Программируемые контроллеры SYSMAC

SQM1/CPM1/CPM1A/SRM1

Руководство по программированию

Cat No. W228-E1-4 R1.20

Программируемые контроллеры CQM1/CPM1/CPM1A/SRM1

Инструкция по программированию

Пересмотрена в июне 1997 года

Cat. No. W228-E1-4 (R1.20)

Перед тем, как читать инструкцию:

Продукция фирмы OMRON создана для использования согласно разрешенным процедурама 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 прописной буквы. Слово “Блок” также пишется с большой 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,1993

Все права сохранены. Никакую часть данного документа нельзя размножать, 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 несется. 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.

О данной инструкции:

В данной инструкции описано программирование программируемых контроллеров CQM1/CPM1/CPM1A/SRM1, включая структуру памяти, содержимое памяти, команды релейно-контактных схем и т. д.

Информацию об аппаратной части и о работе с программатором смотрите Инструкцию по работе CQM1 или Инструкцию по работе CPM1;

для работы с SSS см. Инструкцию по работе с SSS: ПК серии C.

Внимательно прочтите данную инструкцию до полного понимания изложенных сведений, прежде чем пытаться программировать и работать с CQM1 и CPM1.

Глава 1 описывает Установочные параметры ПК и связанные с ними функции ПК, включая обработку прерываний и связь. Установочные параметры служат для управления рабочими параметрами ПК.

В Главе 2 дано описание новых возможностей CQM1 и CPM1, включая новые дополнительные команды и новый режим индикации, называемый просмотром фронтов,

В Главе 3 описана структура областей памяти ПК и их использование. Также описаны операции с кассетой памяти для обмена данных между CQM1 и кассетой памяти.

В Главе 4 описаны основные действия и понятия, используемые для написания программы в виде релейно-контактной схемы. Она знакомит с командами, используемыми для построения базовой структуры релейно-контактной схемы и для управления ее выполнением.

В Главе 5 объясняется индивидуально каждая команда программирования в виде релейно-контактной схемы, которые используются в ПК.

В Главе 6 описаны методы и процедуры использования команд связи с верхним уровнем (HOST LINK), которые служат для связи HOST LINK через порты ПК.

В Главе 7 описаны внутренняя работа ПК и время, требуемое для обработки и исполнения. Читайте данную главу для понимания синхронизации операций ПК.

В Главе 8 описано, как диагностировать и исправлять ошибки аппаратуры и программы, которые могут произойти во время работы ПК.

Предупреждение! Если приступить к работе с ПК, не прочитав и не поняв информацию в данной инструкции, это может привести к несчастным случаям или смерти персонала, повреждению или поломке изделия. Перед тем, как пытаться проделать любую процедуру или операцию, полностью прочтите каждую главу до полного понимания.

Содержание

1. Установочные параметры ПК и связанные с ними характеристики 13
1.1 Установочные параметры ПК. 14
1.1.1 Изменение установочных параметров ПК 14
1.1.2 Установочные параметры программируемого контроллера CQM1. 15
1.1.3 Установочные параметры CPM1/CPM1A. 20
1.1.4 Установочные параметры SRM1 23
1.2 Базовые операции ПК и процессы входов/выходов. 26
1.2.1 Режим пуска 26
1.2.2 Состояние бита удержания 27
1.2.3 Защита от записи памяти программ (только для CPM1/CPM1A) 27
1.2.4 Время обслуживания порта RS-232C (только CQM1/SRM1) 27
1.2.5 Время обслуживания периферийного порта 28
1.2.6 Время цикла 28
1.2.7 Константы времени входа 29
1.2.8 Высокоскоростные таймеры (Только CQM1) 30
1.2.9 Количество вводимых цифр для DSW(87) и Метод обновления выходов (Только CQM1) 31
1.2.10 Параметры протокола ошибок. 31
1.3 Функции выдачи импульсов (только CQM1) 34
1.3.1 Типы выдачи импульсов. 34
1.3.2 Выдача стандартных импульсов с простого выхода. 34
1.3.3 Выдача стандартных импульсов с портов 1 и 2 36
1.3.4 Выдача импульсов с переменным коэффициентом заполнения с портов 1 и 2. 44
1.3.5 Определение состояния портов 1 и 2 46
1.4 Функции прерываний CQM1 48
1.4.1 Типы прерываний 48
1.4.2 Входные прерывания 50
1.4.3 Маскирование всех прерываний 55
1.4.4 Прерывания интервального таймера. 55
1.4.5 Прерывания высокоскоростного счетчика (ВСч) 0. 58
1.4.6 Переполнение “+”/ Переполнение “- “ высокоскоростного счетчика 0 64
1.4.7 Прерывания высокоскоростных счетчиков (ВСч) 1 и 2 (CQM1-CPU43-E) 65
1.4.8 Прерывания абсолютного высокоскоростного счетчика (CQM1-CPU44-E). 72
1.5 Функции прерываний CPM1/CPM1A 80
1.5.1 Типы прерываний 80
1.5.2 Входные прерывания 81
1.5.3 Маскирование всех прерываний 86
1.5.4 Прерывания интервальных таймеров 87
1.5.5 Прерывания высокоскоростного счетчика 89
1.6 Функции прерываний SRM1 97
1.6.1 Типы прерываний 97
1.6.2 Прерывания по интервальному таймеру 97
1.7 Функции распределенных входов/выходов CompoBus/S (Только SRM1) 99
1.8 Функции связи 101
1.8.1 Установочные параметры CQM1 101
1.8.2 Подключение портов. 104
1.6.3 Связь Host Link CQM1 104
1.8.4 Связь Host Link CPM1/CPM1A. 106
1.8.5 Связь Host Link SRM1 108
1.8.6 Связь по RS-232C (только CQM1/SRM1) 111
1.8.7 Связь CQM1 1:1 113
1.8.8 Связь 1:1 CPM1/CPM1A 115

1.8.9 Связь CPM1/CPM1A NT LINK . . . 116
1.8.10 Связь 1:1 SRM1 . . . 117
1.8.11 Связь NT LINK в SRM1 . . . 119
1.8.12 Свободный протокол в SRM1 . . . 120
1.8.13 Конфигурация передаваемых данных . . . 122
1.8.14 Флаги передачи . . . 122
1.8.15 Пример программы связи без протокола . . . 122
1.9 Вычисления с двоичными данными со знаком . . . 124
1.9.1 Задание двоичных данных со знаком . . . 124
1.9.2 Арифметические флаги . . . 125
1.9.3 Ввод двоичных чисел со знаком с использованием десятичных значений . . . 125
1.9.4 Использование дополнительных команд двоичных со знаком (только CQM1) . . . 126
1.9.5 Пример применения двоичных чисел со знаком . . . 126
2. Новые возможности . . . 129
2.1 Дополнительные команды (только CQM1/SRM1) . . . 130
2.1.1 Дополнительные инструкции CQM1 . . . 130
2.1.2 Дополнительные инструкции SRM1 . . . 132
2.2 Новые команды входа/выхода (только CQM1) . . . 132
2.2.1 Ввод с клавиатуры на 10 клавиш TKY(18) . . . 132
2.2.2 Ввод с клавиатуры (16 клавиш) HKY(-) . . . 134
2.2.3 Ввод с цифрового переключателя - DSW(87) . . . 137
2.2.4 Вывод на 7-сегментный индикатор - 7SEG(88) . . . 141
2.2.5 Переназначение битов входов/выходов . . . 143
2.3 Функция MACRO . . . 144
2.4 Просмотр фронтов (изменения состояния) . . . 145
2.5 Аналоговые задания (только CQM1-CPU42-E/CPM1/CPM1A) . . . 146
2.6 Входы быстрой реакции (только CPM1/CPM1A) . . . 147
3. Области памяти . . . 149
3.1 Функции области памяти CQM1 . . . 150
3.1.1 Структура областей памяти . . . 150
3.1.2 Область IR . . . 151
3.1.3 Область SR . . . 153
3.1.4 Область TR . . . 153
3.1.5 Область HR . . . 154
3.1.6 Область AR . . . 154
3.1.7 Область LR . . . 154
3.1.8 Область таймеров/ счетчиков . . . 154
3.1.9 Область DM . . . 155
3.1.10 Область UM . . . 155
3.2 Функции областей памяти CPM1/CPM1A . . . 155
3.2.1 Структура области памяти . . . 155
3.2.2 Область IR . . . 156
3.2.3 Область SR . . . 157
3.2.4 Область TR . . . 157
3.2.5 Область HR . . . 158
3.2.6 Область AR . . . 158
3.2.7 Область LR . . . 158
3.2.8 Область таймеров/ счетчиков (TC) . . . 158
3.2.9 Область DM . . . 158
3.3 Функции областей памяти SRM1 . . . 158
3.3.1 Структура областей памяти . . . 158
3.3.2 Область IR . . . 160
3.3.3 Область SR . . . 160
3.3.4 Область TR . . . 160
3.3.5 Область HR . . . 160

3.3.6 Область AR . . . 160
3.3.7 Область LR . . . 161
3.3.8 Область таймеров/ счетчиков . . . 161
3.3.9 Область DM . . . 161
3.4 Флэш-память SRM1 . . . 161
3.5 Использование кассет памяти (только CQM1) . . . 162
3.5.1 Типы кассет памяти и содержание . . . 162
3.5.2 Емкость кассеты памяти и размер области UM . . . 163
3.5.3 Запись в кассету памяти . . . 164
3.5.4 Чтение из кассеты памяти . . . 164
3.5.5 Сравнение содержания кассеты памяти . . . 165
4. Программирование в виде релейно-контактных схем . . . 167
4.1 Основной алгоритм . . . 168
4.2 Терминология команд . . . 168
4.3 Основные релейно-контактные схемы . . . 169
4.3.1 Базовые термины . . . 169
4.3.2 Мнемокод . . . 170
4.3.3 Команды, расположенные на релейно-контактной схеме (РКС) . . . 171
4.3.4 Вывод и Вывод инверсии (OUTPUT и OUTPUT NOT) . . . 174
4.3.5 Команда END . . . 174
4.3.6 Команды логического блока . . . 175
4.3.7 Кодирование нескольких выходных команд . . . 183
4.3.8 Командные линии ветвления . . . 184
4.3.9 Переходы . . . 189
4.4 Управление состоянием битов . . . 191
4.4.1 Установить в 1 (SET) и Сбросить в 0 (RESET) . . . 191
4.4.2 Включить на 1 цикл (DIFFERENTIATE UP и DIFFERENTIATE DOWN) . . . 192
4.4.3 Сохранить (KEEP) . . . 192
4.4.4 Биты самоподдержки . . . 193
4.5 Рабочие биты (внутренние реле) . . . 193
4.6 Обратите внимание при программировании . . . 195
4.7 Исполнение программы . . . 197
5. Набор команд . . . 199
5.1 Система обозначений . . . 200
5.2 Формат команды . . . 200
5.3 Области данных, значения определителей и флаги . . . 200
5.4 Команды, срабатывающие по фронту 0/1 . . . 202
5.5 Кодирование выходных команд (“правосторонних”) . . . 203
5.6 Таблицы команд . . . 205
5.6.1 Функциональные коды CQM1 . . . 205
5.6.2 Функциональные коды CPM1/CPM1A . . . 206
5.6.2 Функциональные коды SRM1 . . . 206
5.6.3 Алфавитный список мнемоники . . . 206
5.7 Команды релейно-контактной схемы . . . 212
5.7.1 LOAD, LOAD NOT, AND, AND NOT, OR и OR NOT . . . 212
5.7.2 AND LOAD и OR LOAD . . . 213
5.8 Команды управления битами . . . 214
5.8.1 OUT и OUT NOT - Вывод и Вывод с инверсией . . . 214
5.8.2 SET и RESET (Установка и сброс) . . . 215
5.8.3 KEEP(11) (KEEP) - Сохранить состояние . . . 216
5.8.4 DIFU(13) и DIFD(14) - установка бита на 1 цикл (после фронта 0/1 и 1/0) . . . 217
5.9 NOP(00) (NO OPERATION) - Нет операции . . . 218

5.10 END (01) (END) - Конец программы 218
5.11 IL(02) и ILC(03) - Секция INTERLOCK и INTERLOCK CLEAR 218
5.12 JMP(04) и JME(05) - Переход и Конец перехода 220
5.13 FAL(06) и FALS(07) - Команды ошибок пользователя 221
5.14 STEP(08) / SNXT(09) - Команды STEP 223
5.15 Команды таймеров и счетчиков 226
5.15.1 TIM - таймер. 227
5.15.2 CNT - Счетчик 227
5.15.3 CNTR(12) - Реверсивный счетчик 229
5.15.4 TIMH(15) - Высокоскоростной таймер 230
5.15.5 STIM(69) - Интервальный таймер 231
5.15.6 CTBL(63) - Зарегистрировать таблицу сравнения 233
5.15.7 INI(61) - Управление режимами выдачи импульсов 238
5.15.8 PRV(62) - Чтение текущего значения высокоскоростного счетчика 239
5.16 Команды сдвига 242
5.16.1 SFT(10) - Сдвиговый регистр 242
5.16.2 WSFT(16) - Сдвиг слова 243
5.16.3 ASL(25) - Арифметический сдвиг влево 244
5.16.4 ASR(26) - Арифметический сдвиг вправо 244
5.16.5 ROL(27) - Циклический сдвиг влево 245
5.16.6 ROR(28) - Циклический сдвиг вправо 245
5.16.7 SLD(74) - Сдвиг влево на одну цифру 246
5.16.8 SRD(75) - Сдвиг вправо на одну цифру 247
5.16.9 SFTR(84) - Реверсивный регистр сдвига 247
5.16.10 ASFT(17) - Асинхронный регистр сдвига 249
5.17 Команды пересылки данных 251
5.17.1 MOV(21) - Пересылка. 251
5.17.2 MVN(22) - Пересылка инверсии 252
5.17.3 XFER(70) - Пересылка блока. 253
5.17.4 BSET(71) - Заполнение блока 254
5.17.5 XCHG(73) - Обмен данными 255
5.17.6 DIST(80) - Распределение одного слова 255
5.17.7 COLL(81) - Сбор данных 257
5.17.8 MOVB(82) - Переслать бит 260
5.17.9 MOVD(83) - Переслать цифру 261
5.17.10 XFRB(-) - Переслать биты 262
5.18 Команды сравнения 264
5.18.1 CMP(20) - Сравнение 264
5.18.2 TCMP(85) - Сравнение таблицы 265
5.18.3 BCMP(68) - Сравнение блока 267
5.18.4 CMPL(60) - Сравнение слов двойной длины 268
5.18.5 MCMP(19) - Сравнение нескольких слов 270
5.18.6 CPS(-) - Сравнение двоичных чисел со знаком 271
5.18.7 CPSL(-) - Сравнение двоичных слов двойной длины со знаком 273
5.18.8 ZSP(-) - Сравнение с зоной 274
5.18.9 ZCPL(-) - Сравнение с зоной чисел двойной длины 276
5.19 Команды преобразования 277
5.19.1 BIN(23) - Преобразование двоично-десятичного числа в двоичное) 277
5.19.2 BCD(24) - Преобразование из двоичного вида в двоично-десятичный 277
5.19.3 BINL(58) - Преобразование двоично-десятичного числа двойной длины в двоичное двойной длины 278
5.19.4 BCDL(59) - Преобразование двоичного числа двойной длины в двоично-десятичное число двойной длины 279
5.19.5 MLPX(76) - Декодер 4-в-16 279
5.19.6 DMPX(77) - Преобразовать 16-в-4 281
5.19.7 SDEC(78) - 7-сегментный декодер 284

5.19.8 ASC(86) - Преобразование в коды ASCII . . . 287
5.19.9 HEX(-) - Преобразование из ASCII в 16-ричное число . . . 288
5.19.10 SCL(66) - Масштабирование . . . 291
5.19.11 SCL2(-) - Преобразование двоичных значений со знаком в двоично-десятичное с масштабированием. . . . 293
5.19.12 SCL3(-) - Масштабирование двоично-десятичных чисел в двоичные со знаком . . . 295
5.19.13 SEC(-) - Преобразование Часы-в-секунды. . . . 297
5.19.14 HMS(-) - Преобразование Секунды-в-часы . . . 298
5.19.15 LINE(-) - Столбец-в-строку . . . 299
5.19.16 COLM(-) - Строка-в-столбец . . . 300
5.19.17 NEG(-) - Дополнение до 2. . . . 302
5.19.18 NEGL(-) - Дополнение до 2 слова двойной длины . . . 303
5.20 Команды двоично-десятичных вычислений . . . 305
5.20.1 STC(40) - Установка флага переноса . . . 305
5.20.2 CLC(41) - Очистка флага переноса . . . 305
5.20.3 ADD(30) - Двоично-десятичное сложение. . . . 305
5.20.4 SUB(31) - Двоично-десятичное вычитание . . . 307
5.20.5 MUL(32) - Двоично-десятичное умножение . . . 309
5.20.6 DIV(33) - Деление двоично-десятичных чисел . . . 310
5.20.7 ADDL(54) - Двоично-десятичное сложение чисел двойной длины . . . 311
5.20.8 SUBL(55) - Двоично-десятичное вычитание чисел двойной длины . . . 313
5.20.9 MULL(56) - Двоично-десятичное умножение чисел двойной длины. . . . 315
5.20.10 DIVL(57) - Двоично-десятичное деление чисел двойной длины . . . 315
5.20.11 ROOT(72) - Квадратный корень . . . 316
5.21 Команды двоичных вычислений . . . 317
5.21.1 ADB(50) - Сложение двоичных чисел . . . 317
5.21.2 SBB(51) - Вычитание двоичных чисел. . . . 319
5.21.3 MLB(52) - Умножение двоичных чисел . . . 320
5.21.4 DVB(53) - Деление двоичных чисел . . . 321
5.21.5 ADBL(-) - Сложение двоичных чисел двойной длины . . . 321
5.21.6 SBBL(-) - Вычитание двоичных чисел двойной длины . . . 323
5.21.7 MBS(-) - Умножение двоичных чисел со знаком . . . 325
5.21.8 MBSL(-) - Умножение двоичных чисел двойной длины. . . . 326
5.21.9 DBS(-) - Деление двоичных чисел со знаком . . . 327
5.21.10 DBSL(-) - Деление двоичных чисел двойной длины со знаком . . . 328
5.22 Специальные математические команды . . . 330
5.22.1 MAX(-) - Найти максимум. . . . 330
5.22.2 MIN(-) - Найти минимум . . . 331
5.22.3 AVG(-) - Среднее значение. . . . 332
5.22.4 SUM(-) - Сумма . . . 334
5.22.5 APR(-) - Математические вычисления. . . . 336
5.23 Логические команды . . . 341
5.23.1 COM(29) - Дополнение . . . 341
5.23.2 ANDW(34) - Логическое И . . . 341
5.23.3 ORW(35) - Логическое ИЛИ . . . 342
5.23.4 XORW(36) - Исключающее ИЛИ . . . 343
5.23.5 XNRW(37) - Исключающее ИЛИ - НЕ . . . 343
5.24 Команды инкрементирования/ декрементирования . . . 345
5.24.1 INC(38) - Инкремент двоично-десятичного числа . . . 345
5.24.2 DEC(39) - Декремент двоично-десятичного числа . . . 345
5.25 Команды подпрограмм . . . 346
5.25.1 SBS(91) - Войти в подпрограмму . . . 346
5.25.2 SBN(92) - Начало подпрограммы и RET(93) - возврат . . . 347
5.26 Специальные команды . . . 349
5.26.1 TRSM(45) - Трассировка выборки памяти. . . . 349

5.26.2 MSG(46) - Индикация сообщения . . . 350
5.26.3 IORF(97) - Обновление входов/выходов . . . 352
5.26.4 MCRO(99) - Макро . . . 352
5.26.5 BCNT(67) - Счетчик битов . . . 353
5.26.6 FCS(-) - Контрольная сумма кадра . . . 354
5.26.7 FPD(-) - Поиск точки сбоя . . . 356
5.26.8 INT(89) - Управление прерываниями . . . 360
5.26.9 PULS(65) - Задание импульсов . . . 364
5.26.10 SPED(64) - Запуск выдачи импульсов . . . 365
5.26.11 PLS2(-) - Выдача импульсов . . . 368
5.26.12 ACC(-) - Управление ускорением . . . 369
5.26.13 PWM(-) - Импульсы с переменным коэффициентом заполнения . . . 372
5.26.14 SRCH(-) - Поиск данных . . . 373
5.26.15 PID(-) - ПИД-регулирование(пропорционально-интегрально-дифференциальное) . . . 374
5.27 Команды связи . . . 377
5.27.1 RXD(47) - Прием . . . 377
5.27.2 TXD(48) - Передача . . . 378
5.27.3 STUP Изменение установочных параметров RS-232C . . . 380
5.28 Дополнительные команды ввода/вывода . . . 383
5.28.1 7SEG(88) - Вывод на 7-сегментный индикатор . . . 383
5.28.2 DSW(87) - Ввод с цифрового переключателя . . . 384
5.28.3 HKY(-) - Ввод с клавиатуры на 16 клавиш . . . 384
5.28.4 TKY(18) - Ввод с клавиатуры на 10 клавиш . . . 385
6. Команды Host Link . . . 387
6.1 Процедура связи . . . 388
6.2 Форматы команд и ответов . . . 390
6.2.1 Команды с управляющего компьютера . . . 390
6.2.2 Команды от ПК (только CQM1) . . . 394
6.2.3 Коды окончания ответа . . . 394
6.3 Команды Host Link (связи с верхним уровнем) . . . 394
6.3.1 Чтение области IR/SR - RR . . . 395
6.3.2 Чтение области LR - RL . . . 395
6.3.3 Чтение области HR - RH . . . 396
6.3.4 Чтение текущего значения - RC . . . 396
6.3.5 Чтение состояния TC - RG . . . 397
6.3.6 Чтение области DM - RD . . . 397
6.3.7 Чтение области AR - RJ . . . 398
6.3.8 Запись в области IR/SR - WR . . . 398
6.3.9 Запись в область LR - WL . . . 399
6.3.10 Запись в область HR - WH . . . 399
6.3.11 Запись текущего значения - WC . . . 400
6.3.12 Запись состояния Т/Сч - WG . . . 401
6.3.13 Запись в область DM - WD . . . 401
6.3.14 Запись в область AR - WJ . . . 402
6.3.15 Чтение 1 заданного значения - R# . . . 403
6.3.16 Чтение 2 заданного значения - R$ . . . 403
6.3.17 Чтение 3 заданного значения - R% (Только CQM1) . . . 404
6.3.18 Изменение заданного значения (первое)- W# . . . 405
6.3.19 Изменение 2 заданного значения - W$ . . . 406
6.3.20 Изменение 3 заданного значения - W% . . . 407
6.3.21 Чтение состояния - MS . . . 408
6.3.22 Запись состояния - SC . . . 409
6.3.23 Чтение признаков ошибок - MF . . . 410
6.3.24 Принудительная установка - KS . . . 411
6.3.25 Принудительный сброс - KR . . . 412
6.3.26 Принудительные сброс/установка нескольких адресов - FK . . . 413

6.3.27 Отмена принудительных установки/сброса - KC . . . 414
6.3.28 Читать модели ПК - MM . . . 415
5.3.29 Тест - TS . . . 415
5.3.30 Чтение программы - RP . . . 416
6.3.31 Запись программы - WP . . . 416
6.3.32 Составная команда - QQ . . . 417
6.3.33 Прервать - XZ . . . 419
6.3.34 Инициировать - ** . . . 419
6.3.35 Неопределенная команда - IC . . . 420
7. Операции ПК и распределение времени . . . 421
7.1 Время цикла CQM1 и время реакции на вход . . . 422
7.1.1 Цикл CQM1 . . . 422
7.1.2 Время цикла CQM1 . . . 423
7.1.3 Время реакции на вход . . . 425
7.1.4 Время реакции на вход при связи 1+1 . . . 426
7.1.5 Время обработки прерываний . . . 428
7.1.6 Время исполнения команд CQM1 . . . 430
7.2 Время цикла CPM1/CPM1A и время реакции входов/выходов . . . 443
7.2.1 Цикл CPM1/CPM1A . . . 443
7.2.2 Время цикла CPM1/CPM1A . . . 443
7.2.3 Время реакции на вход . . . 445
7.2.4 Время реакции на вход при связи 1:1 . . . 446
7.2.5 Время обработки прерываний . . . 448
7.2.6 Время исполнения команд CPM1/CPM1A . . . 449
7.3 Время цикла SRM1 и время реакции входов/выходов . . . 456
7.3.1 Цикл SRM1 . . . 456
7.3.2 Время цикла SRM1 . . . 456
7.3.3 Время реакции вход/выход . . . 458
7.3.4 Время реакции вход/выход при связи 1:1 . . . 459
7.3.5 Время обработки прерываний . . . 460
7.3.6 Время исполнения команд SRM1 . . . 461
8. Поиск неисправностей . . . 467
8.1 Введение . . . 468
8.2 Неисправности при работе с программатором . . . 468
8.3 Ошибки программирования . . . 469
8.4 Ошибки, определяемые пользователем . . . 470
8.5 Ошибки при работе . . . 471
8.5.1 Нефатальные ошибки . . . 471
8.5.2 Фатальные ошибки . . . 473
8.6 Протокол ошибок . . . 474
8.7 Ошибки HOST LINK . . . 476
8.8 Алгоритмы поиска неисправностей . . . 478
8.8.1 Алгоритмы для CPM1 . . . 478
8.8.1 Алгоритмы для CPM1A . . . 478
8.8.1 Алгоритмы для SRM1 . . . 478
8.8.2 Алгоритмы для CQM1 . . . 478
Приложение A . . . 485
Команды программирования ЛД . . . 485
Дополнительные команды . . . 489
Приложение B . . . 493
Операции с флагами ошибок и арифметики . . . 493
Дополнительные команды (Все CQM1/SRM1) . . . 494
Дополнительные команды (Только CQM1-CPU4_-EV1) . . . 495

Приложение C 497
Области памяти 497
Функции областей памяти CQM1 497
Структура области памяти 497
Область SR 498
Пояснения к битам SR 501
Область AR 502
Функции областей памяти CPM1/CPM1A 506
Структура области памяти 506
Область SR 506
Область AR 509
Функции областей памяти SRM1 511
Структура областей памяти 511
Область SR 512
Область AR 513
Приложение D 517
Применение функции часов 517
Слова , используемые с функцией часов 517
Установка времени 517
Приложение E 519
Распределение входов/выходов 519
Приложение F 521
Мнемокод программы 521
Приложение G 523
Мнемокод программы 523
Приложение H 525
Расширенные коды ASCII 525
Глоссарий 527

1. Установочные параметры ПК и связанные с ними характеристики

В данной главе описаны установочные параметры ПК и связанные с ними функции CQM1/CPM1/CPM1A/SRM1, включая обработку прерываний и связь. Установочные параметры служат для управления оперативными параметрами ПК. Для изменения установочных параметров смотрите:

для процедур программатора - Инструкцию по работе CQM1, Инструкцию по работе CPM1, Инструкцию по работе CPM1A, Инструкцию по работе SRM1;

для процедур SSS - Инструкцию по работе с SSS: ПК серии C.

Если Вы не знакомы с ПК OMRON или с релейно-контактным программированием, Вы можете прочитать п. 1-1 как общий обзор имеющихся параметров CQM1/CPM1/CPM1A/SRM1, но для полного понимания данной главы Вам, вероятно, понадобиться прочесть гл. 3, гл. 4 и в гл. 5 описание команд, встречающихся в гл. 1.

1.1 Установочные параметры ПК

К установочные параметрам ПК относятся различные рабочие параметры, которые управляют работой CQM1/CPM1/CPM1A/SRM1. Для максимального использования функциональных возможностей CQM1/CPM1/CPM1A/SRM1 при использовании функций обработки прерываний и связи установочные параметры должны быть откорректированы под конкретного пользователя согласно условиям работы.

При отгрузке с завода установлены параметры по умолчанию для общих условий работы, так что CQM1/CPM1/CPM1A/SRM1 можно использовать и без изменения установочных параметров. Однако перед работой рекомендуется проверить установленные значения по умолчанию.

Значения по умолчанию

Значения по умолчанию установочных параметров ПК = 0000 для всех слов. Значения установочных параметров можно в любой момент сбросить, включив параметр SR 25210 в 1.

Внимание! При очистке области памяти DM с программирующего устройства, все установочные параметры также сбросятся в 0.

1.1.1 Изменение установочных параметров ПК

Установочные параметры доступны в различные моменты времени в зависимости от этих параметров следующим образом:

- DM 6600..DM 6614: Доступны только при включении ПК.
- DM 6615..DM 6644: Доступны только в начале исполнения программы.
- DM 6645..DM 6655: Доступны постоянно при включенном ПК.

Поскольку изменения в параметрах вступают в силу только в указанных случаях, для того, чтобы изменения в параметрах DM 6600..DM 6614 вступили в силу, нужно перезапустить ПК, для того, чтобы изменения в параметрах DM 6615..DM 6644 вступили в силу, нужно перезапустить исполнение программы.

Изменение параметров с периферийного устройства

Установочные параметры можно читать, но не записывать, из программы пользователя. Писать можно только с программирующего устройства.

Хотя установочные параметры хранятся в DM 6600..DM 6655, их можно задавать и изменять только с программирующего устройства (например SSS или программатора). DM 6600..DM 6644 можно задавать или изменять только в режиме PROGRAM. DM 6645..DM 6655 можно задавать или изменять в режимах PROGRAM или MONITOR.

Далее приведены установки параметров, которые можно произвести в режиме PROGRAM с использованием операций меню SSS. Все другие установки нужно делать с использованием операций над 16-ричными числами.

- Режим при пуске (DM 6600)
- Состояние бита удержания входа/выхода и состояние бита удержания принудительного состояния (DM 6601)
- Время контроля цикла (DM 6618)
- Время цикла (DM 6619)
- Установочные параметры порта RS-232C (DM 6645..DM 6649)

Замечание Параметры порта RS-232C (DM 6645..DM 6649) не используются в CPM1/CPM1A, поскольку на нем нет порта RS-232C.

Ошибки в установочных параметрах ПК

При доступе к некорректно заданному установочному параметру будет генерироваться признак нефатальной ошибки (код ошибки 9B), включится в 1 соответствующий флаг ошибки (AR 2400..AR 2402 для CQM1 и AR 1300..AR 1302 для CPM1/CPM1A), вместо некорректного значения будет использоваться значение по умолчанию.

1.1.2 Установочные параметры программируемого контроллера CQM1

Установочные параметры программируемого контроллера делятся на 4 категории:

- Относящиеся к базовым операциям CQM1 и процессам входов/ выходов;
- Относящиеся к функциям выдачи импульсов;
- Относящиеся к прерываниям;
- Относящиеся к связи.

В данной главе Установочные параметры будут описаны в соответствии с этой классификацией.

В следующей таблице установочные параметры представлены в порядке их расположения в области DM. Подробное описание параметра смотрите в пункте, указанном в правом столбце.

Слово (а)	Бит(ы)	Функция	Пункт
Обработка пуска (DM 6600 - DM 6614) Введенные в ПК значения параметров вступают в силу только после перезапуска ПК.			
DM 6600	00 - 07	Режим пуска (действует, когда биты 08..15 = 2). 00: PROGRAM 01: MONITOR 02: RUN	1.2.1
	08 - 15	Задание режима пуск 00: С переключателя программатора 01: Продолжить работу в режиме, который был перед отключением питания 02: Из битов 00..07 данного параметра.	
DM 6601	00 - 07	Резерв (= 00)	
	08 - 11	Состояние бита сохранения входов/выходов (SR 25212) 0: Сброшен 1: Установлен	1.2.2
	12 - 15	Состояние бита удержания принудительной установки (SR 25211) 0: Сброшен 1: Установлен	
DM 6602 - DM 6610	00 - 15	Резерв	
DM 6611	00 - 15	CQM1-CPU43-EV1: Режим портов 1 и 2 0000: режим высокоскоростных счетчиков; 0001: Режим выдачи импульсов CQM1-CPU44-EV1: Смещение исходного для порта 1 (4 двоично-десятичных цифры)	1.3.3, 1.4.8
DM 6612	00 - 15	CQM1-CPU44-EV1: Смещение исходного для порта 2 (4 двоично-десятичных цифры)	1.4.8
Настройка параметров выдачи импульсов и времени цикла (DM 6615 - DM 6619) Введенные в ПК значения параметров вступают в силу при пуске программы.			
DM 6615	00 - 07	Слово для выдачи импульсов 00: IR 100; 01: IR 101; 02: IR 102 ..... 15: IR 115	1.3.2
	08 - 15	Резерв (= 0)	
DM 6616	00 - 07	Время обслуживания порта RS-232C (действуют, когда биты 08..15 = 01) 00..99:(двоично-десятичные цифры): процент от времени цикла на обслуживание порта RS-232C .	1-2-4
	08 - 15	Задание времени обслуживания порта RS-232C из параметров 00: 5% от времени цикла 01: время, заданное битами 00 - 07	
DM 6617	00 - 07	Время обслуживания периферийного порта (действуют, когда биты 08..15 = 01) 00..99:(двоично-десятичные цифры): процент от времени цикла.	1-2-5
	08 - 15	Задание времени обслуживания периферийного порта из параметров 00: 5% от времени цикла 01: время, заданное битами 00 - 07	

Слово (а)	Бит(ы)	Функция	Пункт
DM 6618	00 - 07	Время контроля цикла (действует, когда биты 08..15 = 01, 02 или 03) 00 - 99 (двоично-десятичные цифры): Задание (см. биты 08 - 15)	1.2.10
	08 - 15	Разрешение контроля за циклом (задание в битах 00 - 07 х 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	
DM 6619	00 - 15	Время цикла 0000: Непостоянное (минимальное не задано) 0001 - 9999 (двоично-десятичные цифры): минимальное время в мс	1.2.6
Обработка прерываний (DM 6620 - DM 6639) Введенные в ПК значения параметров вступают в силу при пуске программы.			
DM 6620	00 - 03	Выдержка времени входа IR 00000..IR 00007 0: 8 мс; 1: 1 мс; 2: 2 мс; 3: 4 мс; 4: 8 мс; 5:16 мс; 6: 32 мс; 7: 64 мс; 8: 128 мс;	1.2.7
	04 - 07	Выдержка времени входа IR 00008 - IR 00015 (значения те же, что и для бит 00 - 03)	
	08 - 15	Выдержка времени входа IR 001 00: 8 мс; 01: 1 мс; 02: 2 мс; 03: 4 мс; 04: 8 мс; 05: 16 мс; 06: 32 мс; 07: 64 мс; 08: 128 мс;	
DM 6621	00 - 07	Выдержка времени входа IR 002 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 003 (значения те же, что и для IR 001)	
DM 6622	00 - 07	Выдержка времени входа IR 004 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 005 (значения те же, что и для IR 001)	
DM 6623	00 - 07	Выдержка времени входа IR 006 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 007 (значения те же, что и для IR 001)	
DM 6624	00 - 07	Выдержка времени входа IR 008 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 009 (значения те же, что и для IR 001)	
DM 6625	00 - 07	Выдержка времени входа IR 0010 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 011 (значения те же, что и для IR 001)	
DM 6626 - DM 6627	00 - 15	Резерв	
DM 6628	00 - 03	Разрешение прерывание для IR 00000 (0: нормальный вход; 1: вход с прерыванием)	1.4.2
	04 - 07	Разрешение прерывание для IR 00001 (0: нормальный вход; 1: вход с прерыванием)	
	08 - 11	Разрешение прерывание для IR 00002 (0: нормальный вход; 1: вход с прерыванием)	
	12 - 15	Разрешение прерывание для IR 00003 (0: нормальный вход; 1: вход с прерыванием)	
DM 6629	00 - 07	Число высокоскоростных счетчиков для обновления входов по прерываниям 00..15:(двоично-десятичные цифры; например, задайте 15 для 00..14)	1.2.8
	08 - 15	Разрешение прерываний высокоскоростных таймеров 00: 16 таймеров (задание бит 00..07 запрещено) 01: использование задание бит 00..07	
DM 6630	00 - 07	Первое слово обновления входов для входного прерывания 0: 00..11 (двоично-десятичные цифры)	1.4.2
	08 - 15	Число слов обновления входов для входного прерывания 0: 00..12 (двоично-десятичные цифры)	
DM 6631	00 - 07	Первое слово обновления входов для входного прерывания 1: 00..11 (двоично-десятичные цифры)	
	08 - 15	Число слов обновления входов для входного прерывания 1: 00..12 (двоично-десятичные цифры)	

Слово (а)	Бит(ы)	Функция	Пункт
DM 6632	00 - 07	Первое слово обновления входов для входного прерывания 2: 00..11 (двоично-десятичные цифры)	
	08 - 15	Число слов обновления входов для входного прерывания 2: 00..12 (двоично-десятичные цифры)	
DM 6633	00 - 07	Первое слово обновления входов для входного прерывания 3: 00..11 (двоично-десятичные цифры)	
	08 - 15	Число слов обновления входов для входного прерывания 3: 00..12 (двоично-десятичные цифры)	
DM 6634	00 - 07	Первое слово обновления входов для высокочастотного счетчика 1: 00..11 (двоично-десятичные цифры)	1.4.8
	08 - 15	Число слов обновления входов для высокочастотного счетчика 1: 00..12 (двоично-десятичные цифры)	
DM 6635	00 - 07	Первое слово обновления входов для высокочастотного счетчика 2: 00..11 (двоично-десятичные цифры)	
	08 - 15	Число слов обновления входов для высокочастотного счетчика 2: 00..12 (двоично-десятичные цифры)	
DM 6636	00 - 07	Первое слово обновления входов для интервального таймера 0: 00..07 (двоично-десятичные цифры)	1.4.4 1.4.5
	08 - 15	Число слов обновления входов для интервального таймера 0: 00..08 (двоично-десятичные цифры)	
DM 6637	00 - 07	Первое слово обновления входов для интервального таймера 1: 00..07 (двоично-десятичные цифры)	
	08 - 15	Число слов обновления входов для интервального таймера 1: 00..08 (двоично-десятичные цифры)	
DM 6638	00 - 07	Первое слово обновления входов для интервального таймера 2 (также используется для высокоскоростного счетчика 0): 00..07 (двоично-десятичные цифры)	
	08 - 15	Число слов обновления входов для интервального таймера 2: 00..08 (двоично-десятичные цифры) (также используется для высокоскоростного счетчика 0)	
DM 6639	00 - 07	Метод обновления выходов 00: циклический; 01: прямой	1.2.9 7.1.1
	08 - 15	Количество цифр для команды ЦИФРОВОЙ ПЕРЕКЛЮЧАТЕЛЬ (DSW(87)) 00: 4 цифры; 01: 8 цифр	1.2.9 2.2.3
Параметры высокоскоростных счетчиков (DM 6640 - DM 6644) Введенные в ПК значения параметров вступают в силу при пуске программы.			
DM 6640 - DM 6641	00 - 15	Резерв	
DM 6642	00 - 03	Режим высокоскоростного счетчика 0 0: режим плюс/минус; 4: режим инкремента	1.4.5
	04 - 07	Режим сброса высокоскоростного счетчика 0 0: сброс фазой Z и системной программой; 1: сброс только системной программой	
	08 - 15	Разрешение высокоскоростного счетчика 0 00: не используйте Всч 01: ВСч используется с заданиями 00..07	
DM 6643	00 - 03	CQM1-CPU43-E: Настройка входа порта 1 0: вход разности фаз; 1: вход импульс/направление; 2: вход плюс/минус CQM1-CPU44-E: Настройка входа порта 1 0: вход 8 бит; 1: вход 10 бит; 2: вход 12 бит	1.4.7 1.4.8

Слово (а)	Бит(ы)	Функция	Пункт
	04 - 07	CQM1-CPU43-E: Настройка сброса порта 1 0: сброс фазой Z и системной программой; 1: сброс только системной программой CQM1-CPU44-E: Не используется. Установлен в 0.	1.4.7
	08 - 11	CQM1-CPU43-E: Настройка метода счета порта 1 0: линейный метод; 1: циклический метод CQM1-CPU44-E: Настройка режима порта 1 0: режим двоично-десятичный; 1: режим 360°.	1.4.7 1.4.8
	11 - 15	CQM1-CPU43-E: Настройка типа импульсов порта 1 0: Выдача стандартных импульсов (коэффициент заполнения = 0.5); 1: Выдача импульсов с переменным коэффициентом заполнения. CQM1-CPU44-E: Не используется. Установлен в 0.	1.3.3, 1.3.4
DM 6644	00 - 15	Настройка порта 2 (идентична порту 1 в DM 6643)	
Установочные параметры порта RS-232C Следующие параметры вступают в силу после передачи в ПК			
DM 6645	00 - 07	Задание порта 00: стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: заданное в DM 6646	1.6.1
	08 - 11	Слова связи (действуют, когда биты 12..15 =3) 0: LR 00 - LR 63, 1: LR 00 - LR 31; 2: LR 00 - LR 15.	
	12 - 15	Режим связи 0 связь с верхним уровнем (Host Link) 1 RS-232C (без протокола) 2: связь 1+1 (ведомый) 3 связь 1+1 (ведущий)	
DM 6646	00 - 07	Скорость связи 00: 1.2 К, 01: 2.4 К, 02: 4.8 К, 03: 9.6 К, 04: 19.2 К	
	08 - 15	Формат кадра 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 бита / четн. 10: 1 бит / 8 бит / 2 бита / нечетн. 11: 1 бит / 8 бит / 2 бита / нет	
DM 6647	00 - 15	Задержка передачи (Host Link) 0000..9999 (двоично-десятичные цифры): задается в единицах 10 мс. Напр. задание 0001 = 1 мс.	
DM 6648	00 - 07	Номер узла (Host link, действуют, когда биты DM 6645 12 - 15=0) 00..31 (двоично-десятичные цифры)	
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен 1: задан	
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: разрешен (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	

Слово (а)	Бит(ы)	Функция	Пункт
DM 6649	00 - 07	Код старта (RS-232C) 00.. FF (двоичное число)	1.6.1
	08 - 15	Когда биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01..FF: 1..256 байт Когда биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00 .. FF: (двоичное число)	
Установочные параметры периферийного порта Следующие параметры вступают в силу после передачи в ПК. Данные УППК действуют при применении кабеля CQM1-CIF02. Данные УППК не действуют, когда применяется кабель CQM1-CIF11 или программатор.			
DM 6650	00 - 07	Задание порта 00: стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: заданное в DM 6646	1.6.1
	08 - 11	Резерв	
	12 - 15	Режим связи 0: Host Link 1: RS-232C	1.6.1
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	
	08 - 15	Формат кадра 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 бита / четн. 10: 1 бит / 8 бит / 2 бита / нечетн. 11: 1 бит / 8 бит / 2 бита / нет	
DM 6652	00 - 15	Задержка передачи (Host Link) 0000..9999 (в мс)	
DM 6653	08 - 07	Номер узла (Host link, действуют, когда биты DM 6650 12..15 =0) 00..31 (двоично-десятичные цифры)	
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6650 12..15 =1) 0: не разрешен 1: установлен	
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6650 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: задан (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	
DM 6654	00 - 07	Код старта (RS-232C, действует, когда биты DM 6653 08..11 =1) 00 .. FF (двоичное число)	
	08 - 15	Когда биты 12..15 DM 6653 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01.. FF: 1..255 байт Когда биты 12..15 DM 6653 = 1: Код окончания (RS-232C) 00.. FF (двоичное число)	
Установочные параметры протокола ошибок (DM 6655) Следующие параметры вступают в силу после передачи в ПК.			

Слово (a)	Бит(ы)	Функция	Пункт
DM 6655	00 - 03	Стиль 0: сдвиг после сохранения 10 записей 1: сохранение только 10 записей (без сдвига) 2.. F: не сохранять записи	1.2.10
	04 - 07	Резерв	
	08 - 11	Разрешение контроля за временем цикла 0: обнаруживать длинные циклы, как нефатальные ошибки 1: не обнаруживать длинные циклы	1.2.10
	12 - 15	Разрешение предупреждения о неисправности аккумулятора 0: обнаруживать падение напряжения аккумулятора как нефатальную ошибку. 1: не обнаруживать падение напряжения аккумулятора.	

1.1.3 Установочные параметры CPM1/CPM1A

Установочные параметры ПК делятся на 4 категории:

- Относящиеся к базовым операциям ПК и обработке входов/ выходов;
- Относящиеся к времени цикла
- Относящиеся к прерываниям
- Относящиеся к связи.

В данной главе установочные параметры приведены согласно этой классификации.

В таблице установочные параметры CPM1/CPM1A представлены в порядке их расположения в области DM. Подробное описание параметра Вы можете найти в пункте, указанном в правом столбце.

Слово (a)	Бит(ы)	Функция	Пункт
Обработка пуска (DM 6600 - DM 6614) Введенные в ПК значения параметров вступают в силу только после перезапуска ПК.			
DM 6600	00 - 07	Режим пуска (действует, когда биты 08 - 15 = 2). 00: PROGRAM 01: MONITOR 02: RUN	1.2.1
	08 - 15	Режим запуска 00: переключатель программатора 01: продолжить работу в режиме перед отключением питания 02: задание в битах 00 - 07 данного параметра.	
DM 6601	00 - 07	Резерв (= 00)	
	08 - 11	Состояние бита сохранения входов/выходов (SR 25212) 0: сброшен 1: установлен	1.2.2
	12 - 15	Состояние бита удержания принудительной установки (SR 25211) 0: сброшен 1: установлен	
DM 6602	00 - 03	Защита памяти от записи 0: память программ не защищена 1: память программ защищена от записи (за исключением самого DM 6602)	1.2.3
	04 - 07	Язык программатора 0: английский 1: японский	
	08 - 15	Резерв	
DM 6603 - DM 6614	00 - 15	Резерв	
Настройка времени цикла (DM 6615 - DM 6619) Введенные в ПК значения параметров вступают в силу при пуске программы.			
DM 6615 - DM 6616	00 - 15	Резерв	

Слово (а)	Бит(ы)	Функция	Пункт
DM 6617	00 - 07	Время обслуживания периферийного порта (действуют, когда биты 08..15 = 01) 00..99 (двоично-десятичные цифры): процент от времени цикла.	1.2.5
	08 - 15	Задание времени обслуживания периферийного порта из параметров 00: 5% от времени цикла 01: время, заданное битами 00 - 07	
DM 6618	00 - 07	Время контроля цикла (действует, когда биты 08 - 15 = 01, 02 или 03) 00 - 99 (двоично-десятичные цифры): Задание (см. биты 08 - 15)	1.2.10
	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	
DM 6619	00 - 15	Время цикла 0000: Непостоянное (минимальное не задано) 0001.. 9999 (двоично-десятичные цифры): минимальное время в мс	1.2.6
Обработка прерываний (DM 6620 - DM 6639) Введенные в ПК значения параметров вступают в силу при пуске программы.			
DM 6620	00 - 03	Выдержка времени входов IR 00000 - IR 00002 0: 8 мс; 1: 1 мс; 2: 2 мс; 3: 4 мс; 4: 8 мс; 5: 16 мс; 6: 32 мс; 7: 64 мс; 8: 128 мс;	1.2.7
	04 - 07	Выдержка времени входов IR 00003 - IR 00004 (значения те же, что и для бит 00 - 03)	
	08 - 11	Выдержка времени входов IR 00005 - IR 00006 (значения те же, что и для бит 00 - 03)	
	12 - 15	Выдержка времени входов IR 00007 - IR 00011 (значения те же, что и для бит 00 - 03)	
DM 6621	00 - 07	Выдержка времени входов IR 001 00: 8 мс; 01: 1 мс; 02: 2 мс; 03: 4 мс; 04: 8 мс; 05: 16 мс; 06: 32 мс; 07: 64 мс; 08: 128 мс;	
	08 - 15	Выдержка времени входов IR 002 (значения те же, что и для IR 001)	
DM 6622	00 - 07	Выдержка времени входов IR 003 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 004 (значения те же, что и для IR 001)	
DM 6623	00 - 07	Выдержка времени входа IR 005 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 006 (значения те же, что и для IR 001)	
DM 6624	00 - 07	Выдержка времени входа IR 007 (значения те же, что и для IR 001)	
	08 - 15	Выдержка времени входа IR 008 (значения те же, что и для IR 001)	
DM 6625	00 - 07	Выдержка времени входа IR 009 (значения те же, что и для IR 001)	
	08 - 15	Резерв	
DM 6626 - DM 6627	00 - 15	Резерв	
DM 6628	00 - 03	Разрешение прерывание для IR 00000 (0: нормальный вход; 1: вход с прерыванием)	1.4.2
	04 - 07	Разрешение прерывание для IR 00001 (0: нормальный вход; 1: вход с прерыванием)	
	08 - 11	Разрешение прерывание для IR 00002 (0: нормальный вход; 1: вход с прерыванием)	
	12 - 15	Разрешение прерывание для IR 00003 (0: нормальный вход; 1: вход с прерыванием)	
DM 6629 - DM 6641	00 - 15	Резерв	
Параметры высокоскоростных счетчиков (DM 6640 - DM 6644) Введенные в ПК значения параметров вступают в силу при пуске программы.			
DM 6640 - DM 6641	00 - 15	Резерв	

Слово (а)	Бит(ы)	Функция	Пункт
DM 6642	00 - 03	Режим высокоскоростного счетчика 0 0: режим плюс/минус 4: режим инкремента	1.4.5
	04 - 07	Режим сброса высокоскоростного счетчика 0 0: Сброс фазой Z и системной программой 1: Сброс только системной программой	
	08 - 15	Разрешение высокоскоростного счетчика 0 00: не используйте Всч 01: ВСч используется с заданиями 00 - 07	
DM 6643 - DM 6644	00 - 15	Резерв	
Параметры периферийного порта Значения параметров вступают в силу после ввода в ПК.			
DM 6645 - DM 6649	00 - 15	Резерв	
DM 6650	00 - 07	Задание порта 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Заданное в DM 6651 (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	1.6.4
	08 - 11	Область связи для связи 1+1 через периферийный порт 0: LR 00 - LR 15.	
	12 - 15	Режим связи 0: Host Link, 2: связь 1+1 (ведомый), 3: связь 1+1 (ведущий) 4: связь NT (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	
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	
	08 - 15	Формат кадр 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 бита четн. 10: 1 бит 8 бит 2 бита нечетн. 11: 1 бит 8 бит 2 бита нет (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	
DM 6652	00 - 15	Задержка передачи (Host Link) 0000..9999: в мс. (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0000)).	
DM 6653	00 - 07	Номер узла (Host Link) 00..31 (двоично-десятичные цифры) (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0000)).	
	08 - 15	Резерв	
DM 6654	00 - 15	Резерв	
Установочные параметры протокола ошибок (DM 6655) Следующие параметры вступают в силу после передачи в ПК.			
DM 6655	00 - 03	Стиль 0: Сдвиг после сохранения 10 записей 1: Сохранение только 10 записей (без сдвига) 2..F: Не сохранять записи	1.2.10

Слово (а)	Бит(ы)	Функция	Пункт
	04 - 07	Резерв	
	08 - 11	Разрешение времени наблюдения за циклом 0: Обнаруживать длинные циклы как нефатальные ошибки 1: Не обнаруживать длинные циклы	1.2.10
	12 - 15	Резерв	

1.1.4 Установочные параметры SRM1

Установочные параметры ПК делятся на 3 категории:

- Относящиеся к базовым операциям ПК и обработке входов/выходов;
- Относящиеся к времени цикла
- Относящиеся к связи.

В данной главе установочные параметры приведены согласно этой классификации.

В таблице установочные параметры SRM1 представлены в порядке их расположения в области DM. Подробное описание параметра Вы можете найти в пункте, указанном в правом столбце.

Слово (а)	Бит(ы)	Функция	Пункт
Обработка пуска (DM 6600 - DM 6614) Введенные в ПК значения параметров вступают в силу только после перезапуска ПК.			
DM 6600	00 - 07	Режим пуска (действует, когда биты 08 - 15 = 2). 00: PROGRAM 01: MONITOR 02: RUN	1.2.1
	08 - 15	Режим запуска 00: переключатель программатора 01: продолжить работу в режиме перед отключением питания 02: задание в битах 00 - 07 данного параметра.	
DM 6601	00 - 07	Резерв (= 00)	
	08 - 11	Состояние бита сохранения входов/выходов (SR 25212) 0: сброшен 1: установлен	1.2.2
	12 - 15	Состояние бита удержания принудительной установки (SR 25211) 0: сброшен 1: установлен	
DM 6602	00 - 03	Защита памяти от записи 0: память программ не защищена 1: память программ защищена от записи (за исключением самого DM 6602)	1.2.3
	04 - 07	Язык программатора 0: английский 1: японский	
	08 - 15	Резерв	
DM 6603	00 - 03	Максимальное число устройств CompoBus/S 0: 32 1: 16	
	04 - 15	Резерв	
DM 6604 - DM 6614	00 - 15	Резерв	
Настройка времени цикла (DM 6615 - DM 6619) Введенные в ПК значения параметров вступают в силу при пуске программы.			
DM 6615	00 - 15	Резерв	
DM 6616	00 - 07	Время обслуживания порта RS-232C (действуют, когда биты 08..15 = 01) 00..99:(двоично-десятичные цифры): процент от времени цикла на обслуживание порта RS-232C .	1-2-4

Слово (а)	Бит(ы)	Функция	Пункт
	08 - 15	Задание времени обслуживания порта RS-232C из параметров 00: 5% от времени цикла 01: время, заданное битами 00 - 07	
DM 6617	00 - 07	Время обслуживания периферийного порта (действуют, когда биты 08..15 = 01) 00..99 (двоично-десятичные цифры): процент от времени цикла.	1.2.5
	08 - 15	Задание времени обслуживания периферийного порта из параметров 00: 5% от времени цикла 01: время, заданное битами 00 - 07	
DM 6618	00 - 07	Время контроля цикла (действует, когда биты 08 - 15 = 01, 02 или 03) 00 - 99 (двоично-десятичные цифры): Задание (см. биты 08 - 15)	1.2.10
	08 - 15	Разрешение контроля за циклом (задание в битах 00 - 07 х 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	
DM 6619	00 - 15	Время цикла 0000: Непостоянное (минимальное не задано) 0001.. 9999 (двоично-десятичные цифры): минимальное время в мс	1.2.6
DM 6620 - DM 6644	00 - 15	Резерв (не использовать)	
Установочные параметры порта RS-232C Следующие параметры вступают в силу после передачи в ПК			
DM 6645	00 - 03	Задание порта 00: стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: заданное в DM 6646	1.6.1
	04 - 07	Контроль CTS: 0: запрещен 1: разрешен	
	08 - 11	Слова связи (действуют, когда биты 12..15 =3) 0: LR 00 - LR 15, другие: не работают	
	12 - 15	Режим связи 0 связь с верхним уровнем (Host Link) 1 RS-232C (без протокола) 2: связь 1+1 (ведомый) 3 связь 1+1 (ведущий)	
DM 6646	00 - 07	Скорость связи 00: 1.2 К, 01: 2.4 К, 02: 4.8 К, 03: 9.6 К, 04: 19.2 К	
	08 - 15	Формат кадра 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 бита / четн. 10: 1 бит / 8 бит / 2 бита / нечетн. 11: 1 бит / 8 бит / 2 бита / нет	
DM 6647	00 - 15	Задержка передачи (Host Link) 0000..9999 (двоично-десятичные цифры): задается в единицах 10 мс. Напр. задание 0001 = 1 мс.	
DM 6648	00 - 07	Номер узла (Host link, действуют, когда биты DM 6645 12 - 15=0) 00..31 (двоично-десятичные цифры)	

Слово (а)	Бит(ы)	Функция	Пункт
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен 1: задан	
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: разрешен (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	
DM 6649	00 - 07	Код старта (RS-232C) 00.. FF (двоичное число)	1.6.1
	08 - 15	Когда биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01..FF: 1..256 байт Когда биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00 .. FF: (двоичное число)	
Параметры периферийного порта Значения параметров вступают в силу после ввода в ПК.			
DM 6650	00 - 07	Задание порта 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Заданное в DM 6651 (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	1.6.4
	08 - 11	Резерв	
	12 - 15	Режим связи 0: Host Link, 2: свободный протокол (другие значения вызовут признак нефатальной ошибки)	
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	
	08 - 15	Формат кадр 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 бита, четн. 10: 1 бит, 8 бит, 2 бита, нечетн. 11: 1 бит, 8 бит, 2 бита, нет (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	
DM 6652	00 - 15	Задержка передачи (Host Link) 0000..9999: в мс. (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0000)).	
DM 6653	00 - 07	Номер узла (Host link, действуют, когда биты DM 6645 12 - 15=0) 00..31 (двоично-десятичные цифры)	
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен 1: задан	

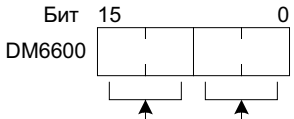
Слово (а)	Бит(ы)	Функция	Пункт
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: разрешен (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	
DM 6654	00 - 07	Код старта (RS-232C) 00.. FF (двоичное число)	
	08 - 15	Код окончания Когда биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01..FF: 1..256 байт Когда биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00 .. FF: (двоичное число)	
Установочные параметры протокола ошибок (DM 6655) Следующие параметры вступают в силу после передачи в ПК.			
DM 6655	00 - 03	Стиль 0: Сдвиг после сохранения 10 записей 1: Сохранение только 10 записей (без сдвига) 2..F: Не сохранять записи	1.2.10
	04 - 07	Резерв	
	08 - 11	Разрешение времени наблюдения за циклом 0: Обнаруживать длинные циклы как нефатальные ошибки 1: Не обнаруживать длинные циклы	1.2.10
	12 - 15	Резерв	

1.2 Базовые операции ПК и процессы входов/выходов

В данной главе описываются установочные параметры, относящиеся к базовым операциям ПК и процессам входов/выходов.

1.2.1 Режим пуска

Режим работы ПК, в который он переходит по включению питания, можно задавать следующим образом:



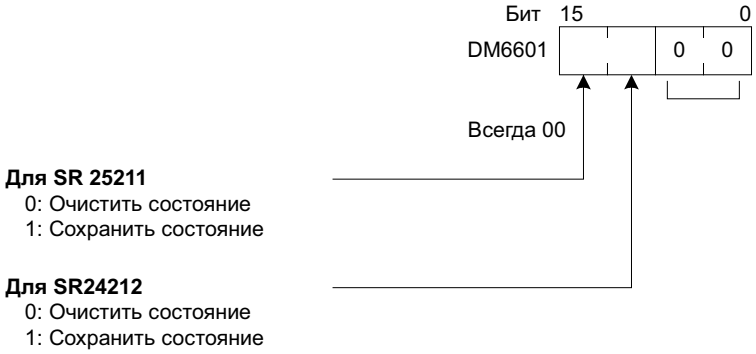
сточник задания при пуске ПК
00: С переключателя режимов на программаторе (если не подключен - режим RUN)
01: Режим, вкотором ПК находился перед отключением
02: Режим, заданный битами 00..07

Режим при пуске: (биты 08..15 действуют только при задании битов 00..07 = 02)
00: Режим PROGRAM
01: Режим MONITOR
02: Режим RUN

По умолчанию: переключатель режимов на программаторе или RUN, если программатор не подключен.

1.2.2 Состояние бита удержания

Установите следующие значения для задания, будет ли при включении питания сохранено значение Бита удержания принудительного состояния (SR 25211) и/или Бита удержания входов/выходов (SR 25212), которое действовало перед выключением питания ПК, или предыдущее состояние будет очищено.



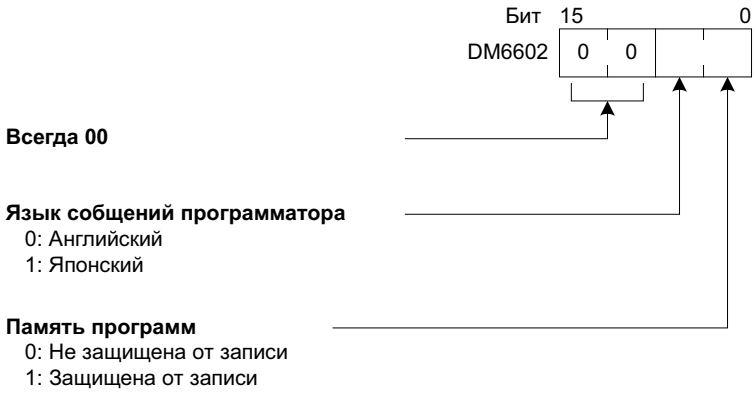
По умолчанию: Очистить оба бита.

Бит удержания принудительного состояния (SR 25211) определяет, будет или нет сохраняться принудительное состояние при переключении режимов с PROGRAM на MONITOR.

Бит удержания входов/выходов (SR 25212) определяет, будет или нет сохраняться состояние битов IR и LR после того, как работа ПК была начата и остановлена.

1.2.3 Защита от записи памяти программ (только для CPM1/CPM1A)

В CPM1/CPM1A память программ можно защитить от записи, установив DM 6602, биты 00..03, в 0. Биты 04..07 определяют язык сообщений ПК (английский или японский).

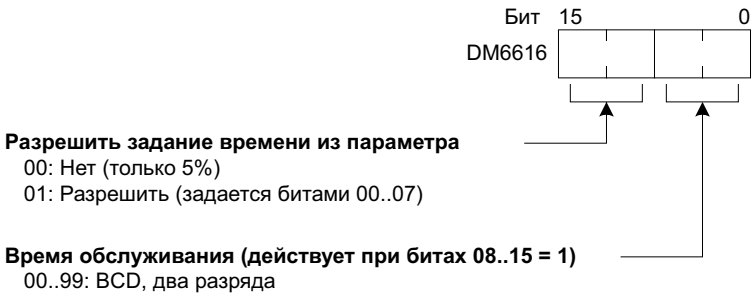


По умолчанию: Английский, не защищена.

Замечание После установки защиты памяти программ от записи (установив биты 04..07 DM 6602 в 1) сам параметр DM 6602 изменять можно.

1.2.4 Время обслуживания порта RS-232C (только CQM1/SRM1)

Служит для задания процента времени цикла, отведенного на обслуживание порта RS-232C.



По умолчанию: 5% времени цикла.

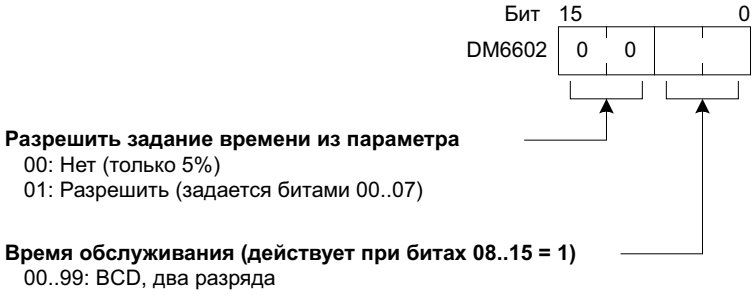
Пример: DM 6616 = 0110, порт RS 232C будет обслуживаться 10% времени цикла.

Минимальное время обслуживания - 0.34 мс.

Полное время обслуживание будет использоваться только когда есть запросы на обслуживание.

1.2.5 Время обслуживания периферийного порта

Служит для задания процента времени цикла для обслуживания периферийного порта.



По умолчанию: 5% времени цикла.

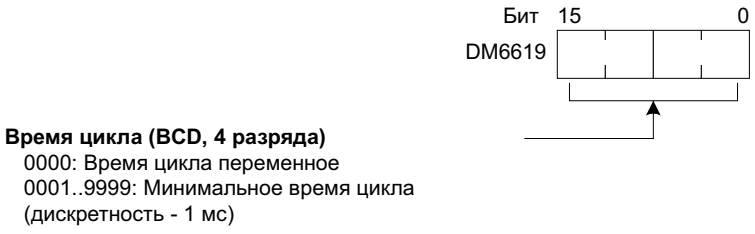
Пример: DM 6617 = 0115, периферийный порт будет обслуживаться 15% времени цикла.

Минимальное время обслуживания - 0.34 мс.

Полное время обслуживание будет использоваться только когда есть запросы на обслуживание.

1.2.6 Время цикла

Устанавливайте следующие параметры для стандартизации времени цикла и устранения “плавания” времени реакции входов/выходов путем задания минимального времени цикла.



По умолчанию: Время цикла переменное.

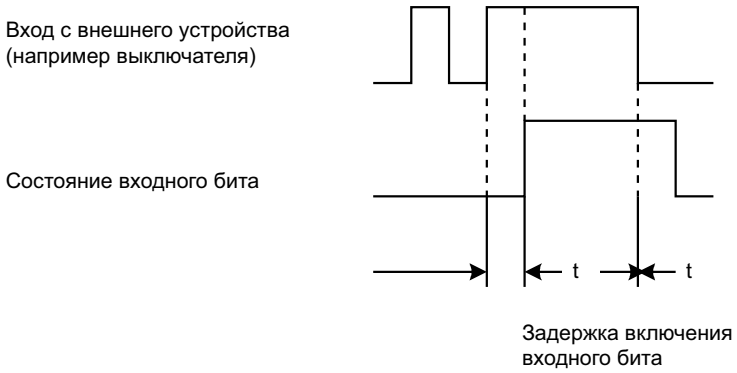
Если фактическое время цикла короче минимального, исполнение программы будет задержано до истечения минимального времени. Если фактическое время цикла

длиннее минимального, работа продолжится согласно фактическому времени цикла. Если превышено минимальное время цикла, AR 2405 установится = 1.

1.2.7 Константы времени входа

Произведите следующие настройки для задания промежутка времени от момента, когда реальные входы с блока входов постоянного тока (DC) установятся в состояние 1 или 0 до того момента, когда эти значения передадутся входным битам (т.е. когда изменится состояние 1 или 0 входных бит). Производите данную настройку тогда, когда нужно настроить время до стабильного срабатывания входов.

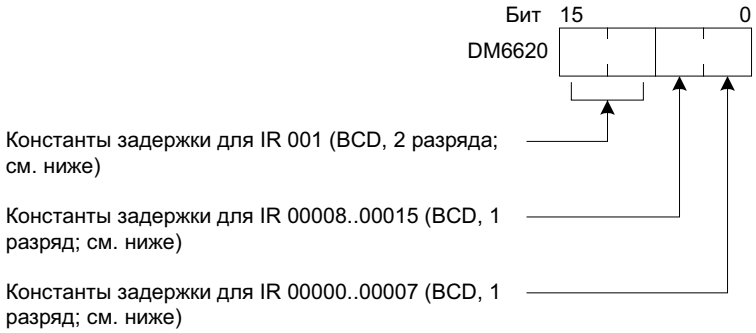
Увеличение константы времени входа сокращает влияние дребезга и внешних помех.



CQM1

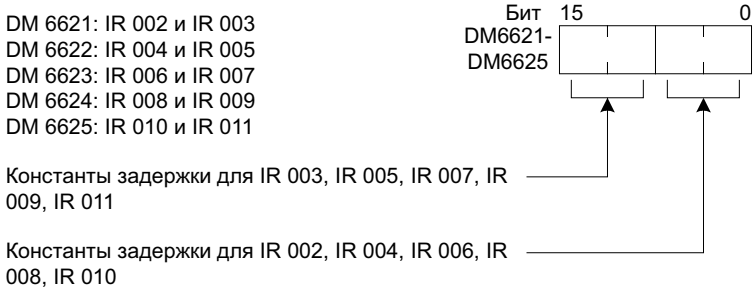
В DM 6620 содержатся константы времени входа для IR 000 и IR 001.

Константы времени входа для IR 000 и IR 001



По умолчанию: 8 мс

Константы времени входа для IR 002 - IR 011



По умолчанию: 8 мс

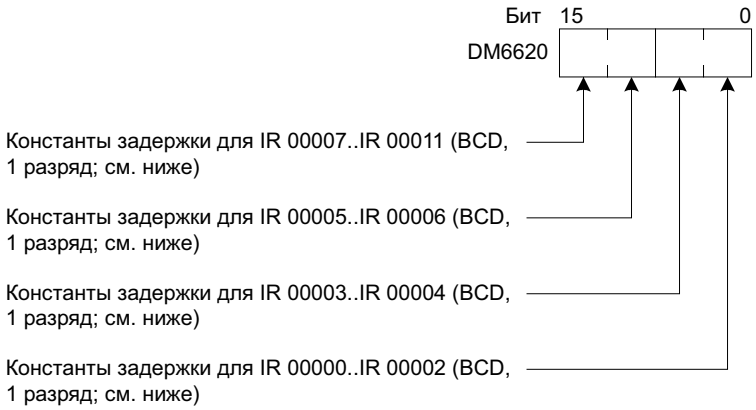
Девять возможных значений константы времени входа показаны ниже. Для IR 000 задавайте только самую правую цифру.

0: 8 мс	1: 1 мс	2: 2 мс	3: 4 мс	4: 8 мс
5: 16 мс	6: 32 мс	7: 64 мс	8: 128 мс	

CPM1/CPM1A

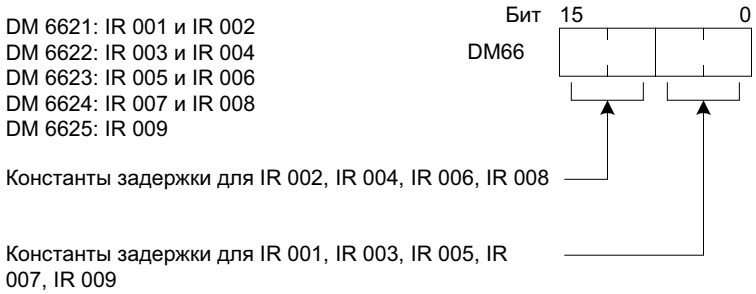
Задавайте константы времени входа для входов CPM1/CPM1A с периферийного устройства.

Константы времени входа для IR 000



По умолчанию: 8 мс

Константы времени входа для IR 001 - IR 009



По умолчанию: 8 мс

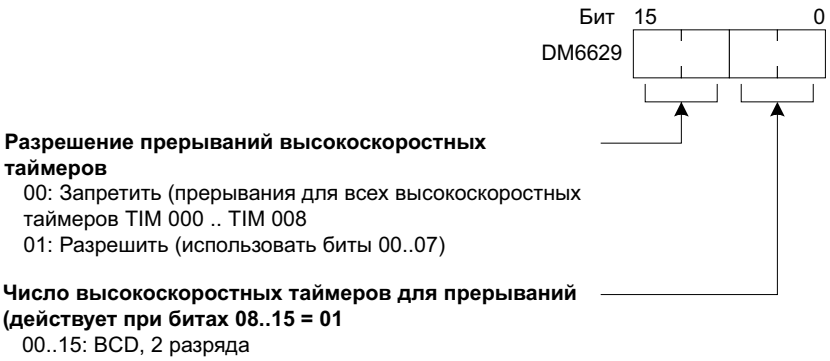
Девять возможных значений константы времени входа показаны ниже. Для IR 000 задавайте только самую правую цифру.

0: 8 м	1: 1 мс	2: 2 мс	3: 4 мс	4: 8 мс
5: 16 мс	6: 32 мс	7: 64 мс	8: 128 мс	

Время реакции на вход CPM1 - это константа времени (1 - 128 мс, по умолчанию 8 мс) + время цикла. Подробности см. 7.2.

1.2.8 Высокоскоростные таймеры (Только CQM1)

Установите значения, приведенные далее, для задания числа высокоскоростных таймеров, созданных командой TIMH(15), которые будут использовать обработку прерываний.



По умолчанию: Разрешение всех высокоскоростных таймеров, TIM 000

Данный параметр указывает число таймеров, которые будут использовать обработку прерываний, начиная с TIM 00. Например, если задано 0108, восемь таймеров - TIM 000 - TIM 007 будут использовать обработку прерываний.

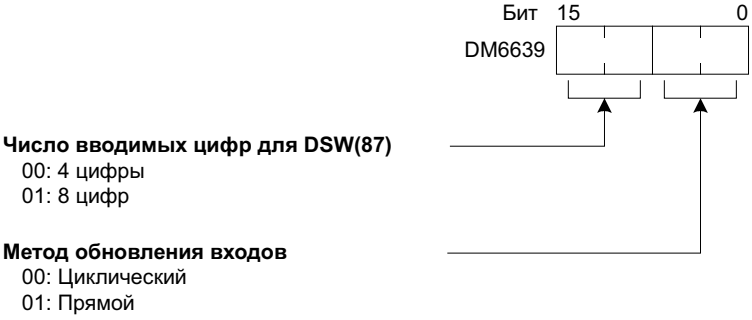
Замечание Высокоскоростные таймеры будут правильно работать без обработки прерываний только в случае, если время цикла 10 мс или меньше.

Время реакции для других прерываний улучшится, если параметр обработки прерываний будет установлен в 0, когда высокоскоростные таймеры не требуются. Это в силе всегда, когда время цикла менее 10 мс.

Замечание Если используется команда SPED (64) и выдаются импульсы частотой 500 Гц или выше, устанавливайте число высокоскоростных таймеров с обработкой прерываний 4 и менее. Подробности см, описание команды SPED (64).

1.2.9 Количество вводимых цифр для DSW(87) и Метод обновления выходов (Только CQM1)

Установите значения, приведенные далее, для задания числа вводимых цифр команды DSW(87) и метода обновления выходов.

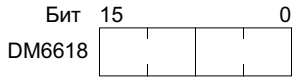


По умолчанию: Число входных цифр для DSW(87) - 4; метод обновления - циклический

Подробности о команде DSW(87) см. гл. 2 и подробности о методе обновления выходов гл. 7.

1.2.10 Параметры протокола ошибок

Установите значения, приведенные далее, для обнаружения ошибок и сохранения протокола ошибок.



Разрешение контроля цикла и еденицы контроля
00: фиксированоое значение (120мс)
01: значения битов 00..07, дискрета 10мс
02: значения битов 00..07, дискрета 100мс
03: значения битов 00..07, дискрета 1с

Время контроля цикла (действует, если биты 08..15 не равны 0)
00..99: две цифры BCD

По умолчанию: 120 мс

Время контроля цикла служит для проверки на очень длинные времена цикла, что может случиться, когда программа заходит в бесконечную петлю. Если время цикла превышает заданное в параметре, генерируется признак фатальной ошибки (FALS 9F).

Замечание 1. Единица времени, используемая для макс. и текущего времени цикла, записанная в области AR (AR 26 и AR 27 в CQM1, AR 14 и AR 15 в CPM1/CPM1A) зависит от параметра DM 6618, как показано далее.

Биты 08 - 15 = 0 или 1: 0.1 мс

Биты 08 - 15 = 2: 1 мс

Биты 08 - 15 = 3: 10 мс

2. Даже если время цикла = 1 с и более, время цикла, читаемое с программных устройств не будет превышать 999.9 мс. Правильное макс. значение и текущее время запишется в область AR.

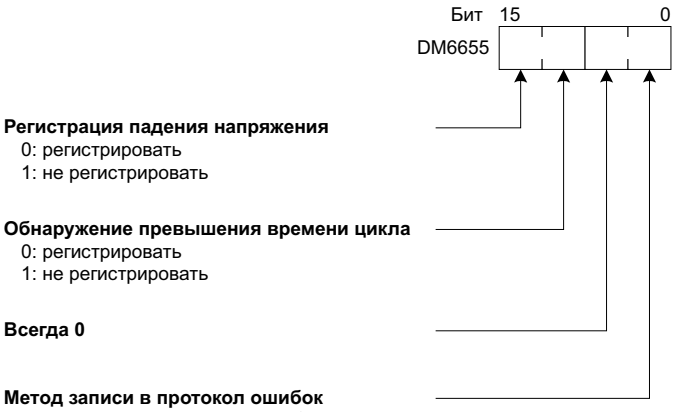
Пример:

Если в DM 6618 задано 0230, ошибка FALS 9F не появится, пока время цикла не превысит 3 с. Если фактическое время цикла = 2.59 с, текущее время цикла, записанное в область AR = 2590 (мс), но время цикла, считанное с программирующего устройства, будет = 999.9 мс.

Признак нефатальной ошибки “время цикла превышено” появится при превышении времени цикла 100 мс, если только обнаружение большого времени цикла не запрещен параметром DM 6655.

Обнаружение ошибок и операции с протоколом ошибок (DM 6655)

Установите значения, приведенные далее, для задания, нужно ли генерировать признак нефатальной ошибки, когда время цикла превысит 100 мс или когда снизится напряжение встроенного аккумулятора (только CQM1) и задания метода сохранения записей в протоколе ошибок при их появлении.



По умолчанию: регистрировать падение напряжения аккамулятора, регистрировать превышения времени цикла, сохранять последние 10 ошибок

Ошибки падения напряжения аккумулятора и превышения времени цикла - нефатальные ошибки.

Подробности о протоколе ошибок см. гл. 8.

Замечание Ошибки падения напряжения аккумулятора есть только в CQM1, в CPM1/CPM1A эта цифра не используется.

1.3 Функции выдачи импульсов (только CQM1)

В данной главе объясняются параметры и методы использования функций выдачи импульсов. Подробности об аппаратном подключении входов и портов см. Инструкцию по работе CQM1.

1.3.1 Типы выдачи импульсов

Все CQM1 могут выдавать стандартные импульсы с выходного бита, а CQM1-CPU43-EV1 может также выдавать стандартные или импульсы с переменным коэффициентом заполнения с портов 1 и 2. У стандартных импульсов коэффициент заполнения (t_{on}/T) 50 %. Коэффициент заполнения для импульсов с переменным коэффициентом заполнения может быть установлен в диапазоне 1% - 99% (через 1%).

Замечание У CQM1-CPU43-EV1 импульсы могут выдаваться с 3 портов одновременно. Кроме того, 2 порта могут быть использованы для счетных входов, независимо от выдачи импульсов.

Выдача стандартных импульсов с простого выхода

Стандартные импульсы (коэффициент заполнения = 50 %) могут выдаваться с обычного выхода с частотой 20 Гц..1 кГц. Слово входа/выхода задается в установочных параметрах, а бит задается в самой команде выдачи импульса. Подробности см. 1-3-2.

Выдача стандартных импульсов с портов 1 и 2

CQM1-CPU43-EV1 может выдавать стандартные импульсы (коэффициент заполнения = 50 %) с портов 1 и/или 2 с частотой 10 Гц..50 кГц. (макс. 20 кГц для шагового двигателя). Выдача импульсов может быть по часовой или против часовой стрелки, а частота может меняться плавно.

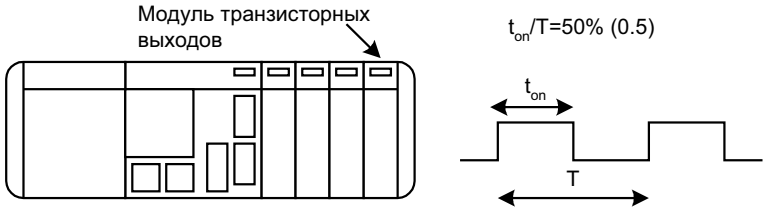
PLS2(-) и режим 0 ACC (-) нельзя использовать при установке DM 6611 в режим высокочастотного счетчика. CTBL(63) нельзя использовать с портами 1 или 2 при установке DM 6611 в режим выдачи импульсов. Подробности см. 1-3-3.

Выдача импульсов с переменным коэффициентом заполнения с портов 1 и 2

CQM1-CPU43-EV1 может выдавать импульсы с переменным коэффициентом заполнения (1 - 99 %) с портов 1 и/или 2 с частотой 91.6 Гц, 1.5 кГц или 5.9 кГц. Импульсы могут выдаваться только одного направления и выдача продолжается до останова командой INI (61). Подробности см. 1-3-4.

1.3.2 Выдача стандартных импульсов с простого выхода

Стандартные импульсы можно выдавать с заданного выходного бита командой SPED(64). Импульсы можно выдавать одновременно только с одного бита. На схеме показана выдача импульсов с обычного блока транзисторного выхода, установленного на CQM1. Коэффициент заполнения - 50 %, частоту можно задавать от 20 Гц до 1 кГц.

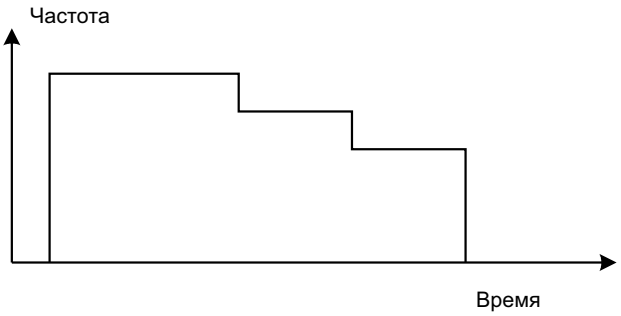


Замечание 1. Для выполнения данной задачи нужно использовать транзисторный выход.

2. Импульсы не выдаются при работе интервального таймера 0.

3. При выдаче импульсов частотой выше 500 Гц следует установить число высокоскоростных таймеров с обработкой прерываний 4 (DM 6629 = 0104).

При выдаче импульсов с обычного выхода частоту можно изменять ступенчато повторным выполнением команды SPED(64) с другой частотой, как показано на диаграмме.



Прекратить выдачу можно 2 способами:

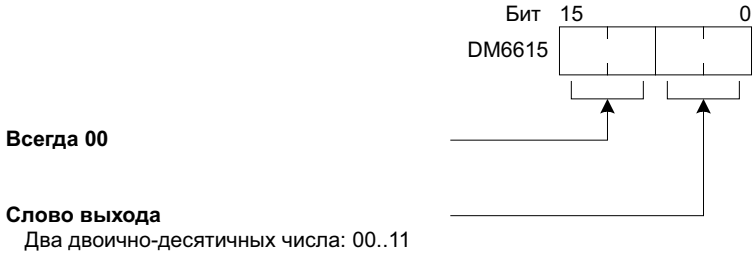
1, 2, 3,... 1. После выполнения SPED(64) выдача прекратится, если выполнить INI(61) с C=003 или снова выполнить SPED(64) с частотой 0.

2. Можно задать общее число импульсов командой PULS(65) перед выполнением SPED(64). В таком случае SPED(64) должна выполняться в “независимом режиме”. Выдача импульсов прекратится автоматически после выдачи числа импульсов, заданных командой PULS(65).

Замечание Подробности о командах PULS(65) и SPED(64) см. описание соответствующих команд.

Настройка установочных параметров

Перед выполнением SPED(64) для выдачи импульсов с Блока Выходов переключите ПК в режим PROGRAM и проведите следующие настройки:



По умолчанию: выдача импульсов с IR 100

В DM 6615 задайте выходное слово, с которого будут выдаваться импульсы. (Бит указывается в первом операнде SPED(64).)

Содержимое DM 6615 (0000..0011) задает выходное слова IR 100..IR 111. Например, если DM 6615 = 0002, импульсы будут выдаваться с IR 102.

В CQM1-CPU11/21-E установите прямое обновление выходов в параметре DM 6639 следующим образом. (В CQM1-CPU4_-E метод обновления может быть установлен либо прямой, либо циклический).

По умолчанию: циклический

Выдача импульсов в непрерывном режиме

При исполнении команды SPED(64) импульсы начинают выдаваться на указанный выход. Задайте бит выхода 00..015 (D=000..150) и частоту (20 Гц..1000 Гц) (F=0002..0100). Режим установите на непрерывный (M=001).

Выдачу импульсов можно остановить, если выполнить INI(61) с C=003 или снова выполнить SPED(64) с частотой 0. Частоту можно изменять повторным исполнением SPED(64) с другим заданием частоты.

Задание числа импульсов

Общее число импульсов, которые будут выданы, можно задать командой PULS(65) перед выполнением SPED(64) в “независимом режиме“. Выдача импульсов прекратится автоматически после выдачи числа импульсов, заданного командой PULS(65).

PULS(65) задает число импульсов 8-разрядным числом P1+1, P в диапазоне 00000001..16777215. Число импульсов, заданных PULS(65), доступно, когда SPED(64) выполняется в “независимом режиме“. (Число импульсов, процесс выдачи которых идет, нельзя изменить).

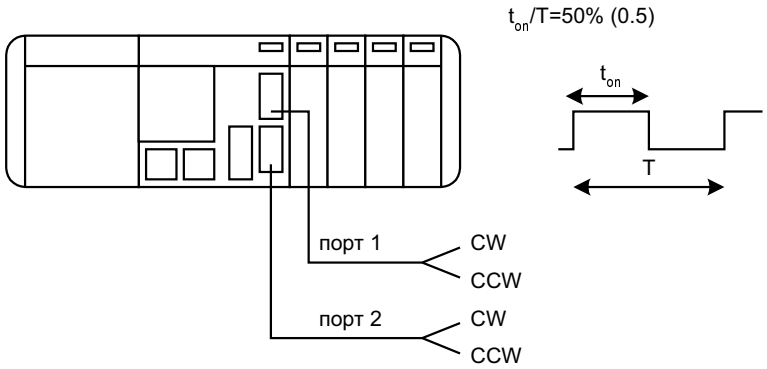
При выполнении SPED (64) импульсы начинают выдаваться с заданного выходного бита (D= 000..150, биты 00 - 15) с заданной частотой (F=0002..0100; 20 Гц - 1000 Гц). Для выдачи числа импульсов согласно PULS(65), задайте “независимый режим” (M=001). Частоту можно изменять, выполнив SPED(64) с новым значением частоты.

Изменение частоты

Частоту можно изменять, повторно выполнив SPED(64) с другим значением частоты. Используйте тот же самый выходной бит (P) и режим (M), которые использовались для пуска выдачи импульсов. Новая частота должна быть от 20 Гц до 1000 Гц (F=0002..0100).

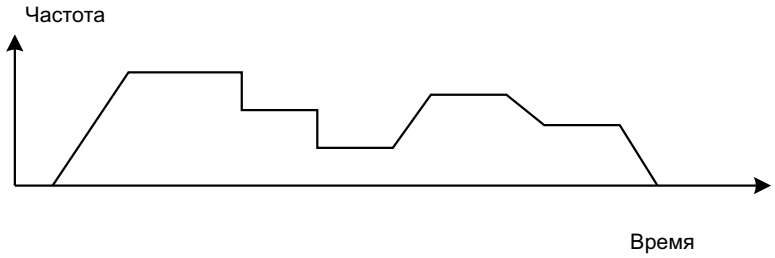
1.3.3 Выдача стандартных импульсов с портов 1 и 2

CQM1-CPU43-EV1 может выдавать стандартные с портов 1 и 2 с помощью команд SPED(64), PLS2(-) или ACC(-). Частоту можно задавать в диапазоне 10 Гц..50 кГц. (макс. 20 кГц для шагового двигателя). Импульсы могут выдаваться по часовой или против часовой, а частота может меняться плавно.



Замечание Только CQM1-CPU43-EV1 может выдавать стандартные с портов 1 и 2.

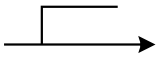
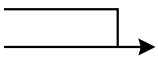
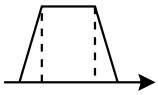
При выдаче импульсов с порта частоту можно менять плавно или ступенчато командами SPED(64), PLS2(-) или ACC(-), как показано на диаграмме.

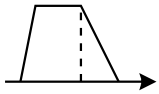
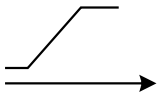
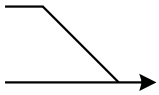
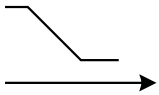


Прекратить выдачу можно 2 способами:

1, 2, 3,... 1. После выполнения SPED(64) выдача прекратится, если выполнить INI(61) с C=003 или снова выполнить SPED(64) с частотой 0.

2. Общее число импульсов, которые будут выданы, можно задать командой PULS(65) перед выполнением SPED(64). В этом случае SPED(64) должна выполняться в “независимом режиме”. Выдача импульсов прекратится автоматически после выдачи числа импульсов, заданных командой PULS(65).

Тип изменения частоты	**Команда**	**Инструкция**	**Пример N**
	Начать выдачу импульсов заданной частоты. Выдача идет непрерывно или до выдачи заданного числа импульсов. (Выполнить PULS(65), затем SPED(64).	PULS(65)	CW/CCW (по чс или против чс) (Число импульсов)
		SPED(64)	Порт Режим Частота
	Прекратить выдачу импульсов командой (Выполнить SPED(64) или INI (61)	SPED(64)	Порт Частота = 0
		INI (61)	Слово управления = 0
	Выдает заданное число импульсов Увеличивает частоту до заданной частоты с заданным ускорением. Уменьшает частоту с тем же коэффициентом замедления.	PLS2(-)	Порт CW/CCW (по чс или против чс) Коэффициент ускорения Заданная частота Число импульсов

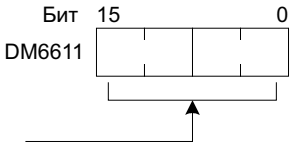
Тип изменения частоты	Команда	Инструкция	Пример N
	Выдает указанное число импульсов Увеличивает частоту до заданной частоты 1 с заданным коэффициентом. Уменьшает частоту до заданной частоты 2 с другим коэффициентом. (Выполнить PULS(65), затем ACC(-).)	PULS(65)	CW/CCW (по чс или против чс) Число импульсов Момент уменьшения
		ACC(-) (Режим 0)	Порт Коэффициент ускорения Заданная частота 1 Коэффициент замедления Заданная частота 2
	Увеличивает частоту до заданной частоты с заданным коэффициентом. Выдача импульсов продолжается. (Выполнить PULS(65), затем ACC(-).)	PULS(65)	CW/CCW (по чс или против чс)
		ACC(-) (Режим 1)	Порт Коэффициент ускорения Заданная частота
	Уменьшает текущую частоту до заданной частоты с заданным коэффициентом. Выдача прекратится после выдачи заданного числа импульсов. (Выполнить PULS(65), затем ACC(-).)	PULS(65)	CW/CCW (по чс или против чс) Число импульсов
		ACC(-) (Режим 2)	Порт Коэффициент замедления Заданная частота
	Уменьшает текущую частоту до заданной частоты с заданным коэффициентом. Выдача импульсов продолжается. (Выполнить PULS(65), затем ACC(-).)	PULS(65)	CW/CCW (по чс или против чс) Число импульсов
		ACC(-) (Режим 3)	Порт Коэффициент замедления Заданная частота

В таблице приведены типы изменения частоты, которые можно произвести комбинацией PULS(65), SPED(64), INI (61), PLS2(-) и ACC(-).

Настройка параметров ПК

Перед выполнением выдачи импульсов с портов 1 или 2 переключите ПК в режим PROGRAM и проведите следующие настройки в установочных параметрах:

В DM 6611 задайте режим для портов 1 и 2.



Реимы портов 1 и 2
0000: режим высокоскоростного счетчика
0001: режим выдачи импульсов

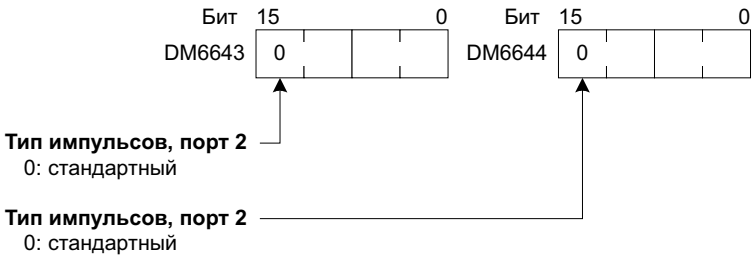
По умолчанию: режим высокоскоростного счетчика

В зависимости от режима, заданного в DM 6611, некоторые команды нельзя использовать

Значение DM 6611	“Пораженные” команды
0000: режим высокоскоростного счетчика	PLS2(-) и режим 0 ACC(-) нельзя использовать
0001: режим выдачи импульсов	CTBL(63) нельзя использовать с портами 1 и 2

Значения DM 6611 читаются только при пуске CQM1. При изменении данного слова обязательно выключите и снова включите ПК.

Задайте выдачу стандартных импульсов в DM 6643 (порт 1) и / или DM 6644 (порт 2).



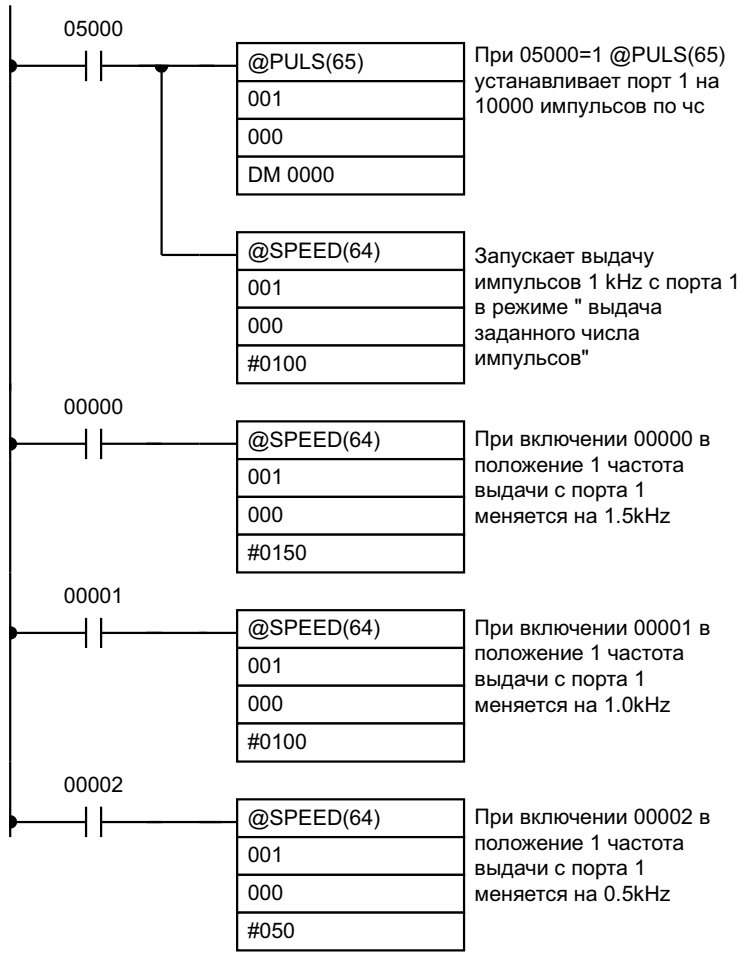
По умолчанию: стандартный

Выдача импульсов с переменным коэффициентом заполнения с портов невозможна, если в словах DM 6643 (порт 1) и DM 6644 установлены стандартные импульсы.

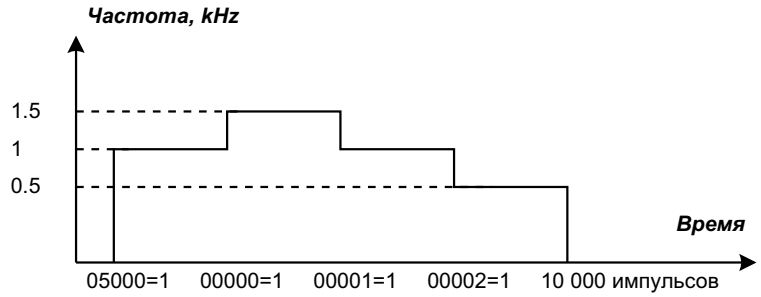
Пример 1: Запуск выдачи импульсов командами PULS(65) и SPED (64)

В следующем примере показано использование команд PULS(65) и SPED (64) для управления выдачей импульсов с порта 1. Выдается число импульсов, заданных PULS(65) (10 000), в то время как частота изменяется командой SPED (64).

Перед выполнением программы установите DM 6611 = 0001 (режим выдачи импульсов), а DM 6643 = 0000 (стандартные импульсы с порта 1).



На диаграмме приведена частота импульсов с порта 1 при выполнении программы.

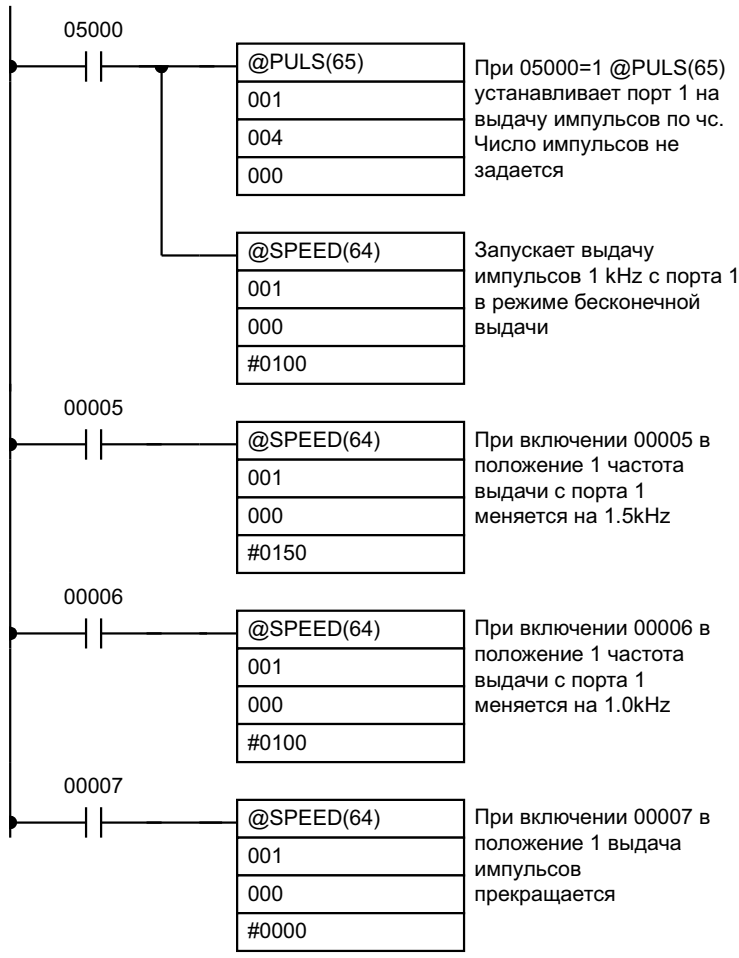


Внимание! При пуске и останове двигателя следите, чтобы частота находилась в пределах пусковой частоты двигателя.

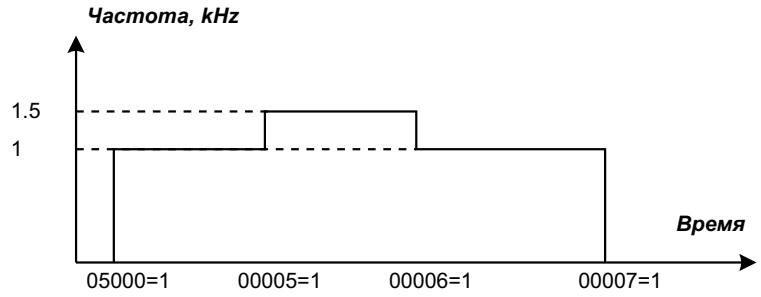
Замечание Синхронизация при управлении скоростью будет очень точной, если изменения частоты выполняются как процессы входных прерываний.

Пример 2: Прекращение выдачи импульсов командой SPED (64)

В примере показано использование команд PULS(65) и SPED (64) для управления выдачей импульсов с порта 1. Частота изменяется командой SPED (64) с другими заданиями частоты и, наконец, выдача прекращается путем задания частоты =0.



На диаграмме приведена частота импульсов порта 1 при выполнении программы.



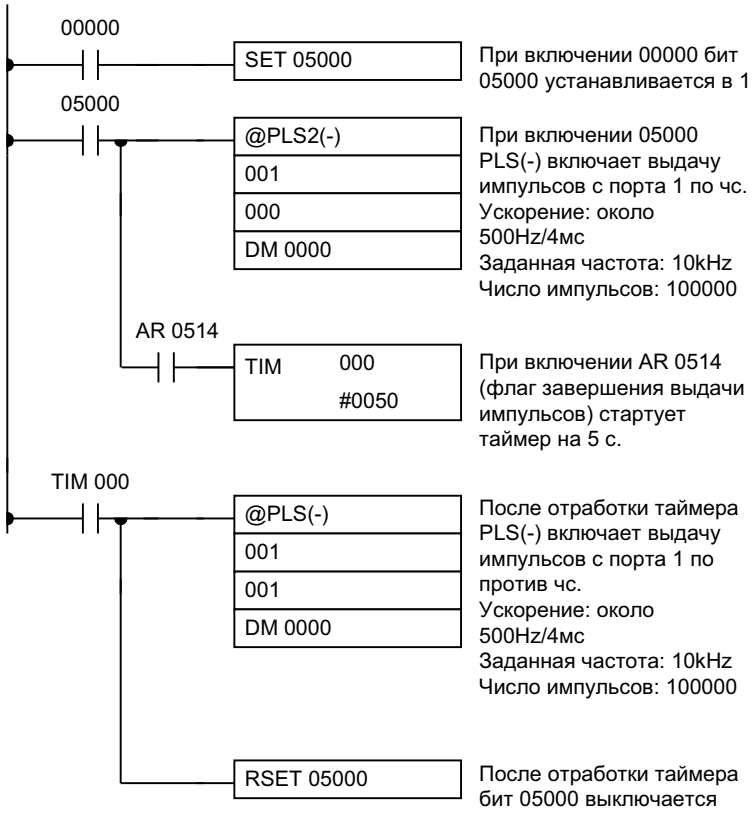
Внимание! При пуске и останове двигателя следите, чтобы частота находилась в пределах пусковой частоты двигателя.

На диаграмме приведена частота выдачи импульсов с порта 1 при выполнении этой программы.

Пример 3: PLS2(-)

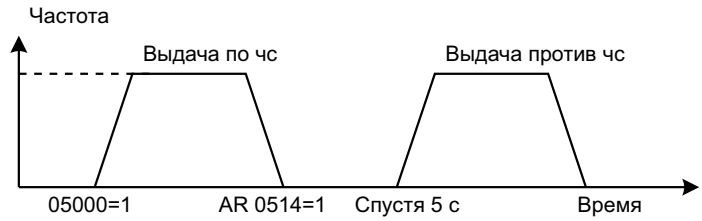
В следующем примере показано использование команды PLS2(-) для выдачи 100 000 импульсов по чс с порта 1. Частота увеличивается до 10 кГц с ускорением 500 Гц/ 4 мс и уменьшается с таким же темпом.

DM 0000	0050
DM 0001	1000
DM 0002	0000
DM 0003	0010



Через 5 с после начала выдачи по чс другая команда PLS2(-) выдает 100 000 импульсов против чс с теми же параметрами.

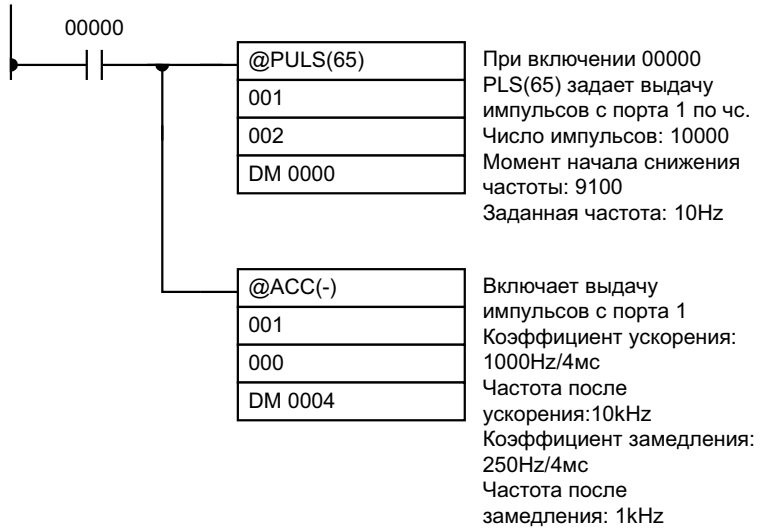
На диаграмме приведена частота выдачи импульсов с порта 1 при выполнении этой программы.



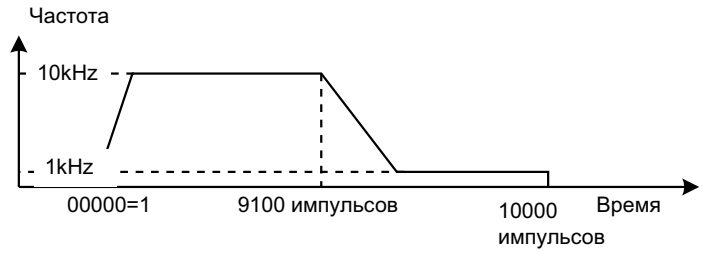
Пример 4: ACC(-), режим 0

В следующем примере показано использование режима 0 команды ACC(-) для выдачи 10 000 импульсов по чс с порта 1. Частота увеличивается до 10 кГц с ускорением около 1 кГц/ 4 мс и уменьшается до 1 кГц с ускорением 250 Гц/4 мс. Замедление начинается после выдачи 9 100 импульсов.

DM 0000	0000	DM 0004	0100
DM 0001	0001	DM 0005	1000
DM 0002	9100	DM 0006	0025
DM 0003	0000	DM 0007	0050

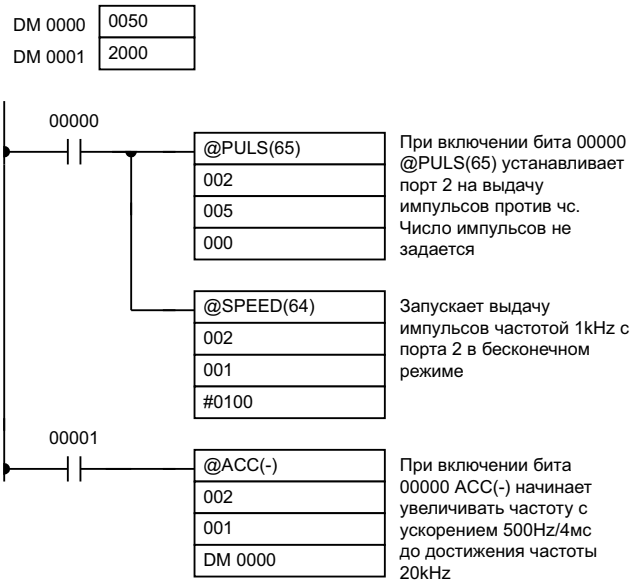


На диаграмме приведена частота импульсов порта при выполнении программы.

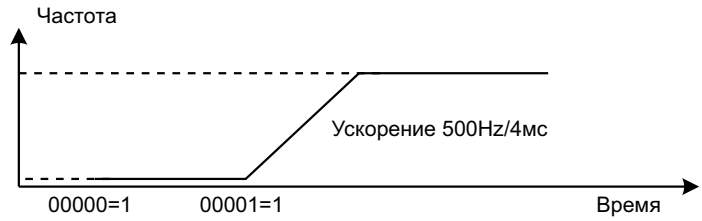


Пример 5: ACC(-), режим 1

В следующем примере показано использование команды ACC(-), режим 1 для увеличения частоты импульсов с порта 2. Частота увеличивается с 1 кГц до 20 кГц с ускорением 500 Гц/ 4 мс.



На следующей диаграмме приведена частота импульсов порта 2 при выполнении программы.

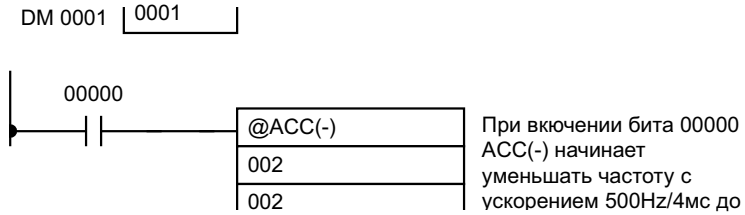


На следующей диаграмме приведена частота импульсов порта 1 при выполнении программы.

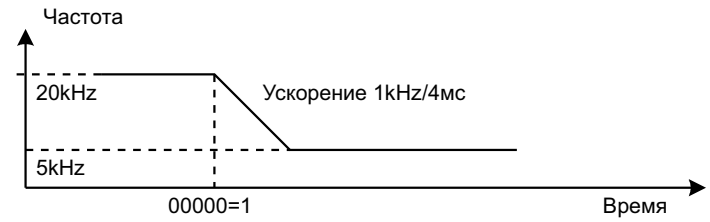
Замечание Выдачу импульсов можно остановить командой ACC(-), режим 2, с конечной частотой 0, однако выдача будет прекращена не точно после выдачи заданного числа импульсов, так что этим методом следует пользоваться только при аварийных остановках.

Пример 6: ACC(-), режим 2

В следующем примере представлено использование команды ACC(-), режим 2 для снижения частоты импульсов порта 1. Выдача импульсов с частотой 2 кГц в “независимом режиме“ была уже начата, и останов произойдет после выдачи заданного числа импульсов.



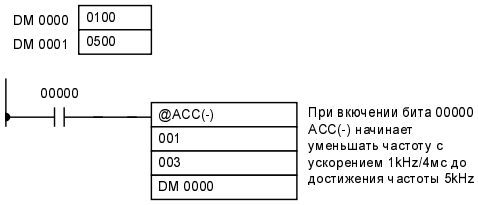
На следующей диаграмме приведена частота импульсов порта 1 при выполнении программы.



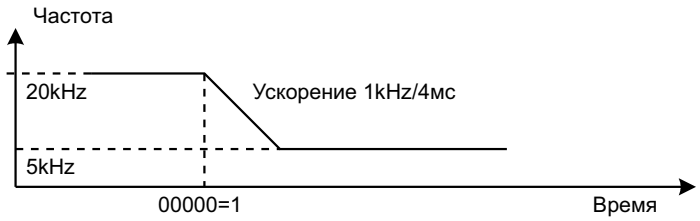
Замечание Выдачу импульсов можно остановить командой ACC(-), режим 2, с конечной частотой 0, однако выдача будет прекращена не точно после выдачи заданного числа импульсов, так что этим методом следует пользоваться только при аварийных остановках.

Пример 7: ACC(-), режим 3

В следующем примере представлено использование команды ACC(-), режим 3 для снижения частоты импульсов порта 1. Выдача импульсов с частотой 20 кГц в непрерывном режиме уже шла.

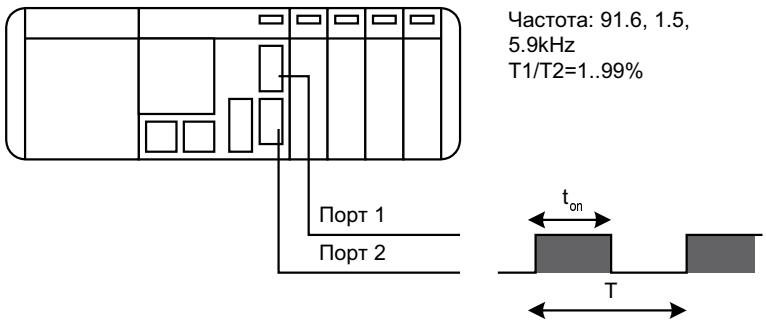


На диаграмме приведена частота импульсов порта при выполнении программы.



1.3.4 Выдача импульсов с переменным коэффициентом заполнения с портов 1 и 2

CQM1-CPU43-EV1 может выдавать импульсы с переменным коэффициентом заполнения с портов 1 и/или 2 с помощью команды PWM(-). Частоту импульсов можно задать 91.6 кГц, 1.5 кГц или 5.9 кГц. Данная функция служит для различных задач управления, таких, как выдача для легкой нагрузки или выдача управления скоростью на преобразователь.

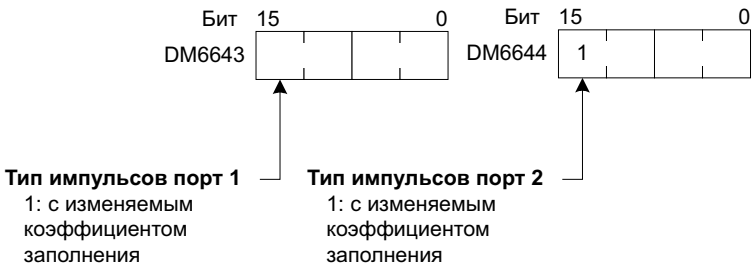


Замечание Только ЦУ CQM1-CPU43-EV1 может выдавать импульсы с портов 1 и 2.

Настройка параметров ПК

Перед выдачи импульсов переменного коэффициента заполнения с портов 1 или 2 переключите ПК в режим PROGRAM и проведите следующие настройки:

Задайте параметры выдачи импульсов с переменным коэффициентом заполнения DM 6643 (порт 1) и / или DM 6644 (порт 2).



По умолчанию: стандартные импульсы

Импульсы с переменным коэффициентом заполнения нельзя выдавать с порта, если в словах DM 6643 и DM 6644 установлены стандартные импульсы.

Пуск выдачи импульсов

Импульсы будут выдаваться с заданного порта исполнением команды PWM(-). Задайте порт 1 или 2 (P=001..002). Задайте частоту 5.9 кГц, 1.5 кГц или 91.6 Гц (F=000, 001 ,002). Задайте коэффициент заполнения от 1% до 99 % (D=0001..0099, двоично-десятичное число).

Выдача будет продолжаться с заданной частотой и коэффициентом заполнения до тех пор, пока не встретится еще одна команда PWM(-) с другими заданиями или команда INI(61) для прекращения выдачи с указанного порта.

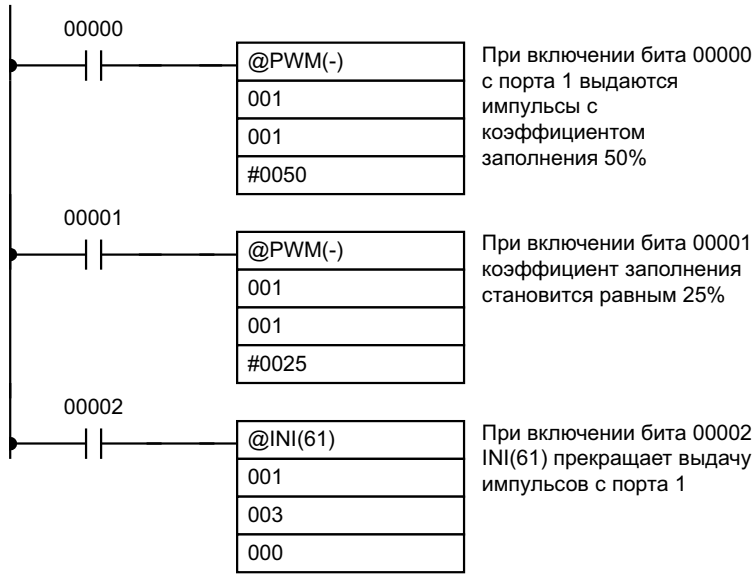
Прекращение выдачи импульсов

Выдача импульсов будет прекращена с заданного порта при выполнении команды INI(61) с C = 003. Задайте порт 1 (P=001) или 2 (P=002).

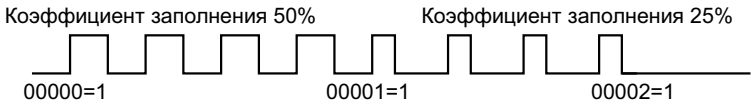
Пример: PWM (-)

В примере представлено использование команды PWM (-) для запуска выдачи с порта 1 импульсов 1.5 кГц и изменения коэффициента заполнения с 50 % до 25 %. Затем выдача останавливается командой INI(61).

Перед выполнением программы задайте DM 6643 = 1000 (выдача импульсов с переменным коэффициентом заполнения с порта 1).



На диаграмме показан коэффициент заполнения импульсов порта 1 при выполнении данной программы.



1.3.5 Определение состояния портов 1 и 2

Состояние выдачи импульсов с портов 1 и 2 (для стандартных импульсов или с переменным коэффициентом заполнения) можно определить либо чтением состояния соответствующих флагов в SR и AR, либо командой PRV(62).

Чтение состояния флагов

Состояние выдачи импульсов можно определить, прочитав содержание слов и флагов, указанных в таблице.

Слово(а)	Бит(ы)	Функция	Описание
SR 236, SR 237	00 - 15	Текущее значение порта 1	Показывает текущее значение (8 цифр) выданных импульсов с порта 1. Старшие 4 цифры числа содержатся в SR 237.
SR 238, SR 239	00 - 15	Текущее значение порта 2	Показывает текущее значение (8 цифр) выданных импульсов с порта 2. Старшие 4 цифры числа содержатся в SR 239.
AR 04	00 - 15	Состояние выдачи импульсов	Показывает состояние выдачи импульсов 00: нормальная выдача 01..02: ошибка аппаратуры 03: ошибка установочных параметров ПК 04: При выдаче импульсов была остановлена работа.
AR05	12	Порт 1, флаг уменьшения частоты	Показывает уменьшение частоты. 0: Не задано 1: Задано
	13	Порт 1, флаг числа импульсов	Задано ли число импульсов 0: Не задано 1: Задано
	14	Порт 1, флаг завершения выдачи импульсов	Показывает, завершена ли выдача импульсов 0: Не завершена 1: Завершена)
	15	Порт 1, флаг состояния выдачи	Показывает, идет ли выдача импульсов 0: Не идет 1: Идет выдача)

Слово(а)	Бит(ы)	Функция	Описание
AR 06	12	Порт 2, флаг уменьшения частоты	Показывает уменьшение частоты. 0: Не задано 1: Задано
	13	Порт 2, флаг числа импульсов	задано ли число импульсов 0: Не задано 1: Задано
	14	Порт 2, флаг завершения выдачи импульсов	Показывает, завершена ли выдача импульсов 0: Не завершена 1: Завершена
	15	Порт 2, флаг состояния выдачи	Показывает, идет ли выдача импульсов 0: Не идет 1: Идет выдача)

Выполнение PRV(62)

Состояние выдачи импульсов можно определить командой PRV(62). Задайте порт 1 (P= 001) или 2 (P =002) и слово назначения D. Информация о состоянии порта будет записана по адресу D, в биты 04 - 07, а биты 08 - 15 обнулятся.

При применении PRV(62) для чтения состояния порта будет читатся самая последняя информация, так что время цикла ПК не будет оказывать влияния.

Биты 4 - 7 содержат информацию об указанном порте.

Бит	Функция	Описание
04	Флаг уменьшения частоты	Показывает уменьшение частоты. (0: Без уменьшения, 1: с уменьшением)
05	Флаг числа импульсов	Задано ли общее число импульсов (0: Не задано, 1: задано)
06	Флаг завершения выдачи импульсов	Показывает, завершена ли выдача импульсов (0: Не завершена, 1: завершена)
07	Флаг состояния выдачи	Показывает, идет ли выдача импульсов (0: Не идет, 1: идет выдача)

1.4 Функции прерываний CQM1

В данной главе описаны установочные параметры и методы применения функций прерываний CQM1.

1.4.1 Типы прерываний

CQM1 имеет 3 типа прерываний, описанных далее.

Входные прерывания:

Обработка прерываний производится, когда вход из внешнего источника включает в 1 один из битов ЦУ 00000..00003.

Прерывания интервального таймера:

Обработка прерываний выполняется по интервальному таймеру с точностью 0.1 мс.

Прерывания высокоскоростного счетчика:

Обработка прерываний выполняется в соответствии с текущим значением (PV) встроенного высокоскоростного счетчика. Все ЦУ CQM1 имеют встроенный высокоскоростной счетчик 0, который считает входные импульсы на одном из битов ЦУ 00004..00006. Можно считать двухфазные импульсы частотой до 2.5 кГц.

ЦУ CQM1-CPU43/44-E может также считать входные импульсы с портов 1 и 2.

- CQM1-CPU43-EV1: высокоскоростные счетчики 1 и 2 считают высокоскоростные входные импульсы портов 1 или 2. Можно считать двухфазные импульсы частотой до 25 кГц.
- CQM1-CPU44-EV1: Высокоскоростные счетчики 1 и 2 считают коды, поступающие с абсолютного датчика вращения на порты 1 и 2.

Обработка прерываний

При генерировании прерывания выполняется заданная программа обработки прерывания. Прерывания имеют следующий порядок приоритетов. (Входное прерывание 0 имеет наивысший приоритет, прерывание высокоскоростного счетчика 0 - самый низший):

1, 2, 3,... 1. Входное прерывания 0 Входное прерывание 1 Входное прерывание 2 Входное прерывание 3.

2. Прерывание высокоскоростного счетчика 1 Прерывание высокоскоростного счетчика 2

3. Прерывание интервального таймера 0 Прерывание интервального таймера 1 Прерывание интервального таймера 2 (Прерывание интервального таймера 2 является прерыванием высокоскоростного счетчика 0).

Когда при обработке прерывания происходит прерывание с более высоким приоритетом, текущий процесс прекращается и начинается обработка нового прерывания. По полного завершения этой программы возобновится обработка предыдущего прерывания.

Когда при обработке прерывания происходит прерывание с низшим или равным приоритетом, новое прерывание будет обрабатываться, как только текущая программа прерывания полностью отработается.

Подобно обычным подпрограммам, программы обработки прерываний задаются командами SBN(92) и RET(93) в конце главной программы.

Во время отработки программ прерываний можно обновлять заданный диапазон входных бит.

Если программа прерываний определена, во время контроля программы появляется сообщение “NO SBS ERROR” (“Отсутствие вызова подпрограммы командой SBS”), но программа будет выполняется нормально. При появлении этого сообщения проверьте все обычные подпрограммы, чтобы убедиться, что для всех обычных подпрограмм запрограммированы SBS(91).

Команды выдачи импульсов и прерывания

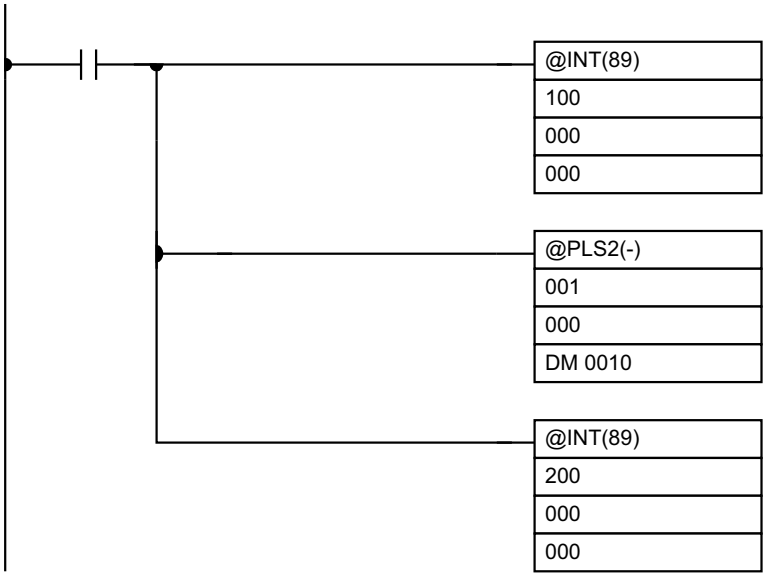
У ЦУ CQM1-CPU43/44-E следующие команды не выполняются в подпрограмме прерывания во время обработки в главной программе команд, которые управляют вводом/выдачей импульсов или высокоскоростными счетчиками: (25503=1)

INI(61), PRV(62), CTBL(63), SPED(64), PULS(65), PWM(-), PLS2(-) и ACC(-)

Для обхода данного ограничения можно использовать следующие методы:

Метод 1

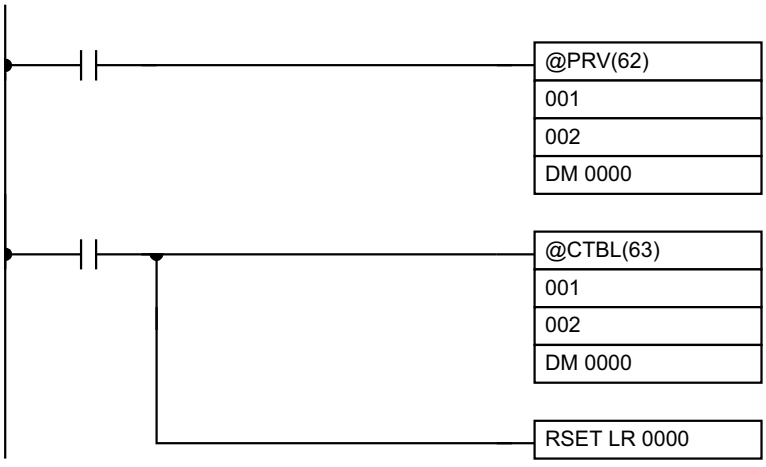
При выполнении команды все прерывания можно замаскировать.



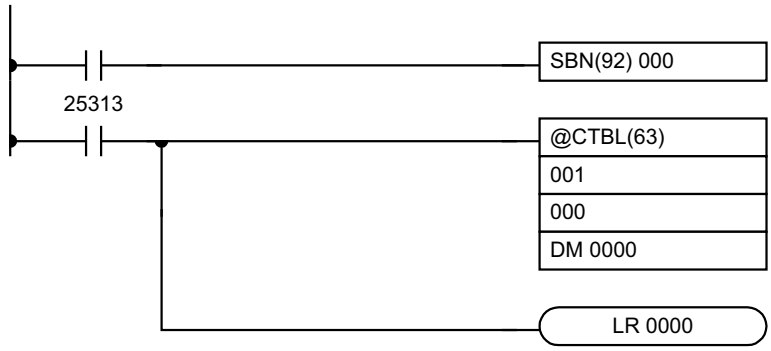
Метод 2

Повторно выполнить команду в главной программе.

Секция в главной программе:



Секция в подпрограмме прерываний:



1.4.2 Входные прерывания

Входы ЦУ IR 00000..IR 00003 можно использовать для прерываний из внешних источников. Входные прерывания 0..3 соответствуют этим битам и всегда служат для вызова подпрограмм с номерами соответственно 000..003. Когда входные прерывания не используются, подпрограммы 000 - 003 можно использовать как обычные подпрограммы.

Обработка

Есть 2 типа обработки входных прерываний. Первый - Режим прерывания по входу, в котором прерывание выполняется в ответ на внешний вход. Второй - Режим счетчика, в котором сигналы из внешнего источника подсчитываются с высокой скоростью, и прерывание обрабатывается один раз после отсчета некоторого заданного числа сигналов.

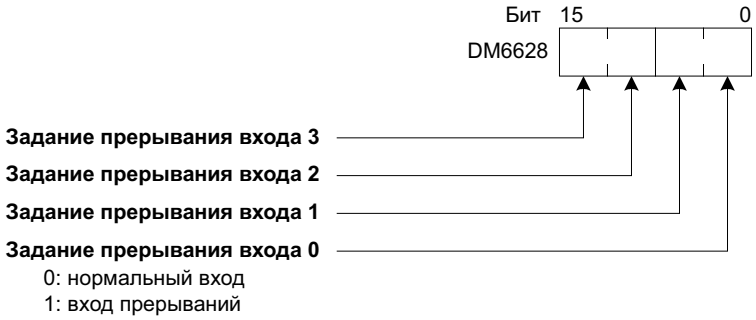
В Режиме прерывания по входу можно обнаружить сигналы длиной 100 мкс и более. В Режиме счетчика можно считать сигналы с частотой до 1 кГц.

Установочные параметры

Перед выполнением программы произведите следующую настройку установочных параметров в режиме PROGRAM.

Параметры входных прерываний (DM 6628)

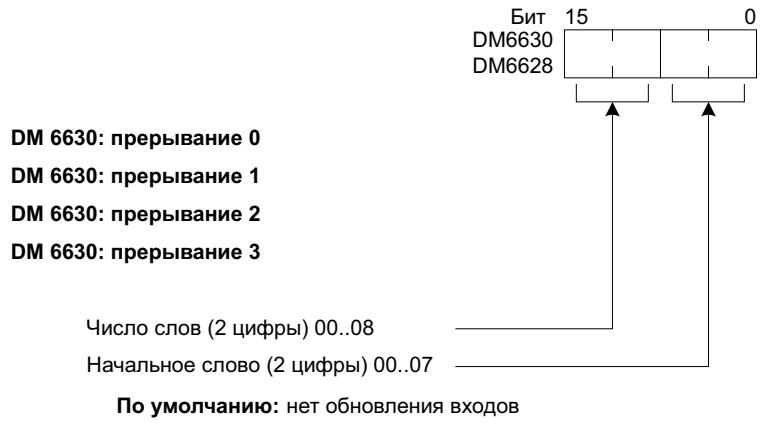
Если не произвести такую настройку, прерывания в программе использовать нельзя.



По умолчанию: нормальный вход

Параметры слов обновления входов (DM 6630 - 6633)

Производите данные настройки в случае необходимости обновления входов.



По умолчанию: нет обновления входов

Пример:

Если в DM 6630 занесено 0100, IR 000 обновится, когда будет получен сигнал прерывания 0.

Замечание Если обновление входа не используется, состояние входного сигнала в программе прерывания будет неопределенным. Это относится и к биту, вызвавшему прерывание. Например, IR 0000 не будет = 1 в программе входного прерывания, если его не обновить (в таком случае вместо IR 00000 можно использовать флаг ВСЕГДА 1 (SR 25313).

Режим прерываний по входу

Используйте следующие команды для программирования режима прерываний по входу.

Маскирование прерываний

Установите или сбросьте маску прерываний командой INT(89).

@INT(89)
000
000
D

Настройки производятся битами 0..3 слова D,
соответствующими входным прерываниям 0..3
0: маска снята (прерывания разрешены)
1: маска установлена (прерывания запрещены)

Перед началом работы все входные прерывания замаскированы.

Стирание замаскированных прерываний

Если бит, соответствующий входному прерыванию, стал = 1 во время замаскированного состояния, это входное прерывание сохранится в памяти и начнет отрабатываться после снятия маски. Чтобы это прерывание не выполнялось после снятия маски, оно должно быть убрано из памяти.

Только один сигнал прерываний будет сохранен в памяти для каждого номера прерывания.

Командой INT(89) сотрите входное прерывание из памяти.

@INT(89)
001
000
D

Если биты 0..3, соответствующие входным
прерываниям 0..3, будут установлены в 1,
то входные прерывания убраны из памяти.
0: входные прерывания сохраняются
1: входные прерывания убраны

Чтение состояния маски

Командой INT(89) читайте состояние маски входного прерывания.

@INT(89)
002
000
D

Состояние правой цифры, загруженной в слово
D (биты 1..3), показывает состояние маски.
0: маска снята (прерывания разрешены)
1: маска установлена (прерывания запрещены)

Режим счетчика

Используйте следующие шаги для программирования входных прерываний в Режиме счетчика.

Замечание Слова SR, используемые в Режиме счетчика (SR 244..SR 245), все содержат двоичные (16-ричные), а не двоично-десятичные данные.

1, 2, 3,... 1. Запишите задание для операций подсчета в слова SR, соответствующие прерываниям 0..3. Диапазон заданий - 0000..FFFF (0..65 535). Значение 0000 запретит операцию счета до занесения нового значения и выполнения шага 2.

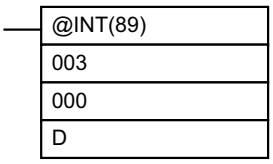
Замечание Перед началом работы биты SR очищаются и должны быть записаны из программы.

Максимальная частота подсчета сигнала - 1 кГц.

Прерывание	**Слово**
Входное прерывание 0	SR 244
Входное прерывание 1	SR 245
Входное прерывание 2	SR 246
Входное прерывание 3	SR 247

Если Режим счетчика не используется, эти биты SR можно использовать как рабочие биты.

2. Командой INT(89) обновите задание для Режима счетчика и разрешите прерывания.



Если биты 0..3 слова D, соответствующие входным прерыавниям 0..3, будут установлены в 0, то заданное значение будет обновлено и прерывание разрешено.
0: обновление заданного значения и маска снята
1: ничего не происходит (задавайте 1 для всех прерываний, которые не будут меняться)

Входное прерывание, у которого обновлено задание, будет разрешено для работы в Режиме счетчика. Когда счетчик досчитает до задания, производится прерывание, счетчик сбрасывается и последовательность счет/прерывания продолжается до остановки счетчика.

Замечание 1. Если команда INT(89) используется во время счета, текущее значение (PV) становится равным заданию (SV). Таким образом, Вы должны использовать форму команды по изменению условия (@), или прерывание может никогда не произойти.

2. Задание будет установлено, когда выполняется команда INT(89). Если уже идут прерывания, задание не изменится простым изменением SR 244..SR 247, т.е. если содержимое изменено, задание должно быть снова обновлено командой INT(89).

Прерывания можно замаскировать таким же образом, как и в режиме прерывания по входу, но если маску и снимать таким же образом, Режим счетчика не сохранится, а вместо него будет вызван режим прерывания по входу. Входные сигналы, принятые для замаскированных прерываний, можно также очистить таким же образом, как и в режиме прерывания по входу.

Текущее значение счетчика в Режиме счетчика

Когда входные прерывания используются в Режиме счетчика, текущие значения счетчика будут сохранены в словах SR, соответствующих входным прерываниям 0..3. Диапазон значений 0000..FFFF (0..65 534) и будет равен текущему значению счетчика текущего значения (PV) минус 1.

Прерывание	Слово
Входное прерывание 0	SR 248
Входное прерывание 1	SR 249
Входное прерывание 2	SR 250
Входное прерывание 3	SR 251

Пример:

Текущее значение для прерывания с заданием 000A сразу после выполнения команды INT(89) будет 0009.

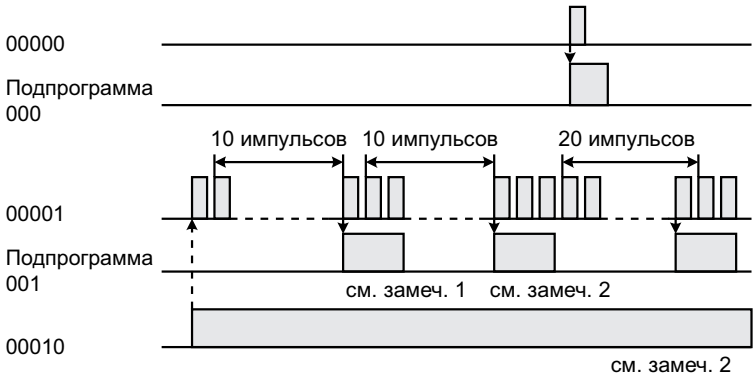
Замечание Даже если входные прерывания не используется в Режиме счетчика, эти биты SR нельзя использовать как рабочие биты.

Пример применения

В данном примере входное прерывание 0 используется в режиме прерывания по входу, а входное прерывание 1 в режиме счетчика. Перед выполнением программы проверьте установочные параметры.

Установочные параметры: DM 6628: 0011 (IR 0000 и IR 0001 будут использоваться для входных прерываний). Для всех других установочных параметров берутся значения по умолчанию. (Входы не обновляются во время обработки прерываний).

При исполнении данной программы работа будет проходить в соответствии со следующей диаграммой:



Замечание 1. Счетчик будет продолжать работу даже во время исполнения программы прерываний.

2. Входное прерывание останется замаскированным

1.4.3 Маскирование всех прерываний

Все прерывания, включая входные прерывания, прерывания интервального таймера и прерывания высокоскоростного счетчика можно замаскировать и размаскировать как группу командой INT(89). Эта маска является дополнительной ко всем другим маскам индивидуальных типов прерываний. Кроме того, очистка масок всех прерываний не снимает маски индивидуальных типов прерываний, а возвращает их в замаскированное состояние, которое существовало до выполнения команды INT(89) для маскирования их как группы.

Пользуйтесь командой INT(89) только когда требуется временно замаскировать все прерывания, и всегда используйте команды INT(89) в паре, используя первую команду для маскирования, а вторую для размаскирования прерываний.

INT(89) нельзя использовать для маскирования и размаскирования всех прерываний из программ прерывания.

Маскирование прерываний

Используйте команду INT(89) для запрещения всех прерываний.

@INT(89)
100
000
000

Если прерывание вызывается, когда прерывания замаскированы, обработка прерывания не произойдет, однако запрос на прерывание будет запомнен для прерываний входа, интервального таймера и высокоскоростного счетчика. Прерывания будут обслужены, как только они будут размаскированы.

Размаскирование прерываний

Используйте команду INT(89) для размаскирования прерываний:

@INT(89)
200
000
000

1.4.4 Прерывания интервального таймера

Высокоскоростная, высокоточная обработка таймерных прерываний можно исполнятся с использованием интервальных таймеров. На CQM1 имеются 3 интервальных таймера с номерами 0..2.

Замечание 1. Интервальный таймер 0 нельзя использовать во время выдачи импульсов на блоки выходов командой SPED(64).

2. Интервальный таймер 2 нельзя использовать одновременно с высокоскоростным счетчиком.

Обработка

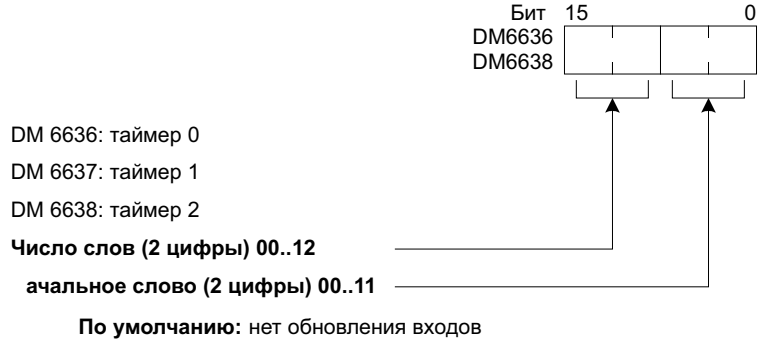
Есть 2 режима работы интервального таймера: Одноразовый режим, в котором только одно прерывание будет исполняться по истечении заданного времени, и Режим прерывания по расписанию, в котором прерывания повторяются через фиксированные интервалы.

Установочные параметры

При использовании прерываний интервального таймера произведите следующие настройки установочных параметров в режиме PROGRAM перед исполнением программы.

Настройка обновления входных слов (DM 6636..6638)

Производите данные настройки в случае необходимости обновления входов.



Настройка высокоскоростного счетчика 0 (DM 6642)

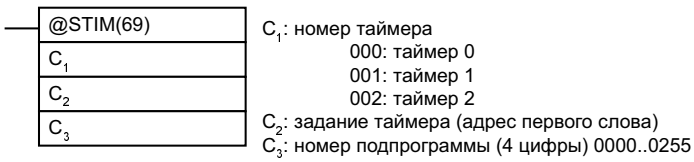
При использовании интервального таймера 2 перед началом работы проверьте, чтобы параметры высокочастотного счетчика (DM 6642) были установлены по умолчанию (0000: высокоскоростной счетчик не используется).

Работа

Используйте следующую команду для запуска и управления интервальным таймером.

Пуск в одноразовом режиме

Используйте команду STIM(69) для пуска интервального таймера в одноразовом режиме.



C2: Задание декрементирующего счетчика (4 двоично-десятичных цифры): 0000..9999.

C2+1: Единица времени декрементирования (4 двоично-десятичных цифры; единица времени: 0.1 мс): 0005..0320 (0.5 мс..32 мс).

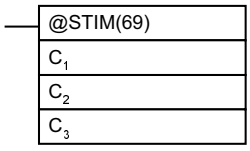
Каждый раз по истечении заданного интервала в слове C2+1 декрементирующий счетчик уменьшает текущее значение на 1. Когда оно станет равным 0, будет вызвана заданная подпрограмма один раз и таймер остановится.

Время от выполнения команды STIM (69) до истечения заданного времени рассчитывается следующим образом:

(Содержание C2) x (Содержание C2 + 1) x 0.1 мс =(0.5..319 968 мс). Если для C2 задана константа, тогда заданное значение декрементального счетчика станет равно этому значению и интервал декрементирования будет = 10 (1 мс). (Задание выражено в мс).

Пуск режима прерываний по расписанию

Используйте команду STIM(69) для пуска интервального таймера в режиме прерываний по расписанию.



C_1: номер таймера+3
003: таймер 0
004: таймер 1
005: таймер 2
C_2: задание таймера
C_3: номер подпрограммы (4 цифры) 0000..0255

C2: Задание декрементирующего счетчика (4 двоично-десятичных цифры): 0000..9999.

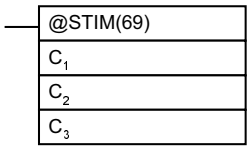
C2+1: Интервал времени декрементирования (4 двоично-десятичных цифры; единица времени: 0.1 мс): 0005..0320 (0.5 мс..32 мс).

Значения параметров аналогичны параметрам для одноразового режима, но в режиме прерываний по расписанию текущее значение таймера будет установлено в значение задания и декрементирование начнется снова после вызова программы прерывания. В режиме прерываний по расписанию прерывания будут повторяться с фиксированными интервалами до тех пор, пока работа не будет остановлена.

Замечание CQM1-CPU11-EV1/CPU21-EV1 поддерживает только номера подпрограмм 0000 - 0127.

Чтение истекшего времени таймера

Используйте команду STIM(69) для чтения истекшего времени таймера.



C_1: номер таймера+6
006: таймер 0
007: таймер 1
008: таймер 2
C_2: первое слово параметра 1
C_3: параметр 2

C2: Число, показывающее сколько раз был декрементирован счетчик (4 двоично-десятичных цифры).

C2+1: интервал времени декрементирующего счетчика (4 двоично-десятичных цифры; единица времени: 0.1 мс).

C3: истекшее время от предыдущего декремента. (4 двоично-десятичных цифры; единица времени: 0.1 мс).

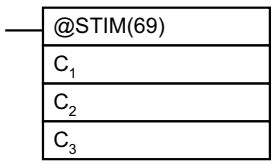
Время от начала отсчета интервала до выполнения данной команды рассчитывается следующим образом:

(Содержание C2) x (Содержание C2 + 1) x 0.1 мс.

Если заданный интервал отработан, будет выдан 0000.

Остановка таймеров

Используйте команду STIM(69) для остановки интервального таймера.

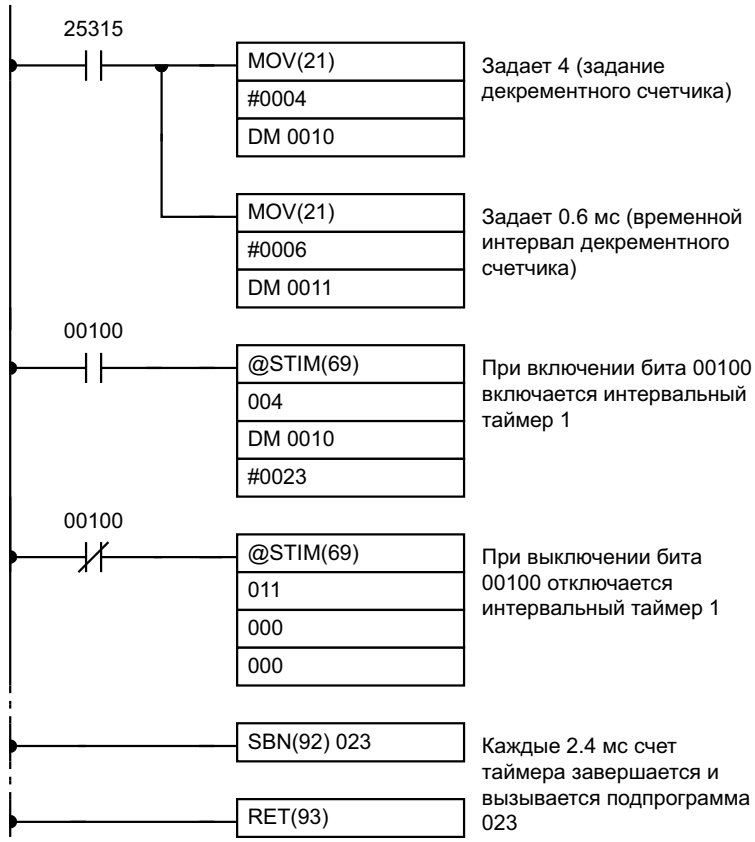


C_1: номер таймера+10
010: таймер 0
011: таймер 1
012: таймер 2

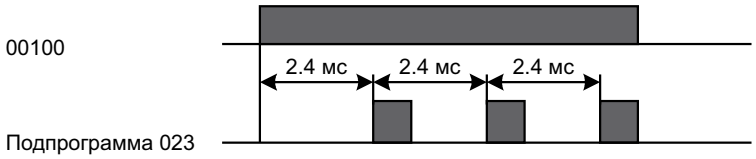
Указанный интервальный таймер остановится.

Пример применения

В данном примере прерывание производится каждые 2.4 мс (0.6 мс x 4) интервальным таймером 1. Предполагается, что все установочные параметры взяты по умолчанию. (Входы для отработки прерываний не обновляются).



При выполнении данной программы подпрограмма 023 будет выполняться каждые 24 мс, пока IR 00100 = 1.



1.4.5 Прерывания высокоскоростного счетчика (ВСч) 0

Импульсные сигналы, поступающие с импульсного датчика положения на биты ЦУ 00004..00006 можно подсчитывать с высокой скоростью, и в соответствии со значением счета можно выполнять обработку прерываний.

Отработка

Типы входных сигналов и режимы счета

С импульсного датчика могут поступать сигналы 2 типов. Режим счета, используемый для высокоскоростного счетчика 0, будет зависеть от типа сигнала.

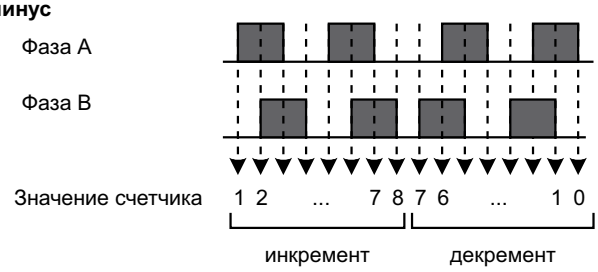
Режим плюс/минус

Входом служит двухфазный сигнал с разностью фаз 4X (фаза A и фаза B) и сигнал фазы Z. Значение счетчика инкрементируется или декрементируется в зависимости от разности двухфазных сигналов.

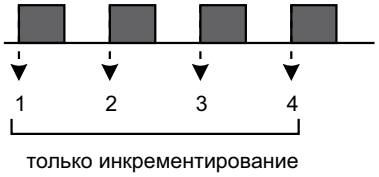
Режим инкрементальный

Входом служит однофазный сигнал и сигнал сброса счета. Значение счетчика только инкрементируется в соответствии с однофазным сигналом.

Режим плюс/минус



Режим инкрементальный



Замечание Один из методов, приведенных в следующем пункте, нужно всегда использовать для сброса счетчика при повторном запуске. Счетчик сбрасывается при пуске или останове исполнения программы.

Следующая последовательность сигналов рассматривается как увеличивающие (инкрементирующие) импульсы: Передний фронт фазы A - Передний фронт фазы B - задний фронт фазы A - задний фронт фазы B.

Следующая последовательность сигналов рассматривается как уменьшающие (декрементирующие) импульсы: Передний фронт фазы B - Передний фронт фазы A - задний фронт фазы B - задний фронт фазы A.

Диапазон счета -32 767..32 767 для режима плюс/минус и 0..65 535 для инкрементального режима. Сигналы можно считать с частотой до 2.5 кГц в режиме плюс/минус и до 5 кГц в инкрементальном режиме.

Режим плюс/минус всегда использует вход с разностью фаз 4X. Число импульсов для каждого оборота датчика = 4 x разрешающую способность датчика. Выбирайте датчик, исходя из диапазонов счета.

Методы сброса

Любой из 2 методов, описанных далее, можно выбрать для сброса текущего значения счетчика (т. е. его установки в 0).

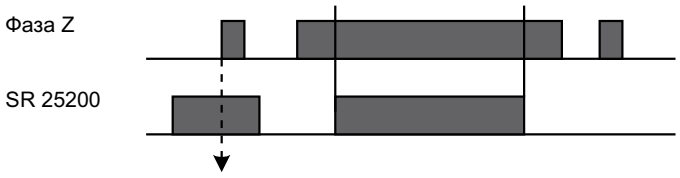
Сигнал фазы Z + сброс системной программой:

Текущее значение сбрасывается, когда сигнал фазы Z (вход сброса) установится в 1 после установки бита сброса высокоскоростного счетчика 0 (SR 25200) в 1.

Сброс системной программой:

Текущее значение сбрасывается когда бит сброса высокоскоростного счетчика 0 (SR 25200) установится в 1.

Сброс фазой Z + сброс системной программой



Замечание Бит сброса высокоскоростного счетчика 0 (SR 25200) обновляется один раз каждый цикл, так что для того, чтобы надежно его прочитать, он должен быть включен в 1 минимум на один цикл.

Z в обозначении “фаза Z” значит zero (нуль). Этот сигнал показывает, что датчик совершил один цикл.

Счет прерываний высокоскоростного счетчика

Для нулевого высокоскоростного счетчика при применении сравнивающих таблиц вместо счета вызывается прерывание. Проверка счета может проводится любым из двух методов, описанных далее. В таблице сравнений сохраняются условия сравнения (для сравнения с текущим значением) и комбинации программ прерываний.

Конечное значение:

В таблице сравнений можно хранить до 16 условий сравнений (конечное значение + направление счета) и комбинаций программ прерываний. Когда текущее значение счетчика и направление счета удовлетворяют условиям сравнения, выполняется заданная программа прерываний.

Сравнение с зоной:

До 8 условий сравнения (верхняя и нижняя границы) хранить и комбинаций программ прерываний хранятся в таблице сравнений. Когда текущее значение станет больше либо равным верхней границе или меньше либо равным нижней границе, выполняется заданная программа прерываний.

Подключение

В зависимости от режима счета входные сигналы с импульсного датчика должны подключаться к входам ЦУ следующим образом:

Клемма	Режим плюс/минус	Режим инкремента
4	Фаза A	Счетный вход
5	Фаза B	-
6	Фаза C	Вход сброса

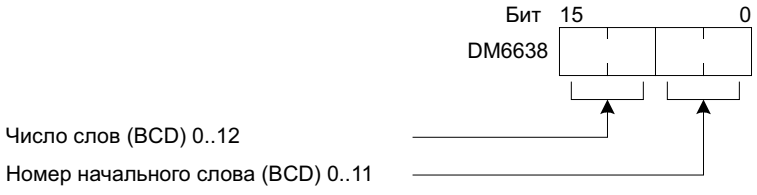
Если используется только сброс системной программой, вход 6 можно использовать как обычный вход. Если используется режим инкремента, вход 5 можно использовать как обычный вход.

Установочные параметры

При использовании прерываний высокоскоростного счетчика 0 перед выполнением программы произведите следующую настройку установочных параметров в режиме PROGRAM.

Параметры слова обновления входов (DM 6638)

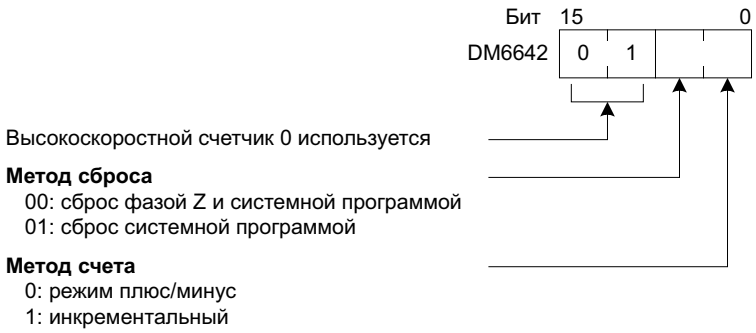
Производите данные настройки в случае необходимости обновления входов. Настройки те же, что и для интервального таймера 2.



По умолчанию: нет обновления входов

Параметры высокоскоростного счетчика 0 (DM 6642)

Если не сделаны данные настройки, высокоскоростной счетчик 0 нельзя использовать в программе.



По умолчанию: высокоскоростной счетчик 0 не используется

Изменения в DM6642 вступают в силу только при включении питания или начале отработки программы.

Программирование:

Для программирования высокоскоростного счетчика 0 проделайте следующие операции.

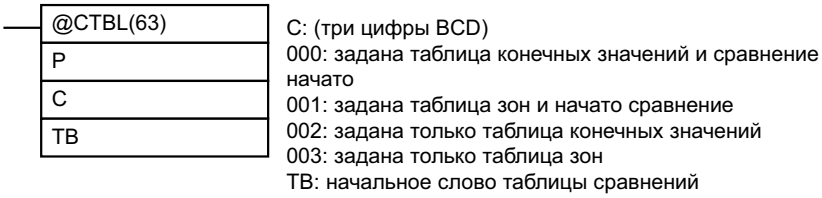
Высокоскоростной счетчик 0 начнет операцию счета при правильной настройке УППК, но сравнения с таблицей сравнений и вызов прерываний начнется только после выполнения команды CTBL(63).

Высокоскоростной счетчик 0 сброшен в 0 при включении питания и при пуска программы.

Текущее значение высокоскоростного счетчика 0 содержится в SR 230..SR 231.

Управление прерываниями высокочастотного счетчика 0

1, 2, 3,... 1. Используйте команду CTBL(63) для сохранения таблицы сравнения в CQM1 и начала сравнения.

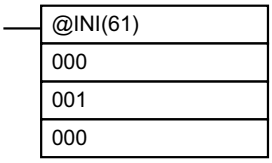


Если задано C = 0000, сравнение будет производится методом соответствия конечных значений, если 0001 - методом сравнения зон. Таблица сравнения будет сохранена и сравнение начнется после завершения операции сохранения. Во время выполнения сравнения высокочастотные прерывания будут выполняться согласно таблицы сравнений. Подробности о содержании сохраненных таблиц сравнений см. объяснения к команде CTBL(63) в Гл. 5 Набор команд.

Замечание При сравнении методом зон результаты сравнения нормально хранятся в AR 1100..AR 1107.

При задании C = 002 сравнение будет производиться методом соответствия конечных значений, если C = 003 - методом сравнения с зоной. В обоих случаях таблица сравнений будет сохранена, но сравнение начнется только после команды INI(61) - начать сравнение.

2. Для прекращения сравнений выполните команду INI (61), как показано ниже:



Для возобновления сравнения задайте второй операнд “000” (выполнить сравнение) и выполните команду INI (61).

Если таблица сохранена, она останется в CQM1 во время работы (т.е. исполнения программы) до загрузки другой таблицы.

Чтение текущего значения (PV)

Есть 2 способа чтения текущего значения.

читать из SR 230..SR 231

командой PRV(62)

Чтение SR 230 и SR 231

Текущее значение высокоскоростного счетчика 0 хранится в SR 230 и SR 231, как показано ниже. Левая цифра = F для отрицательных значений.

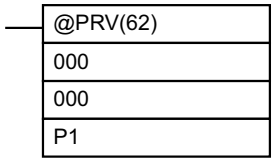
4 левых разряда	4 правых разряда	Режим плюс/минус	Режим инкремент
SR 231	SR 230	F0032767..00032767 (-32767)	до 00065535

Замечание Данные слова обновляются только один раз каждый цикл, так что могут быть отличия от фактических текущих значений.

Когда высокоскоростной счетчик 0 не используется, биты данных слов можно использовать как рабочие биты.

Применение команды PRV(62)

Читайте текущее значение высокоскоростного счетчика 0 командой PRV(62).



P1: первое слово текущего значения

Текущее значение высокоскоростного счетчика 0 хранится в следующем виде. Самая левая цифра = F для отрицательных чисел.

4 левых разряда	4 правых разряда	Режим плюс/минус	Режим инкремент
P1+1	P1	F0032767 до 00032767 (-32767)	до 00065535

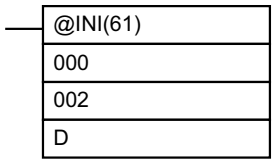
Текущее значение читается, когда команда PRV(62) уже выполнена.

Изменение текущего значения

Есть 2 метода изменения текущего значения высокоскоростного счетчика 0:

Сброс методами сброса (В данном случае новое текущее значение = 0).

Применение команды INI(61).



D: ведущее слово для сохранения текущего значения

Здесь объясняется применение команды INI(61). Для объяснения метода сброса см. описание высокоскоростного счетчика 0.

левых цифры	правых цифры	Режим плюс/минус	Режим инкремент
D1+1	D1	F0032767 до 00032767 (-32767)	до 00065535

Для задания отрицательного числа задавайте F в самой левой цифре.

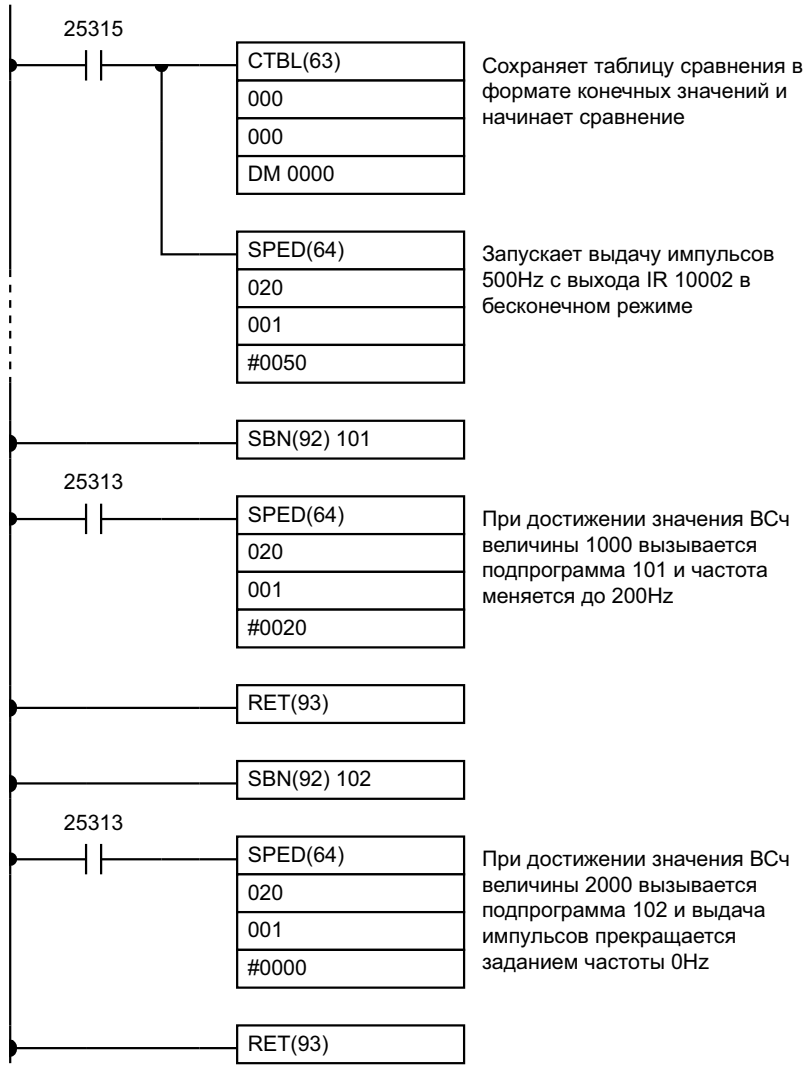
Пример применения

В данном примере приведена программа использования высокоскоростного счетчика 0 в режиме инкремент, сравнение методом конечного значения и изменение частоты выдачи импульсов в зависимости от текущего значения счетчика. Перед выполнением программы настройки установочные параметры следующим образом:

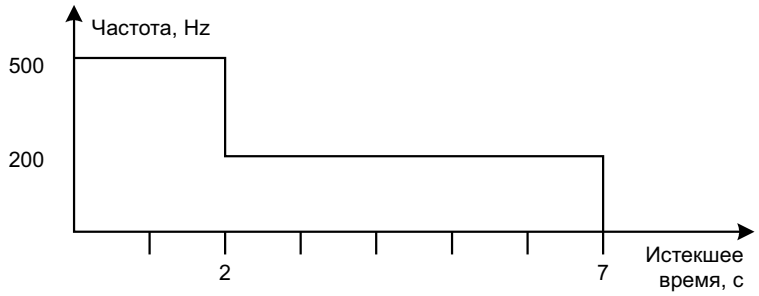
DM 6642: 0114 (Высокоскоростной счетчик 0 используется со сбросом от системной программы и работа в режиме инкремент). Для всех остальных параметров используйте значения по умолчанию. (Входы не обновляются во время обработки прерываний, выдача импульсов производится на IR 100).

Кроме того, в таблице сравнения хранятся следующие данные:

DM 0000	0002	Число условий сравнения: 2
DM 0001	1000	Конечное значение 1: 1000
DM 0002	0000	
DM 0003	0101	Номер программы обработки прерывания сравнения 1: 101
DM 0004	2000	Конечное значение 2: 2000
DM 0005	0000	
DM 0006	0102	Программа обработки прерывания сравнения 2: 102



При выполнении программы порядок работы будет иметь следующий вид:



1.4.6 Переполнение “+”/ Переполнение “- “ высокоскоростного счетчика 0

Если текущее значение высокоскоростного счетчика 0 вышло за заданные границы, появляется состояние переполнение + или переполнение - и текущее значение будет 0FFF FFFF для переполнения + и FFFF FFFF для переполнения - останется до стирания командой сброса. Допустимые значения:

- для режима плюс/минус: F003 2767..0003 2767
- для инкрементального режима 0000 0000..0006 5535

Замечание 1. Вышеприведенные значения являются теоретическими и принимают меньшие значения. Фактически значения будут равны значениям в цикле перед переполнением.

2. Цифры 6-я и 7-я текущего значения нормально 00, но могут использоваться как флаги переполнения + или - при операциях обнаружения значений, вышедших за заданные границы.

Высокоскоростной счетчик 0 можно сбросить как описано в предыдущем пункте или он будет сброшен автоматически при перезапуске программы. Высокоскоростной счетчик 0 и сопутствующие операции не будут работать нормально, пока не произвести очистку переполнения +/-. При переполнении +/- операции будут производится следующим образом:

- Операция с таблицей сравнения будет остановлена.
- Таблица сравнения не будет очищена.
- Программы прерывания высокоскоростного счетчика не будут отрабатываться.
- CTBL(63) можно использовать только для регистрации таблицы сравнения. При попытке начать операцию с таблицей сравнения операция не начнется и таблица не зарегистрируется.
- Нельзя пользоваться командой INI(61) для пуска или останова операций с таблицей сравнения или для изменения текущего значения.
- PRV(62) прочитает только 0FFF FFFF или FFFF FFFF как текущее значение.

Восстановление

Для выхода из состояния переполнения +/- проделайте операции:

С зарегистрированной таблицей сравнения

1, 2, 3,... 1. Произведите сброс счетчика.

2. Если необходимо, установите текущее значение командой PRV(62).

3. Если необходимо, задайте таблицу сравнения командой CTBL(63).

4. Запустите операции с таблицей сравнения командой INI(61).

Без зарегистрированной таблицы сравнения

1, 2, 3,... 1. Произведите сброс счетчика.

2. Если необходимо, установите текущее значение командой PRV(62).

3. Задайте таблицу сравнения и запустите операцию командами CTBL(63) и INI(61).

Замечание Результаты сравнения с зоной в AR 11 останутся после восстановления. Программа прерываний для условий, которые выполняются сразу после

восстановления, выполняться не будет, если условия уже выполнялись перед появлением состояния переполнения +/-. Если необходимо выполнить программу прерывания, перед отработкой очистите AR 11.

Операция сброса

Когда высокоскоростной счетчик 0 сброшен, текущее значение PV будет = 0, счет начнется от 0 и будут сохранены таблица сравнения, состояние исполнения и результаты исполнения.

Состояние пуска счетчика

Когда высокоскоростной счетчик 0 запущен, в установочных параметрах будет читаться и использоваться режим счета, Текущее значение PV будет установлено в 0, состояние переполнения +/- очищено, таблица сравнения зарегистрирована и состояние исполнения будет очищено, результаты сравнения с зоной очищены. (Результаты сравнения с зоной всегда очищаются в начале работы или при регистрации таблицы сравнений).

Состояние остановленного счетчика

Когда высокоскоростной счетчик 0 остановлен, текущее значение сохраняется, состояние регистрации таблицы сравнения и состояние работы будет очищено, результаты сравнения с зоной будут сохранены.

1.4.7 Прерывания высокоскоростных счетчиков (ВСч) 1 и 2 (CQM1-CPU43-E)

Импульсные сигналы с импульсного датчика на порты 1 и 2 CQM1-CPU43-EV1 можно подсчитывать с высокой скоростью, и обработка прерываний может выполняться согласно значениям подсчета.

Оба порта могут работать раздельно. Счетчик порта 1 далее называется ВСч 1, счетчик порта 2 называется ВСч 2. В данном пункте описано использование ВСч 1 и 2. Информацию об аппаратной части и подключении см. Инструкцию по работе CQM1.

Замечание 1. ВСч 1 и 2 могут работать только с CQM1-CPU43-EV1.

2. Некоторые команды нельзя использовать при настройке параметра DM 6611 на режим высокоскоростного счетчика.

Значение DM 6611	**“Пострадавшие” команды**
Режим ВСч (0000)	PLS2(-) и режим 0 команды ACC(-) нельзя использовать
Режим выдачи импульсов (0001)	CTBL(63) нельзя использовать с портами 1 и 2.

Отработка

Входные сигналы и режимы счета типа сигналов могут служить входами портов 1 и 2. Режимы счета, используемые для ВСч 1 и 2, задаются соответственно в DM 6643 и DM 6644.

1, 2, 3,... 1. Режим разности фаз (Диапазон счета - 25 кГц):

Входом служит двухфазный сигнал с разностью фаз 4X (фаза A и фаза B) и сигнал фазы Z . Счетчик инкрементируется или декрементируется в зависимости от разности двухфазных сигналов. Данный режим аналогичен режиму плюс\минус ВСч 0.

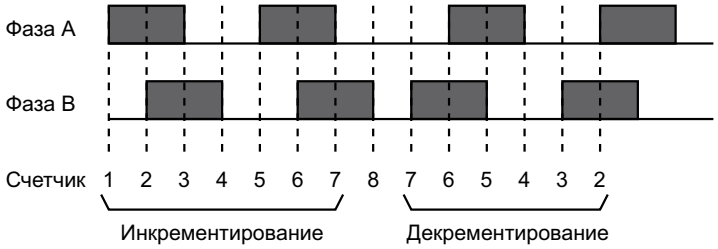
2. Режим импульс/направление (Диапазон счета - 50 кГц).

Фаза A служит сигналом направления, а .фаза B - счетным импульсом. Значение инкрементируется при фазе A = 0 и декрементируется при фазе A = 1.

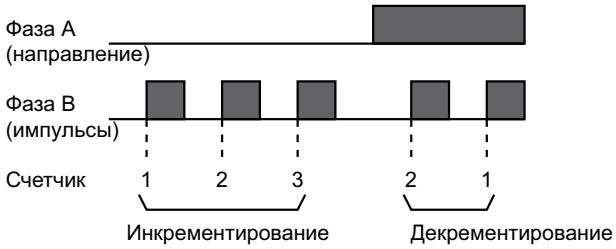
3. Режим плюс/минус (Диапазон счета - 50 кГц).

Фаза A служит декрементирущим сигналом, а .фаза B - инкрементирущим. Значение счетчика декрементируется при поступлении импульса на фазу A и инкрементируется при поступлении импульса на фазу B.

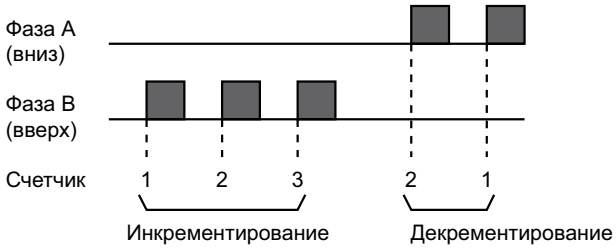
Режим разности фаз



Режим импульс/направление



Режим плюс/минус



Режимы счета

Режимы счета (кольцевой или линейный) для ВСч 1 и 2 задаются соответственно в DM 6643 и DM 6644.

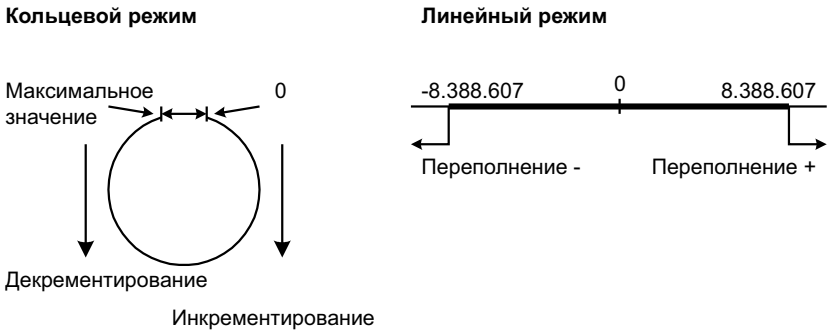
1, 2, 3,... 1. Кольцевой Режим:

В кольцевом режиме максимальное значение счетчика + 1 задается в CTBL(63). Счетчик перейдет из максимума в 0 при инкрементировании и из 0 на максимальное значение при декрементировании. Отрицательных значений нет.

Число точек кольца (максимальное значение + 1) можно задать от 1 до 65000.

2. Линейный Режим:

В линейном режиме диапазон счета от -8 388 607 до 8 388 607. При выходе за допустимые значения ВСч 1 или 2 появляется состояние переполнения (+ или -), текущее значение остается 0838 8607 (переполнение +) или F838 8607 (переполнение -), счет или сравнение прекратятся и AR 0509 (порт 1) или AR 0609 (порт 2) включатся в 1.



Замечание 1. Один из методов, описанных в данной главе, следует всегда применять для сброса счетчика при его повторном запуске. Счетчик автоматически сбросится при пуске или останове отработки программы.

2. Для увеличивающегося (инкрементального) счета используется последовательность импульсов: Передний фронт фазы A - Передний фронт фазы B - задний фронт фазы A - задний фронт фазы B. Для уменьшающегося (декрементального) счета используется последовательность импульсов: Передний фронт фазы B - Передний фронт фазы B - задний фронт фазы B - задний фронт фазы A.

Методы сброса

Для сброса текущего значения счетчика (т. е. его установки в 0) можно выбрать любой из 2 методов: Сигнал фазы Z + сброс системной программой или Сброс системной программой. Данные сбросы работают аналогичном образом, как и для высокоскоростного счетчика 0 (подробности см. 1-4-5).

Замечание 1. Биты сброса для ВСч 1 и 2 (SR 25201 и SR 25202) обновляются один раз каждый цикл. Следите за тем, чтобы бит сброса был 1 как минимум 1 полный цикл, чтобы его надежно прочитать.

2. Таблица сравнений, состояние исполнения и результаты сравнения с зоной при сбросе будут сохранены. (Сравнение продолжится после выполнения сброса).

Счет прерываний высокоскоростного счетчика

Таблицы сравнения, используемые для ВСч 1 и 2 аналогичны таблицам для ВСч 0. (Подробности См. 1-4-5).

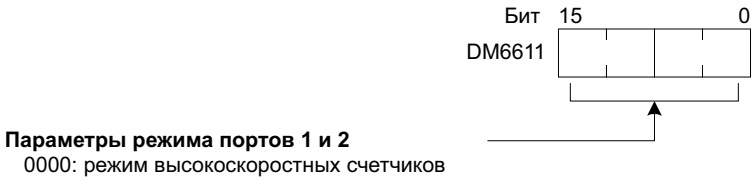
Установочные параметры

При использовании прерываний ВСч 1 и 2 перед выполнением программы произведите следующую настройку установочных параметров в режиме PROGRAM.

Параметры режимов портов 1 и 2 (DM 6611)

Задайте для портов 1 и 2 режим высокоскоростных счетчиков.. Если не задать режим высокоскоростных счетчиков, CTBL(63) нельзя использовать для счета со сравнениями.

Данная настройка читается только при включении ПК. Если они были изменены, следует включить и выключить ПК перед исполнением программы.

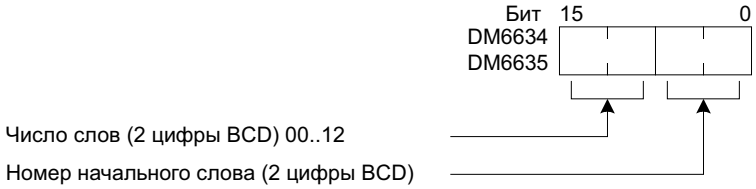


По умолчанию: режим высокоскоростных счетчиков

Замечание Если DM 6611 установлен в режим выдачи импульсов, для сравнения текущего значения ВСч 1 и 2 может использоваться другая команда сравнения, такая как BCMP(68) .

Параметры слов обновления входов (DM 6634 и DM 6635)

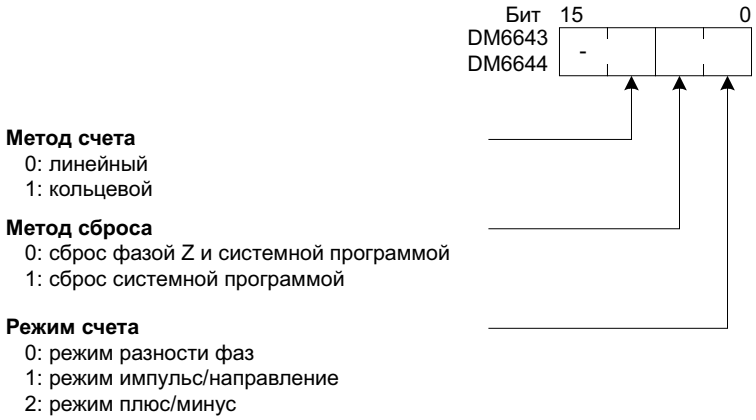
DM 6634 содержит слово обновления входов для ВСч 1, DM 6635 содержит параметры для ВСч 2. Производите данные настройки в случае необходимости обновления входов.



По умолчанию: нет обновления входов

Параметры высокоскоростных счетчиков 1 и 2 (DM 6643 и DM 6644)

DM 6634 содержит параметры для ВСч 1, DM 6634 содержит параметры для ВСч 2. Данные установочные параметры определяют рабочие параметры для этих Всч.



По умолчанию: линейный метод, сброс фазой Z и системной программой, режим разности фаз

Программирование

Для программирования высокоскоростных счетчиков 1 и 2 проделайте следующие операции.

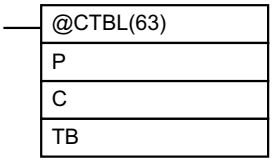
Высокоскоростные счетчики 1 и 2 начнут счет при правильной настройке установочных параметров, но сравнения с таблицей сравнений и вызов прерываний начнется только после выполнения команды CTBL(63).

Высокоскоростные счетчики 1 и 2 сброшены в 0 при включении питания и после пуска или останова программы.

Текущие значения высокоскоростного счетчика 1 содержатся в SR 232 и SR 233, текущие значения высокоскоростного счетчика 2 содержатся в SR 234 и SR 235.

Управление прерываниями высокочастотных счетчиков 1 и 2

1, 2, 3,... 1. Используйте команду CTBL(63) для сохранения таблицы сравнения в CQM1 и начала сравнения.



P: порт
001: порт 1
002: порт 2
C: (3 цифры BCD)
000: задать таблицу конечных значений и начать сравнение
001: задать таблицу зон и начать сравнение
002: задать только таблицу конечных значений
003: задать только таблицу зон
TB: начальное слово таблицы сравнений

Если задано C = 0000, сравнение будет производится методом конечных значений, если 0001 - методом сравнения с зоной. Таблица сравнения будет сохранена и сравнение начнется после завершения операции сохранения.

Во время выполнения сравнения высокочастотные прерывания выполняются согласно таблицы сравнений. Подробности о содержании таблицы сравнений см. объяснения к команде CTBL(63) в Гл. 5 Набор команд.

Замечание 1. При сравнении с зоной результаты сравнения хранятся в AR 1100..AR 1107.

При задании C = 002 сравнение будет производиться методом конечных значений, если C = 003 - методом сравнения с зоной. В обоих случаях таблица сравнений будет сохранена, но сравнение начнется только после команды INI(61).

2. Для прекращения сравнения выполните команду INI (61) следующим образом. Задавайте P = 001 для порта 1 и P = 002 для порта 2.

@INI(61)
P
001
000

Для возобновления сравнения задайте второй операнд = 000 (выполнить сравнение) и выполните команду INI (61).

Сохраненная таблица хранится в CQM1 во время работы до загрузки другой таблицы.

Чтение текущего значения.

Есть 2 способа читать текущее значение.

- читать из SR 232 и SR 233 (порт 1) или SR 234 и SR 235 (порт 2).
- командой PRV(62).

Чтение SR 232 и SR 233 или SR 234 и SR 235

Текущее значение высокоскоростного счетчика 1 хранится в SR 232 и SR 233, текущее значение высокоскоростного счетчика 2 хранится в SR 234 и SR 235.

В линейном режиме для отрицательных значений левая цифра = F.

	4 левых разряда	**4 правых разряда**	**Линейный режим**	**Кольцевой режим**
Порт 1	SR 233	SR 232	F838 8607 до 0838 8607 (-8 388 607 до 8 388 607)	000 до 0000 6499
Порт 2	SR 235	SR 234		

Замечание Данные слова обновляются только один раз за каждый цикл, так что могут быть отличия от фактических текущих значений.

Применение команды PRV(62)

Читайте текущее значение высокоскоростных счетчиков 1 и 2 командой PRV(62).

Задайте для порта 1 = 001 и для порта 2 = 002.

@PRV(62)
P
000
P1

P: порт
001: потр 1
002: порт 2
P1: ведущее слово текущего значения

Текущее значение заданного высокоскоростного счетчика 1 и 2 хранится в следующем виде. При линейном методе самое левое слово = F для отрицательных значений.

4 левых разряда	**4 правых разряда**	**Линейный метод**	**Кольцевой метод**
D+1	D	F838 8607 до 0838 8607 (-8 388 607 до 8 388 607)	0000 до 0000 6499

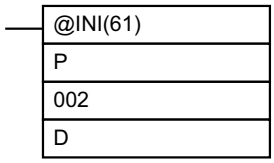
Текущее значение читается, когда команда PRV(62) выполнена.

Изменение текущего значения

Есть 2 метода изменения текущего значения ВСч 1 и 2: сброс методами сброса (В данном случае текущее значение сбрасывается в 0) или командой INI(61).

Здесь объясняется метод применения команды INI(61). Для объяснения метода сброса см. начало данного описания высокоскоростных счетчиков 1 и 2.

Измените текущее значение командой INI(61).



P: порт
001: порт 1
002: порт 2
D: ведущее слово для сохранения значения текущего состояния

4 левых разряда	4 правых разряда	Линейный метод	Кольцевой метод
D+1	D	F838 8607 до 0838 8607 (-8 388 607 до 8 388 607)	0000 до 0000 6499

Для задания отрицательного значения в линейном режиме задайте левую цифру F.

Состояние высокоскоростного счетчика

Состояние ВСч 1 и 2 можно определить либо чтением флагов из области AR, либо командой PRV(62).

В следующей таблице представлены соответствующие флаги AR и их функции.

Слово	Бит(ы)	Функция
AR 04	00 - 15	Показывает состояние высокоскоростного счетчика. 00: Нормально 01, 02: Ошибка аппаратуры 03: Ошибка установочных параметров ПК
AR 05	00 - 07	Флаг результатов сравнения ВСч 1 для зон 1..8. 0: не в зоне 1: в зоне)
	08	Флаг сравнения ВСч 1 0: остановлено 1: идет сравнение)
	09	Флаг переполнения +/- ВСч 1 0: нормально 1: произошло переполнение + или -
AR 06	00 - 07	Флаг результатов сравнения ВСч 2 для зон 1..8. 0: не в зоне 1: в зоне)
	08	Флаг сравнения ВСч 2 0: остановлено 1: идет сравнение)
	09	Флаг переполнения +/- ВСч 2 0: нормально 1: произошло переполнение + или -

Состояние ВСч 1 и 2 можно также определить командой PRV(62). Задайте номер ВСч (1 или 2) и слово назначения D. Информация о состоянии порта будет записана по адресу D в биты 00..01, а биты 02..15 обнулятся.

Биты 00 и 01 слова D содержат информацию об указанном ВСч.

Бит	Функция
00	Флаг сравнения ВСч 1 0: остановлено 1: идет сравнение)
01	Флаг переполнения +/- ВСч 1 0: нормально 1: произошло переполнение + или -

Пример применения

В данном примере приведена программа, которая выдает стандартные импульсы с порта 1, считая их ВСч 0. ВСч работает в режиме плюс/минус, импульсы выдачи по ЧС инкрементируют счетчик (вход фазы В), импульсы выдачи против ЧС декрементируют счетчик (вход фазы А). Перед выполнением программы произведите следующие настройки и перезапустите ПК.

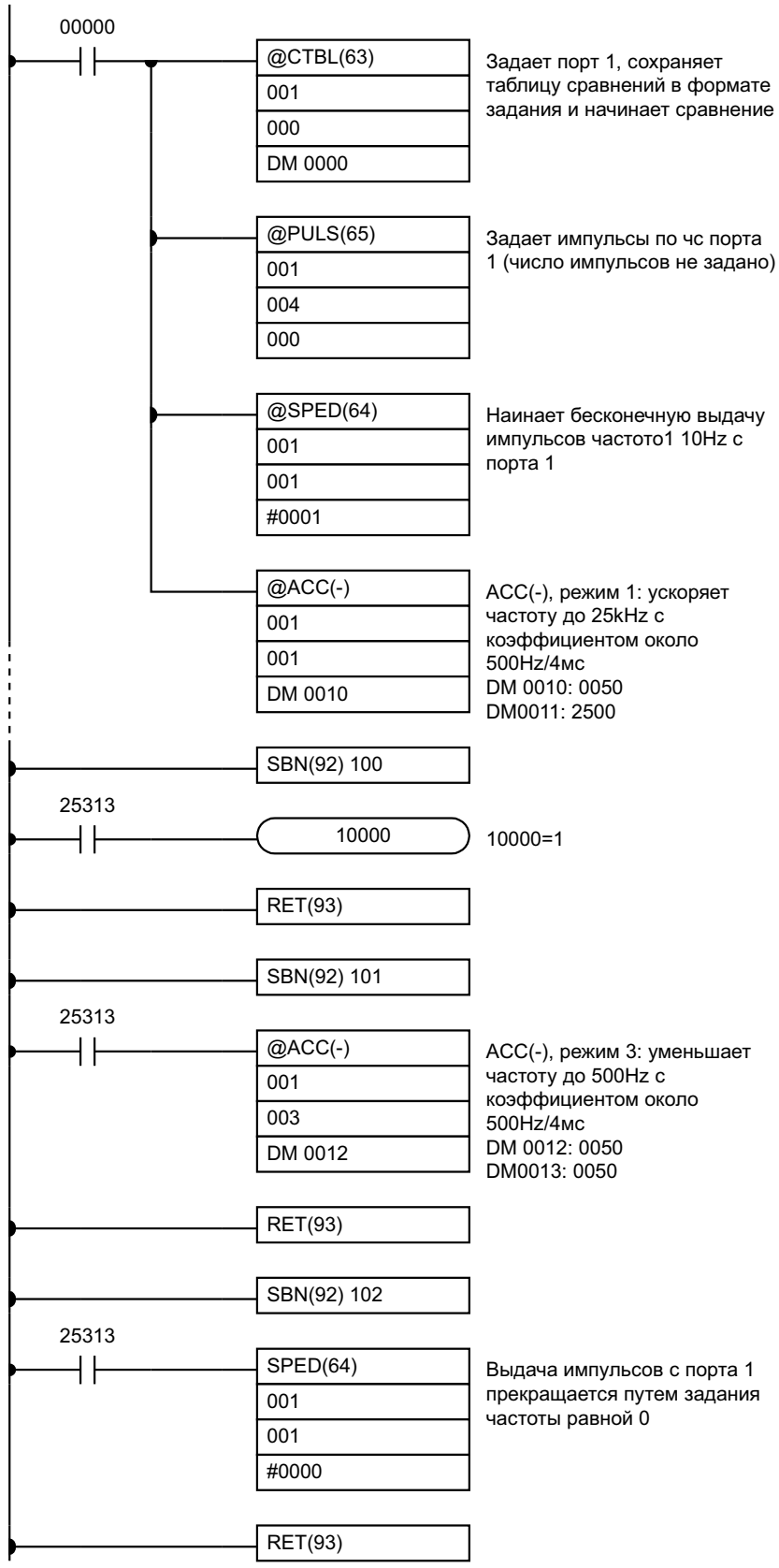
DM 6611: 0000 (режим В Сч).

DM 6643: 0002 (порт 1 : выдача стандартных импульсов, счет линейный, сброс фазой Z и системной программой, режим плюс/минус).

Для всех остальных настроек значения по умолчанию. (Входы не обновляются во время обработки прерываний).

Кроме того, в таблице сравнения хранятся следующие данные:

DM 0000	0003	Число условий сравнения: 3
DM 0001	2500	Конечное значение 1:2 500
DM 0002	0000	
DM 0003	0100	Номер программы обработки прерывания сравнения 1: 100
DM 0004	7500	Конечное значение 2: 7 500
DM 0005	0000	
DM 0006	0101	Номер программы обработки прерывания сравнения 2:101
DM 0007	0000	Конечное значение 3: 10 000
DM 0008	0001	
DM 0009	0102	Номер программы обработки прерывания сравнения 3: 102



1.4.8 Прерывания абсолютного высокоскоростного счетчика (CQM1-CPU44-E)

Сигналы в двоичном виде с абсолютного вращающегося датчика могут служить входами портов 1 и 2 CQM1-CPU44-EV1 и подсчитываться с частотой 4 кГц. Согласно полученным значениям можно выполнять обработку прерываний.

Два порта могут работать раздельно. Счетчик порта 1 далее называется АВСч 1, счетчик порта 2 - АВСч 2. В данной главе описывается применение абсолютных высокоскоростных счетчиков 1 и 2. Информацию об аппаратной части и подключении см. Инструкцию по работе CQM1.

Отработка

Входные сигналы и режимы счета

Есть два режима счета, которые можно использовать для АВСч 1 и 2. Режимы счета и разрешающая способность АВСч 1 и 2 задаются в установочных параметрах (DM 6643 и DM 6644).

1, 2, 3,... 1. Режим двоично-десятичный

Двоичный код вращающегося абсолютного датчика сначала преобразуется в обычное двоичное число и далее в двоично-десятичное.

2. Режим 360^0

Двоичный код вращающегося абсолютного датчика преобразуется в угловое значение ($0^0..359^0$) в зависимости от настройки разрешающей способности. (Настройки CTBL(63) сделаны с дискретой 5^0).

Разрешение	**Возможные текущие значения**	
	Режим BCD	**Режим 360^0**
8 бит	0..255	Текущее значение: $0^0..359^0$ (дискрета 1 град) Параметры таблицы сравнений: 0^0 .. 355^0 (дискрета 5^0)
10 бит	0..1023	
12 бит	0..4095	

Счет прерываний АВСч

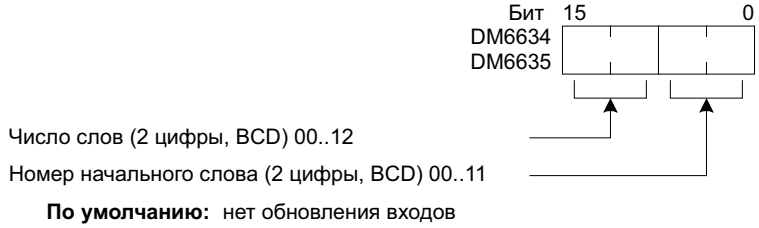
Текущие значение счетчика можно сравнивать с помощью до 16 конечных значений или 8 зон. См. 5-5-16 описание таблиц сравнения, используемых с АВСч 1 и 2.

Установочные параметры

При использовании прерываний АВСч 1 и/или 2 перед выполнением программы произведите следующую настройку установочных параметров ПК в режиме PROGRAM.

Параметры слов обновления входов (DM 6634 и DM 6635)

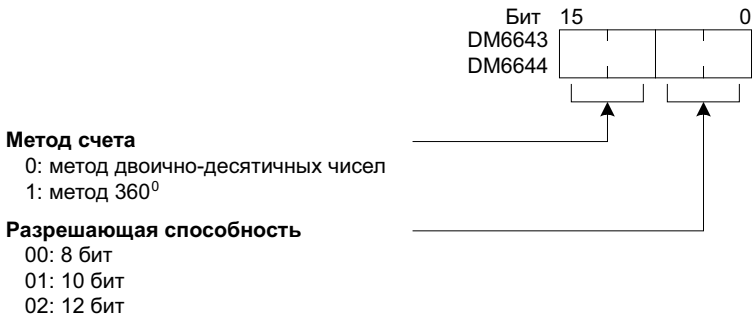
DM 6634 содержит параметры слова обновление для АВСч 1, DM 6635 содержит параметры слова обновление для АВСч 2. Производите данные настройки в случае необходимости обновления входов.



По умолчанию: нет обновления входов

Параметры АВСч (DM 6643 и DM 6644)

DM 6634 содержит параметры для АВСч 1, DM 6634 содержит параметры для АВСч 2. Данные слова определяют режимы счета и параметры разрешающей способности.



По умолчанию: метод двоично-десятичных чисел, разрешающая способность 8 бит

Коррекция исходного

Имеется возможность вносить коррекцию для смещения между исходным абсолютного поворотного датчика и фактическим исходным. Данную настройку можно проводить раздельно для портов 1 и 2.

Для задания коррекции исходного произведите следующие операции. После задания коррекции исходного данные, полученные с датчика положения, пересчитываются перед тем, как занестись в качестве текущего значения.

1, 2, 3,... 1. Установите абсолютный вращающийся датчик положения в желаемое положение исходного.

2. Убедитесь, что секция 1 переключателя DIP находится в положении OFF (разрешение периферийным устройствам переписывать DM 6614..DM 6655) и переключите ПК в режим PROGRAM.

3. Установите разрешающую способность в DM 6643 или DM 6644.

4. Проверьте, нет ли фатальной ошибки FALS 9C.

5. Прочтите текущее значение ВСч с IR 232 и IR 233 (порт 1) и IR 234 и IR 235 (порт 2) для определения текущего значения перед установкой коррекции исходного.

6. Включите в 1 флаг коррекции исходного SR 25201 (порт 1) или флаг коррекции исходного SR 25202 (порт 2) с периферийного устройства.

Значение коррекции будет записано в DM 6611 (порт 1) и DM 6612 (порт 2) и флаг коррекции исходного автоматически сбросится в OFF. Значение коррекции будет записано в двоично -..4095 независимо от того, в каком режиме установлен счетчик - двоично десятичным или 360^0.

7. Прочтите текущее значение ВСч для определения текущего значения после установки смещения исходного. После установки коррекции исходного текущее значение должно быть = 0.

Значение коррекции исходного останется действующим до изменения путем выполнения вышеописанной процедуры.

Программирование

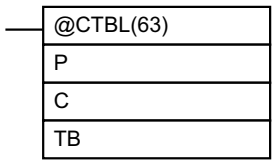
Для программирования абсолютных высокоскоростных счетчиков 1 и 2 проделайте следующие операции.

Абсолютные высокоскоростные счетчики 1 и 2 начнут счет при правильной настройке установочных параметров, но сравнения с таблицей сравнений и вызов прерываний начнется только после выполнения команды CTBL(63).

Текущее значение абсолютного высокоскоростного счетчика 1 содержатся в IR 232 и IR 233, текущие значения абсолютного высокоскоростного счетчика 1 содержатся в IR 234 и IR 235.

Управление прерываниями абсолютного высокоскоростного счетчика

1, 2, 3,... 1. Используйте команду CTBL(63) для сохранения таблицы сравнения в CQM1 и начала сравнений.



P: порт
001: порт 1
002: порт 2
C: (3 цифры BCD)
000: задать таблицу конечных значений и начать сравнение
001: задать таблицу зон и начать сравнение
002: задать только таблицу конечных значений
003: задать только таблицу зон
TB: начальное слово таблицы сравнений

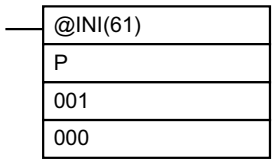
P определяет порт. Для задания АВСч 1 установите P = 001 или для задания АВСч 2 установите P = 002.

Если задано C = 0000, сравнение будет производится методом конечных значений, если 0001 - методом выхода за зону. Таблица сравнения будет сохранена и сравнение начнется после завершения операции сохранения. Во время выполнения сравнения высокочастотные прерывания выполняются согласно таблицы сравнений. Подробности о содержании таблицы сравнений см. объяснения к команде CTBL(63) в гл. 5-15-6.

Замечание При сравнении с зоной результаты сравнения хранятся в AR 1500..AR 0507 (порт 1) и в AR 0600..AR 0607 (порт 2).

При задании C = 002 сравнения будет производиться методом конечных значений, если C = 003 - методом сравнения с зоной. В обоих случаях таблица сравнений будет сохранена, но сравнение не начнется, для начала сравнения нужно выполнить команду INI(61).

2. Для прекращения сравнения выполните команду INI (61), как показано ниже. Задавайте P = 001 для порта 1 и P = 002 для порта 2.



Для возобновления сравнения задайте второй операнд = 000 (выполнить сравнение) и выполните команду INI (61).

Сохраненная таблица будет хранится в CQM1 во время работы (т.е. исполнении программы) до загрузки другой таблицы.

Чтение текущего значения

Есть 2 способа читать текущее значение.

- читать из IR 232 и IR 233 (порт 1) и IR 234 и IR 235 (порт 2).
- командой PRV(62).

Чтение IR 232 и IR 233 или IR 234 и IR 235

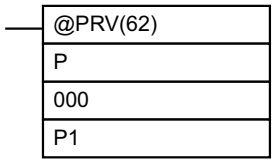
Текущее значение абсолютного высокоскоростного счетчика 1 хранится в IR 232..I R 233, текущее значение абсолютного высокоскоростного счетчика 2 хранится в IR 234..IR 235, как показано ниже:

	4 левых разряда	**4 правых разряда**	**Режим BCD**	**Режим 360^0**
Порт 1	IR 233	IR 232	0000 до 0000 4095	0000 до 0000 0359
Порт 2	IR 235	IR 234		

Замечание Данные слова обновляются только один раз за каждый цикл, так что данные значения могут отличаться от фактических текущих значений.

Применение команды PRV(62)

Читайте текущее значение высокоскоростного счетчика командой PRV(62). Задавайте АВСч 1 или 2 в P (P=001 или 002).



P: порт
001: потр 1
002: порт 2
P1: ведущее слово текущего значения

Текущее значение заданного абсолютного высокоскоростного счетчика хранится в следующем виде:

4 левых разряда	**4 правых разряда**	**Режим BCD**	**Режим 360^0**
D+1	D	0000 0000 до 0000 4095	0000 0000 до 0000 0359

Текущее значение читается когда команда PRV(62) уже выполнена.

Чтение состояния АВСч

Состояние АВСч 1 и 2 можно определить либо чтением соответствующих флагов из области AR, либо командой PRV(62).

В таблице представлены флаги области AR и их функции.

Слово	**Бит(ы)**	**Функция**
AR 04	00 - 15	Показывает состояние абсолютного высокоскоростного счетчика. 00: Нормально 01, 02: Ошибка аппаратуры 03: Ошибка установочных параметров ПК
AR 05	00 - 07	Флаг результатов сравнения ВСч 1 для зон 1..8. 0: не в зоне 1: в зоне)
	08	Флаг сравнения ВСч 1 0: остановлено 1: идет сравнение)
AR 06	00 - 07	Флаг результатов сравнения ВСч 2 для зон 1..8. 0: не в зоне 1: в зоне)
	08	Флаг сравнения ВСч 2 0: остановлено 1: идет сравнение)

Состояние флага сравнения АВСч 1 и 2 можно также определить командой PRV(62). Задайте номер АВСч 1 или 2 (P=001..002) и адрес слова приемника D. Флаг состояния (0: Остановлен, 1: сравнение) будет переписан в бит 00 слова D, а биты 02..15 обнулятся.

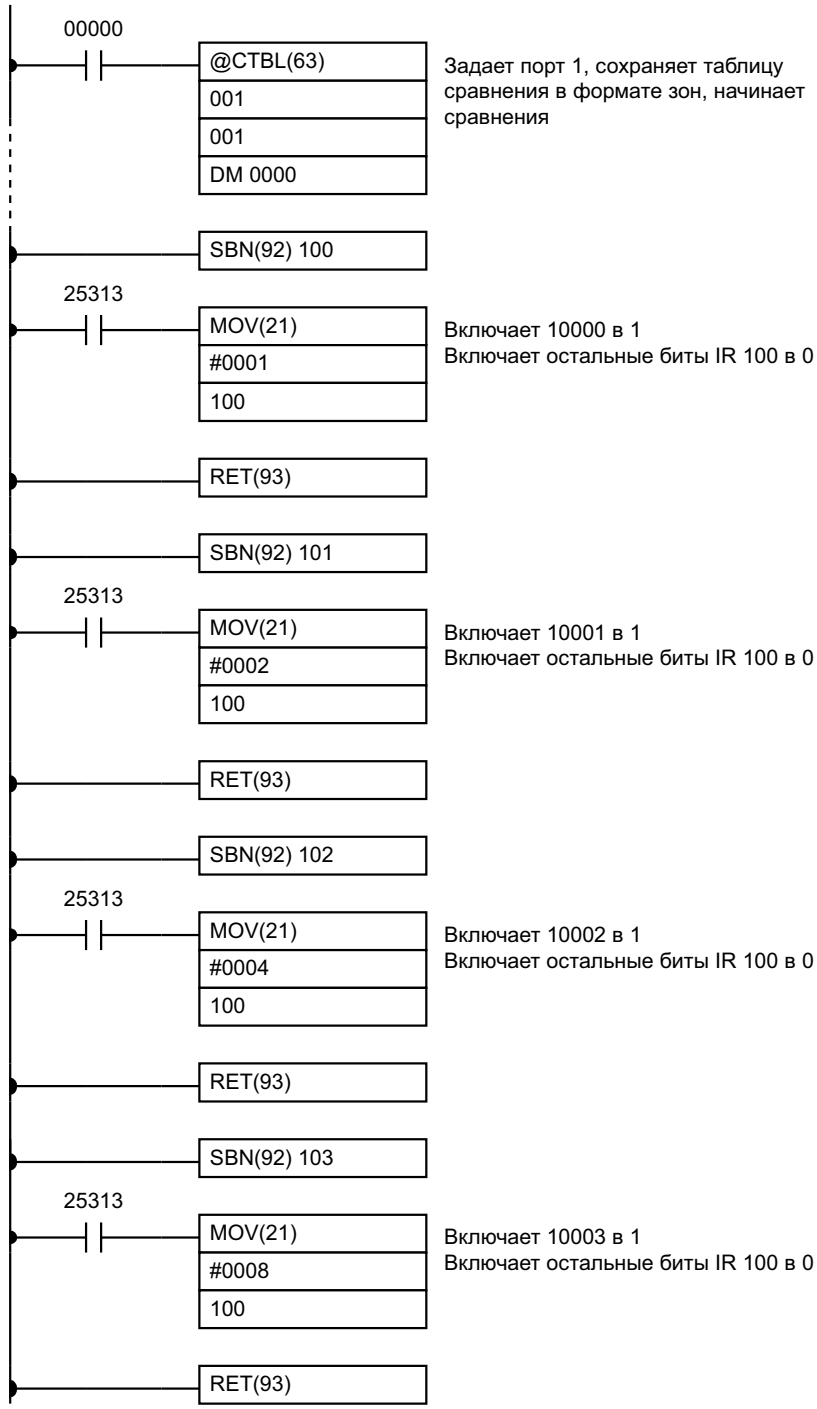
Пример применения

В данном примере приведена программа, которая получает входные сигналы абсолютного вращающегося датчика положения на порт 1, и использует их для управления выходами IR 10000..IR 10003. АВСч установлен с разрешающей способностью 8 бит и режим 360^0, сравнение с зоной. Перед выполнением программы задайте DM 6643 = 0100 (порт 1 : режим 360^0, разрешающая способность 8).

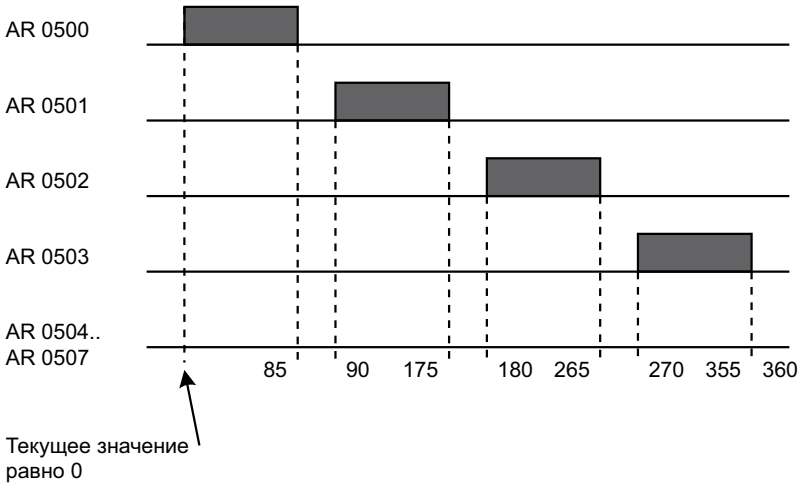
Для всех остальных параметров используйте значения по умолчанию. (Входы не обновляются во время обработки прерываний).

Кроме того, в таблицу сравнения загружены следующие данные:

DM 0000	0000	Нижняя граница #1 (0°)	Зона 1 (0 - 85°)
DM 0001	0085	Верхняя граница #2 (85°)	
DM 0002	0100	Номер подпрограммы: 100	
DM 0003	0090	Нижняя граница #1 (90 град.)	Зона 2 (90 - 175 град)
DM 0004	0175	Верхняя граница # (175°)	
DM 0005	0101	Номер подпрограммы: 101	
DM 0006	0180	Нижняя граница #1 (180°)	Зона 3 (180 - 265 град)
DM 0007	0265	Верхняя граница # (265°)	
DM 0008	0102	Номер подпрограммы: 102	
DM 0009	0270	Нижняя граница #1 (270°)	Зона 4 (270 - 355 град)
DM 0010	0355	Верхняя граница # (355°)	
DM 0011	0103	Номер подпрограммы: 103	
DM 0012	0000	Нижняя граница #1 (0°)	Зона 5 (Не используется)
DM 0013	0000	Верхняя граница # (0°)	
DM 0014	FFFF	Номера подпрограммы нет	
DM 0015	0000	Нижняя граница #1 (0°)	Зона 6 (Не используется)
DM 0016	0000	Верхняя граница # (0°)	
DM 0017	FFFF	Номера подпрограммы нет	
DM 0018	0000	Нижняя граница #1 (0°)	Зона 7 (Не используется)
DM 0019	0000	Верхняя граница # (0°)	
DM 0020	FFFF	Номера подпрограммы нет	
DM 0021	0000	Нижняя граница #1 (0°)	Зона 8 (Не используется)
DM 0022	0000	Верхняя граница # (0°)	
DM 0023	FFFF	Номера подпрограммы нет	



На следующей диаграмме показана зависимость между текущим значением АВСч 1 и флагами результатов сравнения зон AR 0500..AR 0507 при выполнении этой программы.


AR 0500
AR 0501
AR 0502
AR 0503
AR 0504..
AR 0507
85
90
175
180
265
270
355
360
Текущее значение
равно 0

1.5 Функции прерываний CPM1/CPM1A

В данной главе описаны установочные параметры и методы применения функций прерываний CPM1/CPM1A.

1.5.1 Типы прерываний

CPM1/CPM1A имеет 3 типа прерываний, описываемых далее:

Входные прерывания

CPM1/CPM1A может иметь 2 или 4 входа с прерываниями. Обработка прерываний производится при включении одного из этих входов в 1 из внешнего источника.

Прерывания интервального таймера

Обработка прерываний выполняется по интервальному таймеру с дискретностью 0.1 мс.

Прерывания высокоскоростного счетчика

ВСч считает входные импульсы на одном из входных битов ЦУ 00000..00002. Обработка прерываний выполняется, когда значение счета достигает заданного значения встроенного ВСч.

Приоритет прерываний

При вызове прерывания выполняется заданная программа обработки прерываний. Приоритет обработки прерываний имеет следующий вид:

Входные прерывания >интервальные прерывания > прерывания ВСч.

Когда при обработке прерывания происходит прерывание с более высоким приоритетом, текущий процесс будет прекращен и вместо этого начинается обработка нового прерывания. После полного завершения этой программы продолжится обработка предыдущего прерывания.

Когда во время обработке прерывания происходит прерывание с низшим или равным приоритетом, новое прерывание будет обрабатываться как только текущая программа полностью отработается.

Когда одновременно происходят 2 прерывания одного приоритета, они выполняются в следующем порядке:

Входное прерывание 0 > Входное прерывание 1 > Входное прерывание 2 > Входное прерывание 3 > Интервальные прерывания > Прерывания ВСч.

Предосторожности в программе прерываний

При пользовании программ прерываний соблюдайте следующие меры предосторожности:

1, 2, 3,... 1. Новое прерывание можно задать внутри программы прерывания. Более того, прерывание можно стереть из программы прерывания.

2. Другую программу нельзя написать в программе прерывания.

3. Подпрограмму прерываний нельзя записать внутри программы прерывания. Не записывайте команду SBN(92) (определить подпрограмму) в программе прерываний.

4. Программу прерываний нельзя записать внутри подпрограммы. Не записывайте программу прерываний между командой ОПРЕДЕЛИТЬ ПОДПРОГРАММУ (SBN(92)) и команду ВОЗВРАТ (RET(93)).

Входы, используемые как входы прерываний, нельзя использовать как обычные входы.

Команды и прерывания ВСч

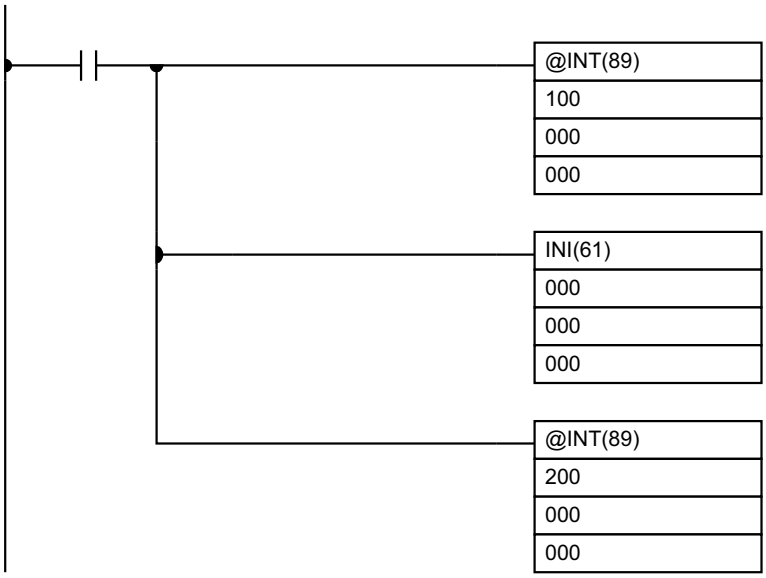
Следующие команды нельзя выполнять в подпрограмме прерываний во время обработки в главной программе команд управления высокоскоростными счетчиками:

INI(61), PRV(62), CTBL(63)

Для обхода данных ограничений можно воспользоваться следующими методами:

Метод 1

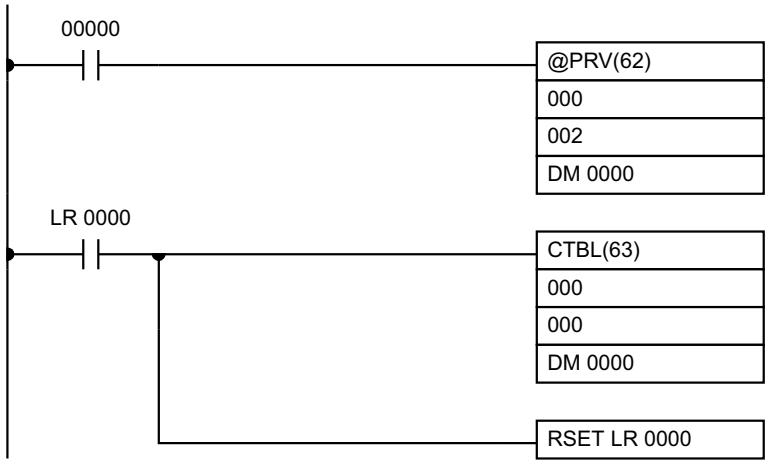
Обработку всех прерываний можно замаскировать во время выполнения команды.



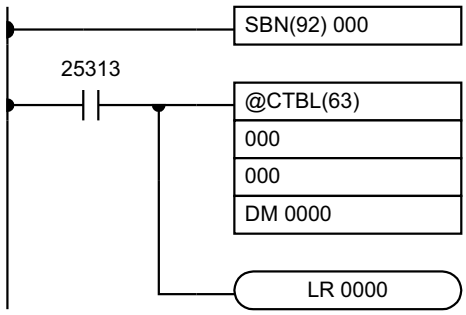
Метод 2

Повторно выполнить команду в главной программе

1, 2, 3,... 1. Секция в главной программе:



2. Секция в подпрограмме прерываний:



Замечание 1. Определяйте программы прерываний в конце главной программы командами SBN(92) и RET(93), как обычные подпрограммы.

2. При задании программы прерывания, во время операции контроля программы может появиться сообщение “SBS UNDEFD”, но программа будет выполняться нормально.

1.5.2 Входные прерывания

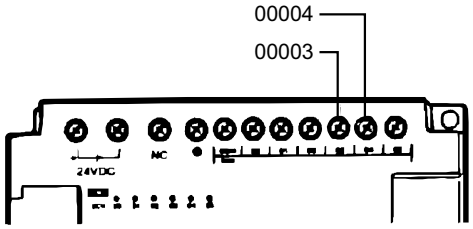
Блоки CPU на 10 точек имеют 2 входа прерываний (00003 и 00004)

Блоки CPU на 20, 30, 40 точек имеют 4 входа прерываний (00003..00006).

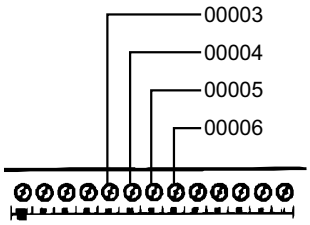
Есть 2 режима обработки входных прерываний: Режим прерывания по входу и режим счетчика.

CPM1

CPM1 10CDR

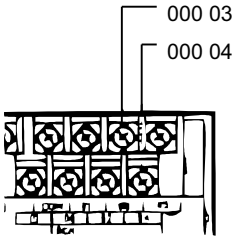


CPM1 20CDR /CPM1 30CDR

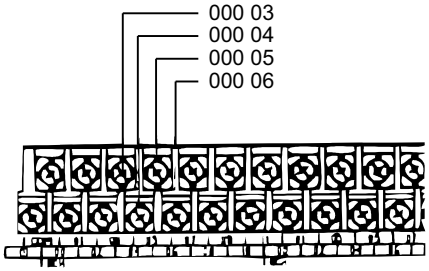


CPM1A

Модули на 10 точек
CPM1A-10CD -



Модули на 20, 30, 40 точек
CPM1A-20CD - ,
CPM1A-30CD - ,
CPM1A-40CD -



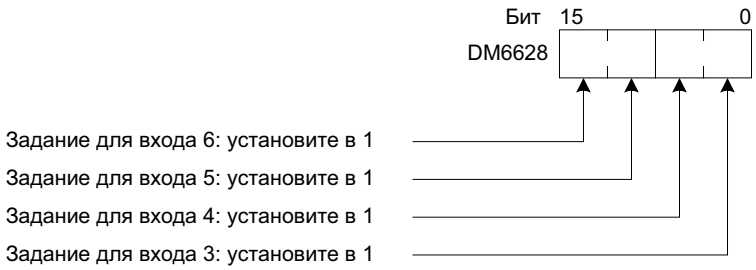
Модель ПК	Вход	Номер прерывания	Время реакции	
			Режим прерывания	Режим счетчика
CPM1-10CDR-_ CPM1A-10CDR-_	0003	00	0.3 ms max (время до выполнения программы прерывания	1 kHz
	0004	01		
CPM1-20CDR-_ CPM1A-20CDR-_	0003	00		
	0004	01		
CPM1-30CDR-_ CPM1A-30CDR-_	0003	02		
CPM1A-40CDR-_	0004	03		

Замечание Если входные прерывания не используются, используйте входы 00003..00006 как обычные входы.

Установочные параметры входных прерываний

Если предполагается использование входов 00003..00006 в качестве входов прерываний, они должны быть заданы как входы прерываний в DM 6628. Задайте соответствующую цифру в 1, если предполагается использование входа в качестве входа прерываний (по входу или в режиме счета). Задайте 0, если бит должен использоваться в качестве обычного входа.

Слово	Значение параметра
DM 6628	0: Обычный вход (по умолчанию) 1: Вход прерывания 2: Вход быстрого реагирования



Подпрограммы прерываний

Прерываниям со входов 00003..00006 отведены номера прерываний 00..03 и они вызывают подпрограммы 000..003. Если входные прерывания не используются, подпрограммы 000..003 можно использовать как обычные подпрограммы.

Номер входа	Номер прерывания	Номер подпрограммы
00003	0	000
00004	1	001
00005	2	002
00006	3	003

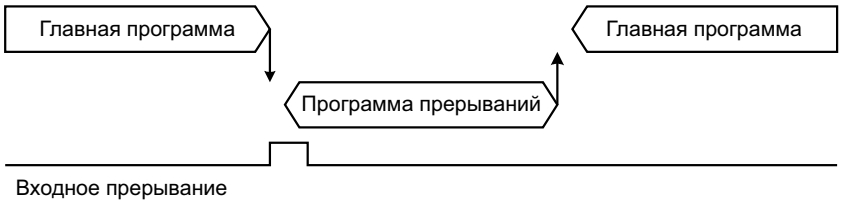
Обновление входов

Если обновление входа не используется, при отработке прерываний состояние входного сигнала будет ненадежным. В зависимости от константы времени входа входные сигналы могут не включиться в 1, даже если использовать обновление входов. Это относится и к состоянию бита входа прерывания, который вызвал прерывание.

Например, IR 00000 не включится в 1 в программе прерываний для входного прерывания 0, если он не обновлен. В таком случае используйте флаг ВСЕГДА 1 (SR 25313) в программе прерываний вместо IR 00000.

Режим прерываний по входу

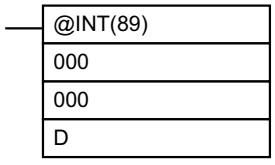
При получении сигнала прерывания главная программа прерывается и сразу же начинается отрабатываться программа прерывания, независимо от момента в цикле, в который принят сигнал прерывания. Сигнал должен быть в состоянии 1 не менее 200 мкс для того, чтобы быть распознанным.



Используйте следующие команды для программирования входных прерываний в режима прерывания по входу.

Маскирование/размаскирование прерываний

Командой INT(89) установите или сбросьте маску прерываний в случае необходимости.



Настройки производите битами 0..3 слова D, которые соответствуют входным прерываниям 0..3
0: маска убрана (входные прерывания разрешены)
1: маска установлена (входные прерывания запрещены)

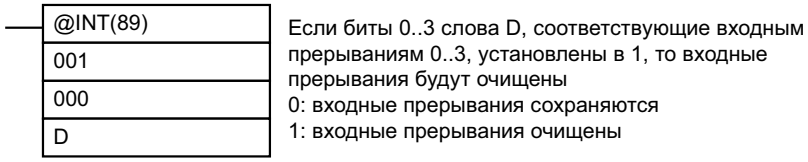
Все входные прерывания замаскированы при начале работы ПК. Если используется режим прерывания по входу, разрешите использование входов командой INT(89), как показано выше.

Стирание замаскированных прерываний

Если бит, соответствующий входному прерыванию, включается в 1 во время замаскированного состояния, это входное прерывание сохранится в памяти и будет выполняться как только будет снята маска. Чтобы это прерывание не выполнялось после очистки маски, прерывание должно быть убрано из памяти.

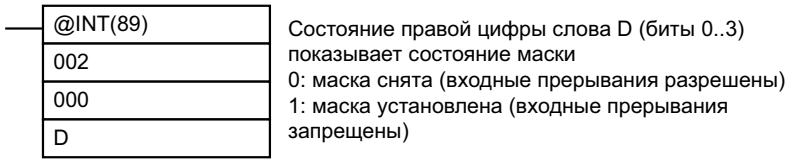
Только один сигнал прерываний будет сохранен в памяти для каждого номера прерывания.

Командой INT(89) сотрите входное прерывание из памяти.



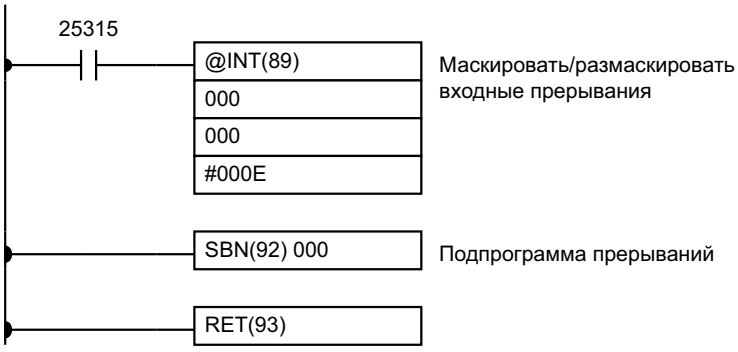
Чтение состояния маски

Командой INT(89) читайте состояние маски входного прерывания.



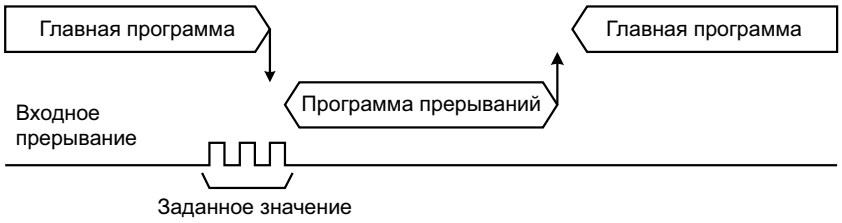
Пример программы

Когда вход 00003 (входное прерывание 0) устанавливается в 1, отработка сразу же переходит к программе прерывания 000. Входы для DM 6628 установлены в 0001.



Режим счета

Входы внешнего сигнала подсчитываются с высокой скоростью и прерывание вызывается, когда значение счетчика достигнет заданного значения. При вызове прерывания главная программа прерывается и начинается отрабатываться программа прерывания. Можно подсчитывать сигналы до 1 кГц.



Проделайте следующие операции для программирования входных прерываний с использованием режима счета.

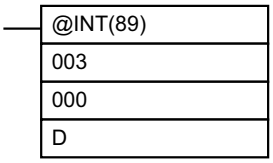
1, 2, 3,... 1. Запишите задания для операции счетчика в слова SR, показанные в следующей таблице. Диапазон заданий - 0000..FFFF (0..65 535). Значение 0000 запретит операцию счета до занесения нового значения и повторения шага 2.

Прерывание	Слово
Входное прерывание 0	SR 240
Входное прерывание 1	SR 241
Входное прерывание 2	SR 242
Входное прерывание 3	SR 243

Слова SR, используемые в режиме счетчика (SR 240..SR 243) содержат 16-ричные, а не двоично-десятичные данные. Если режим счетчика не используется, эти слова можно использовать как рабочие биты.

Замечание Эти слова SR очищаются перед началом работы и должны быть записаны из программы.

2. Командой INT(89) обновите заданное значение режима счетчика и разрешите прерывания.



Если биты 0..3 слова D, соответствующие входным прерываниям 0..3 будут равны 0, то заданное значение будет обновлено и прерывания будут разрешены
0: задание режима счетчика обновлено и маска снята
1: нет обновления

Если входное прерывание не используется, обязательно установите соответствующий бит в 1.

Прерывание, у которого обновлено задание, будет разрешено в режиме счетчика. Когда счетчик отсчитает заданное число, произойдет прерывание, счетчик сбросится и последовательность счет/прерывания продолжится до остановки счетчика.

Замечание 1. Если команда INT(89) используется во время счета, текущее значение станет равным заданию. Таким образом, Вы должны использовать форму команды @ (срабатывающую по франту), или прерывание может не вызваться.

2. Задание будет установлено, когда команда INT(89) выполнена. Если уже идут прерывания, задание не изменится простым изменением SR 240... 243, т.е. если их содержание изменено, задание должно быть обновлено повторным исполнением команды INT(89).

Прерывания можно замаскировать таким же процессом, как и в режиме прерывания по входу, но если замаскированные прерывания очищаются с помощью того же процесса, прерывания будут в режиме прерываний по входу, а не в режиме счетчика.

Сигналы прерывания, принятые для замаскированных прерываний, можно очистить так же, как и в режиме прерываний по входу.

Текущее значение счетчика в режиме счетчика

Когда входные прерывания используются в режиме счетчика, текущее значение счетчика будет загружено в слово SR, соответствующее входным прерываниям 0..3. Диапазон значений 0000..FFFF (0..65 535) и значение равно текущему значению счетчика минус 1.

Прерывание	Слово
Входное прерывание 0	SR 244
Входное прерывание 1	SR 245
Входное прерывание 2	SR 246
Входное прерывание 3	SR 247

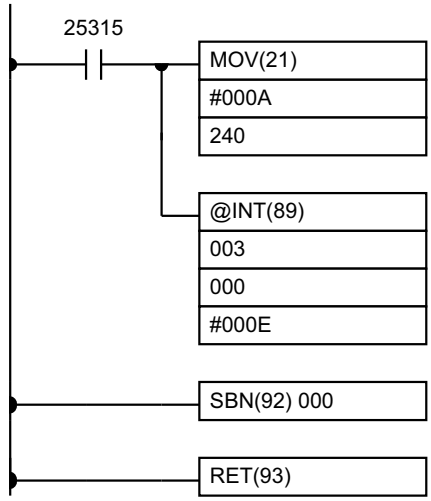
Пример: Текущее значение для прерывания с заданием 000A будет записано как 0009 сразу после выполнения команды INT(89).

Замечание Даже если входные прерывания не используются в режиме счетчика, эти биты SR нельзя использовать как рабочие биты.

Пример программы

Когда вход 00003 (прерывание 0) включится в 1 10 раз, сразу произойдет переход к программе прерываний с номером 000. В следующей таблице показано, где хранятся задания счетчика и текущие значения - 1. Входы для DM 6628 были установлены на 0001.

Прерывание	Слово, содержащее задание	Слово, содержащее текущее значение -1
Вход 00003 (Вх. прерывание 0)	SR 240	SR 244
Вход 00004 (Вх. прерывание 1)	SR 241	SR 245
Вход 00005 (Вх. прерывание 2)	SR 242	SR 246
Вход 00006 (Вх. прерывание 3)	SR 243	SR 247



1.5.3 Маскирование всех прерываний

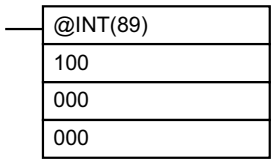
Все прерывания, включая входные прерывания, прерывания интервального таймера и прерывания высокоскоростных счетчиков, можно замаскировать или размаскировать как группу командой INT(89). Эта маска является дополнительной к любым другим маскам индивидуальных типов прерываний. Кроме того, очистка масок всех прерываний не снимает маски индивидуальных типов прерываний, а возвращает их в замаскированное состояние, которое существовало перед выполнением команды INT(89) для маскирования их как группы.

Пользуйтесь командой INT(89) только когда требуется временно замаскировать все прерывания, и всегда используйте команды INT(89) в паре, используя первую команду для маскирования, а вторую для размаскирования прерываний.

INT(89) нельзя использовать для маскирования и размаскирования всех прерываний из программы прерывания.

Маскирование прерывания

Используйте команду INT(89) для запрещения всех прерываний.



Если прерывание вызвано, когда прерывания замаскированы, обработка прерывания не произойдет, однако запрос на прерывание будет запомнен для входных прерываний,

прерываний интервального таймера и прерываний ВСч. Прерывания будут обслужены после снятия маски.

Размаскирование прерываний

Используйте команду INT(89) для размаскирования прерываний:

@INT(89)
200
000
000

1.5.4 Прерывания интервальных таймеров

CPM1/CPM1A оснащен одним интервальным таймером. Когда интервал отсчитан, главная программа прерывается и сразу же выполняется программа прерывания независимо от места в цикле.

Есть 2 режима работы таймерных прерываний, одноразовый режим, в котором по истечении заданного времени прерывание отрабатывается только раз, и режим прерывания по расписанию, когда прерывания повторяются через фиксированные интервалы.

Задания интервальных таймеров можно установить в диапазоне 0.5..319 968 мс, единица времени 0.1 с.

Работа

Используйте следующую команду для вызова и управления интервальным таймером.

Пуск одноразового режима

Используйте команду STIM(69) для пуска интервального таймера в одноразовом режиме.

@STIM(69)
C_1
C_2
C_3

C1: интервальный таймер, одноразовый режим (000)
C2: задание таймера (адрес первого слова)
C_3: номер подпрограммы (4 цифры, BCD) 0000..0049

1, 2, 3,... 1. Когда C2 введено как адрес слова:

C2: Задание декрементирующего счетчика (4 двоично-десятичных цифры): 0000..9999.

C2+1: Интервал времени декрементирования (4 двоично-десятичных цифры; единица времени: 0.1 мс): 0005..0320 (0.5 мс..32 мс).

Каждый раз по истечении заданного интервала C2+1 декрементальный счетчик уменьшит текущее значение на 1. Когда текущее значение достигнет 0, заданная подпрограмма будет вызвана один раз и таймер остановится.

Время от выполнения команды STIM (69) до того, как время истечет, рассчитывается следующим образом:

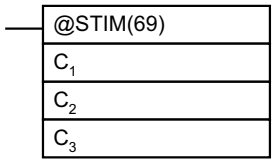
(Содержание C2) x (Содержание C2 + 1) x 0.1 мс = (0.5..319 968) мс.

2. Когда C2 введена как константа:

Заданное значение декрементального счетчика станет равно заданной константе (в мс) и интервал декрементирования будет 10 (1 мс).

Пуск режима прерываний по расписанию

Используйте команду STIM(69) для пуска интервального таймера в режиме прерываний по расписанию.



C1: интервальный таймер, режим прерывания по расписанию (003)
C2: задание таймера (адрес первого слова)
C_3: номер подпрограммы (4 цифры, BCD) 0000..0049

1, 2, 3,... 1. Когда C2 введено, как адрес слова:

C2: заданное значение декрементального счетчика (4 двоично-десятичных цифры): 0000..9999.

C2+1: интервал времени декрементирования (4 двоично-десятичных цифры; единица времени: 0.1 мс): 0005..0320 (0.5 мс..32 мс).

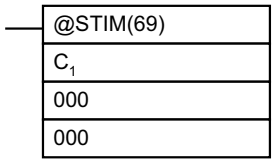
Значения параметров аналогичны значению в одноразовом режиме, но в режиме прерываний по расписанию текущее значение таймера будут сброшено в задание и декрементирование начнется снова после вызова подпрограммы. В режиме прерываний по расписанию прерывания будут повторяться в фиксированными интервалами до остановки работы.

2. Когда C2 введено как константа:

Значения параметров аналогичны значениям в одноразовом режиме, но прерывания будут продолжаться с фиксированными интервалами до остановки работы.

Чтение истекшего времени таймера

Используйте команду STIM(69) для чтения истекшего времени таймера.



C_1: чтение истекшего времени
C_2: первое слово параметра 1
C_3: параметр 2

C2: Количество операций декрементирования счетчика (4 двоично-десятичных цифры).

C2+1: Интервал времени декрементного счетчика (4 двоично-десятичных цифры; единица времени: 0.1 мс).

C3: Истекшее время от предыдущего декрементирования (4 двоично-десятичных цифры; единица времени: 0.1 мс).

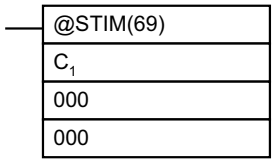
Время от пуска интервального таймера до выполнения данной команды рассчитывается следующим образом:

[(Содержание C2) x (Содержание C2 + 1) + (Содержание C3)] x 0.1 мс.

Если заданный интервальный таймер остановлен, будет сохранено значение “0000”.

Остановка таймера

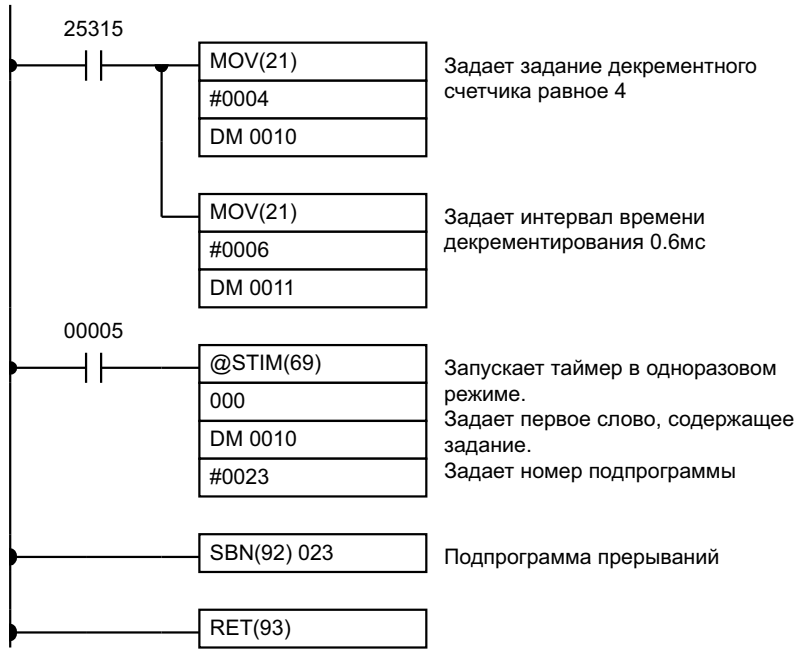
Используйте команду STIM(69) для остановки интервального таймера. Интервальный таймер будет остановлен.



C1: останов интервального таймера

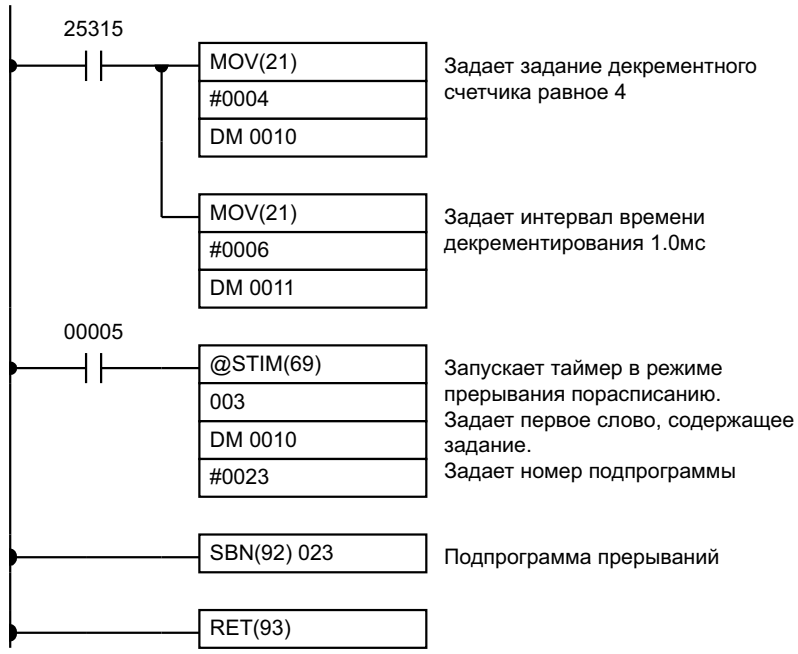
Пример применения (Одноразовый режим)

В данном примере прерывание вызывается через 2.4 мс (0.6 мс x 4) после включения входа 00005 в 1. Прерывание исполняет подпрограмму прерывания номер 23.



Пример применения (Прерывания по расписанию)

В данном примере прерывание вызывается каждые 4 мс (1.0 мс x 4) после включения входа 00005 в 1. Прерывание вызывает подпрограмму 23.

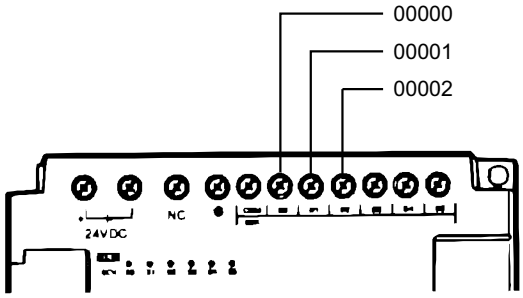


1.5.5 Прерывания высокоскоростного счетчика

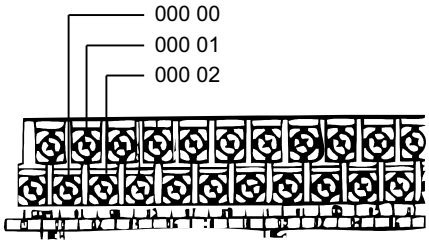
В CPM1/CPM1A есть функция ВСч, которую можно использовать в инкрементальном режиме либо в режиме плюс/минус. ВСч можно комбинировать с входными прерываниями для управления по конечному значению или управления по сравнению с зоной, на которые не оказывает влияния время цикла.

Сигналы ВСч могут поступать на входные биты ЦУ 00000..00002.

CPM1



CPM1A



Режим	Функции входа	Метод входа	Частота счета	Диапазон счета	Метод управления
Плюс/минус	00000: Вход фазы A 00001: Вход фазы B 00002: Вход фазы Z	Разность фаз, 4x входа	2.5 kHz макс.	-32767.. 32767	Управление конечным значением: можно записать до 16 конечных значений и номеров подпрограмм прерываний. Управление сравнением с зоной: можно записать до 8 зон и номеров подпрограмм прерываний.
Инкремен-тальный	00000: Счетный вход 00001: См. Зам. 00002: Вход сброса	Индивидуаль-ные входы	5 кГц макс.	0.. 65535	

Замечание В инкрементальном режиме вход 00001 можно использовать как обычный вход.

Установочные параметры ВСч

При использовании функции ВСч CPM1/CPM1A нужно сделать следующие настройки в DM 6642.

Биты DM 6642	Функция	Значения		
		Инкремент	Плюс/минус	Не используются
00 - 03	Задает режим счетчика 0: Плюс/минус 4: Инкрементальный	4	0	0 или 4
04 - 07	Задает Метод сброса 0: Сброс фазой Z + системной программой 1: Сброс системной программой	0 или 1	0 или 1	0 или 1
08 - 15	Задает счетчик: 00: Счетчик не используется 01: Счетчик используется	01	01	00

Диапазон счета

ВСч CPM1/CPM1A использует операцию линейного счета и значение хранится в SR 248 и SR 249 (старшие 4 цифры хранятся в SR 248 и младшие 4 цифры в SR 249).

Режим	Диапазон счета
Плюс/минус	F003 2767..0003 2767 (-32 767..32 767) Самая левая цифра SR 248 указывает на знак. F - отрицательный, 0 - положительный.
Инкрементальный	00000..0006 5535 (0..65 535)

Если текущее значение ВСч превышает верхнюю границу диапазона счета, происходит переполнение +; если текущее значение ВСч становится меньше нижней границы, происходит переполнение -.

Ошибка	Инкрементирование	Плюс/минус	Текущее значение
Переполнение +	Происходит при инкрементировании от 65 535.	Происходит при инкрементировании от 32 767.	0FFF FFFF
Переполнение -	-	Происходит при декрементировании от -32 767	FFFF FFFF

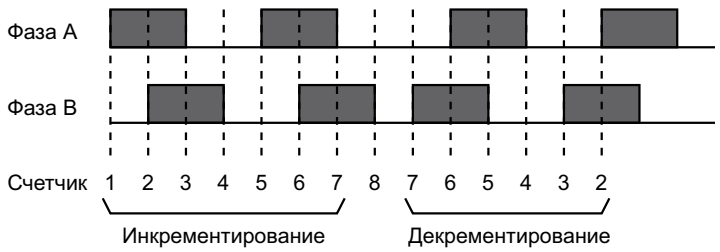
Отработка

Два типа сигналов могут поступать с импульсного датчика положения. Режим счета, используемый для ВСч, будет зависеть от типа сигналов. Режим счета и режим сброса задаются в DM 6642; Эти настройки становятся эффективными при включении ПК или пуске программы.

Режим плюс/минус

Для входов используются двухфазный сигнал с разностью фаз 4X (фаза A и фаза B) и сигнал фазы Z . Значение счетчика инкрементируется или декрементируется в зависимости от разности двухфазных сигналов.

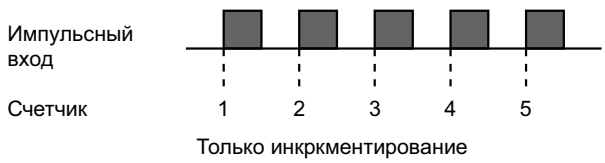
Режим плюс/минус



Режим инкрементальный

Для входов используются однофазный импульсный сигнал и сигнал сброса счета. Значение только инкрементируется в соответствии с однофазным сигналом.

Режим инкрементальный



Только инкркментирование

Замечание При повторном запуске счетчика всегда нужно применять один из режимы сброса, описанных ниже. Счетчик будет автоматически сброшен при пуске или останове программы.

Следующая последовательность сигналов трактуется как увеличивающие (инкрементальные) импульсы: Передний фронт фазы A - Передний фронт фазы B - задний фронт фазы A - задний фронт фазы B.

Следующая последовательность сигналов трактуется как уменьшающие (декрементальные) импульсы: Передний фронт фазы B - Передний фронт фазы B - задний фронт фазы B - задний фронт фазы A.

Режим плюс/минус всегда использует вход с разностью фаз 4X. Число единиц счетна для каждого оборота датчика = 4 x разрешающую способность датчика. Выбирайте датчик, основываясь на диапазонах счета.

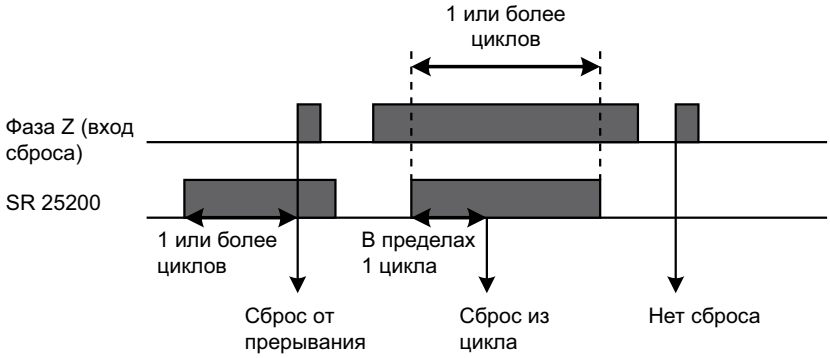
Методы сброса

Для сброса (т. е. установки текущего значения счетчика в 0) можно выбрать любой из 2 методов:

Сброс сигналом фазы Z + системной программой:

Текущее значение сбрасывается, когда фазы Z (вход сброса) включается в 1, в то время как бит сброса высокоскоростного счетчика (SR 25200) = 1.

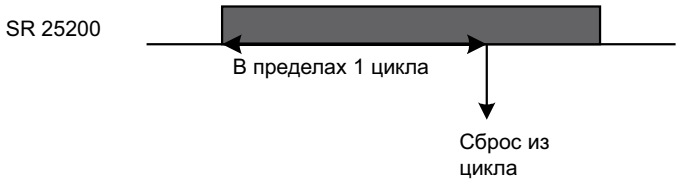
Сброс фазой Z + системной программой



Сброс системной программой:

Текущее значение сбрасывается, когда бит сброса высокоскоростного счетчика (SR 25200) = 1.

Сброс системной программой



Замечание Бит сброса высокоскоростного счетчика (SR 25200) обновляется один раз каждый цикл, так что для того, чтобы надежно его прочитать, он должен быть в состоянии 1 как минимум 1 цикл.

Z в обозначении “фаза Z” - это сокращение от Zero (нуль). Этот сигнал показывает, что датчик сделал 1 оборот.

Счет прерываний высокоскоростного счетчика

Для прерываний высокоскоростного счетчика 0 вместо “инкрементального счета” применяется таблица сравнений. Проверка счета может проводится любым из двух далее описанных метода. В таблице сравнений сохраняются условия сравнения (для сравнения с текущим значением) и комбинации программ прерываний.

Сравнение по конечному значению:

В таблице сравнений хранится максимум 16 условий сравнений (конечное значение + направление счета) и комбинаций программ прерываний. Когда текущее значение счетчика и направление совпадают с табличными, выполняется заданная программа прерываний.

Сравнение с зоной:

В таблице сравнений хранится 8 условий сравнений (верхняя и нижняя граница) и комбинаций программ прерываний. Когда текущее значение станет больше либо равным верхней границе или меньше либо равным нижней границе, выполняется заданная программа прерываний.

Программирование:

Для программирования высокоскоростного счетчика проделайте следующие операции.

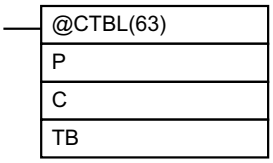
Высокоскоростной счетчик начнет операцию счета при правильной настройке установочных параметров, но сравнения с таблицей сравнений и вызов прерываний начнется только после выполнения команды CTBL(63).

Высокоскоростной счетчик сброшен в 0 при включении питания и при пуске программы.

Текущее значение высокоскоростного счетчика сохраняются в SR 248 и SR 249.

Управление прерываниями высокоскоростного счетчика

1, 2, 3,... 1. Используйте команду CTBL(63) для сохранения таблицы сравнения в CPM1/CPM1A и начала сравнения.



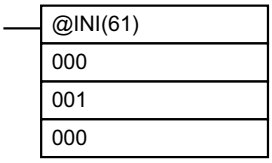
C: три цифры BCD
000: задать таблицу конечных значений и начать сравнение
001: задать таблицу зон и начать сравнение
002: задать только таблицу конечных значений
003: задать только таблицу зон
TB: начальное слово таблицы сравнений

Если задано C = 0000, сравнение будет производится методом конечных значений, если 0001 - методом сравнения с зоной. Таблица сравнения будет сохранена и сравнение начнется после завершения операции сохранения. Во время выполнения сравнения высокочастотные прерывания будут выполняться согласно таблицы сравнений. Подробности о содержании таблиц сравнений см. объяснения к команде CTBL(63) в Главе 5 Набор команд.

Замечание При сравнении с зоной результаты сравнения хранятся в AR 1100..AR 1107.

При задании C = 002 сравнение будет производиться методом конечных значений, если C = 003 - методом сравнения с зоной. Для любого из заданий таблица сравнений будет сохранена, но для начала сравнения нужно использовать после команду INI(61).

2. Для остановки сравнения выполните команду INI (61), как показано далее.



Для возобновления сравнения задайте второй операнд = 000 (выполнить сравнение) и выполните команду INI (61).

Когда таблица сохранена, она будет храниться в CQM1 во время работы (т. е. исполнении программы) до сохранения другой таблицы.

Чтение текущего значения.

Есть 2 способа читать текущее значение.

- читать из SR 248 и SR 249.
- командой PRV(62).

Чтение из SR 248..SR 249

Текущее значение высокоскоростного счетчика хранится в SR 248 и SR 249 как показано ниже. Самая левая цифра будет равна F для отрицательных значений.

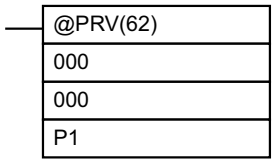
4 левых разряда	4 правых разряда	Режим плюс/минус	Режим инкремент
SR 248	SR 249	F003 2767 (-32767) до 0003 2767	0000 до 0006 5535

Замечание 1. Данные слова обновляются только один раз каждый цикл, так что может быть отличие от фактических текущих значений.

2. Когда высокоскоростной счетчик не используется, биты данных слов можно использовать как рабочие.

Применение команды PRV(62)

Читайте текущее значение высокоскоростного счетчика 0 при помощи команды PRV(67)



P1: начальное слово текущего значения

4 левых разряда	4 правых разряда	Режим плюс/минус	Режим инкремент
P1+1	P1	F003 2767 (-32767) до 0003 2767	0000 до 0006 5535

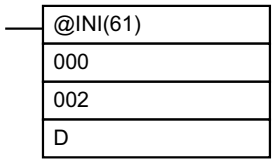
Текущее значение читается, когда команда PRV(62) уже выполнена.

Изменение текущего значения

Есть 2 метода изменения текущего значения высокоскоростного счетчика:

- Сброс методами сброса (В данном случае текущее значение сбрасывается в 0).
- Использование команды INI(61).

Здесь объясняется метод с использованием команды INI(61). Для объяснения метода сброса смотрите начало этого описания высокоскоростных счетчиков.



D: ведущее слово для сохранения нового текущего значения

4 левых разряда	4 правых разряда	Режим плюс/минус	Режим инкремент
D+1	D	F003 2767 (-32767) до 0003 2767	0000 до 0006 5535

Для задания отрицательного числа в режиме плюс/минус задавайте F в самой старшей цифре.

Пример применения

(Режим инкрементирования)

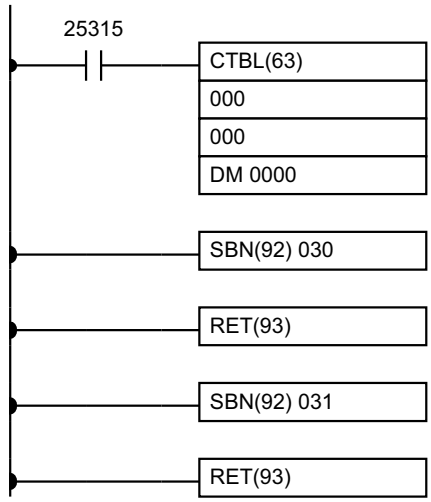
В данном примере показана программа, которая использует высокоскоростной счетчика с входом одной фазы в инкрементальном режиме, сравнение методом совпадения конечных значений.

Условия сравнения (конечные значения и направление счета) хранятся в таблице сравнений с номерами подпрограмм. Можно сохранить до 16 конечных значений. Соответствующие подпрограммы выполняются при совпадении текущих значений счетчика и конечных значений.

В таблице сравнения хранятся следующие данные:

DM 0000	0002	Число условий сравнения:- 2
DM 0001	1000	Конечное значение 1: 1000
DM 0002	0000	
DM 0003	0030	Номер подпрограммы обработки прерывания сравнения 1: 30
DM 0004	2000	Конечное значение 2: 2000
DM 0005	0000	
DM 0006	0031	Номер подпрограммы обработки прерывания сравнения 2: 31

На диаграмме показан пример релейно-контактной схемы. DM 6642 должно быть установлено в 014, где обозначает метод сброса, который может быть 0 или 1.



Пример применения (Режим плюс/минус)

В данном примере приведена программа, использующая высокоскоростной счетчик с входами разности фаз в режиме плюс/минус, сравнение методом зон.

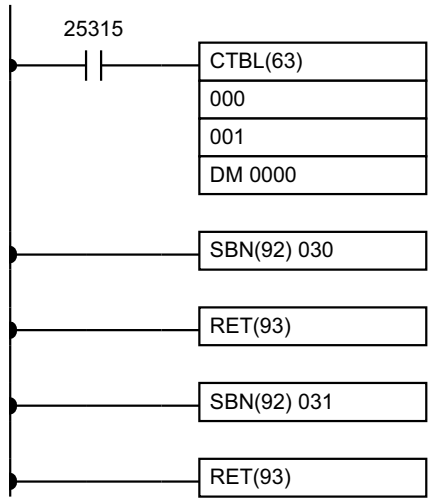
Условия сравнения (верхняя и нижняя граница зон) загружены в таблице сравнений с номерами подпрограмм. Можно задать до 8 зон. Соответствующая подпрограмма выполняется при выходе текущих значений счетчика за границы зоны.

Замечание Всегда задавайте 8 зон. Если требуется менее 8 зон, задайте номера подпрограмм FFFF. Значение FFFF показывает, что подпрограммы не должны выполняться.

В таблице сравнения хранятся следующие данные:

DM 0000	1500	
DM 0001	0000	Нижняя граница 1: 1 500
DM 0002	3000	
DM 0003	0000	Верхняя граница 1: 3 000
DM 0004	0040	Номер подпрограммы прерывания зоны 1: 40
DM 0005	7500	
DM 0006	0000	Нижняя граница 2: 7 500
DM 0007	0000	
DM 0008	0001	Верхняя граница 2: 10 000
DM 0009	0041	Номер подпрограммы прерывания зоны 2: 41
DM 0010	0000	
DM 0011	0000	
DM 0012	0000	
DM 0013	0000	
DM 0014	FFFF	Подпрограмма прерывания зоны 3 не исполняется
...		
DM 0035	0000	
DM 0036	0000	
DM 0037	0000	
DM 0038	0000	
DM 0039	FFFF	Подпрограмма прерывания зоны 8 не исполняется

На следующем рисунке показан пример релейно-контактной схемы. DM 6642 должен быть установлен в 010, где обозначает метод сброса, который может быть 0 или 1.


25315
CTBL(63)
000
001
DM 0000
SBN(92) 030
RET(93)
SBN(92) 031
RET(93)

1.6 Функции прерываний SRM1

В данном разделе описаны установочные параметры и методы применения функций прерываний SRM1.

1.6.1 Типы прерываний

В SRM1 имеется только 1 тип обработки прерываний, описанный далее.

Прерывания по интервальному таймеру:

Обработка прерываний выполняется по интервальному таймеру с точностью 0.1 мс.

1.6.2 Прерывания по интервальному таймеру

В SRM1 есть 1 интервальный таймер. Когда время интервального таймер истекает, главная программа прерывается и сразу же выполняется программа прерывания, независимо от места в цикле.

Есть два режима работы интервального таймера: Одноразовый режим, в котором только одно прерывание будет исполняться по истечении заданного времени, и Режим прерывания по расписанию, в котором прерывания повторяются через фиксированные интервалы.

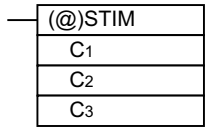
Задание интервального таймера можно давать любое в диапазоне 0.5..319968, единица задания - 0.1 мс.

Отработка программы

Используйте следующую команду для запуска и управления интервальным таймером.

Пуск одноразового режима

Используйте команду STIM(69) для пуска интервального таймера в одноразовом режиме.



C_1: интервальный таймер, одноразовый режим (000)
C_2: заданное значение таймера (адрес первого слова)
C_3: номер подпрограммы (BCD), 0000..0049

1, 2, 3,... 1. Когда C2 введено, как адрес слова: 1

C2: Задание декрементирующего счетчика (4 двоично-десятичных цифры): 0000..9999.

C2+1: Интервал времени декрементирования (4 двоично-десятичных цифры; единица времени: 0.1 мс): 0005..0320 (0.5 мс..32 мс).

Каждый раз по истечении заданного интервала в слове C2+1 декрементирующий счетчик уменьшает текущее значение на 1. Когда оно станет равным 0, будет вызвана заданная подпрограмма только один раз и таймер остановится.

Время от выполнения команды STIM (69) до истечения заданного времени рассчитывается следующим образом:

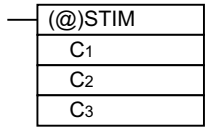
(Содержание C2) x (Содержание C2 + 1) x 0.1 мс =(0.5..319 968 мс). 2

2. Если C2 введено, как константа:

Заданное значение декрементирующего счетчика станет равно заданной константе (в мс) и интервал декрементирования будет = 10 (1 мс).

Пуск в режиме прерываний по расписанию

Используйте команду STIM(69) для пуска интервального таймера в режиме прерываний по расписанию.



C_1: интервальный таймер, режим прерываний по расписанию(000)
C_2: заданное значение таймера (адрес первого слова)
C_3: номер подпрограммы (BCD), 0000..0049

1, 2, 3,... 1. Когда C2 введено, как адрес слова:

C2: Задание декрементирующего счетчика (4 двоично-десятичных цифры): 0000..9999.

C2+1: Интервал времени декрементирования (4 двоично-десятичных цифры; единица времени: 0.1 мс): 0005..0320 (0.5 мс..32 мс).

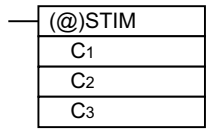
Значения параметров аналогичны параметрам для одноразового режима, но в режиме прерываний по расписанию текущее значение таймера будет установлено в значение задания и декрементирование начнется снова после вызова программы прерывания. В режиме прерываний по расписанию прерывания будут повторяться с фиксированными интервалами до тех пор, пока работа цикла не будет остановлена.

2. Если C2 введено, как константа:

Значения параметров аналогичны параметрам для одноразового режима, но прерывания будут повторяться с фиксированными интервалами до тех пор, пока работа цикла не будет остановлена.

Чтение истекшего времени таймера

Используйте команду STIM(69) для чтения истекшего времени таймера.



C_1: чтение истекшего времени (006)
C_2: первое слово параметра 1
C_3: параметр 2

C2: Число, показывающее сколько раз был декрементирован счетчик (4 двоично-десятичных цифры).

C2+1: интервал времени декрементирующего счетчика (4 двоично-десятичных цифры; единица времени: 0.1 мс).

C3: истекшее время от предыдущего декремента. (4 двоично-десятичных цифры; единица времени: 0.1 мс).

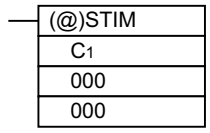
Время от начала отсчета интервала до выполнения данной команды рассчитывается следующим образом:

[(Содержание C2) x (Содержание C2 + 1) + (Содержание C3)] x 0.1 мс.

Если выбранный интервальный таймер остановлен, будет сохранено значение 0000.

Остановка таймера

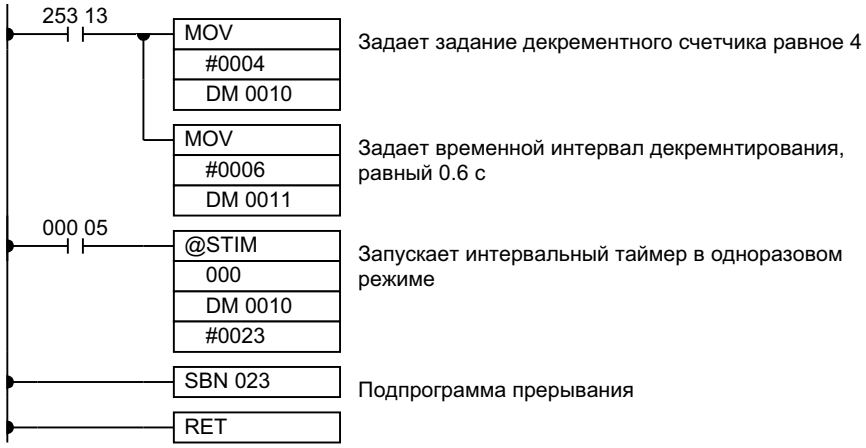
Используйте команду STIM(69) для остановки интервального таймера.



C_1: останов интервального таймера

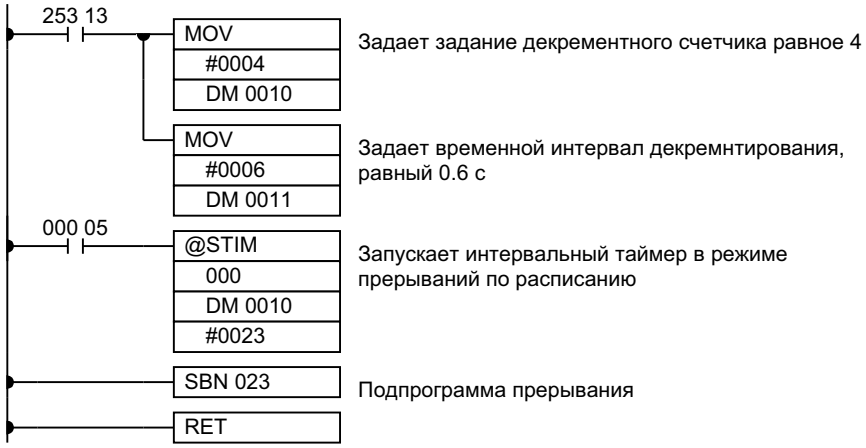
Пример применения (одноразовый режим)

В данном примере прерывание генерируется каждые 2.4 мс (0.6 мс x 4) после установки в 1 входа 0005; при прерывании отрабатывается программа прерываний 23. 25315 Флаг первого цикла



Пример применения (режим прерываний по расписанию)

В данном примере прерывание генерируется каждые 4.0 мс (0.1 мс x 4) после установки в 1 входа 0005; при прерываниях отрабатывается программа прерываний 23. 25315 Флаг первого цикла



1.7 Функции распределенных входов/выходов CompoBus/S (Только SRM1)

Число подключенных узлов Можно подключить максимум 16 или 32 узла CompoBus/S

Число узлов	**Время реакции связи**
32	0.8 мс
16	0.5 мс

Максимально число узлов можно задать с программирующего устройства путем следующих установок параметра DM 6603.

Слово (а)	**Бит (ы)**	**Функция**	**Задание**
DM 6603	00..03	Максимальное число узлов CompoBus/S 16 или 32 00: 32 узла 01: 16 узлов	00 или 01
	04..15	Резерв	00

Замечание После изменения данного параметра обязательно выключите и включите питание, чтобы сделать эти значения активными.

Прерывания терминалов входов/выходов CompoBus/S

Входные биты в IR 000..007 и выходные биты в IR 010..017 служат в качестве битов прерывания для терминалов входов/выходов CompoBus/S. Прерывания терминалов входов/выходов CompoBus/S (IN0..IN15 и OUT0..OUT 15) связаны в соответствии со следующей таблицей.

IN0..IN15 являются адресами узлов для терминалов входов, а OUT0..OUT 15 являются адресами узлов для терминалов выходов.

Слово		Бит															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Вход	IR 000	IN1								IN0							
	IR 001	IN3								IN2							
	IR 002	IN5								IN4							
	IR 003	IN7								IN6							
	IR 004	IN9								IN8							
	IR 005	IN11								IN10							
	IR 006	IN13								IN12							
	IR 007	IN15								IN14							
Выход	IR 010	OUT1								OUT0							
	IR 011	OUT3								OUT2							
	IR 012	OUT5								OUT4							
	IR 013	OUT7								OUT6							
	IR 014	OUT9								OUT8							
	IR 015	OUT11								OUT10							
	IR 016	OUT13								OUT12							
	IR 017	OUT15								OUT14							

Замечание 1. Когда макс. число узлов CompoBus/S задано 16, входы IN8..IN15 можно использовать в качестве рабочих битов.

2. Терминалам CompoBus/S с числом точек менее 8 выделены адреса битов начиная либо с 0, либо с 8.

3. Терминалы CompoBus/S с числом точек больше 16 можно назначать только четные адреса.

Флаги состояния Входные биты в IR 000..007 и выходные биты в IR 010..017 служат в качестве битов прерывания для терминалов входов/выходов CompoBus/S.

Слово	Старшие биты: Флаги ошибки связи Ведомых								Младшие биты: Флаги сложения Ведомых							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR04	OUT 7	OUT 6	OUT 5	OUT 4	OUT 3	OUT 2	OUT 1	OUT 0	OUT 7	OUT 6	OUT 5	OUT 4	OUT 3	OUT 2	OUT 1	OUT 0
AR05	IN 7	IN 6	IN 5	IN 4	IN 3	IN 2	IN 1	IN 0	IN 7	IN 6	IN 5	IN 4	IN 3	IN 2	IN 1	IN 0
AR06	OUT 15	OUT 14	OUT 13	OUT 12	OUT 11	OUT 10	OUT 9	OUT 8	OUT 15	OUT 14	OUT 13	OUT 12	OUT 11	OUT 10	OUT 9	OUT 8
AR07	IN 15	IN 14	IN 13	IN 12	IN 11	IN 10	IN 9	IN 8	IN 15	IN 14	IN 13	IN 12	IN 11	IN 10	IN 9	IN 8

Замечание IN0..IN15 - терминалы входов, OUT0..OUT15 - терминалы выходов.

Когда макс. число узлов CompoBus/S задано 16, входы IN8..IN15 и OUT8..OUT15 нельзя использовать.

Флаги сложения Ведомого включены в 1, когда Ведомый подключен к сети. Когда питание Модуля ЦПУ выключить и снова включить, все биты будут выключены в 0.

Флаги ошибки связи Ведомых включены в 1, когда Ведомый - участник сети - отключен от сети. Бит будет выключен в 0, когда Ведомый снова возобновляет работу в сети. Прим. Если использованы некорректные параметры, возникает нефатальная ошибка, AR 1302 включится в 1 и будет использоваться значение по умолчанию (0 или 00).

Информацию о параметрах сети NT link для других изделий OMRON смотрите Инструкции по работе с данными ПК.

1.8 Функции связи

Связь в CQM1

Через порты CQM1 можно осуществить следующие типы связи.

- Связь Host link с управляющим компьютером
- Связь RS-232C с компьютером или другим устройством
- Связь 1:1 с другим контроллером
- Связь NT Link

Замечание Данных режимов связи нет у CQM1-CPU11-E, у которого есть только периферийный порт.

В данной главе описаны начальная настройка ПК и методы применения этих типов связи.

Связь в CPM1/CPM1A

CPM1/CPM1A может осуществлять различные виды связи по периферийному порту через адаптер RS-232C или адаптер RS-422.

Связь с верхним уровнем HOST LINK

ПК CPM1/CPM1A совместимы с системой Host Link System, которая позволяет управлять до 32 ПК с одного управляющего компьютера. Адаптер RS-232C служит для связи 1:1, а адаптер RS-422 и адаптер B500-AL004 Link Adapter служат для связи 1:n.

CPM1/CPM1A, оборудованный адаптером RS-232C, может также сообщаться с программатором путем использования команд Host link.

Подробности см. 1.6.4 данной инструкции и 1.2.2 Инструкции по работе.

Связь 1 :1

Можно создать линию обмена данных с областью данных в другом ПК. Для создания связи 1 :1 применяется адаптер RS-232C.

Подробности см. 1-6-6 данной инструкции и 1-2-3 Инструкции по работе.

NT LINK

Используя NT LINK, CPM1 можно подключить к программируемому терминалу (интерфейс NT LINK) через адаптер RS-232C.

Подробности см. 1-6-8 данной инструкции и 1-2-4 Инструкции по работе.

Связь в SRM1

Через порты CQM1 можно осуществить следующие типы связи.

- Связь Host link с управляющим компьютером
- Связь RS-232C с компьютером или другим устройством
- Связь 1:1 с другим контроллером
- Связь NT Link

Замечание Связь NT Link невозможна при использовании SRM1-C01, который имеет только периферийный порт. SRM1-C01 может быть присоединен к NT терминалу через адаптер RS232C в режиме Host Link.

1.8.1 Установочные параметры CQM1

Для настройки параметров портов связи служат параметры DM 6645..DM 6654. Параметры для порта RS-232C в DM 6645..DM 6649 можно задать командами меню, используя SSS.

Если секция 5 переключателя DIP CQM1 установлена в ON, параметры связи ПК будут игнорироваться, а использоваться будут следующие параметры:

Режим:	Host Link
Номер узла:	00
Стартовые биты:	1 бит
Длина данных:	7 бит
Стоповые биты:	2 бита

Контроль на четность: четн.
Скорость передачи: 9 600 бод
Задержка передачи: нет

Замечание Приведенные параметры действительны для ЦУ, изготовленных после июля 1995 (партия **75 июля 1995). Для ЦУ, изготовленных перед июлем 1995 (партия **65 июня 1995), будет установлен только 1 стоповый бит и скорость передачи данных 2400 бод.

Значения в DM 6645..DM 6650 определяют основные параметры связи, как показано на следующей диаграмме.

По умолчанию: Host Link со стандартными параметрами

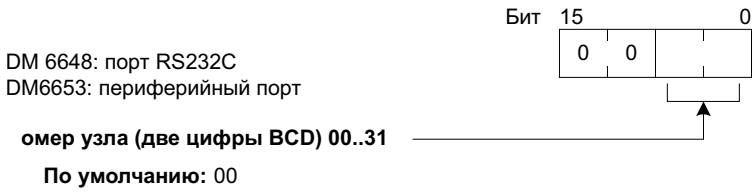
Связь 1:1

Для использования связи 1:1 нужно произвести только настройку режимов связи и слов связи. Задайте режим связи для одного ПК: ведущего 1:1 и для другого ПК: ведомого 1:1, затем задайте слова связи в ПК, заданном как ведущий. Биты 08..11 действуют только для ведущих при связи 1:1.

Замечание Связь 1:1 возможна только для порта RS-232C. Эта настройка невозможна для периферийного порта.

Номер узла Host Link

Для Host Link должен быть задан номер узла для различия между узлами, когда в связи участвуют несколько узлов. Данная настройка требуется только для Host Link. Для использования Host Link в качестве режима связи должен быть задан Host Link и должны быть заданы параметры связи (см. Далее).



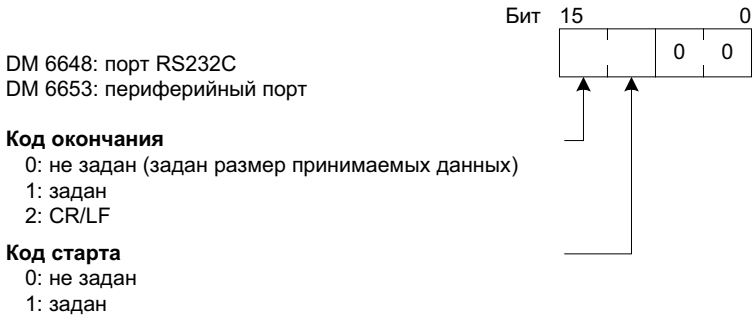
По умолчанию: 00

Задавайте номер узла 00, если только в сеть не объединено несколько узлов.

Коды пуска и окончания RS-232C и принимаемые данные

Коды пуска и окончания или число подлежащих приему данных можно задать согласно следующей диаграмме, если требуется связь RS-232C. Данная настройка требуется только при связи RS-232C. Для использования RS-232C требуется задать режим RS-232C как режим связи и должны быть заданы параметры связи (см. далее).

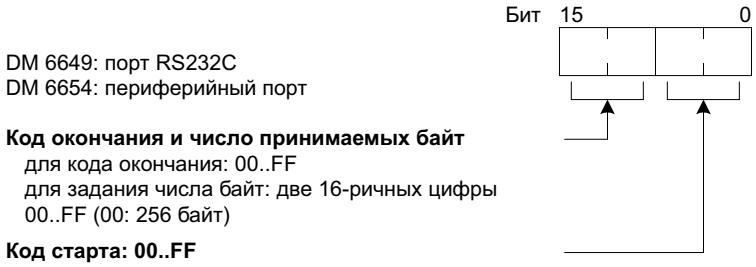
Разрешение кодов старта и окончания



По умолчанию: нет кода старта, прием данных оканчивается посла приема 256 байт

Укажите, нужно ли в начале блока данных задавать стартовый код и код окончания в конце. Вместо задания кода окончания можно задать число байт, которые должны быть приняты до завершения операции. И коды, и число байт принимаемых данных задаются в DM 6649 и DM 6654.

Задание кода старта, кода окончания и размера принимаемых данных



По умолчанию: нет кода старта, прием данных оканчивается после приема 256 байт

Параметры связи Host Link и RS-232C

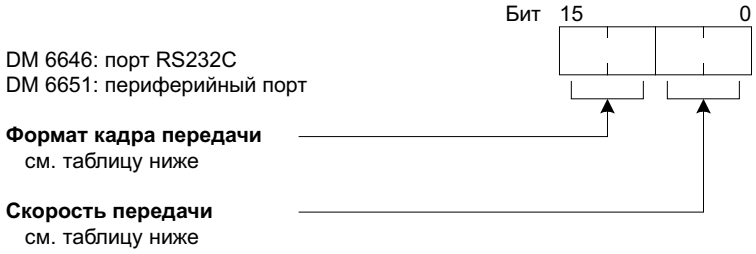
Выберите либо Host Link, либо RS-232C и затем задайте параметры связи, как будет показано далее. Согласовывайте условия связи с установочными параметрами и устройством, с которым осуществляется связь.

Стандартная связь

Если следующие условия достаточны для условий связи, для DM 6645 и DM 6650 установите две правых цифры в 00. Настройки DM 6646 и DM 6651 при таком задании будут игнорироваться.

Стартовые биты:	1 бит
Длина данных:	7 бит
Стоповые биты:	2 бита
Контроль на четность:	четн.
Скорость передачи:	9 600 бод

Задание условий связи



По умолчанию: стандартные условия связи

Формат кадра передачи

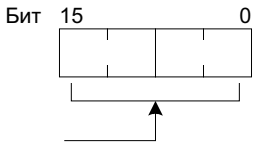
Задание	Стартовые биты	Длина данных	Стоповые биты	Контроль четности
00	1	7	1	четн.
01	1	7	1	нечетн.
02	1	7	1	нет
03	1	7	2	четн.
04	1	7	2	нечетн.
05	1	7	2	нет
06	1	8	1	четн.
07	1	8	1	нечетн.
08	1	8	1	нет
09	1	8	2	четн.
10	1	8	2	нечетн.
11	1	8	2	нет

Скорость передачи

Задание	Скорость передачи
00	1 200 бод
01	2 400 бод
02	4 800 бод
03	9 600 бод
04	19 200 бод

Время задержки передачи

В зависимости от типа устройств, подключенных к порту RS-232C, может возникнуть необходимость разрешить задержку передачи. В таком случае задайте задержку передачи для регулирования интервала разрешенного времени.



DM 6647: порт RS232C
DM 6652: периферийный порт

Задержка передачи (4 цифры BCD, единица времени 10мс)

По умолчанию: нет задержки

1.8.2 Подключение портов

Информацию о подключении портов см. Инструкцию по работе CQM1, Инструкцию по работе CPM1, Инструкцию по работе CPM1A, Инструкцию по работе SRM1.

1.6.3 Связь Host Link CQM1

Связь Host Link была разработана на фирме OMRON для соединения ПК с одним или несколькими управляющими компьютерами по кабелю RS-232C и управлению связью ПК с управляющего компьютера. Обычно Управляющий компьютер выдает команду на ПК, ПК автоматически посылает ответ. Таким образом, связь идет без активной роли ПК. У ПК есть также возможность вызвать передачу данных в случае необходимости непосредственного участия в связи.

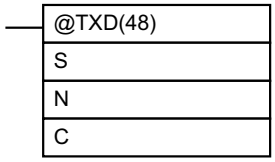
Вообще, имеется 2 способа осуществления связи Host Link. Один основан на командах C-режима, другой на командах FINS (режим CV). CQM1 поддерживает только команды режима C. Подробности о связи Host Link см. Главу 6 Команды Host Link.

Процедура связи

В данной пункте описано использование Host Link для осуществления передачи данных из CQM1. Применение данного метода разрешает автоматическую передачу данных из CQM1 после изменения данных и таким образом упрощает процесс связи, устраняя необходимость постоянного контроля со стороны компьютера.

1, 2, 3,... 1. Проверьте, чтобы AR 0805 (флаг готовности к передаче порта RS-232C) был 1.

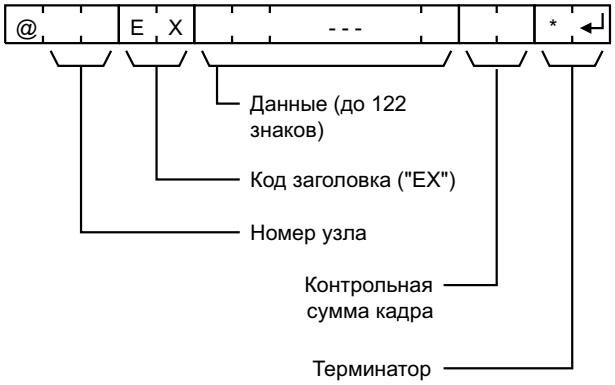
2. Используйте команду TDX(48) для передачи данных.



S: адрес первого слова передаваемых данных
C: параметры управления
биты 00..03
0: левые байты первые
1: правые байты первые
биты 12..15
0: порт RS232C
1: периферийный порт
N: число байт, подлежащих передаче (4 цифры BCD)
0000..0256

От времени исполнения этой команды до завершения передачи данных флаг AR 0805 (или AR 0813 для периферийного порта) останется = 0. Он снова включится в положение 1 после завершения передачи данных. Команда TDX(48) не обеспечивает сигнал ответа, так что для получения подтверждения того, что данные приняты компьютером программа в компьютере должна быть написана таким образом, чтобы извещала ПК после записи данных из CQM1.

Кадр передаваемых данных, передаваемых в режиме Host Link командой TDX(48), имеет следующий вид:

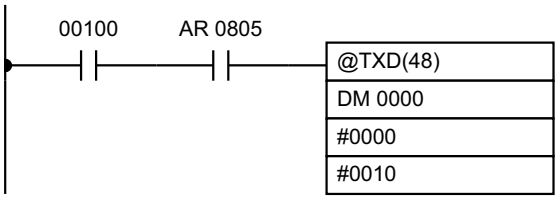


Для сброса порта RS-232C (т.е. для восстановления исходного состояния) включите в 1 SR 25209. Для сброса периферийного порта включите в 1 SR 25208. Эти биты автоматически установятся в 0 после сброса.

Если команда TDX(48) выполняется, когда CQM1 находится в процессе ответа на команду от компьютера, сначала будет завершится передача ответа перед выполнением передачи, вызванной командой TDX(48). Во всех других случаях передаче данных по команде TDX(48) буден отдан высший приоритет.

Пример применения

В данном примере приведена программа использования порта RS-232C в режиме Host Link для передачи 10 байтов данных (DM 0000..DM 0004) в компьютер. Все установочные параметры ПК берутся по умолчанию (т.е. порт RS-232C используется в режиме Host Link, номер узла 00, используются стандартные условия связи). В каждом слове DM 0000..DM 0004 загружено “1234”. С компьютера выполните программу приема данных CQM1 со стандартными условиями связи.



Если AR 0805 (флаг готовности передачи) = 1 при IR 00100 = 1, то будут переданы десять байт данных (DM 0000..DM 0004)

В управляющем компьютере должна быть готова следующая программа для приема данных. Данная программа позволяет компьютеру читать и индикатировать данные, принятые из ПК, при выполнении команды чтения данных из ПК (в режиме HOST LINK).

```
10 ‘CQM1 SAMPLE PROGRAM FOR EXCEPTION
20 CLOSE 1
30 CLS
40 OPEN ‘’COM: E73” AS #1
50 *KEYIN
60 INPUT “DATA ————” , S $
70 IF S$=“ “ THEN GOTO 190
80 PRINT “SEND DATA = “ ; S $
90 ST$ = S $
100 INPUT “SEND OK ? Y OR N ?=“, B $
110 IF B$=“Y” THEN GOTO 130 ELSE GOTO *KEYIN
120 S$=ST$
130 PRINT #1,S$ ‘Посылает команду в ПК
140 INPUT #1,R$ ‘Получает ответ из ПК
150 PRINT “RECV DATA = “;R$
160 IF MID$ (R$,4,2) =“EX’’ THEN GOTO 210 ‘Идентифицирует команду из ПК
170 IF RIGHT$ ( R$,1) “*” THEN S$=“ “:GOTO 130
180 GOTO *KEYIN
190 CLOSE 1
200 END
210 PRINT “EXCEPTION ! ! DATA”
220 GOTO 140
```

Данные, принятые компьютером, имеют следующий вид: (КСК (Контрольная сумма кадра) = “59”) “@00EX123412341234123459*CR”

1.8.4 Связь Host Link CPM1/CPM1A

Связь Host Link была разработана фирмой OMRON для соединения программируемых контроллеров и одного или нескольких управляющих компьютеров по кабелю RS-232C и для управления связью ПК с управляющего компьютера. Обычно Управляющий компьютер выдает команду на ПК, и ПК автоматически посылает ответ. Таким образом, связь идет без активной роли ПК. У ПК есть также возможность вызвать передачу данных в случае необходимости непосредственного участия в связи.

Вообще, имеется 2 способа осуществления связи Host Link. Один основан на командах С-режима, другой на командах FINS (режим CV). CPM1/CPM1A поддерживает только команды С-режима. Подробности о связи HOST LINK см. Главу 6 Команды Host Link.

Установочные параметры ПК

Параметры периферийного порта CPM1 должны быть настроены должным образом для использования в Host Link, как показано в следующей таблице.

Слово (а)	Бит(ы)	Функция	Значение
DM 6650	00 - 07	Задание порта 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Заданное в DM 6651 (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	00
	08 - 11	Область связи для связи 1+1 через периферийный порт 0: LR 00 - LR 15.	0
	12 - 15	Режим связи 0: Host Link, 2: связь 1+1 (ведомый), 3: связь 1+1 (ведущий) 4: связь NT (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	00
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	00
	08 - 15	Формат кадр 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 бита / четн. 10: 1 бит / 8 бит / 2 бита / нечетн. 11: 1 бит / 8 бит / 2 бита / нет (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	00
DM 6652	00 - 15	Задержка передачи (Host Link) 0000..9999: в мс. (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0000)).	0000
DM 6653	00 - 07	Номер узла (Host Link) 00..31 (двоично-десятичные цифры) (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0000)).	00 .. 31
	08 - 15	Резерв	00

Замечание 1. Если используются некорректные параметры, появляется признак нефатальной ошибки, AR 1302 включится в 1 и используются настройки по умолчанию (0, 00 или 0000).

2. Информацию о параметрах HOST LINK для других ПК производства OMRON смотри руководство по работе с данными ПК.

Пример программы

В данном примере приведена программа на BASIC, которая читает состояние входов CPM1 в IR 000. Подробности см. Главу 6, команды Host Link.

Проверка контрольной суммы кадра не выполняется в данной программе. Перед выполнением программы убедитесь, что порт RS-232C ведущего компьютера сконфигурирован правильно.

```
1010 ‘CPM1/CPM1A SAMPLE PROGRAM
1020 ‘SET THE COMMAND DATA
1030 S$=“@00RR00000001”
1040 FCS=0
1050 FOR I=1 TO LEN (S$)
1060 FCS=FCS XOR ASC (MID$ (S$, I, 1))
1070 NEXT I
```

```
1080 FCS$= (FCS): IF LEN (FCS$) =1 THEN FCS$="0"+FCS$
1090 CLOSE 1
1100 CLS
1110 PRINT "SENDING COMMAND"
1120 OPEN "COM:E 73" AS #1
1130 PRINT #1, S$ + FCS + CHR$ (13);
1140 CLS
1150 PRINT "RECEIVING RESPONSE DATA"
1160 LINE INPUT #1, A$
1170 PRINT A$
1180 END
```

1.8.5 Связь Host Link SRM1

Связь Host Link была разработана фирмой OMRON для соединения программируемых контроллеров и одного или нескольких управляющих компьютеров по кабелю RS-232C и для управления связью ПК с управляющего компьютера. Обычно Управляющий компьютер выдает команду на ПК, и ПК автоматически посылает ответ. Таким образом, связь идет без активной роли ПК. У ПК есть также возможность вызвать передачу данных в случае необходимости непосредственного участия в связи.

Вообще, имеется 2 способа осуществления связи Host Link. Один основан на командах C-режима, другой на командах FINS (режим CV). SRM1 поддерживает только команды C-режима. Подробности о связи HOST LINK см. Главу 6 Команды Host Link.

Установочные параметры ПК

Параметры периферийного порта CPM1 должны быть настроены должным образом для использования в Host Link, как показано в следующей таблице.

Слово (а)	Бит(ы)	Функция	Пункт
Установочные параметры порта RS-232C Следующие параметры вступают в силу после передачи в ПК			
DM 6645	00 - 07	Задание порта 00: стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: заданное в DM 6646	1.6.1
	04 - 07	Контроль CTS: 0: запрещен 1: разрешен	
	08 - 11	Слова связи (действуют, когда биты 12..15 =3) 0: LR 00 - LR 15, другие: не работают	
	12 - 15	Режим связи 0 связь с верхним уровнем (Host Link) 1 RS-232C (без протокола) 2: связь 1+1 (ведомый) 3 связь 1+1 (ведущий) 4: Nt Link	
DM 6646	00 - 07	Скорость связи 00: 1.2 К, 01: 2.4 К, 02: 4.8 К, 03: 9.6 К, 04: 19.2 К	

Слово (а)	Бит(ы)	Функция	Пункт
	08 - 15	Формат кадра 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 бита / четн. 10: 1 бит / 8 бит / 2 бита / нечетн. 11: 1 бит / 8 бит / 2 бита / нет	
DM 6647	00 - 15	Задержка передачи (Host Link) 0000..9999 (двоично-десятичные цифры): задается в единицах 10 мс. Напр. задание 0001 = 1 мс.	
DM 6648	00 - 07	Номер узла (Host link, действуют, когда биты DM 6645 12 - 15=0) 00..31 (двоично-десятичные цифры)	
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен 1: задан	
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: разрешен (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	
DM 6649	00 - 07	Код старта (RS-232C) 00.. FF (двоичное число)	1.6.1
	08 - 15	Когда биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01..FF: 1..256 байт Когда биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00 .. FF: (двоичное число)	
Параметры периферийного порта Значения параметров вступают в силу после ввода в ПК.			
DM 6650	00 - 03	Задание порта 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Заданное в DM 6651 (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	1.6.4
	08 - 11	Резерв	
	12 - 15	Режим связи 0: Host Link, 1: свободный протокол (другие значения вызовут признак нефатальной ошибки).	

Слово (а)	Бит(ы)	Функция	Пункт
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	
	08 - 15	Формат кадр 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 бита / четн. 10: 1 бит / 8 бит / 2 бита / нечетн. 11: 1 бит / 8 бит / 2 бита / нет (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	
DM 6652	00 - 15	Задержка передачи (Host Link) 0000..9999: в мс. (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0000)).	
DM 6653	00 - 07	Номер узла (Host link, действуют, когда биты DM 6645 12 - 15=0) 00..31 (двоично-десятичные цифры)	
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен 1: задан	
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: разрешен (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	
DM 6654	00 - 07	Код старта (RS-232C) 00.. FF (двоичное число)	
	08 - 15	Код окончания Когда биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01..FF: 1..256 байт Когда биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00 .. FF: (двоичное число)	
	04 - 07	Резерв	
	08 - 11	Разрешение времени наблюдения за циклом 0: Обнаруживать длинные циклы как нефатальные ошибки 1: Не обнаруживать длинные циклы	1.2.10
	12 - 15	Резерв	

Пример программы

В данном примере приведена программа на BASIC, которая читает состояние входов SRM1 в IR 000. Подробности см. Главу 6, команды Host Link.

Проверка контрольной суммы кадра не выполняется в данной программе. Перед выполнением программы убедитесь, что порт RS-232C ведущего компьютера сконфигурирован правильно.

```
1000 '---
1010 'CPM1/CPM1A SAMPLE PROGRAM
1020 '---
1050 '---
```

```
1060 '---Set RS232C Speed: 9600BPS, Parity: Even, Data:7, Stop: 2
1070 OPEN "COM:E73" AS #1
1080 *REPEAT
1090 '---Transmission Data Input
1100 INPUT "send data:" SEND$
1110 '---FCS Calculation
1120 FCS=0
1130 FOR IFCS=1 TO LEN (SEND$)
1140 FCS=FCS XOR ASC (MID$ (SEND$, IFCS, 1))
1150 NEXT
1160 FCS$= RIGHT$("0"+HEX$(FCS),2)
1170 '---Communications execute
1180 ZZZ$=SEND$+FCS$+"*"+CHR$(13)
1190 PRINT #1,ZZZ$
1200 '---Response check
1210 RECCNT=0:TMP$=""
1220 *DRECLOOP
1230 IF LOC(1)<>0 THEN *DREC1
1240 RECCNT=RECCNT+1
1250 IF RECCNT=5000 THEN DRECERR ELSE *DRECLOOP
1260 *DREC1
1270 TMP$=TMP$+INPUT$(LOC(1),#1)
1280 IF RIGHT$(TMP$,1)=CHR$(13) THEN *DRECEND ELSE RECCNT=0:GOTO
        *DRECLOOP
1290 *DRECERR
1300 TMP$="No Response!!"+CHR$(13)
1310 *DRECEND
1320 RECV$+TMP$
1330 PRINT "recive data";RESV$
1340 '---Go to transmission data input
1350 GOTO *REPEAT
1360'---Processing complete
1370 CLOSE #1
1380 END
```

1.8.6 Связь по RS-232C (только CQM1/SRM1)

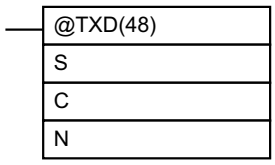
В данной главе описана связь RS-232C. Используя связь RS-232C данные можно распечатать на принтере или считать со считывателя штрихового кода. Подтверждение для RS-232C не поддерживается.

Процедура связи

Передача

1, 2, 3,... 1. Проверьте, чтобы AR 0805 (флаг готовности к передаче порта RS-232C) был включен в 1.

2. Используйте команду TDX(48) для передачи данных.



S: номер ведущего слова передаваемых данных
C: параметры управления
N: число байт, подлежащих передаче (4 цифры BCD)
0000..0256

От времени исполнения этой команды до полной передачи данных флаг AR 0805 (или AR 0813 для периферийного порта) останется = 0. (Он снова включится в положение 1 после завершения передачи данных.)

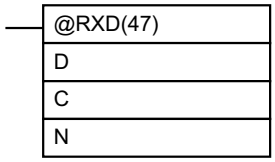
Коды старта и окончания не включены, когда задается число байт, подлежащих передаче. Наибольшее число бит, которое может быть переслано с кодами или без кодов старта и окончания - 256 байт, N будет от 254 до 256 в зависимости от задания кодов старта и окончания. Если задать число байт для передачи задано 0000, будут посланы только коды старта и окончания.

Для сброса порта RS-232C (т.е. для восстановления исходного состояния) включите SR 25209 в 1. Для сброса периферийного порта включите SR 25208 в 1. Эти биты автоматически сбросятся в 0 после сброса.

Прием

1, 2, 3,... 1. Убедитесь, что AR 0806 (флаг завершения приема RS-232C) или AR 0814 (флаг завершения приема периферийного порта) = 1.

2. Используйте команду RDX(47) для приема данных.



D: номер ведущего слова для хранения принимаемых данных
C: параметры управления
биты 00..03
0: левые байты первые
1: правые байты первые
биты 12..15
0: порт RS232C
1: периферийный порт
N: число сохраняемых байт (4 цифры BCD)
0000..0256

3. Результаты чтения полученных данных будут сохранены в области AR. Проверьте, что операция завершена успешна. Содержимое этих битов будет сброшено каждый раз при выполнении команды RDX(47).

Порт RS-232C	**Периферийный порт**	**Ошибка**
AR 0800.. AR 0803	AR 0808.. AR 0811	Код ошибки порта RS-232C (1 двоично-десятичная цифра) 0: Нормальное завершение 1: Ошибка четности 2: Ошибка шаблона 3: Ошибка переполнения
AR 0804	AR 0812	Ошибка связи
AR 0807	AR 0815	Флаг переполнения связи (после завершения приема принят следующий блок, прежде чем данные были прочитаны командой RXD(47)).
AR 09	AR 10	Число принятых байт.

Для сброса порта RS-232C (т.е. для восстановления исходного состояния) включите SR 25209 в 1. Для сброса периферийного порта включите SR 25208 в 1. Эти биты автоматически сбросятся в 0 после сброса.

Код старта и окончания не включены в AR 09 или AR 10 (число принятых байт).

Пример применения

В данном примере приведена программа использования порта RS-232C в режиме RS-232C для передачи 10 байт данных (DM 0100..DM 0104) в компьютер и загрузки полученных данных из компьютера в область DM, начиная с DM 0200. Перед исполнением программы нужно произвести следующие настройки установочных параметров.

DM 6645: 1000 (порт RS-232C в режиме RS-232C; стандартные условия связи).

DM 6648: 2000 (нет стартового кода; код окончания: CR/LF).

Для всех остальных установочных параметров значения берутся по умолчанию. В DM 0100..DM 0104 в каждом слове загружено “3132”. В компьютере выполните программу приема данных CQM1 со стандартными условиями связи.

Данные будут иметь следующий вид: “313231323132313231323132CR LF”

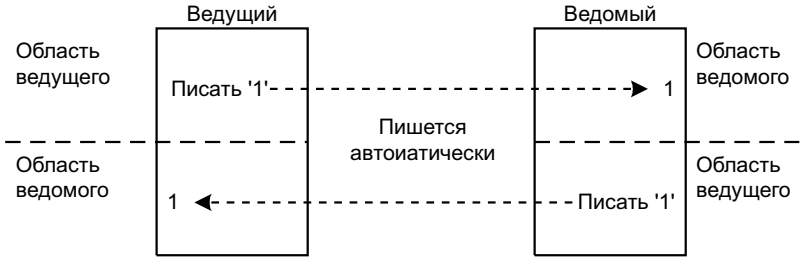
1.8.7 Связь CQM1 1:1

Если два CQM1 связаны методом 1:1 по портам RS-232C, у них общие области LR. При связи двух CQM1 методом 1:1 один из них служит в качестве ведущего, другой - в качестве ведомого.

Замечание Периферийный порт нельзя использовать для связи 1:1.

Связь 1:1

Связь 1:1 позволяет двум CQM1 иметь общие данные в их областях LR. Как показано в следующей диаграмме, когда данные переписываются в слово области LR одного из связанных Блоков, оно автоматически перепишется в другой блок. В каждом ПК имеются определенные слова, в которые он может записать и определенные слова, которые записываются другим ПК. Каждый ПК может читать, но не может писать слова, записанные другим ПК.



Слово, используемое каждым ПК, будет иметь вид, указанный в таблице, согласно настройки слов ведущего ПК, ведомого ПК и слов связи.

Задание DM 6645	**LR 00..LR 63**	**LR 00..LR 31**	**LR 00..LR 15**
Слова ведущего	LR 00..LR 31	LR 00..LR 15	LR 00..LR 07
Слова ведомого	LR 32..LR 63	LR 16..LR 31	LR 08..LR 15

Процедура связи

Если параметры для ведущего и ведомого ПК сделаны корректно, связь 1:1 стартует автоматически просто при включении питания в обоих ПК и операция будет производиться независимо от режимов работы CQM1.

Ошибки связи

Если ведомый не получает ответа от ведущего в течение 1 секунды, флаг ошибки 1:1 (AR 0802) и флаг ошибки связи (AR 0804) будет включен в 1.

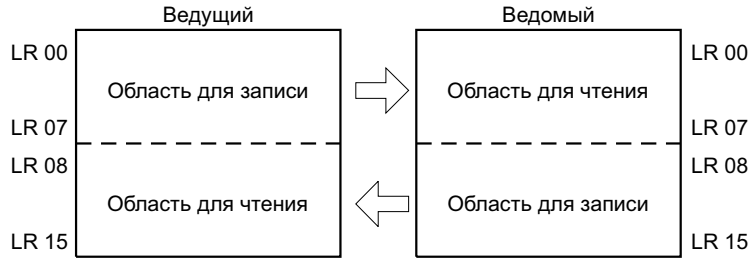
Пример применения

В данном примере приведена программа для проверки условий выполнения связи 1:1 с использованием портов RS-232C. Перед выполнением программы задайте следующие установочные параметры ПК.

Ведущий : DM 6645: 3200 (ведущий в связке 1:1; используемая область: LR 00..LR 15)

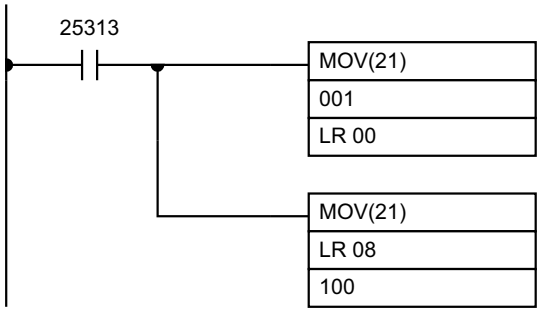
Ведомый: DM 6645: 2000 (ведомый в связке 1:1).

Для всех остальных параметров ПК настройки берутся по умолчанию. Слова, используемые для режима связи 1:1, показаны на рисунке.

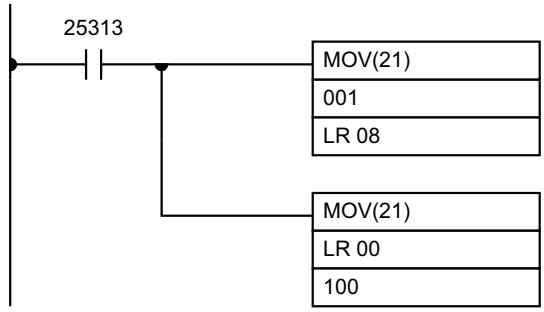


Когда программа исполняется в обоих - ведущем и ведомом ПК, состояние IR 001 каждого Блока будет отражено в IR 100 другого Блока. Аналогично состояние IR 001 другого блока будет отражено в IR 100 каждого ПК. IR 001 является входным словом, а IR 100 - выходным словом.

Программа ведущего



Программа ведомого

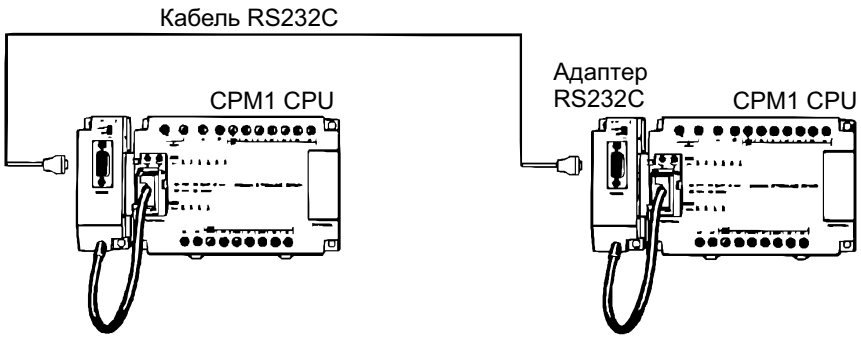


1.8.8 Связь 1:1 CPM1/CPM1A

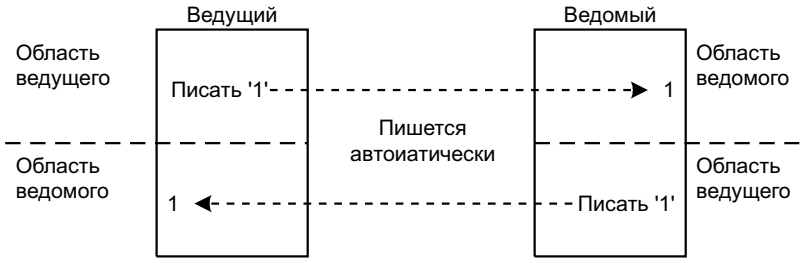
В связке 1:1 один CPM1/CPM1A связан с другим ПК через адаптер RS-232C и стандартный кабель RS-232C. Один из ПК служит в качестве ведущего, другой - в качестве ведомого. Режим связи 1:1 может объединять до 256 бит (LR 0000..LR 1515) в двух ПК.

Связь 1:1 CPM1/CPM1A

На следующей диаграмме показана связь 1:1 между двумя CPM1.



Слова, используемые для связи 1:1, показаны ниже:



Ограничения на связь 1:1 с CPM1

Только 16 слов LR от LR 00 до LR 15 могут быть общими в CPM1, поэтому в CQM1 или C200HS используйте только эти слова при организации связи с этими ПК. Связь 1:1 нельзя организовать, используя в CQM1 или C200HS слова LR 16..LR 63 для связи с CPM1.

Установочные параметры ПК

Параметры, относящиеся к связи 1:1, должны быть настроены, как показано в следующей таблице.

Слово	Бит	Функция	Настройка ведущего	Настройка ведомого
DM 6650	00 - 07	Задание порта[1] 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Заданное в DM 6651	00 (опция)	00 (опция)

Слово	Бит	Функция	Настройка ведущего	Настройка ведомого
	08 - 11	Область связи для связи 1:1 через периферийный порт 00: LR 00..LR 15.	0	0 (опция)
	12 - 15	Режим связи[1] 0: Host Link 2: 1:1 (ведомый) 3: 1:1 (ведущий) 4: NT LINK	3	2

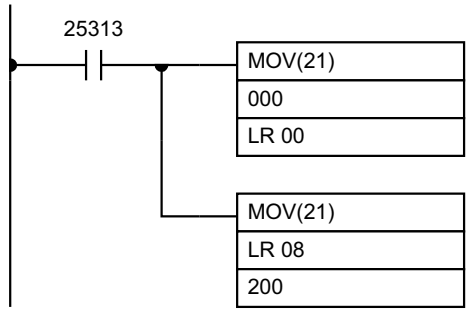
Замечание 1. Если используются некорректные параметры, появится признак нефатальной ошибки, AR 1302 включится в 1 и будут использоваться настройки по умолчанию (0 или 00).

2. Информацию о настройке связи 1:1 других ПК производства OMRON см. руководство по работе данных ПК.

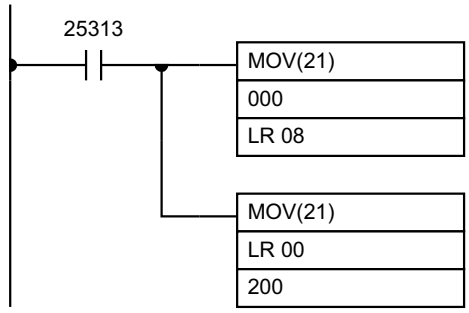
Пример применения

В данном примере приведена программа релейно-контактной схемы, копирующая состояние IR 000 каждого CPM1 в SR 200 другого CPM1.

Программа ведущего



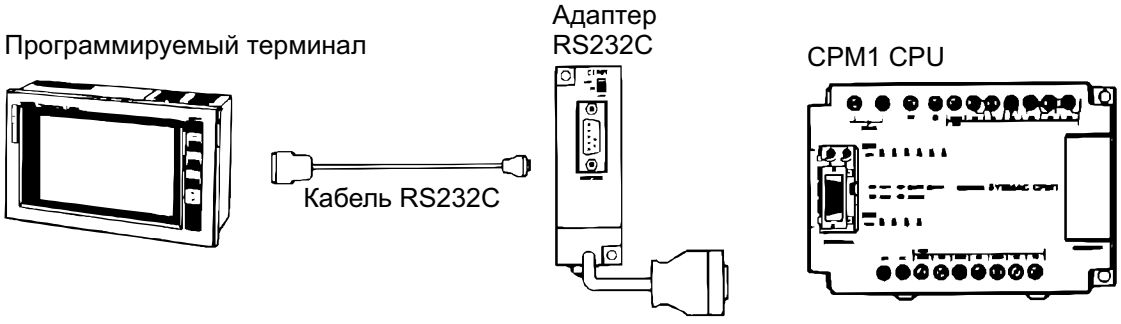
Программа ведомого



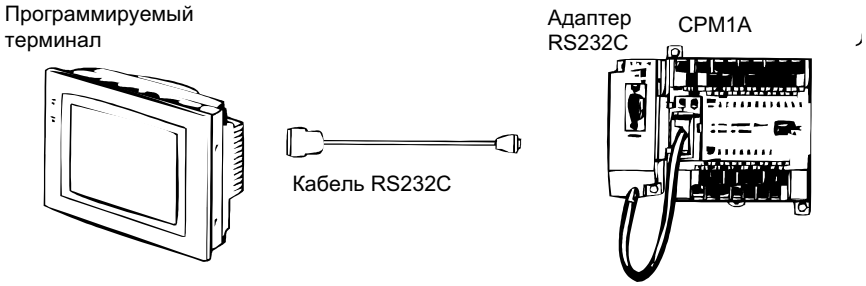
1.8.9 Связь CPM1/CPM1A NT LINK

Используя режим связи NT LINK, CPM1 может подключаться к программируемому Терминалу (интерфейс NT LINK) через адаптер RS-232C.

CPM1



CPM1A



Установочные параметры ПК

Параметры, относящиеся к режиму связи NT, должны быть настроены в соответствии с таблицей.

Слово	Бит	Функция	Задание
DM 6650	00 - 07	Задание порта[1] 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Задания в DM 6651	00 (опция)
	08 - 11	Область связи для связи 1:1 через периферийный порт 0: LR 00..LR 15	0 (опция)
	12 - 15	Режим связи[1] 0: Host Link 2: 1:1 (ведомый) 3: 1:1 (ведущий) 4: NT LINK	4

Замечание 1. Если используются некорректные параметры, появится признак нефатальной ошибки, AR 1302 включается в 1, будут использоваться значения по умолчанию (0 или 00).

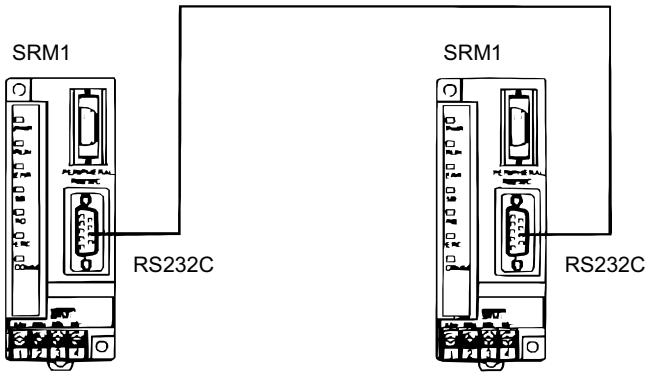
2. Информацию о параметрах NT LINK для других ПК производства OMRON см. руководство по работе данных ПК.

1.8.10 Связь 1:1 SRM1

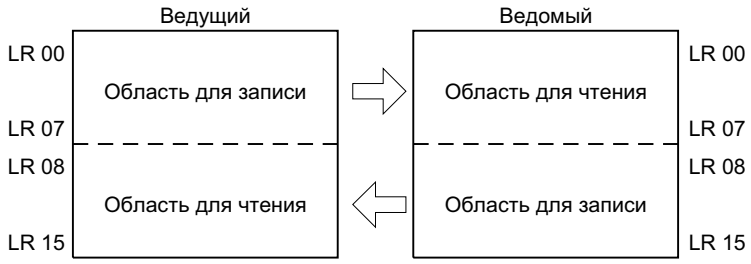
При связи 1+1 один SRM1 связан с другим ПК через адаптер RS-232C и стандартный кабель RS-232C. Один из ПК служит в качестве ведущего, другой - в качестве ведомого. Связь 1+1 может объединять до 256 бит (LR 0000..LR 1515) в двух ПК.

Связь 1:1 в SRM1

На следующей схеме показана связь 1+1 между двумя SRM1.



Слова, используемые для связи 1+1, показаны ниже:



Ограничения при связи 1:1 SRM1

Только 16 слов LR (LR 00 .. LR 15) могут быть общими в SRM1, поэтому в CQM1 или C200HS используйте только эти слова при организации связи с этими ПК. Связь 1+1 с SRM1 нельзя организовать, используя в CQM1 или C200HS слова LR 16 .. LR 63.

Установочные параметры ПК

Параметры, относящиеся к связи 1+1, должны быть настроены в соответствии со следующей таблицей.

Слово	Бит	Функция	Настройка ведущего	Настройка ведомого
DM 6645	00 .. 03	Задание порта[1] 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Заданное в DM 6651	Любая	Любая
	04 .. 07	Разрешение CTS (Clear to send - сброс передатчика) 0: Запретить 1: Разрешить	0	0
	08 .. 11	Область связи для связи 1+1 через периферийный порт 0: LR 00..LR 15.	0	0
	12 .. 15	Режим связи[1] 0: Host Link 1: без протокола 2: 1:1 (ведомый) 3: 1:1 (ведущий) 4: NT LINK	3	2

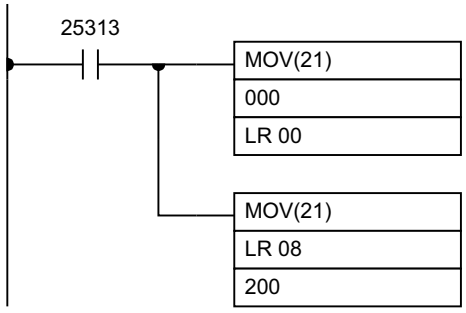
Замечание 1. Если используются некорректные параметры, появится признак нефатальной ошибки, AR 1302 включится в 1 и будут использоваться настройки по умолчанию (0 или 00).

2. Информацию о настройке связи 1+1 других ПК производства OMRON см. руководство по работе данных ПК.

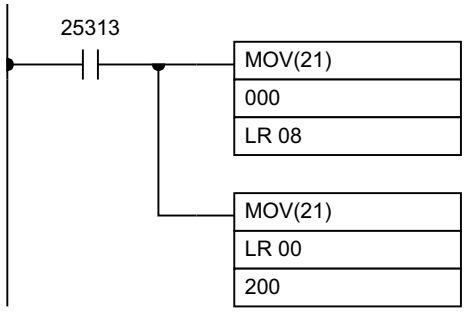
Пример применения

В данном примере приведена программа релейно-контактной схемы, копирующая состояние IR 000 каждого SRM 1 в SR 200 другого SRM1.

Программа ведущего

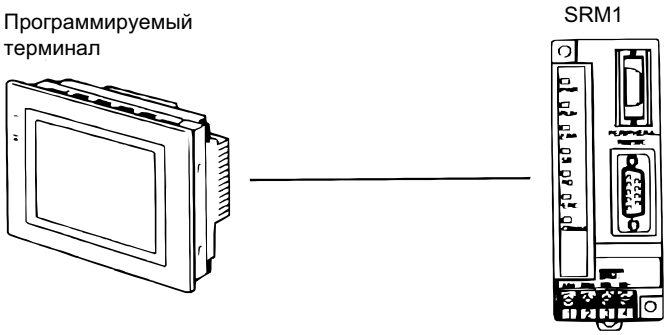


Программа ведомого



1.8.11 Связь NT LINK в SRM1

Используя режим связи NT LINK, SRM1 может подключаться к программируемому Терминалу (интерфейс NT LINK). Для NT LINK можно использовать порт RS-232C.



NT LINK возможен только у SRM1-C02, в котором есть порт RS-232C.

Установочные параметры ПК

Параметры, относящиеся к режиму связи NT, должны быть настроены в соответствии с таблицей.

Слово (а)	Бит(ы)	Функция	Задание
DM 6645	00 - 07	Задание порта 00: стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: заданное в DM 6646	00
	04 - 07	Контроль CTS: 0: запрещен 1: разрешен	0
	08 - 11	Слова связи (действуют, когда биты 12..15 =3) 0: LR 00 - LR 15, другие: не работают	0
	12 - 15	Режим связи 0 связь с верхним уровнем (Host Link) 1 RS-232C (без протокола) 2: связь 1+1 (ведомый) 3 связь 1+1 (ведущий) 4: Nt Link	4

Замечание 1. Если используются некорректные параметры, появится признак нефатальной ошибки, AR 1302 включается в 1, будут использоваться значения по умолчанию (0 или 00).

2. Информацию о параметрах NT LINK для других ПК производства OMRON см. руководство по работе данных ПК.

1.8.12 Свободный протокол в SRM1

Параметры периферийного порта

Когда периферийный порт используется для осуществления связи без протокола, в SRM1 должны быть настроены следующие параметры с периферийного устройства в DM 6650 .. DM 6653.

Слово (а)	Бит(ы)	Функция	Задание
Параметры периферийного порта Значения параметров вступают в силу после ввода в ПК.			
DM 6650	00 - 03	Задание порта 00: Стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: Заданное в DM 6651 (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	-
	08 - 11	Резерв	0
	12 - 15	Режим связи 0: Host Link, 1: свободный протокол (другие значения вызовут признак нефатальной ошибки).	1
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	-
	08 - 15	Формат кадр 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 бита, четн. 10: 1 бит, 8 бит, 2 бита, нечетн. 11: 1 бит, 8 бит, 2 бита, нет (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0)).	-
DM 6652	00 - 15	Задержка передачи (Host Link) 0000..9999: в мс. (другие значения вызовут признак нефатальной ошибки и будет использовано значение по умолчанию (0000)).	-
DM 6653	00 - 07	Номер узла (Host link, действуют, когда биты DM 6645 12 - 15=0) 00..31 (двоично-десятичные цифры)	-
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен 1: задан	-
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: разрешен (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	-
DM 6654	00 - 07	Код старта (RS-232C) 00.. FF (двоичное число)	-
	08 - 15	Код окончания Когда биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01..FF: 1..256 байт Когда биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00 .. FF: (двоичное число)	-

Параметры порта RS232C

Когда порт RS-232C используется для осуществления связи без протокола, в SRM1 должны быть настроены следующие параметры с периферийного устройства в DM 6645..DM 6649.

Слово (а)	Бит(ы)	Функция	Значение
Установочные параметры порта RS-232C Следующие параметры вступают в силу после передачи в ПК			
DM 6645	00 - 07	Задание порта 00: стандартное (1 стартовый бит, данные 7 бит, четн., 2 стоповых бита, 9 600 бод) 01: заданное в DM 6646	-
	04 - 07	Контроль CTS: 0: запрещен 1: разрешен	-
	08 - 11	Слова связи (действуют, когда биты 12..15 =3) 0: LR 00 - LR 15, другие: не работают	0
	12 - 15	Режим связи 0 связь с верхним уровнем (Host Link) 1 RS-232C (без протокола) 2: связь 1+1 (ведомый) 3 связь 1+1 (ведущий) 4: Nt Link	1
DM 6646	00 - 07	Скорость связи 00: 1.2 К, 01: 2.4 К, 02: 4.8 К, 03: 9.6 К, 04: 19.2 К	-
	08 - 15	Формат кадра 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 бита / четн. 10: 1 бит / 8 бит / 2 бита / нечетн. 11: 1 бит / 8 бит / 2 бита / нет	-
DM 6647	00 - 15	Задержка передачи (Host Link) 0000..9999 (двоично-десятичные цифры): задается в единицах 10 мс. Напр. задание 0001 = 1 мс.	-
DM 6648	00 - 07	Номер узла (Host link, действуют, когда биты DM 6645 12 - 15=0) 00..31 (двоично-десятичные цифры)	-
	08 - 11	Разрешение кода пуска (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен 1: задан	-
	12 - 15	Разрешение кода окончания (RS-232C, действуют, когда биты DM 6645 12..15 =1) 0: не разрешен (прием заданного числа байт) 1: разрешен (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	-
DM 6649	00 - 07	Код старта (RS-232C) 00.. FF (двоичное число)	-
	08 - 15	Когда биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: значение по умолчанию (256 байт) 01..FF: 1..256 байт Когда биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00 .. FF: (двоичное число)	-

1.8.13 Конфигурация передаваемых данных

Когда используется режим связи без протокола, команда TXD(48) служит для посылки данных, а RXD(47) для приема данных. Максимальное количество данных, которые можно передать или принять - 259 байт, включая код старта/окончания.

Нет кода старта или окончания

Данные (256 байт макс.)

Только стартовый код

ST	Данные (256 байт макс.)

Только конечный код

Данные (256 байт макс.)	ED

И стартовый и конечный код

ST	Данные (256 байт макс.)	ED

CR, LF заданы в качестве конечного кода

Данные (256 байт макс.)	CR	LF

Стартовый код 00-FF/ конечный код CR,LF

ST	Данные (256 байт макс.)	CR	LF

Замечание 1. Код старта и окончания задаются в установочных параметрах DM 6648 .. DM 6649, DM 6653 .. DM 6654.

2. Когда имеется несколько кодов старта и окончания, первая часть каждого будет действующей.

3. Когда код окончания дублирует передаваемые данные и передача остановлена на полпути, используйте в качестве кода окончания CR или LF.

4. Коды старта и окончания не сохраняются.

1.8.14 Флаги передачи

При передаче данных с SRM1 проверяйте, чтобы Флаг Разрешения Передачи = 1 для выполнения команды TDX(48). Флаг Разрешения Передачи будет = 0 во время передачи данных и снова включится в 1 по завершении передачи.

После того, как SRM1 принял данные, Флаг Разрешения Приема включается в 1. Когда исполняется команда RXD, принимаемые данные будут записаны в указанные слова и Флаг Завершения Приема включится в 0.

Флаг	**Периферийный порт**	**Порт RSS-232C**
Флаг разрешения передачи	AR 0813	AR 0805
Флаг завершения приема	AR 0814	AR 0806

Замечание Распределение времени от начала приема в SRM1 до завершения показано далее.

Начало приема	
Без стартового кода:	Состояние нормального приема
Со стартовым кодом:	После приема стартового кода

Завершение приема

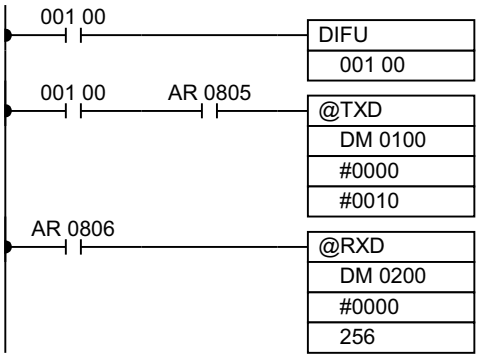
При приеме либо кода окончания, либо заданного числа байт, либо 256 байт.

1.8.15 Пример программы связи без протокола

Следующий пример иллюстрирует связь без протокола по порту RS-232C с помощью команд TXD(48) и RXD(47).

Когда используется режим связи без протокола, команда TXD(48) служит для посылки данных, а RXD(47) для приема данных. Максимальное количество

Если AR 0805 (Флаг разрешения передачи) = 1, когда 00100 = 1, данные из DM0100 .. DM0104 передаются, начиная со старших к младшим значениям. Когда AR 0806 (Флаг разрешения приема) = 1, 256 байт принимаемых данных читаются и сохраняются в DM 0200 от старших к младшим значениям



1.9 Вычисления с двоичными данными со знаком

CQM1 и CPM1 позволяют вести расчеты с двоичными данными со знаком. Следующие команды работают с двоичными данными со знаком. Расчеты с данными со знаком ведутся с использованием дополнения до 2.

Команды CQM1

В CQM1 есть следующие команды, работающие с двоичными данными со знаком.

Команды над словами обычной длины

- ДОПОЛНЕНИЕ ДО 2 - NEG(-)
- СЛО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ - ADB(50)
- ВЫЧИТАНИЕ ДВОИЧНЫХ ЧИСЕЛ - SBB(51)
- УМНО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ СО ЗНАКОМ - MBS (-)
- ДЕЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ СО ЗНАКОМ - DBS(-)

Команды с словом двойной длины

- ДОПОЛНЕНИЕ ДО 2 ЧИСЛА ДВОЙНОЙ ДЛИНЫ - NEGL(-)
- СЛО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ - ADBL(-)
- ВЫЧИТАНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ - SBBL(-)
- УМНО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ СО ЗНАКОМ MBSL(-)
- ДЕЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ СО ЗНАКОМ - DBSL (-)

Команды CPM1/CPM1A

В CPM1/CPM1A есть следующие команды, работающие с двоичными данными со знаком:

- СЛО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ - ADB(50)
- ВЫЧИТАНИЕ ДВОИЧНЫХ ЧИСЕЛ - SBB(51)

Вычисления с числами со знаком

Сложение	**Вычитание**
7 + 3 = 10	7- 3 = 4
(-7) + 3 = -4	(-7) - 3 = -10
7+ (-3) = 4	7- (-3) = 10
(-7) + (-3) = -10	(-7) - (-3) = -4
Умножение	**Деление**
7 x 3 = 21	7 / 3 = 2 и 1 в остатке
(-7) x 3 = -21	(-7) / 3 = -2 и -1 в остатке
7 x (-3) = -21	7 / (-3) = -2 и 1 в остатке
(-7) x (-3) = 21	(-7) / (-3) = 2 и -1 в остатке

1.9.1 Задание двоичных данных со знаком

В CQM1 есть команды, работающие с данными длиной либо в одно, либо в 2 слова. В CPM1 есть только две команды, работающие с данными длиной в одно слово. Расчеты с двоичными данными со знаком ведутся с использованием дополнением до 2, а старший разряд числа из одного или двух слов используется в качестве знакового бита. Диапазон данных, которые могут быть выражены с использованием одного или двух слов:

- Данные длиной 1 слово: -32 768..32 767 (16-ричные 8000..7FFF)
- Данные длиной 2 слов: -2 147 483 648..2 147 483 647 (16-ричные 8000 0000..7FFF FFFF)

В следующей таблице показаны эквиваленты десятичных и 16-ричных чисел.

Десятичные	**16-ричные 16 бит**	**16-ричные 32 бита**
2 147 483 647	-	7FFF FFFF
2 147 483 646	-	7FFF FFFE
.	.	.

Десятичные	16-ричные 16 бит	16-ричные 32 бита
.	.	.
.	.	.
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

1.9.2 Арифметические флаги

Результаты выполнения команд двоичных чисел со знаком отражены в арифметических флагах. Флаги и условия, при котором они включаются в 1, приведены в следующей таблице. Флаги включаются в 0, когда эти условия не выполнены.

Флаг	Условия включения в 1
Флаг переноса (SR 25504)	Перенос при сложении. При вычитании отрицательное значение.
Флаг равно (SR 25506)	Результаты сложения, вычитания, умножения или деления равны 0.
Флаг переполнения + (SR 25404)	Превышено значение 32 767 (7FFF) в результате 16-битового сложения или вычитания. Превышено значение 2 147 483 647 (7FFF FFFF) в результате 32-битового сложения или вычитания.
Флаг переполнения - (SR 25405)	Превышено значение -32 768 (8000) в результате 16-битового сложения или вычитания или преобразования дополнения до 2. Превышено значение -2 147 483 648 (8000 0000) в результате 32-битового сложения или вычитания или преобразования дополнения до 2.

1.9.3 Ввод двоичных чисел со знаком с использованием десятичных значений

Хотя вычисления с двоичными числами со знаком используют 16-ричные выражения, ввод с программатора или SSS можно произвести с использованием десятичного ввода и мнемоники для команд. Процедура использования программатора для ввода десятичных значений описана в Инструкции по работе CQM 1 и Инструкции по работе CPM1/CPM1A; Подробности об использовании SSS см. Инструкцию по работе с SSS, ПК серии C.

Команды ввода

Только операнды 16 бит могут вводиться для следующих команд: NEG(–), ADB(50), SBB(50), MBS(-), DBS(-). Подробности о вводе команд с программатора см. Инструкцию по работе CQM1 или Инструкцию по работе CPM1.

1.9.4 Использование дополнительных команд двоичных со знаком (только CQM1)

Следующим командам CQM1 необходимо перед применением присвоить функциональные коды в таблице команд.

- ДОПОЛНЕНИЕ ДО 2 - NEG(-)
- ДОПОЛНЕНИЕ ДО 2 ЧИСЛА ДВОЙНОЙ ДЛИНЫ - NEGL(-)
- СЛО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ - ADBL(-)
- ВЫЧИТАНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ - SBBL(-)
- УМНО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ СО ЗНАКОМ - MBS (-)
- УМНО ЕНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ СО ЗНАКОМ MBSL(-)
- ДЕЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ СО ЗНАКОМ - DBS(-)
- ДЕЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ ДВОЙНОЙ ДЛИНЫ СО ЗНАКОМ - DBSL (-)

Присвоение функциональных кодов

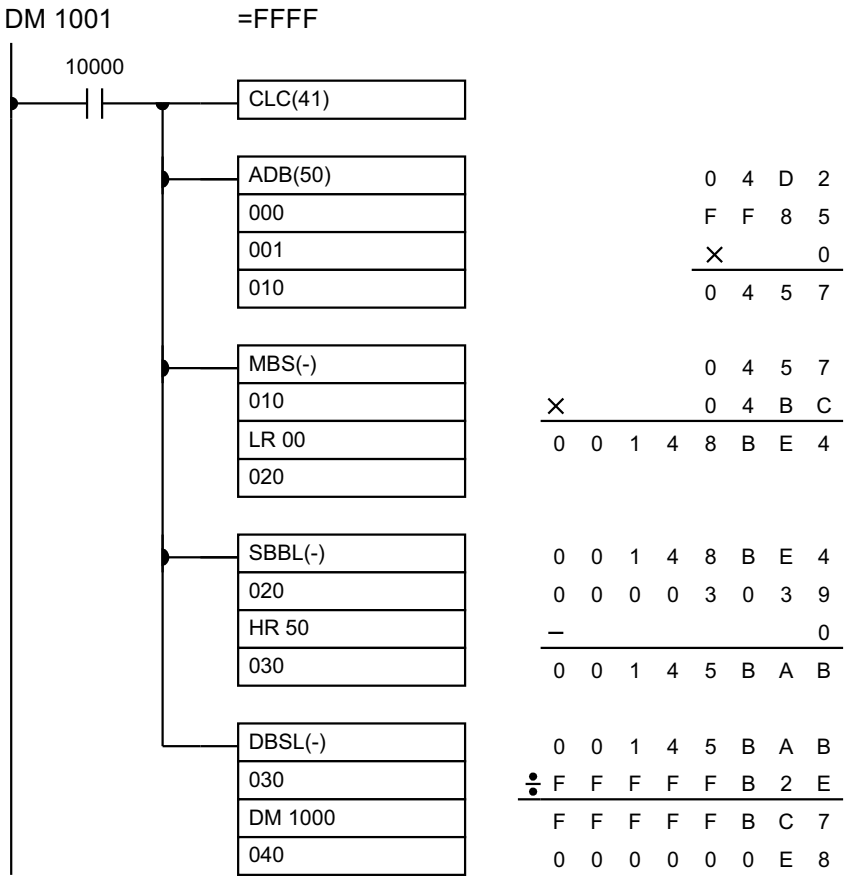
Процедура использования программатора для присвоения функциональных кодов описана в Инструкции по работе CQM1. Следите за тем, чтобы перед началом операции секция 4 на переключателя DIP была в состоянии ON для разрешения использования таблицы команд.

1.9.5 Пример применения двоичных чисел со знаком

Следующие приемы программирования можно использовать для выполнения подобных расчетов в CQM1:

((1234+(-123)) x 1212 - 12345) (-1234) = -1081, остаток 232

000	=04D2	1234
001	=FF85	-123
LR00	=04BC	1212
HR50	=3039	12345
HR51	=0000	
DM 1000	=FB2E	-1234


DM 1001 =FFFF
10000
CLC(41)
ADB(50)
000
001
010
MBS(-)
010
LR 00
020
SBBL(-)
020
HR 50
030
DBSL(-)
030
DM 1000
040
0 4 D 2
F F 8 5
× 0
0 4 5 7
0 4 5 7
× 0 4 B C
0 0 1 4 8 B E 4
0 0 1 4 8 B E 4
0 0 0 0 3 0 3 9
– 0
0 0 1 4 5 B A B
0 0 1 4 5 B A B
÷ F F F F F B 2 E
F F F F F B C 7
0 0 0 0 0 0 E 8

2. Новые возможности

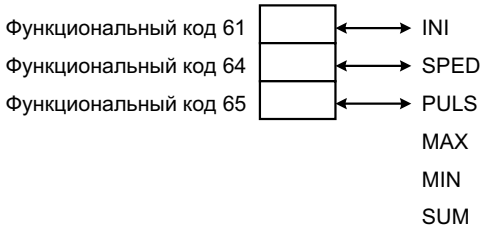
В данной главе дано описание новых возможностей CQM1/CPM1/CPM1A/SRM1, включая новые дополнительные команды, новые возможности, называемые просмотром фронтов, и функции аналоговых регуляторов в CQM1-CPU42-EV1 и CPM1/CPM1A.

Если Вы не знакомы с ПК OMRON или программированием в релейно-контактном виде, Вы можете только просмотреть данную главу для общего знакомства с новыми возможностями, но перед детальным знакомством с данной главой Вам следует ознакомиться с Главами 3 и 4. Подробности о командах программирования можно найти в Главе 5 Набор команд.

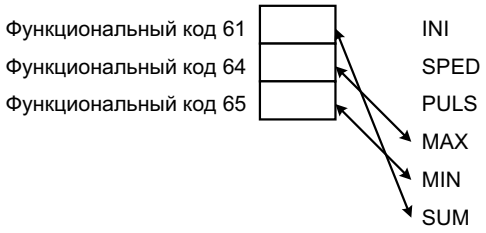
2.1 Дополнительные команды (только CQM1/SRM1)

Для специальных задач программирования в CQM1 имеется набор дополнительных команд. Функциональные коды можно присвоить до 18 дополнительным командам для их разрешения их использования в программах. Это позволяет пользователю выбирать команды, требуемые для каждой программы CQM1, чтобы более эффективно использовать функциональные коды, требуемые для команд ввода.

Мнемоники дополнительных команд сопровождается знаком “(-)” в качестве функционального кода чтобы показать, что перед тем, как будет разрешено из использование в программе, пользователь должен присвоить им функциональные коды в таблице команд.



При поставке функциональные коды присвоены следующим образом (в данном примере команды относятся к выдаче импульсов)



Если импульсные выходы не используются, а требуются специальные математические функции, то можно прменить операцию 'Присвоение команд' для переприсваивания команд в таблице команд

2.1.1 Дополнительные инструкции CQM1

Для дополнительных команд можно использовать 18 функциональных кодов: 17, 18, 19, 47, 48, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 87, 88, и 89.

Дополнительные команды, которые можно использовать, приведены в таблице, наряду с функциональными кодами по умолчанию, которые присваиваются при поставке CQM1. Команды, помеченные * есть только у ЦУ CQM1-CPU4_-EV1.

Мнемоника	**Функциональный код**
ASFT	17
TKY	18
MCMP	19
RXD	47
TXD	48
CMPL	60
INI	61
PRV	62
CTBL	63
SPEED	64
PULS	65
SCL	66
BCNT	67
BCMP	68

Мнемоника	Функциональный код
STIM	69
DSW	87
7SEG	88
INT	89
HKY	-
FPD	-
SRCH	-
MAX	-
MIN	-
APR	-
LINE	-
COLM	-
SEC	-
HMC	-
SUM	-
FCS	-
HEX	-
AVG	-
PVM*	-
PID*	-
ADBL*	-
SBBL*	-
MBS*	-
DBS*	-
MBSL*	-
DBSL*	-
CPS*	-
NEG*	-
NEGL*	-
ZCP*	-
ZCPL*	-
XFRB*	-
PLS2*	-
ACC*	-
SCL2*	-
SCL3*	-

Любой команде с неприсвоенными функциональными кодами нужно присвоить функциональные коды в таблице команд, которую используют устройство программирования и CQM1, прежде чем их можно будет использовать в программировании. Присвоения дополнительных команд в таблице команд изменит значение команд и операндов, поэтому перед программированием обязательно составьте таблицу команд и перед выполнением программы передайте нужную таблицу команд в CQM1.

Таблицу команд можно также хранить в кассете памяти. Будьте внимательны при использовании кассеты памяти с другим CQM1 и следите за тем, чтобы присутствовала нужная таблица команд.

Внимание! Если секция 4 переключателя DIP CQM1 стоит в положении OFF, можно использоваться только таблицу дополнительных команд по умолчанию, а

таблица команд пользователя будет игнорироваться. Таблица команд по умолчанию будет также установлена после включения питания, удаляя предыдущую.

Убедитесь, что секция 4 переключателя DIP ЦУ находится в положение ON при чтении программы с кассеты памяти, в которой находится пользовательская таблица дополнительных команд. Если секция 4 стоит в положении OFF, в программе, считанной с кассеты памяти, будет применяться таблица команд по умолчанию. (В таком случае при сравнении программа, считанная с кассеты памяти и находящаяся на кассете памяти, не будут совпадать).

2.1.2 Дополнительные инструкции SRM1

Для дополнительных команд можно использовать 18 функциональных кодов: 17, 18, 19, 47, 48, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 87, 88, и 89.

Дополнительные команды, которые можно использовать, приведены в таблице, наряду с функциональными кодами по умолчанию, которые присваиваются при поставке SRM1.

Мнемоника	**Функциональный код**
ASFT	17
RXD	47
TXD	48
CMPL	60
BCNT	67
BCMP	68
STIM	69
FCS	-
HEX	-
STUP	-

2.2 Новые команды входа/выхода (только CQM1)

Новые команды входа/выхода позволяют производить одной командой ранее сложные операции с внешними устройствами входа/выхода (цифровые переключатели, 7-сегментные индикаторы и т. д.). Данная глава знакомит с новыми командами входа/выхода, которые также освещены в Главе 5 Набор команд.

Есть 4 новых команд ввода/вывода, как показано в следующей таблице,. Все они являются дополнительными командами и перед использованием им должны быть присвоены функциональные коды.

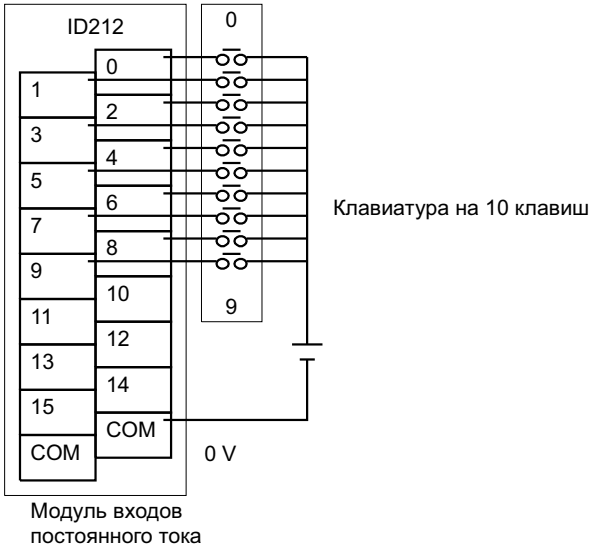
Название	**Мнемоника**	**Функция**
Ввод с клавиатуры (10 клавиш)	TKY(18)	Ввод двоично-десятичных цифр с клавиатуры 10 клавиш
Ввод с клавиатуры (16 клавиш)	HKY(-)	Ввод 16-ричных цифр с клавиатуры 16 клавиш
Ввод с цифрового переключателя	DSW(87)	Ввод задания с цифрового переключателя
Вывод на 7-сегментный индикатор	7SEG(88)	Вывод на 7-сегментный индикатор двоично-десятичных цифр

2.2.1 Ввод с клавиатуры на 10 клавиш TKY(18)

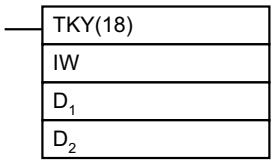
Данная команда вводит 8 двоично-десятичных цифр с клавиатуры 10 клавиш и использует 10 точек входов.

Аппаратная часть

Приготовьте клавиатуру 10 клавиш и подключите их так, чтобы цифровые клавиши 0..9 являлись входами 0..9, как показано на следующей схеме. Можно пользоваться и входами на ЦУ и входами с блоков на 16 и более входов постоянного тока.

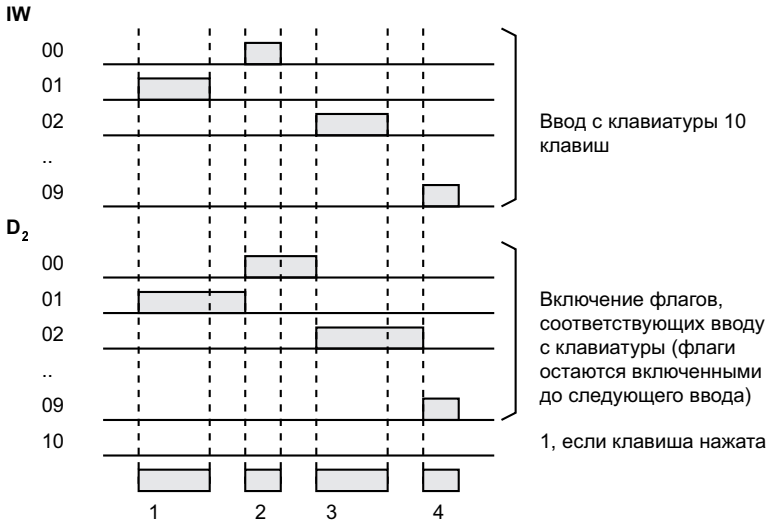


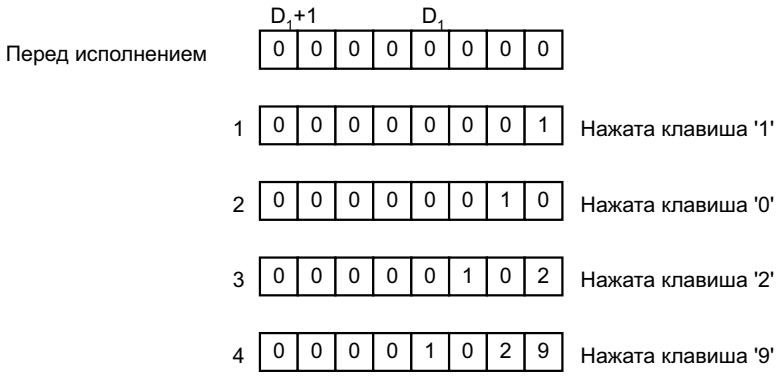
Применение команды



IW: входное слово
D1: адрес первого слова регистра
D2: слово ввода клавиш

Если для IW задано слово входа с клавиатурой 10 клавиш, при выполнении программы работа будет происходить следующим образом:





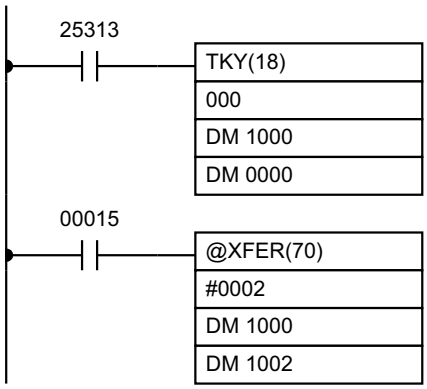
Замечание 1. Когда нажата одна клавиша, ввод с других клавиш не происходит.

2. Если вводится более 8 цифр, они будут удаляться, начиная с самой левой.

3. Входные биты, не используемые здесь, можно использовать как обычные входные биты.

Пример применения

В данном примере показана программа для ввода с клавиатуры на 10 клавиш. Предполагается, что клавиатура-10 подключена к IR 000.



Информация, введенная в IR 000 с помощью TKY(18), преобразуется в двоично-десятичный вид и хранится в DM 1000..DM 1001. Информация о клавишах хранится в DM 1002.

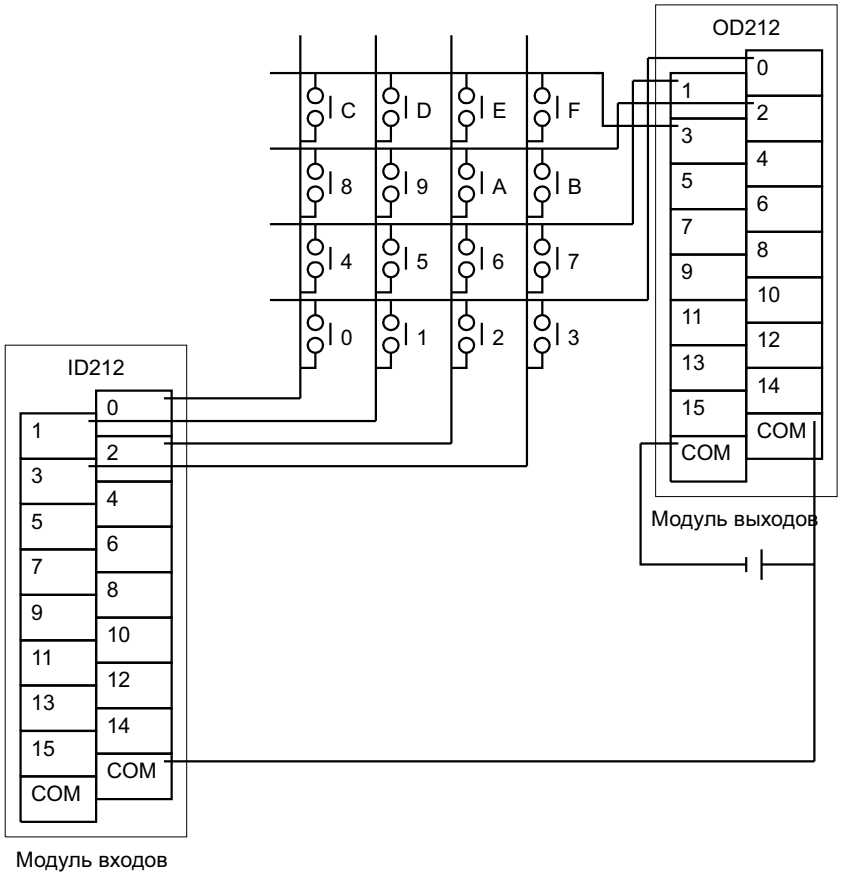
IR 00015 используется в качестве клавиши “ВВОД”, и когда IR 00015 включено в 1, данные, хранящиеся в DM 1000 и DM 1001 будут переписаны в DM 0000 и DM 0001.

2.2.2 Ввод с клавиатуры (16 клавиш) HKY(-)

Данная команда вводит восемь 16-ричных цифр клавиатуры 16 клавиш. Она использует 5 выходных бита и 4 входных бита.

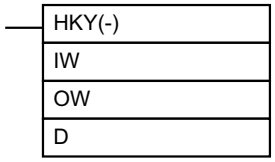
Аппаратная часть

Приготовьте клавиатуру 16 клавиш и подключите клавиши 0..F в соответствии со следующей схемой, к входам 0..3 и выходам 0..3. Выход 4 будет = 1 при нажатии любой клавиши, но подключать его не требуется.



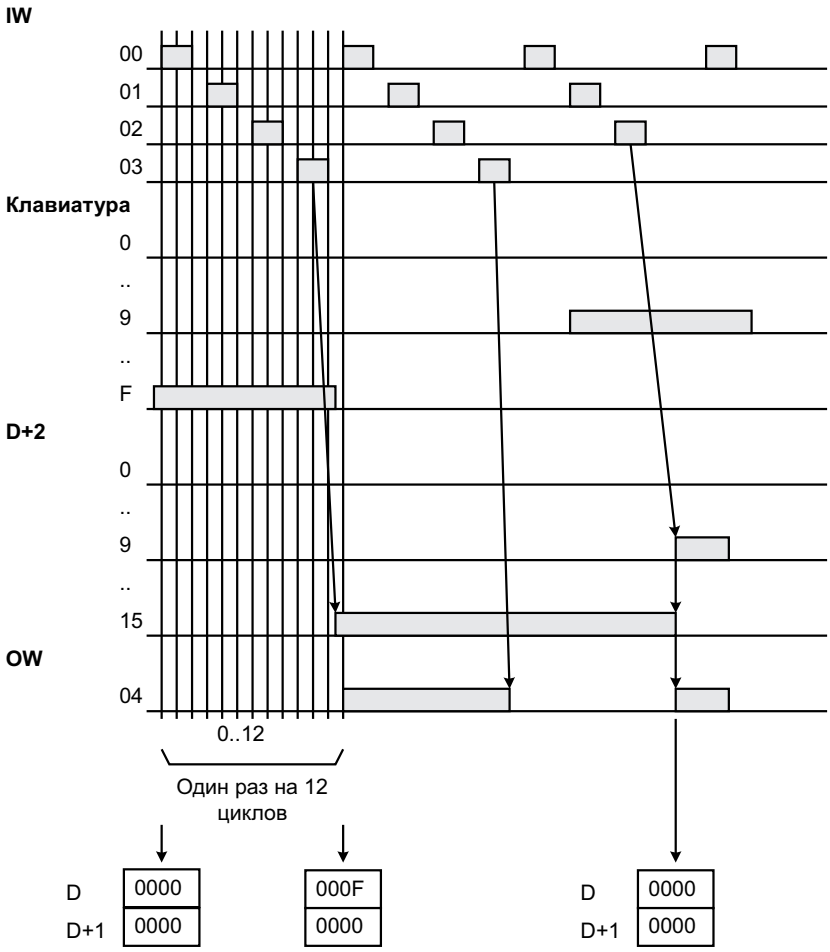
Входы можно подключать к входным клеммам на ЦУ или блоков на 8 и более входов постоянного тока, выходы можно подключать к блокам транзисторных выходов на 8 выходов и более.

Применение команды



IW: входное слово
OW: выходное слово сигнала управления
D: слово первого регистра

Если входное слово для подключения клавиатуры 16 клавиш задано как IW, а выходное слово задано в OW, при выполнении программы работа будет происходить следующим образом:



SR 25408 будет = 1 во время выполнения HKY(-).

Замечание 1. Не используйте HKY(-) в одной и той же программе больше чем 1 раз.

2. При использовании HKY(-) задайте входную константу соответствующего входа меньше, чем время цикла (Входные константы можно изменять от DM 6620 и далее).

3. Когда нажата одна клавиша, ввод с других клавиш будет недоступен.

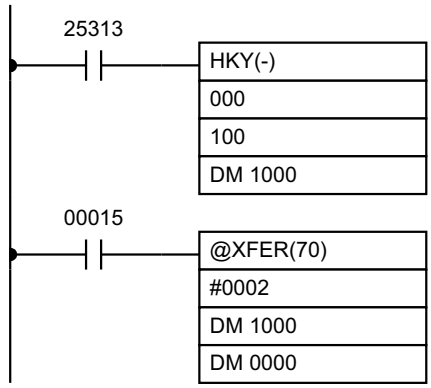
4. Если вводится более 8 цифр, они будут удаляться, начиная с самой левой.

5. Биты входа и выхода, не использованные здесь, можно использовать как обычные входные и выходные биты.

При данной команде одна клавиша читается в течение 3..12 циклов. Более одного цикла требуется потому, что состояние 1 клавиши можно будет определить только после включения выходов в 1 для их опроса.

Пример применения

В данном примере показана программа для ввода с клавиатуры на 16 клавиш. Предполагается, что клавиатура подключена к входам IR 000 и выходам IR 100.



Информация клавиатуры-16, введенная в IR 000 с помощью HKY(-), преобразуется в 16-ричный вид и хранится в DM 1000 и DM 1001.

IR 00015 используется в качестве клавиши “ВВОД”, и когда IR 00015 = 1, числа, хранящиеся в DM 1000 и DM 1001, передаются в DM 0000 и DM 0001.

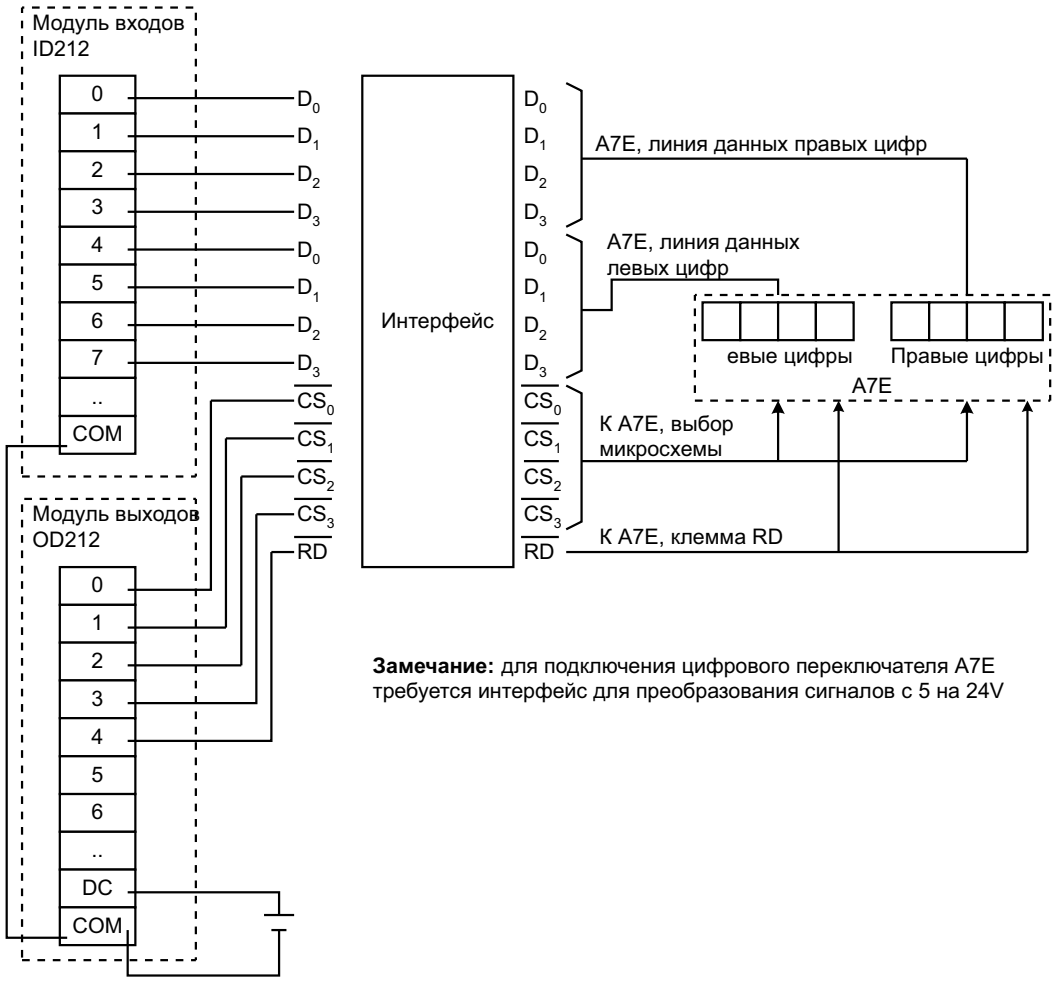
2.2.3 Ввод с цифрового переключателя - DSW(87)

Данной командой читаются значения из 4 или 8 двоично-десятичных цифр с цифрового переключателя. DSW(87) использует 5 выходных бита и либо 4 входных бита (для 4 цифр) или 8 входных бит (для 8 цифр).

Аппаратная часть

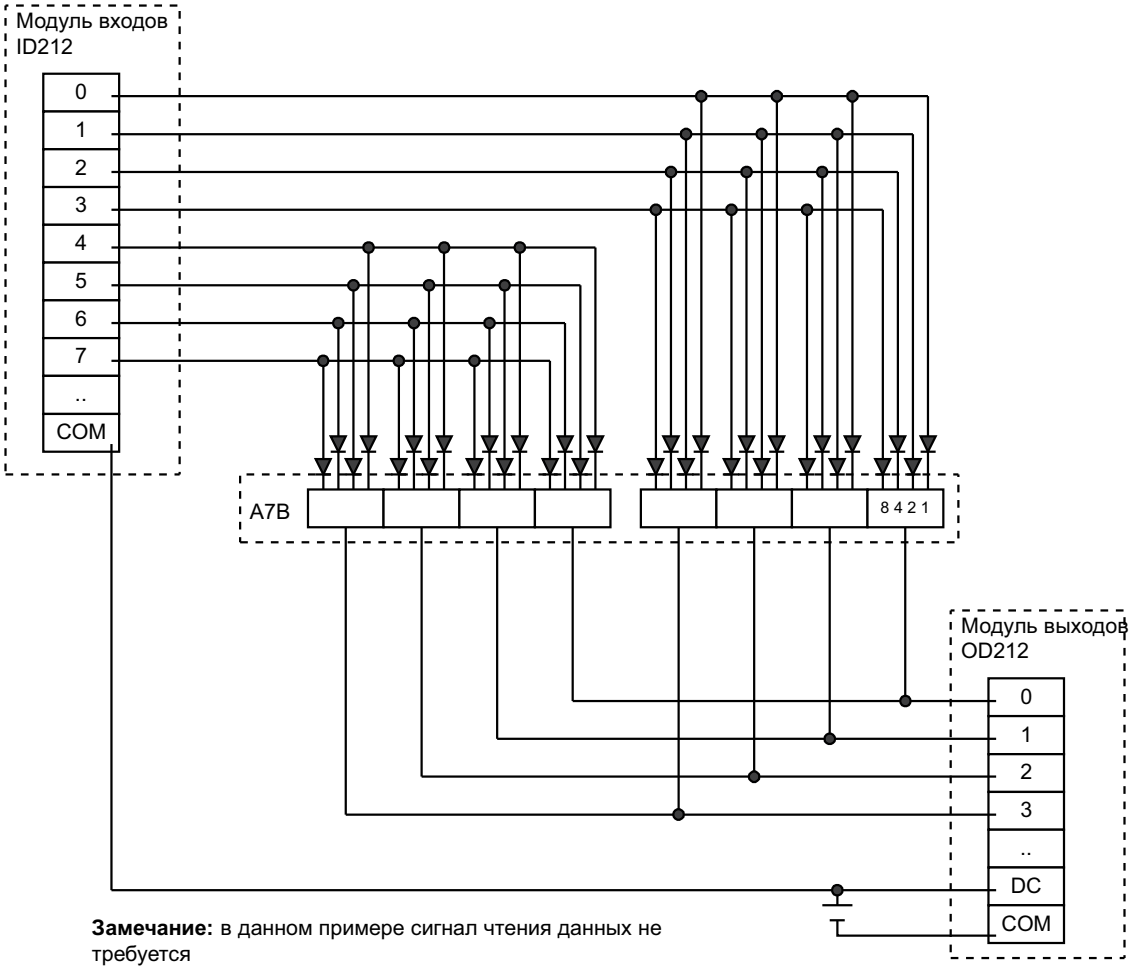
Приготовьте цифровой переключатель и блок входа и выхода как показано на следующей схеме. В случае ввода 4 цифр подключите D0..D3 с цифрового переключателя к входам 0..3.

В любом случае выход 5 будет включен в 1 после чтения одного полного цикла, но подключать выход 5 не обязательно, если не требуется для конкретной задачи.



Замечание: для подключения цифрового переключателя A7E требуется интерфейс для преобразования сигналов с 5 на 24V

На схеме показано подключение Кодирующего колеса (Цифрового переключателя) A7B.

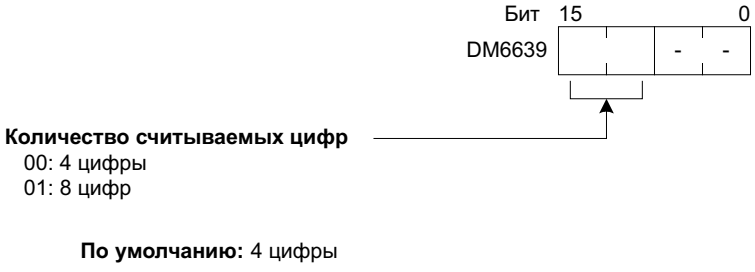


Можно пользоваться входами как на ЦУ, так и входами с блоков на 8 и более входов постоянного тока, выходы можно подключать к блокам транзисторных выходов на 8 выходов и более.

Подготовка

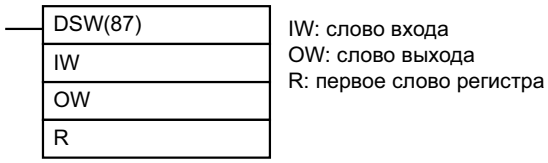
При использовании DSW(87) перед исполнением программы нужно произвести следующие настройки.

Параметры Цифрового переключателя

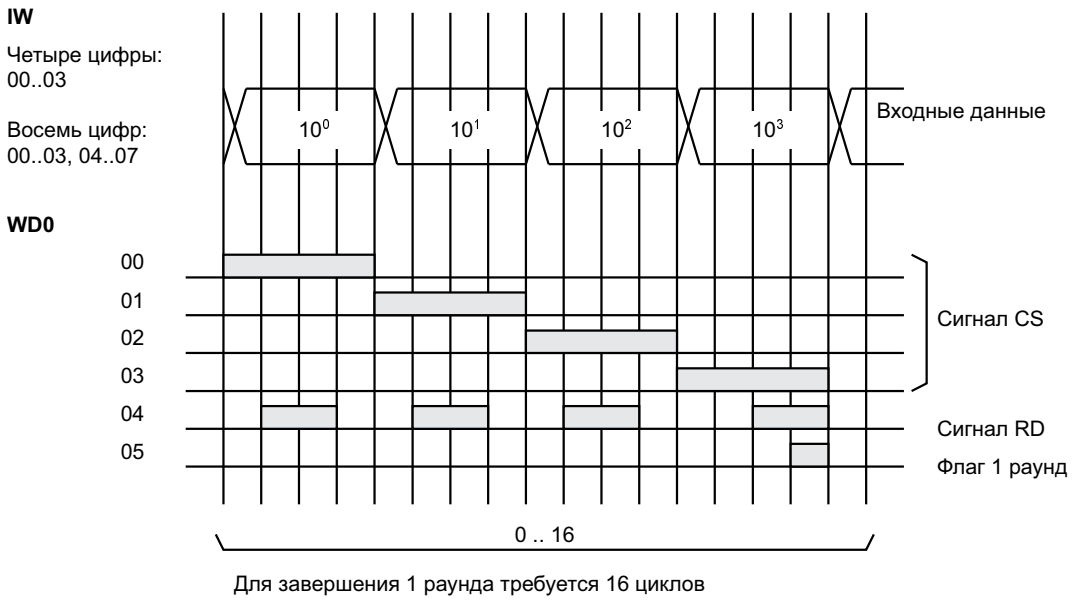


Не изменяйте биты 0..7, они не относятся к DSW(87).

Применение команды



Если входное слово для подключения цифрового переключателя определено как IW, слово выхода как OW, при выполнении программы работа будет происходить следующим образом:



Для завершения 1 раунда требуется 16 циклов

SR 25410 будет включено в 1 при выполнении DSW(87).

Замечание 1. Не используйте DSW(87) в программе больше чем 1 раз.

2. При использовании DSW(87) задавайте константу входа для соответствующего входного слова менее одного цикла. (Константы входа можно изменять начиная со слова DM 6620 и далее). Характеристики цифрового переключателя должны быть также принять во внимание при разработке системы и программы.

3. Биты входа и выхода, неиспользованные здесь, можно использовать как обычные входные и выходные биты.

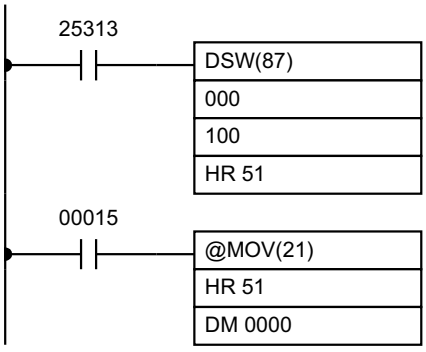
Данная команда читает 4 или 8 цифр за 16 циклов.

Пример применения

В данном примере показана программа для чтения 4 цифр в двоично-десятичном виде с цифрового переключателя. Предполагается, что цифровой переключатель подключен к IR 000 (входы) и IR 100 (выходы), все установочные параметры ПК взяты по умолчанию (4 цифры для чтения).

Данные, взятые с цифрового переключателя командой DSW(87), сохраняются в HR 51.

Когда IR 00015 = 1, значения из HR 51 передаются в DM 0001.



Замечание Выход 5 (IR 10005) включается в 1, когда считан один раунд данных и может использоваться для переключения по времени области хранения данных и

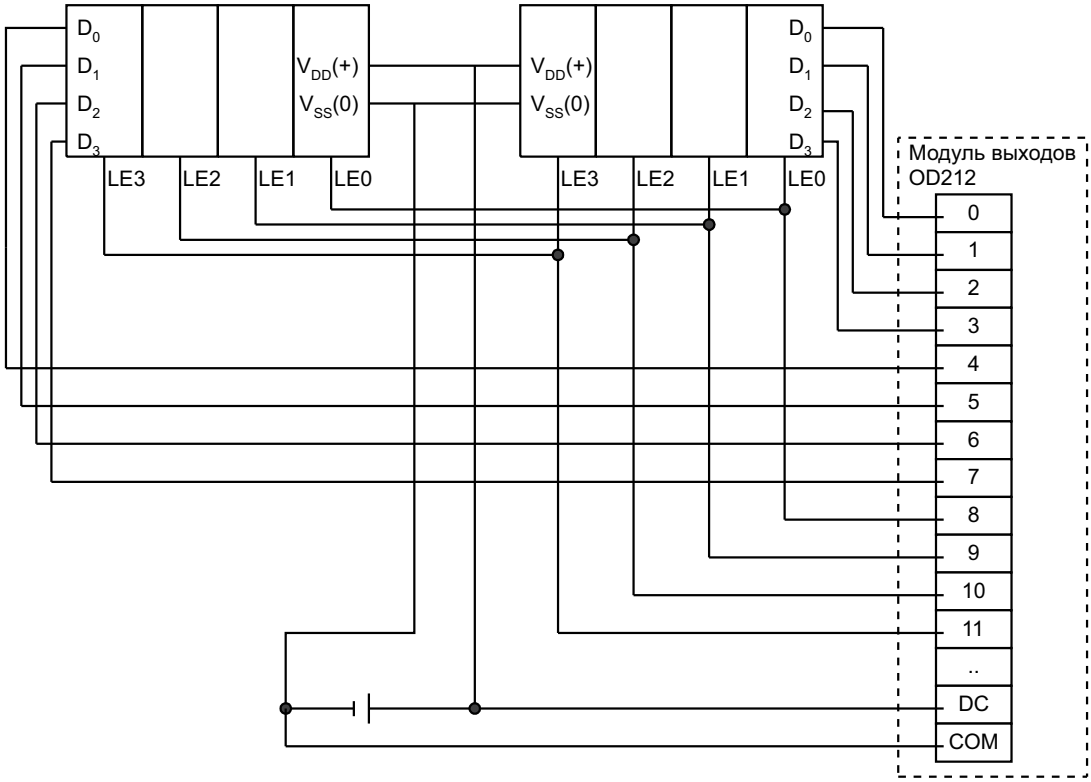
сигнала CS, когда DSW(87) используется для ввода данных в различные области памяти.

2.2.4 Вывод на 7-сегментный индикатор - 7SEG(88)

Данная команда выводит слово данных на 7-сегментный индикатор. Она использует либо 8 (для 4 цифр) либо 12 (для 8 цифр) выходных бит.

Аппаратная часть

7-сегментный индикатор подключается к Блоку выходов в соответствии со схемой. Для индикациии 4 цифр, выходы данных (D0..D3) подключаются к выходам 0..3, выходы регистра (CS0..CS3) подключаются к выходам 4..7. Выход 12 (для индикации 8 цифр) или 8 (для индикации 4 цифр) будет включен в 1, когда индикатируется 1 раунд данных, но подключать их не обязательно, если не требуется решаемой задачей.



Выходы можно подключать к блокам транзисторных выходов на 8 выходов и более для вывода 4 цифр и 16 выходов и более для вывода 8 цифр.

Замечание 1. Блок 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.

Применение команды

7SEG(88)
S
O
C

S: первое слово источника
O: выходное слово
C: слово управления

Если первое слово, содержащее данные для индикации, определено как S, Слово выхода как O, и константа из следующей таблицы как C, при выполнении программы работа при исполнении программы будет происходить следующим образом:

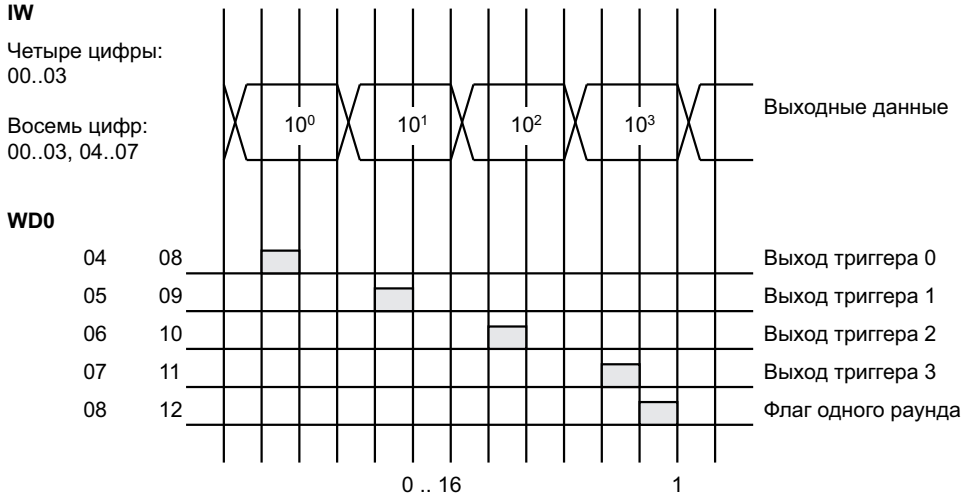
Формат хранения данных

Если выводится только 4 цифры, используется только слово S.

Задание C в зависимости от типа логики и количества выводимых цифр

Число выводимых цифр	Логика входных данных блока индикатора и логика блока выхода	Логика регистра индикатора и логика блока выхода	Значение параметра C
4 цифры (4 цифры, 1 блок)	Одинаковы	Одинаковы	000
		Различны	001
	Различны	Одинаковы	002
		Различны	003
8 цифры (4 цифры, 2 блока)	Одинаковы	Одинаковы	004
		Различны	005
	Различны	Одинаковы	006
		Различны	007

Замечание Не задавайте значения C кроме 000..007.



Для завершения 1 раунда требуется 12 циклов

SR 25409 включится в 1 при выполнении 7SEG(88).

Замечание 1. Не используйте 7SEG(88) в одной и той же программе больше чем 1 раз.

2. Принимайте во внимание время цикла и характеристики 7-сегментного индикатора при разработке системы.

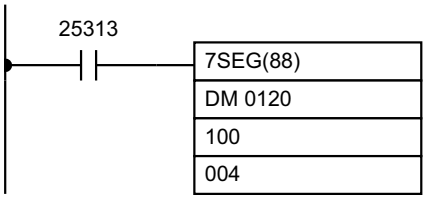
3. Выходные биты, не использованные здесь, можно использовать как обычные выходные биты.

Данной командой 4 или 8 цифр индикатируются за 12 циклов.

Операция будет производиться с первого исполнения, независимо от состояния перед исполнением.

Пример применения

В данном примере показана программа для индикации 8-разрядных двоично-десятичных цифр на 7-сегментном светодиодном индикаторе. Предполагается, что 7-сегментный индикатор подключен к выходному слову IR 100. Также предполагается, что Выходной блок использует негативную логику, и что 7-сегментный индикатор также с негативной логикой для данных и для сигналов триггера.



8-разрядные двоично-десятичные числа DM 0120 (4 правые цифры) и DM 0121 (4 левые цифры) постоянно индикатируются командой 7SEG(88). При изменении DM 0120 и DM 0121 индикация также изменяется.

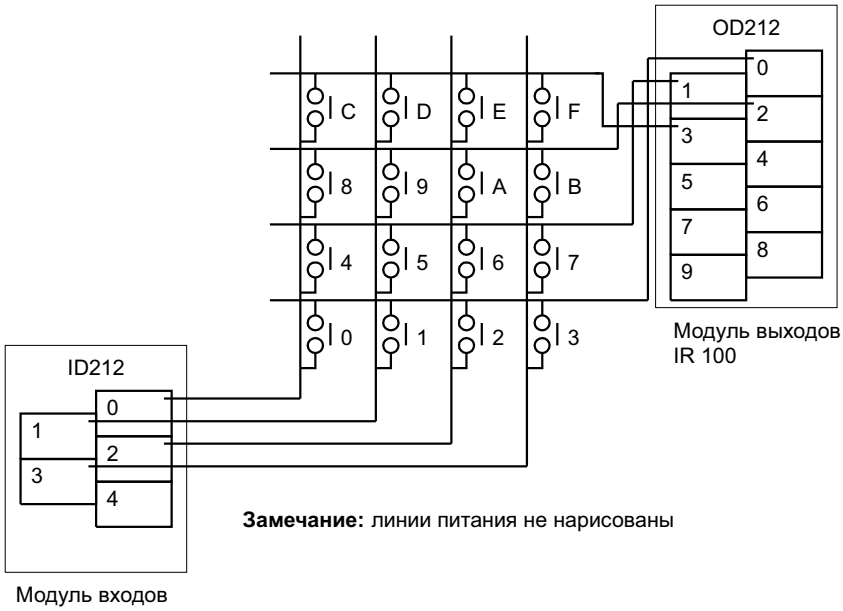
2.2.5 Переназначение битов входов/выходов

Хотя новые команды ввода/вывода обычно используют биты входов/выходов начиная с бита 00 указанных слов, они могут программироваться с помощью промежуточных слов для использования других битов входов/выходов. В следующем примере показано, как это можно сделать для команды HKY(-).

Пример

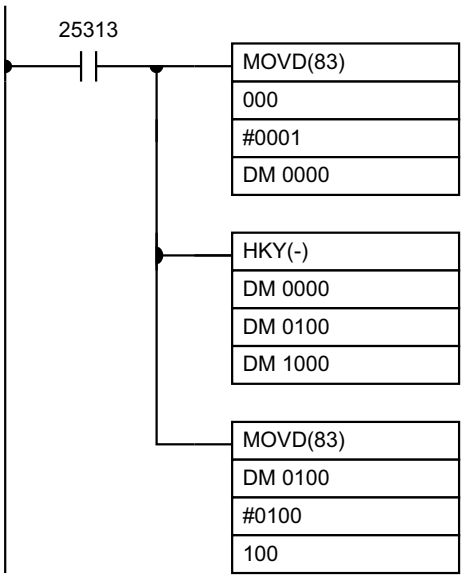
Следующие примеры схемы и программы показывают, как использовать входные биты IR 00004..IR 00007 и выходные IR 10004..IR 10007 для ввода значений с клавиатуры-16.

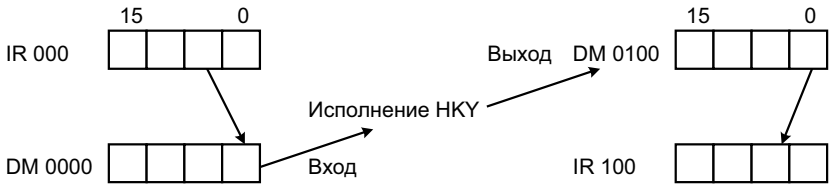
Схема подключения (Фрагмент)



Замечание: линии питания не нарисованы

Пример программы



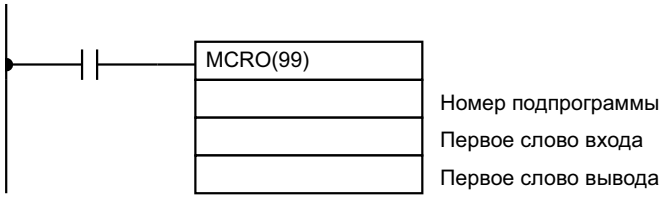


2.3 Функция MACRO

Функция MACRO позволяет использовать одну подпрограмму (шаблон программирования) простым изменением слов входов/выходов. Можно управлять несколькими секциями программы одной подпрограммой, тем самым значительно сокращая число шагов программы и упростив ее понимание.

Применение MACRO

Для использования MACRO вызовите подпрограмму, как показано на рисунке, командой MCRO(99), вместо SBS(91).



При исполнении MCRO(99) работа происходит следующим образом:

1, 2, 3,... 1. Содержимое четырех последовательных слов, начиная с первого слова входа будут переданы в IR 096..IR 099 (для CPM1 - SR 232..SR 235). Содержимое четырех последовательных слов, начиная с первого слова выхода, будут переданы в IR 196..IR 199 (для CPM1 - SR 236..SR 239).

2. Указанная подпрограмма будет выполняться до выполнения команды RET(93). (Возврат из прерывания).

3. Содержимое IR 196..IR 199 (для CPM1 - SR 236..SR 239) будут переданы в четыре последовательных слова, начинающиеся с первого слова выхода.

4. MCRO(99) окончится.

При выполнении MCRO(99) один и тот же формат команды можно использовать при необходимости, заменяя первое слово входа и первое слово выхода.

При использовании функции MACRO существуют следующие ограничения:

- Для каждого выполнения макрокоманды можно использовать только 4 последовательных слова, начинающиеся с первого входного слова (для входов) и 4 последовательных слова, начинающиеся с первого выходного слова (для выходов).
- Заданные входы и выходы должны соответствовать словам, используемым в подпрограмме.
- Даже когда для выходов применяется метод прямой выдачи, результаты подпрограммы будут фактически выданы на заданные выходы только после завершения подпрограммы (шаг 3).

Замечание 1. В CQM1 IR 096..IR 099 и IR 196..IR 199 можно использовать как рабочие биты, когда MCRO (99) не используется.

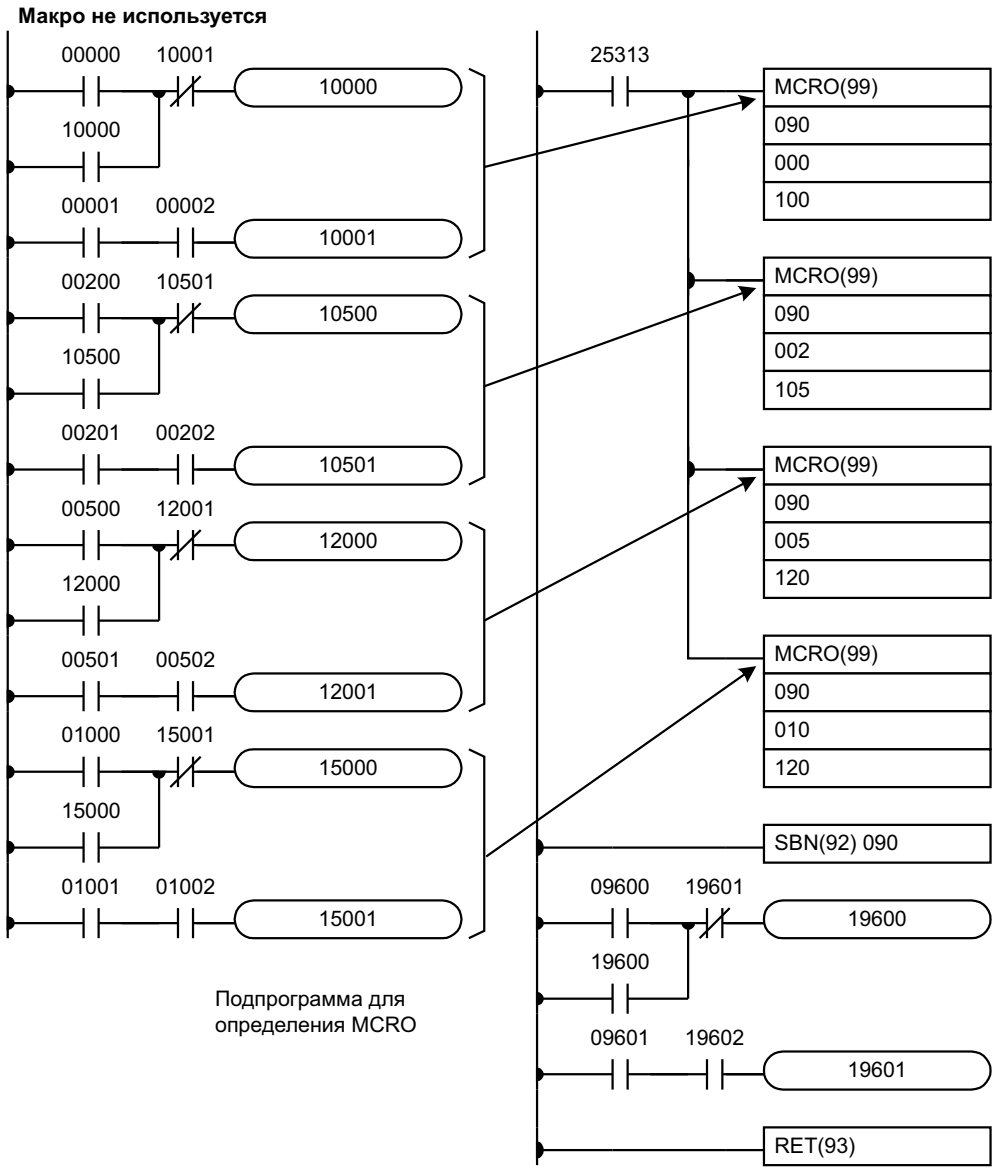
2. В CPM1 SR 232..SR 239 можно использовать как рабочие биты, когда MCRO (99) не используется.

В качестве первого входного слова и первого выходного слова можно задавать не только биты входов/выходов, но также и другие биты (такие как HR, рабочие биты и т. д.) или слова DM.

Подпрограммы, вызываемые MCRO(99), задаются как обыкновенные подпрограммы (командами SBN(92), RET(93)).

Пример применения в CQM1

При использовании MACRO программу можно упростить следующим образом.



Пример применения в CPM1/CPM1A/SRM1

В CPM1/CPM1A/SRM1 программа может быть упрощена таким же образом, но вместо IR 096..IR 099 используются слова SR 232..SR 235, а вместо IR 196..IR 199 используются слова SR 236..SR 239.

2.4 Просмотр фронтов (изменения состояния)

CQM1/CPM1/CPM1A/SRM1 поддерживают просмотр фронтов с программатора или с SSS. Оператор может обнаружить переход заданного бита 0/1 или 1/0. Когда происходит данный переход, это отображается на дисплее и звучит зуммер.

Подробности процедуры просмотра фронтов с программатора см. Инструкцию по работе CQM1, Инструкцию по работе CPM1, Инструкцию по работе CPM1A, Инструкцию по работе SRM1или процедуру с SSS см. Инструкцию по работе с SSS: ПК серии C.

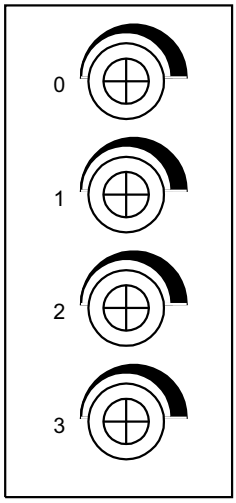
2.5 Аналоговые задания (только CQM1-CPU42-E/CPM1/CPM1A)

В CQM1-CPU42-EV1 и CPM1/CPM1A функция аналоговых заданий автоматически передает значения с аналоговых регуляторов в слова IR 220..IR 223. Данная функция очень полезна, когда во время работы требуется настройка заданий (для таймеров и счетчиков) во время работы. Эта задания легко изменять, вращая регуляторы на ЦУ.

Значения хранятся в двоично-десятичном виде в диапазоне 0000..0200. Для настройки используйте мини-отвертку. (Для увеличение значений поворачивайте регулятор по часовой стрелке.

Задания CQM1-CPU42-EV1

В CQM1-CPU42-EV1 имеется 4 аналоговых регулятора На следующем рисунке показаны регуляторы CQM1-CPU42-EV1 и соответствующие слова IR , в которых содержится их значения.



Значение регулятора 0 (IR 220)

Значение регулятора 1 (IR 221)

Значение регулятора 2 (IR 222)

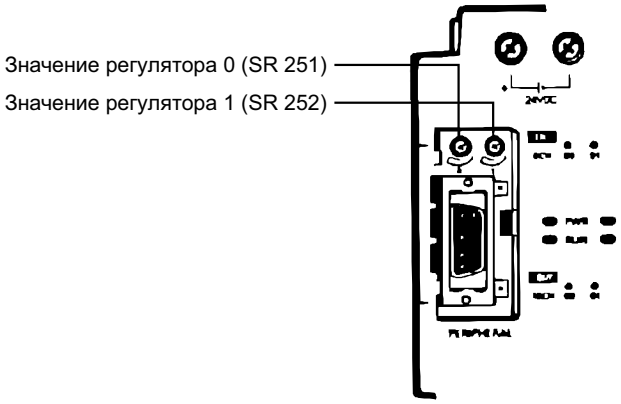
Значение регулятора 3 (IR 223)

В остальных моделях CQM1 (кроме CQM1-CPU42-EV1) IR 220..IR 223 не используются для специальных целей. Их можно использовать, как обычные слова в программе.

Внимание! CQM1-CPU42-EV1 постоянно обновляет слова IR 220..IR 223 значениями с аналоговых регуляторов, пока включено питание. Не производите в эти слова запись из программы или периферийного устройства.

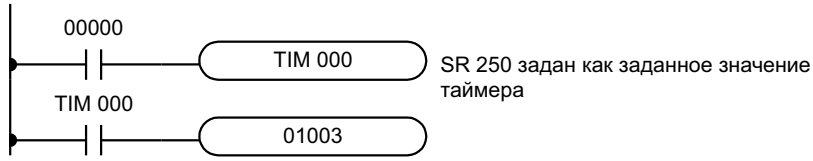
Заданния CPM1/CPM1A

В CPM1 имеется 2 аналоговых регулятора На рисунке показаны регуляторы CPM1 и соответствующие слова SR, в которые заносятся их значения. Используйте отвертку с головкой PHILIPS для настройки задания.



Внимание! Аналоговые регуляторы могут менять свои значения с изменением температуры. Не используйте регуляторы там, где требуются высокая точность.

Пример программы CPM1.



На следующей релейно-контактной схеме применяются аналоговые регуляторы CPM1/CPM1A. Значение слова SR 250 (двоично-десятичные 0000..0200) определяется настройкой регулятора 0. Данное значение используется для настройки задания таймера от 0.0 до 20.0 с.

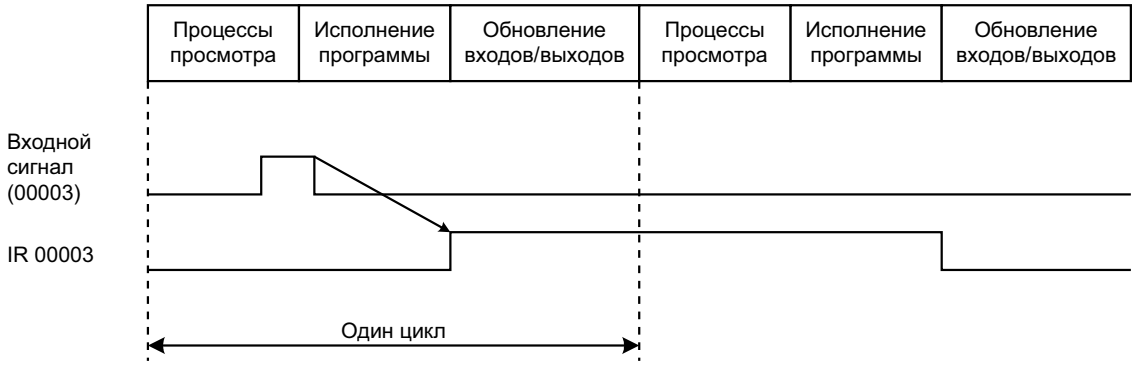
2.6 Входы быстрой реакции (только CPM1/CPM1A)

Все CPM1/CPM1A имееют входы быстрой реакции, которыеможно использовать для ввода коротких сигналов.

Все модули на 10 точек имееют 2 входа быстрой реакции, а модули на 20, 30 и 40 точек - 4 входа быстрой реакции. (Одни и те же входы используются в качестве входов быстрой реакции и входов прерываний).

Работа быстрой реакции

Входы быстрой реакции имеют внутренний буфер, так что можно обнаружить сигналы длительностью менее 1 цикла. Можно обнаружить импульсы шириной до 0.2 мс, независимо от времени их прихода в цикле.



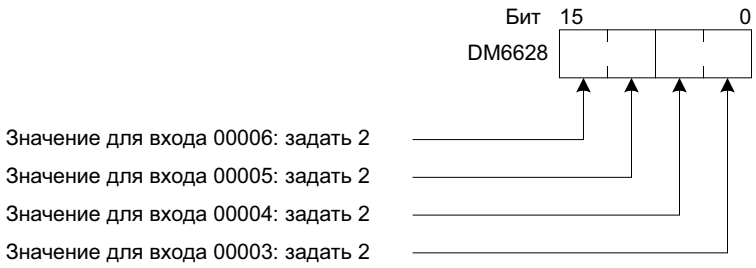
Модель ПК	Входные биты	Минимальная ширина импульса
10-ти точечные модули	IR 00003 - IR 00004	0.2 мс
20, 30 и 40 точечные модули	IR 00003 - IR 00006	

Задание параметров

Настройка входов коротких импульсов

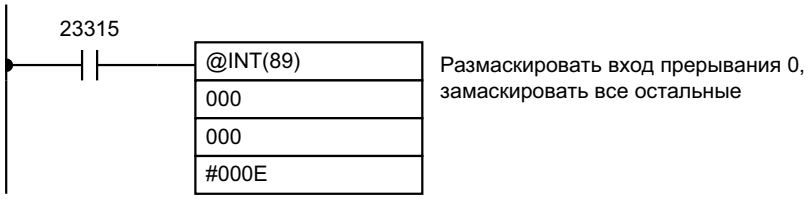
Входы 00003..00006 (00003 и 00004 для CPM1-10CDR-) можно установить в режим коротких импульсов в DM 6628 как показано в следующей таблице.

Слово	Значение
DM 6628	0: нормальный вход 1: вход прерывания 2: вход быстрой реакции



Пример программы

Входы для DM 6628 установлены в 0002.



3. Области памяти

В данной главе описана структура областей памяти ПК и их использование. Также описаны операции с кассетой памяти CQM1, которые используются для обмена данных между ЦУ CQM1 и кассетой памяти.

3.1 Функции области памяти CQM1

3.1.1 Структура областей памяти

В CQM1 можно использовать следующие области памяти.

Область данных		Размер	Слова	Биты	Функции
Область IR[1]	Область входов	128 или 192 бит	IR 000 - IR 011	IR 00000 - IR 01115	CQM1-CPU11/21-E: до 8 слов (128 бит) могут использоваться как биты входа/выхода CQM1-CPU4-E: до 12 слов (192 бит) могут использоваться как биты входа/выхода
	Область выходов		IR 100 - IR 111	IR 10000 - IR 11115	
	Рабочие области	Минимум 720 бит[2]	IR 012 - IR 095	IR 01200 - IR 09515	Рабочие биты не имеют определенного назначения и могут произвольно использоваться внутри программы
			IR 112 - IR 195	IR 11200- IR 19515	
			IR 216 - IR 219	IR 21600 - IR 21915	
			IR 224 - IR 229	IR 22400 - IR 22915	
Область операнда MACRO	Область входов	64 бита	IR 096 - IR 099	IR 09600 - IR 09915	Применяется при использовании команды MACRO (MCRO(99)). Когда команда MACRO не используется, данные биты можно использовать в качестве рабочих бит.
	Область выходов	64 бита	IR 196 - IR 199	IR 19600 - IR 19915	
Область аналоговых заданных значений[1]		64 бита	IR 220 - IR 223	IR 22000 - IR 22315	CQM1-CPU42-E: Служит для введения задания с аналоговых регуляторов (нельзя использовать в качестве рабочих бит). В других ЦУ можно использовать в качестве рабочих бит.
Текущее значение высокоскоростного счетчика 0[1]		32 бита	IR 230 - IR 231	IR 23000 - IR 23115	Служат для сохранения текущего значения высокоскоростного счетчика 0
Текущие значения импульсных выходов портов 1 и 2[1]		64 бита	IR 236 - IR 239	IR 23600 - IR 23915	CQM1-CPU43-E: Служат для сохранения текущих значений импульсных выходов портов 1 и 2 (Нельзя использовать в качестве рабочих бит) CQM1-CPU44-E: Используются системой. (Нельзя использовать в качестве рабочих бит) В других ЦУ можно использовать в качестве рабочих бит.
Текущее значение высокоскоростных счетчиков 1 и 2 (прим. 1)		64 бита	IR 232 - IR 235	IR 23200 - IR 23515	CQM1-CPU43/44-E: Служат для сохранения текущего значения высокоскоростных счетчиков 1 и 2 портов 1 и 2 (Нельзя использовать в качестве рабочих бит) В других ЦУ можно использовать в качестве рабочих бит.
Области для дополнительных команд[1]		320 бит	IR 200 - IR 215 IR 240 - IR 243	IR 20000 - IR 21515 IR 24000 - IR 24315	Служат для использования в дополнительных командах.
Область SR		184 бита	SR 244 - SR 255	SR 24400 - SR 25515	Выполняют специальные функции, такие, как флаги и биты контроля. Можно использовать в качестве рабочих бит.
Область TR		8 бит	-	TR 0 - TR 7	Данные биты служат для временного хранения состояния 1 или 0 в ответвлениях программы.

Область данных		Размер	Слова	Биты	Функции
Область HR		1600 бит	HR 00 - HR 99	HR 0000 - HR 9915	Данные биты сохраняют данные и состояние 1 или 0 при отключении питания.
Область AR		448 бит	AR 00 - AR 27	AR 0000 - AR 2715	Выполняют специальные функции, такие, как флаги и биты контроля.
Область LR[1]		1024 бита	LR 00 - LR 63	LR 0000 - LR 6315	Служат для связи 1+1 через порт RS-232C.
Область таймеров/счетчиков[3]		512 бит	TC 000 - TC 511 (номера таймеров/ счетчиков)		Одни и те же номера используются и для таймеров и для счетчиков. TC 000 - TC 002 служат для интервальных таймеров
Область DM	Чтение/ запись	1024 слов	DM 0000-DM 1023	-	Данные области DM доступны только в виде слов. Значения слов сохраняются при отключении питания
		5120 слов	DM 1024-DM 6143	-	Есть только у ЦУ CQM1-CPU4_-E
	Только чтение[5]	425 слов	DM 6144-DM 6568	-	Не могут быть изменены из программы.
	Область записи протоко-ла ошибок[5]	31 слово	DM 6569-DM 6599	-	Служат для сохранения времени и кода появления признака ошибки
	Установочные парамет ры ПК[5]	56 слов	DM 6600-DM 6655	-	Служат для сохранения параметров, управляющих работой ПК.
Область программы пользователя (область UM)		3200 или7200 слов	-		Служат для сохранения программы Сохраняется при отключении питания. CQM1-CPU11/21-E: 3 200 слов CQM1-CPU4-E: 7 200 слов

Замечание 1. Биты IR и LR, которые не используются для выделенных им функций, можно использовать как рабочие биты.

2. Как минимум 2 720 бит можно использовать в качестве рабочих бит. Общее количество бит, которые можно использовать в качестве рабочих, зависит от конфигурации системы ПК.

3. При доступе к текущим значениям номера таймеров/ счетчиков используются как слова; при доступе к флагам завершения они используются как биты.

4. Хотя CQM1-CPU11-E и CQM1-CPU21-E не поддерживают DM 1024..DM 6143, при адресации к данным областям признака ошибки не появляется. Любая попытка записи в данные слова не приведет в результату и любое чтение даст нуль.

5. Значения DM 6144..DM 6655 нельзя переписать из программы

3.1.2 Область IR

Далее описаны функции области IR

Область входов и выходов

Биты области входов/выходов приписаны к разъемам блоков входа и выхода. Они отражают состояние 1 или 0 входных и выходных сигналов. Биты входов начинаются с IR 00000, биты выходов - с IR 10000. У CQM1 в качестве входных битов можно использовать только IR 0000..IR 01515 и в качестве выходных битов можно использовать только IR 1000..IR 11515.

Информацию о распределении битов входа и выхода см. далее.

Замечание Биты входа нельзя использовать для команд выхода. Не используйте один и тот же выходной бит в более чем одной команде OUT и/или OUT NOT, в противном случае программа не будет выполняться должным образом.

Рабочие области

У CQM1-CPU11/21/41-E любой бит в области IR 001..IR 243, не используемый для специальных функций, можно использовать в качестве рабочих битов. Для CQM1-CPU42/43/44- E есть исключения, приведенные в таблице.

ЦПУ	Биты, которые нельзя использовать в качестве рабочих
CQM1-CPU42- E	IR 22000..IR 22315
CQM1-CPU43/44- E	IR 23200..IR 23915

Рабочие биты можно использовать в программе произвольно. Их можно использовать только внутри программы, а не в качестве прямых внешних входов/выходов. Рабочие биты сбрасываются (т.е. устанавливаются в 0) при отключении питания или пуске и останове программы.

Биты, приведенные в таблице, имеют специальные функции, но могут использоваться в качестве рабочих, если их специальные функции не используются.

Диапазон	Функции
IR 001..IR 015	Когда биты присвоены блокам входа, они служат как входные биты.
IR 096..IR 099	При применении команды MACRO данные биты служат в качестве битов операндов входов.
IR 100..IR 115	Когда биты присвоены блокам выхода, они служат как выходные биты.
IR 196..IR 199	При применении команды MACRO данные биты служат в качестве битов операндов выходов.
IR 220..IR 223	В CQM1-CPU42-E служат для хранения задания с аналоговых регуляторов. В других ЦУ можно использовать в качестве рабочих бит.
IR 230..IR 231	Когда применяется высокоскоростной счетчик 0, данные биты служат для сохранения текущего значения.
IR 232..IR 235	В CQM1-CPU43/44-E данные биты служат для хранения текущего значения высокоскоростных счетчиков 1 и 2 портов 1 и 2. В других ЦУ можно использовать в качестве рабочих бит.
IR 236..IR 239	В CQM1-CPU43-E данные биты служат для хранения текущих значений импульсных выходов портов 1 и 2. В CQM1-CPU44-E они используются системой. В других ЦУ можно использовать в качестве рабочих бит.

IR 200..IR 215 и IR 240..IR 243 будут иметь специальное назначение в будущих функциях, но теперь их можно использовать в качестве рабочих битов.

LR 00..LR 63 служат в качестве битов связи, но если ПК не соединен с другим CQM1, их можно использовать в качестве рабочих битов.

Распределение битов входов/выходов

Слова входов/выходов располагаются по порядку слева направо, начиная с IR 001 для блоков входа и IR 100 для блоков выхода. Входы ЦУ присвоены битам IR 000. Даже если блоки входов и выходов расположены не по порядку, слова входов и выходов располагаются в отдельных частях области IR .

Одно слово выделяется даже для блока на 8 входов/выходов. Использование бит на блоке 8 входов/выходов показано в таблице.

Блок	Биты 0..7	Биты 8..15
Блок входа	Биты входа	Всегда 0
Блок выхода	Биты выхода	Рабочие биты

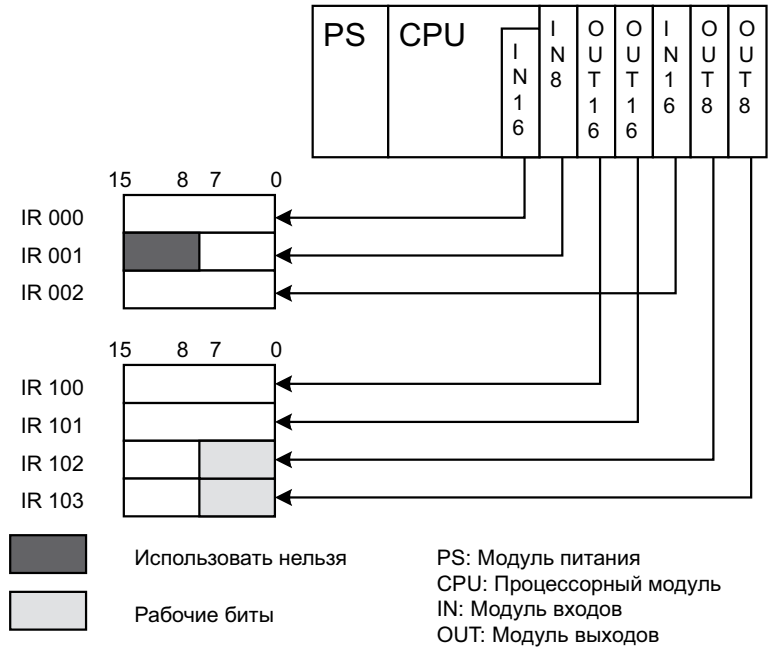
Число отведенных бит входов/выходов зависит от используемого ЦУ CQM1, как показано в таблице:

Модель ЦУ	Число бит входов/выходов
CQM1-CPU11/21-E	В качестве бит входов/выходов можно использовать до 128 бит (8 слов)
CQM1-CPU4-E	В качестве бит входов/выходов можно использовать до 192 бит (12 слов)

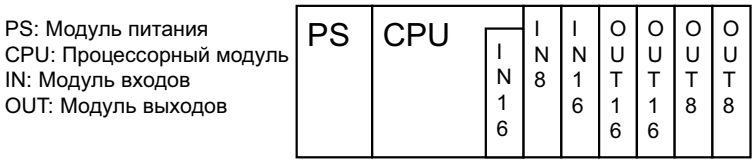
Для CQM1 можно выделить 128 бит (8 слов) входов/выходов. Блоку на 16 входов/выходов выделяется два байта и при подсчете общего числа для блока нужно считать 16.

Пример распределения слов

Все биты в словах после последнего присвоенного слова входа и выхода можно использовать в качестве рабочих бит.



Чтобы распределение слов было более понятным и для уменьшения проблем с помехами рекомендуется все блоки входа располагать сразу после ЦУ. Для данного примера конфигурация в таком случае будет иметь следующий вид:



Число выделенных слов входов хранится в двоично-десятичном виде в AR 2200..AR 2207, число занятых слов выходов хранится в двоично-десятичном виде в AR 2208..AR 2215. CQM1 не использует таблицу входов/выходов.

3.1.3 Область SR

Данные биты главным образом служат в качестве флагов, относящимся к работе CQM1. Подробности см. также соответствующие пункты Приложение C, области памяти.

SR 244..SR 247 могут также использоваться в качестве рабочих бит, когда входные прерывания не используются в режиме счетчика.

3.1.4 Область TR

В случае, когда сложную релейно-контактную схему нельзя запрограммировать в мнемокоде, эти биты служат для временного хранения условий исполнения 1 или 0 ветвей программы. Они используются только для мнемокода. При программировани непосредственно в виде релейно-контактной схемы с помощью SSS, биты TR

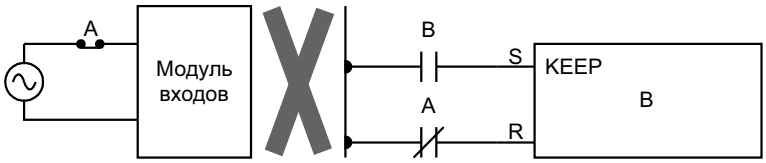
обрабатываются автоматически. Одни и те же биты нельзя использовать более одного раза внутри одного блока команд, но могут использоваться повторно в разных блоках команд. Состояние 1 или 0 битов TR нельзя просмотреть с периферийного устройства.

Пример программирования с использованием TR приведен в п. 4-3-8.

3.1.5 Область HR

Данные биты сохраняют свое состояние 1/ 0 даже при отключении питания CQM1 или при пуске или останове программы. Они используются таким же образом, как и рабочие биты.

Внимание! Никогда не используйте входной бит нормально открытого условия для входа сброса (R) для команды KEEP(11), когда входное устройство использует питание переменного тока (см. схему ниже). Задержка при отключении питания постоянного тока ПК по сравнению с питанием переменного тока входного устройства может вызвать сброс указанного бита KEEP(11).



3.1.6 Область AR

Данные биты главным образом служат в качестве флагов, относящимся к работе CQM1. Подробности о функциях битов см. соответствующие пункты в данной главе или Приложение C, области памяти.

За исключением AR 23 (Счетчик отключения питания), состояние слов и бит AR обновляется каждый цикл. (AR 23 обновляется только прерыванием питания).

3.1.7 Область LR

Когда CQM1 подключен к другому CQM1 методом 1:1, данные биты служат для образования общего поля данных. Подробности см. 1-6-6.

Если данные биты не используются в качестве битов связи, их можно использовать в качестве рабочих битов.

3.1.8 Область таймеров/ счетчиков

Данная область используется для управления таймерами и счетчиками, созданными командами TIM, TIMH(15), CNT и CNTR(12). Одни и те же номера используются и для таймеров и для счетчиков и каждый номер можно использовать только раз в программе пользователя. Не используйте одинаковые номера таймеров/ счетчиков дважды даже для разных команд.

Номера TC служат для создания таймеров и счетчиков, а также для доступа к флагам завершения и текущему значению. Если номер TC задан для данных типа слово, он будет обращаться к текущим значениям; если номер задан для данных типа бит, он будет обращаться к флагам завершения таймера/счетчика.

Флаг завершения включается в 1, когда текущее значение используемого таймера/счетчика достигает 0.

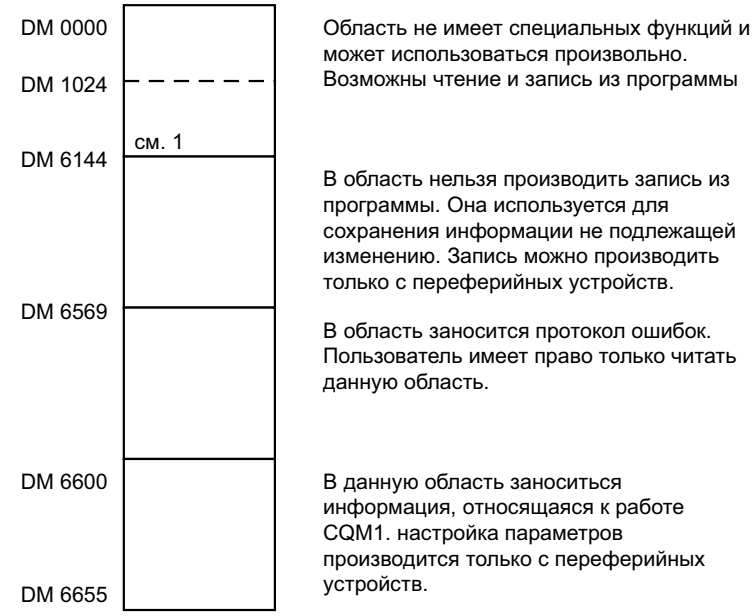
Подробности о таймерах и счетчиках см. данную инструкцию, п. 5-15.

Замечание 1. TC 000..TC 015 и обработка с прерываниями следует использовать для TIMH(15), если время цикла больше 10 мс. Использование других номеров таймеров/счетчиков или неиспользование прерываний приведет к неточности работы высокоскоростных таймеров. Обработка прерываний задается в установочном параметре DM 6629.

2. Когда входное условие для TIM или TIMH(15) включается в 0, текущее значение сбрасывается и становится равным заданию. Текущее значение сброшено также в начале исполнения программы или когда условие INTERLOCK включается в 0 в секции программы INTERLOCK (IL-ILC). Текущее значение для CNT или CNTR(12) не сбрасывается как у таймера, но сбрасывается только при включении входе сброса в состояние 1.

3.1.9 Область DM

Данные доступны в виде слов. Как показано ниже, в области DM есть область, которая может использоваться произвольно и области со специальными функциями.



Замечание 1. CQM1-CPU11-E и CQM1-CPU21-E не поддерживают DM 1024..DM 6143.

2. Включение переключателя DIP, секция 1 на ЦУ в положение ON предотвратит запись даже с периферийных устройств.

Содержание фиксированных DM, установочные параметры, программу пользователя и таблицу команд можно сохранять и списывать с кассеты памяти как единый блок. Подробности см. 3-3-1.

Замечание Хотя CQM1-CPU11-E и CQM1-CPU21-E не поддерживают DM 1024..DM 6143, при адресации к ним признака ошибки не возникает. Любая попытка записи в данные слова не приведет к результату, а попытка чтения даст все нули.

3.1.10 Область UM

В области UM хранится программа пользователя. Содержимое области UM можно прочесть и записать только как данные программы, а не слова. В следующей таблице показан размер области UM для разных ЦУ.

Модель ЦУ	**Размер области UM**
CQM1-CPU11/21-E	3.2 Кслов
CQM1-CPU4-E	7.2 Кслов

3.2 Функции областей памяти CPM1/CPM1A

3.2.1 Структура области памяти

В CPM1/CPM1A можно использовать следующие области памяти.

Область данных		Размер	Слова	Биты	Функции
Область IR[1]	Область входов	160 бит	IR 000 - IR 009	IR 00000 - IR 00915	Данные биты могут быть выделены внешним входам/выходам.
	Область выходов	160 бит	IR 010 - IR 019	IR 01000 - IR 01915	
	Рабочая область	512 бит	IR 200 - IR 231	IR 20000 - IR 23115	Рабочие биты могут произвольно использоваться внутри программы
Область SR		384 бита	SR 232 - SR 255	SR 23200 - SR 25515	Выполняют специальные функции,такие, как флаги и биты контроля.
Область TR		-	-	TR 0 - TR 7	Данные биты служат для временного хранения состояния 1 или 0 при ветвлениях программы.
Область HR[2]		320 бит	HR 00 - HR 19	HR 0000 - HR 1915	Данные биты сохраняют данные и сохраняют состояние 1 или 0 при отключении питания
Область AR[2]		256 бит	AR 00 - AR 15	AR 0000 - AR 1515	Выполняют специальные функции, такие, как флаги и биты контроля
Область LR[1]		256 бит	LR 00 - LR 15	LR 0000 - LR 1515	Служат для связи 1+1 с другим ПК
Область таймеров/счетчиков[2]		TC 000 - TC 127 (номера таймеров/счетчиков)[3]			Одни и те же номера используются и для таймеров и для счетчиков
Область DM	Чтение/ запись	1002 слова	DM 0000 - DM 0999 DM 1022 - DM 1023		Данные области DM доступны только в виде слов. Значения слов сохраняются при отключении питания.
	Область протокола ошибок (прим. 4)	22 слова	DM 1000 - DM 1021		Служат для сохранения времени и кода появления признака ошибки. Если функции протокола ошибки не используются, данные слова можно использовать как обычные слова чтения/записи.
	Только читать (прим. 4)	456 слов	DM 6144 - DM 6599		Не могут быть переписаны из программы.
	Установочные параметры ПК (прим. 4)	56 слов	DM 6600 - DM 6655		Служат для сохранения параметров, управляющих работой ПК.

Замечание 1. Биты IR и LR, которые не используются для отведенных им функций, можно использовать как рабочие биты.

2. Содержание областей HR, AR, счетчиков и DM чтения/записи поддерживается конденсатором. При 25 град. конденсатор сохраняет память 20 суток. График зависимости времени сохранения от температуры см. 2-1-2 Инструкции по работе CPM1/CPM1A.

3. При обращении к текущим значениям номера таймеров/ счетчиков используются как слова; при обращении к флагам завершения они используются как биты.

4. Значения DM 6144..DM 6655 нельзя переписать из программы, но можно изменить с периферийных устройств.

3.2.2 Область IR

Далее описаны функции области IR.

Биты входов/ выходов

Биты области IR 00000..IR 01915 выделены точкам входов/выходов на ЦУ и на Блоке входа/выхода. Они отражают состояние 1/0 входных и выходных сигналов. Биты входов начинаются с IR 00000, биты выходов - с IR 01000..

В следующей таблице приведено распределение битов IR по точкам входов/выходов ЦУ CPM1 и блока входов/выходов CPM1-20EDR.

ЦУ CPM1	Входы/выходы	Клеммы ЦПУ	Клеммы блока входов/выходов
CPM1-10CDR -	Входы	6 точек: 00000..00005	12 точек: 00100..00111
	Выходы	4 точки: 01000..01003	8 точек: 01100..01107
CPM1-20CDR -	Входы	12 точек: 00000..00011	12 точек: 00100..00111
	Выходы	8 точек: 01000..01007	8 точек: 01100..01107
CPM1-30CDR -	Входы	18 точек: 00000..00011, 00100..00105	12 точек: 00200..00211
	Выходы	12 точек: 01000..01007, 01100..01103	8 точек: 01200..01207

В следующей таблице приведено распределение битов IR по точкам входов/выходов ЦУ CPM1A и блока расширения CPM1A-20EDR.

Модель	Входы/выходы на ЦПУ		Блок расширения					
	Входы	Выходы	Входы	Выходы	Входы	Выходы	Входы	Выходы
CPM1A-10CD_-_	6 00000.. 00005	4 01000.. 01003	-	-	-	-	-	-
CPM1A-20CD_-_	12 00000.. 00011	8 01000.. 01007	-	-	-	-	-	-
CPM1A-30CD_-_	18 00000.. 00011 00100.. 00105	12 01000.. 01007 01100.. 01103	12 00200.. 00211	8 01200.. 01207	12 00300.. 00311	8 01300.. 01307	12 00400.. 00411	8 01400.. 01407
CPM1A-40CD_-_	24 00000.. 00011 00100.. 00111	16 01000.. 01007 01100.. 01107	12 00200.. 00211	8 01200.. 01207	12 00300.. 00311	8 01300.. 01307	12 00400.. 00411	8 01400.. 01407

Рабочие биты

Рабочие биты можно использовать в программе произвольно. Их можно использовать только внутри программы, но не как прямые внешние входы/выходы.

3.2.3 Область SR

Данные биты главным образом служат в качестве флагов, относящихся к операциям CPM1 или содержат текущее значение и задание для различных функций. Подробности о различных функциях битов см. соответствующие пункты в данной главе или в Приложении C, области памяти.

SR 244..SR 247 могут также использоваться в качестве рабочих бит, когда входные прерывания не используются в режиме счетчика.

3.2.4 Область TR

В случае, когда сложную релейно-контактную схему нельзя запрограммировать в мнемокоде, эти биты служат для временного хранения условий исполнения 1/0 при ветвлениях программы. Они используются только для мнемокода. Когда программирование производится непосредственно релейно-контактной схемы с помощью SSS, биты TR обрабатываются автоматически. Одни и те же биты нельзя использовать более одного раза внутри одного блока команд, но могут использоваться повторно в разных блоках команд. Состояние 1/0 битов TR нельзя наблюдать с периферийного устройства.

Примеры, показывающие использование битов TR в программировании, приведены в п. 4-3-8.

3.2.5 Область HR

Данные биты сохраняют свое состояние 1/0 даже при отключении питания CPM1 или при пуске или останове программы. Они используются таким же образом, как и рабочие биты.

3.2.6 Область AR

Данные биты главным образом служат в качестве флагов, относящимся к работе CPM1. Данные биты сохраняют свое состояние даже при отключении питания CPM1 или при пуске и останове программы. Подробности о функциях битов см. Приложение C, Области памяти.

3.2.7 Область LR

Когда CPM1 подключен в режиме 1:1 к другому CPM1, CQM1 или C200HS данные биты служат для образования общего поля данных. Подробности см. 1-6-7.

Если биты LR не используются в качестве битов связи, их можно использовать в качестве рабочих битов.

3.2.8 Область таймеров/ счетчиков (TC)

Данная область используется для управления таймеры и счетчиками, созданными командами TIM, TIMH(15), CNT и CNTR(12). Одни и те же номера используются и для таймеров и для счетчиков и каждый номер можно использовать только раз в программе пользователя. Не используйте одинаковые номера таймеров/ счетчиков дважды даже для разных команд.

Номера TC служат для создания таймеров и счетчиков, а также для доступа к флагам завершения и текущим значений. Если номер TC задан для данных типа слово, он будет работать с текущим значением; если номер задан для данных типа бит, он будет обращаться к флагам завершения таймера/счетчика.

Подробности о таймерах и счетчиках см. данную инструкцию, п. 5-15.

3.2.9 Область DM

Данные области DM доступны только в виде слов. Содержание области DM сохраняется даже при отключении питания CPM1 или при пуске и останове программы.

Слова DM 0000..DM 0999, DM 1022 и DM 1023 можно использовать в программе произвольно. Другие слова DM выполняют специальные функции, описанные далее.

Протокол ошибок

DM 1000..DM 1021 содержит информацию протокола ошибок. Подробности см. Главу 8, поиск ошибок.

Установочные параметры ПК

DM 6600..DM 6655 содержат установочные параметры ПК. Подробности см. Главу 1-1, Установочные параметры ПК.

3.3 Функции областей памяти SRM1

3.3.1 Структура областей памяти

В SRM1 можно использовать следующие области памяти.

Область данных		Размер	Слова	Биты	Функция
Область IR[1]	Область входов	160 бит	IR 000.. IR 009	IR 00000.. IR 00915	Данные биты могут быть выделены внешним входам/выходам. Состояние 1 или 0 битов входов/выходов будут аналогичны состоянию на клеммах.
	Область выходов	160 бит	IR 010.. IR 019	IR 01000.. IR 01915	Все биты, которые не используются для входов/выходов можно использовать в качестве рабочих бит.
	Рабочая область	512 бит	IR 200.. IR 239	IR 20000.. IR 23915	Рабочие биты могут произвольно использоваться внутри программы. Однако биты IR 232..IR 239 использовать в качестве области входов MACRO для команды MACRO.
Область SR		248 бит	SR 240.. SR 255	SR 24000.. SR 25515	Эти биты служат в качестве ячеек сохранения флагов и заданных/текущих значений при отработке программы SRM1. См. Область AR.
Область TR		8 бит	-	TR 0..TR 7	Когда сложную РКС нельзя записать в мнемокоде эти биты служат для временного сохранения состояния 1 / 0 в ветвях программы. Эти временные биты нельзя использовать в пределах одного блока, но если блоки разные, то можно. Состояние 1/0 данных битов нельзя просмотреть с помощью функций просмотра периферийного устройства.
Область HR [2]		320 бит	HR 00.. HR 19	HR 0000.. HR 1915	Данные биты служат для хранения данных и сохраняют состояние 1/ 0 при отключении питания или запуске и останове программы. Они используются так же, как и рабочие биты.
Область AR [2]		256 бит	AR 00.. AR 15	AR 0000.. AR 1515	Эти биты обслуживают специальные функции, такие, как флаги и биты управления. Биты AR 04..07 служат ведомыми. См. Область AR.
Область LR [1]		256 бит	LR 00.. LR 15	LR 0000.. LR 1515	Служат для связи 1:1 с другими ПК SRM1, CQM1 или C200HS.
Область таймеров/счетчиков[2]		TC 000..TC 127[3]			Таймеры и счетчики используют команды TIM, TIMH(15) CNT, CNTR(12). Одинаковые номера используются и для таймеров и для счетчиков.
Область DM	Читать/ писать[2]	2000 слов	DM0000.. DM1999	-	Данные в области DM доступны только в виде слов. Значения слов сохраняются при отключении питания или при пуске / останове программы. В область чтения/записи можно читать и писать свободно в программе.
	Протокол ошибок[4]	22 слова	DM1000.. DM 1021	-	Служит для сохранения времени появления и кода произошедших ошибок. См. 5-5.
	Только читать[4]	456 слов	DM6144.. DM 6599	-	Не могут быть переписаны из программы.
	Установочные параметры ПК[4]	56 слов	DM6600.. DM 6655	-	Служат для сохранения параметров, управляющих работой ПК.

Замечание 1. Биты IR и LR, которые не используются для отведенных им функций, можно использовать как рабочие биты.

2. Содержание областей HR, AR, счетчиков и DM чтения/записи поддерживается конденсатором. При 25 град. конденсатор сохраняет память 20 суток. График зависимости времени сохранения от температуры см. 2-1-2 Инструкции по работе с Мастер-Модулем SRM1.

3. При обращении к текущим значениям номера таймеров/ счетчиков используются как слова; при обращении к флагам завершения они используются как биты.

4. Значения DM 6144..DM 6655 нельзя переписать из программы, но можно изменить с периферийных устройств.

3.3.2 Область IR

Далее описаны функции области IR.

Биты входов/ выходов

Биты области IR 00000..IR 01915 выделены точкам входов/выходов на Модуле ЦПУ и на Модуле входов/выходов. Они отражают состояние 1/0 входных и выходных сигналов. Биты входов начинаются с IR 00000, биты выходов - с IR 01000.

Подробности см. 1.4 в Инструкции по работе с Мастер-Модулем SRM1.

Рабочие биты

Рабочие биты можно использовать в программе произвольно. Их можно использовать только внутри программы, но не как прямые внешние входы/выходы.

3.3.3 Область SR

Данные биты главным образом служат в качестве флагов, относящихся к операциям SRM1, или содержат текущее значение и задание для различных функций. Подробности о различных функциях битов см. соответствующие пункты в данной главе или в Приложении C, области памяти.

SR 244..SR 247 и SR 250, 251 могут также использоваться в качестве рабочих бит, когда входные прерывания не используются в режиме счетчика. SR 232..SR 239 могут также использоваться в качестве рабочих бит, когда не используется команда MCRO (99).

3.3.4 Область TR

В случае, когда сложную релейно-контактную схему нельзя запрограммировать в мнемокоде, эти биты служат для временного хранения условий исполнения 1/0 при ветвлениях программы. Они используются только для мнемокода. Когда программирование производится непосредственно релейно-контактной схемы с помощью LSS или SSS, биты TR обрабатываются автоматически. Одни и те же биты нельзя использовать более одного раза внутри одного командного блока, но могут использоваться повторно в разных командных блоках. Состояние 1/0 битов TR нельзя просматривать с периферийного устройства.

Примеры, показывающие использование битов TR в программировании, приведены в п.4.3.8

3.3.5 Область HR

Данные биты сохраняют свое состояние 1/0 даже при отключении питания SRM1 или при пуске или останове программы. Они используются таким же образом, как и рабочие биты.

3.3.6 Область AR

Данные биты главным образом служат в качестве флагов отработки программы SRM1. Данные биты сохраняют свое состояние даже при отключении питания SRM1 или при пуске и останове программы. Подробности о функциях битов см. Приложение C, Области памяти.

3.3.7 Область LR

Когда SRM1 подключен в режиме 1:1 к другим ПК данные биты служат для образования общего поля данных. Подробности см.1.8.7

Если биты LR не используются в качестве битов связи, их можно использовать в качестве рабочих битов.

3.3.8 Область таймеров/ счетчиков

Данная область используется для управления таймерами и счетчиками, созданными командами TIM, TIMH(15), CNT и CNTR(12). Одни и те же номера используются и для таймеров и для счетчиков и каждый номер можно использовать только раз в программе пользователя. Не используйте одинаковые номера таймеров/ счетчиков дважды даже для разных команд.

Номера TC служат для создания таймеров и счетчиков, а также для доступа к флагам завершения и текущим значениям. Если номер TC задан для данных типа слово, он будет работать с текущим значением; если номер задан для данных типа бит, он будет обращаться к флагам завершения таймера/счетчика.

Подробности о таймерах и счетчиках см. данную инструкцию, п.5.15

3.3.9 Область DM

Данные в области DM доступны только в виде слов. Содержание области DM сохраняется даже при отключении питания CPM1 или при пуске и останове программы.

Слова DM 0000..DM 1999, DM 2022 и DM 2047 можно использовать в программе произвольно. Другие слова DM выполняют специальные функции, описанные далее.

Протокол ошибок

DM 2000..DM 2021 содержит информацию протокола ошибок. Подробности см. Главу 8, поиск ошибок.

Установочные параметры ПК

DM 6600..DM 6655 содержат установочные параметры ПК. Подробности см. Главу 1-1, Установочные параметры ПК.

3.4 Флэш-память SRM1

Для использования области флэш-памяти в SRM1 нужно задать следующие установочные параметры.

Запись данных

Для записи содержимого области UM, области только чтения DM (DM 6144..6599) и области установочных параметров (DM 6600..6655) во флэш-память нужно произвести одну из следующих операций:

- Переключить SRM1 в режим MONITOR или PROGRAM.
- Выключить и снова включить SRM1 в сеть.

Замечание Если в данных областях были сделаны изменения, они не будут записаны во флэш-память, и если питание отключить более чем на 20 дней (при 25 C), изменения в ОЗУ будут потеряны. В этом случае при запуске ПК из флэш-памяти будет считано старое (до изменения) содержание.

Изменения в областях памяти

При первом пуске программы после внесенных изменений в области UM, области только чтения DM (DM 6144..6599) и области установочных параметров (DM 6600..6655) обратите внимание на последствия задержки отработки программы SRM1.

Первый цикл отработки программы SRM1 после внесения изменений в вышеуказанные области будет максимум на 850 мс дольше, чем нормальный первый цикл без изменений.

Времена отработки цикла

Предупреждение превышения времени цикла не будет выдаваться при исполнении следующих операций в режиме MONITOR или OPERATION. Обращайте внимание на результаты влияния он-лайнового редактирования на время реакции на входы/выходы SRM1.

- Изменения в программе с помощью он-лайнового редактирования .
- Изменения в области только чтения DM (DM 6144..6599)
- Изменения в области установочных параметров (DM 6600..6655)

При выполнении одной из этих операций время цикла SRM1 увеличится максимум на 850 мс. В течение этого времени прерывания будут запрещены при записи программы или содержимого памяти.

3.5 Использование кассет памяти (только CQM1)

Когда используется кассета памяти (опция), установочные параметры, программу пользователя, фиксированные DM и таблицы команд можно хранить в ROM. Это предотвращает нежелательные изменения, которые может произвести случайная запись. Кроме того, при изменении процессов управления установочные параметры и программу можно легко изменить простой заменой кассеты памяти.

В данной главе описаны чтение, запись и сравнение информации в и из кассеты памяти.

3.5.1 Типы кассет памяти и содержание

Кассеты памяти

Имеется 6 типов кассет памяти, как показано в следующей таблице.

Модель	Примечания
CQM1-ME04K	Тип EEPROM (без часов) 4K слов
CQM1-ME04R	Тип EEPROM (с часами) 4K слов
CQM1-MP08K	Тип EPROM (без часов) 8K слов
CQM1-MP08R	Тип EPROM (с часами) 8K слов
CQM1-ME08K	Тип EEPROM (без часов) 8K слов
CQM1-ME08R	Тип EEPROM (с часами) 8K слов

Следующие микросхемы EEPROM (продаются отдельно) требуются для кассет памяти типа EEPROM.

Модель	Версия ROM	Емкость	Скорость доступа
ROM-ID-B	27128 или эквивалент	8K слов	150 нс
ROM-JD-B	27256 или эквивалент	16K слов	150 нс
ROM-KD-B	27512 или эквивалент	32K слов	150 нс

Для CQM1 можно записать максимум 8K слов, поэтому все перечисленные типы микросхем имеют достаточную емкость, и выбирайте те, которые Вам легче приобрести.

Об использовании кассеты памяти см. Инструкцию по работе CQM1.

Содержание

Информация, которую можно записать в кассету памяти, приведена в таблице.

Информация	Содержание
Фиксированные DM	Фиксированные DM нельзя записать из программы. Область записи: DM 6144..DM 6568. Данные слова доступны пользователю.
Установочные параметры ПК	Установочные параметры ПК задают рабочие параметры CQM1 и хранятся в DM 6600..DM 6655.
Таблица команд	Таблица команд присваивает функциональные коды дополнительным командам и разрешает их использование при программировании.

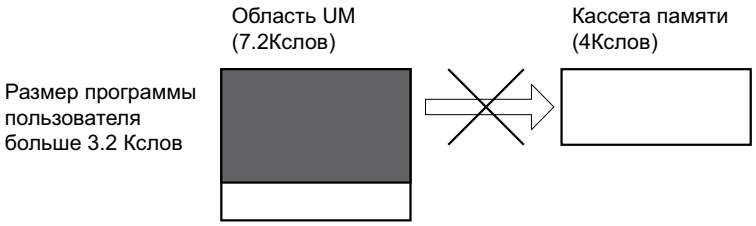
Информация	Содержание
Память программы пользователя (UM)	Область UM содержит программу пользователя.

Данную информацию нельзя прочесть, записать и сравнить по частям, но нужно работать как с единым блоком.

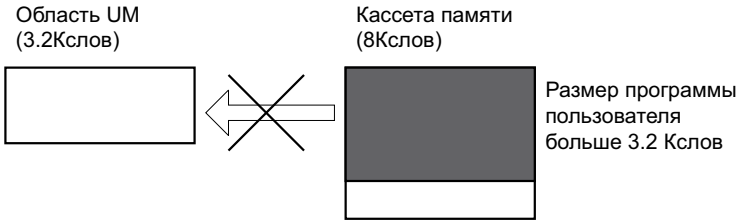
3.5.2 Емкость кассеты памяти и размер области UM

При попытке записать слишком длинную программу появляется признак нефатальной ошибки. Есть 2 случая, в которых возникают ошибки.

1, 2, 3,... 1. Когда в CQM1-CPU4-E с областью UM 7.2 К слов установлена кассета памяти с EEPROM 4К слов, программа длиной до 3.2 К слов можно записать в кассету памяти. При попытке записать программу длиной более 3.2 К слов в кассету памяти появляется признак нефатальной ошибки.



2. Когда в CQM1-CPU11/21-E с областью UM 3.2 К слов установлена кассета памяти объемом памяти 8К слов и выше, программа длиной до 3.2 К слов можно считать с кассеты. При попытке считать программу длиной более 3.2 К слов с кассеты памяти появляется признак нефатальной ошибки.



Замечание Если размер программы менее 3.2 К слов, передача в вышеприведенных случаях завершится нормально.

Примерный размер программы в области UM и кассеты памяти можно определить по содержимому AR 15, как показано в следующей таблице.

Местонахождение программы	Биты	Содержание	Значение
Кассета памяти	AR 1500..AR 1507	00	Кассета памяти не установлена или в кассете программа не записана.
		04	Объем программы менее 3.2 К слов и может быть считан с любого CQM1.
		08	Объем программы менее 7.2 К слов и может быть считан только с CQM1-CPU4-E.
Область UM	AR 1508..AR 1515	04	Объем программы менее 3.2 К слов и может быть записана на любую кассету памяти.
		08	Объем программы менее 7.2 К слов и может быть записана на кассету памяти емкостью 8К слов и более.

В CQM1-CPU11/21-E содержимое AR 1508..AR 1515 нормально 04, а содержание AR 1500..AR 1507 нормально 04 при установке кассеты 4 К слов.

Размер программы, указанный в AR 15, не включает команды NOP(00) после END(01), но включает все другие команды, отличные от NOP(00). Позаботьтесь о том, чтобы

очистить все ненужные команды после NOP(00) для того, чтобы получить реальный объем программы.

3.5.3 Запись в кассету памяти

Запись в кассету памяти типа EPROM выполняется с помощью SSS и устройства записи в PROM. Команды по использованию SSS см. Инструкцию по работе с SSS.

Для записи произведите следующие действия.

1, 2, 3,... 1. Проверьте, чтобы переключатель защиты от записи на кассете памяти был OFF (запись разрешена). Если он в положении ON (запись запрещена), отключите питание CQM1 и вытащите кассету, а потом переключите переключатель.

2. Проверьте, чтобы CQM1 находился в режиме PROGRAM. Если он в режимах RUN или MONITOR, воспользуйтесь командами SSS для смены режима.

3. Включите AR 1400 в 1 с SSS.

Информация перепишется с CQM1 в кассету памяти. По окончании операции AR 1400 автоматически переключится в 0.

Внимание! Если появился признак ошибки памяти, данные нельзя записать в кассету.

Замечание Если появился признак ошибки при передачи данных, будет генерироваться признак нефатальной ошибки (FAL9D) и соответствующий бит AR (AR 1412..AR 1415) переключится из 1 в 0. В таком случае см. Главу 8 Поиск ошибок, и сделайте необходимые исправления.

3.5.4 Чтение из кассеты памяти

Есть 2 метода чтения с кассеты памяти в CQM1:

- с помощью периферийного устройства (например, SSS);
- автоматическое чтение при включении ПК.

Если программа в кассете памяти имеет дополнительные команды с функциональными кодами, отличающихся от значений по умолчанию, проверьте, чтобы секция 4 переключателя DIP ЦУ была в положении 1 (функциональные коды пользователя).

Замечание Когда данные читаются с кассеты памяти в CQM1, секция 1 переключателя DIP CQM1 должна быть в положении OFF (т.е. разрешена запись в DM). Перед переключением в положение OFF отключите питание CQM1.

Чтение с кассеты памяти можно выполнять независимо от типа кассеты памяти.

Если при передаче данных появился признак ошибки, будет генерироваться признак нефатальной ошибки (FAL9D) и соответствующий бит AR (AR 1412..AR 1415) будет установлен из 1 в 0. (В таком случае см. Главу 8 Поиск ошибок, и сделайте необходимые исправления).

Работа периферийного устройства

Для использования периферийного устройства для чтения с кассеты памяти произведите следующие действия.

1, 2, 3,... 1. Проверьте, чтобы CQM1 находился в режиме PROGRAM. Если он в режимах RUN или MONITOR, воспользуйтесь периферийным устройством для смены режима.

2. Используйте периферийное устройство для включения AR 1401 в 1.

Информация будет считана из кассеты памяти в CQM1. По окончании операции AR 1401 автоматически переключится в 0.

Автоматическое чтение

Если секция 2 переключателя DIP CQM1 находится в положении ON (автозагрузка), данные автоматически считаются в CQM1 из кассеты памяти при включении питания.

Работа программы невозможна, если во время обмена данных между кассетой памяти и CQM1 произошла ошибка.

Внимание! Обязательно выключите питание перед изменением положения переключателя DIP.

3.5.5 Сравнение содержания кассеты памяти

Содержание кассеты памяти можно сравнить с содержанием памяти CQM1 для проверки, одинаково ли их содержание. Данное сравнение возможно с любым типом кассет.

Для сравнения произведите следующие действия.

1, 2, 3,... 1. Проверьте, чтобы CQM1 находился в режиме PROGRAM. Если он в режимах RUN или MONITOR, воспользуйтесь периферийным устройством для переключения в режим PROGRAM.

2. Включите AR 1402 в 1 с периферийного устройства. Содержание кассеты памяти будет сравниваться с содержанием памяти CQM1. AR 1402 автоматически переключится в 0 после завершения сравнения.

3. Проверьте состояние AR 1403 для проверки результатов сравнения. AR 1403 будет равен 1, если содержание разное или сравнение невозможно из-за того, что CQM1 находится не в режим PROGRAM. Если AR 1403 = 0, сравнение завершилось успешно и содержание одинаково.

Параметром AR 1403 нельзя управлять из программы либо с периферийного устройства. Он управляется только результатами сравнения.

Если была попытка сравнения, когда CQM1 находился не в режиме PROGRAM, появляется признак нефатальной ошибки (FAL9D) и AR 1412 становится = 1. Хотя и AR 1403 будет также 1, сравнения не произойдет. AR 1403 также станет =1 при попытке сравнения, когда кассета памяти не установлена на CQM1.

4. Программирование в виде релейно-контактных схем

В данной главе описаны основные действия и понятия, используемые при написании программ в виде релейно-контактных схем. Она знакомит с командами, используемыми для построения базовой структуры релейно-контактной схемы и управления ее выполнением. Полный набор команд, используемый в программировании, описан в Главе 5, Набор команд.

4.1 Основной алгоритм

Существует несколько основных операций, используемых при написании программы. Бланки, которые можно скопировать для помощи в программировании, приведены в Приложении E, Бланк распределения входов/выходов и Приложение F, 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 Вы применяете биты LR для связи двух ПК, приготовьте таблицу, в которой будет показано применение данных битов.

3. Определите, какие слова доступны в качестве рабочих, и приготовьте таблицу, в которой показано их распределение.

4. Также приготовьте таблицы номеров TC и номеров переходов, чтобы распределить их в соответствии с планом использования. Не забывайте, что функцию номера TC можно задать только один раз в программе; Номера переходов 01..99 можно использовать один раз каждый. (Номер TC описан в п. 5-15, номера переходов в данной главе далее).

5. Нарисуйте релейно-контактную схему.

6. Введите программу в ЦУ. При использовании программатора это повлечет 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 настройку.

Основы программирования в виде релейно-контактных схем и преобразование в мнемокод описаны в п. 4.3. Приготовления для ввода программы с программатора описаны в Инструкции по работе CQM1 и Инструкции по работе CPM1, а с помощью SSS Инструкции по работе с SSS, ПК серии C.

Остальная часть главы 4 освещает более сложное программирование, меры предосторожности при программировании и исполнение программы. Все команды специального применения описаны в Главе 5, Набор команд. Отладка описана в Инструкции по работе CQM1, Инструкции по работе CPM1, Инструкции по работе CPM1A, Инструкции по работе SRM1, Инструкции по работе с SSS ПК серии C. В Главе 8 Поиск ошибок приведена информация, требуемая для отладки.

4.2 Терминология команд

Имеется два типа команд при программировании в виде релейно-контактных схем:

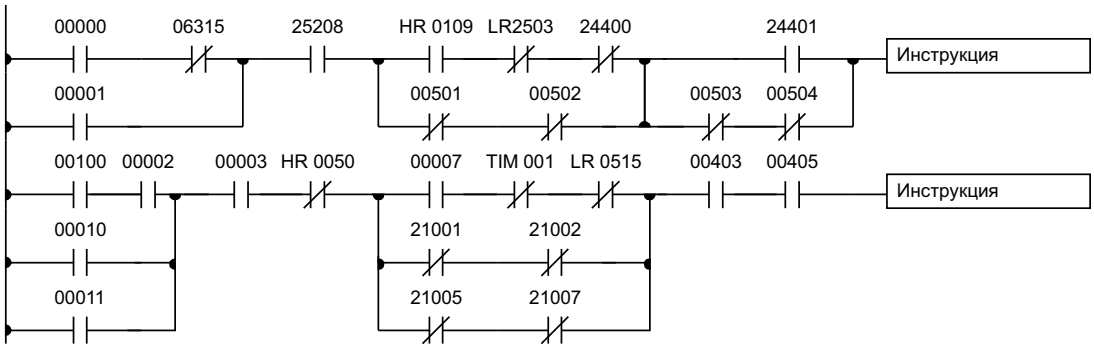
- команды РКС - команды, которые соответствуют условиям на релейно-контактной схеме и используются в форме команд только при преобразовании программы в мнемокод;
- команды, которые располагаются с правой стороны релейно-контактной схеме и выполняются согласно условиям на командных линиях, ведущих к ним.

Большинство команд имеет минимум один или больше связанных с ними операндов. Операнды указывают или содержат данные, над которыми должна исполняться команда. Операнды иногда введены, как численные значения, но обычно представляют собой адреса слов или битов, которые содержат данные, которые будут использоваться. 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 ка константа, ей предшествует # для указания того, что это не адрес.

Другие термины, применяемые для описания команд, приведены в Главе 5, Набор команд.

4.3 Основные релейно-контактные схемы

Релейно-контактная схема состоит из одной вертикальной линии с левой стороны с линиями, отходящими направо. Вертикальная линия слева называется шиной, ответвление - командной линией или ступенькой. На командной линии располагаются условия, ведущие к командам на правой стороне. Логические комбинации этих условий определяют, когда и как выполняются “правосторонние” команды. На рисунке приведен пример релейно-контактной схемы.



Как показано на схеме, командные линии могут разветвляться и снова соединяться. Две рядом расположенные вертикальные линии называются условием. Условия без диагональной черты называются нормально открытыми условиями и соответствуют командам LOAD, AND или OR. Условия с диагональной чертой называются нормально закрытыми условиями и соответствует командам LOAD NOT, AND NOT или OR NOT. Число над каждым условием указывает битовый операнд для 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 некоторые базовые термины.

Замечание При индикации релейно-контактной схемы с помощью SSS с правой стороны будет индикатироваться 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. Инструкцию по работе с SSS, ПК серии C.

4.3.1 Базовые термины

Нормально открытые и нормально закрытые условия

Каждое условие на релейно-контактной схеме может находится 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. Короче говоря, 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 программировании методом релейно-контактных схем логическая комбинация условий 1 и 0 перед командой определяет общее условие, при котором команда выполняется. Данное общее условие (либо 0, либо 1) называется условием исполнения команды. Все команды, за исключением команд LOAD, имеют условия исполнения.

Битовые операнды

Операндами для любой команды релейно-контактной схемы 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 при специальных применениях. Подробности см. 4-3-8.

Логические блоки

Условие исполнения команды определяется отношением между условиями внутри командных строк, которые их соединяют. Любая группа условий, работающих совместно для выработки логического результата, называется логическим блоком. Хотя релейно-контактную схему можно написать без анализа отдельных логических блоков, понимание работы логических блоков необходимо для эффективного программирования и существенно в случае, когда программы должны вводиться в виде мнемокода.

Блок команд

Блок команд состоит из всех команд, которые соединяются между собой, пересекая РКС. Таким образом, один блок команд состоит из всех команд от того места, где Вы можете провести горизонтальную линию через релейно-контактную схему, не пересекая ни одной вертикальной линии, до следующего места, где Вы можете провести другую такую линию.

4.3.2 Мнемокод

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

Из-за важности программатора как периферийного устройства и важности мнемокода в понимании программы, наряду с РКС мы вводим и описываем мнемокоды. Помните, что при вводе с SSS знание мнемокода не требуется (хотя Вы, если предпочитаете, можете использовать его с SSS).

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. Любые 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 все команды, кроме РКС, 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 (21)	
		000
		DM 0000
00007	CMP (20)	
		DM 0000
		HR 00
00008	LD	25505
00009	OUT	10000
00010	MOV (21)	
		DM 0000
		DM 0500
00011	DIFU (13)	00502
00012	AND	00005
00013	OUT	10003

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.

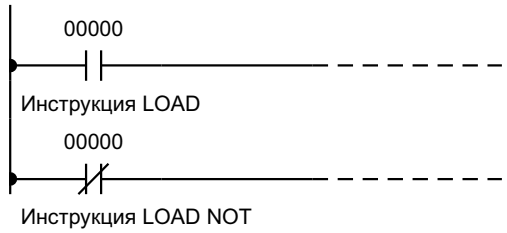
При программировании адреса индикатируются автоматически и нет необходимости вводить адрес, если только по какой-либо причине нет необходимости разместить команду в каком-либо другом месте. При преобразовании в мнемокод лучше всего начинать с адреса памяти программ 00000, если нет особых причин размещать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

4.3.3 Команды, расположенные на релейно-контактной схеме (РКС)

Команды РКС - такие команды, которые соответствуют условиям на РКС. Команды РКС, 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

Первым условием, которым начинается любой логический блок РКС, являются команды LOAD и LOAD NOT. Каждая команда требует одной строки в мнемокоде. В следующем примере “Команда” может быть любой “правосторонней” командой, , которые описаны далее в данной инструкции.

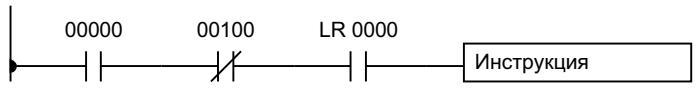


Адрес	Инструкция	Операнд
00000	LD	00000
00001	Инструкция	
00002	LD NOT	00000
00003	Инструкция	

Когда данное условие является единственным в командной строке, 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 соответствует команде LOAD или LOAD NOT. Остальные условия командам AND и AND NOT. В следующем примере показаны три условия, расположенные последовательно: LOAD, AND и AND NOT . 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 исполнение 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 соответствует команде LOAD или LOAD NOT. Остальные условия соответствуют командам 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.

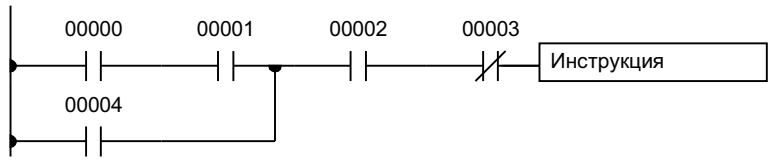
Адрес	Инструкция	Операнд
00000	LD NOT	00000
00001	OR NOT	00100
00002	OR	LR 0000
00003	Инструкция	

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 схемы, их иногда можно рассматривать индивидуально, каждая команда выполняет логическую операцию с предыдущим условием и состоянием битового операнда. Далее приведен пример такой ситуации. Изучите данный пример, пока не поймете, что мнемокод следует той же логике, что и РКС.



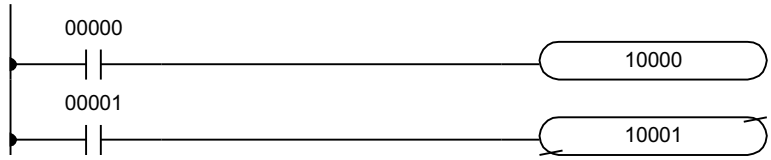
Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	OR	00200
00003	AND	00002
00004	AND NOT	00003
00005	Инструкция	

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 (AND NOT) с состоянием IR 00003.

Однако в более сложных РКС необходимо рассматривать логические блоки перед определением условий исполнения для конечной команды, и именно здесь используются команды AND LOAD и OR LOAD. Перед рассмотрением более сложных схем, мы рассмотрим команды, которые требуются для завершения простейшей программы ”ввод/вывод”.

4.3.4 Вывод и Вывод инверсии (OUTPUT и OUTPUT NOT)

Простейший способ выдать результат комбинаций условий исполнения - прямая выдача командами OUTPUT и OUTPUT NOT. Данные команды служат 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, пока условие исполнения = 0. Командой OUTPUT NOT битовому операнду будет присваиваться значение 1, пока условие исполнения = 0 и 0, пока условие исполнения = 1. Это происходит в соответствии с схемой. В мнемокоде каждая из этих команд требует одной строки.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	10000
00002	LD	00001
00003	OUT NOT	10001

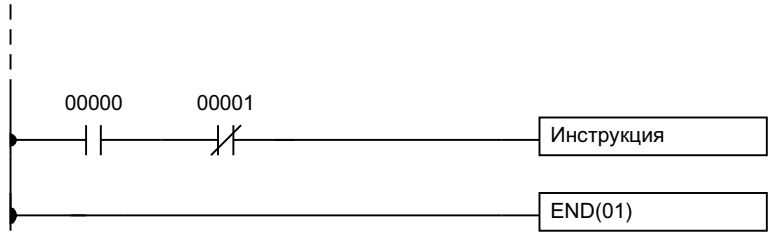
В данных примерах IR 10000 будет в положении 1, пока IR 00000 = 1, а IR 10001 будет = 0, пока IR 00001 = 1. В данном примере IR 00000 и IR 00001 - входные биты, а IR 10000 и IR 100001 - выходные биты, приписанные к блокам ПК. Т.е. сигналы, приходящие с входов, приписанных к IR 00000 и IR 00001, управляют выходами, приписанными к IR 10000 и IR 100001.

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

Подробности см. 5-15-1.

4.3.5 Команда END

Последняя команда, необходимая для завершения простейшей программы -команда END. Когда ЦУ сканирует 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	

Адрес	Инструкция	Операнд
00503	END(01)	-

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

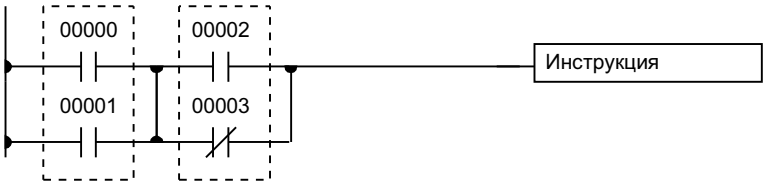
Теперь у Вас есть все команды, требующиеся для написания простейшей программы ввода/вывода. Перед тем, как окончить с основами РКС и перейти к вводу программ в ПК, давайте рассмотрим команды логического блока (AND LOAD и OR LOAD), которые иногда необходимы даже в простых РКС.

4.3.6 Команды логического блока

Команды логического блока не соответствуют конкретным условиям на РКС, а описывают отношения между блоками. Команда AND LOAD производит операцию ЛОГИЧЕСКОЕ И над условиями исполнения, произведенными двумя логическими блоками. Команда OR LOAD производит операцию ЛОГИЧЕСКОЕ ИЛИ над условиями исполнения, произведенными двумя логическими блоками.

AND LOAD

Хотя данная РКС 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 блока отмечены пунктиром. Изучение данного примера показывает, что условие исполнение команды будет = 1,

- когда какое-либо из условий левого блока = 1 (т.е. либо IR 00000, либо IR 00001 = 1) и
- когда какое-либо из условий правого блока = 0 (т.е. либо IR 00002 = 1, либо IR 00003 = 0).

Вышеприведенную РКС нельзя преобразовать в мнемокод, используя только команды AND и OR. Если пытаться 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 (НЕТ) независимо и далее объединить результаты.

Для выполнения данной задачи мы можем пользоваться командой LOAD или LOAD NOT в середине командной строки. При выполнении LOAD или LOAD NOT таким образом, 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 РКС 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 НЕТ, т.е. ИЛИ между состоянием IR 00002 и инверсией IR 00003. Для получения условия исполнения “правосторонней” команды нужно проделать операцию И над этими двумя блоками. Это делает команда AND LOAD. Мнемокод для РКС приведен в таблице под РКС. 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

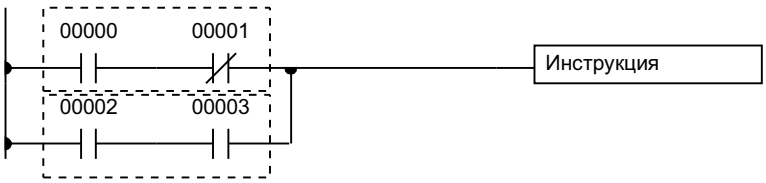
Следующая схема требует команды OR LOAD между верхним и нижним логическими блоками. Условие исполнения для “правосторонней” команды будет = 1,

- либо когда IR 00000 = 1 и IR 00001 = 0
- либо когда IR 00002 = 1 и IR 00003 = 1.

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

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

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



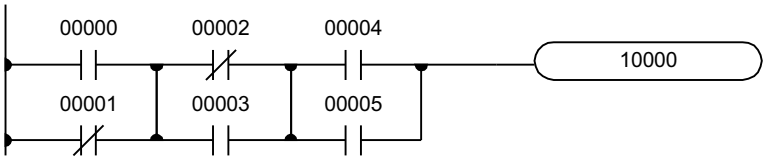
Последовательное включение команд логического блока

Для кодирования РКС с последовательным включением блоковых команд РКС должна быть разбита на логические блоки. Каждый блок кодируется с использованием команды LOAD для кодирования первого условия, затем AND LOAD или OR LOAD для логического объединения блоков. И для AND LOAD, и для OR LOAD есть 2 метода.

Закодировать блоковую команду после первых двух блоков и далее после каждого нового блока.

Кодирование всех блоков, подлежащих объединению, начиная каждый блок с LOAD или LOAD NOT, а затем блоковые команды для их объединения. В этом случае команды для последней пары блоков должны быть заданы первыми, и далее для каждого предшествующего блока, возвращаясь к первому блоку. Хотя оба этих метода производят одинаковый результат, второй метод может применяться только в случае, когда число блоков не более 8.

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

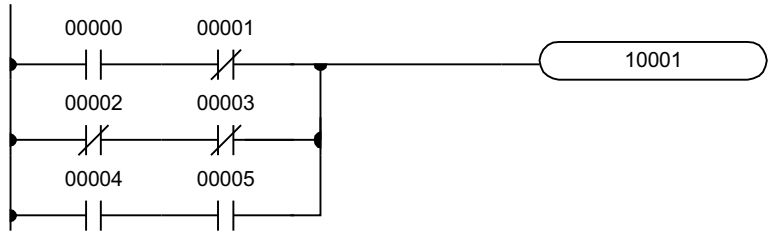


Адрес	Инструкция	Операнд
00000	LD	00000
00001	OR NOT	00001
00002	LD NOT	00002
00003	OR	00003
00004	AND LD	-
00005	LD	00004
00006	OR	00005
00007	AND LD	-
00008	OUT	10000

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	10000

Методом, приведенным в таблице справа, можно объединить максимум 8 логических блоков. Для первого метода число логических блоков не ограничено.

Следующая схема 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 блока, затем OR LOAD, затем последний блок и другая OR LOAD, или можно закодировать сначала три блока, 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	10001

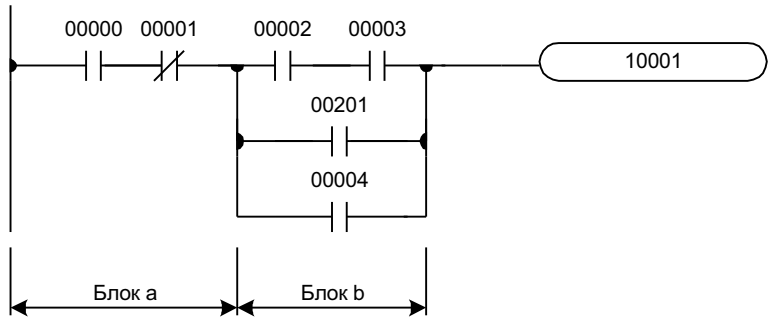
Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND NOT	00003
00004	LD	00004
00005	AND	00005
00006	OR LD	-
00007	OR LD	-
00008	OUT	10001

Методом, приведенным в таблице справа, можно объединить максимум 8 логических блоков. Для первого метода число логических блоков не ограничено.

Комбинирование AND LOAD и OR LOAD

Оба вышеописанных метода можно также использовать при использовании AND LOAD и OR LOAD, пока число блоков не превышает 8.

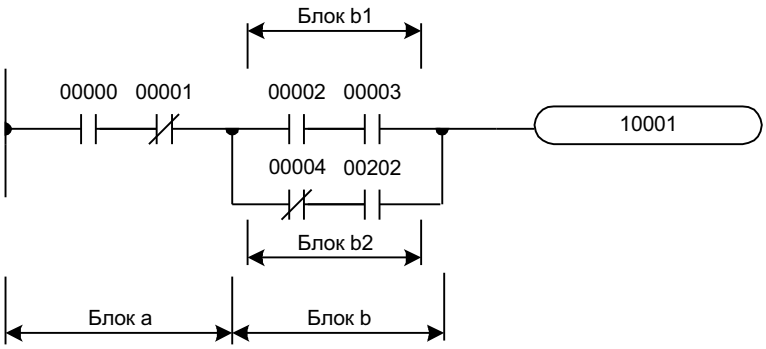
Следующая схема содержит только 2 логических блока. Нет необходимости дальнейшего дробления компонентов блоков, поскольку их можно закодировать простыми AND и OR.



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	10001

Хотя следующая схема похожа на предыдущую, блок b нельзя закодировать, не разделив его на 2 блока, объединенных OR LOAD. В данном примере сначала закодированы три блока и далее применена команда OR LOAD для объединения двух последних блоков, а потом AND LOAD для объединения условия исполнения, полученного командой AND LOAD с условием исполнения блока a.

При кодировании команд логических блоков в конце логических блоков, которые они объединяют, они должны кодироваться в обратном порядке, т.е. сперва идет блоковая команда для последних двух блоков, далее команда, объединяющая условие исполнения, полученное первой блоковой командой, и условие исполнения блока, третьего от конца.



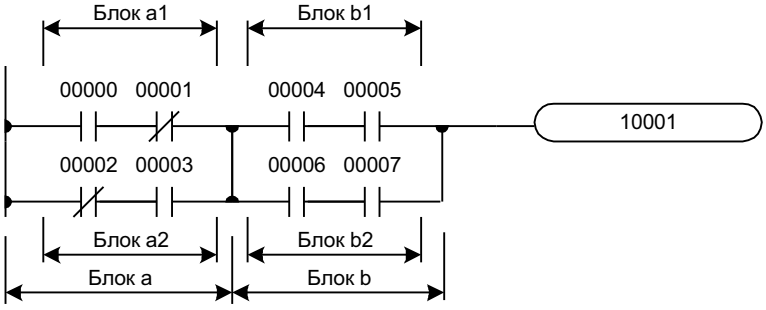
Адрес	Инструкция	Операнд
00000	LD NOT	00000
00001	AND	00001
00002	LD	00002
00003	AND NOT	00003
00004	LD NOT	00004
00005	AND	00202
00006	OR LD	-
00007	AND LD	-
00008	OUT	10002

Сложные схемы

При определении, какие блоковые команды будут необходимы для кодирования схем, иногда необходимо разбить схему на большие блоки и затем продолжать дробить большие блоки до тех пор, пока не будут сформированы блоки, которые можно программировать без блоковых команд. Затем данные блоки кодируются, сначала объединяются маленькие блоки, затем объединяются большие. Для объединения блоков 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 блока или предыдущих блоковых команд.

При работе со сложными схемами блоки 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.

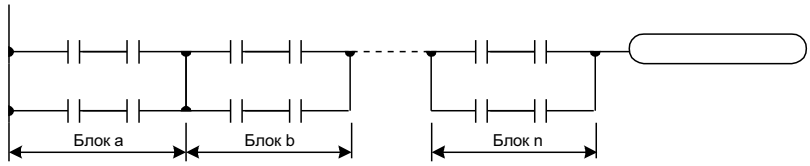
Следующую схему перед тем, как ее можно будет кодировать, нужно разбить на 2 блока и каждый из этих блоков разбить на 2 блока. Как показано, блоки a и b требуют AND LOAD. Перед применением AND LOAD нужно применить OR LOAD для объединения верхних и нижних блоков с каждой стороны, т.е. для объединения a1 и a2; b1 иb2 .



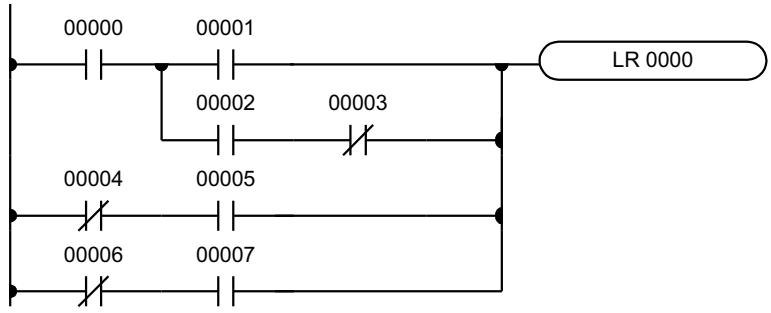
Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	AND NOT	00001
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 1 и 2)***
00010	AND LD	- ***(блок a и)***
00011	OUT	10003

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

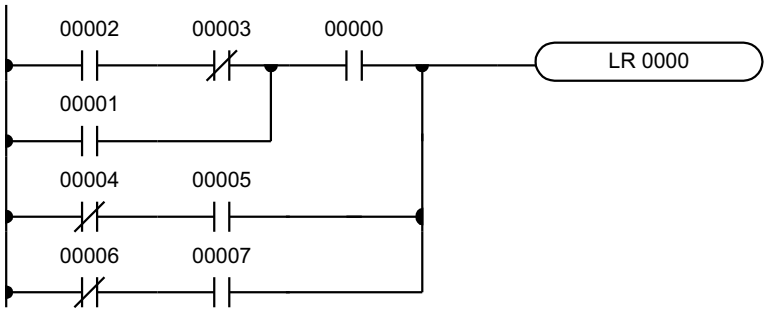


Следующая схема требует команды OR LOAD, затем AND LOAD для кодирования верхних трех блока, и далее два OR LOAD для завершения мнемокода.



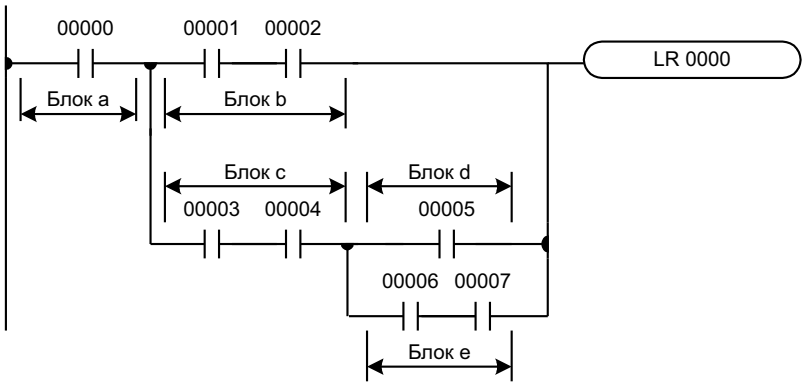
Адрес	Инструкция	Операнд
00000	LD	00000
00001	LD	00001
00002	LD	00002
00003	AND NOT	00003
00004	OR LD	-
00005	AND LD	-
00006	LD NOT	00004
00007	AND	00005
00008	OR LD	-
00009	LD NOT	00006
00010	AND	00007
00011	OR LD	-
0012	OUT	LR 0000

Хотя программа будет выполняться, как и написано, для устранения необходимости первого OR LOAD и AND LOAD данную схему 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

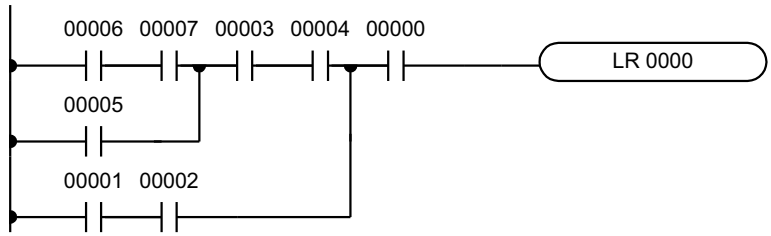
Следующая схема требует 5 блоков, которые здесь кодируются по порядку перед использованием OR LOAD и AND LOAD для их объединения, начиная с последних двух блоков и двигаясь назад. OR LOAD по адресу 00008 объединяет блоки d и e, следующая AND LOAD объединяет результат с результатом блока c и т. Д.



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

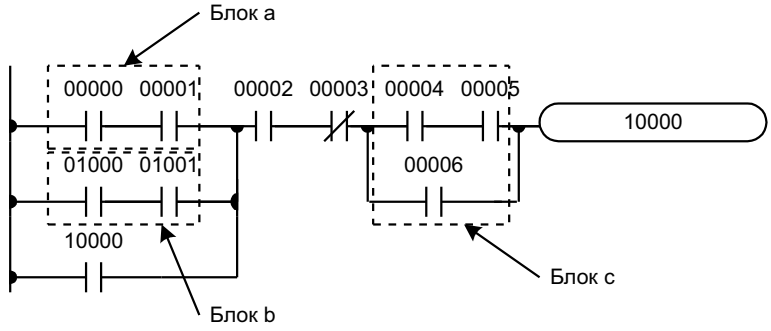
Адрес	Инструкция	Операнд
00007	AND	00007
00008	OR LD	-
00009	AND LD	- ***(блок и)***
00010	OR LD	- ***(блок с результатом предыду ей операции)***
00011	AND LD	- ***(блок с результатом предыду ей операции)***
00012	OUT	LR 0000 ***(блок a с результатом предыду ей операции)***

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

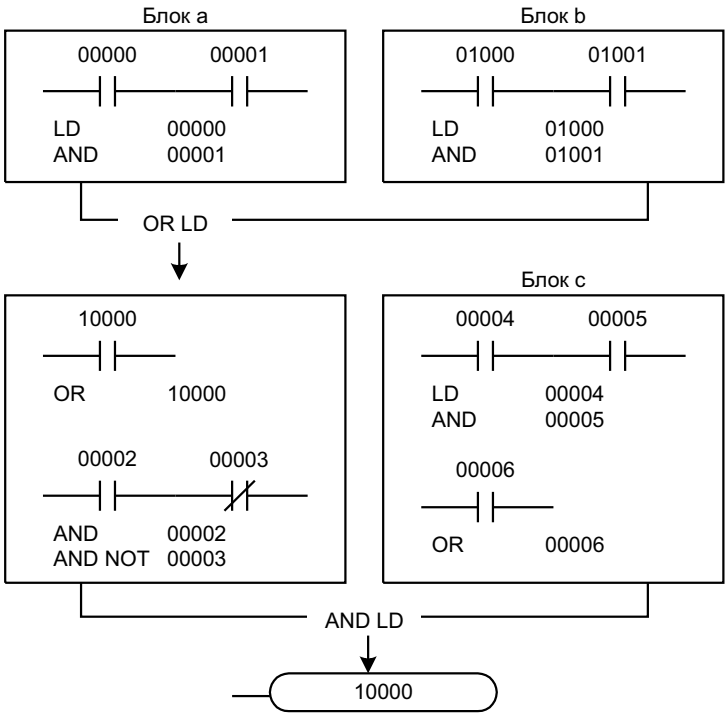


Адрес	Инструкция	Операнд
00000	LD	00006
00001	AND	00007
00002	OR	00005
00003	AND	00003
00004	AND	00004
00005	LD	00001
00006	AND	00002
00007	OR LD	-
00008	AND	00000
00009	OUT	LR 0000

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



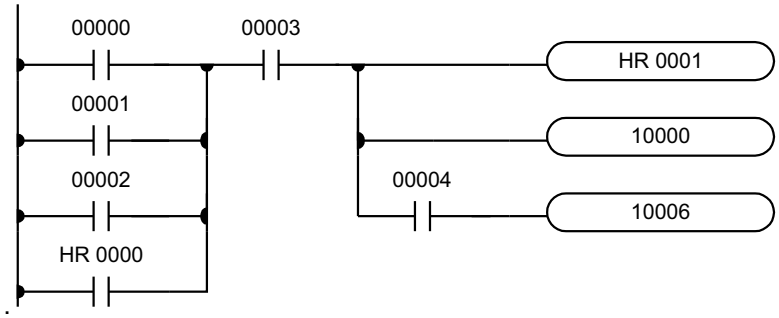
Первая блоковая команда используется для объединения условий исполнения, полученных от блоков a и b, а вторая - для объединения условия исполнения блока “c” с условием исполнения, получающегося от нормально закрытого условия IR 00003. Оставшуюся часть схемы можно закодировать командами OR, AND и AND 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	10000
00006	AND	00002
00007	AND NOT	00003
00008	LD	00004
00009	AND	00005
00010	OR	00006
00011	AND LD	-
00012	OUT	10000

4.3.7 Кодирование нескольких выходных команд

Если несколько выходных команд выполняются с одним условием 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
00002	OR	00002
00003	OR	HR 0000
00004	AND	00003
00005	OUT	HR 0001
00006	OUT	10000
00007	AND	00004
00008	OUT	10006

4.3.8 Командные линии ветвления

Когда командная линия разветвляется на 2 и более линий, иногда необходимо применить либо секцию INTERLOCK, либо биты TR для сохранения условия исполнения, которое существовало в точке ветвления. Это требуется от того, 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 схема иллюстрирует это. В обеих схемах команда 1 выполняется перед возвращением в точку ветвления и далее ветвление на команду 2.

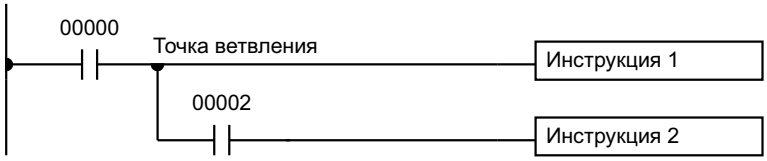


Диаграмма A: коректная операция

Адрес	Инструкция	Операнд
00000	LD	00000
00001	Инструкция 1	
00002	AND	00002
00003	Инструкция 2	

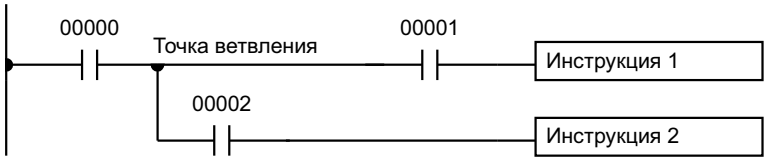


Диаграмма B: некоректная операция

Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	Инструкция 1	
00003	AND	00002
00004	Инструкция 2	

Если, как показано на схеме A, 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 схеме B, 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(02)/ ILC(03)).

Биты TR

Область TR имеет 8 бит, 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 схему B 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.

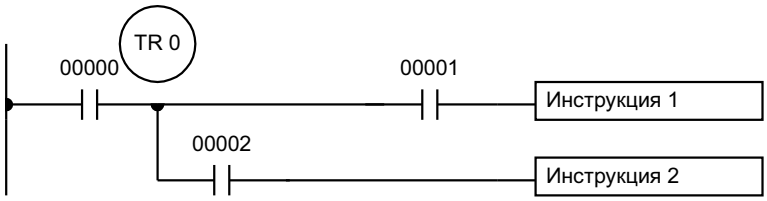
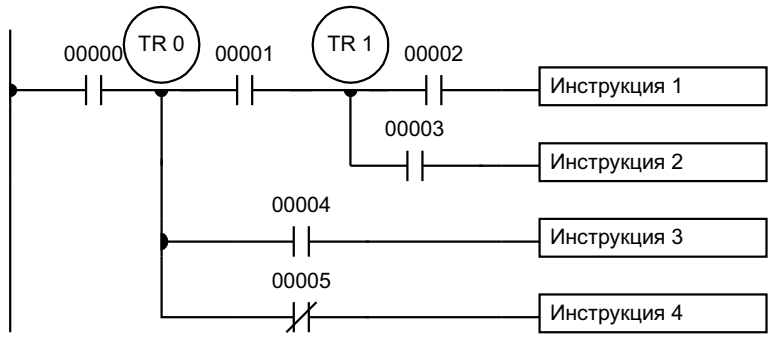


Диаграмма B: корректное использование бита TR

Адрес	Инструкция	Операнд
00000	LD	00000
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 схема 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 выполнится 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.

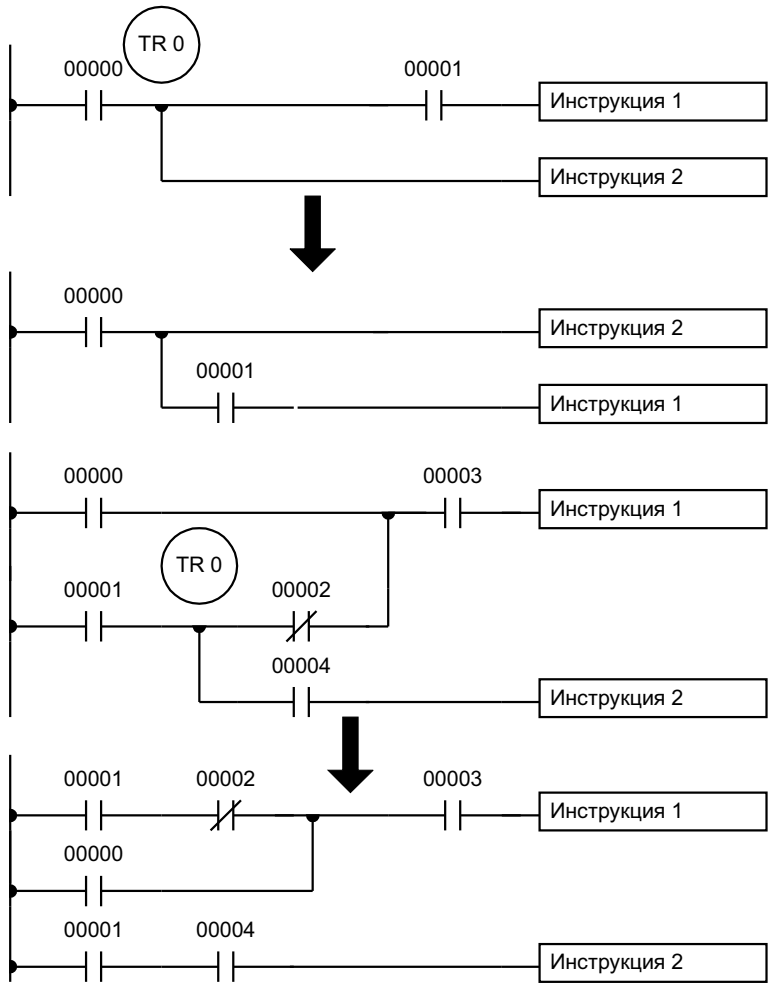


Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	TR 0
00002	AND	00001
00003	OUT	TR 1
00004	AND	00002
00005	Инструкция 1	
00006	LD	TR 1
00007	AND	00003
00008	Инструкция 2	
00009	LD	TR 0
00010	AND	00004
00011	Инструкция 3	
00012	LD	TR 0
00013	AND NOT	00005
00014	Инструкция 4	

В данном примере TR 0 и TR 1 служат 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 блоке команд. Здесь, новый блок команд начинается каждый раз, когда происходит возврат на шину. Если в одном блоке команд необходимо иметь более 8 точек ветвления, которые требуют сохранения результата, нужно применять команды INTERLOCK (описаны далее).

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.



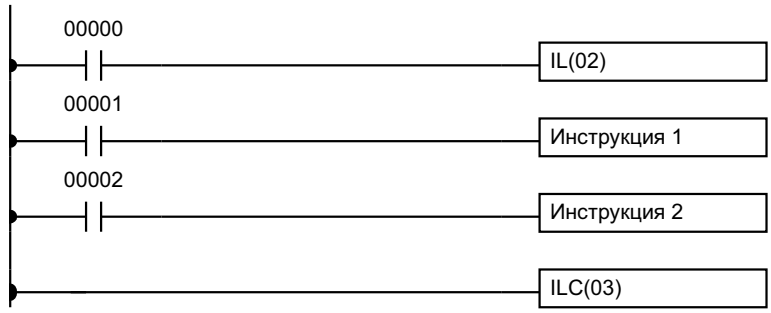
Замечание Биты TR должны вводиться только при программировании в мнемокоде. При вводе РКС их вводить не нужно, поскольку это делается автоматически. Однако вышеуказанные ограничения на число ветвлений, требующих TR, и на методы сокращения команд сохраняются.

“Сгруппировать” - INTERLOCK

Проблемы сохранения условий исполнения в точках ветвления можно решить также командами INTERLOCK (сгруппировать) IL(02) и INTERLOCK CLEAR (разгруппировать) IL(03) 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 секцией релейно-контактной схемы, 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 РКС). Влияние, которое они оказывают на конкретные команды, смотрите гл. 5 -11 INTERLOCK и INTERLOCK CLEAR.

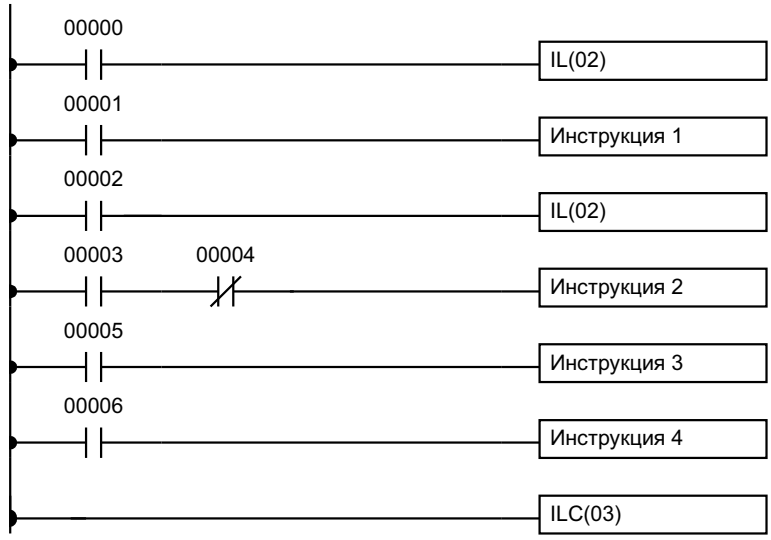
Схему B можно откорректировать также с помощью INTERLOCK. Здесь условия исполнения до точки ветвления ставятся в командную строку для команды INTERLOCK , все строки от точки ветвления записываются как отдельные командные строки и добавляется еще одна команда INTERLOCK CLEAR. Обратите внимание, что ни INTERLOCK, ни INTERLOCK CLEAR не требуют операнда.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL(02)	-
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002
00005	Инструкция 2	
00006	ILC(03)	-

Если в новой версии схемы 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 схеме, внутри блока можно применить более одной команды INTERLOCK. Каждая действует до первой INTERLOCK CLEAR.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL(02)	-
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002

Адрес	Инструкция	Операнд
00005	IL(02)	-
00006	LD	00003
00007	AND NOT	00004
00008	Инструкция 2	
00009	LD	00005
00010	Инструкция 3	
00011	LD	00006
00012	Инструкция 4	
00013	ILC(03)	-

Если в данной схеме 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 схеме 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 определят первое условие исполнения на командных строках.

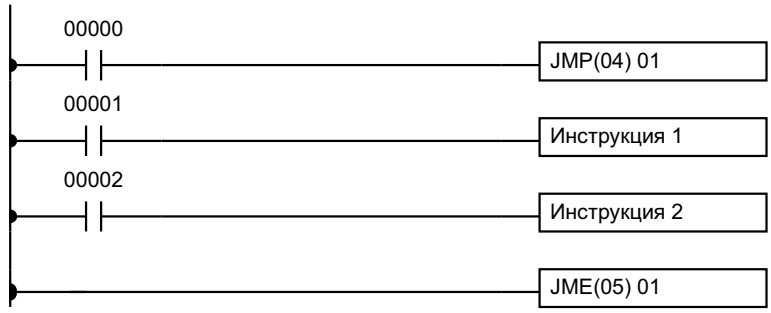
4.3.9 Переходы

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(04) и JME(05) (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. Схема B 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.



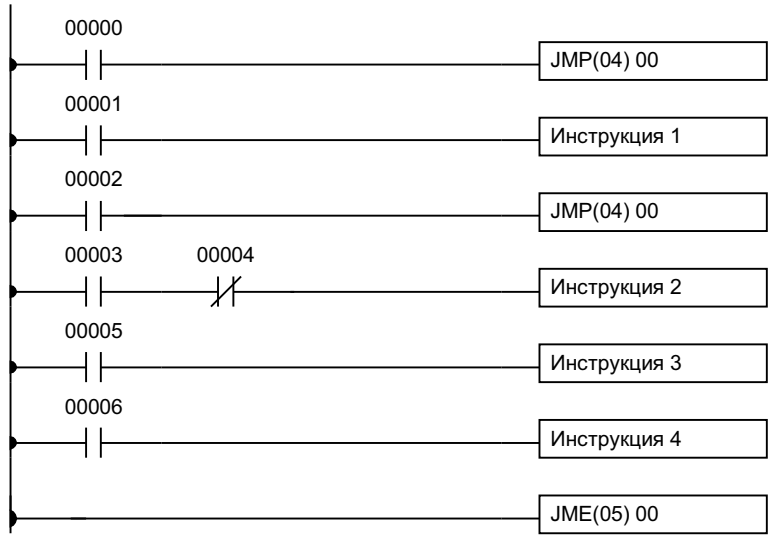
Адрес	Инструкция	Операнд
00000	LD	00000
00001	JMP(04)	01
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002
00005	Инструкция 2	
00006	JME(05)	01

Данная версия схемы B 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 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 схема 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(04)	00
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002

Адрес	Инструкция	Операнд
00005	JMP(04)	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(05)	00

4.4 Управление состоянием битов

Есть 7 базовых команд, используемых для управления состоянием отдельного бита. Это OUTPUT, OUTPUT NOT, SET, RESET, DIFFERENTIATE UP, DIFFERENTIATE DOWN и KEEP. Все эти команды ставятся самыми последними в командной строке 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 служат 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 управлением состоянием других битов в области IR или в других областях.

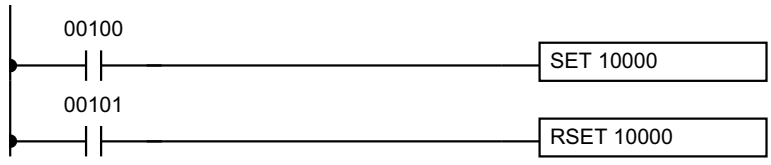
4.4.1 Установить в 1 (SET) и Сбросить в 0 (RESET)

Команды SET и RESET очень похожи на команды OUTPUT и OUTPUT NOT за исключением того, что они изменяют состояние битовых операндов только при условиях исполнения = 1. Данные команды не влияют на состояние операндов, если их условия исполнения = 0.

SET включит операнд в 1 при условии исполнения = 1, но в отличие от команды OUTPUT SET не выключит его в состояние 0 при условии исполнения = 0

RESET выключит битовый операнд в 0 при условии исполнения = 1, но в отличие от команды OUTPUT RESET не включит его в состояние 1, когда условие исполнения = 0.

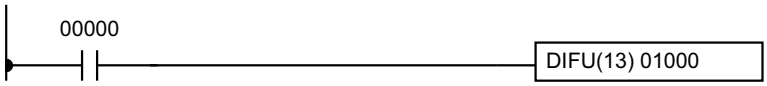
В следующем примере IR 10000 будет = 1 при IR 00100 = 1 и останется = 1, пока IR 00101 не станет =1, несмотря на состояние IR 00100. Когда IR 00101 станет = 1, RESET сбросит IR 10000 в 0.



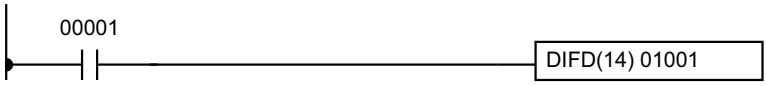
Адрес	Инструкция	Операнд
00000	LD	00100
00001	SET	10000
00002	LD	00101
00003	RSET	10000

4.4.2 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) служат 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 включает битовый операнд в состояние 1 после того, как условие исполнения изменится с 1 на 0; Обе этих команды требуют только одной строки в мнемокоде.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	DIFU(13)	01000



Адрес	Инструкция	Операнд
00000	LD	00001
00001	DIFU(14)	01001

Здесь, IR 01000 будет включен в 1 на 1 цикл после включения IR 00000 в состояние 1. На следующем цикле после выполнения DIFU (13) IR 01000 будет выключено в 0 независимо от состояния IR 00000. Командой DIFFERENTIATE DOWN IR 01001 будет включен в 1 на 1 цикл после включения IR 00001 в состояние 0 (до того времени IR 01001 будет = 0), и будет выключен в 0 после выполнения DIFD (14).

4.4.3 Сохранить (KEEP)

Команда KEEP служит для сохранения состояния битового операнда и требует двух входов условий исполнения. К команде KEEP проводятся 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 схемы 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 либо IR 00004 либо 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

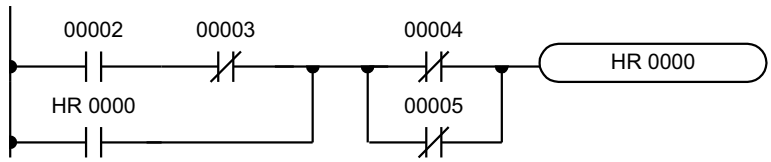
Адрес	Инструкция	Операнд
00003	OR	00005
00004	KEEP(11)	HR 0000

4.4.4 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 схему с командой KEEP можно переписать, как показано на схеме ниже. Единственная разница в схемах будет только при работе в группе с командой INTERLOCK, когда условие команды INTERLOCK = 1. Здесь, как и на схеме с командой KEEP, используются 2 бита сброса, т.е. HR 0000 можно переключить в 0 включением 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

4.5 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 словами.

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 бит

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.

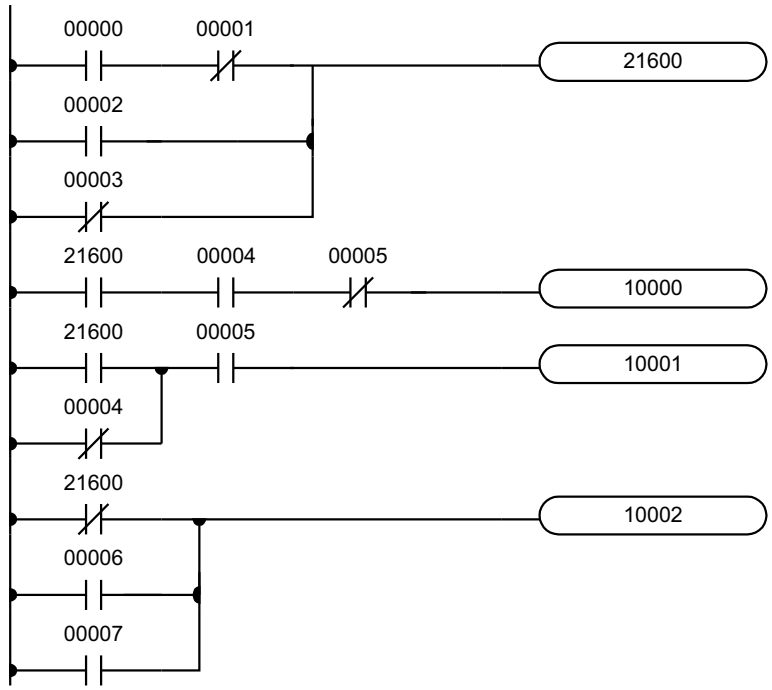
Рабочие биты часто используются с командами OUTPUT, OUTPUT NOT, DIFFERENTIATE UP, DIFFERENTIATE DOWN и KEEP. Сначала рабочие биты

используются в качестве операндов для этих команд, чтобы позднее использовать их в качестве условий для выполнении команд. Рабочие биты можно также использовать и с другими командами, напр. SHIFT REGISTER (регистр сдвига) (SFT(10)). Пример использования рабочих слов с командой. SHIFT REGISTER (SFT(10)) приведен в п. 5-1-16.

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 следующим примере IR 00000, IR 00001, IR 00002 и IR 00003 объединяются в логический блок, который сохраняет результирующее условие исполнение в IR 216000. Далее IR 21600 объединяется в логические блоки 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.



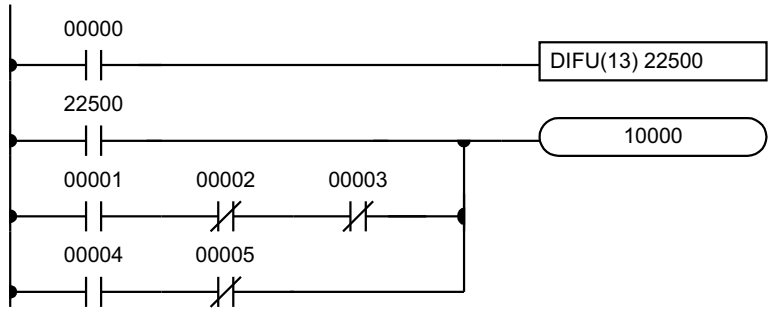
Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001
00002	OR	00002
00003	OR NOT	00003
00004	OUT	21600
00005	LD	21600
00006	AND	00004
00007	AND NOT	00005
00008	OUT	10000
00009	LD	21600
00010	OR NOT	00004
00011	AND	00005
00012	OUT	10001
00013	LD NOT	21600

Адрес	Инструкция	Операнд
00014	OR	00006
00015	OR	00007
00016	OUT	10002

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 00003 = 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(13)). Когда IR 00000 включается в 1, IR 22500 включится в 1 на время 1 цикла и в следующем цикле сбросится в 0 командой DIFU(13). 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(13)	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	10000

4.6 Обратите внимание при программировании

Число условий, которые можно применять последовательно или параллельно, ограничен только емкостью памяти ПК. Так что используйте столько условий, сколько требуется для написания понятных схем. Хотя с помощью командных линий можно написать очень сложные схемы, не должно быть условий на вертикальных линиях между двумя командными линиями. Схема A недопустима, ее нужно перерисовать так, как представлено на схеме B. Мнемокоды даны только для схемы B; кодирование схемы A невозможно.

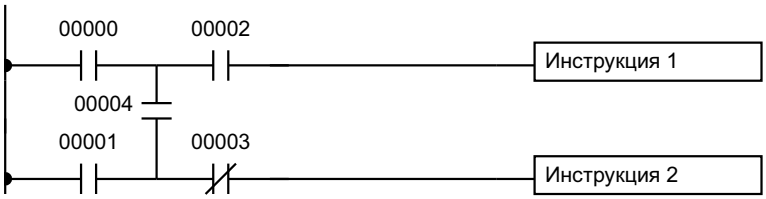


Диаграмма A: е программируется

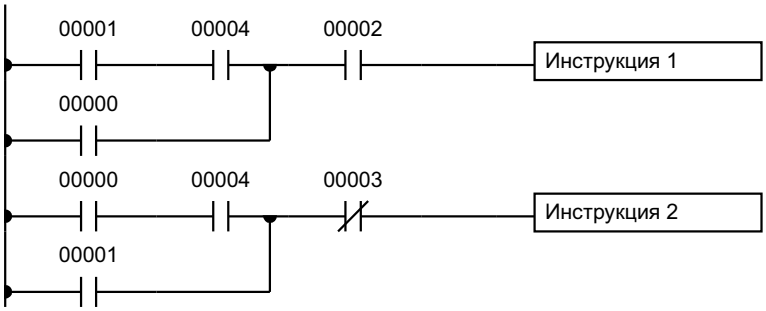


Диаграмма B: Корректная версия

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. Схема A должна быть переписана в виде схемы B. Если команда должна выполняться непрерывно (например, выход должен быть постоянно 1 при выполнении программы), используйте флаг Всегда 1 (SR 25313).

Диаграмма A: е программируется для большинства инструкций

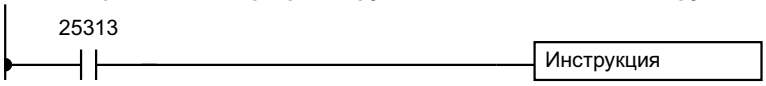


Диаграмма B: Корректная версия

Адрес	Инструкция	Операнд
00000	LD	25313

Адрес	Инструкция	Операнд
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 РКС всегда нужно помнить о количестве команд, которые потребуются для ее ввода. На схеме А потребуется команда OR LOAD для объединения верхней и нижней командной линий. Этого можно избежать, переписав ее в виде схемы В, в которой AND LOAD или OR LOAD не потребуется. Подробности см. 5-7-2.

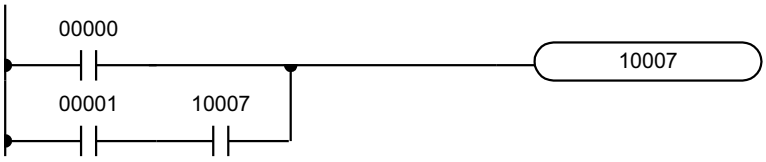


Диаграмма А

Адрес	Инструкция	Операнд
00000	LD	00000
00001	LD	00001
00002	AND	10007
00003	OR LD	-
00004	OUT	10007

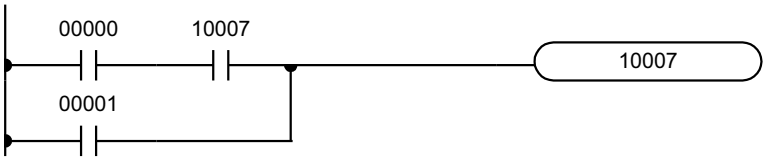


Диаграмма В

Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	10007
00002	OR	00001
00003	OUT	10007

4.7 Исполнение программы

Когда начато исполнение программы, ЦУ 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 ЦУ выполняет за время цикла. Подробности см. гл. 7.

5. Набор команд

CQM1, CPM1/CPM1A и SRM1 располагают большим числом команд программирования, позволяющих просто запрограммировать сложные процессы управления. В данной главе объясняется каждая команда и приводятся ее символ на релейно-контактной схеме, области памяти и флаги, используемые с каждой командой.

Все команды ПК делятся на группы. Эти группы включают команды релейно-контактной схемы, команды с фиксированными функциональными кодами и команды установки (битов).

Некоторые команды, такие как команды таймера и счетчика, используются для управления исполнением других команд, например, Флаг завершения TIM, можно использовать для включения в 1 бита, когда истечет заданное время. Хотя эти команды часто используются для управления выходными битами командами выхода OUTPUT, их можно также использовать для управления исполнением других команд. Выходные команды, использованные в примерах данной инструкции, можно заменить другими командами для модификации программы для специфических задач, а не управлять непосредственно битами.

5.1 Система обозначений

В данной инструкции все обращения к командам будет производится по их мнемонике. Например, команда OUTPUT будет названа OUT, команда AND LOAD - AND LD. Если Вы не уверены, какую команду означает мнемоника, см. Приложение А, 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 ЦУ. Таблица команд в порядке их функциональных кодов также приведена в Приложение А, команды программирования. Списки команд также приведены в 5-6, таблица команд.

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 одно 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 одном слове с операндом. Такие операнды определяют команду, а не указывают на данные. Примеры определитей - 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

- символ на релейно-контактной схеме
- области данных, которые могут использоваться операндами
- 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. 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.

Подраздел Флаги перечисляет флаги, на которые влияет команда. Эти флаги включают следующие флаги области SR

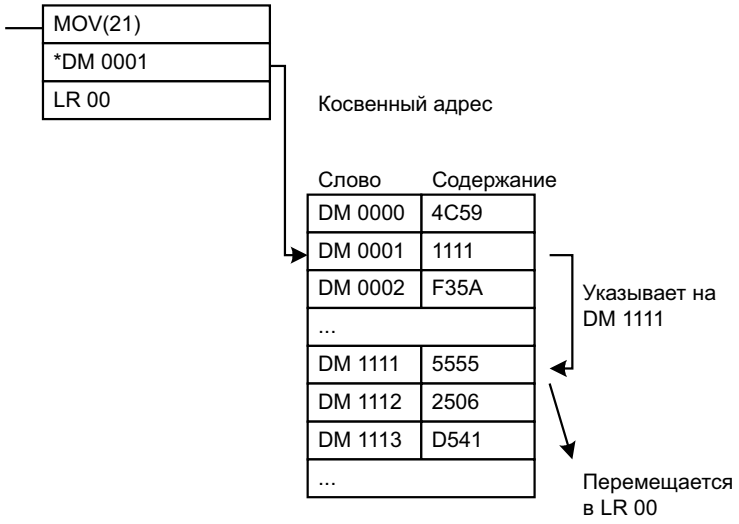
Сокращение	Название	Бит
ER	Instruction Execution Error Flag Флаг ошибки исполнения команды	25503
CY	Carry Flag Флаг переноса	25504
GR	Greater Than Flag Флаг больше чем	25505
EQ	Equals Flag Флаг равно	25506
LE	Less Than Flag Флаг меньше чем	25507

Флаг ER наиболее часто используется 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 B.

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(21) 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..9999.

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, срабатывающие по фронту 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 исполнения = 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 изменилось с 1 на 0, команда не выполняется. Следующие 2 примера показывают работу команд MOV(21) и @MOV(21) , которые используются для пересылки данных по адресу, заданному первым операндом по адресу, заданному вторым операндом.

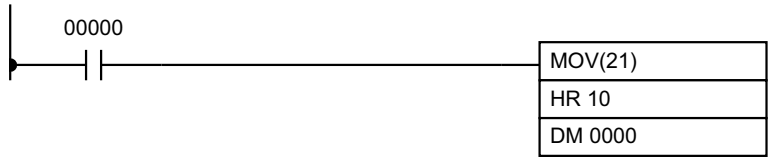


Диаграмма A

Адрес	Инструкция	Операнд
00000	LD	00000
00001	MOV(21)	
		HR 10
		DM 0000

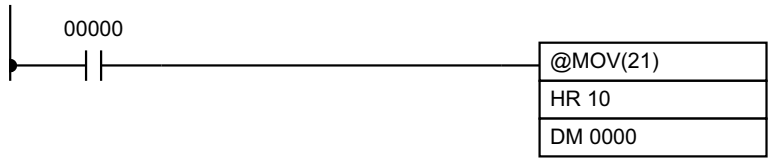


Диаграмма B

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@MOV(21)	
		HR 10
		DM 0000

На схеме A нормальная команда MOV(21) 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, эта операция будет выполняться 25 раз, и только последнее значение, перенесенное в DM 0000, будет там сохраняться.

На схеме B команда @MOV(21) 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, после включения 00000 в 1, содержание DM 0000 после 2 с может отличаться в зависимости от того, какая команда использовалась: MOV(21) или @MOV(21).

Все операнды, символы РКС 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 секций INTERLOCK на команды фронта 0/1 см. 5-11.

В CQM1 имеются еще две команды, срабатывающие при изменении условия исполнения: DIFU(13) и DIFD(14). DIFU(13) работает так же, как и команда фронта 0/1, но применяется для включения бита в 1 на 1 цикл. DIFD(14) также включает бит в 1 на 1 цикл, но при изменении условия исполнения с 1 на 0. Подробности см. 5-8-4.

5.5 Кодирование выходных команд (“правосторонних”)

Написание мнемокода для команд РКС описано в Гл. 4 Программирование РКС. Преобразование информации символов РКС для всех других команд производятся по образцу, описанному ниже, и не указывается для каждой команды индивидуально.

Первое слово любой команды задает команду и включает 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 обозначении команды на релейно-контактной схеме.

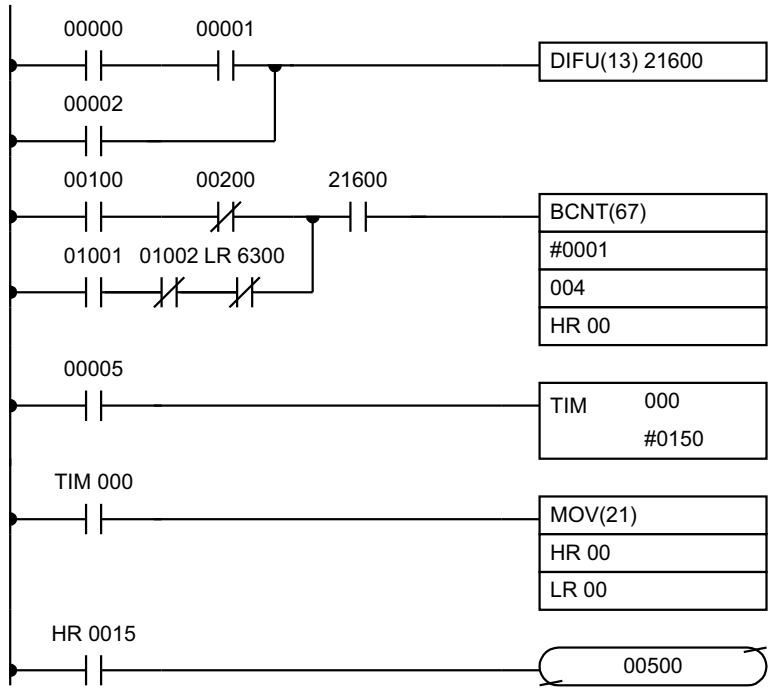
Столбцы адреса и команды таблицы мнемокодов заполняются только для слов команд. Для всех других строк два левых столбца остаются пустыми. Если команда не требует определителя или битового операнда, столбец данных остается пустым для первой строки. Рекомендуем зачеркивать все пустые места столбца данных(для 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 пишите это функциональный код, который необходим при вводе команд с программатора. Также обязательно обозначайте команды фронта 0/1 символом @.

Замечание После мнемоники дополнительных команд идет (-) в качестве функционального кода для указания того, что перед программированием им должен быть присвоен функциональный код в таблице команд. Подробности см. 2-1.

Следующая схема и соответствующий мнемокод иллюстрируют вышеизложенное.

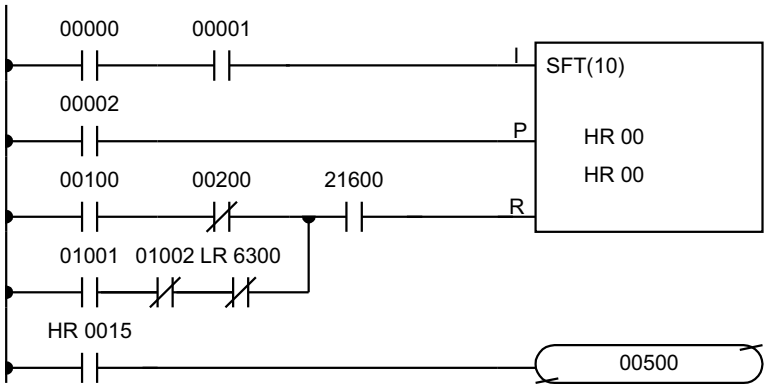


Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	OR	00002
00003	DIFU(13)	21600
00004	LD	00100
00005	AND NOT	00200
00006	LD	01001
00007	AND NOT	01002
00008	AND NOT	LR 6300
00009	OR LD	-
00010	AND	21600
00011	BCNT(67)	
		# 0001
		004
		HR 00
00012	LD	HR 00005
00013	TIM	000
		# 0150
00014	LD	TIM 000
00015	MOV(21)	
		HR 00
		LR 00
00016	LD	HR 0015
00017	OUT NOT	00500

Команда с несколькими командными линиями

Если выходная команда требует несколько командных линий (такие, как KEEP(11)), все командные линии команды вводятся перед самой командой. Каждая командная линия

кодируется начиная с LD или LD NOT для формирования логического блока, который объединяет выходная команда. Далее в качестве примера приведена команда SFT(10).



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	21600
00010	SFT(10)	
		HR 00
		HR 00
00011	LD	HR 0015
00012	OUT NOT	00500

END(01)

Если Вы закончили кодирование программы, обязательно поместите END(01) в последний адрес.

5.6 Таблицы команд

В данной главе приводятся таблицы команд, существующих в CQM1. Первая таблица используется для нахождения команд по функциональному коду. Вторая таблица служит для нахождения команд по мнемокоду. В обеих таблицах значок @ показывает вариант команд фронта 0/1.

5.6.1 Функциональные коды CQM1.

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	(@)TKY	(@)MCMP

Левый разряд	Правый разряд									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
2	CMP	(@)MOV	(@)MVN	(@)BIN	(@)BCD	(@)ASL	(@)ASR	(@)ROL	(@)ROR	(@)COM
3	(@)ADD	(@)SUB	(@)MUL	(@)DIV	(@)ANDW	(@)ORW	(@)XORW	(@)XNRW	(@)INC	(@)DEC
4	(@)STC	(@)CLC	-	-	-	TRSM	(@)MSG	(@)RXD	(@)TXD	-
5	(@)ADB	(@)SBB	(@)MLB	(@)DVB	(@)ADDL	(@)SUBL	(@)MULL	(@)DIVL	(@)BINL	(@)BCDL
6	CMPL	(@)INI	(@)PRV	(@)CTBL	(@)SPEED	(@)PULS	(@)SCL	(@)BCNT	(@)BCMP	(@)STIM
7	(@)XFER	(@)BSET	(@)ROOT	(@)XCHG	(@)SLD	(@)SRD	(@)MLPX	(@)DMPX	(@)SDEC	-
8	(@)DIST	(@)COLL	(@)MOVB	(@)MOVD	(@)SFTR	(@)TCMP	(@)ASC	(@)DSW	(@)7SEG	(@)INT
9	-	(@)SBS	SBN	RET	-	-	-	(@)IORF	-	(@)MCRO

Замечание TRSM(45) нельзя использовать с ЦУ CQM1-CPU11/21-E.

5.6.2 Функциональные коды CPM1/CPM1A.

В следующей таблице перечислены команды CPM1, 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	-	-
2	CMP	(@)MOV	(@)MVN	(@)BIN	(@)BCD	(@)ASL	(@)ASR	(@)ROL	(@)ROR	(@)COM
3	(@)ADD	(@)SUB	(@)MUL	(@)DIV	(@)ANDW	(@)ORW	(@)XORW	(@)XNRW	(@)INC	(@)DEC
4	(@)STC	(@)CLC	-	-	-	-	(@)MSG	-	-	-
5	(@)ADB	(@)SBB	(@)MLB	(@)DVB	(@)ADDL	(@)SUBL	(@)MULL	(@)DIVL	-	-
6	CMPL	(@)INI	(@)PRV	(@)CTBL	-	-	-	(@)BCNT	(@)BCMP	(@)STIM
7	(@)XFER	(@)BSET	-	(@)XCHG	(@)SLD	(@)SRD	(@)MLPX	(@)DMPX	(@)SDEC	-
8	(@)DIST	(@)COLL	(@)MOVB	(@)MOVD	(@)SFTR	(@)TCMP	(@)ASC	-	-	(@)INT
9	-	(@)SBS	SBN	RET	-	-	-	(@)IORF	-	(@)MCRO

5.6.2 Функциональные коды SRM1.

В следующей таблице перечислены команды SRM1, 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	-	-
2	CMP	(@)MOV	(@)MVN	(@)BIN	(@)BCD	(@)ASL	(@)ASR	(@)ROL	(@)ROR	(@)COM
3	(@)ADD	(@)SUB	(@)MUL	(@)DIV	(@)ANDW	(@)ORW	(@)XORW	(@)XNRW	(@)INC	(@)DEC
4	(@)STC	(@)CLC	-	-	-	-	(@)MSG	-	-	-
5	(@)ADB	(@)SBB	(@)MLB	(@)DVB	(@)ADDL	(@)SUBL	(@)MULL	(@)DIVL	-	-
6	CMPL	-	-	-	-	-	-	(@)BCNT	(@)BCMP	(@)STIM
7	(@)XFER	(@)BSET	-	(@)XCHG	(@)SLD	(@)SRD	(@)MLPX	(@)DMPX	(@)SDEC	-
8	(@)DIST	(@)COLL	(@)MOVB	(@)MOVD	(@)SFTR	(@)TCMP	(@)ASC	-	-	-
9	-	(@)SBS	SBN	RET	-	-	-	(@)IORF	-	(@)MCRO

5.6.3 Алфавитный список мнемоники

Мнемоника	Код	Слов	Наименование	Тип ЦУ	Пункт
7 SEG	88	4	7-segment display output Выдача на 7-сегментный индикатор	Только CQM1	5.28.1
ACC(@)	-	4	Acceleration control Управление ускорением	Только CQM1-CPU43-EV1	5.26.12

Мнемоника	Код	Слов	Наименование	Тип ЦУ	Пункт
ADB(@)	50	4	Binary ADD Сложение двоичных чисел	Все	5.21.1
ADBL(@)	-	4	Double Binary ADD Сложение двоичных чисел двойной длины	Только CQM1-CPU4_-EV1	5.21.5
ADD(@)	30	4	BCD ADD Сложение двоично-десятичных чисел	Все	5.20.3
ADDL(@)	54	4	DOUBLE BCD ADD Сложение двоично-десятичных чисел двойной длины	Все	5.20.7
AND	Нет	1	AND Логическое И	Все	5.7.1
AND LD	Нет	1	AND LOAD Логическое И + загрузка	Все	5.7.2
AND NOT	Нет	1	AND NOT Логическое И с инверсией	Все	5.7.1
ANDW (@)	34	4	LOGICAL AND Логическое И двух слов	Все	5.23.2
APR (@)	-	4	ARITHMETIC PROCESS Арифметические расчеты	Только CQM1	5.22.5
ASC (@)	86	4	ASCII CONVERT Преобразование в код ASCII	Все	5.19.8
ASFT (@)	17	4	ASYNCHRONOUS SHIFT REGISTER Асинхронный регистр сдвига	Все	5.16.10
ASL (@)	25	2	ARITHMETIC SHIFT LEFT Арифметический сдвиг влево	Все	5.16.3
ASR (@)	26	2	ARITHMETIC SHIFT RIGHT Арифметический сдвиг вправо	Все	5.16.4
AVG	-	4	AVERAGE VALUE Среднее значение	Только CQM1	5.22.3
BCD (@)	24	3	BINARY TO BCD Двоичное в двоично-десятичное	Все	5.19.2
BCDL (@)	59	3	DOUBLE BINARY-TO-DOUBLE BCD Двоичное двойной длины в двоично-десятичное двойной длины	Только CQM1	5.19.4
BCMP (@)	68	4	BLOCK COMPARE Сравнение блока	Все	5.18.3
BCNT (@)	67	4	BIT COUNTER Битовый счетчик	Все	5.26.5
BIN (@)	23	3	BCD-TO-BINARY Двоично-десятичное в двоичное	Все	5.19.1
BINL (@)	58	3	DOUBLE BCD-TO-DOUBLE BINARY Двоично-десятичное двойной длины в двоичное двойной длины	Только CQM1	5.19.3
BSET (@)	71	4	BLOCK SET Заполненние блока	Все	5.17.4
CLC (@)	41	1	CLEAR CARRY Очистка переноса	Все	5.20.2
CMP	20	3	COMPARE Сравнить	Все	5.18.1
CMPL	60	4	DOUBLE COMPARE Сравнить числа двойной длины	Все	5.18.4
CNT	Нет	2	COUNTER Счетчик	Все	5.15.2
CNTR	12	3	REVERSIBLE COUNTER Реверсивный счетчик	Все	5.13.3

Мнемоника	Код	Слов	Наименование	Тип ЦУ	Пункт
COLL (@)	81	4	DATA COLLECT Собрать данные	Все	5.17.7
COLM (@)	-	4	LINE TO COLUMN Строку в столбец	Только CQM1	5.19.16
COM (@)	29	2	COMPLEMENT Дополнение	Все	5.23.1
CPS	-	4	SIGNED BINARY COMPARE Сравнение двоичных чисел со знаком	Только CQM1-CPU4_-EV1	5.18.6
CPSL	-	4	DOUBLE SIGNED BINARY COMPARE Сравнение двоичных чисел двойной длины со знаком	Только CQM1-CPU4_-EV1	5.18.7
CTBL (@)	63	4	COMPARISON TABLE LOAD Загрузка таблицы сравнений	Все	5.15.6
DBS (@)	-	4	SIGNED BINARY DIVIDE Деление двоичных чисел со знаком	Только CQM1-CPU4_-EV1	5.21.9
DBSL (@)	-	4	DOUBLE SIGNED BINARY DIVIDE Деление двоичных чисел двойной длины со знаком	Только CQM1-CPU4_-EV1	5.21.10
DEC (@)	39	2	BCD DECREMENT Декремент двоично-десятичного числа	Все	5.24.2
DIFD	14	2	DIFFERENTIATE DOUN Включение на 1 цикл по фронту 1/0	Все	5.8.4
DIFU	13	2	DIFFERENTIATE UP Включение на 1 цикл по фронту 0/1	Все	5.8.4
DIST (@)	80	4	SINGLE WORD DISTRIBUTE Распределение одного слова	Все	5.17.6
DIV (@)	33	4	BCD DIVIDE Деление двоично-десятичного числа	Все	5.20.6
DIVL (@)	57	4	DOUBLE BCD DIVIDE Деление двоично-десятичного числадвойной длины	Все	5.20.10
DMPX (@)	77	4	16-TO-4 ENCODER Преобразование 16-в-4	Все	5.19.6
DSW	87	4	DIGITAL SWITCH Ввод с кодового колеса	Только CQM1	5.28.2
DVB (@)	53	4	BINARY DIVIDE Деление двоичных чисел	Все	5.21.4
END	01	1	END Конец программы	Все	5.10
FAL (@)	06	2	FAILURE ALARM AND RESET Признак оошибки и сброс	Все	5.13
FALS	07	2	SEVERE FAILURE ALARM Признак фатальной обшибки	Все	5.13
FCS (@)	-	4	FCS CALCULATE Подсчет КСК (контрольной суммы конверта)	Только CQM1/SRM1	5.26.6
FPD	-	4	FAILURE POINT DETECT Поиск точки сбоя	Только CQM1	5.26.7
HEX (@)	-	4	ASCII-TO-HEXADECIMAL ASCII в 16-ричный вид	Только CQM1/SRM1	5.19.9
HKY	-	4	HEXADECIMAL KEY INPUT Ввод с клавиатуры 16 клавиш	Только CQM1	5.28.3
HMS	-	4	SECOND TO HOURS Секудны в часы	Только CQM1	5.19.14
IL	02	1	INTERLOCK Секция INTERLOCK	Все	5.11

Мнемоника	Код	Слов	Наименование	Тип ЦУ	Пункт
ILC	03	1	INTERLOCK CLEAR Секция INTERLOCK CLEAR	Все	5.11
INC (@)	38	2	INCREMENT Инкремент	Все	5.24.1
INI (@)	61	4	MODE CONTROL Упарвление режимами	Все	5.15.7
INT (@)	89	4	INTERRUPT CONTROL Упарвление прерываниями	Все	5.26.8
IORF (@)	97	3	I/O REFRESH Обновление входов/выходов	Все	5.26.3
JME	05	2	JUMP END Конец перехода	Все	5.12
JMP	04	2	JUMP Переход	Все	5.12
KEEP	11	2	KEEP Сохранить значение	Все	5.8.3
LD	Нет	1	LOAD Загрузка	Все	5.7.1
LD NOT	Нет	1	LOAD NOT Загрузка инверсии	Все	5.7.1
LINE	-	4	COLUMN-TO-LINE Столбец в строку	Только CQM1	5.19.15
MAX (@)	-	4	FIND MAXIMUM Найти максимум	Только CQM1	5.22.1
MBS (@)	-	4	SIGNED BINARY MULTIPLY Умножение двоичных чисел со знаком	Только CQM1-CPU4_-EV1	5.21.7
MBSL (@)	-	4	DOUBLE SIGNED BINARY MULTIPLY Умножение двоичных чисел двойной длины со знаком	Только CQM1-CPU4_-EV1	5.21.8
MCMP (@)	19	4	MULTI-WORD COMPARE Сравнение нескольких слов	Только CQM1	5.18.5
MCRO (@)	99	4	MACRO Макрокоманда	Все	5.26.4
MIN (@)	-	4	FIND MINIMUM Найти минимум	Только CQM1	5.22.2
MLB (@)	52	4	BINARY MULTIPLY Умножение двоичных чисел	Все	5.21.3
MLPX (@)	76	4	4-TO-16 DECODER Преобразование 4-в-16	Все	5.19.5
MOV (@)	21	3	MOVE Пересылка	Все	5.17.1
MOVB (@)	82	4	MOVE BIT Пересылка бита	Все	5.17.8
MOVD (@)	83	4	MOVE DIGIT Пересылка слова	Все	5.17.9
MSG (@)	46	2	MESSAGE Сообщение	Все	5.26.2
MUL (@)	32	4	BCD MULTIPLY Умножение двоично-десятичных чисел	Все	5.20.5
MULL (@)	56	4	DOUBLE BCD MULTIPLY Умножение двоично-десятичных чисел двойной длины	Все	5.20.9
MVN (@)	22	3	MOVE NOT Пересылка инверсии	Все	5.17.2
NEG (@)	-	4	2'S COMPLEMENT Дополнение до 2	Только CQM1-CPU4_-EV1	5.19.17

Мнемоника	Код	Слов	Наименование	Тип ЦУ	Пункт
NEGL (@)	-	4	DOUBLE 2'S COMPLEMENT Дополнение до 2 чисел двойной длины	Только CQM1-CPU4_-EV1	5.19.18
NOP	00	1	NO OPERATION Нет операции	Все	5.9
OR	Нет	1	OR Логическое ИЛИ	Все	5.7.1
OR LD	Нет	1	OR LOAD Логическое Или С Загрузить	Все	5.7.2
OR NOT	Нет	1	OR NOT Логическое Или С Инверсией	Все	5.7.1
ORW (@)	35	4	LOGICAL OR Логическое Или Двух Слов	Все	5.23.3
OUT	Нет	2	OUTPUT Выход	Все	5.8.1
OUT NOT	Нет	2	OUTPUT NOT Выход инверсии	Все	5.8.1
PID	-	4	PID CONTROL ПИД-регулятор	Только CQM1-CPU4_-EV1	5.26.15
PLS2 (@)	-	4	PULSE OUTPUT Выдача импульсов	Только CQM1-CPU43-EV1	5.26.11
PRV (@)	62	4	HIGH-SPEED COUNTER PV READ Чтение текущего состояний высокоскоростного счетчика	Все	5.15.8
PULS (@)	65	4	SET PULSES Задание импульсов	Только CQM1	5.26.9
PWM (@)	-	4	PULSE WITH VARIABLE DUTY RATIO Импульсы с переменным коэффициентом заполнения.	Только CQM1-CPU43-EV1	5.26.13
RET	93	1	SUBROUTINE RETURN Возврат из подпрограммы	Все	5.25.2
ROL (@)	27	2	ROTATE LEFT Циклический сдвиг влево	Все	5.16.5
ROOT (@)	72	3	SQUARE ROOT Квадратный корень	Только CQM1	5.20.11
ROR (@)	28	2	ROTATE RIGHT Циклический сдвиг вправо	Все	5.16.6
RSET	Нет	2	RESET Сброс	Все	5.8.2
RXD (@)	47	4	RECEIVE Прием	Только CQM1/SRM1	5.27.1
SBB (@)	51	4	BINARY SUBTRACT Вычитание двоичных чисел	Все	5.21.2
SBBL (@)	-	4	DOUBLE BINARY SUBTRACT Вычитание двоичных чисел двойной длины	Только CQM1-CPU4_-EV1	5.21.6
SBN	92	2	SUBROUTINE DEFINE Определить подпрограмму	Все	5.22.2
SBS (@)	91	2	SUBROUTINE ENTER Вызвать подпрограмму	Все	5.25.1
SCL (@)	66	4	SCALING Масштабирование	Только CQM1	5.19.10
SCL2 (@)	-	4	SIGNED BINARY TO BCD SCALING Масштабное преобразование двоичных значений в двоично-десятичные	Только CQM1-CPU4_-EV1	5.19.11
SCL3 (@)	-	4	BCD TO SIGNED BINARY SCALING Масштабное преобразование двоично-десятичных значений в двоичные со знаком	Только CQM1-CPU4_-EV1	5.19.13

Мнемоника	Код	Слов	Наименование	Тип ЦУ	Пункт
SDEC (@)	78	4	7-SEGMENT DECODER Декодирование 7-сегментного индикатора.	Все	5.19.7
SEC	-	4	HOURS TO SECONDS Часы в секунды	Только CQM1	5.19.13
SET	Нет	2	SET Установить в 1	Все	5.8.2
SFT	10	3	SHIFT REGISTER Регистр сдвига	Все	5.16.1
SFTR (@)	84	4	REVERSIBLE SHIFT REGISTER Реверсивный регистр сдвига	Все	5.16.9
SLD (@)	74	3	ONE DIGIT SHIFT LEFT Сдвиг одной цифры влево	Все	5.16.7
SNXT	09	2	STEP START Пуск секции STEP	Все	5.14
SPED (@)	64	4	SPEED OUTPUT Задание скорости импульсов	Только CQM1	5.26.10
SRCH (@)	-	4	DATA SEARCH Поиск данных	Только CQM1	5.26.14
SRD (@)	75	3	ONE DIGIT SHIFT RIGHT Сдвиг одной цифры вправо	Все	5.16.8
STC (@)	40	1	SET CARRY Установить перенос	Все	5.20.1
STEP	08	2	STEP DEFINE Определить секцию STEP	Все	5.14
STIM (@)	69	4	INTERVAL TIMER Интервальный таймер	Все	5.15.5
STUP	-	3	CHANGE RS232C SETUP Изменить настройки RS232C	Только SRM1	5.27.3
SUB (@)	31	4	BCD SUBTRACT Вычитание двоичных чисел	Все	5.20.4
SUBL (@)	55	4	DOUBLE BCD SUBTRACT Вычитание двоичных чисел двойной длины	Все	5.20.8
SUM (@)	-	4	SUM Сумма	Только CQM1	5.22.4
TCMP (@)	85	4	TABLE COMPARE Сравнение значение с таблицей	Все	5.18.2
TIM	Нет	2	TIMER Таймер	Все	5.15.1
TIMH	15	3	HIGH-SPEED TIMER Высокоскоростной таймер	Все	5.15.4
TKY (@)	18	4	TEN KEY INPUT Ввод с клавиатуры 10 клавиш	Только CQM1	5.28.4
TRSM	45	1	TRACE MEMORY SAMPLE Выборка для трассировки памяти	Только CQM1-CPU4_-EV1	5.26.1
TXD (@)	48	4	TRANSMIT Передача	Только CQM1	5.27.2
WSFT (@)	16	3	WORD SHIFT Сдвиг слова	Все	5.16.2
XCHG (@)	73	3	DATA EXCHANGE Обмен данными	Все	5.17.5
XFER (@)	70	4	BLOCK TRANSFER Передача блока	Все	5.17.3
XFRB (@)	—	4	TRANSFER BITS Передача битов	Только CQM1-CPU4_-EV1	5.17.10
XNRW (@)	37	4	EXCLUSIVE NOR Исключая ИЛИ-НЕ	Все	5.23.5

Мнемоника	Код	Слов	Наименование	Тип ЦУ	Пункт
XORW (@)	36	4	EXCLUSIVE OR исключая ИЛИ	Все	5.23.4
ZCP	-	4	AREA RANGE COMPARE Сравнение с зоной	Только CQM1-CPU4_-EV1	5.18.8
ZCPL	-	4	DOUBLE AREA RANGE COMPARE Сравнение с зоной чисел двойной длины	Только CQM1-CPU4_-EV1	5.18.9

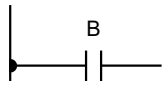
5.7 Команды релейно-контактной схемы

Команды релейно-контактной схемы включают команды, располагающиеся на РКС и блоковые команды. Блоковые команды служат для связи сложных частей схемы.

5.7.1 LOAD, LOAD NOT, AND, AND NOT, OR и OR NOT

LOAD - LD

Обозначение на схеме

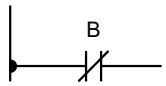


Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

LOAD NOT - LD NOT

Обозначение на схеме

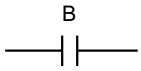


Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

AND - AND

Обозначение на схеме

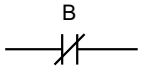


Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

AND NOT - AND NOT

Обозначение на схеме

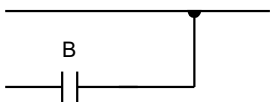


Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

OR - OR

Обозначение на схеме

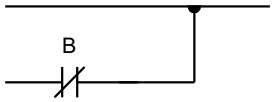


Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

OR NOT - OR NOT

Обозначение на схеме



Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

Ограничения

Ограничений количества любой из этих команд или порядка их применения нет, если программа вмещается в отведенную память.

Описание

Данные шесть базовых команд соответствуют условиям на РКС. Как описано в гл. 4, состояние битов, приписанных 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 между 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;

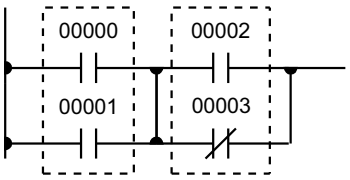
Флаги

Флагов, на которые оказывают воздействие данные команды, нет.

5.7.2 AND LOAD и OR LOAD

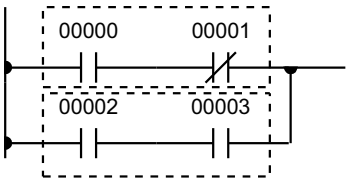
AND LOAD - AND LD

Обозначение на схеме



OR LOAD - OR LD

Обозначение на схеме



Описание

Когда команды объединены в блоки, которые нельзя запрограммировать только командами AND и OR, применяются AND LD и OR LD. В то время, как AND и OR служат 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 AND LD и OR LD нет необходимости применять, не нужны они и при прямом вводе РКС с SSS. Однако они требуются при преобразовании программы в мнемонику и вводе в данной форме.

Для сокращения числа команд, требующихся при программировании, требуется понимание блоковых команд. Введения в логические блоки см. 4-3-6.

Флаги

Флагов, на которые оказывают воздействие данные команды, нет.

5.8 Команды управления битами

Есть 7 команд, которые служат для управления состоянием индивидуального бита. Это OUT, OUT NOT, DIFU(13), DIFD(14), SET, RESET и KEEP(11). Данные команды служат для переключения бита в состояние 1 или 0 различными методами.

5.8.1 OUT и OUT NOT - Вывод и Вывод с инверсией

OUTPUT - OUT

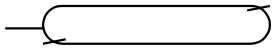
Обозначение на схеме

Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

OUTPUT NOT - OUT NOT

Обозначение на схеме



Область операндов

B	бит	IR, SR, AR, HR, TC, LR, TR

Ограничения

Любой выходной бит можно использовать только с одной командой, которая управляет его состоянием.

Описание

OUT и OUT NOT служат для управления состоянием заданного бита в соответствии с условием исполнения.

OUT включает указанный бит в 1, когда условие исполнения = 1 и устанавливает его в 0, когда условие исполнения = 0. С битом TR OUT появляется в точке ветвления, а не в конце командной линии. Подробности см. 4-3-8.

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 исполнением, устанавливая в 0 или 1 биты, которые заданы в качестве битов условия на РКС, 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.

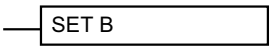
Промежутком времени, в котором бит находится в состоянии 0 или 1, можно управлять, комбинируя OUT и OUT NOT с TIM. Примеры см. 5-15-1.

Флаги

Флагов, на которые оказывают воздействие данные команды, нет.

5.8.2 SET и RESET (Установка и сброс)

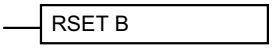
Обозначение на схеме



Область операндов

B	бит	IR, SR, AR, HR, LR

Обозначение на схеме



Область операндов

B	бит	IR, SR, AR, HR, 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.

Внимание!

Состояние битовых операндов команд SET и RESET, запрограммированных между IL(002) и ILC(003) или JMP(004) и JME(005), не изменяется, когда выполняется условие блока INTERLOCK или переходов (т.е. когда IL(002) или JMP(003) выполняется с условием исполнением 0).

Флаги

Флагов, на которые оказывают воздействие данные команды, нет.

Примеры

Следующие примеры демонстрируют разницу между OUT и SET/RESET. В первом примере (схема A) IR 10000 установится в 1 или 0 когда IR 00000 установится в 1 или 0.

Во втором примере (схема B) 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.

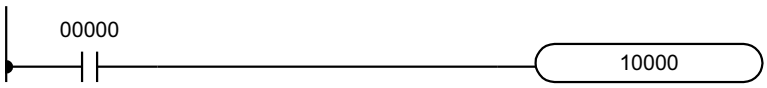


Диаграмма A

Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	10000

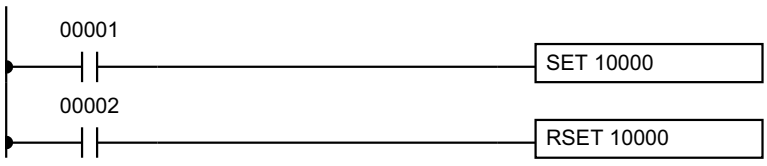


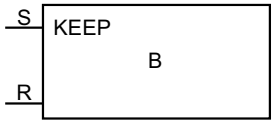
Диаграмма B

Адрес	Инструкция	Операнд
00000	LD	00001
00001	SET	10000

Адрес	Инструкция	Операнд
00002	LD	00002
00003	RSET	10000

5.8.3 KEEP(11) (KEEP) - Сохранить состояние

Обозначение на схеме



Область операндов

B	бит	IR, SR, AR, HR, LR

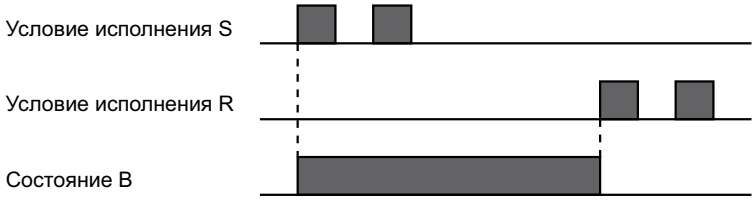
Ограничения

Любой выходной бит можно использовать только с одной командой, которая управляет его состоянием.

Описание

KEEP (11) служит для поддержания состояния заданного бита исходя из двух условий. Данные условия обозначаются S и R. S - вход установки в 1, R - вход сброса. KEEP (11) 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(11) и состоянием бита KEEP(11) показана на рисунке.

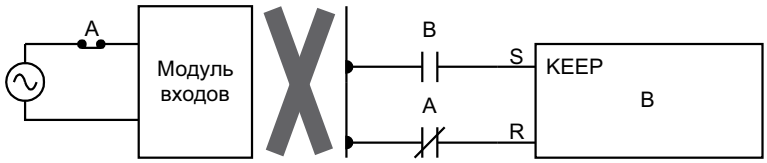


Флаги

Флагов, на которые оказывает воздействие данная команда, нет.

Предосторожности

Будьте осторожны, когда линией сброса KEEP(11) управляется нормально закрытое внешнее устройство. Не используйте входной бит с инверсным условием для линии сброса (R) KEEP(11), 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 KEEP(11). Ситуация показана ниже.



Биты, используемые с KEEP, не сбрасываются в INTERLOCK. Подробности см. 5-11.

5.8.4 DIFU(13) и DIFD(14) - установка бита на 1 цикл (после фронта 0/1 и 1/0)

Обозначение на схеме

DIFU(13) B

Область операндов

B	бит	IR, SR, AR, HR, LR

Обозначение на схеме

DIFD(14) B

Область операндов

B	бит	IR, SR, AR, HR, LR

Ограничения

Любой выходной бит можно использовать только с одной командой, которая управляет его состоянием.

Описание

DIFU(13) и DIFD(14) служат для установки указанного бита в 1 только на 1 цикл.

При выполнении DIFU(13) сравнивает текущее условие исполнения с предыдущим условием исполнения. Если предыдущее условие исполнения было = 0, а текущее условие исполнение = 1, DIFU(13) устанавливает в 1 заданный бит. Если предыдущее условие исполнения было = 1, то независимо от текущего состояния (0 или 1) DIFU(13) 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, предполагая, что команда выполняется каждый цикл (см. далее Внимание!). .

При выполнении DIFD(14) сравнивает текущее условие исполнения с предыдущим условием исполнения. Если предыдущее условие исполнения было = 1, а текущее = 0, DIFD(14) установит в 1 заданный бит. Если предыдущее условие исполнения было = 0, а текущее либо 0, либо 1, DIFD(14) 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, предполагая, что команда выполняется каждый цикл (см. далее Внимание!).

Данные команды используются в случае, если у команд версии фронта 0/1 (т.е. с префиксом @) 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

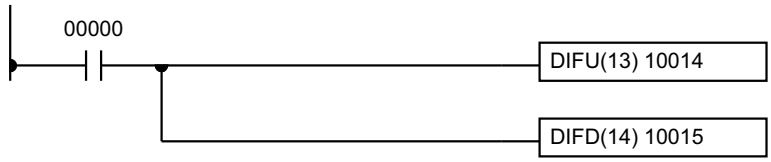
Флагов, на которые оказывают воздействие данные команды, нет.

Предосторожности

Операции DIFU(13) и DIFD(14) могут работать неустойчиво, когда команды запрограммированы между IL и ILC, между JMP и JME, или в подпрограммах. Смотрите 5-11, 5-12, 5-25, 5-28.

Пример

В этом примере IR 10014 установится в 1 на один цикл, когда IR 00000 переходит из 0 в 1. IR 10015 установится в 1 на один цикл, когда IR 00000 переходит из 1 в 0.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	DIFU(13)	10014
00002	DIFFD(14)	10015

5.9 NOP(00) (NO OPERATION) - Нет операции

Описание

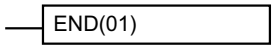
NOP(00) при написании РКС не требуется и не имеет символа для обозначения на РКС. Когда NOP(00) обнаруживается в программа, действий не производится и программа переходит к следующей команде. Когда память очищена перед программированием, во всех адресах записана команда NOP(00). Команда NOP(00) вводится функциональным кодом 00.

Флаги

Флагов, на которые оказывает воздействие NOP(00), нет.

5.10 END (01) (END) - Конец программы

Обозначение на схеме



Описание

END (01) требуется в качестве последней команды в любой программе. Если есть подпрограммы, END (01) помещается после последней подпрограммы. Команды, записанные после END (01), не выполняются. END (01) 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.

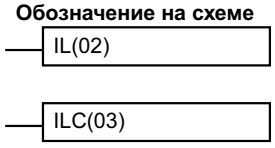
Если в программе отсутствует END (01), никакие команды не выполнятся и появится сообщение “NO END INST”.

Флаги

END (01) сбрасывает в 0 флаги ER, CY, GR, EQ и LE.

5.11 IL(02) и ILC(03) - Секция INTERLOCK и INTERLOCK CLEAR

Обозначение на схеме



Описание

IL(02) всегда используется совместно с ILC(03) для создания секции INTERLOCK. Секции INTERLOCK служат для разрешения ветвления таким же образом, как это достигается битами TR, но отработка команд между IL(02) и ILC(03) отличается от отработки с битами TR, когда условие исполнения для IL(02) = 0. Если условие исполнения для IL(02) = 1, программа будет выполняться как написано, когда это условие, равное 1, служит для пуска всех командных линий после того места, где расположена команда IL(02) до того места, где расположена ближайшая ILC(03). Базовые описания для обоих методов см. 4-3-8.

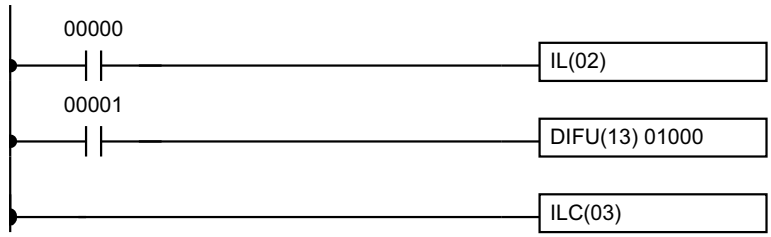
Если условие исполнения для IL(02) = 0, секция INTERLOCK между IL(02) и ILC(03) будет обрабатываться, как указано в следующей таблице.

Команда	**Обработка**
OUT и OUT NOT	Заданный бит устанавливается в 0
TIM и TIMH(15)	Сброс
CNT, CNTR(12)	Сохраняется текущее значение
KEEP(11)	Сохраняется состояние бита
DIFU(13) и DIFD(14)	Не выполняются (смотри ниже)
Все другие команды	Команды не выполняются, и все биты и слова IR, AR, LR, R и SR, записанные, как операнды команд, устанавливаются в 0.

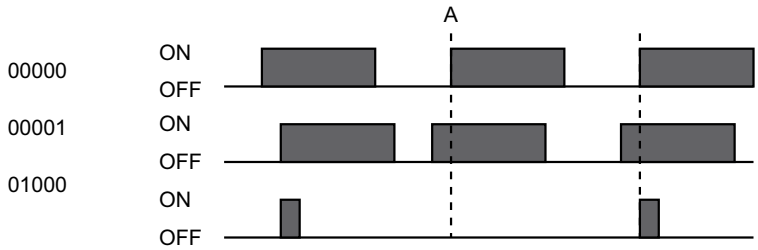
IL(02) и ILC(03) не обязательно использовать в паре. IL(02) можно использовать несколько раз, каждая IL(02) создает секцию INTERLOCK до ближайшей ILC(03). ILC(03) можно использовать только когда для нее имеется хотя бы одна IL(02) между ней и любой предыдущей ILC(03).

DIFU(13), DIFU(14) в секциях INTERLOCK

Изменение условий для DIFU(13) или DIFD(14) не запоминаются, если DIFU(13) или DIFD(14) находятся в секции INTERLOCK и условие исполнения для IL(02) = 0. Когда DIFU(13) или DIFD(14) выполняются в секции INTERLOCK сразу после установки условия исполнения для IL(02) в 1, условие для DIFU(13) или DIFD(14) будет сравниваться с тем условием исполнения, которое существовало до того, как секция INTERLOCK стала работать, (т.е. перед тем, как условие исполнения для IL(02) стало = 0). РКС и изменение состояния бита показаны на следующем рисунке. Секция INTERLOCK действует, пока 00000 = 0. Обратите внимание, что 01000 не включается в 1 в точке A, хотя 00001 стал = 0 и затем снова = 1.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL(02)	
00002	LD	00001
00003	DIFU(13)	01000
00005	ILC(03)	



Предосторожности

ILC(03) должно быть расположено после одной или нескольких IL(02).

Хотя с одним ILC(03) можно использовать столько IL(02), сколько необходимо, команды ILC(03) нельзя использовать друг за другом, без хотя бы одной IL(02) между ними, т.е. вложения невозможны. Когда выполнилась ILC(03), секция INTERLOCK между действующей ILC(03) и предыдущей ILC(03) очищается.

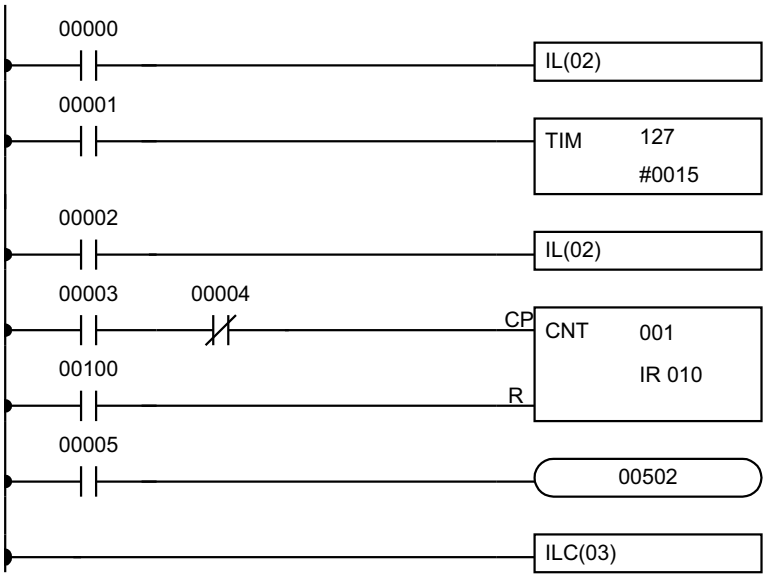
Когда с одной ILC(03) используется больше одной IL(02), появится сообщение об ошибке при проведении контроля программы, но исполнение будет проходить нормально.

Флаги

Флагов, на которые оказывают воздействие данные команды, нет.

Пример

На следующей схеме показано использование двух IL(02) с одним ILC(03).



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL(02)	
00002	LD	00001
00003	TIM	127
		# 0015
00004	LD	00002
00005	IL(02)	
00006	LD	00003
00007	AND NOT	00004
00008	LD	00100
00009	CNT	001
		010
00010	LD	00005
00011	OUT	00502
00012	ILC(03)	

Когда условие исполнения для первой IL(02) = 0, в TIM 127 будет занесено значение 1.5 с, CNT 001 не изменится, 00502 станет = 0. Когда условие исполнения для первой IL(02) = 1, а условие исполнения для второй IL(02) = 0, TIM 127 будет отрабатываться в зависимости от состояния 00001, CNT 001 не изменится, 00502 станет = 0. Когда условие исполнения для второй IL(02) = 1, программа будет исполняться, как написана.

5.12 JMP(04) и JME(05) - Переход и Конец перехода

Обозначение на схеме

JMP(04) N

Область операндов

N	номер	#

Обозначение на схеме

JME(05) N

Область операндов

N	номер	#

Ограничения

Номера переходов 01..99 (в CPM1 00..49) можно использовать только один раз в JMP(04) и JME(05), т.е. каждый может использоваться для определения только одного перехода. Переход с номером 00 можно использовать столько раз, сколько необходимо.

Номера переходов имеют значения 01..99 в CQM1 (в CPM1 00..49).

Описание

JMP(04) всегда применяется совместно с JME(05) для создания переходов, т.е. пропуска от одной до другой точки на РКС . JMP(04) определяет точку, с которой будет делаться переход. JME(05) определяет адрес, куда производится переход. Когда условие исполнения для JMP(04) = 1, перехода не происходит и программа выполняется последовательно, как написана (без пропусков). Когда условие исполнения для JMP(04) = 0, происходит переход к JME(05) с номером, таким же, как и у JMP(04), и далее выполняются команды, находящиеся после JME(05).

Если номер перехода для JMP(04) находится в диапазоне 01..99, при выполнении переходов сразу же следует переход к JME(05) с тем же номером, что и JMP(04) без выполнения каких-либо команд между ними. Состояние таймеров, счетчиков и битов, используемых в OUT и OUT NOT и всех других битов, которые управляются командами между JMP(04) и JME(05), не будет изменяться. Поскольку все команды между JMP(04) и JME(05) будут пропущены, номера переходов 01..99 (00..49 для CPM1) используются для сокращения времени цикла.

JMP(04) с номером 00

Если номер перехода для JMP(04) = 00, ЦУ будет искать ближайшую JME(05) 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 и OUT NOT, и всех других битов, которые контролируются командами между JMP(04) 00 и JME(05) 00 не будет изменяться. Переход JMP(04) 00 можно использовать столько раз, сколько необходимо. Переход JMP(04) 00 всегда переходит к ближайшему следующему JME(05) 00. Таким образом, можно использовать JMP(04) 00 последовательно и завершать их одним JME(05) 00. Однако нет смысла последовательно использовать JME(05) 00, поскольку все переходы окончатся на первом JME(05) 00.

DIFU(13) и DIFU(14) в переходах

Хотя DIFU(13) и DIFU(14) предназначены для установки в 1 указанные биты на один цикл, они могут этого не сделать, если записаны между JMP(04) и JME(05).

Как только либо DIFU(13), либо DIFU(14) установит бит в 1, он останется в 1 до следующего выполнения DIFU(13) или DIFU(14). При нормальном программировании, это значит в следующем цикле. При переходе это значит следующий раз, когда нет пропуска от JMP(04) к JME(05), т.е. если DIFU(13) или DIFU(14) установила бит в 1, а в последующих циклах происходит переход, так что DIFU(13) или DIFU(14) пропускаются, указанный бит останется в 1 до следующего раза, когда условие исполнения для JMP(04), управляемого переходом станет = 1.

Предосторожности

Когда JMP(04) и JME(05) используются не в паре, при контроле программы выдается сообщение об ошибке. Это сообщение появляется также, если JMP(04) 00 или JME(05) 00 используются не в паре, но программа будет выполнятся без переходов.

Флаги

Флагов, на которые оказывают воздействие данные команды, нет.

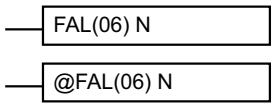
Примеры

Примеры программ с переходами приведены в п. 4-3-9.

5.13 FAL(06) и FALS(07) - Команды ошибок пользователя

(Предупреждение + сброс и предупреждения о фатальной ошибке)

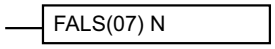
Обозначение на схеме



Область операндов

N	номер	# (00..99)

Обозначение на схеме



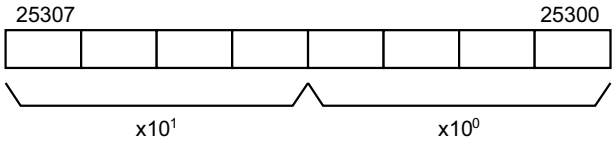
Область операндов

N	номер	# (00..99)

Описание

FAL(06) и FALS(07) служат для того, чтобы программист мог выдать номера ошибок для использования при отработке программы, обслуживании и наладке. Когда условие исполнения для этих команд = 1, они выдают номер FAL в биты 00..07 SR 253. Выдаваемый номер FAL может иметь значения 01..99 и является определителем команд FAL(06) и FALS(07). FAL(06) с определителем 00 используется для сброса данной области. (см. Ниже).

Область ошибок FAL



FAL(06) выдает признак нефатальной ошибки, FALS(07) выдает признак фатальной ошибки. Когда выполняется FAL(06) с условием 1, мигает индикатор ALARM/ERROR на передней панели ЦУ, но отработка программы ПК продолжается. Когда выполняется FALS(07) с условием 1, индикатор ALARM/ERROR на передней панели ЦУ горит, и отработка программы ПК прекращаются.

Система также генерирует коды ошибок в области FAL.

Сброс признаков ошибок

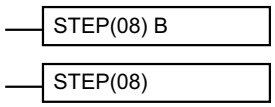
В памяти может храниться максимум 3 кода ошибок FAL, хотя только одна из них будет присутствовать в области FAL. Для доступа к другим кодам FAL нужно сбросить область FAL командой FAL(06) 00. Каждый раз после выполнения FAL(06) 00 код следующий ошибки будет посылаться в область FAL, стирая код той, которая была раньше.

FAL(06) 00 используется также для стирания сообщений, запрограммированных командой MSG(46).

Если область FAL нельзя очистить, что происходит после выполнения FALS(07), сначала устраните причину ошибки и далее очистите область FAL с помощью программатора или SSS.

5.14 STEP(08) / SNXT(09) - Команды STEP

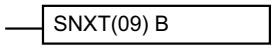
Обозначение на схеме



Область операндов

B	бит	IR, AR, HR, LR

Обозначение на схеме



Область операндов

B	бит	IR, AR, HR, LR

Ограничения

Все биты управления должны быть в одном и том же слове и должны располагаться последовательно.

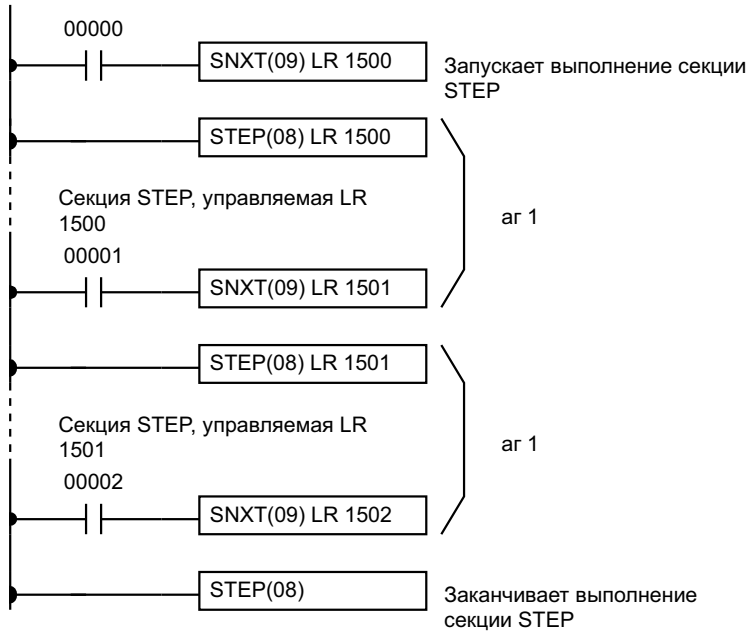
Описание

Команды STEP(08) и SNXT(09) используются совместно для задания точек прерывания между секциями в длинных программах, чтобы секция могла быть отработана как единый блок и сброшена после исполнения. Секция программы обычно определяется соответствующей фактическому процессу. (Примеры применения см. далее в данной главе). Команды секции STEP аналогичны обычным командам, за исключением того, что некоторые команды (END(01), FAL(06), FALS(07), JMP(04) /JME(05) и SBN(92)) нельзя включать в секцию.

STEP(08) использует бит управления в областях IR и HR для определения начала секции программы, называемой STEP. Команда STEP(08) не требует условий исполнения, т.е. ее исполнение управляется битом управления. Для пуска исполнения секции STEP служит SNXT(09) с тем же самым управляющим битом, что и в STEP(08). Если условие исполнения для SNXT(09) = 1, выполняется секция STEP с тем же самым битом управления. Если условие исполнения = 0, секция STEP не выполняется. Команда SNXT(09) 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(09) , не выполняется.

Когда в программе используется SNXT(09), исполнение секции STEP будет продолжаться, пока STEP(08) не выполнится без бита управления. Перед STEP(08) без управляющего бита должна стоять SNXT(09) с фиктивным управляющим битом. Фиктивным управляющим битом может служить любой неиспользуемый бит IR или HR. Им не должен быть бит, используемый STEP(08).

Исполнение секции STEP завершается либо исполнением следующей команды SNXT(09), 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(11) сохраняют свое состояние. В примере показаны две простых секции STEP.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	SNXT(09)	LR 1500
00002	STEP(08)	LR 1500
Секция STEP, управляемая LR 1500		
00100	LD	00001
00101	SNXT(09)	LR 1501
00102	STEP(08)	LR 1501
Секция STEP, управляемая LR 1501		
00200	LD	00002
00201	SNXT(09)	LR 1502
00202	STEP(08)	..

Секции STEP программируются последовательно. Каждая секция должна начинаться с STEP(08) и обычно заканчиваться SNXT(09) (см. далее пример 3, исключение). Когда секции программируюстя последовательно, возможны 3 режима исполнения: последовательный, с ветвлением или параллельный. Условия исполнения и позиция для SNXT(09) определяют, как будут исполняться секции. Три примера иллюстрируют три режима исполнения секций.

Предосторожности

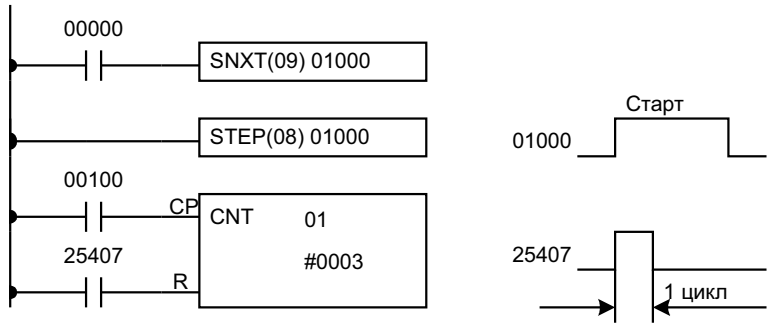
Команды секции INTERLOCK, переходы, SBN(92), END (01) нельзя использовать внутри секций STEP.

Биты, служащие управляющими битами, нельзя использовать больше нигде в программе, (если только они не используются для управления отработкой секции 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(08) и служит в случае необходимости для сброса счетчиков в секции STEP, как показано на следующем примере.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	SNXT(09)	01000
00002	STEP(08)	01000
00003	LD	00100
00004	LD	25407
00005	CNT	01
		# 0003

5.15 Команды таймеров и счетчиков

TIM и TIMH(15) являются командами декрементирующего таймера, включающегося в 1 с задержкой, которым требуются номера TC и заданное значение (SV). STIM(69) используется для управления интервальными таймерами, которые используются для вызова подпрограмм прерываний.

CNT - команда декрементирущего счетчика, а команда CNTR(12) - команда реверсивного счетчика. Обе команды требуют номер TC и заданного значения (SV). Обе подключены к нескольким командным линиям, которые служат в качестве входного сигнала (сигналов) и сброса. CTBL(63), INT(89) и PRV(62) используются для управления высокоскоростным счетчиком. INT(89) служит также для остановки выдачи импульсов.

Номер TC нельзя задать дважды, т.е. если он уже задан в качестве определителя в какой-либо команде таймера или счетчика, его нельзя использовать снова. Если номер TC задан, его можно использовать столько раз, сколько требуется, в качестве операнда в командах, кроме задания таймера и счетчиках.

В CQM1 номера TC имеют диапазон 000..511, в CPM1 диапазон 000..127. Префикс не требуется при использовании номера TC в качестве определителя в команде задания таймера или счетчика. Когда номер TC задан как таймер, его можно использовать с префиксом TIM для использования в качестве операнда в определенных командах. Префикс TIM применяется независимо от команды таймера, которой задавался таймер. Когда номер TC задан как счетчик, его можно использовать с префиксом CNT для использования в качестве операнда для определенных команд. Префикс CNT применяется также независимо от команды счетчика, которой задавался счетчик.

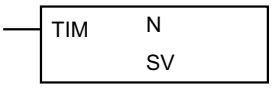
Номера TC можно задать как операнды, требующие либо битовых, либо словных данных. Когда номер TC задан в качестве операнда, требующего битовых данных, данный номер TC получает доступ к биту, выполняющему функцию “флаг завершения”, который указывает, что заданное время отработано/счет завершен, т.е. бит, нормально 0, включится в 1, когда заданное значение (SV) истечет. Когда номер TC задан в качестве операнда, требующего данных в виде слова, данный номер TC 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 CMP(20) или других команд, для которых разрешена область TC. Это делается назначением номера TC, служащего для задания этого таймера или счетчика, для доступа к области памяти, в которой хранится текущее значение (PV).

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 этого термина в контексте должно быть ясным, т.е. первое всегда - команда, второе всегда - битовый операнд и третье всегда словный операнд. То же самое верно для всех остальных номеров TC с префиксами TIM или CNT.

Задание (SV) может вводится константой или адресом слова в области данных. Если слово области IR, присвоенное блоку входов, задано в качестве адреса слова, блок входа можно подключить таким образом, чтобы задание (SV) можно было вводить извне с цифровых переключателей или других аналогичных устройств. 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.

5.15.1 TIM - таймер

Обозначение на схеме



Область операндов

N	номер таймера	#
SV	заданное значение (слово, BCD)	IR, SR, AR, HR, LR, DM, #

Ограничения

Заданные значения лежат в диапазоне 000.0..999.9. Десятичная точка не вводится.

Каждый номер TC можно использовать в качестве определителя только для одной команды таймера или счетчика. В CQM1 номера TC могут быть в диапазоне 000..511, в CPM1 в диапазоне 000..127.

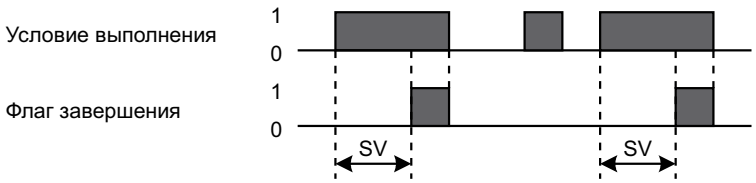
TC 000..TC 015 (TC 000..TC 003 в CPM1) рекомендуется использовать в команде задания TIM только если они требуются для команды TIMH(15). Подробности см. 5-15-4.

Описание

Таймер запускается, когда условие срабатывания устанавливается в 1 и сбрасывается (в заданное значение), когда условие срабатывания = 0. После запуска TIM отсчитывает время, вычитая по дискрете (в 0.1 с) от задания.

Если условие исполнения 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).

Следующий рисунок иллюстрирует соотношение между условием исполнения TIM и Флага завершения, связанного с ним.



Предупреждения

Таймеры в секциях INTERLOCK сбрасываются, когда условие исполнения IL(02) = 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. 5-15-2.

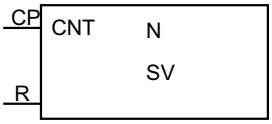
Флаги

ER: Заданные значения не в двоично-десятичном виде.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM).

5.15.2 CNT - Счетчик

Обозначение на схеме



Область операндов

N	номер счетчика	#
SV	заданное значение (слово, BCD)	IR, SR, AR, HR, LR, DM, #

Ограничения

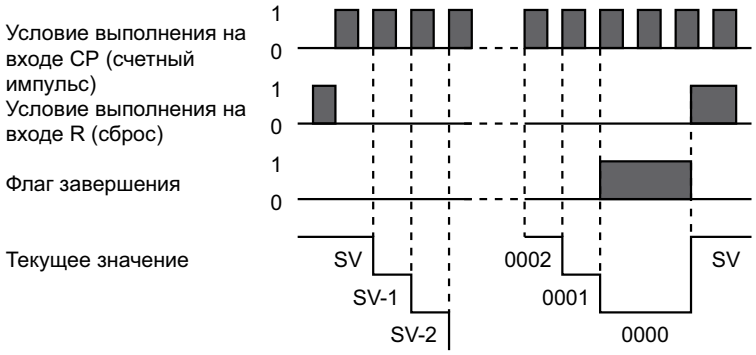
Каждый номер ТС можно использовать в качестве определителя только для одной команды таймера или счетчика. В CQM1 номера ТС задаются в диапазоне 000..511, в CPM1/CPM1A/SRM1 в диапазоне 000..127.

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 условии исполнения счетного выхода = 1 и условии исполнения в прошлом цикле = 0. Если условие исполнения не изменяется или изменяется из 1 в 0, текущее значение счетчика не изменяется. Флаг завершения счетчика устанавливается в 1, когда текущее значение достигает 0 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 сброшен в секциях INTERLOCK или при прерывании питания.

На следующем рисунке показаны изменения условий исполнения, 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 некорректным.

Флаги

ER: Задание не в двоично-десятичном виде.

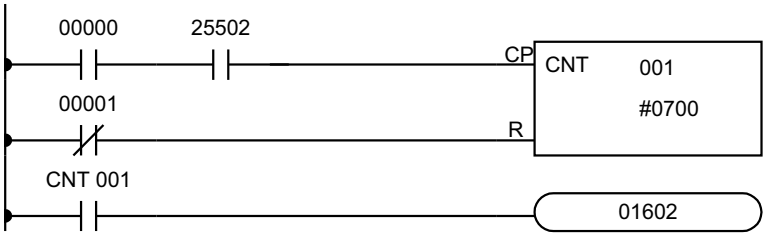
Косвенно адресуемое слова DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

Пример

В следующем примере CNT служит для создания расширенных таймеров путем счета битов импульсов часов в области SR.

CNT 001 считает, сколько раз бит часов частотой 1 сек (SR 25502) переключается из 0 в 1. Здесь IR 00000 используется для управления числом импульсов при работе CNT.

Поскольку в данном примере задание для CNT 001 = 700, флаг завершения для CNT 002 включается в 1, когда истекут 1 x 700 раз или 11 минут и 40 сек. Это приведет к включению IR 01602 в 1.

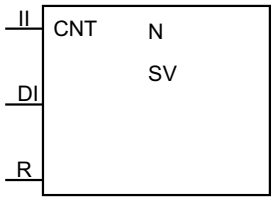


Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	25502
00002	LD NOT	00001
00003	CNT	001
		# 0700
00004	LD	CNT 001
00005	OUT	01602

Внимание! Очень короткие импульсы не обязательно вызовут точный отсчет времени, поскольку их короткое состояние 1 может не просчитаться во время длинных циклов. В частности, импульсы часов 0.02 с и 0.1 с не следует использовать для создания таймеров с помощью команд CNT.

5.15.3 CNTR(12) - Реверсивный счетчик

Обозначение на схеме



Область операндов

N	номер счетчика	#
SV	заданное значение (слово, BCD)	IR, SR, AR, HR, LR, DM, #

Ограничения

Каждый номер ТС можно использовать в качестве определителя только для одной команды таймера или счетчика. В CQM1 номера ТС задаются в диапазоне 000..511, в CPM1/CPM1A/SRM1 в диапазоне 000..127.

Описание

CNTR(12) - реверсивный, двухсторонний кольцевой счетчик, т.е. он служит для счета от 0 до задания (SV) в зависимости от изменений двух условий исполнения, на входе инкрементирования (II) и входе декрементирования (DI).

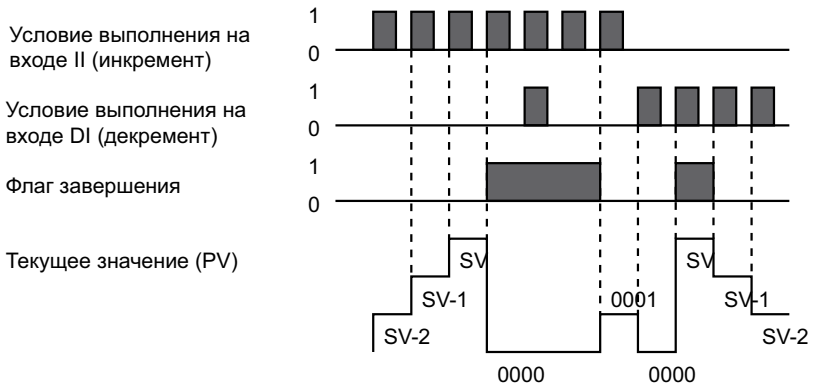
Текущее значение (PV) будет инкрементироваться на 1 при выполнении CNTR(12) с текущим условием исполнения входа II = 1, и прошлом значении II = 0. Текущее значение (PV) будет декрементироваться на 1 при выполнении CNTR(12) с текущим условием исполнения входа DI = 1, и прошлом значении DI = 0. Если условия исполнения на обоих входах изменились из 0 в 1, текущее значение счетчика не изменяется.

Если условия исполнения на обоих входах не изменяются или изменяются из 1 в 0, текущее значение счетчика не изменяется.

При декрементировании от 0000 текущее значение установится в заданное, а флаг завершения устанавливается в 1 до следующего декрементирования текущего значения. При инкрементировании выше задания текущее значение установится в 0000, а флаг завершения устанавливается в 1 до следующего инкрементирования.

CNTR(12) 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(12) не будет сброшен в секциях INTERLOCK или при прерывании питания.

На рисунке показаны условия исполнения на входах II и DI, Флаг завершения и текущее значение счетчика при работе счетчика (т. е. при сбросе, 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.

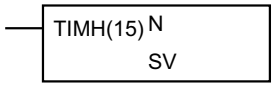
Флаги

ER: Задание не в двоично-десятичном виде.

Косвенно адресуемое слово DM не существуют. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

5.15.4 TIMH(15) - Высокоскоростной таймер

Обозначение на схеме



Область операндов

N	номер таймера	#
SV	заданное значение (слово, BCD)	IR, SR, AR, HR, LR, DM, #

Ограничения

Заданные значения лежат в диапазоне 00.00..99.99. (Хотя можно ввести 00.00 и 00.01, 00.00 запретит таймер, т.е. сразу включит флаг завершения, а 00.01 не будет надежно считан.) Десятичная точка не вводится.

Каждый номер TC можно использовать в качестве определителя только для одной команды таймера или счетчика. В CQM1 номера TC задаются в диапазоне 000..015, в CPM1/CPM1A/SRM1 в диапазоне 000..003.

Высокоскоростные таймеры с номерам 016..511 для CQM1 (003..127 для CPM1) не рекомендуется использовать, если время цикла превышает 10 мс.

Описание

TIMH(15) работает так же, как и TIM, за исключением того, что дискрета TIMH(15)..0.01 с. Подробности см. 5-15-1.

Предосторожности

Таймеры в секциях программы INTERLOCK сбрасываются, когда условия для IL(02) = 0. Таймеры также сбрасываются при сбросе питания. Если нужен таймер, который не сбрасывался бы при таких условиях, биты таймерных импульсов времени из области SR можно подсчитывать для создания таймеров, использующих CNT. Подробности см. 5-15-2.

Таймеры в секциях с командами перехода не будут сбрасываться при условии исполнения JMP(04) = 0, но таймер остановит отсчет времени, если используется номер перехода 00. Таймеры будут продолжать отсчет времени, если используются номера переходов 01..99 (для CPM1 01..49).

Меры предосторожности в CQM1

Высокоскоростные таймеры с номерами TC 000..TC 015 не будут работать некорректно, если установочный параметр (DM 6629) установлен на выполнение обработки прерываний этих таймеров.

Высокоскоростные таймеры с номерами TC 016..TC 511 будут работать неточно, если время цикла превышает 10 мс. Если время цикла больше 10 мс, используйте TC 000..TC 015 и задавайте DM 6629 для обработки прерываний используемых номеров таймеров.

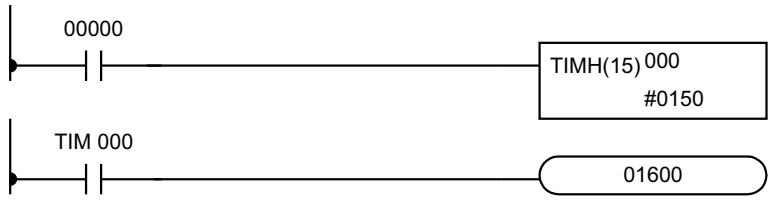
Флаги

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

В следующем примере показан таймер с заданием, установленным как константа. 01600 станет = 1 после того, как 00000 станет = 1 и останется в состоянии 1 как минимум на 1.5 с. Когда 00000 станет = 0, таймер сбросится и 01600 установится в 0.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	TIMH(15)	000
		# 0150
00002	LD	TIM 000
00003	OUT	01600

5.15.5 STIM(69) - Интервальный таймер

Обозначение на схеме

STIM(69)
C1
C2
C3

@STIM(69)
C1
C2
C3

Область операндов

C1	данные управления #1	000..008, 010..012
C2	данные управления #2	IR, SR, AR, HR, TC, LR, DM, #
C3	данные управления #3	IR, SR, AR, HR, TC, LR, DM, #

Замечание STIM(69) - дополнительная команда для SRM1. Номер инструкции 69 установлен по умолчанию и может быть изменен пользователем.

Ограничения CQM1

- C1 должен быть 000..008 или 010..012
- Если C1 = 000..005, для C3 нельзя использовать константу, большую чем 0255
- Если C1=006..008, для C2 и C3 нельзя задавать константы и DM 6143..DM 6655.
- Если C1 = 010..012, и C2 и C3 должны быть = 000.

Ограничения CPM1/CPM1A/SRM1

- C1 должен быть 000, 003, 006 или 010.
- Если C1 = 003, для C3 нельзя использовать константу, большую чем 0049

- Если C1 = 006 для C2 и C3 нельзя задавать константы и DM 6143..DM 6655.
- Если C1 = 010 и C2 и C3 должны быть = 000.

Описание

STIM(69) используется для управления интервальными таймерами путем выполнения четырех базовых команд:

- пуска таймера для разовых прерываний,
- пуска таймера для прерываний по расписанию,
- останова таймера,
- чтения текущего значения таймера.

Задавайте C1 для указания, какие их этих функций будут выполняться и какой из трех интервальных таймеров будет запущен. (см. таблицу). Более подробное описание применения прерываний интервального таймера см. 1-4-4. STIM(69) описан также далее после таблицы.

Функция	Таймер	Задание	ПК
Пуск таймеров	0	000	CQM1/CPM1/CPM1A/ SRM1
	1	001	Только CQM1
	2	002	
Пуск прерываний по расписанию	0	003	CQM1/CPM1/CPM1A/ SRM1
	1	004	Только CQM1
	2	005	
Чтение текущего значения таймера	0	006	CQM1/CPM1/CPM1A/ SRM1
	1	007	Только CQM1
	2	008	
Остановка таймеров	0	010	CQM1/CPM1/CPM1A/ SRM1
	1	011	Только CQM1
	2	012	

Замечание 1. В CQM1 интервальный таймер 0 нельзя применять, когда выдача импульсов производится командой SPED(64).

2. В CPM1 интервальный таймер 2 нельзя применять, когда в установочном параметре DM разрешена работа высокоскоростного счетчика 0.

Пуск прерываний

Установите C1 = 000..002 для пуска таймеров 0..2 в режиме разовых прерываний. Установите C1 = 003..005 для пуска прерываний по расписанию, которые используют таймеры 0..2.

C2, определяющая задание таймера, может быть константой или адресом первого из двух слов, содержащих задание. Значения C2 будет различаться в зависимости от используемых методов.

Если C2 - константа, она задает начальное значение декрементирующего счетчика (двоично-десятичные числа 0000..9999). Интервал декрементирования - 1 мс.

Если C2 - адрес слова, C2 задает начальное значение декрементирующего счетчика (двоично-десятичные числа 0000..9999), C2+1 задет интервал для декрементирования времени (двоично-десятичные числа 0000..9999) в 0.1 мс. Временной интервал может быть 0.5..32 мс.

C3 задает номер подпрограммы 0000..0255 (0000..0127 в CQM1-CPU11/21-E, 0000..0049 в CPM1/CPM1A).

Замечание Время, требующееся от пуска интервального времени до окончания отработки:

(содержание C2) x (содержание C2+1) x 0.1 мс.

Чтение текущих значений таймеров

Установите C1 = 006..008 для чтения текущего значения таймеров 0..2

C2 задает адрес первого из двух слов, в которое будет приниматься текущее значение. C2 будет принимать количество интервалов, которые декреметнировал таймер (двоично-десятичное число 0000..9999), а в C2+1 будет принимать интервал времени декрементирования (двоично-десятичное число, в 0.1 мс.).

C3 задает слово назначения, которое будет принимать время, истекшее с последнего декрементирования таймера (двоично-десятичное число, в 0.1 мс.). (Должно быть меньше либо равно интервалу времени декрементирования, заданному в C2+1).

Замечание Время, истекшее с момента пуска таймера, можно подсчитать следующим образом:

[(содержание C2) x (содержание C2+1)+ (содержание C3)] x 0.1 мс.

Остановка таймеров

Установите C1 = 010..012 для останова таймеров 0..2

C2 и C3 не имеют функций и должны быть заданы = 000.

Флаги

ER: Интервальный таймер 0 запущен во время выдачи импульсов. (C1 = только 000).

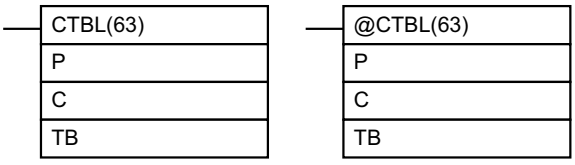
Интервальный таймер 2 запущен во время работы высокоскоростного счетчика 0 (C1 = только 002).

Косвенно адресуемые слова DM не существуют. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM).

Превышена граница области данных.

5.15.6 CTBL(63) - Зарегистрировать таблицу сравнения

Обозначение на схеме



Область операндов

P	указатель порта	IR, SR, AR, HR, TC, LR, TR, DM
C	данные управления	IR, SR, AR, HR, TC, LR, TR, DM
TB	первое слово таблицы сравнения	IR, SR, AR, HR, TC, LR, TR, DM

Замечание Инструкция не используется в SRM1

Ограничения

Первое и последнее слова таблицы сравнения должны лежать в одной области данных. (Длина таблицы сравнения изменяется в зависимости от установочных параметров).

В CQM1-CPU43-EV1 CTBL(63) нельзя использовать, если установочным параметром (DM 6611) задает режим выдачи импульсов.

В CPM1/CPM1A P должно быть 000.

Описание

Когда условие исполнения = 0, CTBL(63) не исполняется. Когда условие исполнения = 1, CTBL(63) регистрирует таблицу сравнения для использования с текущим значением высокоскоростного счетчика. В зависимости от величины C сравнение с текущим значением высокоскоростного счетчика может начаться сразу или может начаться после команды INI(61).

Определитель порта (P) задает высокоскоростной счетчик, который будет использоваться при сравнении.

P	Функция	ПК
000	Задает высокоскоростной счетчик 0	CQM1 и CPM1

P	Функция	ПК
001	Задает высокоскоростной счетчик 1	Только CQM1
002	Задает высокоскоростной счетчик 2	

Функция CTBL(63) задается параметром управления, C, как показано в следующей таблице. Данные функции описаны после таблицы.

C	Функция CTBL(63)
000	Регистрирует таблицу сравнения по значению и начинает сравнение
001	Регистрирует таблицу сравнения с зоной и начинает сравнение
002	Регистрирует таблицу сравнения по значению. Сравнение начинается после INI(61).
003	Регистрирует таблицу сравнения с зоной. Сравнение начинается после INI(61).

Когда текущее значение достигает заданного значения или входит в заданную зону, вызывается и исполняется заданная подпрограмма. Подробности о таблице сравнения см. 1-4-5 (CQM1) или 1-5-5 (CPM1/CPM1A).

Если высокоскоростной счетчик разрешен в установочном параметре (DM 6642), он начинает считать от нуля, когда CQM1 начинает операцию. Текущее значение не будет сравниваться с таблицей сравнения только когда таблица зарегистрирована и сравнение вызвано командой INI(61) или CTBL(63). Сравнение можно остановить или запустить, или текущее значение можно сбросить командой INI(61).

Когда таблица сравнения зарегистрирована, она действует либо до остановки CQM1, либо до ошибки при попытке зарегистрировать новую таблицу. Для сокращения времени цикла рекомендуется, если возможно, версия @ команды CTBL(63) (“по фронту 0/1”).

Сравнение с заданным значением

Таблица сравнений с заданным значением содержит до 16 заданных значений и номеров подпрограмм, соответствующих каждому заданному значению. Для CQM1-CPU4_-EV1 таблица может содержать до 48 заданных значений. Соответствующая подпрограмма вызывается и выполняется, когда текущее значение становится равным заданному значению. (Когда не требуется обработка прерываний, можно ввести неопределенный номер подпрограммы).

Сравнения по заданному значению выполняются одно значения за раз в порядке размещения в таблице. Когда текущее значение достигнет первого заданного значения в таблице, выполняется подпрограмма прерываний и сравнение продолжается со следующим значением в таблице. Когда завершится обработка последнего конечного значения в таблице, происходит возврат к значению в таблице и процесс повторяется.

На следующем рисунке показана структура таблицы сравнения по конечному значению для использования с высокоскоростным счетчиком 0, или высокоскоростными счетчиками 1 и 2 в линейном режиме.

TB	Число заданных значений (двоично-десятичное число)	0001..0016
TB+1	Заданное значение #1, младшие 4 цифры (двоично-десятичные)	Один блок заданий
TB+2	Заданное значение #1, старшие 4 цифры (двоично-десятичные)	
TB+3	Номер подпрограммы (см. прим.)	
...		

В следующей таблице показана структура таблицы сравнения по заданному значению для использования с высокоскоростными счетчиками 1 и 2 в кольцевом режиме. Вводите конечные значения в возрастающем или убывающем порядке.

Величина кольца задает число точек на кольце и максимальное значение счета (величина кольца = макс. значение счета + 1). Не изменяйте величину кольца, когда идет сравнение.

ТВ	Величина кольца, младшие 4 цифры(двоично-десятичное число)	Задание величины кольца
ТВ+1	Величина кольца, старшие 4 цифры(двоично-десятичное число)	
ТВ+2	Число заданных значений (двоично-десятичное число)	0001..0016
ТВ+3	Заданное значение #1, младшие 4 цифры (двоично-десятичные)	Один блок заданий
ТВ+4	Заданное значение #1, старшие 4 цифры (двоично-десятичные)	
ТВ+5	Номер подпрограммы (см. прим.)	
...		

На следующем рисунке показана структура таблицы сравнения с заданным значением для использования с абсолютными высокоскоростными счетчиками 1 и 2 (только CQM1-CPU44-EV1). Вводите заданные значения в возрастающем или убывающем порядке.

ТВ	Число заданных значений (двоично-десятичное число)	0001..0016
ТВ+1	Заданное значение #1 (двоично-десятичное число)	Один блок заданий
ТВ+2	Номер подпрограммы (см. прим.)	
...		

Замечание 1. Номер подпрограммы может быть 0000..0049 для CPM1.

В CQM1 номера подпрограмм могут быть F000..F255 (F000..F127 для CQM1-CPU11/21-EV1) для вызова подпрограмм при декрементировании 0000..0255 (000..127 для CQM1-CPU11/21-EV1) для вызова подпрограмм при инкрементировании

2. Оставьте интервал как минимум 0.2 мс для обработки прерываний при задании заданных значений для высокоскоростных счетчиков 1 и 2.

Сравнение с зоной

Таблица сравнений с зоной содержит 8 зон, которые задаются нижней границей (8 цифр) и верхней границей (8 цифр) и соответствующие номера подпрограмм. Соответствующая подпрограмма вызывается и выполняется, когда текущее значение попадает в зону. (Когда не требуется обработка прерываний, можно ввести неопределенный номер подпрограммы).

Всегда задавайте 8 зон. Если требуется менее 8 зон, задайте остальные номера подпрограмм FFFF. Если требуется более 8 зон, другая команда сравнения, такая как BCMP(68), используется для сравнения зон со значением высокоскоростного счетчика в IR 230..IR 235 (SR 248..SR 249 в CPM1). Помните, что эти слова обновляются один раз каждый цикл.

В области AR имеются флаги, которые показывают, когда текущие значения находятся в одной или более из 8 зон. Флаги = 1, когда текущие значения находятся в соответствующей зоне.

Счетчик	**Флаги области AR**
Высокоскоростной счетчик 0	AR 1100 .. AR 1107 соответствуют зонам 1 .. 8
Высокоскоростной счетчик 1	AR 0500 .. AR 0507 соответствуют зонам 1 .. 8
Высокоскоростной счетчик 2	AR 0600 .. AR 0607 соответствуют зонам 1 .. 8

Замечание У CPM1 есть только высокоскоростной счетчик 0.

На следующем рисунке показана структура таблицы сравнения с зоной для использования с высокоскоростным счетчиком 0, или высокоскоростными счетчиками 1 и 2 в линейном режиме.

ТВ	Нижняя граница #1, младшие 4 цифры (двоично-десятичные)	Задание первого блока
ТВ+1	Нижняя граница #1, старшие 4 цифры (двоично-десятичные)	
ТВ+2	Верхняя граница #1, младшие 4 цифры (двоично-десятичные)	
ТВ+3	Верхняя граница #1, старшие 4 цифры (двоично-десятичные)	
ТВ+4	Номер подпрограммы (см. прим. 1)	
...		
ТВ+35	Нижняя граница #8, младшие 4 цифры (двоично-десятичные)	Задание восьмого блока
ТВ+36	Нижняя граница #8, старшие 4 цифры (двоично-десятичные)	
ТВ+37	Верхняя граница #8, младшие 4 цифры (двоично-десятичные)	
ТВ+38	Верхняя граница #8, старшие 4 цифры (двоично-десятичные)	
ТВ+39	Номер подпрограммы (см. прим. 1)	

В следующей таблице показана структура таблицы сравнения с зоной для использования с высокоскоростными счетчиками 1 и 2 в кольцевом режиме (только CQM1). Величина кольца задает число точек на кольце и максимальное значение счета (Величина кольца = макс. значение счета + 1). Не изменяйте величину кольца , когда идет сравнение.

ТВ	Величина кольца, младшие 4 цифры(двоично-десятичное число)	Задание величины кольца
ТВ+1	Величина кольца, старшие 4 цифры(двоично-десятичное число)	
ТВ+3	Нижняя граница #1, младшие 4 цифры (двоично-десятичные)	
ТВ+4	Нижняя граница #1, старшие 4 цифры (двоично-десятичные)	Задание первой зоны
ТВ+5	Верхняя граница #1, младшие 4 цифры (двоично-десятичные)	
ТВ+6	Верхняя граница #1, старшие 4 цифры (двоично-десятичные)	
ТВ+7	Номер подпрограммы (см. прим. 1)	
...		
ТВ+37	Нижняя граница #8, младшие 4 цифры (двоично-десятичные)	Задание восьмой зоны
ТВ+38	Нижняя граница #8, старшие 4 цифры (двоично-десятичные)	
ТВ+38	Верхняя граница #8, младшие 4 цифры (двоично-десятичные)	
ТВ+40	Верхняя граница #8, старшие 4 цифры (двоично-десятичные)	
ТВ+41	Номер подпрограммы (см. прим. 1)	

На следующем рисунке показана структура таблицы сравнения с зоной для использования с абсолютными высокоскоростными счетчиками 1 и 2 (только CQM1-CPU44-EV1)..

ТВ	Нижняя граница #1, (двоично-десятичные)	Задание первой зоны
ТВ+2	Верхняя граница #1, (двоично-десятичные)	
ТВ+4	Номер подпрограммы (см. прим. 2)	
...		
ТВ+21	Нижняя граница #8 (двоично-десятичные)	Задание восьмой зоны
ТВ+22	Верхняя граница #8, (двоично-десятичные)	
ТВ+23	Номер подпрограммы (см. прим. 2)	

Замечание 1. Номер подпрограммы может быть 0000..0255 (0000..0127 для CQM1-CPU11/21-E, 0000..0049 для CPM1) и подпрограмма будет выполняться, пока текущее значения счетчиков находятся в заданной зоне. Значение FFFF указывает на то, что программу не надо выполнять.

2. Номер подпрограммы может быть 0000..0255 (0000..0127 для CQM1-CPU11/21-E, 0000..0049 для CPM1) для вызова подпрограммы при инкрементировании.

3. Оставляйте интервал как минимум 2 мс между верхней и нижней границей (верхняя граница - нижняя граница 0002 x частота входных импульсов) при задании сравнений для высокоскоростных счетчиков 1 и 2.

В следующей таблице показаны возможные значения заданных значений, нижней и верхней границ зон. 16-ричная цифра F в старшем разряде показывает, что значение отрицательно.

Счетчик	**Возможные значения**
Высокоскоростной счетчик 0	Режим плюс/минус: F003 2767..0003 2767 Инкрементальный: 0000 0000..0006 5535
Высокоскоростные счетчики 1 и 2	Линейный режим: F838 8607..0838 8607 Кольцевой: 0000 0000..0006 4999
Абсолютные Высокоскоростные счетчики 1 и 2	Двоично-десятичный режим: 0000..4095 Режим 360 ^{0}: 0000..0355 (дискрета 5 ^{0})

В режиме 360^0 угловое значение абсолютного высокоскоростного счетчика внутренне преобразуется в двоичное значение. Двоичное значение после преобразования зависит от разрешающей способности, выбранной в установочных параметрах DM 6643 и/или DM 6644. В таблице представлены преобразованные значения для 5^0..45^0.

Разрешающая способность	**Преобразованное значение**								
	5^0	**10^0**	**15^0**	**20^0**	**25^0**	**30^0**	**35^0**	**40^0**	**45^0**
8 бит (0..255)	4	7	11	14	18	21	25	28	32
10 бит (0..1023)	14	28	43	57	71	85	100	114	128
12 бит (0..4095)	57	114	171	228	284	341	398	455	512

Для больших значений найдите преобразованное значение к ближайшим 45^0 и добавьте остаток из таблицы. Например, для преобразования 145^0 с разрешением 8-бит: 32 x 3 (для 135^0) + 7 (для 10^0) = 103.

Внимание! С разрешением 10 и 12 бит обработка прерываний может не переключиться из-за того, что преобразованное угловое значение может точно не соответствовать величине в таблице сравнения.

Флаги

ER: Ошибка в установочных параметрах высокоскоростного счетчика.

Указанный порт и функция несовместимы.

В подпрограмме, вызванной CTBL(63) , имеется еще одна CTBL(63).

Во время сравнения выполняется CTBL(63), использующая другой формат сравнения.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

Таблица сравнений превысила границы области памяти или ошибка в задании таблицы сравнений.

CTBL(63) выполняется в подпрограмме прерываний когда в главной программе выполняется ввод/вывод импульсов или команда высокоскоростного счетчика.

AR 05: Флаги AR 0500..AR 0507 включаются в 1 для указания, когда текущее значение высокоскоростного счетчика 1 находится в зонах 1..8.

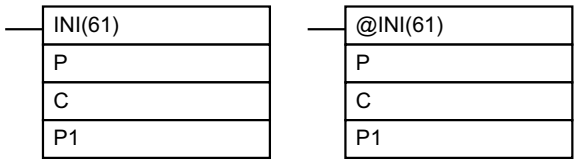
AR 06: Флаги AR 0600..AR 0607 включаются в 1 для указания, когда текущее значение высокоскоростного счетчика 2 находится в зонах 1..8.

AR 11: Флаги AR 1100..AR 1107 включаются в 1 для указания, когда текущее значение высокоскоростного счетчика 1 находится в зонах 1..8.

Подпрограммы выполняются только раз, когда условия их вызова выполнены. Состояние AR обновляется только раз за цикл. Если выполняются условия более одного блока таблицы, приоритетом пользуется первое.

5.15.7 INI(61) - Управление режимами выдачи импульсов

Обозначение на схеме



Область операндов

P	указатель порта	000, 001, 002
C	параметр управления	000..003
P1	первое слово текущего значения	IR, SR, AR, HR, LR, DM

Замечание Инструкция не используется в SRM1

Ограничения

В CPM1/CPM1A P должно быть 000 и C должно быть 000..003.

В CQM1 P должно быть 000, 001 или 002 и C должно быть 000..003.

P1 должно быть 000, если только C не 0002.

P1 и P1+1 должны лежать в одной области.

DM 6143..DM 6655 нельзя использовать для P1.

Описание

Когда условие исполнения = 0, INI(61) не исполняется. Когда условие исполнения = 1, INI(61) используется для управления работой высокоскоростного счетчика и остановки выдачи импульсов.

Определитель порта (P) указывает высокоскоростной счетчик или импульсный выход, над которыми будет производиться управление.

P	**Функция**
000	Задает высокоскоростной счетчик 0 или выдачу импульсов с бита простого выхода
001	Задает высокоскоростной счетчик 1 или выдачу импульсов с порта 1
002	Задает высокоскоростной счетчик 2 или выдачу импульсов с порта 2

Замечание В CPM1 есть только высокоскоростной счетчик 0.

Функция INI(61) определяется параметром управления, C.(В P1 и P1+1 содержится новое текущее значение высокоскоростного счетчика при изменении текущего значения).

C	P1	Функция INI(61)
000	000	Запускает сравнение таблицы CTBL(63)
001	000	Останавливает сравнение таблицы CTBL(63)
002	Новое текущее значение высокоскоростного счетчика	Изменяет текущее значение высокоскоростного счетчика
003	000	Останавливает выдачу импульсов

Сравнение с таблицей CTBL(63)

Если C = 000 или 001, INI(61) запускает или останавливает сравнение текущего значения высокоскоростного счетчика с таблице сравнения, зарегистрированной командой CTBL(63). Подробности о таблице сравнения см. 1-4-5 (CQM1) или 1-5-5 (CPM1).

Изменение текущего значения

Если C = 002, INI(61) изменяет текущее значение высокоскоростного счетчика на значение в P1 и P1+1.

У высокоскоростного счетчика 0 текущее значение может быть в диапазоне F003 2767..0003 2767 в режиме плюс/минус или 0000 0000..0006 5535 в инкрементальном режиме. 16-ричная цифра F в старшем разряде указывает, что текущее значение отрицательно.

4 левых цифры	4 правых цифры	Режим плюс/минус	Режим инкремент
P1+1	P1	F003 2767..0003 2767	0000 0000..0006 5535

У высокоскоростных счетчиков 1 и 2 (только у CQM1), текущее значение может быть в диапазоне F838 8607..0838 8607 в линейном режиме или 0000 0000..0006 4999 в кольцевом режиме. 16-ричная цифра F в старшем разряде указывает, что текущее значение отрицательно.

4 левых цифры	4 правых цифры	Линейный метод	Кольцевой метод
P1+1	P1	F838 8607..0838 8607 (-8 388 607..8 388 607)	0000 0000..0006 4999

Замечание Текущее значение абсолютных высокоскоростных счетчиков 1 и 2 (только CQM1) не может быть изменено.

Остановка выдачи импульсов

Если C = 003, INI(61) останавливает выдачу импульсов. Подробности о выдачи импульсов см 1-3.

Флаги

ER: Указанный порт и функция несовместимы.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

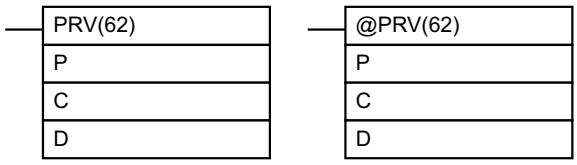
P1+1 превышает границы области данных (C=002).

Ошибка в задании операнда.

INI(61) выполняется в подпрограмме прерываний когда в главной программе выполняется вывод импульсов или команда высокоскоростного счетчика.

5.15.8 PRV(62) - Чтение текущего значения высокоскоростного счетчика

Обозначение на схеме



Область операндов

P	указатель порта	000, 001, 002
C	параметр управления	000..002
D	первое слово приемника	IR, SR, AR, HR, LR, DM

Замечание Инструкция не используется в SRM1

Ограничения

В CPM1/CPM1A P и C должны быть 000.

В CQM1 P и C должно быть 000, 001 или 002.

D1 и D1+1 должны лежать в одной области

DM 6143..DM 6655 нельзя использовать для D

Описание

Когда условие исполнения = 0, PRV(62) не исполняется. Когда условие исполнения = 1, PRV(62) читает данные, указанные P и C, и записывает их в D или в D и D+1

Определитель порта (P) задает высокоскоростный счетчик или импульсный выход.

P	Функция
000	Задает высокоскоростной счетчик 0 или выдачу импульсов с бита простого выхода
001	Задает высокоскоростной счетчик 1 или выдачу импульсов с порта 1
002	Задает высокоскоростной счетчик 2 или выдачу импульсов с порта 2

Замечание В CPM1/CPM1A есть только высокоскоростный счетчик 0.

Параметр управления C определяет, к какому типу данных будет доступ.

C	Данные	Слово(а) приемника
000	Текущее значение высокоскоростного счетчика	D и D+1
001	Состояние высокоскоростного счетчика или выдачи импульсов	D
002	Результаты сравнения по зоне	D

Текущее значение высокоскоростного счетчика (C=000)

Если C = 000, PRV(62) читает текущее значение указанного высокоскоростного счетчика и записывает 8-разрядное значение по адресу D и D+1.

У высокоскоростного счетчика 0 текущее значение может быть в диапазоне F003 2767..0003 2767 в режиме плюс/минус или 0000 0000..0006 5535 в инкрементальном режиме. 16-ричная цифра F в старшем разряде указывает, что текущее значение отрицательно.

4 левых цифры	4 правых цифры	Режим плюс/минус	Режим инкремент
D+1	D	F003 2767..0003 2767	0000 0000..0006 5535

У высокоскоростных счетчиков 1 и 2 (только у CQM1), текущее значение может быть F838 8607..0838 8607 в линейном режиме или 0000 0000..0006 4999 в кольцевом режиме. 16-ричная цифра F в старшем разряде указывает, что текущее значение отрицательно.

4 левых цифры	4 правых цифры	Линейный метод	Кольцевой метод
D+1	D	F838 8607..0838 8607 (-8 388 607..8 388 607)	0000 0000 до 0006 4999

У абсолютных высокоскоростных счетчиков 1 и 2 текущее значение может быть 0000 0000..0000 4095 при двоично-десятичном методе и 0000 0000..0000 0359 при методе 360^0.

4 левых цифры	4 правых цифры	Двоично-десятичный метод	Метод 360^0
D+1	D	0000 0000..0000 4095	0000 0000..0000 0359

Состояние высокоскоростного счетчика или выдачи импульсов

Если C = 001 (только CQM1), PRV(62) читает рабочее состояние указанного высокоскоростного счетчика или выдачи импульсов и записывает данные в D.

В следующей таблице показана функция битов в D для высокоскоростных счетчиков 1 и 2 и выдачи импульсов с портов 1 и 2 (только CQM1-CPU43-E). Биты, не перечисленные в таблице, не используются и всегда = 0.

Бит	Функция
00	Состояние сравнения высокоскоростного счетчика 0: Остановлено 1: Сравнение
01	Переполнение+/переполнение - высокоскоростного счетчика 0: Нормально 1: Произошло переполнение+/переполнение -
04	Увеличение частоты импульсов 0: Не задано 1: Задано
05	Общее число импульсов 0: Не задано 1: Задано
06	Выдача импульсов завершена 0: Не завершена 1: Завершена

Бит	Функция
07	Состояние выдачи импульсов 0: Остановлена 1: Выдаются

Для абсолютных высокоскоростных счетчиков 1 и 2 (только CQM1-CPU43-EV1) бит 00 D указывает на состояние сравнения (0: Остановлено, 1: Идет сравнение). Другие биты в D (0..15) не используются и всегда = 0.

Замечание Данные флаги имеются в AR 05 и AR 06, но эти слова нормально обновляются только раз за цикл, так что данные, полученные PRV(62) будут новее.

Результаты сравнения с зоной (C=002)

Если C = 002 (только CQM1), PRV(62) читает результаты сравнения текущих значений с 8 зонами, заданными CTBL(63) и записывает эти..07 слова D содержат флаги результатов сравнения зон 1..8 (0: не в зоне, 1: в зоне).

Замечание Данные флаги имеются в AR 05 и AR 06, но эти слова нормально обновляются только раз за цикл, так что данные, полученные PRV(62), будут новее.

Флаги

ER: Указанный порт и функция несовместимы.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

D+1 превышает границы области данных. (C=000)

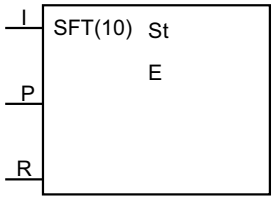
Ошибка в задании операнда.

PRV(62) выполняется в подпрограмме прерываний в то время, когда в главной программе выполняется вывод импульсов или команда высокоскоростного счетчика.

5.16 Команды сдвига

5.16.1 SFT(10) - Сдвиговый регистр

Обозначение на схеме



Область операндов

St	первое слово	IR, SR, AR, HR, LR
E	последнее слово	IR, SR, AR, HR, LR

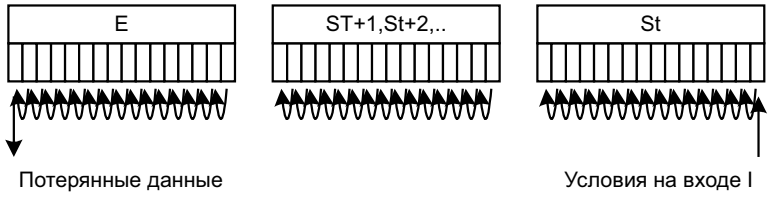
Ограничения

E должно быть больше либо равно St, а St и E должны быть в одной области данных.

Если адрес бита в одном из слов, используемых при сдвиге регистра, также задействован в команде, которая управляет состоянием индивидуального бита (напр. OUT, KEEP(11)) появится сообщение об ошибке “COIL/OUT DUPL” при контроле программы на синтаксис с программматора или других устройств программирования. Программа, однако, будет выполняться, как и написана. Пример программирования см. пример 2.

Описание

SFT(10) управляется тремя условиями, I, P и R. Если SFT(10) выполняется и условие исполнения P = 1 и было = 0 в прошлом цикле R = 0, тогда условие I переместится в самый правый (младший) бит регистра сдвига, заданного 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 “вдвинуто” в регистр, только когда P = 1 и был = 0 в последний раз исполнения SFT(10). Если условие P не изменялось или изменилось с 1 на 0, регистр сдвига не изменяется.

St определяет правое (младшее) слово регистра сдвига, E - левое (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).

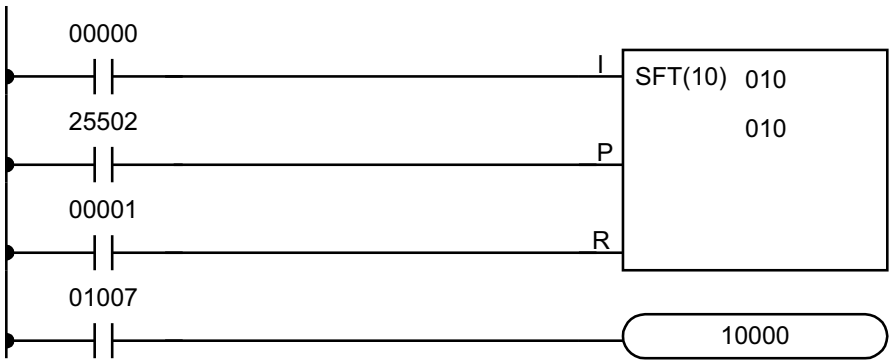
Когда условие исполнения на R устанавливается в 1, все биты в регистре устанавливаются в 0, и регистр сдвига не будет работать, пока R не станет = 0.

Флаги

Флагов, на которые воздействует команда SFT(10), нет.

Пример

Следующий пример использует бит часов 1 с (25502), чтобы условие, определяемое 00000, сдвигалось в IR 010 каждую секунду. Выход 10000 установится в 1, когда 1 “вдвигается” в 01007.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	LD	25502
00002	LD	00001
00003	SFT(10)	
		010
		010
00004	LD	01007
00005	OUT	10000

5.16.2 WSFT(16) - Сдвиг слова

Обозначение на схеме

WSFT(16)
St
E

@WSFT(16)
St
E

Область операндов

St	первое слово	IR, SR, AR, HR, LR, DM
E	последнее слово	IR, SR, AR, HR, LR, DM

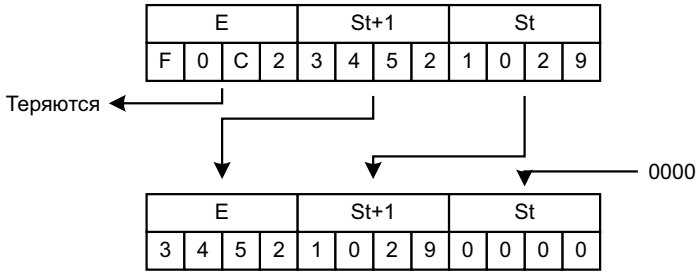
Ограничения

St и E должны быть в одной области данных и E должно быть больше либо равно St.

DM 6144..DM 6655 нельзя использовать для St или E.

Описание

Когда условие исполнения = 0, WSFT(16) не выполняется. Когда условие исполнения = 1, WSFT(16) сдвигает данные между St и E на слово. Нули записываются в St , а содержание E теряется.



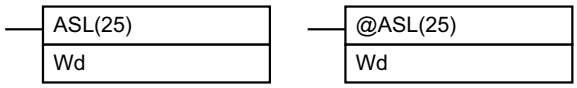
Флаги

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)

5.16.3 ASL(25) - Арифметический сдвиг влево

Обозначение на схеме



Область операндов

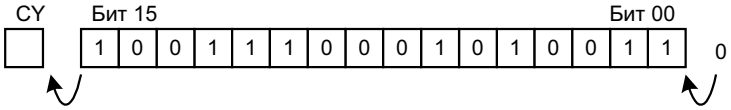
Wd	слово сдвига	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для Wd.

Описание

Когда условие исполнения = 0, ASL(25) не выполняется. Когда условие исполнения = 1, ASL(25) “вдвигает” 0 в бит 00 Wd, сдвигает биты слова Wd на 1 бит влево, и “вдвигает” состояние бита 15 в CY.



Предосторожности

В бит 00 будет заносится нуль каждый цикл, если используется нормальная версия ASL(25) (не фронта 0/1). Используйте версию фронта 0/1 (@ASL(25)) или комбинируйте ASL(25) с DIFU(13) , DIFU(14) для сдвига только один раз.

Флаги

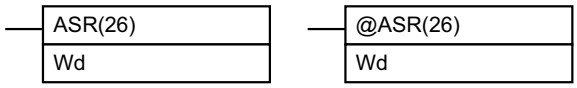
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

CY: Получает состояние бита 15

EQ: 1, когда содержимое Wd равно нулю, в противном случае = 0.

5.16.4 ASR(26) - Арифметический сдвиг вправо

Обозначение на схеме



Область операндов

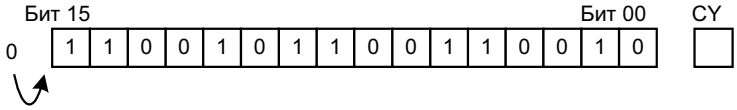
Wd	слово сдвига	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для Wd

Описание

Когда условие исполнения = 0, ASR(26) не выполняется. Когда условие исполнения = 1, ASR(26) “вдвигает” 0 в бит 15 Wd, сдвигает биты слова Wd на 1 бит вправо, и “вдвигает” состояние бита 00 в CY.



Предосторожности

В бит 00 будет заносится нуль каждый цикл, если используется нормальная версия ASR(26) (не фронта 0/1). Используйте версию фронта 0/1 (@ASR(26)) или комбинируйте ASR(26) с DIFU(13) , DIFU(14) для сдвига только один раз.

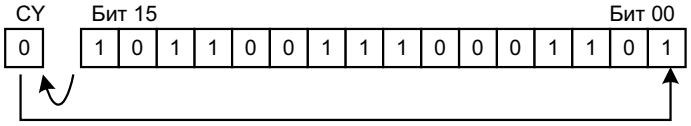
Флаги

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 равно нулю, в противном случае = 0.

5.16.5 ROL(27) - Циклический сдвиг влево

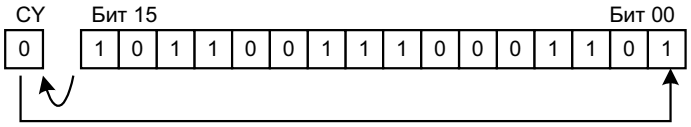


Ограничения

DM 6144..DM 6655 нельзя использовать для Wd

Описание

Когда условие исполнения = 0, ROL(27) не выполняется. Когда условие исполнения = 1, ROL(27) сдвигает все биты слова Wd на 1 бит влево, сдвигает CY в бит 00 Wd, и сдвигает состояние бита 15 в CY.



Предосторожности

Используйте команду SEC (41) для установки состояния CY в 1 или CLC(14) для очистки CY перед командой сдвига ROL(27), чтобы быть уверенным, что в CY содержатся правильное значение.

В бит 00 будет заносится CY каждый цикл, если используется нормальная версия ROL(27) (не фронта 0/1). Используйте версию фронта 0/1 (@ROL(27) или комбинируйте ROL(27) с DIFU(13) , DIFU(14) для сдвига только один раз.

Флаги

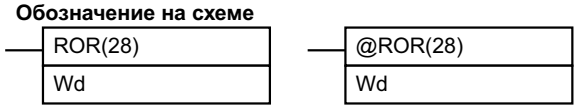
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM).

CY: Получает состояние бита 15.

EQ: 1, когда содержимое Wd равен нулю, в противном случае = 0.

5.16.6 ROR(28) - Циклический сдвиг вправо

Обозначение на схеме



Область операндов

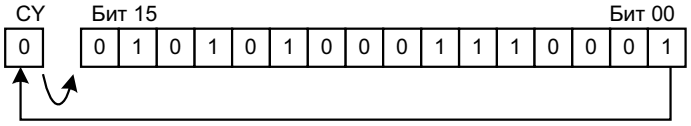
Wd	слово сдвига	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144 .. DM 6655 нельзя использовать для Wd.

Описание

Когда условие исполнения = 0, ROR(28) не выполняется. Когда условие исполнения = 1, ROR(28) сдвигает все биты слова Wd на 1 бит вправо, сдвигает CY в бит 15 Wd, и сдвигает бит 00 слова Wd в CY.



Предосторожности

Используйте команду SEC (41) для установки состояния CY в 1 или CLC(14) для очистки CY перед командой сдвига ROR(28) , чтобы быть уверенным, что в CY содержатся правильное значение.

CY будет сдвигаться в бит 15 каждой цикл, если используется нормальная версия ROR(28) (не фронта 0/1). Используйте версию фронта 0/1 (@ROR(28)) или комбинируйте ROR(28) с DIFU(13) , DIFU(14) для сдвига только один раз.

Флаги

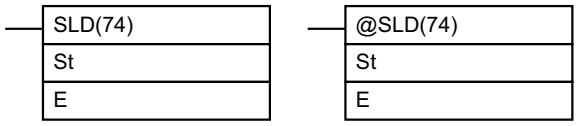
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

CY: Получает состояние бита 00

EQ: 1, когда содержимое Wd равен нулю, в противном случае = 0.

5.16.7 SLD(74) - Сдвиг влево на одну цифру

Обозначение на схеме



Область операндов

St	первое слово	IR, SR, AR, HR, LR, DM
E	последнее слово	IR, SR, AR, HR, LR, DM

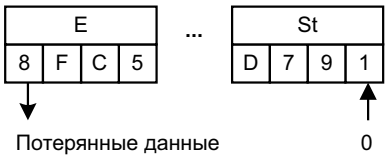
Ограничения

St и E должны быть в одной области данных, E должно быть больше либо равно St.

DM 6144 .. DM 6655 нельзя использовать для St и E.

Описание

Когда условие исполнения = 0, SLD(74) не выполняется. Когда условие исполнения = 1, SLD(74) сдвигает данные между St и E (включительно) на одну цифру (4 бита) влево. Нули записываются в правую (младшую) цифру St , а левая (старшая) цифра слова E теряется.



Предосторожности

Если во время операции сдвига более 50 слов происходит прерывания питания, операция сдвига может не завершиться.

В младшую цифру слова St будет заноситься нуль каждый цикл, если используется нормальная версия SLD(74) (не фронта 0/1). Используйте версию команды, работающую по фронту 0/1 (@SLD(74)) или комбинируйте SLD(74) с DIFU(13) или DIFU(14) для сдвига только один раз.

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)

5.16.8 SRD(75) - Сдвиг вправо на одну цифру

Обозначение на схеме

SRD(75)
E
St

@SRD(75)
E
St

Область операндов

E	последнее слово	IR, SR, AR, HR, LR, DM
St	первое слово	IR, SR, AR, HR, LR, DM

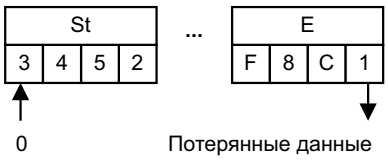
Ограничения

St и E должны быть в одной области данных, E должно быть меньше либо равно St.

DM 6144..DM 6655 нельзя использовать для St и E.

Описание

Когда условие выполнения = 0, SRD(75) не выполняется. Когда условие = 1, SRD(75) сдвигает данные между St и E (включительно) на одну цифру (4 бита) вправо. Нуль записывается в левую (старшую) цифру слова St, а правая (младшая) цифра слова E теряется.



Предосторожности

Если во время операции сдвига более 50 слов происходит прерывания питания, операция сдвига может не завершиться.

В старшую цифру St будет заноситься нуль каждый цикл, если используется нормальная версия SRD(75) (не фронта 0/1). Используйте версию команды, работающую по фронту 0/1 (@SRD(75)) или комбинируйте SRD(75) с DIFU(13) или DIFU(14) для сдвига только один раз.

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)

5.16.9 SFTR(84) - Реверсивный регистр сдвига

Обозначение на схеме

SFTR(84)
C
St
E

@SFTR(84)
C
St
E

Область операндов

C	слово состояния	IR, SR, AR, HR, LR, DM
St	первое слово	IR, SR, AR, HR, LR, DM
E	последнее слово	IR, SR, AR, HR, LR, DM

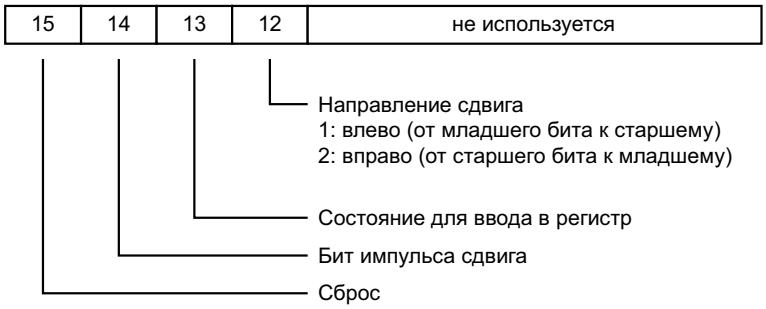
Ограничения

St и E должны лежать в одной области данных. St должно быть меньше либо равно E.

DM 6144..DM 6655 нельзя использовать для C, St и E.

Описание

SFTR(84) используется 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 задаются направление сдвига, состояние, которое должно быть “вдвинуто” в регистр, импульс сдвига и вход 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(84) выполняется (до тех пор, пока бит сброса = 0 и пока бит 14 = 1). Если SFTR(84) выполняется с условием 0 или с битом 14 = 0, регистр сдвига останется в прежнем состоянии. Если SFTR(84) выполняется с условием 1 и бит сброса (бит 15) = 1, весь регистр сдвига и CY сбросятся в 0.

Флаги

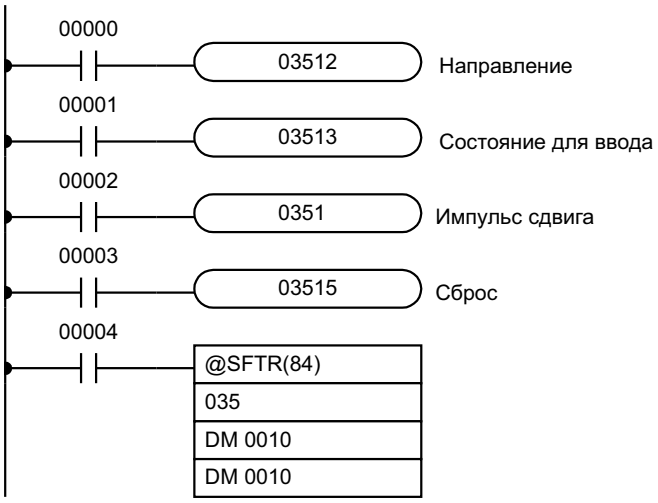
ER: St и E лежат в разных областях или St больше E.

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 следующем примере IR 00000, IR 00001, IR 00002 и IR 00003 используются для управления битами слова C, используемого в @SFTR(84). Регистр сдвига - DM 0010, и он управляется с помощью IR 00004.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	03512
00002	LD	00001
00003	OUT	03513
00004	LD	00002

Адрес	Инструкция	Операнд
00005	OUT	00514
00006	LD	00003
00007	OUT	03515
00008	LD	00004
00009	@SFT(10)	
		035
		DM 0010
		DM 0010

5.16.10 ASFT(17) - Асинхронный регистр сдвига

Обозначение на схеме

ASFT(17)
C
St
E

@ASFT(17)
C
St
E

Область операндов

C	слово состояния	IR, SR, AR, HR, LR, DM, #
St	первое слово	IR, SR, AR, HR, LR, DM
E	последнее слово	IR, SR, AR, HR, LR, DM

Ограничения

St и E должны лежать в одной области данных. E должно быть больше либо равно St.

DM 6144..DM 6655 нельзя использовать для St и E.

Описание

Когда условие исполнения = 0, ASFT(17) действий не производит и программа переходит к следующей команде. Когда условие исполнения = 1, ASFT(17) используется для создания и управления реверсивным асинхронным регистром сдвига между ST и E. Данный регистр сдвигает слова, только когда следующее слово в регистре = 0, например, если в регистре нет слов, содержащих нули, ничего не сдвигается. Итак, только одно слово сдвигается для каждого слова регистра, содержащего нуль. 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 - сдвиг вниз, к младшим словам, 0 - сдвиг вверх, 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 включите бит 15 в 0.

Замечание Если используется вариант команды без фронта 0/1 ASFT(17), данные будут сдвигаться каждый цикл, когда условия исполнения = 1. Для предотвращения этого используйте версию команды фронта 0/1.

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 следующем примере показана команда ASFT(17), используемая для сдвига слов в регистре сдвига из 11 слов между DM 0100..DM 0110 с C = # 6000. Ненулевые данные сдвигаются к St (DM 0110).

Адрес	Инструкция	Операнд
00000	LD	00000
00001	ASFT(17)	
		# 6000
		DM 0100
		DM 0110

Замечание Нули сдвигаются “вперед” (“вверх”), если C = 4000, а весь регистр станет = 0 при C = 8000.

5.17 Команды пересылки данных

5.17.1 MOV(21) - Пересылка

Обозначение на схеме

MOV(21)
S
D

@MOV(21)
S
D

Область операндов

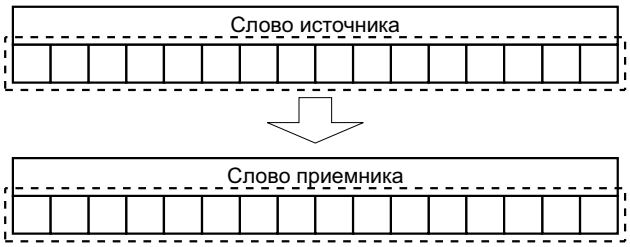
S	слово источника	IR, SR, AR, HR, TC, LR, DM, #
D	слово приемника	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать D.

Описание

Когда условие исполнения = 0, MOV(21) не выполняется. Когда условие исполнения = 1, MOV(21) копирует содержимое S в D.



Состояние битов не изменяется

Предосторожности

Номера TC нельзя задавать в качестве D для изменения текущего значения таймера или счетчика. Вы можете, однако, легко изменить текущее состояние таймера или счетчика командой BSET(71).

Флаги

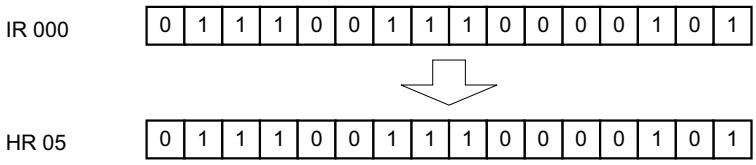
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

EQ: 1, когда в D передаются все нули.

Пример

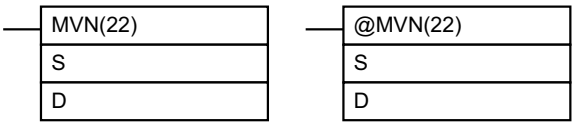
В следующем примере показано использование @MOV(21) для копирования содержимого IR 00 в HR 05, когда IR 00000 изменятся из 0 в 1.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@MOV(21)	
		001
		HR 05



5.17.2 MVN(22) - Пересылка инверсии

Обозначение на схеме



Область операндов

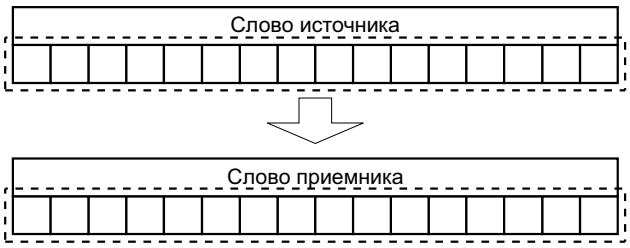
S	слово источник	IR, SR, AR, HR, TC, LR, DM, #
D	слово приемник	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для D.

Описание

Когда условие исполнения = 0, MVN(22) не выполняется. Когда условие исполнения = 1, MVN(22) передает инверсное содержимое S (заданное слово или 4-разрядную 16-ричную константу) в D, т.е. для каждого бита S в 1 соответствующий бит в D будет = 0, а для каждого бита S в 0 соответствующий бит в D будет 1.



Состояние битов инвертируется

Предосторожности

Номера TC нельзя задавать в качестве D для изменения текущего значения таймера или счетчика. Однако текущее состояние легко изменить командой BSET(71).

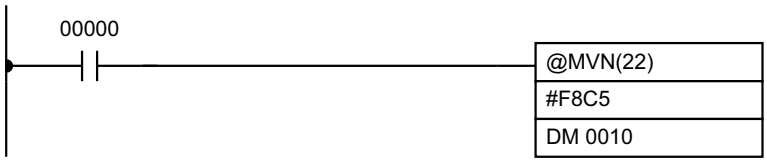
Флаги

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.

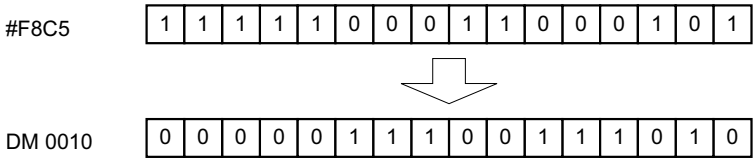
Пример

В следующем примере показано использование @MVN(22) для копирования дополнения #F8C5 в DM 0010, когда IR 00001 переключается из 0 в 1.



Адрес	Инструкция	Операнд
00000	LD	00001

Адрес	Инструкция	Операнд
00001	@MVN(22)	
		# F8C5
		DM 0010



5.17.3 XFER(70) - Пересылка блока

Обозначение на схеме

XFER(70)
N
S
D

@XFER(70)
N
S
D

Область операндов

N	количество слов (BCD)	IR, SR, AR, HR, TC, LR, DM, #
S	первое слово источника	IR, SR, AR, HR, TC, LR, DM
D	первое слово приемника	IR, SR, AR, HR, TC, LR, DM

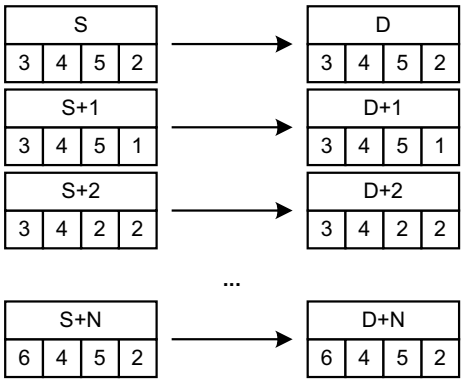
Ограничения

S и S+N должны лежать в одной области данных, также как и D и D+N.

DM 6144..DM 6655 нельзя использовать для D.

Описание

Когда условие исполнения = 0, XFER(70) не выполняется. Когда условие исполнения = 1, XFER(70) копирует содержимое S , S+1, ...S+N в D, D+1, ...D+N.



Флаги

ER: N не двоично-десятичное число.

Слова S и S+N или D и 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)

5.17.4 BSET(71) - Заполнение блока

Обозначение на схеме

BSET(71)
S
St
E

@BSET(71)
S
St
E

Область операндов

S	данные источника	IR, SR, AR, HR, TC, LR, DM, #
St	первое слово	IR, SR, AR, HR, TC, LR, DM
E	последнее слово	IR, SR, AR, HR, TC, LR, DM

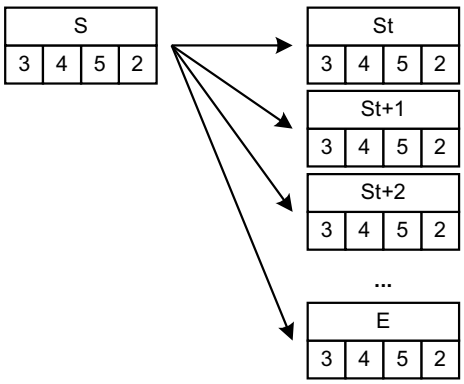
Ограничения

St должно быть меньше либо равно E, St и E должны лежать в одной области данных.

DM 6144..DM 6655 нельзя использовать для St и E.

Описание

Когда условие исполнения = 0, BSET(71) не выполняется. Когда условие исполнения = 1, BSET(71) копирует содержимое S во все слова от St до E.



BSET(71) можно использовать для изменения текущего значения таймеров и счетчиков. (Это нельзя сделать командами MOV(21) и MVN(22)). BSET(71) можно также использовать для очистки секций области данных, т.е. области DM, для подготовки исполнения других команд. Ее также можно использовать для очистки слов путем посылки туда всех нулей.

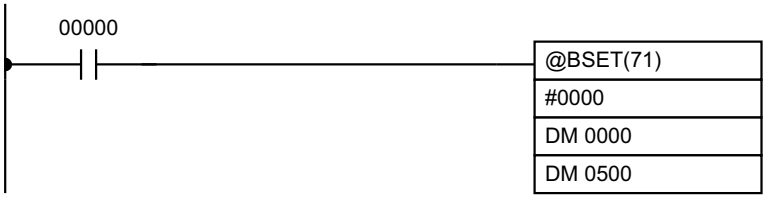
Флаги

ER: 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(71) для копирования константы (#0000) в блок области DM (DM 0000..DM 0500), когда IR 0000 = 1.

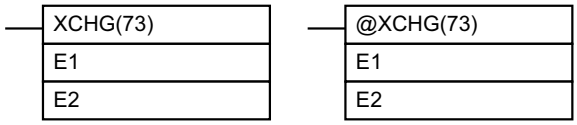


Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	@BSET(71)	
		# 0000
		DM 0000
		DM 0500

5.17.5 XCHG(73) - Обмен данными

Обозначение на схеме



Область операндов

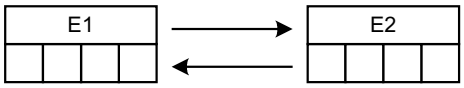
E1	слово обмена 1	IR, SR, AR, HR, TC, LR, DM
E2	слово обмена 2	IR, SR, AR, HR, TC, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для E1 и E2.

Описание

Когда условие исполнения = 0, XCHG(73) не выполняется. Когда условие исполнения = 1, XCHG(73) обменивает содержимое E1 и E2.



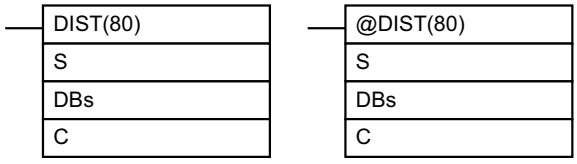
Если Вы хотите обменять содержимое блоков с размером больше 1 слова, используйте рабочие слова как промежуточный буфер для хранения одного из блоков и используйте команду XFER(70) три раза.

Флаги

ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

5.17.6 DIST(80) - Распределение одного слова

Обозначение на схеме



Область операндов

S	данные источника	IR, SR, AR, HR, TC, LR, DM, #
DBs	базовое слово приемника	IR, SR, AR, HR, TC, LR, DM
C	слово состояния (BCD)	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

C должно быть двоично-десятичным числом.

DM 6144..DM 6655 нельзя использовать для DBs или C.

Описание

DIST(80) служит для распределения одного слова или операций со стеком в зависимости от содержания слова управления C.

Распределение одного слова

Когда биты 12..15 слова C = 0..8, DIST(80) можно использовать для операции распределения одного слова. Полное содержание C Задает смещение Of.

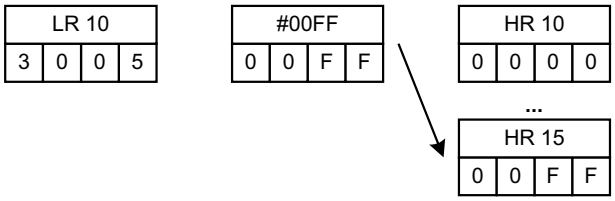
Когда условие исполнения = 0, DIST(80) не выполняется. Когда условие исполнения = 1, DIST(80) копирует содержимое S в DBs + Of, т.е. Of добавляется к DBs для определения слова приемника.

Замечание DBs и DBs+Of должны находится в одной области и не могут быть DM 6144..DM 6655.

Пример

Следующий пример показывает, как использовать DIST(80) для копирования #00FF в HR 10 + Of. Содержание LR 10 равно #3005, так что #00FF копируется в HR 15 (HR 10 + 5), когда IR 00000 = 1.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@DIST(80)	
		# 00FF
		HR 10
		LR 10



Операция со стеком

Когда биты 12..15 слова C = 9, DIST(80) можно использовать для работы со стеком. Другие 3 цифры указывают количество слов в стеке (000..999). Содержание DBs является указателем стека.

Когда условие исполнения = 0, DIST(80) не выполняется. Когда условие исполнения = 1, DIST(80) копирует содержимое S в DBs + 1 + содержание Dbs. Иначе говоря, для определения слова приемника к DBs добавляется 1 + содержание Dbs. Затем содержимое Dbs инкрементируется на 1.

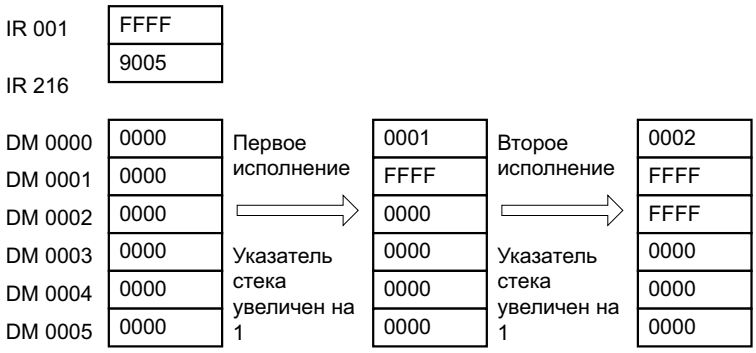
Замечание 1. DIST(80) будет выполняться каждый цикл, если применяется версия не фронта 0/1 (@DIST(80)) или DIST(80) применяется без DIFU(13) или DIFD(14).

2. Обязательно проинициализируйте указатель стека перед использованием DIST(80) в качестве операции стека.

Пример

Следующий пример показывает, как использовать DIST(80) для создания стека из слов DM 0001..DM 0005. DM 00000 служит указателем стека.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@DIST(80)	
		001
		DM 0000
		216



Флаги

ER: Сдвиг или длина стека в слове управления не являются в двоично-десятичными числами.

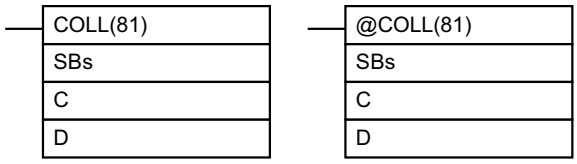
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

Во время стековой операции значение указателя стека +1 превысило длину стека.

EQ: 1, когда содержание S = 0; в противном случае = 0.

5.17.7 COLL(81) - Сбор данных

Обозначение на схеме



Область операндов

SBs	базовое слово источника	IR, SR, AR, HR, TC, LR, TR, DM
C	слово состояния (BCD)	IR, SR, AR, HR, TC, LR, TR, DM
D	слово приемника	IR, SR, AR, HR, TC, LR, TR, DM

Ограничения

C должно быть двоично-десятичным числом.

DM 6144..DM 6655 нельзя использовать для D.

Описание

COLL(81) служит для сбора данных, операции со стеком FIFO или операции со стеком LIFO в зависимости от содержания слова управления C.

Сбор данных

Когда биты 12..15 слова C = 0..7, COLL(81) используется для сбора данных. Все содержание C Задает смещение, Of.

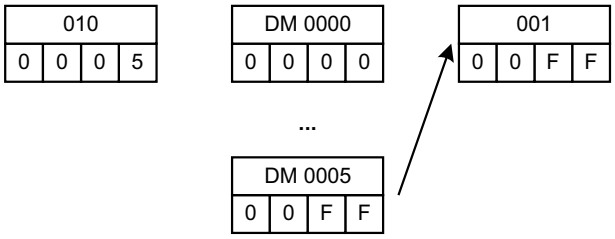
Когда условие исполнения = 0, COLL(81) не выполняется. Когда условие исполнения = 1, COLL(81) копирует содержимое SBs + Of в D, т.е. для определения слова источника к DBs добавляется Of.

Замечание SBs и SBs+Of должны находится в одной области данных.

Пример

Следующий пример показывает, как использовать COLL(81) для копирования содержания DM 0000 + Of в IR 001. Содержание 010 = #0005, так что содержание DM 0005 (DM 0000 +5) копируется в IR 001, когда IR 00001 = 1.

Адрес	Инструкция	Операнд
00000	LD	00001
00001	@COLL(81)	
		DM 0000
		010
		001



Операция со стеком FIFO

Когда биты 12..15 слова C = 9, COLL(81) можно использовать для операции со стеком FIFO. Другие 3 цифры слова C задают количество слов в стеке (000..999). Содержание SBs является указателем стека.

Когда условие исполнения = 1, COLL(81) сдвигает содержание каждого слова в стеке вниз на один адрес, последними сдвигаются данные из SBs + 1 (первое значение, занесенное в стек), в слово приемника (D). Содержание указателя стека (SBs) уменьшается на 1.

Замечание COLL(81) будет выполняться каждый цикл, если применяется версия не фронта 0/1 или COLL(81) применяется без DIFU(13) или DIFD(14).

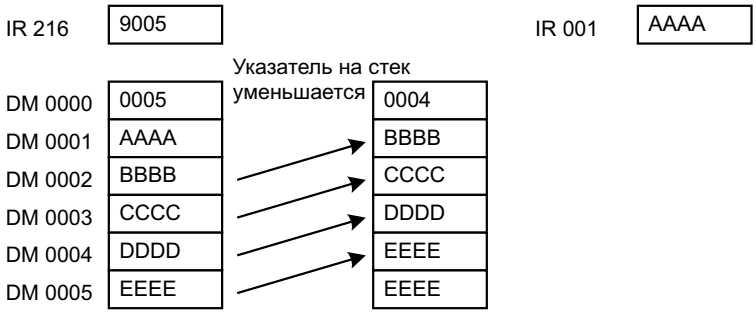
Пример

Следующий пример показывает, как использовать COLL(81) для создания стека в словах DM 0001..DM 0005. DM 0000 действует, как указатель стека.

Когда IR 00000 изменяется из 0 в 1, COLL(81) сдвигает содержание DM 0002..DM 0005 вниз на 1 адрес, и переносит данные из DM 0001 в IR 001. Содержание указателя стека (DM 0000) уменьшается на 1.

Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	@COLL(81)	
		DM 0000
		216
		001



Операция со стеком LIFO

Когда биты 12..15 слова C = 8, COLL(81) служит для операции со стеком LIFO. Другие 3 цифры C задает количество слов в стеке (000..999). Содержание SBs является указателем стека.

Когда условие исполнения = 1, COLL(81) копирует данные из слова, указанного указателем стека (SBs + содержание Sbs) в слово приемника (D). Указатель стека (SBs) уменьшается на 1.

Указатель стека - единственное слово, которое изменяется в стеке.

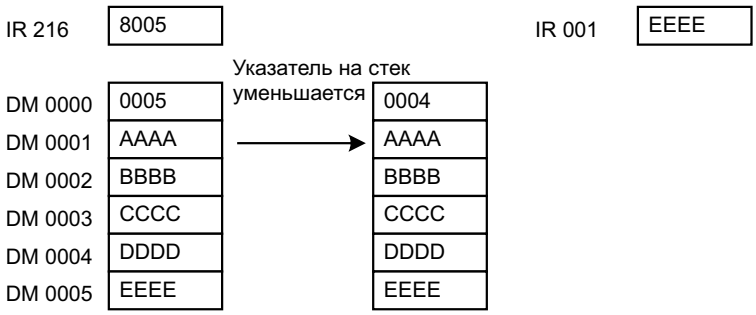
Замечание COLL(81) будет выполняться каждый цикл, если применяется версия не фронта 0/1 или COLL(81) применяется без DIFU(13) или DIFD(14).

Пример

Следующий пример показывает, как использовать COLL(81) для создания стека между DM 0001 и DM 0005. DM 0000 действует, как указатель стека.

Когда IR 00000 изменяется из 0 в 1, COLL(81) копирует содержание DM 0005 (DM 0000+ 5) в IR 001. Содержание указателя стека (DM 0000) уменьшается на 1.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@COLL(81)	
		DM 0000
		216
		001



Флаги

ER: Сдвиг или длина стека в слове управления не являются двоично-десятичным числом.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

Во время стековой операции значение указателя стека превышает длину стека; была сделана попытка записать в слово за пределами конца стека.

EQ: 1, когда содержание S = 0; в противном случае = 0.

5.17.8 MOVB(82) - Переслать бит

Обозначение на схеме

MOVB(82)
S
Bi
D

@MOVB(82)
S
Bi
D

Область операндов

S	слово источник	IR, SR, AR, HR, LR, DM, #
Bi	определитель бита (BCD)	IR, SR, AR, HR, TC, LR, DM, #
D	слово назначения	IR, SR, AR, HR, LR, DM

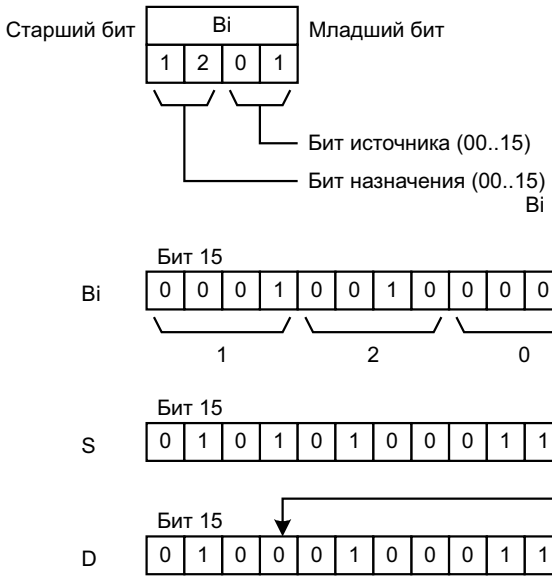
Ограничения

Две младших цифры и две старших цифры BI должны быть в диапазоне 00..15.

DM 6144..DM 6655 нельзя использовать для BI или D.

Описание

Когда условие исполнения = 0, MOVB(82) не выполняется. Когда условие исполнения = 1, MOVB(82) копирует указанный бит S в указанный бит D. Биты S и D задаются в BI. Две правые (младшие) цифры слова BI задают бит источника, две левые (старшие) цифры слова BI задают бит приемника.



Флаги

ER: BI не в двоично-десятичном виде, или определяет несуществующий бит (т.е. указатель бита должен быть между 00 и 15).

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или превышена граница области DM)

5.17.9 MOVD(83) - Переслать цифру

Обозначение на схеме

MOVD(83)
S
Di
D

@MOVD(83)
S
Di
D

Область операндов

S	слово источник	IR, SR, AR, HR, TC, LR, DM, #
Di	определитель слова (BCD)	IR, SR, AR, HR, TC, LR, DM, #
D	слово приемник	IR, SR, AR, HR, TC, LR, DM

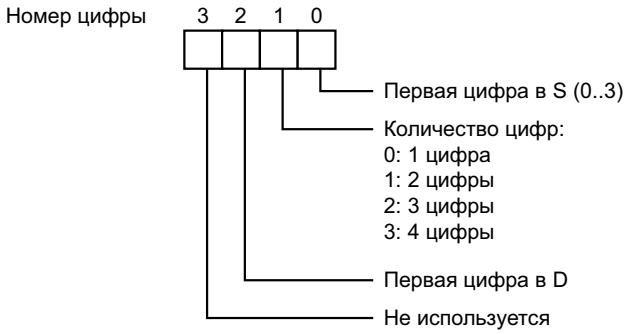
Ограничения

Три правые (младшие) цифры DI должны быть в диапазоне 0..3.

DM 6144..DM 6655 нельзя использовать для DI или D.

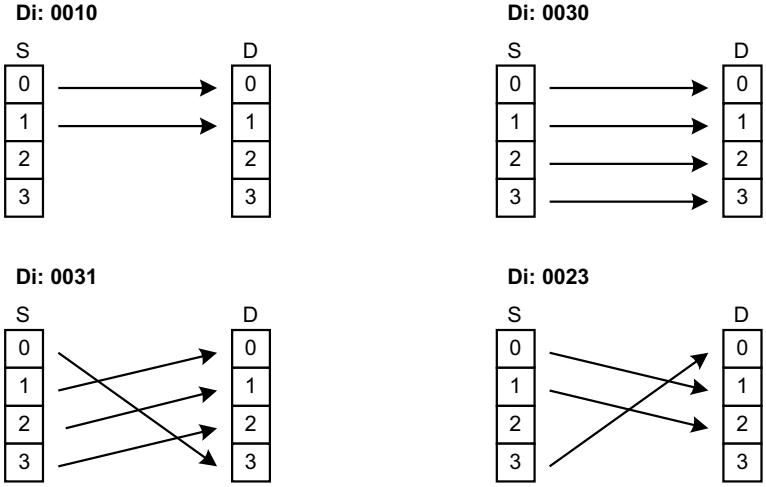
Описание

Когда условие исполнения = 0, MOVD(83) не выполняется. Когда условие исполнения = 1, MOVD(83) копирует содержание указанной цифры (цифр) из S в указанную цифру (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 первой цифры и продолжается указанное число цифр. Если достигнута последняя цифра в S либо D, дальнейшие операции продолжаются с цифры 0.



Определитель цифр

На рисунке показаны примеры перемещения данных для разных значений DI.



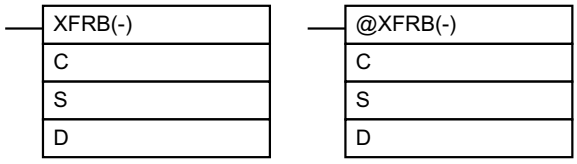
Флаги

ER: По крайней мере одна из трех правых (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)

5.17.10 XFRB(-) - Переслать биты

Обозначение на схеме



Область операндов

C	слово состояния	IR, SR, AR, HR, TC, LR, DM, #
S	первое слово источника	IR, SR, AR, HR, TC, LR, DM
D	первое слово приемника	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда имеется только в CQM1-CPU4-E.

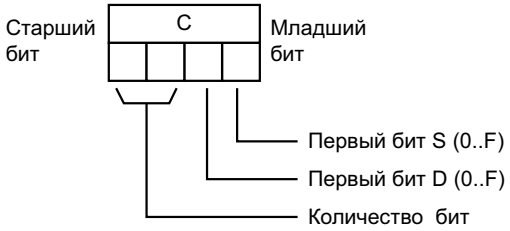
Заданные биты источника должны находиться в одной области данных.

Заданные биты приемника должны находиться в одной области данных.

DM 6144..DM 6655 нельзя использовать для D.

Описание

Когда условие исполнения = 0, XFRB(-) не выполняется. Когда условие исполнения = 1, XFRB(-) копирует указанные биты источника в указанные биты приемника. Две правые (младшие) цифры слова C задают начальные биты в S и D, а две левые (старшие) цифры слова C задают количество битов, подлежащих копированию.

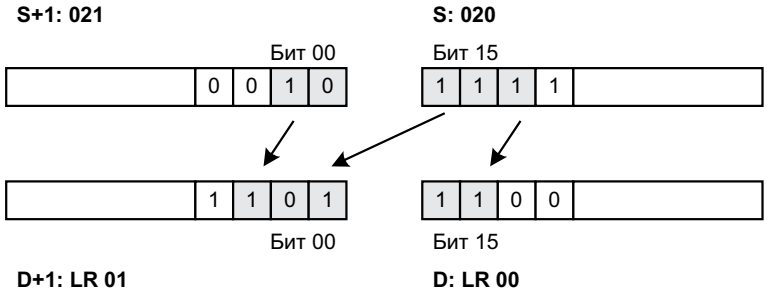


Замечание До 255 (FF) битов можно скопировать за один раз.

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 - D (13), а стартовый бит в LR 00 - E (14), так что IR 02013..IR 02101 копируются в LR 0014..LR 0102.

Адрес	Инструкция	Операнд
00000	LD	00001
00001	XFRB(-)	
		# 05ED
		020
		LR 00



Флаги

ER: Заданные биты источника находятся в разных областях.

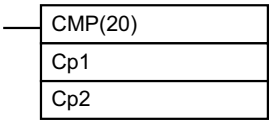
Заданные биты приемника находятся в разных областях.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

5.18 Команды сравнения

5.18.1 CMP(20) - Сравнение

Обозначение на схеме



Область операндов

Cp1	первое слово сравнения	IR, SR, AR, HR, TC, LR, DM, #
Cp2	второе слово сравнения	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

При сравнении значений с текущим значением (PV) таймера или счетчика сравниваемое значение должно быть двоично-десятичным числом.

Описание

Когда условие исполнения = 0, CMP(20) не выполняется. Когда условие исполнения = 1, CMP(20) сравнивает слова Cp1 и Cp2 и выдает результат во флаги GR, EQ и LE в области SR.

Предосторожности

Размещение других команд между командой CMP(20) и командами, которые используют флаги GR, EQ и LE, могут изменить состояние этих флагов. Обязательно обращайтесь к этим флагам перед тем, как это состояние 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 = Cp2.

LE: 1, если Cp1 меньше Cp2.

GR: 1, если Cp1 больше Cp2.

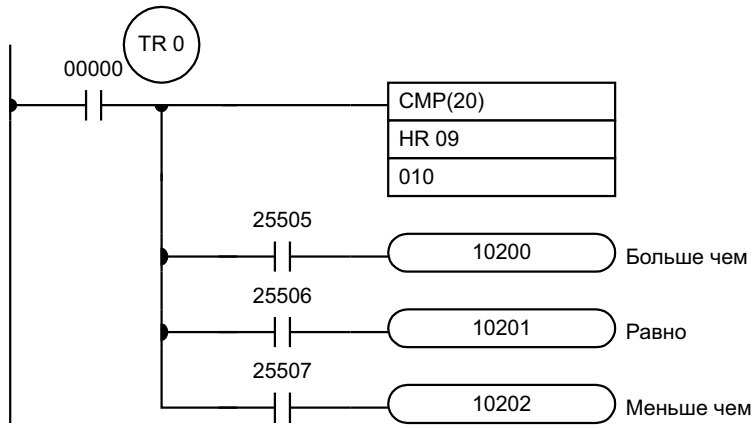
Результат сравнения	**Флаг**		
	GR	**EQ**	**LE**
Cp1 < Cp2	0	0	1
Cp1 = Cp2	0	1	0
Cp1 > Cp2	1	0	0

Пример

В следующем примере показано, как немедленно сохранить результаты сравнения.

- Если содержание HR 09 < содержания 010, 10200 устанавливается в 1.
- Если содержание HR 09 = содержанию 010, 10201 устанавливается в 1
- Если содержание HR 09 > содержания 010, 10202 устанавливается в 1.

В некоторых задачах требуется использование только одного из трех значений, тогда использование TR 0 необязательно. При таком стиле программирования 10200, 10201 и 10202 изменяются только при выполнении CMP(20).



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	TR 0
00002	CMP(20)	
		010
		HR 09
00003	LD	TR 0
00004	AND	25505
00005	OUT	10200
00006	LD	TR 0
00007	AND	25506
00008	OUT	10201
00009	LD	TR 0
00010	AND	25507
00011	OUT	10202

5.18.2 TCMP(85) - Сравнение таблицы

Обозначение на схеме

TCMP(85)
CD
TB
R

@TCMP(85)
CD
TB
R

Область операндов

CD	сравниваемые данные	IR, SR, HR, TC, LR, DM, #
TB	первое слово таблицы сравнения	IR, SR, HR, TC, LR, DM
R	слово результата	IR, SR, HR, TC, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, TCMP(85) не выполняется. Когда условие исполнения = 1, TCMP(85) сравнивает CD с содержанием TB, TB+1, TB+2,....TB+15. Если CD равно содержимому одному из этих слов, устанавливается соответствующий бит в R, например, если содержание CD = содержанию TB, бит 00 установится в 1, если содержание CD = содержанию TB+1, бит 01 установится в 1 и т.д. Остальные биты в R будут установлены в 0.

Флаги

ER: Таблица сравнения (т.е. TB..TB+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(85). Здесь, сравнение происходит каждый цикл, когда IR 00000 = 1.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	TCMP(85)	
		00
		DM 0000
		216

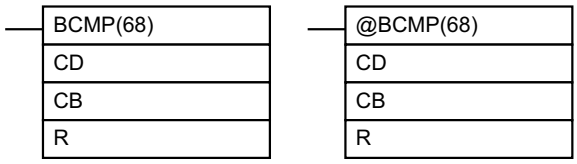
CD: 001 Верхние границы R: 216

IR 001 0210

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

5.18.3 BCMP(68) - Сравнение блока

Обозначение на схеме



Область операндов

CD	сравниваемые данные	IR, SR, AR, HR, TC, LR, DM, #
CB	первое слово блока сравнения	IR, SR, HR, TC, LR, DM
R	слово результата	IR, SR, AR, HR, TC, LR, DM

Ограничения

Каждое слово нижней границы в блоке сравнения должно быть меньше либо равно верхней границе.

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, BCMP(68) не выполняется. Когда условие исполнения = 1, BCMP(68) 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 находится внутри одной из таких зон, (включая верхнюю и нижнюю границы) устанавливается соответствующий бит в слове R. Проделанное сравнение и соответствующие биты R, установленные для результата сравнения “истина”, показаны далее. Остальные биты слова R = 0.

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

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 сравнения и полученные результаты BCMP(68). Здесь, сравнение происходит каждый цикл, когда IR 0000 = 1.

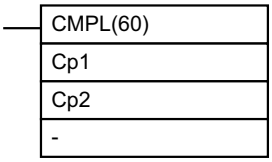
Адрес	Инструкция	Операнд
00000	LD	00000
00001	BCMP(68)	
		001
		DM 0010
		LR 05

CD: 001 IR 001 0210 Сравнить данные в IR 001 (там содержится 210) с приведенными зонами

Верхние границы		Нижние границы		R: LR 05	
DM 0010	0000	DM 0011	0100	LR 0500	0
DM 0012	0101	DM 0013	0200	LR 0501	0
DM 0014	0201	DM 0015	0300	LR 0502	1
DM 0016	0301	DM 0017	0400	LR 0503	0
DM 0018	0401	DM 0019	0500	LR 0504	0
DM 0020	0501	DM 0021	0600	LR 0505	0
DM 0022	0601	DM 0023	0700	LR 0506	0
DM 0024	0701	DM 0025	0800	LR 0507	0
DM 0026	0801	DM 0027	0900	LR 0508	0
DM 0028	0901	DM 0029	1000	LR 0509	0
DM 0030	1001	DM 0031	1100	LR 0510	0
DM 0032	1101	DM 0033	1200	LR 0511	0
DM 0034	1201	DM 0035	1300	LR 0512	0
DM 0036	1301	DM 0037	1400	LR 0513	0
DM 0038	1401	DM 0039	1500	LR 0514	0
DM 0040	1501	DM 0041	1600	LR 0515	0

5.18.4 CMPL(60) - Сравнение слов двойной длины

Обозначение на схеме



Область операндов

Cp1	первое слово первой пары сравниваемых слов	IR, SR, AR, HR, TC, LR, DM
Cp2	первое слово второй пары сравниваемых слов	IR, SR, AR, HR, TC, LR, DM

Ограничения

Cp1 и Cp1 + 1 должны лежать в одной области данных

Cp2 и Cp2 + 1 должны лежать в одной области данных

Третий операнд задайте равным нулю.

Описание

Когда условие исполнения = 0, CMPL(60) не выполняется. Когда условие исполнения = 1, CMPL(60) объединяет 4-значное 16-ричное содержание Cp1 с 4-значным содержимым Cp1+1 и содержимое Cp2 с Cp2+1 для создания 8-значных 16-ричных чисел (Cp1+1,Cp1 и Cp2+1, Cp2). Два восьмизначных числа сравниваются и результатом являются флаги GR, EQ, LE в области SR .

Предосторожности

Размещение других команд между командой CMP(20) и командами, которые используют флаги GR, EQ и LE, могут изменить состояние этих флагов. Обязательно произведите обращения к этим флагам перед тем, как их состояние изменится.

Флаги

ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

GR: 1, если Cp1+1, Cp1 больше Cp2+1,Cp2.

EQ: 1, если Cp1+1, Cp1 равно Cp2+1,Cp2.

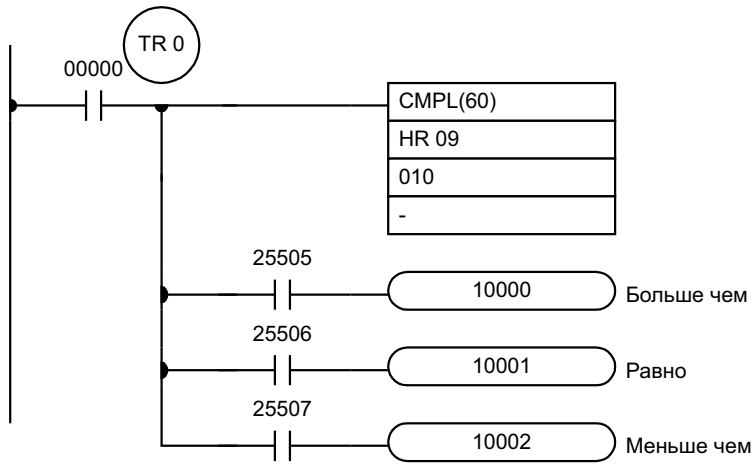
LE: 1, если Cp1+1, Cp1 меньше Cp2+1,Cp2.

Пример

В следующем примере показано, как немедленно сохранить результаты сравнения.

- Если содержание HR 10, HR 09 больше содержания 011, 010, тогда 10000 устанавливается в 1.
- Если содержание HR 10, HR 09 равно содержанию 011, 010, тогда 10001 устанавливается в 1.
- Если содержание HR 10, HR 09 меньше содержания 011, 010, тогда 10002 устанавливается в 1.

В некоторых задачах требуется использование только одного из трех значений, тогда использование TR 0 необязательно. При таком стиле программирования 10000, 10001 и 10002 изменяются только при выполнении CMPL(60).

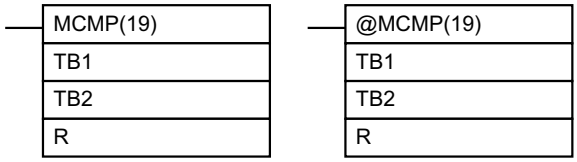


Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	TR 0
00002	CMPL(60)	
		HR 09
		010
		-
00003	AND	25505
00004	OUT	10000
00005	LD	TR 0

Адрес	Инструкция	Операнд
00006	AND	25506
00007	OUT	10001
00008	LD	TR 0
00009	AND	25507
00010	OUT	10002

5.18.5 MCMP(19) - Сравнение нескольких слов

Обозначение на схеме



Область операндов

TB1	первое слово таблицы 1	IR, SR, AR, HR, TC, LR, DM
TB2	первое слово таблицы 2	IR, SR, AR, HR, TC, LR, DM
R	слово результата	IR, SR, AR, HR, TC, LR, DM

Ограничения

Данная команда имеется только в CQM1.

TB1 и TB1 + 15 должны лежать в одной области данных

TB1 и TB1 + 15 должны лежать в одной области данных

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, MCMP(19) не выполняется. Когда условие исполнения = 1, MCMP(19) 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 установлены в 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)

EQ: 1, если равно содержимое двух таблиц, и R = 0000.

Пример

В следующем примере показаны сравнения и результаты MCMP(19). Сравнение производится в каждом цикле, когда 00000 = 1.

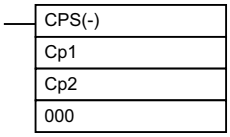
Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	MCMP(19)	
		100
		DM 0200
		DM 0300

	TB1: IR 100		TB2: DM 0200		R: DM 0300	
IR 100	0100	↔	DM 0200	0100	→ DM 030000	0
IR 101	0200	↔	DM 0201	0200	→ DM 030001	0
IR 102	0210	↔	DM 0202	0210	→ DM 030002	0
IR 103	ABCD	↔	DM 0203	0400	→ DM 030003	1
IR 104	ABCD	↔	DM 0204	0500	→ DM 030004	1
IR 105	ABCD	↔	DM 0205	0600	→ DM 030005	1
IR 106	ABCD	↔	DM 0206	0700	→ DM 030006	1
IR 107	0800	↔	DM 0207	0800	→ DM 030007	0
IR 108	0900	↔	DM 0208	0900	→ DM 030008	0
IR 109	1000	↔	DM 0209	1000	→ DM 030009	0
IR 110	ABCD	↔	DM 0210	0210	→ DM 030010	1
IR 111	ABCD	↔	DM 0211	1200	→ DM 030011	1
IR 112	ABCD	↔	DM 0212	1300	→ DM 030012	1
IR 113	1400	↔	DM 0213	1400	→ DM 030013	0
IR 114	0210	↔	DM 0214	0210	→ DM 030014	0
IR 115	1212	↔	DM 0215	1600	→ DM 030015	1

5.18.6 CPS(-) - Сравнение двоичных чисел со знаком

Обозначение на схеме



Область операндов

Cp1	первое сравниваемое слово	IR, SR, AR, HR, TC, LR, DM, #
Cp2	второе сравниваемое слово	IR, SR, AR, HR, TC, LR, DM, #
000	не используется, установить в 000	

Ограничения

Данная команд есть только в CQM1-CPU4-EV1.

Описание

Когда условие исполнения = 0, CPS(-) не выполняется. Когда условие исполнения = 1, CPS(-) сравнивает 16-битовые (4 цифры) двоичные числа со знаком Cp1 и Cp2 и выдает результат во флаги GR, EQ, LE в области SR .

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.

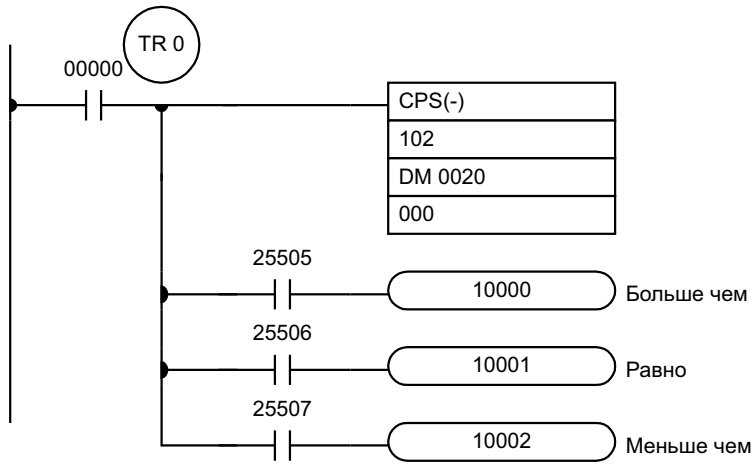
LE: 1, если Cp1 меньше Cp2.

GR: 1, если Cp1 больше Cp2.

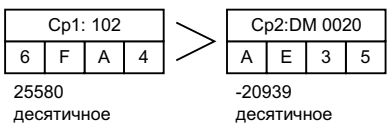
Результат сравнения	**Флаг**		
	GR	**EQ**	**LE**
Cp1 < Cp2	0	0	1
Cp1 = Cp2	0	1	0
Cp1 > Cp2	1	0	0

Пример

В следующем примере содержимое 102 больше содержимого DM 0020, так что 10000 устанавливается в 1, а другие биты, 10001 и 10002, устанавливаются в 0.

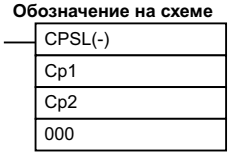


Адрес	**Инструкция**	**Операнд**
00000	LD	00500
00001	OUT	TR 0
00002	CPS(-)	
		102
		DM 0020
		000
00003	AND	25505
00004	OUT	10000
00005	LD	TR 0
00006	AND	25506
00007	OUT	10001
00008	LD	TR 0
00009	AND	25507
00010	OUT	10002



5.18.7 CPSL(-) - Сравнение двоичных слов двойной длины со знаком

Обозначение на схеме



Область операндов

Cp1	первое сравниваемое слово	IR, SR, AR, HR, TC, LR, DM, #
Cp2	второе сравниваемое слово	IR, SR, AR, HR, TC, LR, DM, #
000	не используется, установить в 000	

Ограничения

Данная команд есть только в CQM1-CPU4_-EV1.

Описание

Когда условие исполнения = 0, CPSL(-) не выполняется. Когда условие исполнения = 1, CPSL(-) сравнивает 32-битовые (8 цифр) двоичные слова со знаком Cp+1,Cp1 и Cp2+1, Cp2 и выдает результат в флаги GR, EQ, LE в области SR .

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 Cp1+1, Cp1 равно Cp2+1,Cp2.

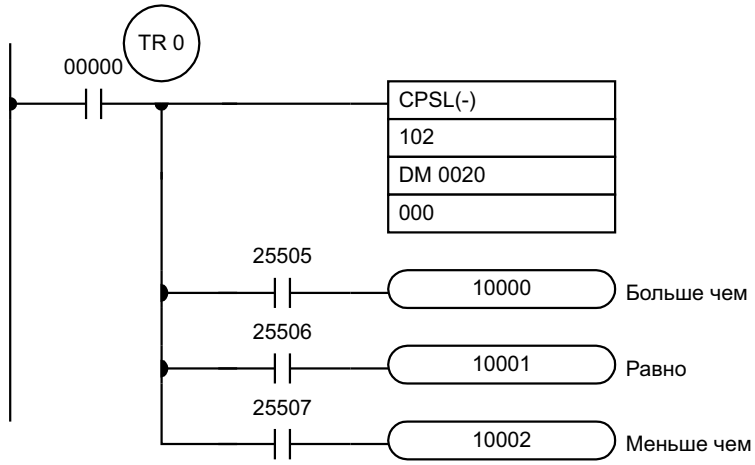
LE: 1, если Cp1+1, Cp1 меньше Cp2+1, Cp2.

GR: 1, если Cp1+1, Cp1 больше Cp2+1, Cp2.

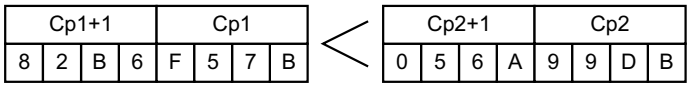
Результат сравнения	**Флаг**		
	GR	**EQ**	**LE**
Cp+1, Cp1 < Cp2+1, Cp2	0	0	1
Cp+1, Cp1 = Cp2+1, Cp2	0	1	0
Cp+1, Cp1 > Cp2+1, Cp2	1	0	0

Пример

В следующем примере содержимое 103, 102 меньше содержимого DM 021, DM 0020, так что 10002 устанавливается в 1, а другие биты, 10000 и 10001, устанавливаются в 0.



Адрес	Инструкция	Операнд
00000	LD	00500
00001	OUT	TR 0
00002	CPS(-)	
		102
		DM 0020
		000
00003	AND	25505
00004	OUT	10000
00005	LD	TR 0
00006	AND	25506
00007	OUT	10001
00008	LD	TR 0
00009	AND	25507
00010	OUT	10002

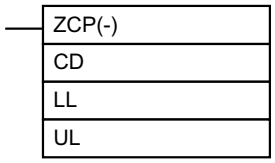


-2101938823
десятичное

90872283
десятичное

5.18.8 ZSP(-) - Сравнение с зоной

Обозначение на схеме



Область операндов

CD	сравниваемые данные	IR, SR, AR, HR, TC, LR, DM, #
LL	нижняя граница зоны	IR, SR, AR, HR, TC, LR, DM, #
UL	верхняя граница зоны	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

Данная команд есть только в CQM1-CPU4_-EV1.

LL должна быть меньше либо равна UL.

Описание

Когда условие исполнения = 0, ZSP(-) не выполняется. Когда условие исполнения = 1, ZSP(-) сравнивает CD с зоной, заданной нижней границей LL и верхней границей UL, и выдает результат в флаги GR, EQ, LE в области SR . Результирующее состояние флагов приведено в таблице.

Результат сравнения	Флаг		
	GR	EQ	LE
CD < LL	0	0	1
LL≤CD≤UL	0	1	0
UL<CD	1	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)

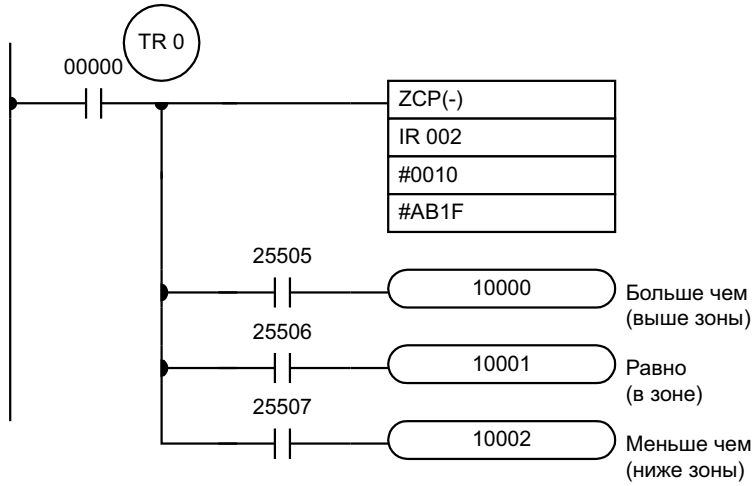
EQ: 1, если LLCDUL

LE: 1, если CD

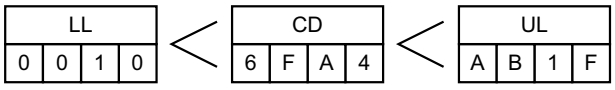
GR: 1, если CDUL.

Пример

В следующем примере содержимое IR 102 (#6FA4) сравнивается с зоной #0010..#AB1F. Поскольку #0010 #6FA4 #AB1F, флаг EQ и IR 10001 установятся в 1.



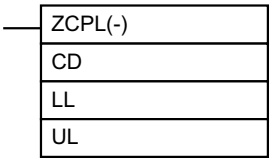
Адрес	Инструкция	Операнд
00000	LD	00500
00001	OUT	TR 0
00002	ZCP(-)	
		002
		# 0010
		# AB1F
00003	AND	25505
00004	OUT	10000
00005	LD	TR 0
00006	AND	25506
00007	OUT	10001
00008	LD	TR 0
00009	AND	25507
00010	OUT	10002



10000: 0(OFF)
10001: 1(ON)
10002: 0(OFF)

5.18.9 ZCPL(-) - Сравнение с зоной чисел двойной длины

Обозначение на схеме



Область операндов

CD	сравниваемые данные	IR, SR, AR, HR, LR, DM
LL	нижняя граница зоны	IR, SR, AR, HR, LR, DM
UL	верхняя граница зоны	IR, SR, AR, HR, LR, DM

Ограничения

Данная команд есть только в CQM1-CPU4_-EV1.

Значение LL+1,LL (8 цифр) должно быть меньше либо равно UL+1, UL.

Описание

Когда условие исполнения = 0, ZSPL(-) не выполняется. Когда условие исполнения = 1, ZSPL(-) сравнивает 8-разрядное число CD+1,CD с зоной, заданной нижней границей LL+1,LL и верхней границей UL+1,UL и выдает результат в флаги GR, EQ, LE в области SR . Результирующее состояние флагов приведено в таблице.

Результат сравнения	**Флаг**		
	GR	**EQ**	**LE**
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

GR: 1, если CD,CD+1UL+1,UL

5.19 Команды преобразования

5.19.1 BIN(23) - Преобразование двоично-десятичного числа в двоичное)

Обозначение на схеме

BIN(23)
S
R

@BIN(23)
S
R

Область операндов

S	слово источник (BCD)	IR, SR, AR, HR, TC, LR, DM
R	слово результата	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, BIN(23) не выполняется. Когда условие исполнения = 1, BIN(23) преобразует двоично-десятичное содержание S в двоичный эквивалент и помещает двоичное значение в R. Изменяется только содержание R, содержание S не изменяется.

BIN(23) можно использовать 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(23) можно также использовать 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.

5.19.2 BCD(24) - Преобразование из двоичного вида в двоично-десятичный

Обозначение на схеме

BCD(24)
S
R

@BCD(24)
S
R

Область операндов

S	слово источник (двоичное)	IR, SR, AR, HR, LR, DM
R	слово результата	IR, SR, AR, HR, LR, DM

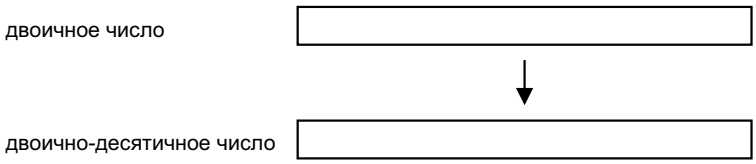
Ограничения

Если содержимое S превышает 270F, преобразуемый результат будет больше 9999 и BCD(24) не выполнится. Когда команда не выполняется, содержание R остается неизменным.

DM 6144..DM 6655 нельзя использовать для R.

Описание

BCD(24) 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(24) можно использовать 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(24) можно также использовать 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.

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 0.

5.19.3 BINL(58) - Преобразование двоично-десятичного числа двойной длины в двоичное двойной длины

Обозначение на схеме

BINL(58)
S
R

@BINL(58)
S
R

Область операндов

S	первое слово источника (BCD)	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM

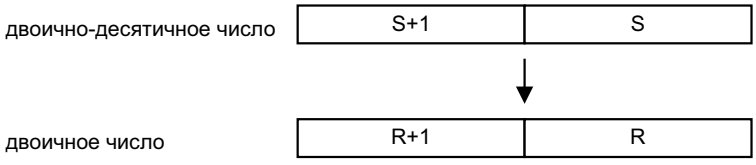
Ограничения

Данная команда есть только у CQM1.

DM 6143..DM 6655 нельзя использовать для R

Описание

Когда условие исполнения = 0, BINL(58) не выполняется. Когда условие исполнения = 1, BINL(58) преобразует 8-разрядное число (S и S+1) в 32-битовое двоичное число и помещает преобразованное 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.

5.19.4 BCDL(59) - Преобразование двоичного числа двойной длины в двоично-десятичное число двойной длины

Обозначение на схеме

BCDL(59)
S
R

@BCDL(59)
S
R

Область операндов

S	первое слово источника (двоичное)	IR, SR, AR, HR, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

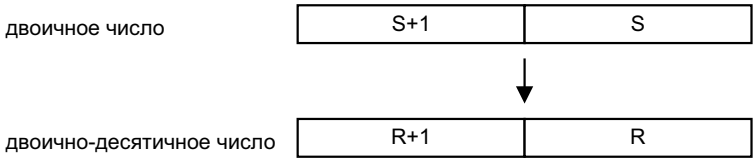
Данная команда есть только у CQM1.

Если содержимое S превышает 05F5 E0FF, результат будет больше 9999 9999 и BCDL(59) не выполнится. Когда команда не выполняется, содержание R и R+1 не изменяется.

DM 6143..DM 6655 нельзя использовать для R

Описание

BCDL(59) преобразует двоичное 32-битовое содержимое S и S+1 в 8 двоично-десятичных цифр и помещает преобразованные данные в R и R+1.



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.

5.19.5 MLPX(76) - Декодер 4-в-16

Обозначение на схеме

MLPX(76)
S
Di
R

@MLPX(76)
S
Di
R

Область операндов

S	слово источника	IR, SR, AR, HR, TC, LR, DM
Di	определитель цифры	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

Две правые (младшие) цифры DI должны быть в диапазоне 0..3.

Все слова результата должны находиться в одной области памяти.

DM 6144..DM 6655 нельзя использовать для R

Описание

Когда условие исполнения = 0, MLPX(76) не выполняется. Когда условие исполнения = 1, MLPX(76) 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,

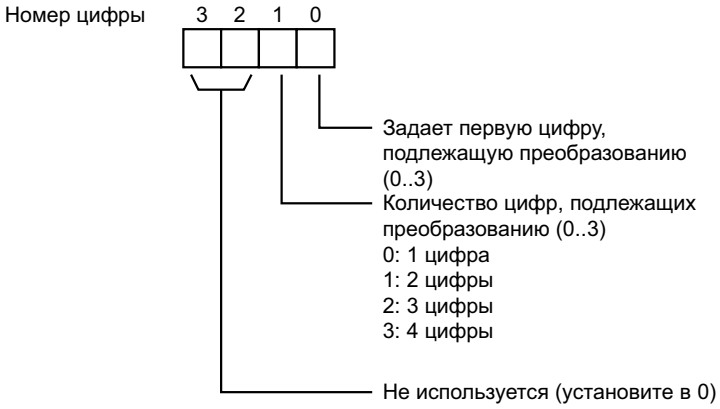
устанавливается в 1. Если задано более одной цифры, тогда один бит будет установлен в 1 в каждом из последовательных слов, начинающихся с R. (см. пример ниже).

Далее приведен пример операции декодирования одной цифры (цифры номер 1 в слове S, т.е. здесь DI = 0001).

Первая цифра и количество цифр для преобразования указаны в DI. Если задано больше цифр, чем остается в S (считая от первой указанную цифру), 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 нельзя назначать. Если преобразуются три цифры, два последних слова в области данных нельзя назначать.

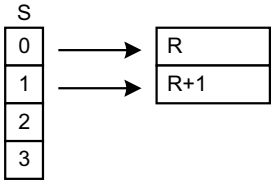
Описатель цифр

Цифры в DI задаются следующим образом.

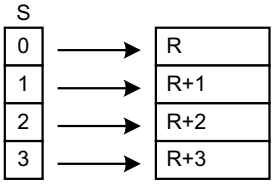


Далее приведены примеры значений DI и преобразования цифра-в-слово, которые производится при данных значениях.

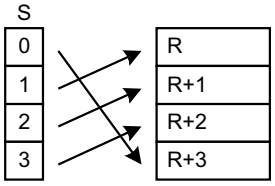
Di: 0010



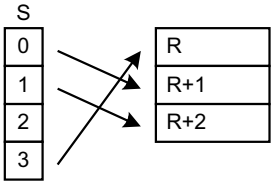
Di: 0030



Di: 0031



Di: 0023



Флаги

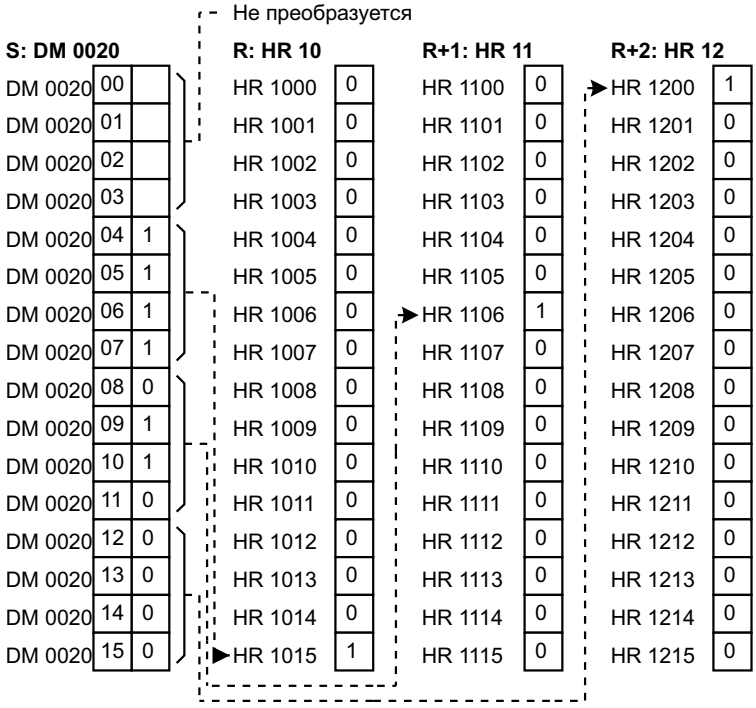
ER: Неопределенный определитель цифр, или R + количество цифр выходит за границы области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

Пример

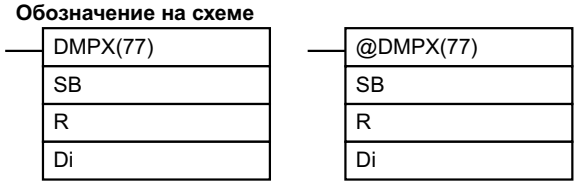
Следующая программа преобразует цифры 1..3 из DM 0020 в позицию битов и устанавливает в 1 соответствующие биты в трех последовательных словах, начинающихся с HR 10. Цифра 0 не преобразуется.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	MLPX(76)	
		DM 0020
		# 0021
		HR 10



5.19.6 DMPX(77) - Преобразовать 16-в-4

Обозначение на схеме



Область операндов

SB	первое слово источника	IR, SR, AR, HR, TC, LR, DM
R	слово результата	IR, SR, AR, HR, LR, DM
Di	определитель цифры	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

Две правые (младшие) цифры DI должны быть в диапазоне 0..3.

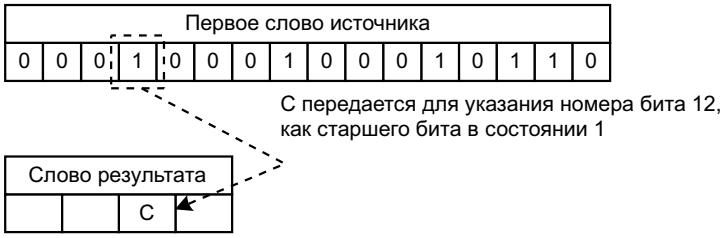
Все слова источника должны находиться в одной области памяти.

DM 6144..DM 6655 нельзя использовать для SB, R или DI.

Описание

Когда условие исполнения = 0, DMPX(77) не выполняется. Когда условие исполнения = 1, DMPX(77) определяет позицию старшего бита, который установлен в 1, слова S и кодирует его в одну 16-ричную цифру, соответствующую номеру старшего бита, установленного в 1, затем передает 16-ричное значение в заданную цифру в R. Цифры, которым передаются результат, задаются в DI, которое также задает количество преобразуемых цифр.

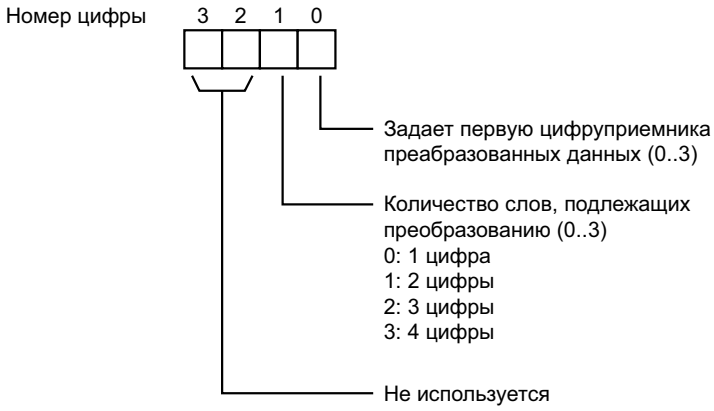
Далее приведен пример операции кодирования одной цифры в цифру номер 1 слова R,(т.е. здесь DI = 0001).



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, (SB плюс количество цифр, подлежащих преобразованию) должно быть в той же области, что и SB.

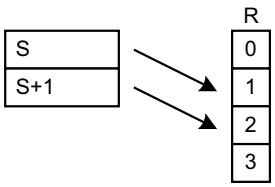
Описатель цифр

Цифры в DI расположены следующим образом.

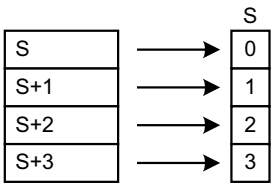


Далее приведено несколько примеров значений DI и преобразований слово-в-цифру, которые производится при данных значениях.

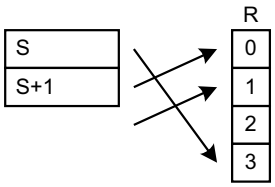
Di: 0011



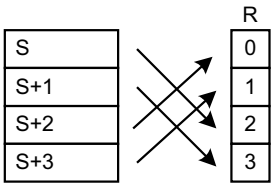
Di: 0030



Di: 0013



Di: 0023



Флаги

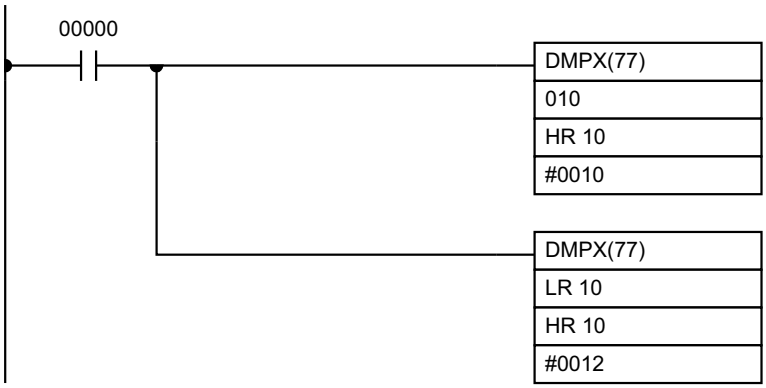
ER: Неопределенный определитель цифр, или S + количество цифр выходит за границы области данных.

Содержание слова источника равно нулю

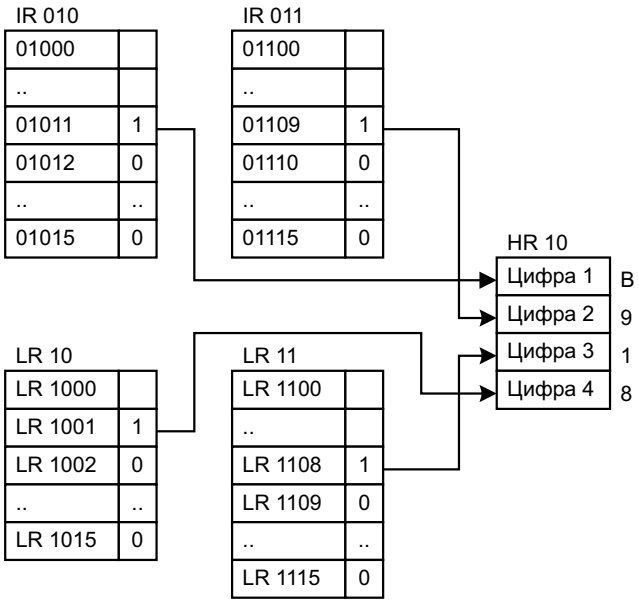
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

Пример

Когда 00000 = 1, следующая релейно-контактная схема кодирует слова IR 010 и IR 011 в две первые цифры HR 10 и далее кодирует LR 10 и LR 11 в 2 последние цифры HR 10. Хотя состояние каждого бита слов источника не показано, предполагается, что бит в состоянием 1 является старшим битом в состоянии 1 в слове.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	DMPX(77)	
		010
		HR 10
		# 0010
00002	DMPX(77)	
		LR 10
		HR 10
		# 0012



5.19.7 SDEC(78) - 7-сегментный декодер

Обозначение на схеме

SDEC(78)
S
Di
D

@SDEC(78)
S
Di
D

Область операндов

S	слово источника (двоичное)	IR, SR, AR, HR, TC, LR, DM
Di	определитель цифры	IR, SR, AR, HR, TC, LR, DM, #
D	первое слово приемника	IR, SR, AR, HR, LR, DM

Ограничения

DI должны быть в далее приведенном диапазоне.

Все слова приемника должны находиться в одной области памяти.

DM 6144..DM 6655 нельзя использовать для D.

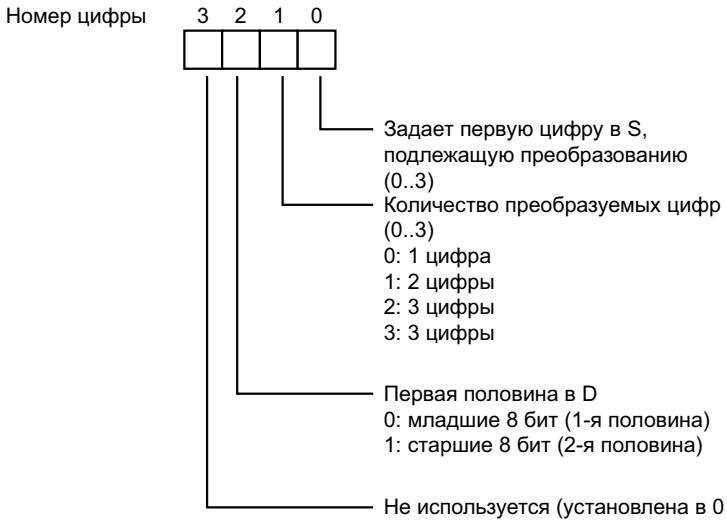
Описание

Когда условие исполнения = 0, SDEC(78) не выполняется. Когда условие исполнения = 1, SDEC(78) преобразует указанные цифры (цифру) слова S в 8-битовый эквивалент, код 7-сегментного дисплея и помещает его в слово(слова) приемника, начинающиеся с D.

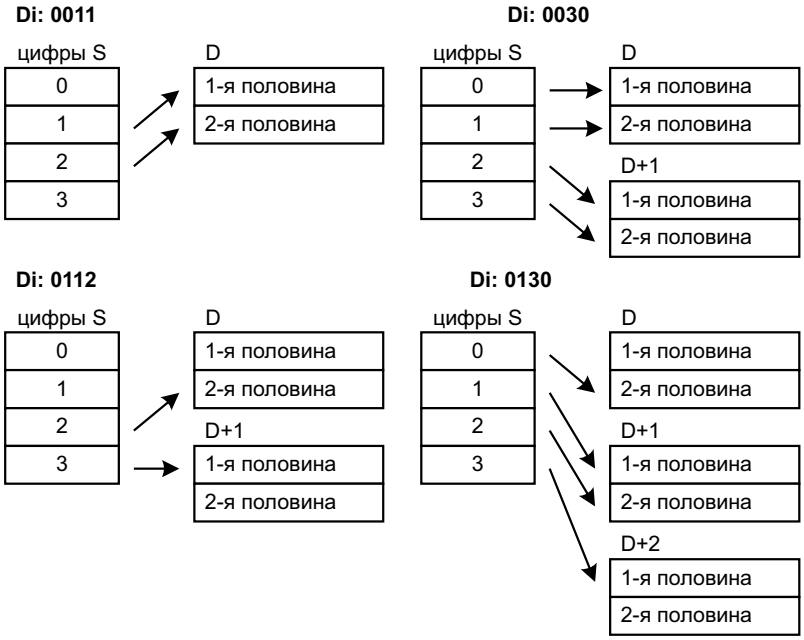
Любая или все цифры в S преобразуются последовательно, начиная от заданной первой цифры. Первая цифра, количество цифр, подлежащих преобразованию, и половина D, принимающая первый код 7-сегментного дисплея (старшие или младшие 8 бит) указаны в DI. Если заданы несколько цифр, они должны быть помещены 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 задаются следующим образом.

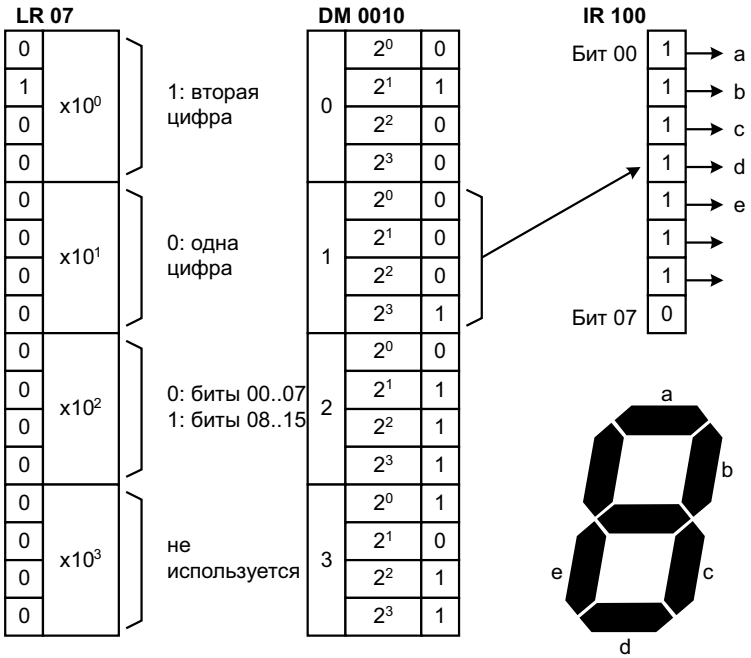


На рисунке приведены примеры значения DI и результаты преобразования 4-битовых двоичных значений в коды 7-сегментного индикатора, которые производится при данных значениях.



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	1	1	1	1	0
1	0	0	0	1	0	0	0	0	0	1	1	0	1
2	0	0	1	0	0	1	0	1	1	0	1	1	2
3	0	0	1	1	0	1	0	0	1	1	1	1	3
4	0	1	0	0	0	1	1	0	0	1	1	0	4
5	0	1	0	1	0	1	1	0	1	1	0	1	5
6	0	1	1	0	0	1	1	1	1	1	0	1	6
7	0	1	1	1	0	0	1	0	0	1	1	1	7
8	1	0	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	0	1	1	0	1	1	1	1	9
A	1	0	1	0	0	1	1	1	0	1	1	1	A
B	1	0	1	1	0	1	1	1	1	1	0	0	b
C	1	1	0	0	0	0	1	1	1	0	0	1	C
D	1	1	0	1	0	1	0	1	1	1	1	0	d
E	1	1	1	0	0	1	1	1	1	0	0	1	E
F	1	1	1	1	0	1	1	1	0	0	0	1	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)

5.19.8 ASC(86) - Преобразование в коды ASCII

Обозначение на схеме

ASC(86)
S
Di
D

@ASC(86)
S
Di
D

Область операндов

S	слово источник	IR, SR, AR, HR, TC, LR, DM
Di	определитель цифры	IR, SR, AR, HR, TC, LR, DM, #
D	первое слово приемника	IR, SR, AR, HR, LR, DM

Ограничения

Значения DI должны быть в далее приведенном диапазоне.

Все слова приемника должны находиться в одной области данных.

DM 6144..DM 6655 нельзя использовать для D.

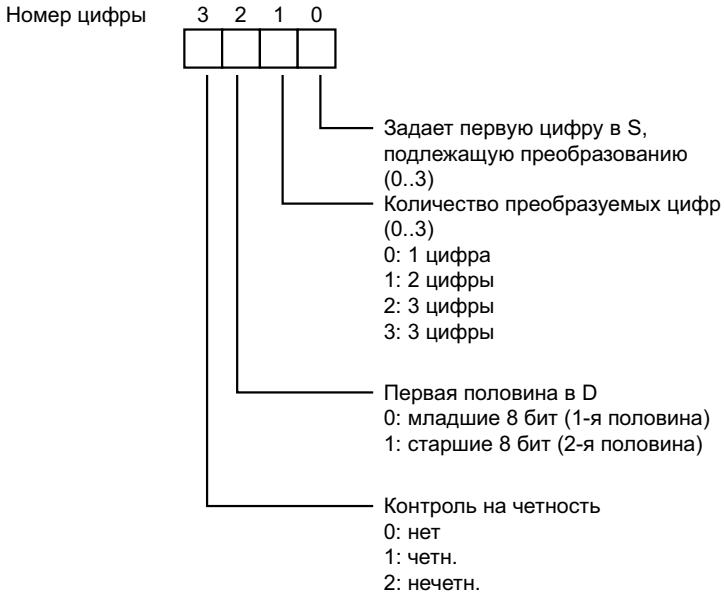
Описание

Когда условие исполнения = 0, ASC(86) не выполняется. Когда условие исполнения = 1, ASC(86) преобразует указанные цифры (цифру) слова S в 8-битовый эквивалент кода ASCII и помещает его в слово(слова) приемника, начинающиеся с D.

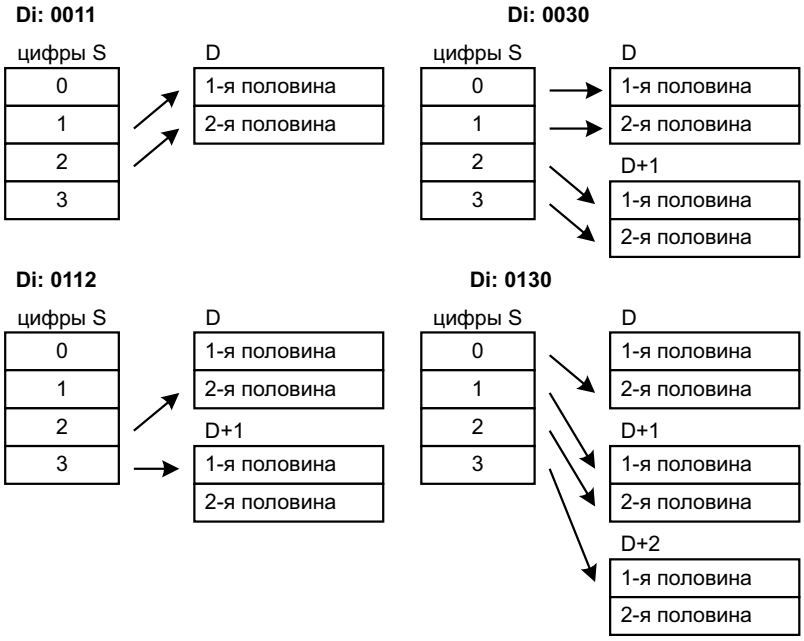
Любая или все цифры в S преобразуются последовательно, начиная от указанной первой цифры. Первая цифра, количество цифр, подлежащих преобразованию, и половина D, принимающая первый код ASCII (старшие или младшие 8 бит) указаны в DI. Если заданы несколько цифр, они должны быть помещены в порядке, начиная 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.

Описатель цифр

Цифры в DI расположены следующим образом.



Далее приведены примеры значений DI и преобразования 4-битовых двоичных значений в 8-битовые коды ASCII, которые производится при данных значениях.



Контроль на четность

Левый (старший) бит каждого символа ASCII (2 цифры = 8 бит) может быть автоматически настроен на четность или нечетность. Если контроль по четности не задан, старший бит всегда = 0.

Когда контроль четности задан на четно, левый (старший) бит будет принимать такое значение, чтобы общее количество битов, находящихся в состоянии 1, было четным, например, при настройке на четн, ASCII “31” (00110001) будет равным “B1” (10110001): бит четности установится в 1 для того, чтобы общее количество битов в состоянии 1 было четным. 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, левый (старший) бит будет принимать такое значение, чтобы общее количество битов, находящихся в состоянии 1, было нечетным,

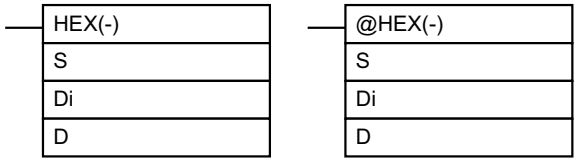
Флаги:

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)

5.19.9 HEX(-) - Преобразование из ASCII в 16-ричное число

Обозначение на схеме



Область операндов

S	первое слово источника	IR, SR, AR, HR, TC, LR, DM, #
Di	определитель цифры	IR, SR, AR, HR, TC, LR, DM, #
D	слово приемника	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1/SRM1.

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).

DM 6144..DM 6655 нельзя использовать для D.

Описание

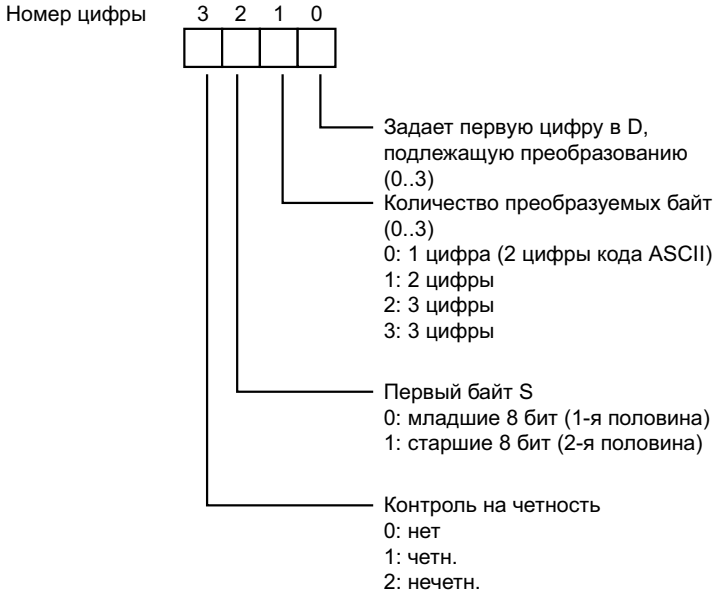
Когда условие исполнения = 0, HEX(-) не выполняется. Когда условие исполнения = 1, HEX(-) преобразует указанные бит(биты) кода ASCII из слова(слов) источника в 16-ричный эквивалент и помещает его в D.

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 необходим, S+1 и S+2.

Если задано больше цифр, чем осталось в D, (считая от заданной первой цифры), последующие цифры будут использоваться, начиная с D. Цифры в D, которые не получают преобразованных данных, не изменятся.

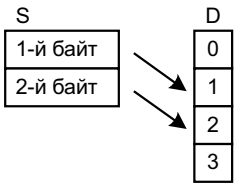
Описатель цифр

Цифры в DI заданы следующим образом.

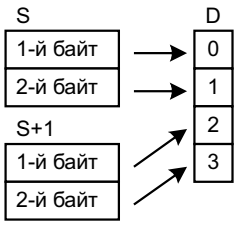


Далее приведены примеры значений DI и результаты преобразования 8 битовых кодов ASCII в 4-битовые 16-ричные, которые производится при данных значениях.

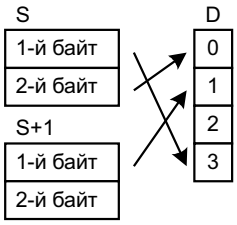
Di: 0011



Di: 0030



Di: 0023



Di: 0133

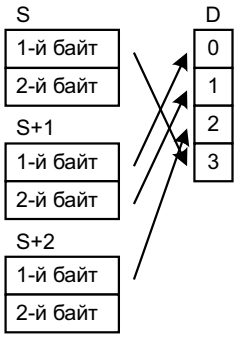


Таблица кодов ASCII

В таблице приведены коды ASCII перед преобразованием и 16-ричные значения после преобразования. Таблицу кодов ASCII см. Приложение H.

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
31	*	0	1	1	0	0	0	1	1	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	1	A	1	0	1	0
42	*	1	0	1	0	0	1	0	B	1	0	1	1
43	*	1	0	1	0	0	1	1	C	1	1	0	0
44	*	1	0	1	0	1	0	0	D	1	1	0	1
45	*	1	0	1	0	1	0	1	E	1	1	1	0
46	*	1	0	1	0	1	1	0	F	1	1	1	1

Замечание Левый (старший) бит каждого кода ASCII настраивается на контроль четности.

Контроль четности

Левый (старший) бит каждого символа ASCII (2 цифры = 8 бит) автоматически настраивается на четность или нечетность.

Когда контроль четности не задан, левый (старший) бит будет всегда установлен в 0. Когда контроль четности установлен на нечетно или четно, старший бит каждого символа ASCII будет принимать такое значение, чтобы общее количество битов, находящихся в состоянии 1, было нечетным или четным

Если контроль по четности кода ASCII в S не совпадает с заданным в DI, флаг ошибки (SR 25503) устанавливается в 1 и команда не выполнится.

Флаги

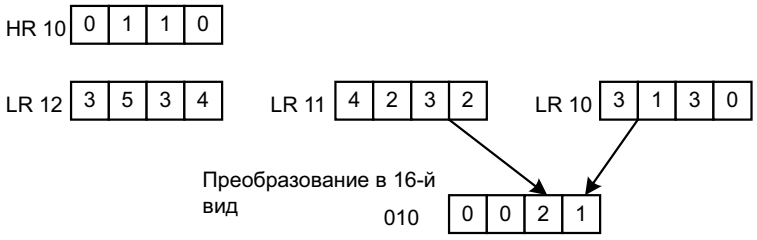
ER: Некорректный определитель цифр или адрес приемника выходит за границы области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

Пример

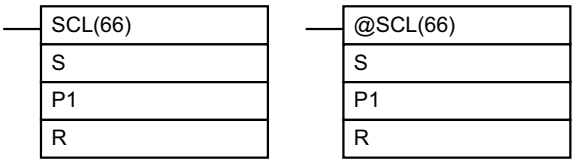
В следующие примере 2-й байт LR 10 и 1-й байт LR 11 преобразуются в 16-ричные значения и эти значения записываются в первый и второй байты IR 010.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@HEX(-)	
		LR 10
		HR 10
		010



5.19.10 SCL(66) - Масштабирование

Обозначение на схеме



Область операндов

S	слово источник	IR, SR, AR, HR, TC, LR, DM, #
P1	первое слово параметров	IR, SR, AR, HR, TC, LR, DM
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

S должно быть в двоично-десятичном виде.

P1..P1+3 должны находиться в одной области данных.

DM 6144..DM 6655 нельзя использовать для P1..P1+3 или R.

Описание

SCL(66) служит для линейного преобразования 4-разрядного 16-ричного числа из в 4-разрядное двоично-десятичное число. В отличие он BCD(24), которая преобразует 4-

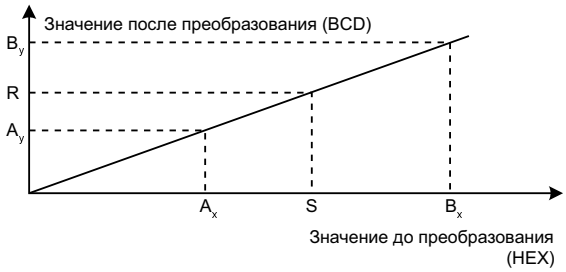
разрядное 16-ричное число в 4-разрядный двоично-десятичный эквивалент (S_{16} - S_{2-10}), SCL(66) может преобразовывать 16-ричное значение с заданным линейным коэффициентом. Линия преобразования задается двумя точками, заданными в словах параметров P1..P1+3.

Когда условие исполнения = 0, SCL(66) не выполняется. Когда условие исполнения = 1, SCL(66) преобразует 4-разрядное 16-ричное значение S в 4-разрядное двоично-десятичное значение с коэффициентом, определяемом на линии, заданной точками (P1, P1+1) и (P1+2, P1+3) и помещает результат в R. Результат округляется до ближайшего целого числа. Если результат меньше 0000, в R заносится 0000, если результат больше 9999, в R заносится 9999.

В следующей таблице показаны функции и диапазоны слов параметров.

Параметр	Функция	Диапазон	Комментарии
P1	Двоично-десятичная точка #1 (A_Y)	0000..9999	-
P1+1	16-ричная точка #1 (A_X)	0000..FFFF	Не задавать P1+1 = P1+3
P1+2	Двоично-десятичная точка #2 (B_Y)	0000..9999	-
P1+3	16-ричная точка #2 (A_X)	0000..FFFF	Не задавать P1+3 = P1+1

На следующем рисунке показаны слово источника S, преобразованное в D по линии, заданной точками (A_Y, A_X) и (B_Y, B_X).



Результаты можно посчитать, преобразовав сначала все значения в двоично-десятичный вид а потом применив формулу:

Результаты = B_Y-[(B_Y-A_Y)/B_X-A_X) x (B_X-S)]

Флаги

ER: Значение P1+1 равно значению P1+3.

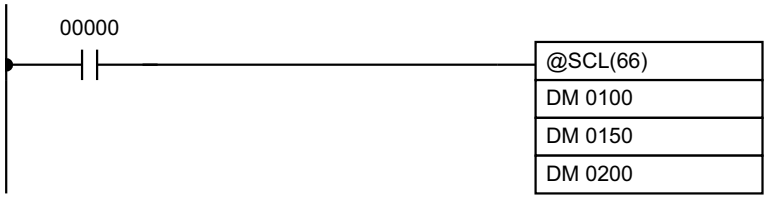
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

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 включится в 1 в следующем примере, данные в двоично-десятичном источнике DM 0100 (#0100) преобразуются в 16-ричные согласно параметров в DM 0150..DM 0153. Результат (#0512) далее записывается в DM 0200.



Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	@SCL(66)	
		DM 0100
		DM 0150
		DM 0200

DM 0150	0010
DM 0151	0005
DM 0152	0050
DM 0153	0019

DM 0100	0100
	↓
DM 0200	0512

5.19.11 SCL2(-) - Преобразование двоичных значений со знаком в двоично-десятичное с масштабированием

Обозначение на схеме

SCL2(-)
S
P1
R

@SCL2(-)
S
P1
R

Область операндов

S	слово источник	IR, SR, AR, HR, LR, DM
P1	первое слово параметров	IR, SR, AR, HR, LR, DM
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1-CPU4_-EV1.

Значения S должны быть в двоично-десятичном виде.

P1..P1+2 должны находиться в одной области данных.

DM 6144..DM 6655 нельзя использовать для R.

Описание

SCL2(-) используется для линейного преобразования 4-разрядного 16-ричного числа со знаком в двоично-десятичное 4-разрядное число. В отличие он BCD(24), которая преобразует 4-разрядное 16-ричное число из 4 цифр в 4-разрядный двоично-десятичный эквивалент (S_{16} - $S_{2\text{-}10}$), SCL2(-) может преобразовывать 16-ричное число со знаком с заданным линейным коэффициентом. Линия преобразования задается пересечением с осью X и наклоном линии, заданным в словах параметров P1..P1+2.

Когда условие исполнения = 0, SCL2(-) не выполняется. Когда условие исполнения = 1, SCL2(-) преобразует 4-разрядное 16-ричное число со знаком из S в 4-разрядное двоично-десятичное число с коэффициентом, определяемом по линии, заданной пересечением с осью X (P1, 0) и наклоном (P1+2 P1+1) и помещает результаты в R. Результат округляется до ближайшего целого числа.

Если результат отрицательный, CY устанавливается в 1.

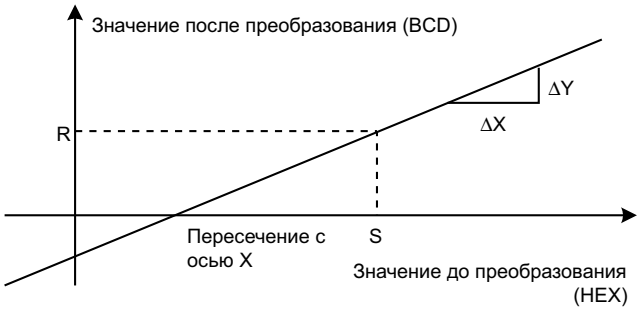
Если результат меньше -9999, в R заносится -9999.

Если результат больше 9999, в R заносится 9999.

В следующей таблице показаны функции и диапазоны слов параметров.

Параметр	Функция	Диапазон
P1	Точка пересечения с осью X (16-ричное число со знаком)	8000 ...7FFF (-32 768..32 767)
P1+1	ΔX (16-ричное число со знаком)	8000..7FFF (-32 768..32 767)
P1+2	ΔY (двоично-десятичное)	0000..9999

На следующей диаграмме показаны слово источника S, преобразованное в R по линии, заданной точкой (P1, 0) и наклоном (Y/ X).



Результат можно посчитать, преобразовав сначала все 16-ричные значения со знаком в двоично-десятичный вид а потом применив формулу:

$$R = \frac{\Delta Y}{\Delta} \times (S - P1)$$

Флаги

ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

P1+1 и P1+2 находятся не в одной области данных или другая ошибка задания.

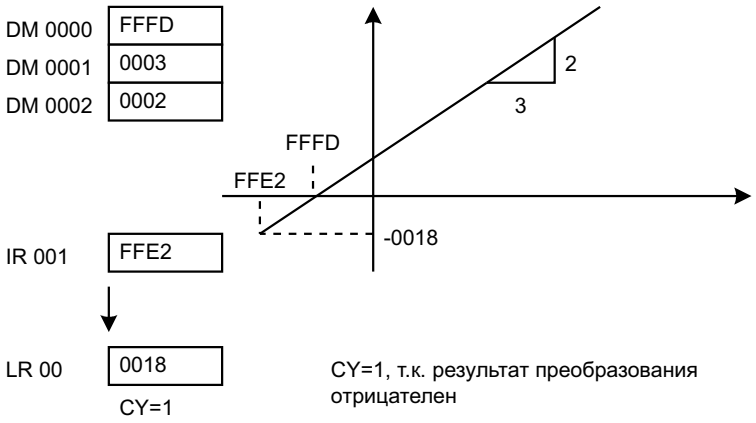
CY: 1, когда результат, R, отрицателен.

EQ: 1, когда результат, R, равен 0000.

Пример

Когда 05000 станет = 0 в следующем примере, двоичные данные со знаком источника 001 (#FFE2) преобразуются в двоично-десятичное число согласно параметров в DM 0000..DM 0002. Результат (#0018) далее заносится в LR 00 и CY включается в 1 из-за того, что результат отрицателен.

Адрес	Инструкция	Операнд
00000	LD	05000
00001	@SCL2	
		001
		DM 0000
		LR 00



5.19.12 SCL3(-) - Масштабирование двоично-десятичных чисел в двоичные со знаком

Обозначение на схеме

SCL3(-)
S
P1
R

@SCL3(-)
S
P1
R

Область операндов

S	слово источник	IR, SR, AR, HR, LR, DM
P1	первое слово параметров	IR, SR, AR, HR, LR, DM
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1-CPU4_-EV1.

Значения P1+1 должны быть в двоично-десятичном виде.

P1..P1+4 должны находиться в одной области данных.

DM 6144..DM 6655 нельзя использовать для R.

Описание

SCL3(-) служит для линейного преобразования 4-разрядного двоично-десятичного числа в 16-ричное 4-разрядное со знаком. SCL3(-)преобразует двоично-десятичное число с заданным линейным соотношением. Линия преобразования задается пересечением с осью Y и наклоном линии, заданными в словах параметров P1 и P1+2.

Когда условие исполнения = 0, SCL3(-) не выполняется. Когда условие исполнения = 1, SCL3(-) преобразует 4-разрядное двоично-десятичное число в слове S в 16-ричное 4-разрядное число со знаком. SCL3(-) преобразует двоично-десятичное число с коэффициентом, определяемом по линии, заданной пересечением с..9999, но если CY = 1, S будет считаться отрицательным числом, так что фактически действующий диапазон S = -9999..9999. Обязательно установите желаемый знак в CY командой STC(40) или CLC(41).

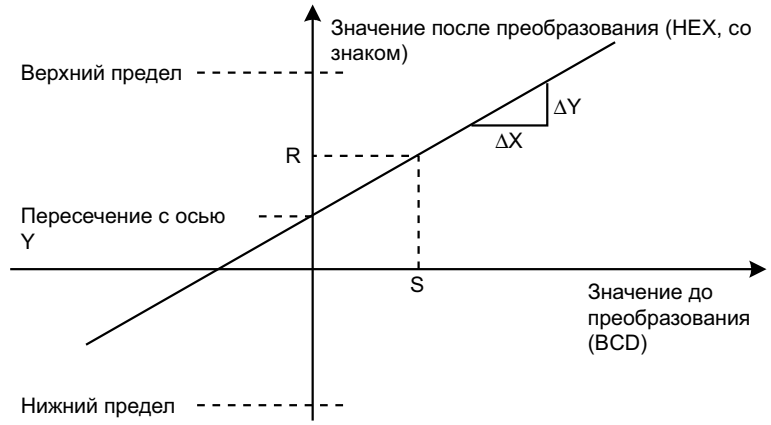
Слова P1+3 и P1+4 определяют верхнюю и нижнюю границы для результата. Если результат больше верхнего предела в P1+3, в R записывается значение верхнего предела. Если результат меньше нижнего предела в P1+4, в R записывается значение нижнего предела.

Замечание Верхний и нижний пределы для 12-битового Блока аналоговых входов будут 07FF и F800.

В таблице показаны функции и диапазоны слов параметров.

Параметр	Функция	Диапазон
P1	Точка пересечения с осью Y (16-ричное число со знаком)	8000 - 7FFF (-32 768 - 32767)
P1+1	X (двоично-десятичное)	0000 - 9999
P1+2	Y(16-ричное число со знаком)	8000 - 7FFF (-32 768 - 32767)
P1+3	Верхняя граница (16-ричное число со знаком)	8000 - 7FFF (-32 768 - 32767)
P1+4	Нижняя граница (16-ричное число со знаком)	8000 - 7FFF (-32 768 - 32767)

На диаграмме показаны слово источника S, преобразованное в R по линии, заданной точкой (P1, 0) и наклоном, равным (X/ Y).



Результат можно посчитать, преобразовав сначала все двоично-десятичные числа в 16-ричные значения со знаком, а потом применив формулу:

$$R = \left(\frac{\Delta Y}{\Delta} \times S \right) + P1$$

Флаги

ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM)

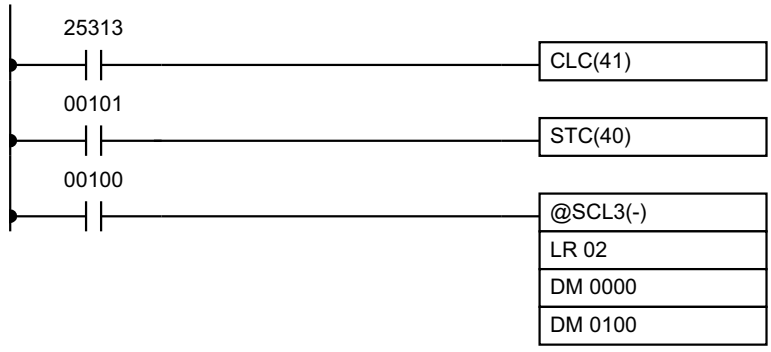
Содержимое S - не в двоично-десятичном виде.

CY: На данный флаг команда воздействия не оказывает. (CY показывает знак S перед исполнением).

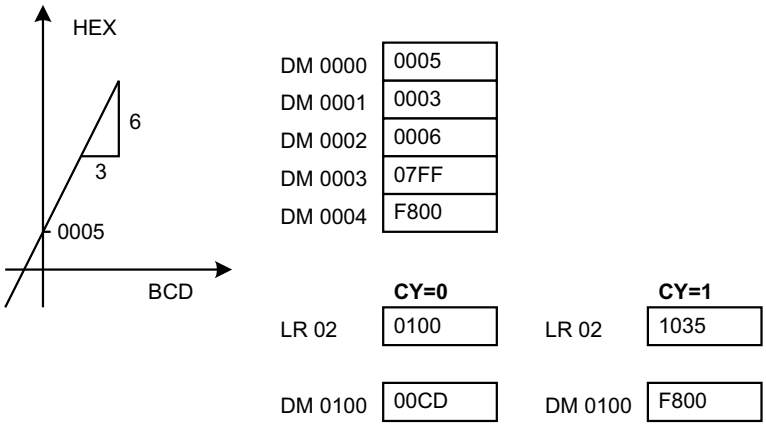
EQ: 1, когда результат, R, равен 0000.

Пример

Состояние 00101 определяет знак двоично-десятичного слова источника в следующем примере. Если 00101 = 1, слово источника отрицательно. Если 00100 = 1, данные из источника (LR 02) преобразуются в двоичный вид со знаком согласно параметрам в DM 0000..DM 0004. Результат заносится в DM 0100. (Во втором преобразовании двоичный эквивалент числа -1035 со знаком меньше нижней границы, заданной в DM 0004, так что в DM 0100 заносится нижняя граница.

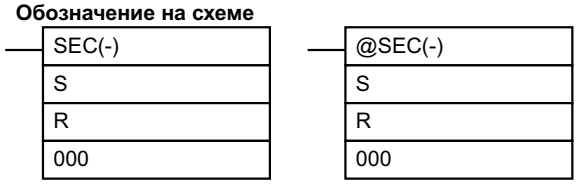


Адрес	Инструкция	Операнд
00000	LD	25313
00001	CLC(41)	
00002	LD	00101
00003	STC(40)	
00004	LD	00100
00005	SCL3(-)	
		LR 02
		DM 0000
		DM 0100



5.19.13 SEC(-) - Преобразование Часы-в-секунды

Обозначение на схеме



Область операндов

S	первое слово источника (BCD)	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата (BCD)	IR, SR, AR, HR, TC, LR, DM
000	функции не имеет	

Ограничения

Данная команда есть только в CQM1.

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.

DM 6143..DM 6655 нельзя использовать для R.

Описание

SEC(-) используется для преобразования времени из формата часы/минуты/секунды в эквивалент в секундах.

В источнике секунды заданы в битах 00..07, минуты в битах 08..15 слова S. Часы заданы 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.

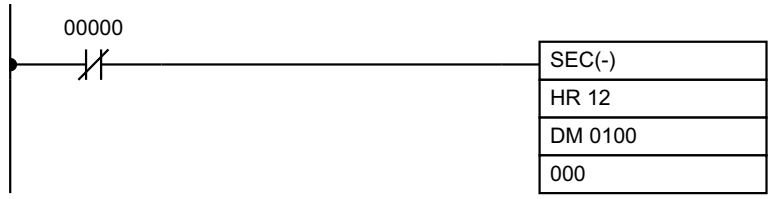
Количество секунд и/или минут превышает 59.

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 результаты в DM 0100 и DM 0101 как показано на рисунке.



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

Адрес	Инструкция	Операнд
00000	LD NOT	00000
00001	SEC(-)	
		HR 12
		DM 0100
		000

5.19.14 HMS(-) - Преобразование Секунды-в-часы

Обозначение на схеме

HMC(-)
S
R
000

@HMC(-)
S
R
000

Область операндов

S	первое слово источника (BCD)	IR, SR, AR, HR, TC, LR, DM
R	первое слово приемника (BCD)	IR, SR, AR, HR, TC, LR, DM
000	функции не имеет	

Ограничения

Данная команда есть только в CQM1.

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 должны иметь значения 0..35 999 999.

DM 6143..DM 6655 нельзя использовать для R.

Описание

HMS(-) служит для преобразования времени из формата секунд в формат часы/минуты/секунды.

Количество секунд, находящиеся в S и S+1, преобразуется в формат часы/минуты/секунды и результат выдается в R и R+1.

В слове результата секунды помещаются в биты 00..07, минуты в битах 08..15 слова R. Часы помещаются 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.

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) следующая 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

Адрес	Инструкция	Операнд
00000	LD NOT	00000
00001	HMC(-)	
		HR 12
		DM 0100
		000

5.19.15 LINE(-) - Столбец-в-строку

Обозначение на схеме

LINE(-)
S
C
D

@LINE(-)
S
C
D

Область операндов

S	первое слово из 16-ти словного блока источника	IR, SR, AR, HR, TC, LR, DM
C	описатель битов столбцов (BCD)	IR, SR, AR, HR, TC, LR, DM, #
D	слово приемника	IR, SR, AR, HR, TC, LR, DM

Ограничения

Данная команда есть только в CQM1.

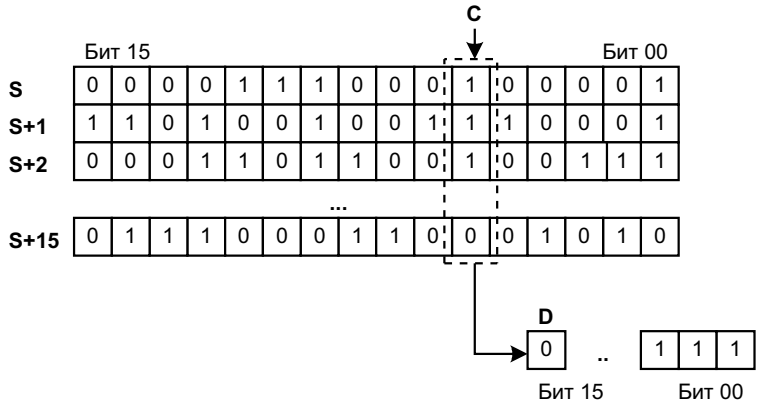
S и S+15 должны находиться в одной области данных.

C должно быть двоично-десятичным числом со значением #0000..#0015.

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, LINE(-) не выполняется. Когда условие исполнения = 1, LINE(-) копирует столбец битов C из 16-словного блока (S..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. значение должно быть 00..15).

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

S и S+15 расположены в разных областях данных.

EQ: 1, когда содержание D равно нулю; в противном случае = 0.

Пример

Следующий пример показывает, как использовать LINE(-) для пересылки битов столбца 07 из блока (IR 100..IR 115) в DM 0100.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	LINE(-)	
		100
		# 0007
		DM 01000

5.19.16 COLM(-) - Строка-в-столбец

Обозначение на схеме

COLM(-)
S
D
C

@COLM(-)
S
D
C

Область операндов

S	слово источник	IR, SR, AR, HR, TC, LR, DM
D	первое слово блока приемника	IR, SR, AR, HR, TC, LR, DM
C	описатель бита столбца	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

Данная команда есть только в CQM1.

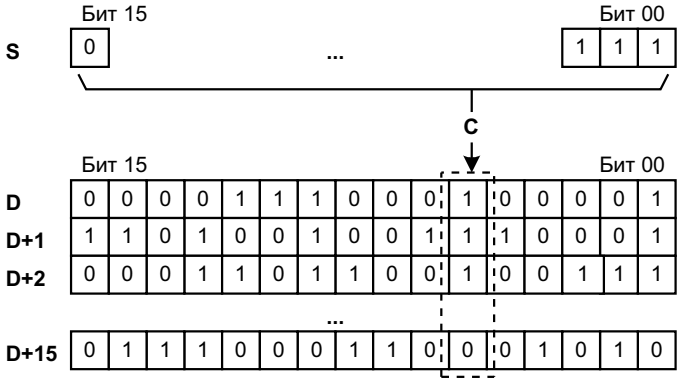
D и D+15 должны находиться в одной области данных.

DM 6129..DM 6655 нельзя использовать для R.

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 (от D до D+15).



Флаги

ER: Описатель битов столбцов C не в двоично-десятичном виде, или определяет несуществующий бит (т.е. должен иметь значение 00..15).

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

D и D+15 расположены в разных областях данных.

EQ: 1, когда содержание S равно нулю; в противном случае = 0.

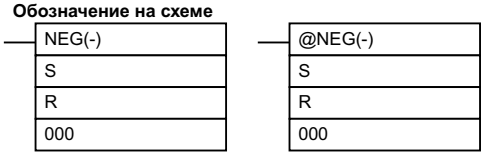
Пример

Следующий пример показывает, как использовать COLM(-) для пересылки содержания слова DM 0100 (00..15) в столбец битов 15 блока DM 0200..DM 0215.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	COLM(-)	
		DM 0100
		DM 0200
		# 0015

5.19.17 NEG(-) - Дополнение до 2

Обозначение на схеме



Область операндов

S	слово источник	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM
000	не используется, установить в 000	

Ограничения

Данная команда есть только в CQM1-CPU4_-EV1.

DM 6144..DM 6655 нельзя использовать для R.

Описание

Преобразует 4-разрядное 16-ричное содержимое источника (S) в дополнение до 2 и выдает результат в слово R. Операция по действию аналогична вычитанию S из 0000 и выдачи результата в R. Она высчитает абсолютное значение отрицательных двоичных данных со знаком.

Если содержание S = 0000, после исполнения содержимое R также будет равным нулю и флаг EQ (SR 25506) установится в 1.

Если содержание S = 8000, после исполнения содержимое R также будет равным 8000 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).

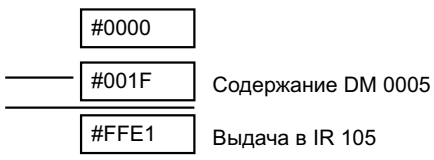
EQ: 1, когда содержание R равно нулю ; в противном случае = 0.

UF: 1, когда содержание S = 8000 ; в противном случае = 0.

Пример

Следующий пример показывает, как использовать NEG(-) для определения дополнения до 2 содержания слова DM 0005 и выдачи результата в IR 105.

Адрес	Инструкция	Операнд
00000	LD	00100
00001	NEG(-)	
		DM 0005
		105
		000



5.19.18 NEGL(-) - Дополнение до 2 слова двойной длины

Обозначение на схеме

NEGL(-)
S
R
000

@NEGL(-)
S
R
000

Область операндов

S	первое слово источника	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM
000	не используется, установить в 000	

Ограничения

Данная команда есть только в CQM1-CPU4_-EV1.

DM 6143..DM 6655 нельзя использовать для R.

S и S+1, должны находиться в одной области данных, как и R и R+1.

Описание

Преобразует 8-разрядное 16-ричное содержимое слов источника (S и S+1) в дополнение до 2 и выдает результат в слова результата R и R+1. Операция по действию аналогична вычитанию 8-разрядного числа S, S+1 из #0000 0000 и выдачи результата в R, R+1. Она будет вычислять абсолютное значение отрицательных двоичных чисел со знаком.

Если содержание S = 0000 0000, после исполнения содержание R также будет 0000 0000 и флаг EQ (SR 25506) установится в 1.

Если содержание S = 8000 0000, после исполнения 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).

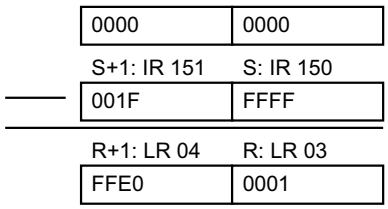
EQ: 1, когда после исполнения содержание R+1, R равно нулю ; в противном случае = 0.

UF: 1, когда содержание S+1, S = 8000 0000; в противном случае = 0.

Пример

Следующий пример показывает, как использовать NEGL(-) для определения дополнения до 2 16-ричного значения в IR 151, IR 150 (001F FFFF) и выдачи результата в HR 04, HR 03.

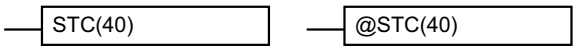
Адрес	Инструкция	Операнд
00000	LD	00000
00001	NEGL	
		150
		LR 03
		000


0000
0000
S+1: IR 151
S: IR 150
001F
FFFF
R+1: LR 04
R: LR 03
FFE0
0001

5.20 Команды двоично-десятичных вычислений

5.20.1 STC(40) - Установка флага переноса

Обозначение на схеме



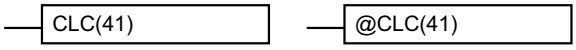
Описание

Когда условие исполнение = 0, STC(40) не выполняется. Когда условие исполнения = 1, STC(40) устанавливает в 1 флаг CY (SR 25504).

Замечание Таблицу с перечислением команд, которые воздействуют на CY, см. Приложении B.

5.20.2 CLC(41) - Очистка флага переноса

Обозначение на схеме



Описание

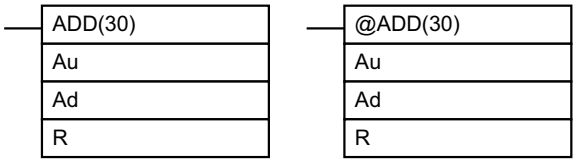
Когда условие исполнение = 0, CLC(41) не выполняется. Когда условие исполнения = 1, CLC(41) устанавливает в 0 флаг CY (SR 25504).

Команда CLC(41) используется для сброса CY (SR 25504) в “0”.

Замечание Таблицу с перечислением команд, которые воздействуют на CY, см. Приложение B.

5.20.3 ADD(30) - Двоично-десятичное сложение

Обозначение на схеме



Область операндов

Au	первое слагаемое (BCD)	IR, SR, AR, HR, TC, LR, DM, #
Ad	второе слагаемое (BCD)	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, ADD(30) не выполняется. Когда условие исполнения = 1, ADD(30) 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).

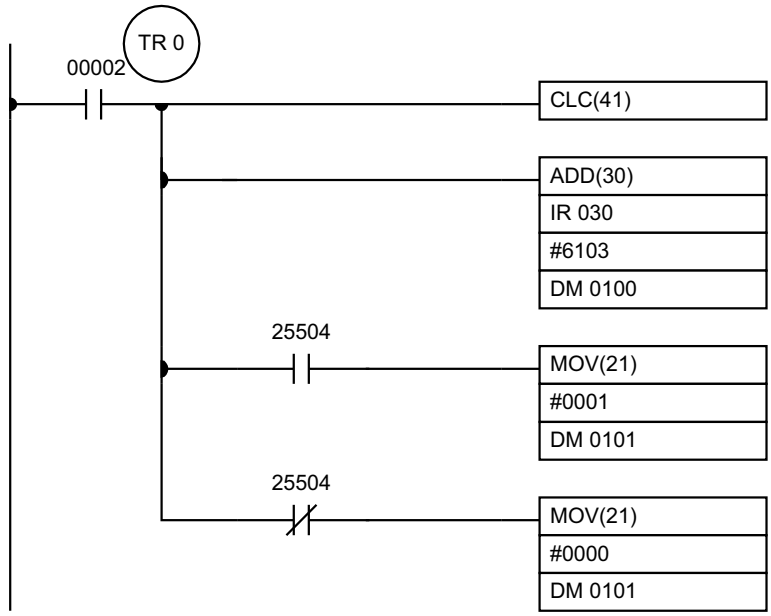
CY: 1, когда есть перенос в результате.

EQ: 1, когда результат равен нулю.

Пример

Если 00002 = 1, программа, представленная на релейно-контактной схеме, очищает CY командой CLC(41), 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 работать с результатом как 8-разрядным числом.



Адрес	**Инструкция**	**Операнд**
00000	LD	00002
00001	OUT	TR 0
00002	CLC(41)	
00003	ADD(30)	
		030
		# 6103
		DM 0100
00004	AND	25504
00005	MOV(21)	
		# 0001
		DM 0101
00006	LD	TR 0
00007	AND NOT	25504
00008	MOV(21)	
		# 0000
		DM 0101

Хотя для сложения 8-разрядного числа можно использовать две команды ADD(30), специально для этой цели предназначена команда ADDL(54).

5.20.4 SUB(31) - Двоично-десятичное вычитание

Обозначение на схеме

SUB(31)
Mi
Su
R

@SUB(31)
Mi
Su
R

Область операндов

Mi	уменьшаемое (BCD)	IR, SR, AR, HR, TC, LR, DM, #
Su	вычитаемое (BCD)	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, SUB(31) не выполняется. Когда условие исполнения = 1, SUB(31) вычитает содержание SU и перенос CY из MI и помещает 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

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(41) перед исполнением SUB(31), если предыдущее состояние не требуется, и проверьте состояние CY после вычитания. Если CY = 1 в результате исполнения SUB(31) (т.е., если результат отрицателен), результат выдается как дополнение до 10 фактического значения. Для преобразования результата в фактическое значения нужно вычесть R из нуля.

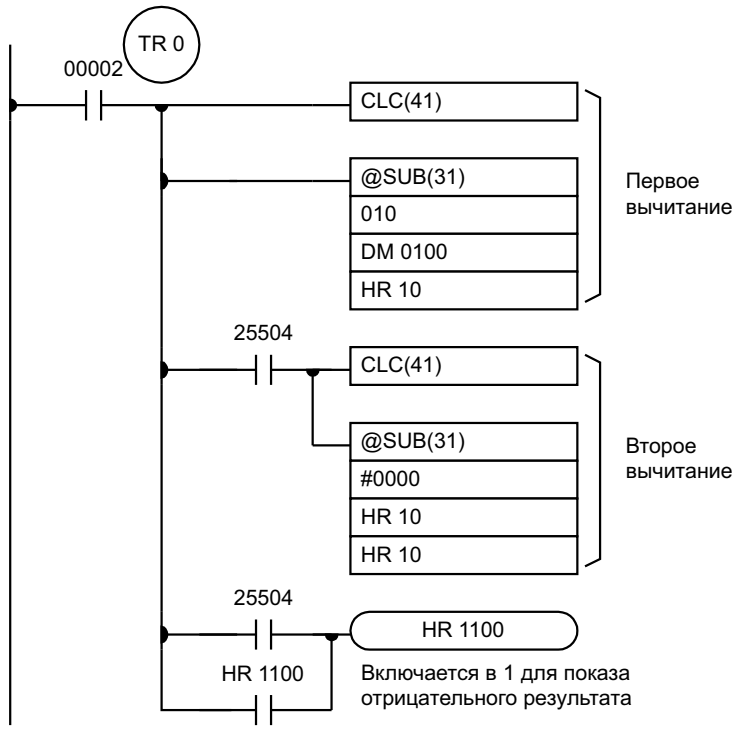
Пример

Если 00002 = 1, программа, представленная на релейно-контактной схеме, очищает CY, вычитает содержимое DM 0100 и CY из содержимого 010 и помещает результат в HR 10.

Если CY установлен после исполнения SUB(31), результат в HR 10 вычитается от нуля (обратите внимание, что снова требуется CLC(41) для получения истинного результата) , результат помещается снова в HR 10, а HR 1100 установлен в 1 для указания того, что результат отрицателен.

Если после исполнения SUB(31) CY не установлен, то результат положителен, второе вычитание не производится, и HR 1100 не устанавливается в 1. HR 1100 запрограммирован как бит самоподдержки, так что изменение состояния CY не установит его в 0 при повторном сканировании программы.

В данном примере применяется вариант SUB(31) фронта 0/1, так что операция вычитания выполняется только раз, каждый раз, когда 0002 устанавливается в 1. Когда требуется новая операция вычитания, требуется хотя бы на 1 цикл установить бит 00002 в состояние 0 (сбросить HR 1100) и затем снова установить в 1.



Адрес	Инструкция	Операнд
00000	LD	00002
00001	OUT	TR 0
00002	CLC(41)	
00003	@SUB(31)	
		010
		DM 0100
		HR 10
00004	AND	25504
00005	CLC(41)	
00006	@SUB(31)	
		# 0000
		HR 10
		HR 10
00007	LD	TR 0
00008	AND	25504
00009	OR	HR 1100
00010	OUT	HR 1100

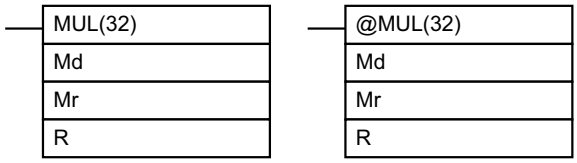
Первое и второе вычитание для этой схемы показаны ниже, они использую данные из примера 010 и DM 0100.

Замечание Текущая операция SUB(31) включает вычитание SU и CY из 10 000 плюс MI. Для положительного результата левая (старшая) цифра отсекается. Для отрицательного результата получается дополнение до 10. Процедура получения правильного ответа показана далее.

В данном случае программа установит HR 1100 в 1 для указания того, что значение в HR 10 отрицательно.

5.20.5 MUL(32) - Двоично-десятичное умножение

Обозначение на схеме



Область операндов

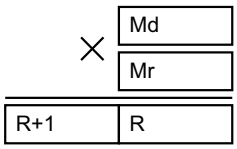
Md	множитель (BCD)	IR, SR, AR, HR, TC, LR, DM, #
Mr	множимое (BCD)	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

DM 6143..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, MUL(32) не выполняется. Когда условие исполнения = 1, MUL(32) умножает MR на MD и посылает результат в R и R+1.



Пример

Если при работе следующей релейно-контактной схемы 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 программы.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	MUL(32)	
		013
		DM 0005
		HR 07

Флаги

ER: MD и/или MR 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.

5.20.6 DIV(33) - Деление двоично-десятичных чисел

Обозначение на схеме

DIV(33)
Dd
Dr
R

@DIV(33)
Dd
Dr
R

Область операндов

Dd	делимое (BCD)	IR, SR, AR, HR, TC, LR, DM, #
Dr	делитель (BCD)	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата, (BCD)	IR, SR, AR, HR, LR, DM

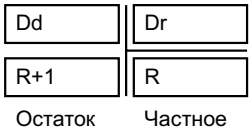
Ограничения

R и R+1 должны лежать в одной области данных.

DM 6143..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, DIV(33) не выполняется. Когда условие исполнения = 1, DIV(33) 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.

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 программы IR 00000 = 1, содержимое IR 216 делится на содержимое HR 09 и результат посылается в DM 0017 и DM 0018. Числовые данные примера и вычисления представлены ниже, после программы.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	DIV(33)	
		216
		HR 09
		DM 0017

Dd: IR 216				Dr: HR 09			
3	4	5	2	0	0	0	3

R+1:DM 0018				R: DM 0017			
0	0	0	2	1	1	5	0

5.20.7 ADDL(54) - Двоично-десятичное сложение чисел двойной длины

Обозначение на схеме

ADDL(54)
Au
Ad
R

@ADDL(54)
Au
Ad
R

Область операндов

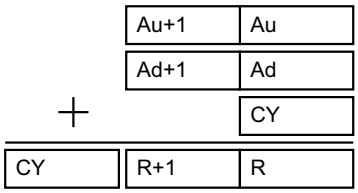
Au	первое слово первого слагаемого (BCD)	IR, SR, AR, HR, TC, LR, DM
Ad	первое слово второго слагаемого (BCD)	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

DM 6143..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, ADDL(54) не выполняется. Когда условие исполнения = 1, ADDL(54) 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).

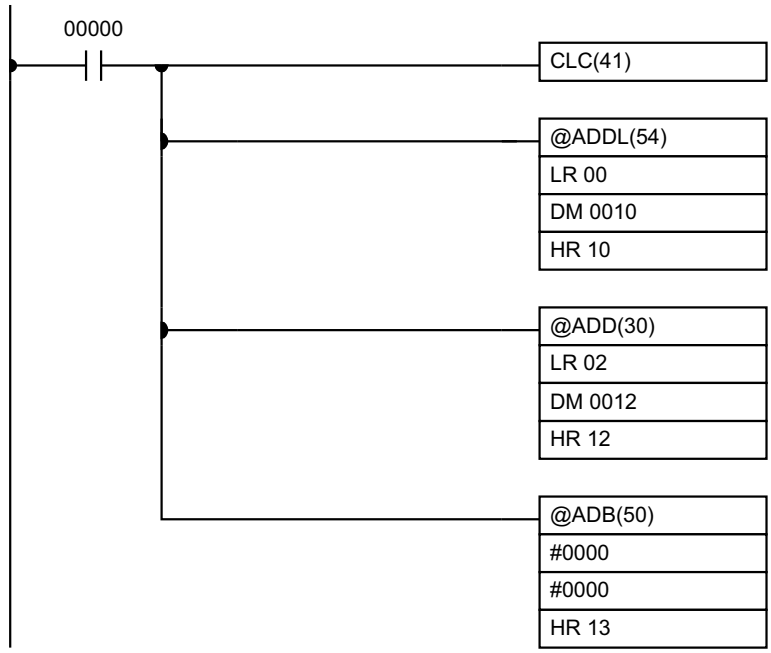
CY: 1, когда есть перенос в результате.

EQ: 1, когда результат равен нулю.

Пример

Если 00002 = 1, программа, представленная на релейно-контактной схеме, складывает два 12-разрядных числа, первое из которых лежит в LR 00..LR 02, а второе в DM 0010..DM 0012. Результат заносится в HR 10..HR 13.

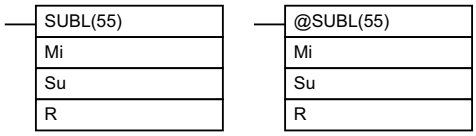
Правые 8 цифр складываются с использованием ADDL(54), т.е. содержимое LR 00, LR 01 складывается с содержимым DM 0010 , DM 0011 и результат заносится в HR 10..HR 11. Вторая операция сложения добавляет 4 старшие цифры каждого числа командой ADD(30) и подключает перенос из первого сложения. Последняя команда, ADB(50) (см. 5-21-1), добавляет две константы со всеми нулями для посылки переноса от второго сложения в HR 13.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	@ADDL(54)	
		LR 00
		DM 0010
		HR 10
00002	@ADD(30)	
		LR 00
		DM 0012
		HR 10
00003	@ADB(50)	
		# 0000
		# 0000
		HR 13

5.20.8 SUBL(55) - Двоично-десятичное вычитание чисел двойной длины

Обозначение на схеме



Область операндов

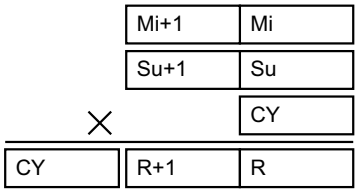
Mi	первое слово уменьшаемого (BCD)	IR, SR, AR, HR, TC, LR, DM
Su	первое слово вычитаемого (BCD)	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

DM 6143..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, SUBL(55) не выполняется. Когда условие исполнения = 1, SUBL(55) 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 помещается 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(71) (5-17-4).



Флаги

ER: MI, MI+1, SU или SU+1 не двоично-десятичное число.

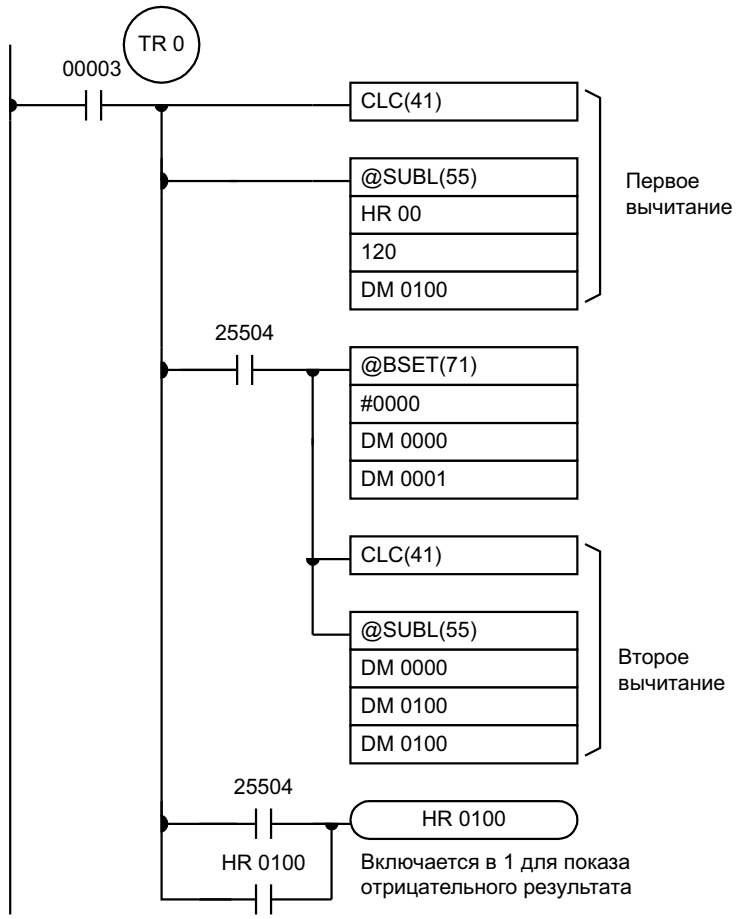
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

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(71), чтобы получить возможность вычитать из нуля отрицательный результат (ввод 8-разрядной константы невозможен).



Адрес	Инструкция	Операнд
00000	LD	00003
00001	OUT	TR 0
00002	CLC(41)	
00003	@SUBL(55)	
		HR 00
		120
		DM 0100
00004	AND	25504
00005	@BSET(71)	
		# 0000
		DM 0000
		DM 0001
00006	CLC(41)	
00007	@SUBL(55)	
		DM 0000
		DM 0100
		DM 0100
00008	LD	TR 0
00009	AND	25504
00010	OR	HR 0100
00011	OUT	HR 0100

5.20.9 MULL(56) - Двоично-десятичное умножение чисел двойной длины

Обозначение на схеме

MULL(56)
Md
Mr
R

@MULL(56)
Md
Mr
R

Область операндов

Md	первое слово множителя (BCD)	IR, SR, AR, HR, TC, LR, DM
Mr	первое слово множимого (BCD)	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

DM 6141..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, MULL(56) не выполняется. Когда условие исполнения = 1, MULL(56) умножает 8-разрядное содержимое MD, MD+1 на MR, MR+1 и посылает результат в R..R+3.

		Md+1	Md
	×	Mr+1	Mr
R+3	R+2	R+1	R

Флаги

ER: MD, MD+1, MR или 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).

CY: 1, когда есть перенос в результате.

EQ: 1, когда результат равен нулю.

5.20.10 DIVL(57) - Двоично-десятичное деление чисел двойной длины

Обозначение на схеме

DIVL(57)
Dd
Dr
R

@DIVL(57)
Dd
Dr
R

Область операндов

Dd	первое слово делимого (BCD)	IR, SR, AR, HR, TC, LR, DM
Dr	первое слово делителя (BCD)	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, DIVL(57) не выполняется. Когда условие исполнения = 1, DIVL(57) 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, DR или 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.

5.20.11 ROOT(72) - Квадратный корень

Обозначение на схеме

ROOT(72)
S
R

@ROOT(72)
S
R

Область операндов

S	первое слово источника (BCD)	IR, SR, AR, HR, TC, LR, DM
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, ROOT(72) не выполняется. Когда условие исполнения = 1, ROOT(72) вычисляет квадратный корень 8-разрядного содержимого SQ,SQ+1 и выдает результат в R. Дробная часть отсекается.

$\sqrt{\text{S +1}\ \text{S}} = \text{R}$

Флаги

ER: SQ не двоично-десятичное число.

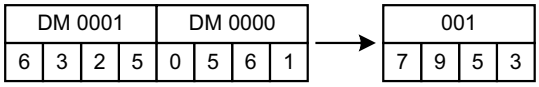
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

EQ: 1, когда результат равен нулю.

Пример

Следующий пример показывает, как взять квадратный корень восьмиразрядного числа. Результатом является 4-разрядное число, округляется остаток, затем число.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@ROOT(72)	
		DM 0000
		001

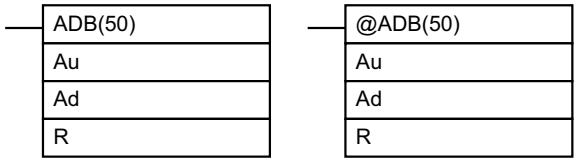


$\sqrt{63250561}$=7953.0221 (результат округляется)

5.21 Команды двоичных вычислений

5.21.1 ADB(50) - Сложение двоичных чисел

Обозначение на схеме



Область операндов

Au	первое слагаемое (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
Ad	второе слагаемое (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, ADB(50) не выполняется. Когда условие исполнения = 1, ADB(50) складывает содержимое слов AU,AD и CY и выдает результат в R. CY установится в 1, когда результат больше FFFF.

ADB(50) можно также использовать для сложения двоичных чисел со знаком. У CQM1-CPU4-E флаги “переполнение -” и “переполнение +” (SR 25405 и SR 25404) указывают, что результат перешел за верхнюю или нижнюю границу 16-разрядного двоичного числа со знаком.

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.

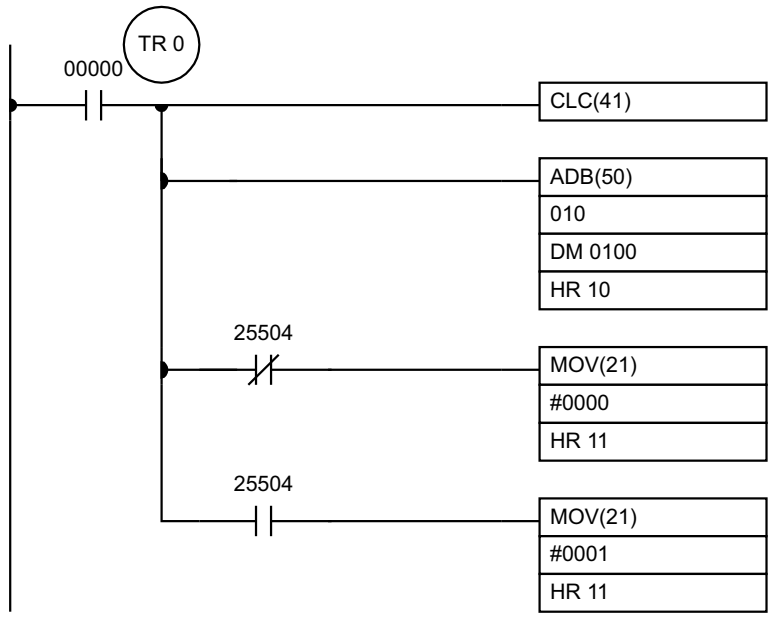
EQ: 1, когда результат равен нулю.

OF: 1, когда результат больше + 32 767 (7FFFF) (только у CQM1-CPU4-E)

UF: 1, когда результат меньше - 32 768 (80000) (только у CQM1-CPU4-E)

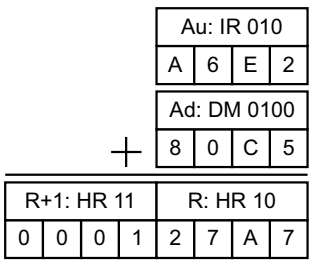
Пример

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	00000
00001	OUT	TR 0
00002	CLC(41)	
00003	ADB(50)	
		010
		DM 0100
		HR 10
00004	AND NOT	25504
00005	MOV(21)	
		# 0000
		HR 11
00006	LD	TR 0
00007	AND	25504
00008	MOV(21)	
		# 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.



Замечание Для двоичных вычислений со знаком состояние флагов UF и OF указывает выходил ли результат за границы двоичных значений со знаком (- 32 768 (8000)..+ 32 767 (7FFFF) (только у CQM1-CPU4-E)

5.21.2 SBB(51) - Вычитание двоичных чисел

Обозначение на схеме

SBB(51)
Mi
Su
R

@SBB(51)
Mi
Su
R

Область операндов

Mi	уменьшаемое (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
Su	вычитаемое (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, SBB(51) не выполняется. Когда условие исполнения = 1, SBB(51) 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.

Mi - Su - CY → CY R

SBB(51) можно также использовать для вычитания двоичных чисел со знаком. У CQM1-CPU4-E флаги переполнение + и переполнение - (SR 25404 и SR 25404) указывают, когда результат перешел за верхнюю или нижнюю границу 16-разрядного двоичного числа со знаком.

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.

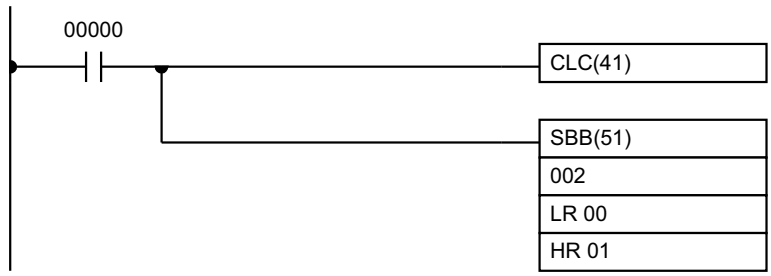
OF: 1, когда результат больше + 32 767 (7FFFF) (только у CQM1-CPU4-E)

UF: 1, когда результат меньше - 32 768 (80000) (только у CQM1-CPU4-E)

Пример

Данный пример показывает вычитание 4 цифр. Когда IR 00001 = 1, содержимое LR 00 и CY вычитаются из содержимого IR 002 и результат записывается в HR 01.

CY устанавливается в 1, если результат отрицателен. Если используются нормальные (беззнаковые) данные, отрицательный результат (двоичное число со знаком) нужно преобразовать в нормальный вид командой NEG(-). Подробности см. 5-19-17.

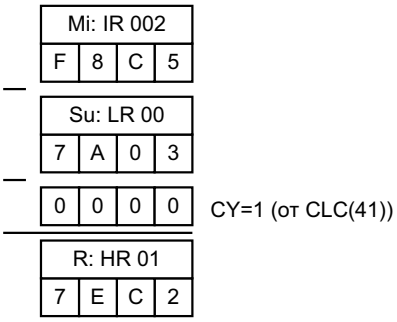


Адрес	Инструкция	Операнд
00000	LD	00001
00001	OUT	TR 1
00002	CLC(41)	

Адрес	Инструкция	Операнд
00003	SBB(51)	
		002
		LR 00
		HR 01

Для следующего случая, содержимое LR 00 (#7A03) и CY вычитаются из IR 002 (#F8C5). Поскольку результат положителен, CY = 0.

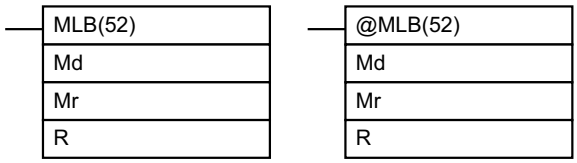
Если бы результат был отрицательным, CY установился бы в 1. Для нормальных (беззнаковых) данных результат нужно преобразовать в дополнение до 2.



Замечание Для двоичных вычислений со знаком состояние флагов UF и OF указывает выходил ли результат за границы двоичных значений со знаком (- 32 768 (8000)..+ 32 767 (7FFFF) (только у CQM1-CPU4-E)

5.21.3 MLB(52) - Умножение двоичных чисел

Обозначение на схеме



Область операндов

Md	множитель (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
Mr	множимое (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

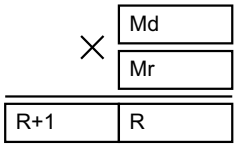
Ограничения

DM 6143..DM 6655 нельзя использовать для R.

MLB(52) нельзя использовать для умножения двоичных чисел со знаком. В CQM1 можно использовать MBS (-). См. 5-21-7.

Описание

Когда условие исполнения = 0, MLB(52) не выполняется. Когда условие исполнения = 1, MLB(52) умножает содержимое MR на содержимое MD и посылает 4 младшие цифры в результата в R и 4 старшие цифры в R+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 результат равен нулю.

5.21.4 DVB(53) - Деление двоичных чисел

Обозначение на схеме

DVB(53)
Dd
Dr
R

@DVB(53)
Dd
Dr
R

Область операндов

Dd	делимое (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
Dr	делитель (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

DM 6143..DM 6655 нельзя использовать для R.

DVB(53) нельзя использовать для умножения двоичных чисел со знаком. В CQM1 можно использовать DBS (-). Подробности см. 5-21-9.

Описание

Когда условие исполнения = 0, DVB(53) не выполняется. Когда условие исполнения = 1, DVB(53) делит содержание DD на содержание DR и посылает результат в R и R+1: частное в R и остаток в R+1.

Dd: IR 216			
3	4	5	2

Dr: HR 09			
0	0	0	3

R+1:DM 0018			
0	0	0	2

R: DM 0017			
1	1	5	0

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.

5.21.5 ADBL(-) - Сложение двоичных чисел двойной длины

Обозначение на схеме

ADBL(-)
Au
Ad
R

@ADBL(-)
Au
Ad
R

Область операндов

Au	первое слово первого слагаемого (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
Ad	первое слово второго слагаемого (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

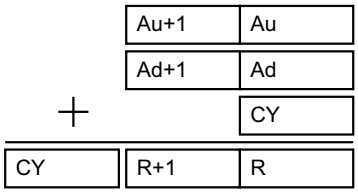
Данная команда есть только в CQM1-CPU4-E.

AU и AU+1 должны лежать в одной области памяти. (То же касается AD и AD+1, R и R+1).

DM 6142..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, ADBL(-) не выполняется. Когда условие исполнения = 1, ADBL(-) складывает 8-разрядное содержимое AU+1 и AU, 8-разрядное содержимое 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.

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.

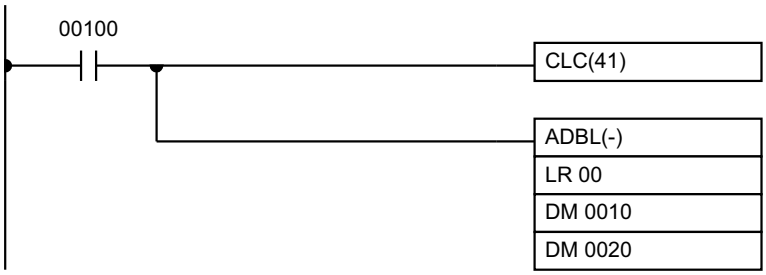
EQ: 1, когда результат равен нулю.

OF: 1, когда результат больше +2 147 483 647 (7FFF FFFF)

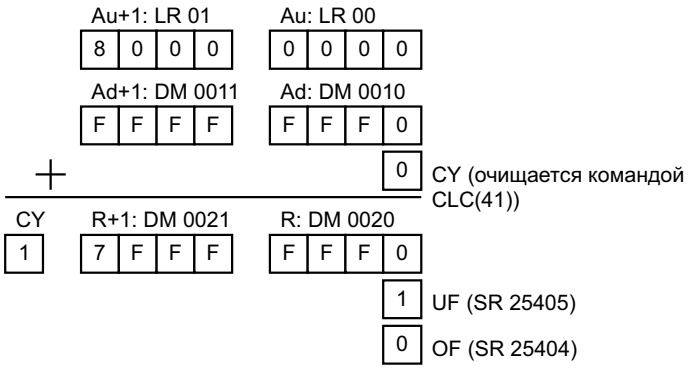
UF: 1, когда результат меньше - 2 147 483 648 (8000 0000)

Пример

Следующий пример показывает сложение 8-разрядных чисел, причем CY (SR 25504) используется для того, чтобы представить состояние 9-й цифры. Состояние флагов UF и OF показывает, не вышел результат за границы зоны двоичных чисел со знаком (от - 2 147 483 648 (8000 0000) до +2 147 483 647 (7FFF FFFF)).



Адрес	Инструкция	Операнд
00000	LD	00100
00001	CLC(41)	
00002	ADBL(-)	
		LR 20
		DM 0010
		DM 0020



Замечание 1. Для беззнакового двоичного сложения CY указывает, что сумма превысила +2 147 483 647 (7FFFF FFFF) (UF и OF можно игнорировать).

2. Для двоичного сложения со знаком флаг UF показывает, что сумма меньше -2 147 483 648 (8000 0000) (CY можно игнорировать).

5.21.6 SBBL(-) - Вычитание двоичных чисел двойной длины

Обозначение на схеме

SBBL(-)
Mi
Su
R

@SBBL(-)
Mi
Su
R

Область операндов

Mi	первое слово уменьшаемого (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
Su	первое слововычитаемого (двоичное)	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

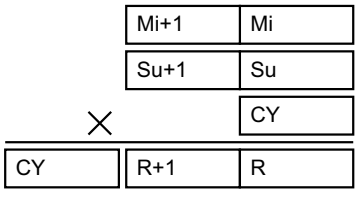
Данная команда есть только в CQM1-CPU4_-E.

MI и MI+1 должны лежать в одной области памяти. (То же касается SU и SU+1, R и R+1).

DM 6142..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 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-битного двоичного числа со знаком.

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.

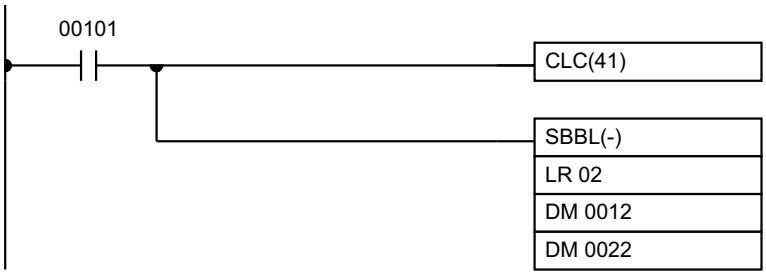
EQ: 1, когда результат равен нулю.

OF: 1, когда результат больше +2 147 483 647 (7FFF FFFF)

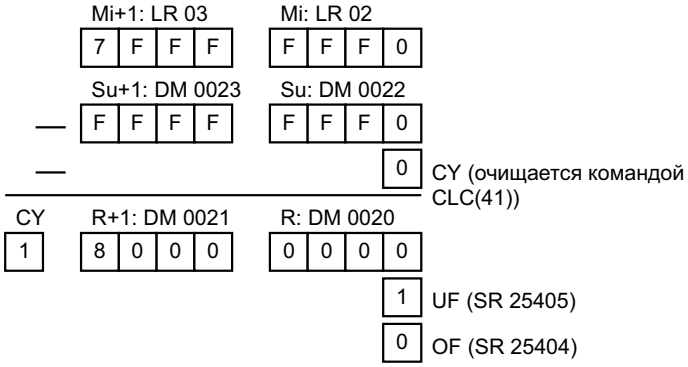
UF: 1, когда результат меньше - 2 147 483 648 (8000 0000)

Пример

Следующий пример показывает вычитание 8-разрядных чисел, причем CY (SR 25504) используется для того, чтобы указать отрицательный результат (с беззнаковыми данными). Состояние флагов UF и OF показывает, не вышел ли результат за границы зоны двоичных чисел со знаком (от - 2 147 483 648 (8000 0000) до +2 147 483 647 (7FFF FFFF)).



Адрес	Инструкция	Операнд
00000	LD	00101
00001	CLC(41)	
00002	SBBL(-)	
		LR 22
		DM 0012
		DM 0022

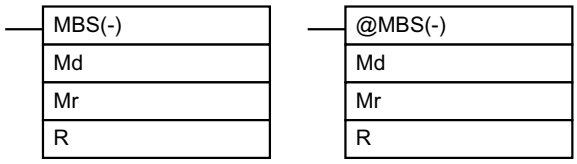


Замечание Для беззнакового двоичного сложения CY указывает, что результат отрицателен. Для получения абсолютного значения истинного результата пользуйтесь командой NEGL(-). (UF и OF можно игнорировать).

Для двоичных данных со знаком флаг OF показывает, что результат превысил +2 147 483 647 (7FFF FFFF) (CY можно игнорировать).

5.21.7 MBS(-) - Умножение двоичных чисел со знаком

Обозначение на схеме



Область операндов

Md	множитель	IR, SR, AR, HR, TC, LR, DM, #
Mr	множимое	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

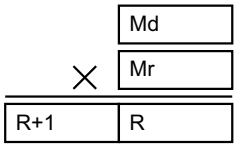
Ограничения

Данная команда есть только в CQM1-CPU4_-E.

DM 6143..DM 6655 нельзя использовать для R.

Описание

MBS(-) умножает двоичное содержимое со знаком двух слов и выдает 8-разрядный двоичный результат в R. 4 младшие цифры результата заносятся в R и 4 старшие цифры в R+1.



Флаги

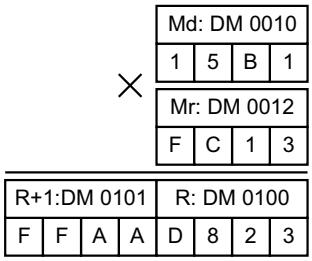
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

EQ: 1, когда результат равен 0000 0000, в противном случае = 1.

Пример

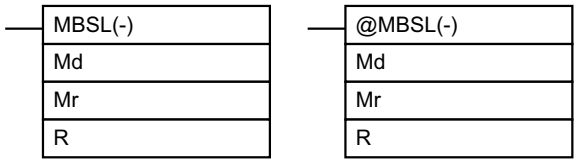
В следующем примере MBS(-) используется для умножения двоичного со знаком содержимого DM 0010 на двоичное со знаком содержимое DM 0012 и выдает результат в DM 0100 и DM 0101.

Адрес	Инструкция	Операнд
00000	LD	00100
00001	MBS(-)	
		DM 0010
		DM 0012
		DM 0100



5.21.8 MBSL(-) - Умножение двоичных чисел двойной длины

Обозначение на схеме



Область операндов

Md	первое слово множителя	IR, SR, AR, HR, TC, LR, DM, #
Mr	первое слово множимого	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

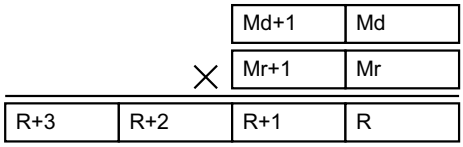
Данная команда есть только в CQM1-CPU4_-E.

MD И MD+1 должны находиться в одной области данных. То же относится и к MR и MR+1, R и R+3.

DM 6143..DM 6655 нельзя использовать для R.

Описание

MBSL(-) умножает 32-разрядное (8 цифр) двоичное число со знаком MD+1 и MD на 32-разрядное (8 цифр) двоичное число со знаком MR+1 и MR и выдает 16-разрядный двоичный результат со знаком в R+3..R.



Флаги

ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

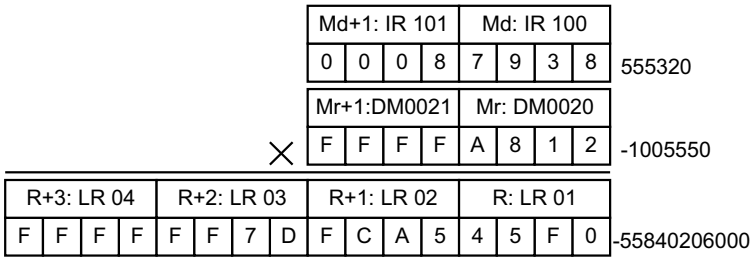
EQ: 1, когда результат равен 0 (содержимое всех слов R..R+3) в противном случае = 1.

Пример

В следующем примере MBSL(-) используется для умножения двоичного со знаком содержимого IR 101 и IR 100 на двоичное со знаком содержимое DM 0021 и DM 0020 и выдает результат в LR 04..LR 01.

Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	MBSL(-)	
		100
		DM 0020
		LR 21



5.21.9 DBS(-) - Деление двоичных чисел со знаком

Обозначение на схеме

DBS(-)
Dd
Dr
R

@DBS(-)
Dd
Dr
R

Область операндов

Dd	делимое	IR, SR, AR, HR, TC, LR, DM, #
Dr	делитель	IR, SR, AR, HR, TC, LR, DM, #
R	первое слово результата	IR, SR, AR, HR, LR, DM

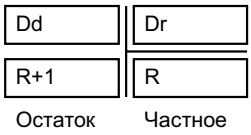
Ограничения

Данная команда есть только в CQM1-CPU4_-E.

DM 6143..DM 6655 нельзя использовать для R.

Описание

DBS(-) делит двоичное число со знаком (содержимое DD) на двоичное число со знаком (содержимое DR) и посылает 8-разрядный результат в R и R+1: частное в R и остаток в R+1.



Флаги

ER: Содержание DR = 0.

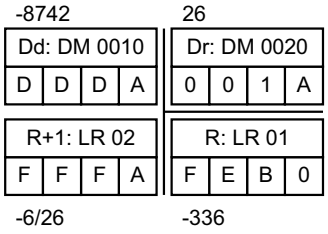
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

EQ: 1, когда содержание R (частное) равно нулю; в противном случае = 0.

Пример

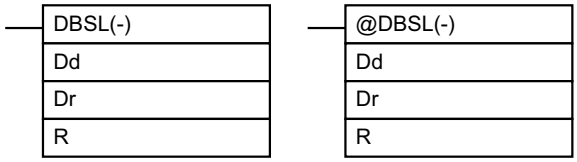
В следующем примере DBS(-) используется для деления двоичного со знаком содержимого DM 0010 на двоичное со знаком содержимое DM 0020 и выдает результат в LR 21 и LR 22.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	DBS(-)	
		DM 0010
		DM 0020
		LR 21



5.21.10 DBSL(-) - Деление двоичных чисел двойной длины со знаком

Обозначение на схеме



Область операндов

Dd	первое слово делимого	IR, SR, AR, HR, TC, LR, DM
Dr	первое слово делителя	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, LR, DM

Ограничения

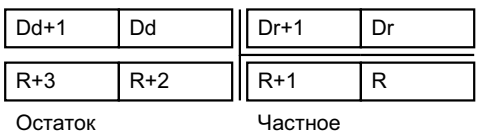
Данная команда есть только в CQM1-CPU4_-E.

DD И DD+1 должны находиться в одной области данных. То же относится и к DR и DR+1, R и R+3.

DM 6143..DM 6655 нельзя использовать для R.

Описание

DBSL(-) делит 32-битное (8-разрядное) двоичное число со знаком DD+1 и DD на 32-битное (8-разрядное) 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.



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.

Пример

В следующем примере DBSL(-) используется для деления двоичного со знаком содержимого IR 101 и IR 100 на двоичное со знаком содержимое DM 0021 и DM 0020 и выдает результат в LR 04..LR 01.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	DBS(-)	
		DM 0010
		DM 0020
		LR 21

Dd+1: IR 101				Dd: IR 100			
F	F	7	A	B	1	5	C

Dr+1: DM 0021				Dr: DM 0020			
0	0	0	0	0	0	1	A

÷

R+3: LR 04				R+2: LR 03				R+1: LR 02				R: LR 01			
F	F	F	F	F	F	F	C	F	F	F	A	D	F	7	0

5.22 Специальные математические команды

5.22.1 MAX(-) - Найти максимум

Обозначение на схеме

MAX(-)
C
R_1
D

@MAX(-)
C
R_1
D

Область операндов

C	слово состояния	IR, SR, AR, HR, TC, LR, DM, #
R_1	первое слово зоны	IR, SR, AR, HR, TC, LR, DM
D	слово приемник	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

N должна быть двоично-десятичным числом от 0000 до 999.

R1 и R1+N-1 должны лежать в одной области памяти.

DM 6144..DM 6655 нельзя использовать для D.

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 слова, содержащего максимальное значение, в D+1. Адрес идентифицируется по-другому для области DM.

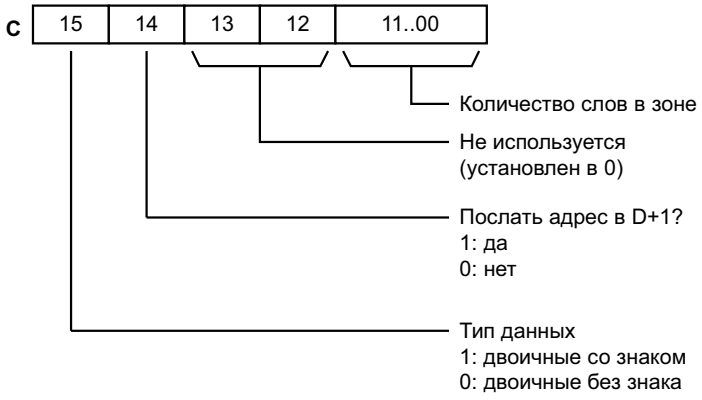
1, 2, 3,... 1. Для адреса в области DM адрес записывается в D+1. Например, если адрес , содержащий максимум, = DM 0114, тогда #0114 запишется в D+1.

2. Для адреса в другой области 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 одинаковое максимальное значение содержится по нескольким адресам, в D+1 будет помещен самый младший адрес. В случае области DM будет записан абсолютный адрес, а при других областях - адрес относительно начала поиска.

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).

R1 и R1+N-1 находятся в разных областях.

EQ: 1, когда максимальное значение равно нулю.

5.22.2 MIN(-) - Найти минимум

Обозначение на схеме

MIN(-)
C
R_1
D

@MIN(-)
C
R_1
D

Область операндов

C	слово состояния	IR, SR, AR, HR, TC, LR, DM, #
R_1	первое слово зоны	IR, SR, AR, HR, TC, LR, DM, #
D	слово приемник	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

N должна быть двоично-десятичным числом от 0000 до 999.

R1 и R1+N-1 должны лежать в одной области памяти.

DM 6144..DM 6655 нельзя использовать для D.

Описание

Когда условие исполнение = 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):

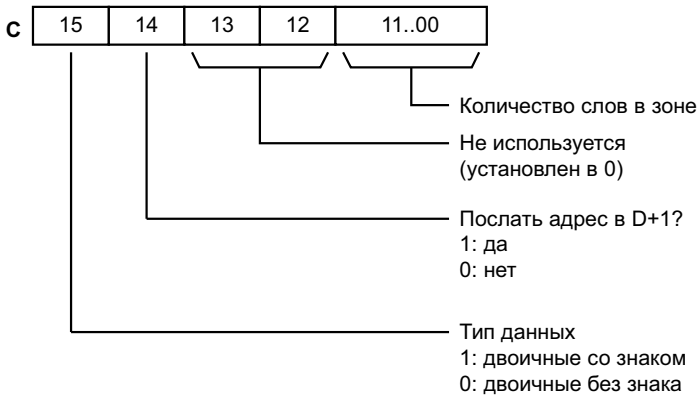
1, 2, 3,... 1. Для области DM в D+1 записывается абсолютный адрес. Например, если адрес , содержащий максимум, = DM 0114, тогда в D+1 запишется #0114.

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. В случае области DM будет записан абсолютный адрес, а при других областях - адрес относительно начала поиска.

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).

R1 и R1+N-1 находятся в разных областях.

EQ: 1, когда результат равен нулю.

5.22.3 AVG(-) - Среднее значение

Обозначение на схеме

AVG(-)
S
N
D

@AVG(-)
S
N
D

Область операндов

S	слово источник	IR, SR, AR, HR, TC, LR, DM
N	число циклов	IR, SR, AR, HR, TC, LR, DM, #
D	слово приемник	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

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.

DM 6144..DM 6655 нельзя использовать для S,N или D..D1+N+1

Описание

AVG(-) используется для вычисления среднего значения S за N циклов.

Когда условие исполнения = 0, AVG(-) не выполняется.

Каждый раз при исполнении AVG(-) содержимое S заносится в слова D+2..D+N+1. При первом исполнении команда записывает S в D+2, при втором содержимое S записывается в D+3 и т. д. При N-м исполнении AVG(-) записывает содержимое S в D+N+1, вычисляет среднее значение ячеек D+2..D+N+1 и записывает его в D. В таблице показаны функции слов D..D +N+1.

D	Среднее значение (после N или более исполнений)
D+1	Используется системой
D+2	Содержимое S от 1-го исполнения AVG(-)
D+3	Содержимое S от 2-го исполнения AVG(-)

...	
D+N+1	Содержимое S от N-го исполнения AVG(-)

Предосторожности

Среднее значение подсчитывается в двоичном виде. Следите за тем, чтобы S было в двоичном виде.

N должно быть двоично-десятичным числом от #0001 до #0064. Если N #0065, AVG(-) будет работать с N= 64.

Среднее значение будет округляться до ближайшего целого. (0.5 округляется до 1).

Пускай в D+1занесется #0000.после первого исполнения AVG(-)

Флаги

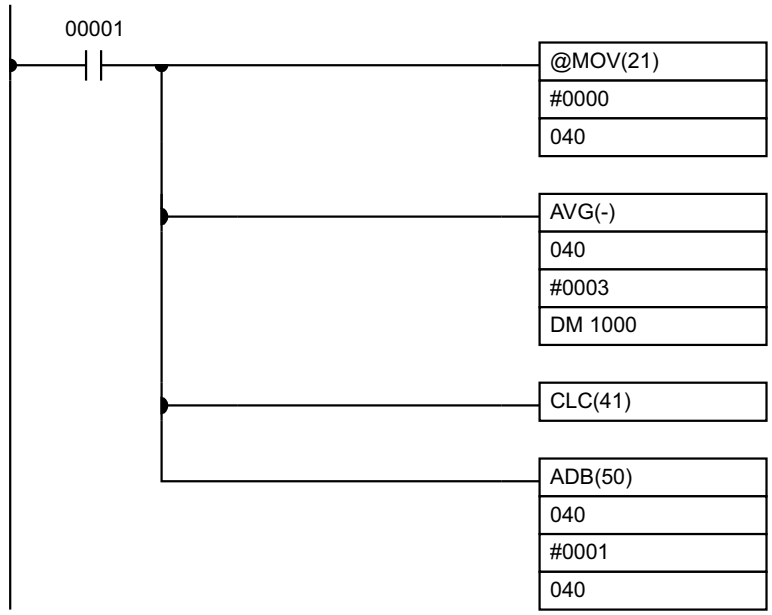
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

Один или более операндов задан некорректно.

D и D+N+1 находятся в разных областях.

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. 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	00001
00001	@MOV(21)	
		# 0000
		040
00002	AVG(-)	
		040
		# 0003
		DM 1000
00003	CLC(41)	

Адрес	Инструкция	Операнд
00004	ADB(50)	
		040
		# 0001
		040

	1 цикл	2 цикл	3 цикл	4 цикл
IR 040	0000	0001	0002	0003

DM 1000	0000	0001	0001	0002	Среднее значение
DM 1001					Используется системой
DM 1002	0000	0000	0000	0003	Предыдущее значение IR 040
DM 1003	..	0001	0001	0001	
DM 1004	..	..	0002	0002	

5.22.4 SUM(-) - Сумма

Обозначение на схеме

SUM(-)
S
R_1
D

@SUM(-)
S
R_1
D

Область операндов

S	слово состояния	IR, SR, AR, HR, TC, LR, DM, #
R_1	первое слово зоны	IR, SR, AR, HR, TC, LR, DM, #
D	первое слово приемника	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

3 правых (младших) цифры C должны быть двоично-десятичными от 001 до 999.

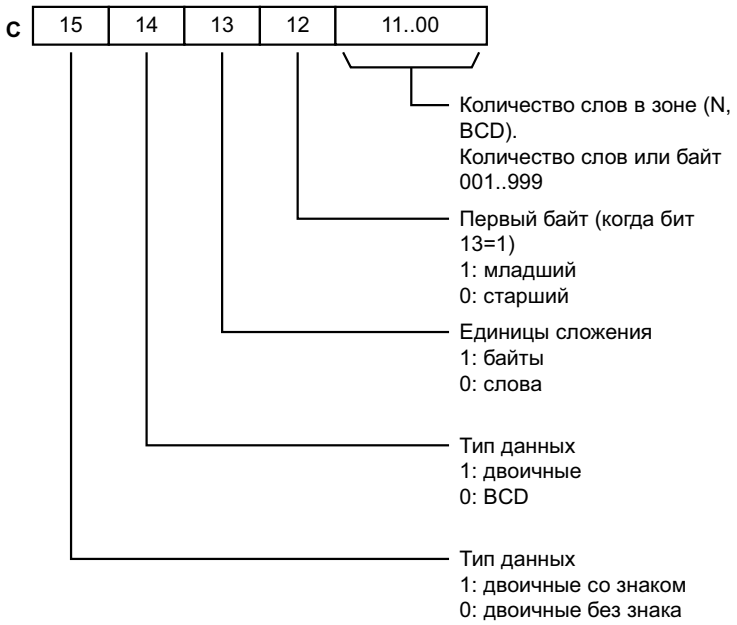
DM 6143..DM 6655 нельзя использовать для D.

Если бит 14 слова C = 0 (задание для 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 от R1 до 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.

Функция битов в C показано на следующем рисунке 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. Количество будет указывать на количество либо слов, либо байт в зависимости от объектов суммирования.

Единицы сложения

Если бит 13 = 0, будут складываться слова, если бит 13 = 1, то байты.

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

Данные в зоне считаются двоичными без знака, если в слове C бит 14 = 1, а бит 15 = 0; данные в зоне считаются двоичными со знаком, если биты 14 и 15 = 1.

Данные в зоне считаются двоично-десятичными, если бит 14 слова C = 0, 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.

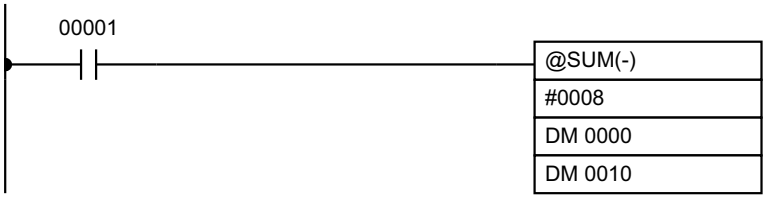
Количество объектов в C не является двоично-десятичным числом от 001 до 999.

Данные, подлежащие суммированию, не являются двоично-десятичными числами, когда задано сложение двоично-десятичных чисел.

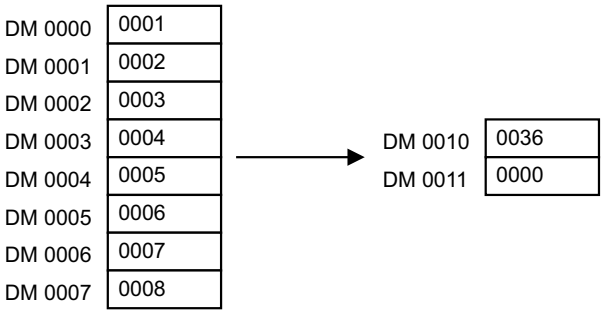
EQ: 1, когда результат равен нулю.

Пример

В следующем примере двоично-десятичное содержимое 8 слов от DM 0000 до DM 0007 складываются, когда IR 00001 = 1, и результат заносится в DM 0010 и DM 0011.

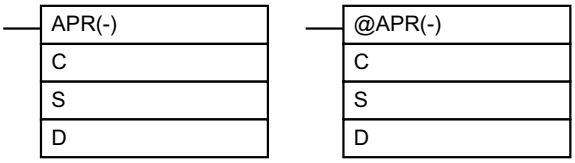


Адрес	Инструкция	Операнд
00000	LD	00000
00001	@SUM(-)	
		# 0008
		DM 0000
		DM 0010



5.22.5 APR(-) - Математические вычисления

Обозначение на схеме



Область операндов

C	слово состояния	IR, SR, AR, HR, TC, LR, DM, #
S	слово источника входных данных	IR, SR, AR, HR, TC, LR, DM
D	слово приемник результата	IR, SR, AR, HR, TC, LR, DM

Ограничения

Данная команда есть только в CQM1.

Для тригонометрических функций S должно быть двоично-десятичным числом от 0000 до 0900 ($0^0 \leq \Theta \leq 90^0$).

DM 6144..DM 6655 нельзя использовать для D.

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 в одной десятой градуса.

Если C является адресом, APR(-) вычисляет f(x) функцию, введенную ранее, имеющую начало в слове C. Функция - это серия отрезков линии (которыми аппроксимируется кривая), заданной 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 C задана константа, отличная от #0000 и #0001.

Данные линейной аппроксимации ненадежны.

EQ: Результат равен нулю.

Примеры

Функция синуса

В следующем примере показано использование функции синуса APR(-) для вычисления синуса 30^0. Функция синуса задана, когда C = #0000.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	APR(-)	
		# 0000
		DM 0000
		DM 0010

Входные данные, x

S: DM 0000			
0	10^1	10^0	10^{-1}
0	3	0	0

Введите данные, не превышающие #0900 в BCD

Результат

D: DM 0100			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
5	0	0	0

Результат имеет 4 значащие цифры, пятая и далее игнорируются. Результат для Sin(90) будет 0.9999 а не 1

Функция косинуса

В следующем примере показано использование функции косинуса APR(-) для вычисления косинуса 30^0. Функция косинуса задана, когда C = #0001.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	APR(-)	
		# 0001
		DM 0010
		DM 0110

Входные данные, x

S: DM 0000			
0	10^{1}	10^{0}	10^{-1}
0	3	0	0

Введите данные, не превышающие #0900 в BCD

Результат

D: DM 0100			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
8	6	6	0

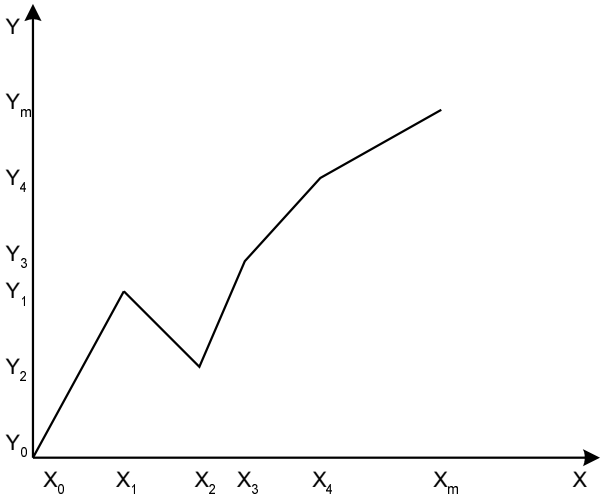
Результат имеет 4 значащие цифры, пятая и далее игнорируются. Результат для Cos(0) будет 0.9999 а не 1

Линейная аппроксимация

Линейная аппроксимация APR(-) задана, когда в C установлен адрес памяти. Слово C является первым словом блока памяти, содержащего параметры линейной аппроксимации.

Содержимое слова C задает количество отрезков линий в аппроксимации, и вид ввода и вывода - двоично-десятичный или двоичный. Биты 00..07 содержат количество сегментов минус 1, в виде двоичных данных. Биты 14 и 15 определяют, соответственно, формы ввода и вывода: 0 задает двоично-десятичную форму, 1 - двоичную.

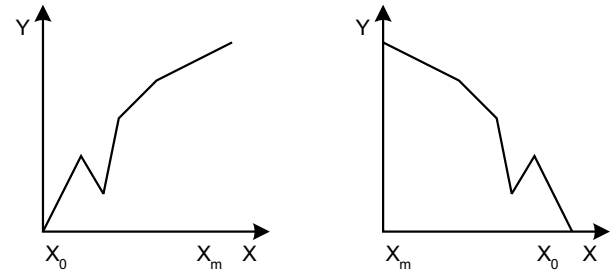
Введите координаты m+1 конечных точек, которые определяют m отрезков линии, как показано в следующей таблице. Все координаты вводите в двоичной форме. Всегда вводите координаты в направлении с наименьшего значения X (X1) до наибольшего (Xm). X_0 = 0000, и его вводить не нужно.



Слово	**Координата**
C+1	X_m
C+2	Y_0
C+3	X_1

Слово	Координата
C+4	Y_1
C+5	X_2
C+6	Y_2
...	
C+(2m+1)	X_m
C+(2m+2)	Y_m

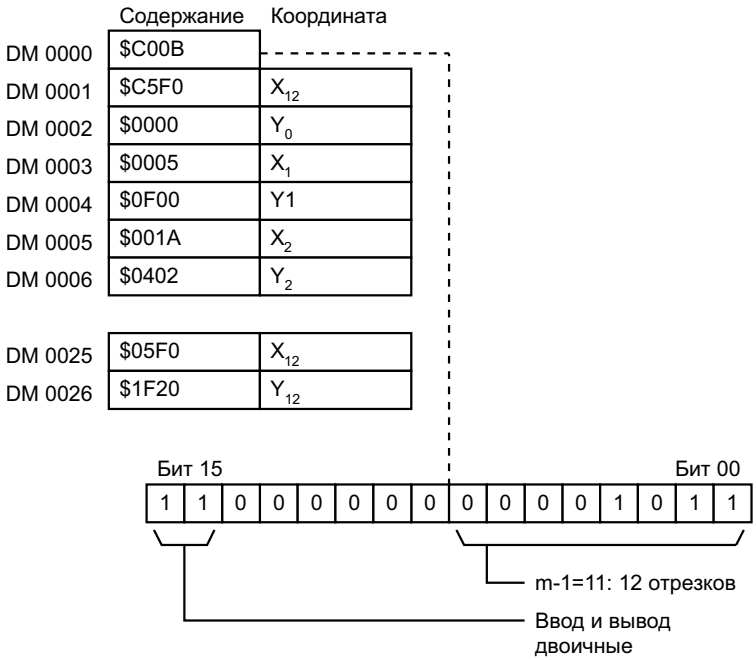
Если бит 13 слова C = 1, график будет зеркально перевернут слева направо, как показано на следующем рисунке.



Следующий пример демонстрирует конструкцию линейной аппроксимации с 12 отрезками прямых. Блок данных, как и положено, непрерывен от DM 0000..DM 0026 (C..C+(2x12+2)). Входные данные берутся из IR 010, и результат выдается в IR 011.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	APR(-)	
		DM 0000
		010
		011

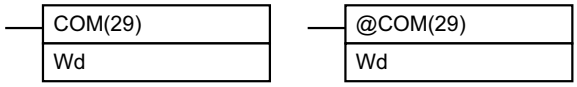
В данном случае слово входных данных, IR 010, содержит #0014 и f(0014) = #0726 выдается в R, IR 011.


Содержание
Координата
DM 0000
$C00B
DM 0001
$C5F0
X12
DM 0002
$0000
Y0
DM 0003
$0005
X1
DM 0004
$0F00
Y1
DM 0005
$001A
X2
DM 0006
$0402
Y2
DM 0025
$05F0
X12
DM 0026
$1F20
Y12
Бит 15
Бит 00
1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1
m-1=11: 12 отрезков
Ввод и вывод
двоичные

5.23 Логические команды

5.23.1 COM(29) - Дополнение

Обозначение на схеме



Область операндов

Wd	слово дополнения	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для WD.

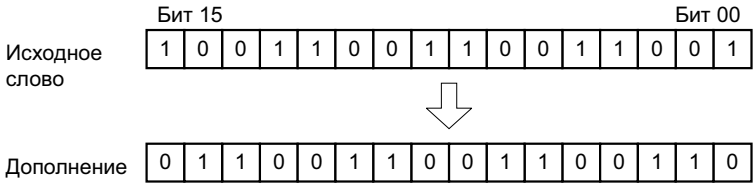
Описание

Когда условие исполнение = 0, COM(29) не выполняется. Когда условие исполнения = 1, COM(29) очищает в слове WD биты, находящиеся в состоянии 1 и устанавливает в 1 все биты, находящиеся в состоянии 0.

Предосторожности

Дополнение Wd будет вычисляться каждый цикл, если используется версия команды не фронта 0/1. Для того, чтобы это действие выполнилось только один раз, используйте версию фронта 0/1 команды (@COM(29)) или комбинируйте команду COM(29) с командами DIFU(13) или DIFD(14) .

Пример



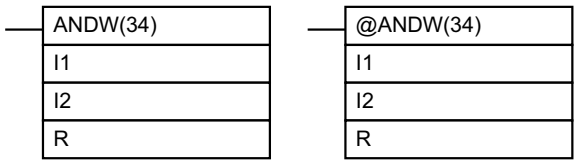
Флаги

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.

5.23.2 ANDW(34) - Логическое И

Обозначение на схеме



Область операндов

I1	вход 1	IR, SR, AR, HR, TC, LR, DM, #
I2	вход 2	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

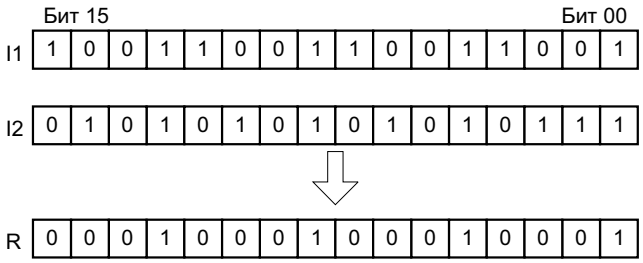
Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, AND(34) не выполняется. Когда условие исполнения = 1, AND(34) 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.

5.23.3 ORW(35) - Логическое ИЛИ

Обозначение на схеме

ORW(35)
I1
I2
R

@ORW(35)
I1
I2
R

Область операндов

I1	вход 1	IR, SR, AR, HR, TC, LR, DM, #
I2	вход 2	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

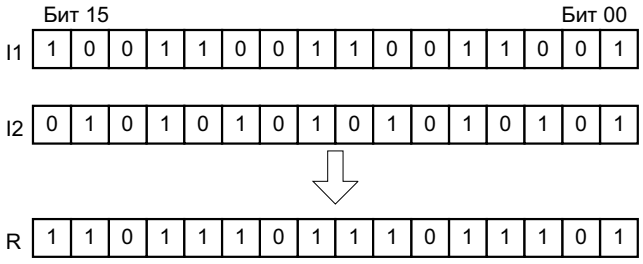
Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнение = 0, ORW(35) не выполняется. Когда условие исполнения = 1, ORW(35) 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.

5.23.4 XORW(36) - Исключающее ИЛИ

Обозначение на схеме

XORW(36)
I1
I2
R

@XORW(36)
I1
I2
R

Область операндов

I1	вход 1	IR, SR, AR, HR, TC, LR, DM, #
I2	вход 2	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

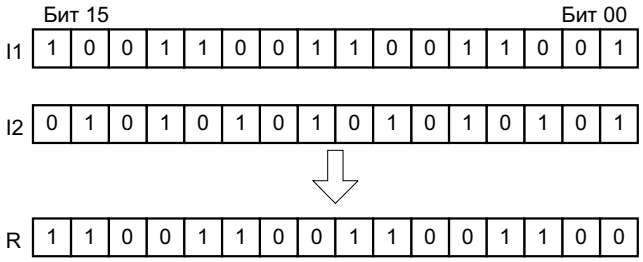
Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, XOR(36) не выполняется. Когда условие исполнения = 1, XOR(36) 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.

5.23.5 XNRW(37) - Исключающее ИЛИ - НЕ

Обозначение на схеме

XNRW(37)
I1
I2
R

@XNRW(37)
I1
I2
R

Область операндов

I1	вход 1	IR, SR, AR, HR, TC, LR, DM, #
I2	вход 2	IR, SR, AR, HR, TC, LR, DM, #
R	слово результат	IR, SR, AR, HR, LR, DM

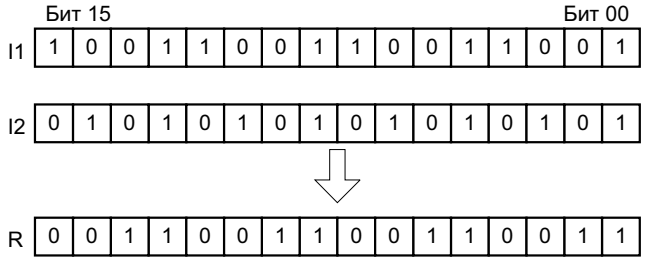
Ограничения

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, XNRW(37) не выполняется. Когда условие исполнения = 1, XNRW(37) производит операцию ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ с содержимым I1 и I2 побитно и выдает результат в R.

Пример



Флаги

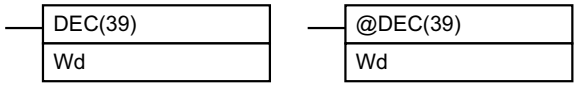
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

EQ: 1, когда результат равен нулю.

5.24 Команды инкрементирования/ декрементирования

5.24.1 INC(38) - Инкремент двоично-десятичного числа

Обозначение на схеме



Область операндов

Wd	слово декрементирования (BCD)	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для WD.

Описание

Когда условие исполнения = 0, INC(38) не выполняется. Когда условие исполнения = 1, INC(38) инкрементирует (увеличивает на 1) WD, без воздействия на перенос (CY).

Предупреждения

Если используется версия команды не фронта 0/1, содержимое слова WD будет инкрементироваться каждый цикл. Для того, чтобы это действие выполнилось только один раз, используйте версию фронта 0/1 команды (@INC(38)) или комбинируйте команду INC(38) не фронта 0/1 с командами DIFU(13) или DIFD(14) .

Флаги

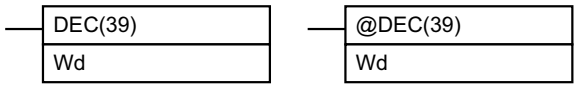
ER: WD не в двоично-десятичном виде.

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.

5.24.2 DEC(39) - Декремент двоично-десятичного числа

Обозначение на схеме



Область операндов

Wd	слово декрементирования (BCD)	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для WD.

Описание

Когда условие исполнения = 0, DEC(39) не выполняется. Когда условие исполнения = 1, DEC(39) декрементирует (уменьшает на 1) WD, без воздействия на перенос (CY). DEC(39) работает аналогично INC(38), за исключением того, что декрементирует значение вместо того, чтобы инкрементировать.

Предосторожности

Если используется версия команды не фронта 0/1, содержимое слова WD будет декрементироваться каждый цикл. Для того, чтобы это действие выполнилось только один раз, используйте версию фронта 0/1 команды (@DEC(39)) или комбинируйте команду DEC(39) не фронта 0/1 с командами DIFU(13) или DIFD(14) .

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 декрементированный результат равен нулю.

5.25 Команды подпрограмм

Подпрограммы разбивают большие задачи управления на небольшие и позволяют повторно использовать набор блоков 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 был вход в подпрограмму. (Если в подпрограмме не назначено что-либо другое).

5.25.1 SBS(91) - Войти в подпрограмму

Обозначение на схеме

SBS(91) N

Область операндов

N	номер подпрограммы	000..255

Ограничения

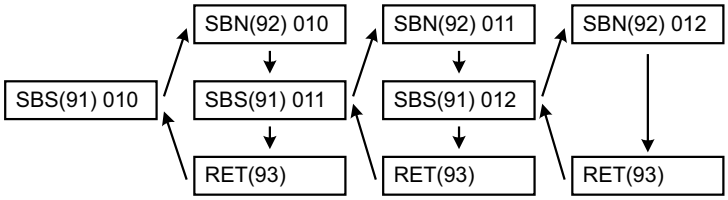
CQM1-CPU11/21 поддерживает только номера подпрограмм 000..127. CPM1 поддерживает только номера подпрограмм 000..049.

Описание

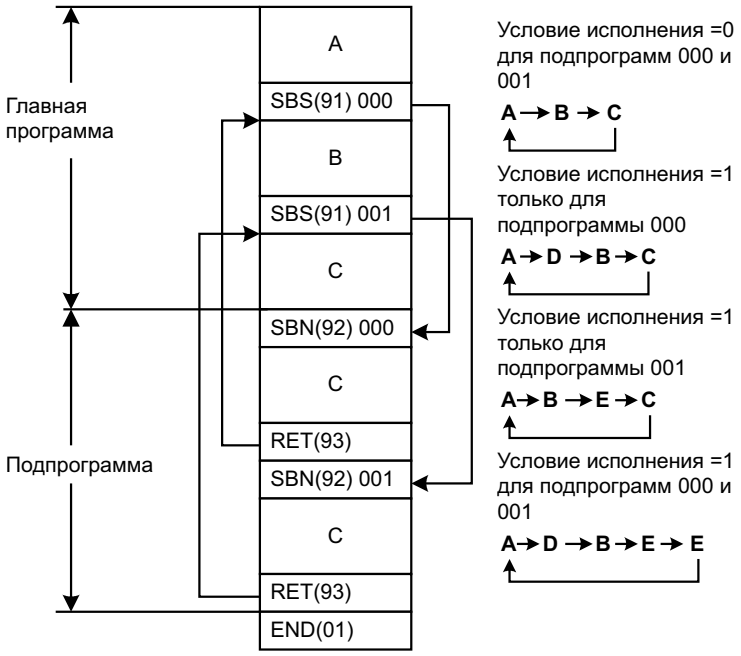
Подпрограмма выполняется путем помещения команды SBS(91) в главной программе в точке, в которой необходим вызов подпрограммы. Номер, который задается вместе с SBS(91), указывает на номер требуемой подпрограммы. Когда выполняется команда SBS(91) (т.е. условие исполнения = 1), то выполняются команды между SBN(92) с таким же самым номером и первой командой RET(93) после нее, прежде чем исполнение программы возвратится к команде, следующей за SBS(91), сделавшей вызов.

SBS(91) можно использовать в программе столько раз, сколько требуется, т.е. одну и ту же подпрограмму можно вызвать из разных мест программы.

SBS(91) можно также поместить в подпрограмму для вызова одной подпрограммы из другой, т.е. подпрограммы могут быть вложенными. После завершения второй подпрограммы (т.е. выполнилась команда RET(93)), выполнение программы возвращается к исходной подпрограмме, которая выполняется и возвращается в главную программу. Возможны 16 уровней вложений. Подпрограмма не может вызывать саму себя. (например, нельзя запрограммировать SBS(91) 000 внутри подпрограммы, определенной командой SBN(92) 000). Следующий рисунок иллюстрирует два уровня вложения.



Следующий рисунок иллюстрирует процесс исполнения программы для различных условий исполнения для двух SBS(91).



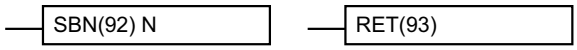
Флаги

ER: Не существует подпрограммы с заданным номером.
Подпрограмма вызывает саму себя
Вызвана активная подпрограмма.

Внимание! SBS(91) не будет выполняться и подпрограмму нельзя вызвать, если флаг ER = 1.

5.25.2 SBN(92) - Начало подпрограммы и RET(93) - возврат

Обозначение на схеме



Область операндов

N	номер подпрограммы	000..255

Ограничения

CQM1-CPU11/21 поддерживает только номера подпрограмм 000..127.

CPM1 поддерживает только номера подпрограмм 000..049.

Каждый номер подпрограммы можно использовать в SBN(92) только раз.

Описание

SBN(92) служит для обозначения начала подпрограммы. RET(93) служит для обозначения конца подпрограммы. Каждая подпрограмма идентифицируется номером подпрограммы N, который программируется как определитель SBN(92). При вызове подпрограммы используется такой же номер (см. 5-25-1). RET(93) 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(92) и затем возвратится на адрес 00000 для следующего цикла. Если подпрограммы не вызываются командой SBS(91), они не выполняются.

END(01) должен ставиться в конце последней подпрограммы, т.е. после последнего RET(93). В других местах программы данная команда не требуется.

Предосторожности

Если SBN(92) ошибочно помещена в главной программе, она запретит исполнение после своего адреса, т.е. выполнение программы возвратится к началу, если встретилась команда SBN(92).

Если в подпрограмме стоят DIFU(13) или DIFD(14), 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.

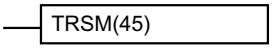
5.26 Специальные команды

5.26.1 TRSM(45) - Трассировка выборки памяти

Внимание! Данная команда не поддерживается CQM1-CPU11-EV1/21-E и CPM1.

Трассировка данных служит для облегчения отладки программ. Для установки и использования трассировки данных необходимо иметь управляющий компьютер с SSS. С программатора трассировка данных невозможна. Подробно трассировка данных описана в Инструкции по работе с SSS, ПК серии C. В данной главе описан символ релейно-контактной схемы TRSM(45) и дан пример программы.

Обозначение на схеме



Описание

TRSM(45) используется 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 SSS, ПК серии C).

TRSM(45) управляется не условиями исполнения, а двумя битами в области AR: AR 2515 и AR 2514. AR 2515 - бит пуска выборки. Этот бит устанавливается в 1 для начала процесса выборки для трассировки. Бит старта выборки не должен устанавливаться в 1 из программы, а только с периферийного устройства.

AR 2514 - бит пуска трассировки. Когда он установлен, назначенные данные регистрируются в памяти трассировки. Бит пуска трассировки можно устанавливать в 1 либо из программы, либо с периферийного устройства. Можно вводить положительное или отрицательное задержке для изменения фактической точки, откуда будет начинаться трассировка.

Данные можно записывать тремя путями. TRSM(45) можно поместить в одном или нескольких местах программы для указания, где нужно трассировать назначенные данные. Если TRSM(45) не используется, назначенные данные будут трассироваться после исполнения END(01). Третий метод включает задание таймерного интервала с периферийных устройств, так что назначенные данные будут трассироваться через регулярные интервалы времени независимо от времени цикла (см. Инструкцию по работе с SSS, ПК серии C).

TRSM(45) можно встроить в любое место программы, любое число раз. Данные в памяти трассировки можно просматривать периферийным устройствам, управляющим компьютером и т. д.

Биты управления и флаги AR

Следующие биты и флаги управления используются во время трассировки. Флаг трассировки должен быть в состоянии 1 во время операции трассировки. Флаг завершения трассировки включится в 1, когда будет достаточно данных для заполнения памяти трассировки.

Флаг	Функция
AR 2515	Бит начала выборки
AR 2514	Бит пуска трассировки
AR 2513	Флаг трассировка
AR 2512	Флаг завершения трассировки

Замечание Не изменяйте состояние AR 2515 из программы.

Предосторожности

TRSM(45) не будет исполняться внутри блока перехода JMP(08) - JME(09), если условия перехода = 0.

Пример

Следующий пример показывает базовую программу и операцию для трасссировки данных. Принудительно включите бит начала выборки (AR 2515) для начала выборки.

Бит пуска выборки не должен быть установлен в 1 из программы. Данные читаются и загружаются в память трассировки.

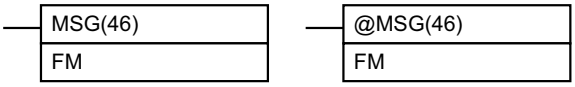
Когда IR 00000 = 1, Бит пуска трассировки (AR 2514) также устанавливается в 1, и ЦУ ищет задержке и соответственно маркирует память трассировки. Это может значить, что некоторые уже готовые образцы будут записаны в память трассировки (отрицательная задержка), или будет сделано еще несколько образцов перед тем, как они будут записаны (положительная задержка).

Выбранные 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 задержка положительна). Отрицательная задержка не может быть такой, что требуемые данные выполнились перед пуском выборки.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	AR 2514
00002	TRSM(45)	
00003	LD	AR 2513
00004	OUT	00200
00005	LD	AR 2512
00006	OUT	00201

5.26.2 MSG(46) - Индикация сообщения

Обозначение на схеме



Область операндов

FM	первое слово сообщения	IR, SR, AR, HR, LR, DM

Ограничения

DM 6649..DM 6655 нельзя использовать для FM.

Описание

Когда условие исполнения = 1, MSG(46) читает 8 слов расширенного кода ASCII из FM..FM+7 и индикатирует сообщение на программаторе. Индикатируемое сообщение может быть длиной до 16 символов, т.е. каждый символ кода ASCII требует 8 бит (2 цифры). Коды ASCII см. Приложение H. В данный код включены также японские знаки катаканы.

Если не все 8 слов требуются для сообщения, его можно прервать в любом месте, вводя “OD”. Когда в сообщении встречается OD, больше слова читаться не будут и ячейки памяти, которые нормально били бы использованы для сообщений, можно использовать для других целей.

Буферизация сообщений и приоритет.

В памяти можно забуферизовать до трех сообщений. После загрузки в буфер они индикатируются на принципе FIFO “первый вошел, первый вышел”. Поскольку возможно, что в одном цикле выполнилось более 3 команд MSG(46), основанная на область, в которой загружены сообщения, для выбора тех, которые требуется загрузить в буфер.

Приоритет областей данных для вывода сообщений имеет следующий вид:

LR>IR>HR>AR>TC>DM

При работе сообщениями из одной области данных высший приоритет имеют с меньшими адресами.

При работе с косвенно заданными сообщениями (т.е. *DM) высший приоритет имеют с меньшими адресами.

Очистка сообщений

Для очистки сообщений выполните FAL(06) 00 или очистите с программатора или SSS.

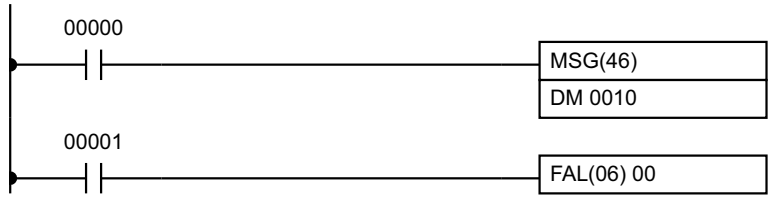
Если сообщения изменяются во время индикациии, индикация также изменится.

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.



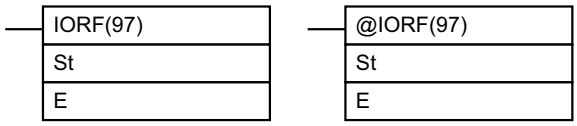
Адрес	Инструкция	Операнд
00000	LD	00000
00001	MSG(46)	
		DM 0010
00002	LD	00001
00003	FAL(06)	00

MSG
ABSDEFGHIJKLMNOP

Содержимое 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	4	F	O	P

5.26.3 IORF(97) - Обновление входов/выходов

Обозначение на схеме



Область операндов

St	начальное слово	IR 000 .. IR 111
E	конечное слово	IR 000 .. IR 111

Ограничения

ST должно быть менее либо равно E.

Описание

Для обновления слов входов/выходов задайте первое (ST) и последнее (E) слово входов/выходов, подлежащих обновлению. Когда условие исполнения для IOFR(97) = 1, все слова от ST до E будут обновлены. Это будет делаться в дополнение к нормальному обновлению входов/выходов, выполняемому во время цикла ПК.

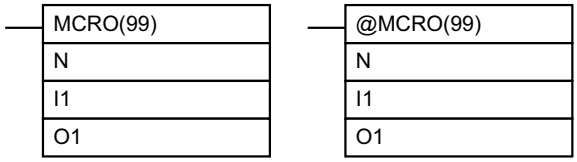
Замечание Данная команда не действует на слова, не использующиеся для входов/выходов.

Флаги

Данная команда на флаги воздействия не оказывает.

5.26.4 MCRO(99) - Макро

Обозначение на схеме



Область операндов

N	номер подпрограммы	000 .. 127
I1	первое слово входа	IR, SR, AR, HR, TC, LR, DM
O1	первое слово выхода	IR, SR, AR, HR, LR, DM

Ограничения

DM 6144..DM 6655 нельзя использовать для O1.

CPM1 поддерживает только подпрограммы 000..049

Описание

Команда MCRO(99) позволяет написать одну подпрограмму вместо нескольких подпрограмм, которые имеют одинаковую структуру, но разные операнды. Есть 4 слова входа IR 096..IR 099 (IR 232..IR 235 для CPM1) и 4 слова выхода IR 196..IR 199 (IR 236..IR 239 для CPM1), которые отведены для MCRO(99). Данные 8 слов используются в подпрограмме и берут свое содержимое из I1..I1+3 и O1..O1+3 при исполнении подпрограммы.

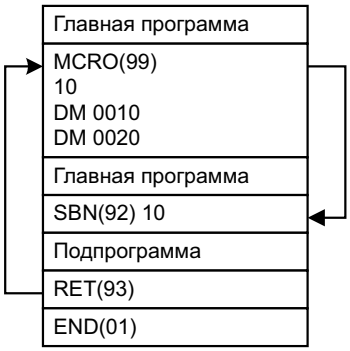
Когда условие исполнения = 0, MCRO(99) не выполняется. Когда условие исполнения = 1, MCRO(99) копирует содержание I1..I1+3 в IR 096..IR 099, копирует содержание O1..O1+3 в IR 196..IR 199 и затем вызывает и исполняет подпрограмму, заданную в N. По завершении подпрограммы содержимое IR 096..IR 099 передаются обратно в O1..O1+3 перед завершением MCRO(99).

Другие подробности о MCRO(99) см. 2-3.

Пример

В данном примере содержимое DM 0010..DM 0013 копируется в IR 096 ..... IR 099, содержимое DM 0020..DM 0023 копируется в IR 196 ..... IR 199 и вызывается на

исполнение подпрограмма 10. После завершения подпрограммы содержимое IR 196..IR 199 передаются обратно в DM 0020..DM 0023.



Флаги

ER: Подпрограмма с заданным номером не существует.

Операнд превысил границу области данных.

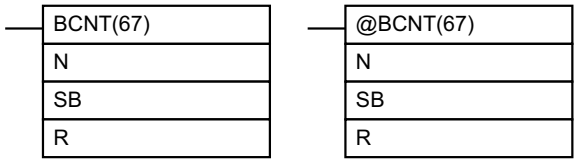
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

Подпрограмма вызывает саму себя.

Вызвана работающая подпрограмма.

5.26.5 BCNT(67) - Счетчик битов

Обозначение на схеме



Область операндов

N	количество слов (BCD)	IR, SR, AR, HR, TC, LR, DM, #
SB	первое слово источника	IR, SR, AR, HR, TC, LR, DM
R	слово приемник	IR, SR, AR, HR, TC, LR, DM

Ограничения

N не может быть 0.

DM 6144..DM 6655 нельзя использовать для R.

Описание

Когда условие исполнения = 0, BCNT(67) не выполняется. Когда условие исполнения = 1, BCNT(67) считает общее количество битов в состоянии 1 во всех словах между SB и SB+(N-1) и помещает результат в R.

Флаги

ER: N не двоично-десятичное число, или N =0; SB и SB+(N-1) находятся в разных области.

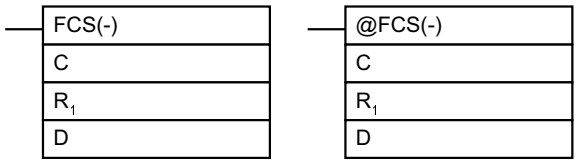
Результат превышает 9999.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

EQ: 1 при результате = 0.

5.26.6 FCS(-) - Контрольная сумма кадра

Обозначение на схеме



Область операндов

C	слово состояния	IR, SR, AR, HR, LR, DM, #
R_1	первое слово в зоне	IR, SR, AR, HR, TC, LR, DM
D	первое слово приемника	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

3 младших цифры C должны быть двоично-десятичными от 000 до 999.

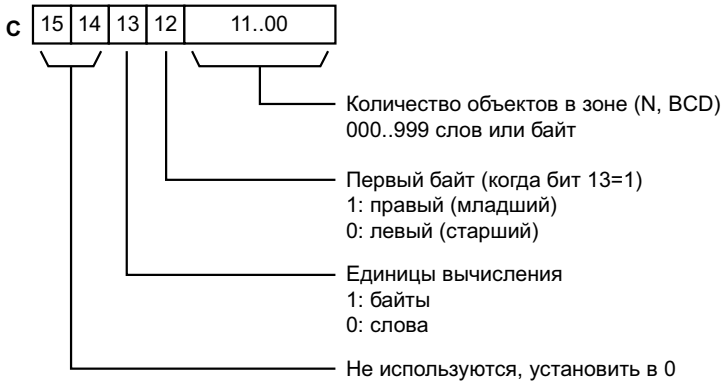
DM 6143..DM 6655 нельзя использовать для D.

Описание

FCS(-) можно использовать 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 кадра (16-ричное) преобразуется в ASCII и выдается в слова приемника (D и D+1).

Функции битов в слове C показано на следующем рисунке и подробно 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 вычислений

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.

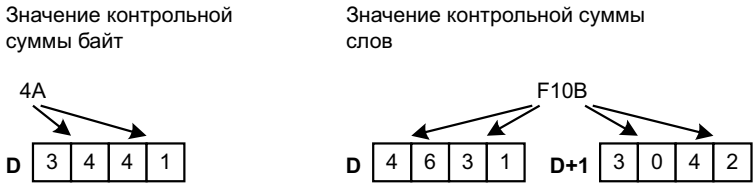
	Старший байт	**Младший байт**
R1	**1**	**2**
R1+1	3	4
R1+2	5	6
R1+3	7	8
...		

Когда бит 12 = 0, над байтами будет производиться операция ИЛИ в следующем порядке: 1, 2, 3, 4...

Когда бит 12 = 1, над байтами будет производиться операция ИЛИ в следующем порядке: 2, 3, 4,5...

Преобразование в 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.



Флаги

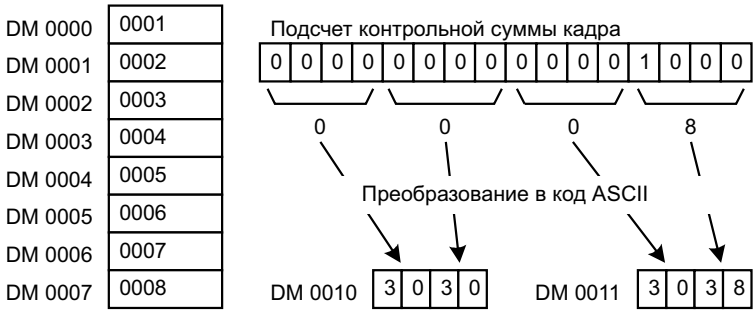
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

Количество объектов в C не является двоично-десятичным числом от 001 до 999.

Пример

Когда в следующем примере IR 0000 = 1, контрольная сумма (0008) вычисляется для 8 слов DM 0000..DM 0007 и эквивалент ASCII (30, 30, 30, 38) заносится в DM 0010 и DM 0011.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	@FCS(-)	
		# 0008
		DM 0000
		DM 0010



5.26.7 FPD(-) - Поиск точки сбоя

Обозначение на схеме

FPD(-)
C
T
D

Область операндов

C	слово состояния	#
T	время наблюдения (BCD)	IR, SR, AR, HR, TC, LR, DM, #
D	первое слово регистра	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

D и D+8 должны лежать в одной области данных когда бит 15 слова C = 1.

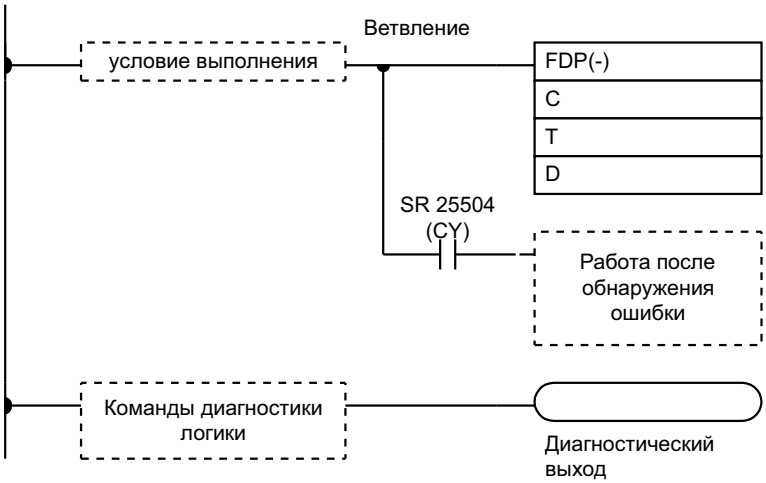
DM 6144..DM 6655 нельзя использовать для T или D.

C должна быть введено 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(06) с номером FAL, заданном в C.

Секции программы, отмеченные пунктиром на следующей релейно-контактной схеме, можно записать по требованию конкретного применения программы. Секция программы, переключаемая CY, является дополнительной и может использовать любую команду кроме LD и LD NOT. Команды диагностики логики и условие исполнения могут состоять из любой комбинации желаемых условий (нормально открытых и нормально закрытых).



Когда условие исполнение = 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(06) с номером FAL, заданном в двух цифрах C. Если задан 00, признак ошибки не генерируется.

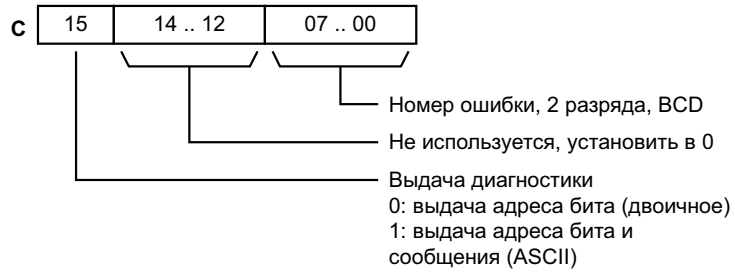
2. Команды диагностики логики опрашиваются на первое входное условие = 0, и адрес бита данного условия выдается в слова приемника, начинающихся с D.

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, предварительно заданное сообщение до 8 символов ASCII будет индикатироваться на программаторе вместе с адресом бита, упомянутого в шаге 2.

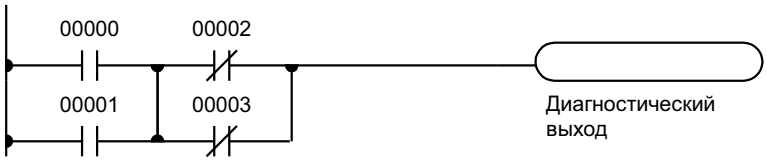
Параметр управления

Функции битов слова управления C показаны на следующем рисунке.



Команды диагностики логики

Если время до включения в 1 условия диагностики логики превысило T, команды диагностики логики опрашиваются на входные условия в состоянии 0. Если в состоянии 0 находятся более одного условия исполнения, выбирается условие на более высокой строке и ближе к левой шине.



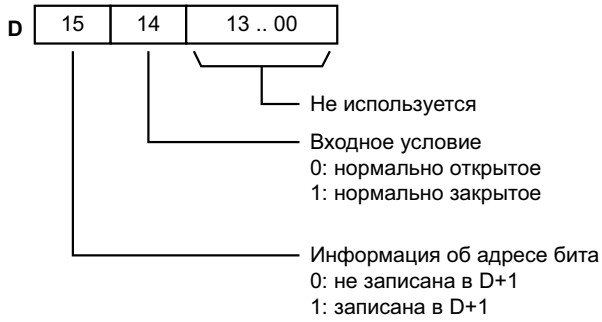
Когда IR 00000..IR 00003 = 1, в качестве причины невключения в состояние 1 выхода диагностики будет указано нормально открытое условие IR 00002.

Вывод диагностики

Есть 2 способа вывода адреса бита в состоянии 0, обнаруженного при диагностике.

1, 2, 3,... 1. Вывод адреса бита (когда бит 15 слова C = 0).

Бит 15 слова D указывает, загружена или нет в D+1 информация об адресе бита. Если да, бит 14 слова D указывает, является ли входное условие нормально открытым или нормально закрытым.



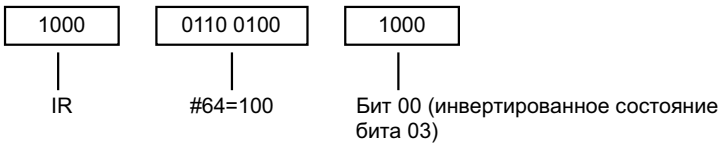
В D+1 содержится код адреса бита входного условия, как показано в таблице. Адреса, номера битов и номера таймеров/счетчиков - в двоичном виде.

Область данных	Состояние бита D+1															
	15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
IR, SR	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			
TC*	1	0	0	1	0	1	*	Номер таймера или счетчика								

Замечание 1. * Для области TC бит 09 слова D1 указывает, является ли номер номером таймера или счетчика. 0 - таймер, 1 - счетчик.

2. Старший бит в графе номер бита (бит 03) - резерв.

Пример: Если D+1 содержит 1000 0110 0100 1000, IR 10000 будет индикатироваться следующим образом:



2. Адрес бита и выдача сообщения (выбирается, когда бит 15 слова C = 1).

Бит 15 слова D 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 содержат информацию в ASCII, которая индикатируется 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 контроля

Далее описанную процедуру можно использовать для автоматического задания времени наблюдения, T, в текущих условиях работы, при задании словного операнда для T. Эту операцию нельзя использовать, если для T задана константа.

1, 2, 3,... 1. Переключите CQM1 в режим MONITOR.

2. Подключите периферийное устройство, например, программатор.

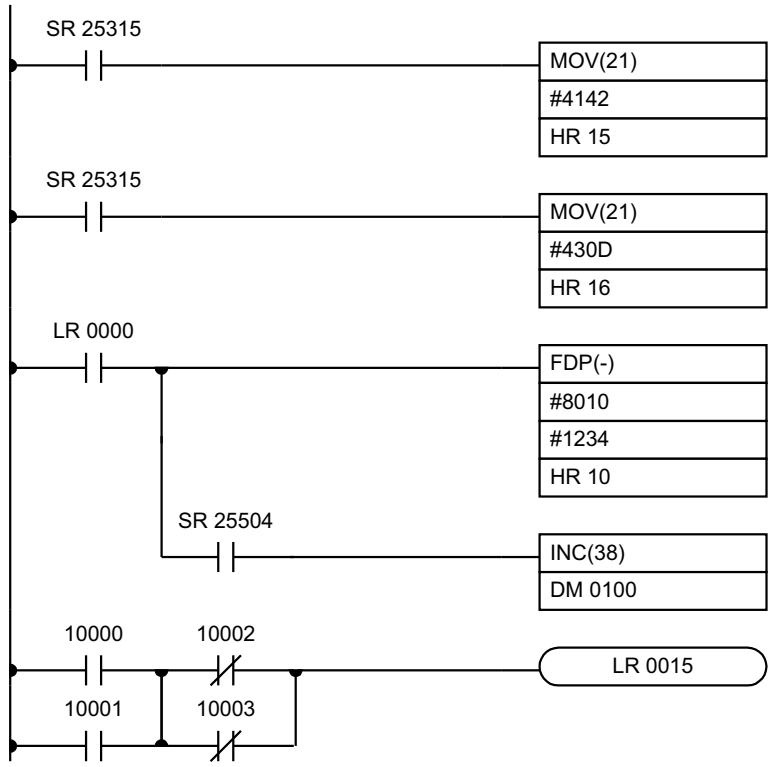
3. Используйте периферийное устройство для включения в 1 бита управление AR 2508.

4. Выполните программу с AR 2508 в положении 1. Если превышено время контроля, которое теперь в T, то в T будет загружено 1.5 фактического времени контроля. Пока AR 2508 = 1, признак ошибки FAL(06) не выдается.

5. Выключите AR 2508, когда в T загрузится приемлемое значение.

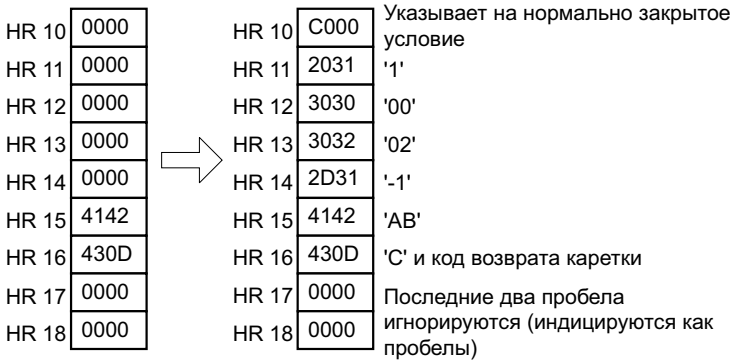
Пример

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	25315
00001	MOV(21)	
		# 4142
		HR 15
00002	LD	25315
00003	MOV(21)	
		# 430D
		HR 16
00004	LD	LR 0000
00005	FPD(-)	
		# 0010
		# 1234
		HR 10
00006	AND	25504
00007	INC(38)	
		DM 0100
00008	LD	10000
00009	OR	10001
00010	LD NOT	10002
00011	OR NOT	10003
00012	AND LD	
00013	OUT	LR 0015

FPD(-) выполняется и начинается контроль 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 ошибки, будет генерироваться признак ошибки FAL(006) с номером FAL 10, и адрес бита в текущем сообщении (“10002-1ABC”) будет индикатироваться на периферийном устройстве.



Флаги

ER: T не двоично-десятичное число

C 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).

CY: 1, когда время между исполнением FPD(-) и исполнением выдачи диагностики превышает T.

5.26.8 INT(89) - Управление прерываниями

Обозначение на схеме

INT(89)
CC
000
D

@INT(89)
CC
000
D

Область операндов

CC	код управления	#(000..003, 100, 200)
000	нет функций	
D	слово состояния	IR, SR, AR, HR, TC, LR, TR, DM, #

Ограничения

DM 6144..DM 6655 нельзя использовать для D когда CC = 002.

Описание

Когда условие исполнение = 0, INT(89) не выполняется. Когда условие исполнения = 1, INT(89) служит для управления прерываниями и исполняет одну из 6 функций, показанных в таблице, в зависимости от значений CC.

Функции INT(89)	**CC**
Маскировать/размаскировать входные прерывания	000
Очистить входные прерывания	001
Читать текущее состоянии маски	002
Возобновить заданное значение счетчика	003
Замаскировать все прерывания	100
Размаскировать все прерывания	200

Данные 6 функций описаны подробно описаны далее. Дополнительную информацию об этих функциях см. 1-4, 1-5.

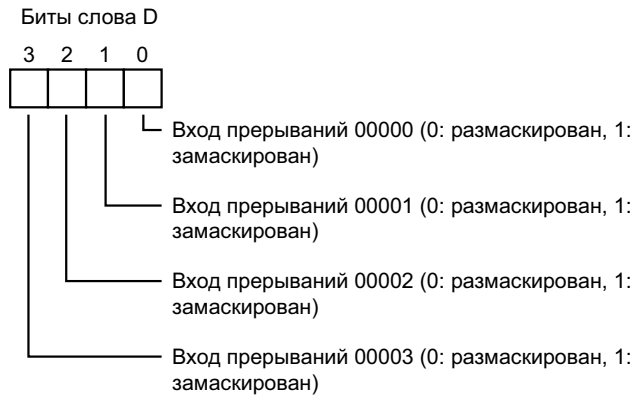
Маскировать/ размаскировать входные прерывания (CC=000)

Данная функция используется для маскирования и размаскирования входов прерываний 00000..00003 (00003..00006 в CPM1). Замаскированные прерывания запоминаются, но игнорируются. Когда один из входов замаскирован, программа прерываний для него

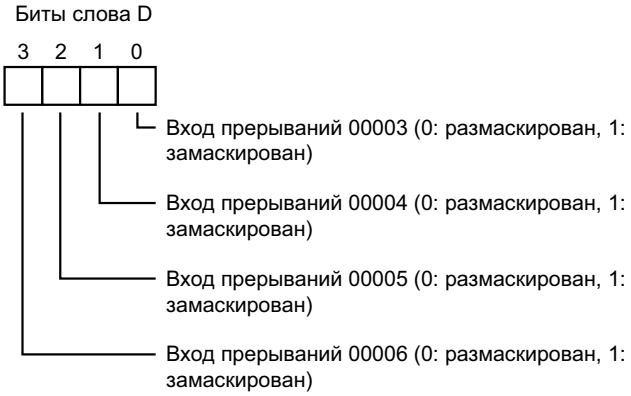
будет исполняться только после размаскирования прерываний (если только он предварительно не очищен командой INT(89) с CC = 001).

Установите соответствующий бит в D в 0 или 1 для маскирования или размаскирования входа прерывания. Биты 00..03 соответствуют входам 00000..00003 (00003..00006 в CPM1). Биты 04..15 должны быть установлены в 0.

CQM1



CPM1

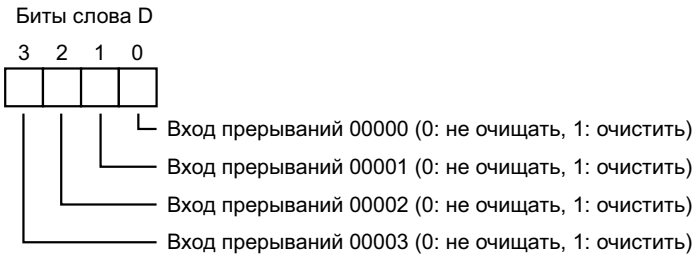


Очистить входные прерывания (CC=001)

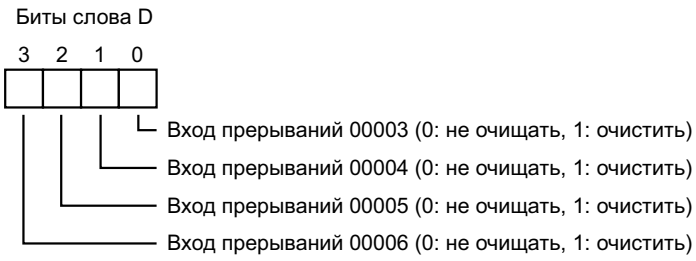
Данная функция используется для очистки входов входных прерываний 00000..00003 (00003..00006 в CPM1). Поскольку входы прерываний запоминаются, замаскированные прерывания будут обслужены после снятия маски, если только они предварительно не очищены.

Установите соответствующий бит в D в 1 для очистки входного прерывания. Биты 00..03 соответствуют входам 00000..00003 (00003..00006 в CPM1). Биты 04..15 должны быть установлены в 0.

CQM1



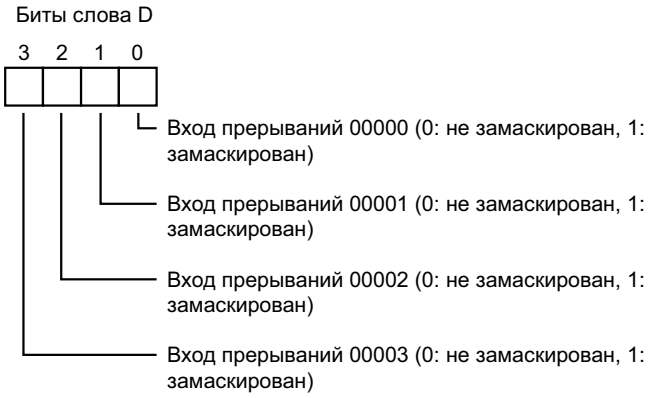
CPM1



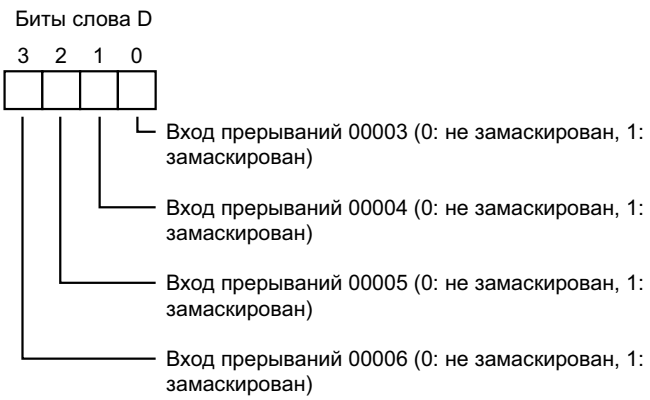
Читать текущее состояние маски (CC=002)

Данная функция используется для записи текущего состояния маски входов прерываний 00000..00003 (00003..00006 в CPM1) в слово D. Соответствующий бит будет 1, если вход замаскирован. (Биты 00..03 соответствуют входам 00000..00003 (00003..00006 в CPM1).

CQM1



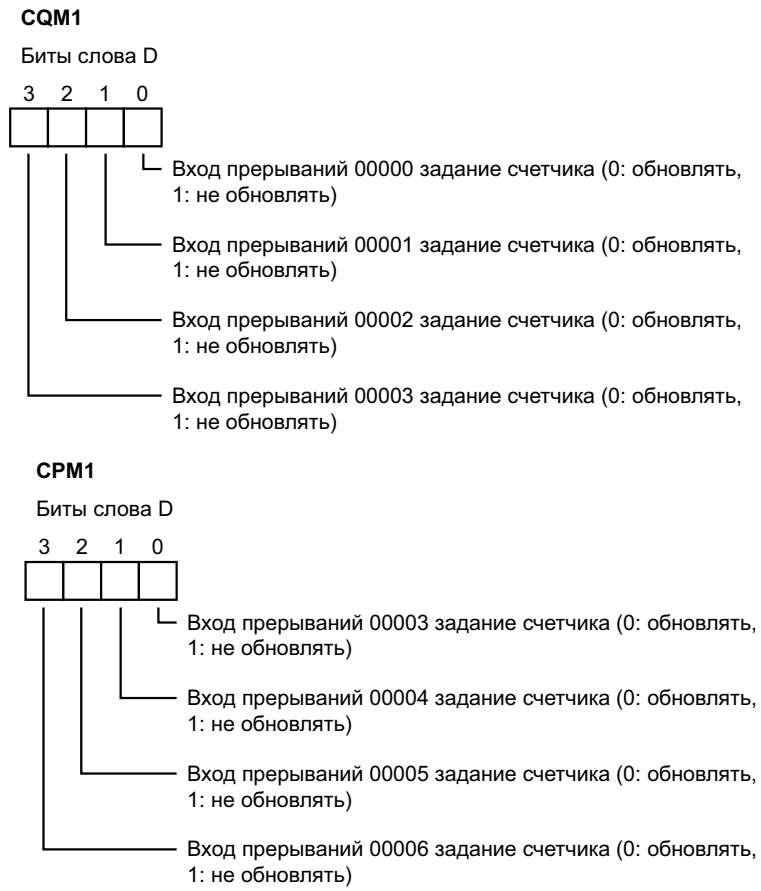
CPM1



Обновить задание счетчиков (CC=003)

Данная функция используется для обновления задания счетчика для входов прерываний 00000..00003 (00003..00006 в CPM1) в соответствии со словом D. Задайте соответствующий бит слова D в 1, чтобы обновить задание счетчика.

(Биты 00..03 соответствуют входам 00000..00003 (00003..00006 в CPM1).



Маскировать/ размаскировать все прерывания (CC=100/200)

Данная функция используется для маскировки/ размаскировки обработки всех прерываний. Запросы прерываний с замаскированных входов запоминаются, но игнорируются. Подробности см. 1-4-3. Параметры управления D в данной функции не используются. Задайте D = #0000.

Флаги

ER: Некорректное задание счетчика (только для CC = 003).

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

CC = 100 или 200, когда выполняется программа прерывания.

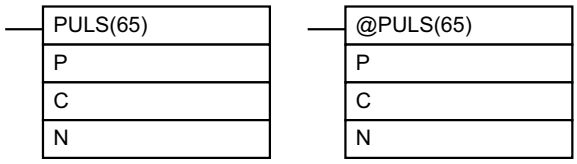
CC = 100, когда все прерывания уже замаскированы.

CC = 200, когда все входы уже размаскированы.

CC и/или D не в допустимых пределах.

5.26.9 PULS(65) - Задание импульсов

Обозначение на схеме



Область операндов

P	указатель порта	000, 001, 002
C	параметры управления	000 .. 005
N	число импульсов	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

N и N+1 должны лежать в одной области данных

DM 6144..DM 6655 нельзя использовать для N.

Описание

PULS(65) используется для установки параметров выдачи импульсов, выдача которых будет запущена позднее в программе командой SPED(64) или ACC(-). Задаются количество импульсов, которые будут выданы в “независимом режиме”, направление выдачи импульсов с портов 1 и 2 и точка замедления выдачи, управляемая командой ACC(-), режим 0.

Поскольку PULS(65) имеет довольно большое время исполнения, при необходимости время цикла можно сократить, применив версию команды @ PULS(65) (фронта 0/1).

Замедление Подробности см. 1-3.

Указатель порта (P)

Указатель порта задает место выдачи импульсов. Параметры, заданные C и N, будут действовать на последующие команды SPED(64) или ACC(-), в которых заданы те же места выдачи импульсов.

P	Место выдачи импульсов
000	Выходной бит
001	Порт 1
002	Порт 2

Параметр управления (C)

Параметр управления определяет направление выдачи импульсов с портов 1 и 2 и указывают, заданы ли в N..N+3 количество импульсов и/или точка замедления. Данный операнд должен быть 000, когда P=000.

C	Направление	Число импульсов	Точка замедления
000	По ЧС	Задано в N и N+1	Не задано
001	Против ЧС	Задано в N и N+1	Не задано
002	По ЧС	Задано в N и N+1	Задано в N+2 и N+3
003	Против ЧС	Задано в N и N+1	Задано в N+2 и N+3
004	По ЧС	Не задано	Не задано
005	Против ЧС	Не задано	Не задано

Задание направление действует до останова выполнения программы или до нового исполнения PULS(65).

Число импульсов и точка замедления

Когда C=000..003, N+1, N содержит 8-разрядное число импульсов для “независимого режима” выдачи импульсов. N+1, N могут быть 00000001..16777215. Выдача импульсов,

запущенная командами SPED(64) или ACC(-), автоматически прекратится, когда будет выдано заданное количество импульсов.

	4 левых цифры	4 правых цифры	Допустимый диапазон
Число импульсов:	N+1	N	0000 0001.. 1677 7215

Когда C=002 или 003, N+3, N+2 содержит 8-разрядное число импульсов для точки замедления, использующееся в режиме 0 ACC(-). N+3, N+2 может быть 0000 0001 до 1677 7215. Выдача импульсов, запущенная командой ACC(-), начнется замедление, когда будет выдано заданное количество импульсов.

	4 левых цифры	4 правых цифры	Допустимый диапазон
Точка замедления:	N+1	N	0000 0001.. 1677 7215

Когда C=004 или 004, не задаются ни количество импульсов, ни точки замедления. Задайте N=000, когда C=004 или 005.

Изменения частоты

Количество импульсов, заданное для выдачи, будет использоваться даже если команда SPED(64) используется для изменения частоты импульсов во время операции.

Например, если задано число импульсов 2 100 и частота меняется с 1 кГц на 100 Гц, выдача импульсов прекратится через:

12 с, если частота меняется после 1 с на 1 кГц.

3 с, если частота меняется после 2 с на 1 кГц.

Флаги

ER: Ошибка в задании команды.

Если превышена граница области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

PULS(65) выполняется в подпрограмме прерываний, в то время как в главной программе проводится ввод/выдача импульсов или команды высокоскоростного счетчика.

5.26.10 SPED(64) - Запуск выдачи импульсов

Обозначение на схеме

SPEED(64)
P
M
F

@SPEED(64)
P
M
F

Область операндов

P	указатель порта	001, 002, 010 .. 150
M	режим выдачи	000 .. 005
F	частота импульсов	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

F должно быть двоично-десятичным числом #0000..#5000, когда задан порт, и #0000 или #0002..#0010, когда задан выходной бит.

DM 6144..DM 6655 нельзя использовать для F.

Описание

SPED(64) используется для задания, изменения или остановки выдачи импульсов с указанного порта или выходного бита. Когда условие исполнения = 0, команда SPED(64)

не выполняется. Когда условие исполнения = 1, команда SPED(64) задает частоту импульсов F на указанный порт или выходной бит, заданную P. M задает режим выдачи.

Поскольку SPED(64) имеет довольно большое время исполнения, при необходимости время цикла можно сократить, применив версию команды @ PULS(65) (фронта 0/1).

Замечание Подробности см. 1.3.

Указатель порта (P)

Указатель порта задает порт или выходной бит, откуда будут выдаваться импульсы.

P	Место выдачи импульсов
001	Порт 1
002	Порт 2
000..150	Выходные биты IR 10000..IR 10015. Первые 2 цифры P определяют, какой бит IR 100 будет битом выдачи, третья цифра P всегда устанавливается в 0. Например, P = 000 задает IR 10000, P = 010 задает IR 10001..и P = 150 задает бит IR 10015.

Режим выдачи (M)

Значение M определяет режим выдачи. Значение 000 задает “независимый режим”, значение 001 - режим непрерывной выдачи.

В “независимом режиме” выдача будет продолжаться до тех пор, пока случится что-либо из следующего:

1, 2, 3,... 1. Количество импульсов, заданное командой PULS(65), достигнуто. (Выполните PULS(65) перед SPED(64) при задании “независимого режима”.)

2. Выполнилась команда INI(61) с C=003.

3. Снова выполнилась SPED(64) с частотой импульсов = 000.

При выдаче импульсов в “независимом режиме” предварительно задайте число импульсов командой PULS(65). При выдаче с порта 1 или 2 задайте также направление по ЧС или против ЧС.

В “независимом режиме” число импульсов, выданных с портов 1 или 2, находятся в IR 236 и IR 237 (Порт 1) или в IR 238 и IR 239 (Порт 2).

	4 старшие цифры	4 младшие цифры
Текущее значение выданных импульсов с порта 1	IR 237	IR 236
Текущее значение выданных импульсов с порта 2	IR 239	IR 238

В непрерывном режиме импульсы будут выдаваться до исполнения команды INI(61) с C=003 или SPED(64) с заданием частоты = 0. Если направление (по ЧС или против ЧС) не задано при выдаче с портов 1 или 2, импульсы будут выдаваться по ЧС.

Частота импульсов (F)

Значение F задает частоту с дискретностью 10 Гц, как показано ниже. Задание частоты 0 прекращает выдачу импульсов с заданного выхода (порта).

Место выдачи	Возможные значения F
Порт 1 или 2	0000 (прекращает выдачу импульсов) или 0001...5000 (10 Гц...50 кГц).
Выходные биты	0000 (прекращает выдачу импульсов) или 0002...0100 (20 Гц...1 кГц).

Замечания по поводу импульсного выхода

Выходная частота с портов 1 и 2 получается делением частоты генератора (500kHz) на целое число, что вызывает разницу между заданной и действительной частотой. Обратите внимания на следующие уравнения для вычисления действительной частоты.

Заданная частота: выходная частота, заданная в инструкции.

Коэффициент деления: коэффициент (целое число), вычисляемый путем деления частоты генератора на заданную частоту

Действительная частота: реальная частота на выходе.

$$F_A = \frac{500kHz}{K_F};\ K_F = Int\ \frac{500kHz}{F_U}$$

FA: действительная частота;

FU: заданная частота;

KF: коэффициент деления.

Разница между заданной и действительной частотой с увеличением частоты растет.

Пример:

Заданная частота, kHz	**Реальная частота, kHz**
45.6..50.00	50.000
41.67..45.45	45.454
38.47..41.66	41.666
..	
31.26..33.33	33.333
29.42..31.25	31.250
27.78..29.41	29.411
..	
20.01..20.83	20.833
19.24..20.00	20.000
18.52..19.23	19.230
..	
10.01..10.20	10.204
9.81..10.00	10.000
9.62..9.80	9.803
..	
5.01..5.05	5.050
4.96..5.00	5.000
4.90..4.95	4.950
..	
3.02..3.03	3.030
3.00..3.01	3.012
2.98..2.99	2.994
..	

Предосторожности

У CQM1-CPU11/21-E метод обновления выходов (в DM 6639) должен быть задан на “прямую выдачу” перед инициализацией выдачи импульсов.

Выдачу импульсов нельзя задать, когда работает интервальный таймер 0.

Когда выдача импульсов частотой 500 Гц и выше с выходного бита задана, задайте обработку прерываний для TIMH(15) с номерами TC 000..003, задавая #0104 в DM 6629.

Выдачу импульсов может осуществлять только на один выход одновременно.

Флаги

ER: SPED(64) выполнилась, когда работает интервальный таймер 0.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

Ошибка в задании команды.

SPED(64) выполняется в подпрограмме прерываний, в то время как в главной программе проводится выдача импульсов или команда высокоскоростного счетчика.

5.26.11 PLS2(-) - Выдача импульсов

Обозначение на схеме

PLS2(-)
P
D
C

@PLS2(-)
P
D
C

Область операндов

P	коммуникационный порт	001, 002
D	указатель направления	001, 002
C	первое слово управления	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1-CPU43-E.

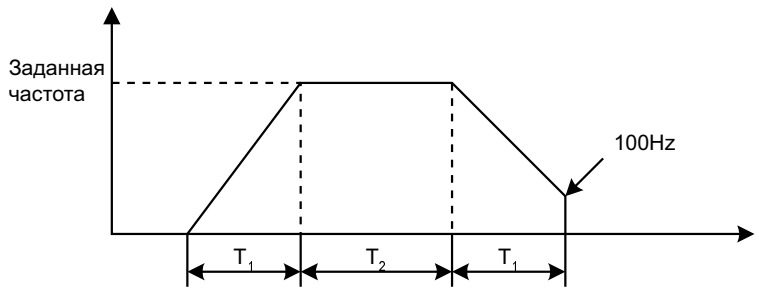
PLS2(-) нельзя использовать, если установочный параметр (DM 6611) задан на режим высокоскоростного счетчика.

P должен быть 001 или 002 и D должен быть 000 или 001.

C..C+3 должны находиться в одной области данных.

Описание

PLS2(-) используется для выдачи заданного числа импульсов по ЧС или против ЧС с портов 1 или 2. Частота импульсов увеличивается от 0 до заданной частоты с заданным ускорением и уменьшается с таким же ускорением. (Выдача прекращается при 100 Гц).



Следующие уравнения показывают, как вычислить приблизительное время ускорения/замедления частоты T1 и время выдачи T2. Время задано в секундах.

$$T_1 \cong 0.004 \times \frac{T}{K};\ T_2 \cong \frac{N-(T_1 \times T)}{T}$$

Где: T - заданная частота; K - коэффициент ускорения/замедления; N - число импульсов.

Замечание 1. Хотя T1 и T2 будут слегка варьировать в зависимости от условий работы, число выданных импульсов будет точным.

2. PLS2(-) не будет работать, если импульсы уже выдаются с заданного порта. Перед выполнением PLS2(-) проверяйте флаги выдачи (AR 0515 для порта 1 и AR 0615 для порта 2).

3. Подробности см. 1-3.

Задание операндов

P задает порт, с которого будут выдаваться импульсы. Импульсы выдаются с порта 1, когда P=001 и с порта 2, когда P=002.

D задает направление импульсов (D = 000 - по ЧС, D = 001 - против ЧС).

Содержание C определяет коэффициент ускорения/замедления. При увеличении или замедлении частоты, частота увеличивается или уменьшается на значение, заданное в C, каждые 4.08 мс. C должно быть двоично-десятичным числом от 0001 до 0200 (10 Гц..2 кГц).

Содержание C+1 задает конечную частоту. C+1 должно быть двоично-десятичным числом от 0010 до 5000 (10 Гц..50 кГц).

8-разрядное содержимое C+3, C+2 определяет количество импульсов, которые должны быть выданы. C+3, C+2 должны быть двоично-десятичным числом от 0000 0001 до 1677 7215.

Флаги

ER: Ошибка в задании операндов.

ЦУ не CQM1-CPU43-E.

Установочные параметры задают не выдачу импульсов.

Заданная частота, коэффициент ускорения/замедления и число импульсов некорректно (Число импульсов T1 x Заданную частоту).

PLS2(-) выполняется в подпрограмме прерываний, в то время как в главной программе проводится ввод/выдача импульсов или команда высокоскоростного счетчика.

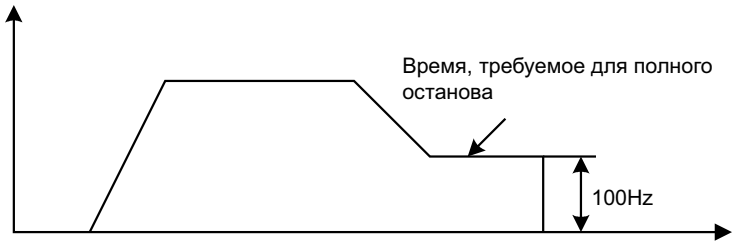
Значения вышли за границу области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

AR 0515 Флаг выдачи с порта 1. 1, когда импульсы выдаются с порта 1.

AR 0615 Флаг выдачи с порта 2. 1, когда импульсы выдаются с порта 2.

Внимание! При работе команды PLS2(-) такие факторы, как скорость ускорения/замедления и заданная частота могут вызвать продолжительную выдачу импульсов низкой частоты (100 Гц) при остановке. Даже если это произойдет, выдано будет заданное число импульсов.



Подкорректируйте систему, настроив темп ускорения/замедления и/или заданную скорость или используя для торможения команду ACC(-) режим 0 для увеличении скорости (конечной частоты) при останове.

5.26.12 ACC(-) - Управление ускорением

Обозначение на схеме

ACC(-)
P
M
C

@ACC(-)
P
M
C

Область операндов

P	коммуникационный порт	001, 002
M	указатель режима	000 .. 003
C	первое слово управления	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1-CPU43-E.

Режим 0 ACC(-) нельзя использовать, если установочный параметр (DM 6611) задан на режим высокоскоростного счетчика.

P должно быть 001 или 002 и M должно быть 000..003.

C..C+3 должны находиться в одной области данных.

Описание

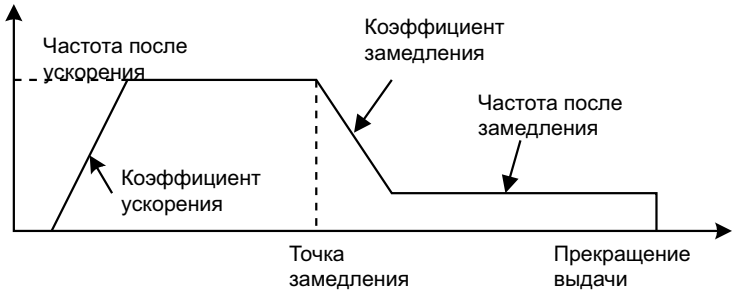
ACC(-) используется совместно с PULS(65) для управления ускорением и/или замедлением импульсов с портов 1 или 2. Есть 4 режима, кратко описанные далее.

Функция слов управления может принимать значение одного из режимов, но P всегда задает порт, с которого выдаются импульсы, а M всегда задает режим. Задавайте P = 001 или 002 для указания порта 1 или 2. Задайте M = 000..003 для задания режимов 0...3.

Замедление Подробности см. 1-3.

Режим 0 (M=000)

Режим 0 служит для выдачи заданного количества импульсов по ЧС или после ЧС с портов 1 или 2. Можно управлять коэффициентом ускорения, частотой после ускорения, точкой замедления, частотой замедления, частотой после замедления.



Задание операндов PULS(65)

PULS(65) должна быть выполнена перед ACC(-), чтобы задать направление, общее число импульсов для выдачи, и точку замедления. Функция операндов описаны ниже. Подробности см. 5-26-9.

1, 2, 3,... 1. Первый операнд PULS(65) задает порт выдачи. Импульсы выдаются с порта 1, когда P=001 и с порта 2, когда P=002.

2. Второй операнд задает направление. Выдача будет по ЧС, когда C=002 и против ЧС, когда C=003.

3. Третий операнд задает первое из 4 слов управления.

а) 8-разрядное содержимое N+1,N (0000 0001..1677 7215) определяет общее число импульсов, которые будут выданы.

б) 8-разрядное содержимое N+3,N+2 (0000 0001..1677 7215) определяет точку замедления.

Слова управления ACC(-)

4 слова управления задают коэффициент ускорения, частоту после ускорения, коэффициент замедления, частоту после замедления.

1, 2, 3,... 1. Содержимое C задает коэффициент ускорения. При разгоне выходная частота увеличивается на величину, заданную в C, каждые 4.08 с. C должно быть двоично-десятичным числом 0001..0200 (10 Гц...2 кГц).

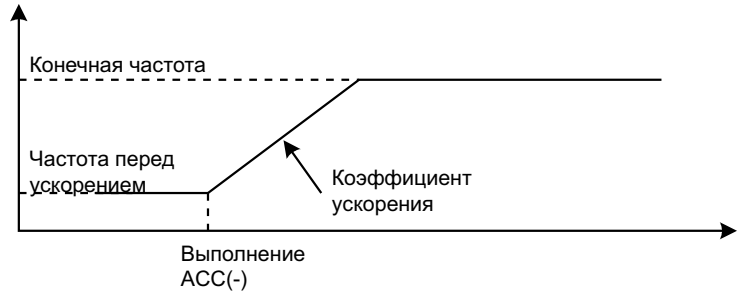
2. Содержимое C+1 задает частоту после ускорения. C+1 должно быть двоично-десятичным числом 0000..5000 (0 Гц...50 кГц).

3. Содержимое C+2 задает коэффициент замедления. При замедлении выходная частота уменьшается на величину, заданную в C, каждые 4.08 с. C+2 должно быть двоично-десятичным числом 0001..0200 (10 Гц...2 кГц).

4. Содержимое C+3 задает частоту после замедления. C+3 должно быть двоично-десятичным числом 0000..5000 (0 Гц...50 кГц).

Режим 1 (M=001)

Режим 1 служит для увеличения частоты до заданной с установленным коэффициентом ускорения.. Выдача импульсов продолжается до остановки.



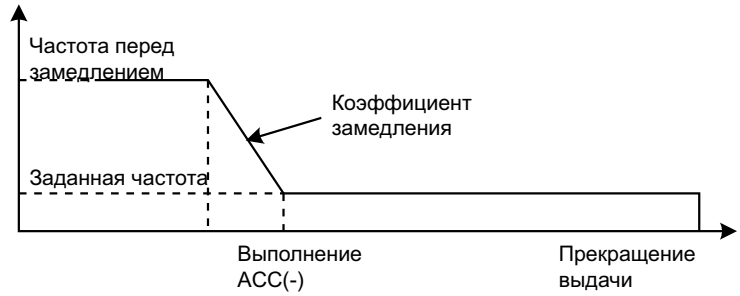
Два слова управления задают коэффициент ускорения и заданную частоту.

1, 2, 3,... 1. Содержимое C задает коэффициент ускорения. При разгоне выходная частота увеличивается на величину, заданную в C, каждые 4.08 с. C должно быть двоично-десятичным числом 0001..0200 (10 Гц...2 кГц).

2. Содержимое C+1 задает конечную частоту. C+1 должно быть двоично-десятичным числом 0000..5000 (0 Гц...50 кГц).

Режим 2 (M=002)

Режим 2 используется для уменьшения частоты до конечной с установленным коэффициентом.. Выдача импульсов прекратится после выдачи количества, заданного в PULS(65).



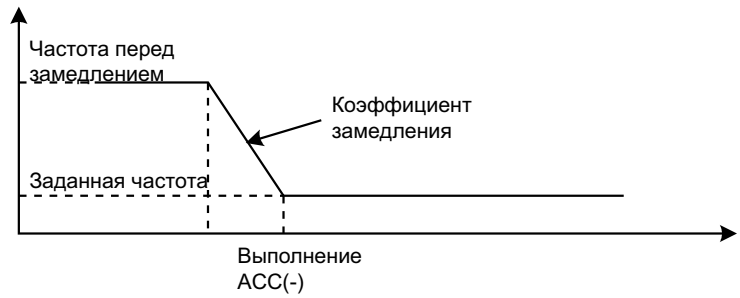
Два контрольных слова управления задают коэффициент замедления и заданную частоту.

1, 2, 3,... 1. Содержимое C задает коэффициент замедления. При замедлении выходная частота уменьшается на величину, заданную в C, каждые 4.08 с. C должно быть двоично-десятичным числом 0001..0200 (10 Гц...2 кГц).

2. Содержимое C+1 задает конечную частоту. C+1 должно быть двоично-десятичным числом 0000..5000 (0 Гц...50 кГц).

Режим 3 (M=003)

Режим 3 служит для уменьшения частоты до заданной с установленным коэффициентом. Выдача импульсов продолжается до задания остановки.



Два слова управления задают коэффициент замедления и заданную частоту.

1, 2, 3,... 1. Содержимое C задает коэффициент замедления. При замедлении выходная частота уменьшается на величину, заданную в C, каждые 4.08 мс. C должно быть двоично-десятичным числом 0001..0200 (10 Гц...2 кГц).

2. Содержимое C+1 задает конечную частоту. C+1 должно быть двоично-десятичным числом 0000..5000 (0 Гц...50 кГц).

Флаги

ER: Ошибка в задании операндов.

ЦУ не CQM1-CPU43-E.

Установочные параметры задают не выдачу импульсов.

ACC(-) выполняется в подпрограмме прерываний, в то время как в главной программе проводится ввод/выдача импульсов или команда высокоскоростного счетчика.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

AR 0515 Флаг выдачи с порта 1. 1, когда импульсы выдаются с порта 1.

AR 0615 Флаг выдачи с порта 2. 1, когда импульсы выдаются с порта 2.

5.26.13 PWM(-) - Импульсы с переменным коэффициентом заполнения

Обозначение на схеме

PWM(-)
P
F
D

@PWM(-)
P
F
D

Область операндов

P	коммуникационный порт	001, 002
F	частота	001, 002, 002
D	коэффициент заполнения	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

Данная команда есть только в CQM1-CPU43-E.

PWM(-) нельзя использовать, если в установочных параметрах (DM 6643 или 6644) не задан режим выдачи импульсов с переменным коэффициентом заполнения.

P должно быть 001 или 002,а F должно быть 000, 002 или 003.

D должно быть двоично-десятичным числом от 0001 до 0099

Описание

PWM(-) используется для выдачи импульсов с переменным коэффициентом заполнения с потов 1 или 2. Выдачу можно задать с тремя значениями частоты: 5.9 кГц, 1.5 кГц или 91.6 Гц. Выдача импульсов продолжается до остановки командой INI(61).

Для того, чтобы выполнялась команда PWM(-), заданный порт должен быть настроен в установочных параметрах на режим выдачи импульсов с переменным коэффициентом заполнения. Задайте старшую цифру DM 6643 в 1 для разрешения выдачи импульсов с переменным коэффициентом заполнения. с порта 1 или старшую цифру DM 6644 в 1 для разрешения выдачи импульсов с переменным коэффициентом заполнения. с порта 2. С порта, настроенного на режим выдачи импульсов с переменным коэффициентом заполнения, нельзя выдавать стандартные импульсы.

Замечание Подробности см. 1.3.

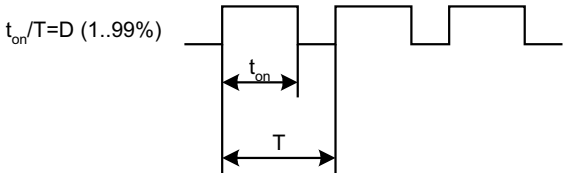
Задание операндов

P задает порт, с которого будут выдаваться импульсы. Импульсы выдаются с порта 1, когда P=001 и с порта 2, когда P=002.

F задает частоту импульсов, как показано в следующей таблице.

F	Частота
000	5.9 kHz
001	1.5 kHz
002	91.6 Hz

D задает коэффициент заполнения импульсов, т.е. процент времени, когда выход находится в состоянии 1. D должно быть двоично-десятичным числом от 0001 до 0099 (1%..99%).



Флаги

ER: Ошибка в задании операндов.

ЦУ не CQM1-CPU43-E.

Установочные параметры настроены не на выдачу импульсов с переменным коэффициентом заполнения.

PWM(-) выполняется в подпрограмме прерываний, в то время как в главной программе проводится ввод/выдача импульсов или команда высокоскоростного счетчика.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

5.26.14 SRCH(-) - Поиск данных

Обозначение на схеме

SRCH(-)
N
R_1
C

@SRCH(-)
N
R_1
C

Область операндов

N	количество слов	IR, SR, AR, HR, TC, LR, DM, #
R_1	первое слово в зоне	IR, SR, AR, HR, TC, LR, DM
C	параметры сравнения, слово результат	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1.

N должно быть двоично-десятичным числом от 0001 до 9999.

R1..R1+N-1 должны быть в одной области.

DM 6143..DM 6655 нельзя использовать для C.

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 по одному или более адресам содержатся сравниваемые данные, 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 слова 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 C+1 остается неизменным.

Флаги

ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

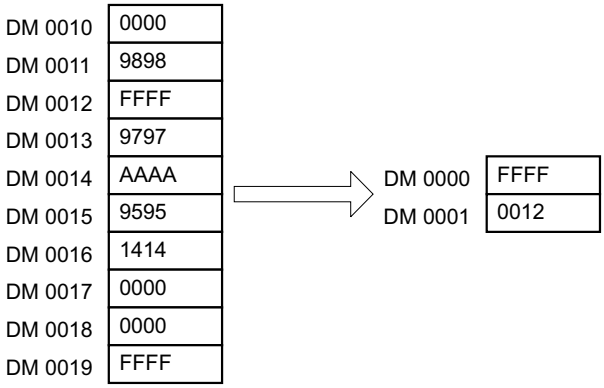
N не двоично-десятичное число от 0001 до 9999.

EQ: 1, когда сравниваемые данные есть в зоне поиска.

Пример

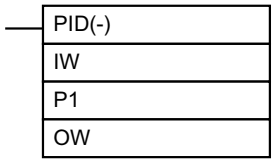
В следующем примере в зоне из 10 слов DM 0010..DM 0019 ищутся адреса, в которых содержатся одинаковые данные с DM 0000 (#FFFF). Поскольку в DM 0012 содержатся такие же данные, флаг EQ (SR 25506) включается в 1 и в DM 0001 заносится #0012.

Адрес	Инструкция	Операнд
00000	LD	00001
00001	@SRCH(-)	
		# 0010
		DM 0010
		DM 0000



5.26.15 PID(-) - ПИД-регулирование(пропорционально-интегрально-дифференциальное)

Обозначение на схеме



Область операндов

IW	слово входных данных	IR, SR, AR, HR, LR, DM
P1	первый параметр	IR, SR, HR, LR, DM
OW	слово выходных данных	IR, SR, AR, HR, LR, DM

Ограничения

Данная команда есть только в CQM1-CPU4_-E.

DM 6144..DM 6655 нельзя использовать для IW, P1..P1+32 или OW.

P1..P1+32 должны быть в одной области.

Внимание! Для правильной работы PID(-) нужны 33 последовательных слова, начиная с P1. Кроме того, PID(-) не может надежно работать в следующих условиях:

в программах прерываний;

в подпрограммах;

между IL(02) и ILC(03);

между JMP(04) и JME(05);

в секциях программы STEP(08)..STEP(09).

Не программируйте PID(-) в данных ситуациях.

Описание

PID(-) осуществляет PID- управление, основанное на параметрах, заданных в P1 ..... P1+6. Данные в IW используются для вычисления выходных параметров, которые заносятся в OW. В следующей таблице показаны функции параметров.

Слово	Биты	Название параметра	Функция/диапазон задания
P1	00 - 15	Задание (SV)	Конечное значение для ПИД-регулирования. Его можно задать любым двоичным числом с номерами битов, заданными параметром входной зоны.
P1+1	00 - 15	Ширина пропорционального диапазона	Данный параметр задает Ширину пропорционального диапазона/коэффициент зоны входа от 0.1% до 999.9%. Должен быть двоично-десятичным числом от 0001 до 9999.
P1+2	00 - 15	Время интегрирования	Задает интегральное время / коэффициент периода выборки, который используется для интегрального управления. Должен быть двоично-десятичным числом от 0001 до 8191 или 9999. (9999 запрещает интегральное управление).
P1+3	00 - 15	Время дифференцирования	Задает Время дифференцирования / коэффициент периода выборки, которая используется в дифференциальном управлении. Должен быть двоично-десятичным числом от 0001 до 8191.
P1+4	00 - 15	Период выборки	Задает интервал между сбором входных данных от 0.1 для 102.3 с. Должен быть двоично-десятичным числом от 0001 до 1023.
P1+5	00 - 03	Определитель операции	Задает нормальную или инверсную операцию. 0: инверсная операция 1: нормальная операция.
	04 - 15	Коэффициент входного фильтра	Задает степень входного фильтра. Чем меньше коэффициент, тем слабее фильтр. Должен быть двоично-десятичным числом от 100 до 999 или 000. Задание 000 задает значения по умолчанию (0.65), а задание от 100 до 999 задает коэффициент от 0.00 до 0.99
P1+6	00 - 07	Выходной диапазон.	Задет число бит выходных данных. Задание должно быть от 00 до 08, что задает выходной диапазон от 8 до 16 бит.
	08 - 15	Входной диапазон.	Задет число бит входных данных. Задание должно быть от 00 до 08, что задает входной диапазон от 8 до 16 бит.
P1+7 P1+32	00 - 15	Рабочая область	Не использовать (Используется системой).

Когда условие исполнения = 0, команда PID(-) не выполняется и параметры команды сохраняются. Когда условие исполнения = 0, желаемые параметры выхода можно записать прямо в OW для ручного управления.

Когда условие исполнения в первый раз переходят из 0 в 1, команда PID(-) считывает параметры и инициализирует рабочую область. Имеется встроенная функция для постоянного изменения выходных данных при пуске, поскольку резкие изменения в выходных данных могут неблагоприятно сказаться на управляемой системе.

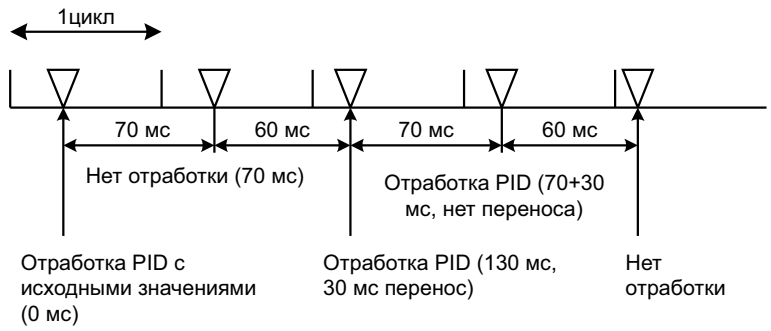
Внимание! Изменения параметров вступят в силу только после переключения условий исполнения PID(-) из 0 в 1.

Замечание Не используйте PID(-) в следующих ситуациях; команда может работать некорректно.

- в программах прерываний;
- в подпрограммы;
- между IL(02) и ILC(03);
- между JMP(04) и JME(05):
- в секциях программ STEP(08) и STEP(09)).

Когда условие исполнения = 1, команда PID(-) осуществляет ПИД-вычисления над входными данными по истечении периода выборки. Период выборки - время, которое должно пройти прежде чем входные данные считаются для обработки.

Следующий рисунок показывает соотношение между периодом выборки и ПИД-обработкой. ПИД-обработка производится только по истечении периода выборки (100 мс в данном примере).



Флаги

ER: Ошибка в задании параметров.

Время цикла более чем в 2 раза превышает период выборки, так что команда PID(-) не будет исполняться корректно. Однако PID(-) будет выполняться.

Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

CY: 1, когда PID(-) завершилась. (0, если период выборки не истек).

5.27 Команды связи

5.27.1 RXD(47) - Прием

Обозначение на схеме

RXD(47)
D
C
N

@RXD(47)
D
C
N

Область операндов

D	первое слово приемника	IR, SR, AR, HR, TC, LR, DM
C	слово состояния	#
N	количество байт	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

Данная команда есть только в CQM1.

D и D+(N2)-1 должны быть в одной области данных.

DM 6144..DM 6655 нельзя использовать для D и N.

N должно быть двоично-десятичным числом от #0000..#0256. (#0000..#0061 для режима HOST LINK).

Описание

Когда условие исполнения = 0, команда RXD(47) не выполняется. Когда условие исполнения = 1, команда RXD(47) считывает N байтов данных, полученных с порта, заданного в параметре управления, и затем записывает данные в слова D..D+(N2)-1. За раз можно читать до 256 байт данных.

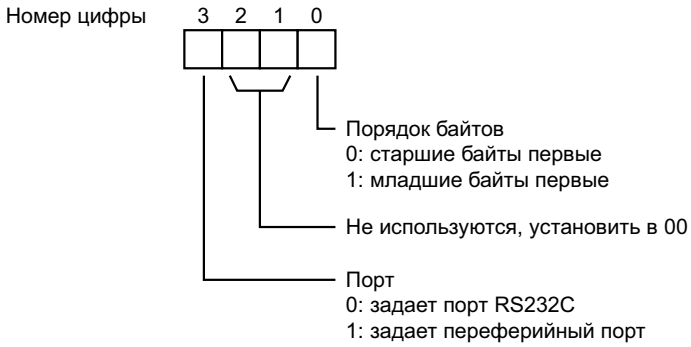
Если получены меньше, чем N байт, будут считаны принятые байты.

Подробности об использовании команды RXD(47), протокола связи в установочных параметрах и т. д. см. 1-4.

Внимание! CQM1 не сможет больше принимать данные, если принятые 256 байтов данных не прочитаны командой RXD(47). Считывайте данные как можно быстрее после установки в 1 Флага Завершения Приема (AR 0806 для порта RS-232C, AR 0814 для периферийного порта).

Слово управления

Значение слова управления определяет порт, с которого будут читаться данные и порядок, в котором данные будут записаны в памяти.



Порядок, в котором данные записываются в память, зависит от значений цифры 0 в слове C. Восемь байтов данных 12345678... будут записаны следующим образом:

Разряд 0 = 0

	MSB	**LSB**
D	1	2

	MSB	LSB
D+1	3	4
D+2	5	6
D+3	7	8
...		

Разряд 0 = 1

	MSB	LSB
D	2	1
D+1	4	3
D+2	6	5
D+3	8	7
...		

Флаги

ER: В ЦУ нет порта RS-232C.

К указанному порту не подключено другое 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+(N2)-1) вышли за пределы области данных.

AR 08: AR 0806 установится в 1, когда данные приняты нормально на порт RS-232C. Сбрасывается после исполнения RXD(47).

AR 0814 установится в 1, когда данные приняты нормально на периферийный порт. Сбрасывается после исполнения RXD(47).

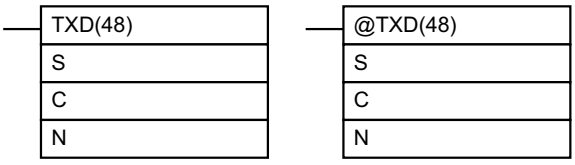
AR 09: Содержит число байтов, принятых на порт RS-232C. Сбрасывается в 0000 после исполнения RXD(47).

AR 10: Содержит число байтов, принятых на периферийный порт. Сбрасывается в 0000 после исполнения RXD(47).

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).

5.27.2 TXD(48) - Передача

Обозначение на схеме



Область операндов

S	первое слово источника	IR, SR, AR, HR, TC, LR, DM
C	слово состояния	#
N	количество байт	IR, SR, AR, HR, TC, LR, DM, #

Ограничения

Данная команда есть только в CQM1.

S и S+(N2)-1 должны быть в одной области

DM 6144..DM 6655 нельзя использовать для S и N.

N должно быть двоично-десятичным числом от #0000..#0256. (#0000..#0061 в режима связи HOST LINK).

Описание

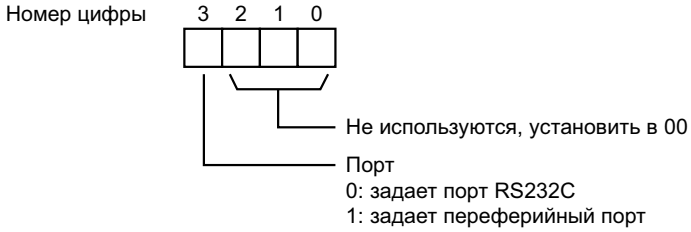
Когда условие исполнения = 0, команда TXD(48) не выполняется. Когда условие исполнения = 1, команда TXD(48) считывает N байтов данных из слов S до S+(N2)-1, преобразует в ASCII и выдает данные на заданный порт. TXD(48) работает по-разному в режиме связи HOST LINK и режиме RS-232C, поэтому данные режимы рассмотрены отдельно.

Подробности об использовании команды RXD(47), задания протокола связи в установочных параметрах и т. д. см. 1-4.

Замечание Флаг AR 0805 будет в состоянии 1, когда CQM1 готов передавать данные по порту RS-232C и Флаг AR 0813 будет в состоянии 1, когда CQM1 готов передавать данные через периферийный порт.

Режим HOST LINK

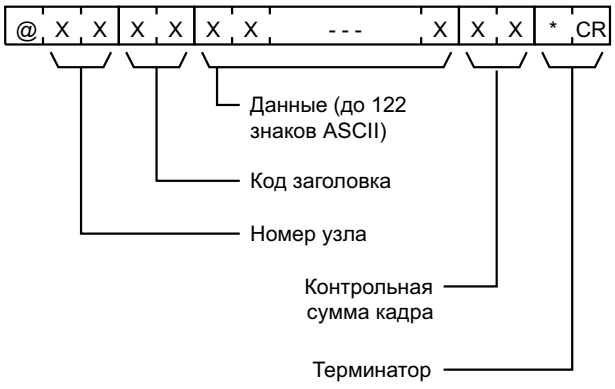
N должно быть двоично-десятичным числом от #0000 до #0061 (т.е. до 122 байт ASCII). Значение слова управления задает порт, с которого должны передаваться данные, как показано ниже.



Количество байт будет считано с S до S+(N2)-1, преобразовано в ASCII и выдано на заданный порт. Байты источника, приведенные в таблице, буду переданы в порядке: 12345678...

	MSB	**LSB**
S	1	2
S+1	3	4
S+2	5	6
S+3	7	8
...		

На следующим рисунке показан формат команды связи HOST LINK (TDX), посланный с CQM1. CQM1 автоматически пристыковывает префиксы и суффиксы, а именно номер узла, заголовок и контрольную сумму кадра.

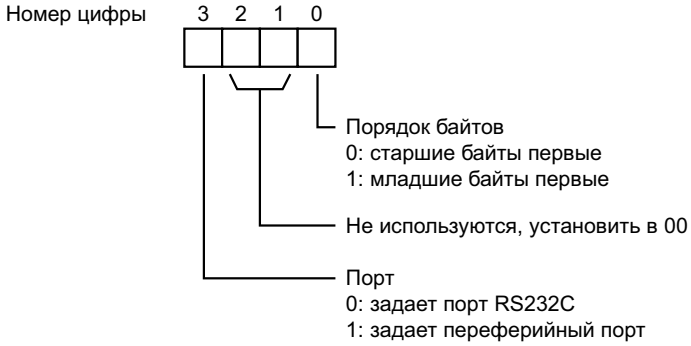


Режим RS-232C

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.

Слово управления

Значение слова управления задает порт, с которого должны передаваться данные, как показано ниже.



Заданное количество байт будет считано с S до S+(N2)-1 и передано через заданный порт.

	MSB	**LSB**
S	1	2
S+1	3	4
S+2	5	6
S+3	7	8
...		

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 макс. 256 байт, включая коды начала и конца.

Флаги

ER: В ЦУ нет порта RS-232C.

К указанному порту не подключено другое устройство.

Ошибка в задании установочных параметров связи или оперативных параметров.

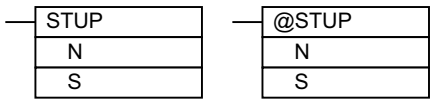
Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

Слова источника (S..S+(N2)-1) вышли за пределы области данных.

AR 08: AR 0805 установится в 1, когда можно передавать данные через порт RS-232C. AR 0813 установится в 1, когда можно передавать данные через периферийный порт.

5.27.3 STUP Изменение установочных параметров RS-232C

Обозначение на схеме



	Номер инструкции	Операнды		
Контроллер	Nr	N	S	
		определитель порта	первое слово источника	
CPM1/CPM1A				
SRM1	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	
CQM1				

Ограничения

Данная команда есть только в SRM1.

N должно быть 000, 001 или 002 для выбора 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 RS-232C будет изменена.

N	Заданный порт
IR 000	Встроенный порт RS-232C (параметры DM 6645 ... DM 6649)
IR 001	Порт A панели связи (параметры DM 6555 ... DM 6559)
IR 002	Порт B панели связи (параметры DM 6550 ... DM 6554)

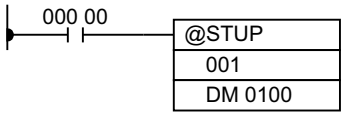
Если S - адрес слова, содержимое S ... S+4 копируется в 5 слов области параметров, которые содержат параметры для порта, заданного в N.

Если S - константа #0000, установочные параметры указанного порта устанавливаются по умолчанию.

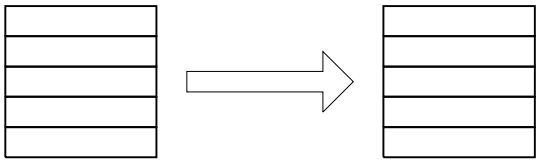
S	**Функция**
Адрес слова	Содержание S ... S+4 копируется в область установочных параметров ПК, которые содержат параметры для порта, заданного в N.
Константа #0000	Установочные параметры указанного порта устанавливаются по умолчанию.

Пример применения

В данном примере показана программ, которая передает содержимое DM 0100 ... DM 0104 в область установочных параметров порта A панели связи (DM 6555 ... DM 6559).



Параметры передаются как показано ниже. Флаг изменения параметров RS-232C (SR 27504) установится в 0 после завершения передачи.



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.

Слово	Содержание	Функция
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).

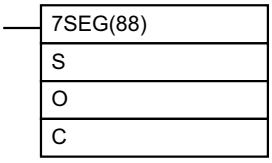
Слова источника превысили область данных.

Команда выполнялась из подпрограммы прерываний.

5.28 Дополнительные команды ввода/вывода

5.28.1 7SEG(88) - Вывод на 7-сегментный индикатор

Обозначение на схеме



Область операндов

S	первое слово источника	IR, SR, AR, HR, TC, LR, DM
O	выходное слово	IR, SR, AR, HR, TC, LR, DM
C	параметры управления	000 .. 007

Ограничения

Данная команда есть только в CQM1.

Не используйте команду 7SEG(88) в программе более, чем два раза.

Описание

Когда условие исполнения = 0, команда 7SEG(88) не выполняется. Когда условие исполнения = 1, команда 7SEG(88) читает данные источника (либо 4, либо 8 цифр), преобразует их в данные 7-сегметнтого 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.

Данные источника	Логика входов данных индикатора	Логика входов триггера индикатора	C
4 цифры (S)	Аналогична блоку выходов	Аналогична блоку выходов	000
		Отлична от блоков выходов	001
	Отлична от блоков выходов	Аналогична блоку выходов	002
		Отлична от блоков выходов	003
8 цифр (S, S+1)	Аналогична блоку выходов	Аналогична блоку выходов	004
		Отлична от блоков выходов	005
	Отлична от блоков выходов	Аналогична блоку выходов	006
		Отлична от блоков выходов	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(88) индикатирует 4-разрядные или 8-разрядные данные за 12 циклов, затем стартует опять и продолжает индикатировать данные.

Дополнительную информацию о команде 7SEG(88) и ее применении см. 2-2-4.

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(88).

5.28.2 DSW(87) - Ввод с цифрового переключателя

Обозначение на схеме

DSW(87)
IW
OW
R

Область операндов

IW	слово входа	IR, SR, AR, HR, TC, LR, DM
OW	слово выхода	IR, SR, AR, HR, TC, LR, DM
R	первое слово результата	IR, SR, AR, HR, TC, LR, DM

Ограничения

Данная команда есть только в CQM1.

DM 6144..DM 6655 нельзя использовать для R.

Описание

DSW(87) служит для чтения значения, набранного на цифровом переключателе, подключенном к входным/выходным блокам. Когда условие исполнения = 0, команда DSW(87) не выполняется. Когда условие исполнения = 1, команда DSW(87) считывает значение (4 или 8 цифр), набранное на цифровом переключателе, из IW и посылает результат в R.

Если значение 8-разрядное, они помещается в R и R+1, старшие цифры в R+1. Количество цифр задается в DM 6639.

DSW(87) считывает 4- или 8- разрядные данные за 12 циклов, затем стартует опять и продолжает считывать данные.

Дополнительную информацию о команде DSW(87) и ее применении см. 2-2-3.

Флаги

ER: Словам IW и OW отведены некорректные входные/выходные блоки.

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 (Когда CQM1 настроен на прием 8-разрядных данных).

SR 25410: 1, когда идет выполнение DSW(87).

5.28.3 HKY(-) - Ввод с клавиатуры на 16 клавиш

Обозначение на схеме

HKY(-)
IW
OW
D

Область операндов

IW	слово входа	IR, SR, AR, HR, TC, LR, DM
OW	слово выдачи управляющего сигнала	IR, SR, AR, HR, TC, LR, DM
D	первое слово регистра	IR, SR, AR, HR, TC, LR, DM

Ограничения

Данная команда есть только в CQM1.

D и D+2 должны находиться в одной области.

Не используйте HKY(-)в программе более, чем 2 раза.

DM 6144..DM 6655 нельзя использовать для D.

Описание

Когда условие исполнения = 0, команда HKY(-) не выполняется. Когда условие исполнения = 1, команда HKY(-) вводит данные с клавиатуры на 16 клавиш, подключенной к входу, заданному в IW. Данные можно вводить двумя способами:

1, 2, 3,... 1. 8-разрядный сдвиговый регистр создается в D и D+1. 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 Когда нажата одна из клавиш, ввод с других клавиш невозможен.

HKY(-) вводит каждую цифру от 3 до 12 циклов, затем стартует опять и продолжает ввод. Дополнительную информацию о команде HKY(-) и ее применении см. 2-2-2.

Флаги

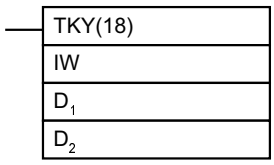
ER: Косвенно адресуемое слово DM не существует. (Содержимое слова *DM не в двоично-десятичном виде, или выходит за границу области DM).

D и D+2 находятся в разных областях.

SR 25408: 1, когда идет выполнение HKY(-).

5.28.4 TKY(18) - Ввод с клавиатуры на 10 клавиш

Обозначение на схеме



Область операндов

IW	слово входа	IR, SR, AR, HR, TC, LR, DM
D_1	первое слово регистра	IR, SR, AR, HR, TC, LR, DM
D_2	слово ввода клавиши	IR, SR, AR, HR, TC, LR, DM

Ограничения

Данная команда есть только в CQM1.

D1 и D1+1 должны находиться в одной области.

DM 6143 и DM 6655 нельзя использовать для D1.

Описание

Когда условие исполнения = 0, команда TKY(18) не выполняется. Когда условие исполнения = 1, команда TKY(18) вводит данные с клавиатуры на 10 клавиш, подключенной к входу, заданному в IW. Данные можно вводить двумя способами:

1, 2, 3,... 1. 8-разрядный сдвиговый регистр создается в D1 и D1+1. 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 указывают на ввод клавиши. Когда одна из клавиш (0..9) нажата, соответствующий бит в D2 (00..09) устанавливается в 1.

Замечание Когда нажата одна из клавиш, ввод с других клавиш невозможен.

TKY(18) можно использовать в нескольких местах в программе путем изменения слова входа, IW. Подробности о команде TKY(18) см. 2-2-1.

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 не в одной области данных.

6. Команды Host Link

В данной главе описаны методы и процедуры использования команд HOST LINK, которые можно использовать для связи HOST LINK через порты CQM1 и CPM1.

Список команд

Команды, перечисленные в следующей таблице, можно использовать для связи HOST LINK с CQM1 и CPM1. Все данные команды посылаются с управляющего компьютера в ПК. В таблице показано, в каких режимах ПК действуют команды HOST LINK:

Код заголовка	Наименование	Режим ПК			Тип ПК	Раздел
		RUN	MON	PRG		
RR	Чтение области IR/SR	Да	Да	Да	Все	6.3.1
RL	Чтение области LR	Да	Да	Да	Все	6.3.2
RH	Чтение области HR	Да	Да	Да	Все	6.3.3
RC	Чтение текущего значения	Да	Да	Да	Все	6.3.4
RG	Чтение состояния Т/Сч	Да	Да	Да	Все	6.3.5
RD	Чтение области DM	Да	Да	Да	Все	6.3.6
RJ	Чтение области AR	Да	Да	Да	Все	6.3.7
WR	Запись в область IR/SR	Нет	Да	Да	Все	6.3.8
WL	Запись в область LR	Нет	Да	Да	Все	6.3.9
WH	Запись в область HR	Нет	Да	Да	Все	6.3.10
WC	Запись текущего значения	Нет	Да	Да	Все	6.3.11
WG	Запись состояния Т/Сч	Нет	Да	Да	Все	6.3.12
WD	Запись в область DM	Нет	Да	Да	Все	6.3.13
WJ	Запись в область AR	Нет	Да	Да	Все	6.3.14
R#	Чтение 1 заданного значения	Да	Да	Да	Все	6.3.15
R$	Чтение 2 заданного значения	Да	Да	Да	Все	6.3.16
R%	Чтение 3 заданного значения	Да	Да	Да	Только CQM1	6.3.17
W#	Изменение 1 заданного значения	Нет	Да	Да	Все	6.3.18
W$	Изменение 2 заданного значения	Нет	Да	Да	Все	6.3.19
W%	Изменение 3 заданного значения	Нет	Да	Да	Только CQM1	6.3.20
MS	Чтение состояния	Да	Да	Да	Все	6.3.21
SC	Запись состояния	Да	Да	Да	Все	6.3.22
MF	Чтение признаков ошибок	Да	Да	Да	Все	6.3.23
KS	Принудительная установка	Нет	Да	Да	Все	6.3.24
KR	Принудительный сброс	Нет	Да	Да	Все	6.3.25
FK	Принудительные сброс/ установка нескольких адресов	Нет	Да	Да	Все	6.3.26
KC	Отмена принудительных установки/сброса	Нет	Да	Да	Все	6.3.27
MM	Читать модели ПК	Да	Да	Да	Все	6.3.28
TS	Тест	Да	Да	Да	Все	6.3.29
RP	Чтение программы	Да	Да	Да	Все	6.3.30
WP	Запись программы	Нет	Нет	Да	Все	6.3.31
QQ	Составная команда	Да	Да	Да	Все	6.3.32
XZ	Прервать (только команда)	Да	Да	Да	Все	6.3.33
**	Инициировать (только команда)	Да	Да	Да	Все	6.3.34
IC	Неопределенная команда (только ответ)	-	1	1	Все	6.3.35

6.1 Процедура связи

Связь HOST LINK осуществляется путем обмена командами и ответами между управляющим компьютером и ПК.

У CQM1 имеется два метода связи.

- Стандартный метод, в котором команды передаются с управляющего компьютера на ПК.
- Метод, который позволяет выдавать команды с ПК на компьютер.

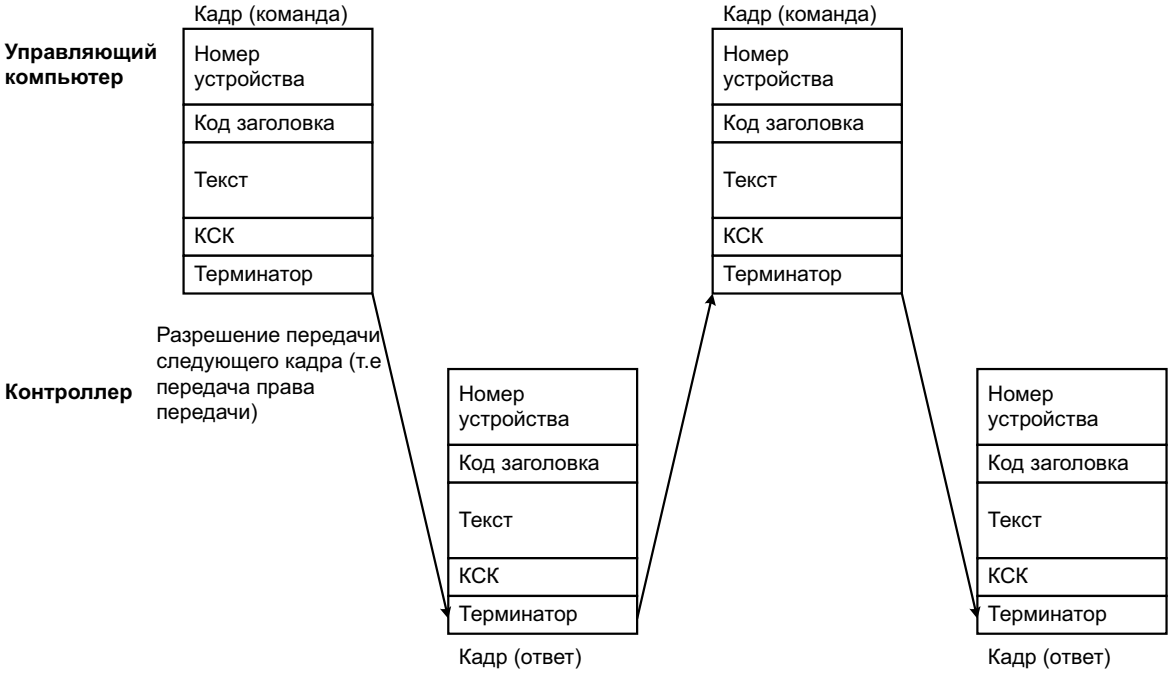
Передача и прием кадра

Обмен командами и ответами производится согласно следующей диаграммы. Блок данных, передаваемых за одну процедуру передачи, называется “Кадр”. Один кадр может состоять максимум из 131 символов.

Право передачи кадра называется “Право передачи”. Устройство, имеющее право передачи, может послать кадр в любое время. Право передачи передается туда и обратно между управляющим компьютером и ПК во время процедуры передачи. Право передачи переходит с передающего устройства на принимающее устройство после каждого терминатора (код, который отмечает конец команды или ответа) или разделителя (код, который разделяет кадры).

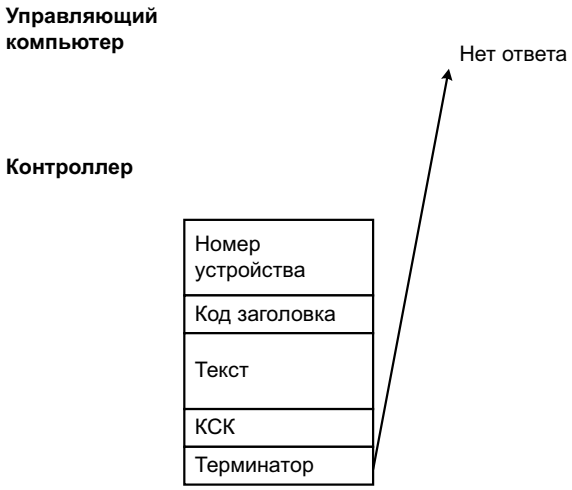
Команды от управляющего компьютера

В режиме связи HOST LINK управляющий компьютер, как правило, первым имеет право передачи и инициирует связь. Далее ПК автоматически посылает ответ.



Команды от ПК (Только CQM1)

У CQM1 в режиме связи HOST LINK также возможно посылать команды от ПК на управляющий компьютер. В данном случае право передачи принадлежит ПК и ПК инициирует процедуру связи.



Когда команды выдаются на управляющий компьютер, данные передаются в одном направлении от ПК на управляющий компьютер. Если требуется ответ на команду, используйте команды HOST LINK для записи ответа управляющего компьютера в ПК.

6.2 Форматы команд и ответов

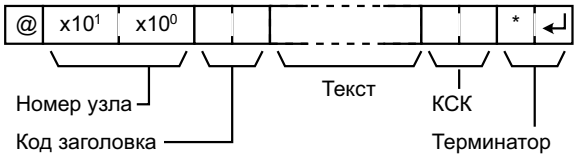
В данной главе описаны форматы команд и ответов, обмен которыми производятся при связи HOST LINK.

6.2.1 Команды с управляющего компьютера

Когда команда выдается с управляющего компьютера, форматы команд и ответов имеет следующий вид:

Формат команды

При передачи команд с управляющего компьютера данные следует готовить в следующем формате.



@

Символ ‘@’ должен быть помещен в начале команды.

Номер узла

Идентифицирует ПК, с которым производится процедура связи. Задавайте номер узла, заданный для ПК в установочных параметрах

Код заголовка

два символа кода заголовка.

Текст

Задайте текст команды.

КСК (Контрольная сумма кадра)

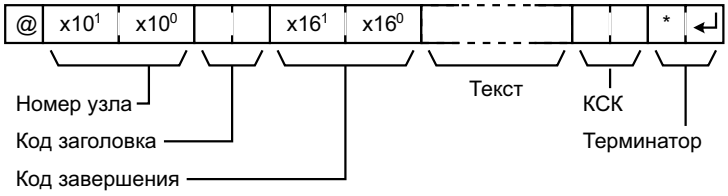
Задайте 2-символьный код контрольной суммы. См. 6-2-1, Контрольная сумма кадра.

Терминатор

Для обозначения окончания команды задайте 2 символа, “” и “возврат каретки” (CHR$(13)).

Формат ответа

Ответ с ПК возвращается в следующем формате. Для интерпретации и обработки параметров ответа приготовьте соответствующую программу.



@

Символ ‘@’ должен быть помещен в начале команды.

Номер узла

Идентифицирует ПК, с которым производится процедура связи. Задавайте номер узла, заданный для ПК в установочных параметрах

Код заголовка

два символа кода заголовка.

Код завершения

Возвращает статус завершения команды (напр., произошла или нет ошибка).

Текст

Задайте текст команды.

КСК (Контрольная сумма кадра)

Задайте 2-символьный код контрольной суммы. См. 6-2-1, Контрольная сумма кадра.

Терминатор

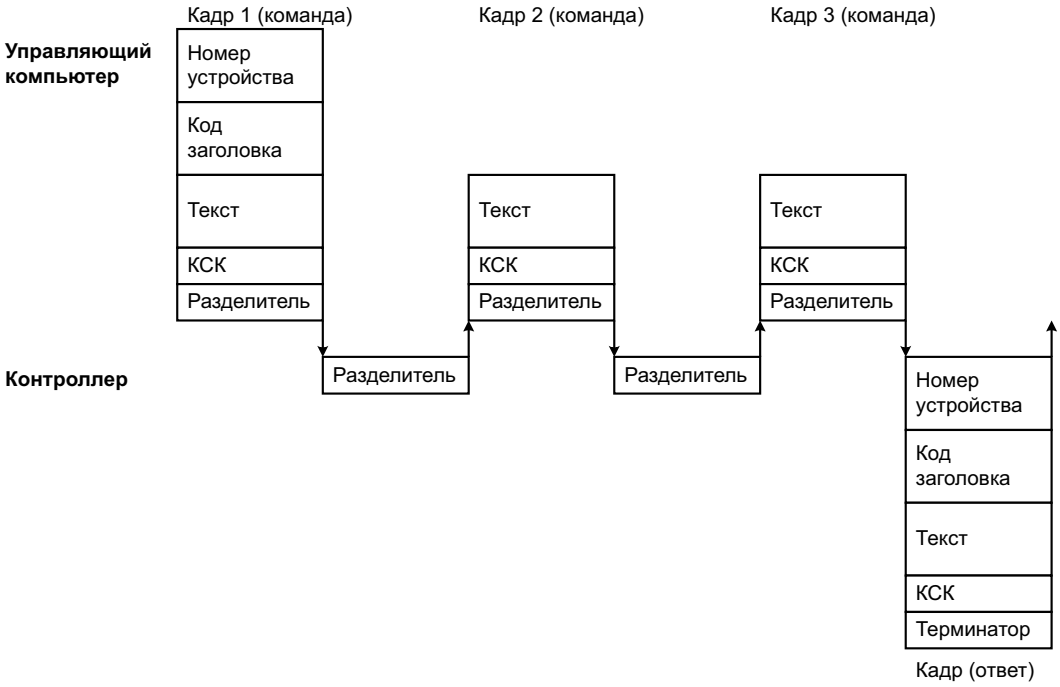
Для обозначения окончания команды задайте 2 символа, “” и “возврат каретки” (CHR$(13)).

Передача длинных блоков данных

Самый длинный блок данных, который можно передать как один кадр, содержит 131 символ. Следовательно, команда или ответ из 132 и более символов перед передачей нужно разделить на более, чем один кадр. При таком разделении окончания первого и соседнего кадров помечаются не терминатором, а разделителем.

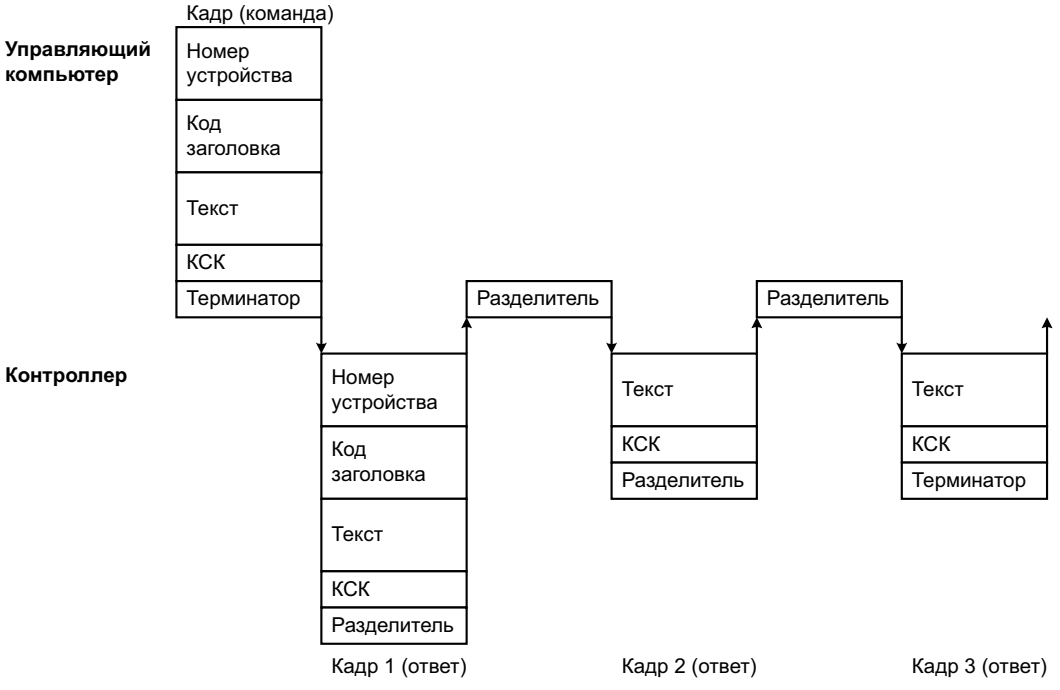
Разделенные команды (от управляющего компьютера к ПК)

Поскольку каждый кадр передается управляющим компьютером, компьютер ожидает передачи разделителя с ПК. После приема разделителя будет послан следующий кадр. Данная процедура повторяется до передачи полной команды.



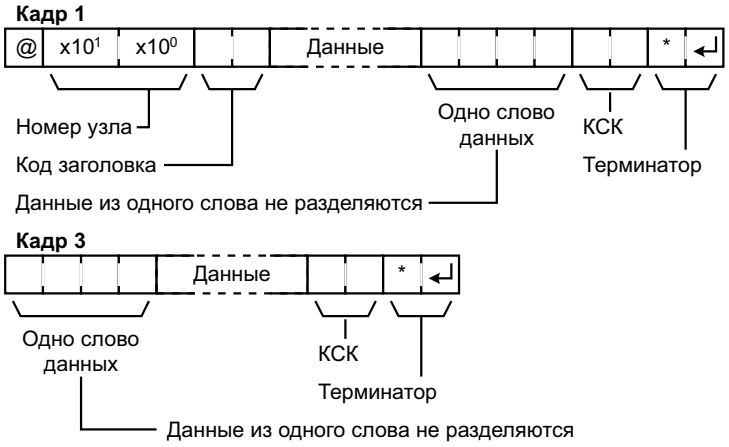
Разделенные ответы (от ПК на управляющий компьютер)

Поскольку каждый кадр получает управляющий компьютер, разделитель передается с ПК. После того, как будет передан разделитель, ПК посылает следующий кадр. Данная процедура повторяется до передачи полного ответа.



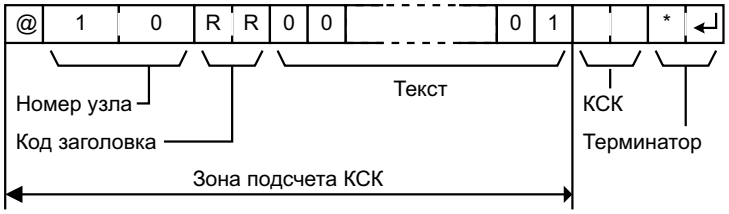
Предосторожности при передаче длинных блоков

При разделении команд таких, как WR, WL, WC или WD, которые выполняют операции записи, следите за тем, чтобы не разносить по разным кадрам данные, которые должны бить записаны в одно слово. Как показано на следующем рисунке, обязательно разделяйте кадры так, чтобы они совпадали с разделением слов.



Контрольная сумма кадра (КСК)

При передаче кадра КСК помещается непосредственно перед разделителем или терминатором для контроля, не возникло ли ошибки в данных. КСК - это 8-бит, преобразованных в два символа ASCII. Эти 8 бит являются результатом команды ИСКЛЮЧАЮЩЕЕ ИЛИ над данными от начала кадра до конца текста кадра (т.е. вплоть до КСК). Подсчет КСК каждый раз при приеме кадра и сравнения ее с переданной КСК позволяет контролировать ошибки в кадре.



Код ASCII			
@	40	0100	0000
			XOR
1	31	0011	0001
			XOR
0	30	0011	0000
			XOR
R	52	0101	0010
			...
1	31	0011	0001
		0100	0010
Результат вычислений		↓	↓
		4	2

Преобразованные в 16-ричный вид
Рассматриваются как символы ASCII

Пример программы для проверки КСК

В данном примере показана подпрограмма на языке BASIC для контроля КСК кадра, полученного управляющим компьютером.

```
400 *FCSCHECK
410 L=LEN (RESPONSE$) ‘ Переданные и принятые данные
420 Q=0: FCSCHCK$=“ “
430 A$=RIGHT$ (RESPONSE$,1)
440 PRINT RESPONSE$, A$, L
450 IF A$=“*” THEN LENGS=LEN(RESPONSE$)-3 ELSE LENGS=LEN(RESPONSE$)-
        2 ‘КСК принятых данных
```

```
460 FCSP$=MID$(RESPONSE$, LENGS+1, 2)
470 FOR I=1 TO LENGS      Число знаков в КСК
480 Q=ASC(MID$(RESPONSE$,I,1)) XOR Q
490 NEXT I
500 FCSD$=HEX$(Q)
510 IF LEN(FCSD$)=1 THEN FCSD$="0"+FCSD$
520 IF FCSD$  FCSP$ THEN FCSK$="ERR"
530 PRINT 'FCSD$=";FCSD$,"FCSP$=";FCSP$,"FCSK$=";FCSK$
540 RETURN
```

Замечание Нормально принятые данные включают КСК, разделитель или терминатор и т. д. При появлении ошибки при передаче КСК или другие данные могут быть пропущены. Предусмотрите в программе и такую ситуацию.

В данном примере код CR (возврат каретки) CHR$(13) не внесен в RESPONSES$. При включении кода CR измените строки 430 и 450.

6.2.2 Команды от ПК (только CQM1)

В режиме связи HOST LINK команды, как правило, посылаются с управляющего компьютера, но также возможно посылать команды с ПК на управляющий компьютер.

В режиме связи HOST LINK любые данные можно передать с ПК на управляющий компьютер. Для посылки команд на управляющий компьютер используйте команду TXD(48) (послать) в программе ПК в режиме связи HOST LINK

TXD(48) выдает данные с указанного порта (RS-232C или периферийного). Подробности об использовании TXD(48) см. 5-27-2.

Формат приема

При выполнении TXD(48) данные, находящиеся в словах, подлежащих пересылке, начинающихся с первого слова, преобразуются в ASCII и выдаются на управляющий компьютер в виде команд в формате, приведенном далее. Символ @, номер узла, код заголовка, КСК и разделитель добавляются автоматически при передаче. В управляющем компьютере предварительно необходимо подготовить программу для интерпретации и обработки данного формата.

Один байт данных (две 16-ричные цифры) преобразуются в 2 символа ASCII для передачи, количество объектов передачи в два раза больше числа слов, заданных для TDX(48). Максимальное число символов для передачи - 122 и максимальное число байт, которые можно задать для TDX(48) - в два раза меньше, 61.

6.2.3 Коды окончания ответа

Таблица с кодами окончания ответа, которые можно послать в управляющий компьютер, находится в 8-7. Код ответа 00 указывает нормальное завершение команды.

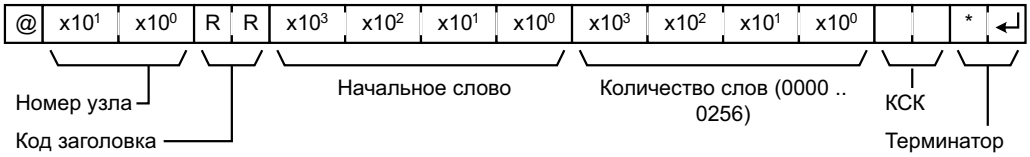
6.3 Команды Host Link (связи с верхним уровнем)

В данной главе описаны команды, которые можно выдавать с управляющего компьютера на ПК.

6.3.1 Чтение области IR/SR - RR

Читает содержимое заданного количества слов IR и SR, начиная от указанного слова.

Формат команды



Замечание Начальные слова: 0000..0255 в CQM1, 0000 ....0019 и 0200..0255 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

Замечание 1. Слова 0020....0199 в CPM1 нельзя задать. При попытке прочесть эти слова будут считываться нули.

2. При чтении более 30 слов ответ будет разделен.

Параметры

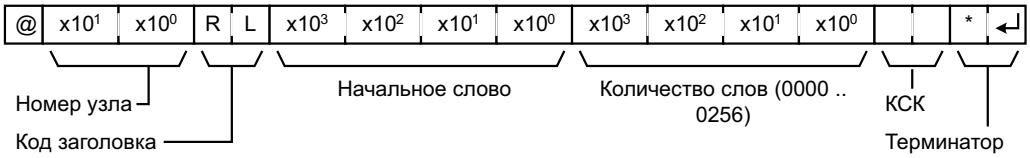
Читаемые данные (ответ)

Содержание нескольких слов, заданных в команде, возвращается в 16-ричном виде в виде ответа. Слова возвращаются последовательно, начиная с указанного начального слова.

6.3.2 Чтение области LR - RL

Читает содержимое заданного количества слов LR, начиная от заданного слова.

Формат команды



Замечание 1. Начальные слова: 0000..0063 в CQM1, 0000 ....0015 в CPM1.

2. Количество слов: 0001..0064 в CQM1, 0001 ....0016 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

Параметры

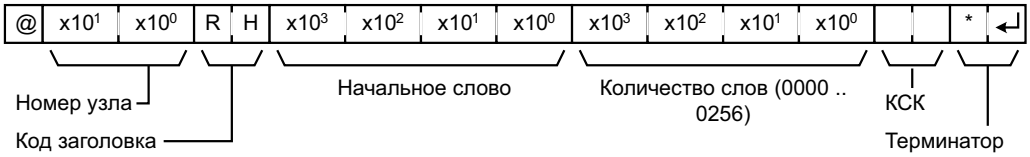
Данные для чтения (ответ)

Содержание нескольких слов, заданных в команде, возвращается в 16-ричном виде в качестве ответа. Слова возвращаются последовательно, начиная с заданного начального слова.

6.3.3 Чтение области HR - RH

Читает содержимое заданного количества слов HR, начиная от заданного слова.

Формат команды



Замечание 1. Начальные слова: 0000..0099 в CQM1, 0000 ....0019 в CPM1.

2. Количество слов: 0001..0100 в CQM1, 0001 ....0020 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

Параметры

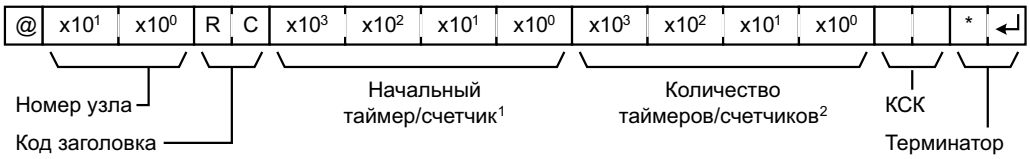
Данные для чтения (ответ)

Содержание нескольких слов, заданных в команде, возвращается в 16-ричном виде в качестве ответа. Слова возвращаются последовательно, начиная с заданного начального слова.

6.3.4 Чтение текущего значения - RC

Читает содержимое заданного количества текущих значений Т/Сч, начиная от заданного таймера/счетчика.

Формат команды



Замечание 1. Начальный Т/Сч: 0000..0511 в CQM1, 0000 ....0127 в CPM1.

2. Количество Т/Сч: 0001..0512 в CQM1, 0001 ....0128 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

При чтении более 30 слов ответ будет разделен.

Параметры

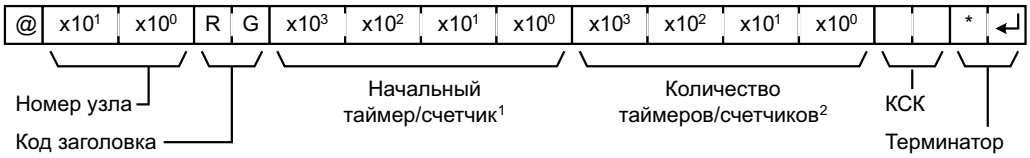
Данные для чтения (ответ)

Несколько текущих значений, заданных в команде, возвращается в 16-ричном виде в качестве ответа. Текущие значения возвращаются последовательно, начиная с заданного начального таймера/счетчика.

6.3.5 Чтение состояния TC - RG

Читает состояние флага завершения заданного количества таймеров/счетчиков, начиная от заданного таймера/счетчика.

Формат команды



Замечание 1. Начальный Т/Сч: 0000..0511 в CQM1, 0000 ....0127 в CPM1.

2. Количество Т/Сч: 0001..0512 в CQM1, 0001 ....0128 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

При чтении состояния более 123 таймеров/счетчиков ответ будет разделен.

Параметры

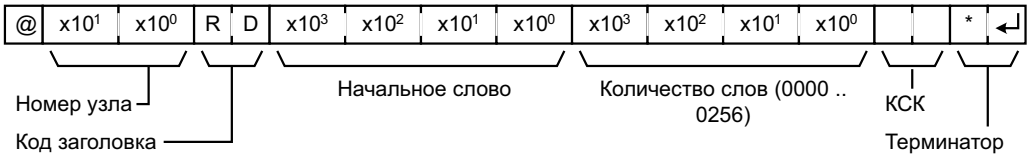
Данные для чтения (ответ)

Состояние нескольких флагов завершения, заданных в команде, возвращается в 16-ричном виде в виде ответа. “1” указывает, что Флаг завершения = 1.

6.3.6 Чтение области DM - RD

Читает содержимое заданного количества слов DM, начиная от заданного слова.

Формат команды



Замечание Начальные слова: 0000..6655 в CQM1, 0000 ....0123 и 6144...6655 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

Замечание 1. Слова 1024....6143 в CPM1 нельзя задать. При попытке прочесть эти слова будут считываться нули.

2. При чтении более 30 слов ответ будет разделен.

Параметры

Данные для чтения (ответ)

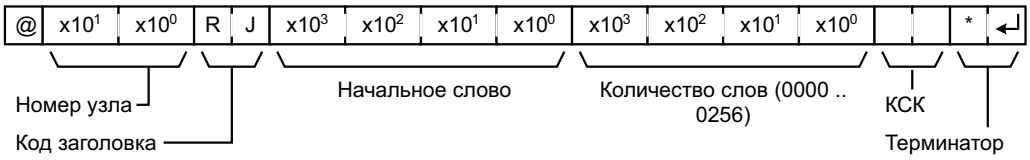
Содержание нескольких слов, заданных в команде, возвращается в 16-ричном виде в виде ответа. Слова возвращаются последовательно, начиная с заданного начального слова.

Замечание Будьте осторожны с конфигурацией области DM, поскольку она меняется в зависимости от модели ЦУ.

6.3.7 Чтение области AR - RJ

Читает содержимое заданного количества слов AR, начиная от заданного слова.

Формат команды



Замечание 1. Начальное слово: 0000..0027 в CQM1, 0000 ....0015 в CPM1.

2. Количество слов: 0001..0028 в CQM1, 0000 ....0016 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

Параметры

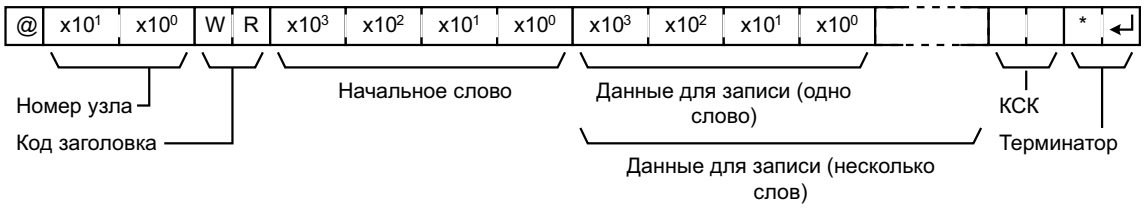
Данные для чтения (ответ)

Содержание нескольких слов, заданных в команде, возвращается в 16-ричном виде в виде ответа. Слова возвращаются последовательно, начиная с заданного начального слова.

6.3.8 Запись в области IR/SR - WR

Записывает данные в области IR и SR, начиная от указанного слова. Запись идет слово за словом.

Формат команды

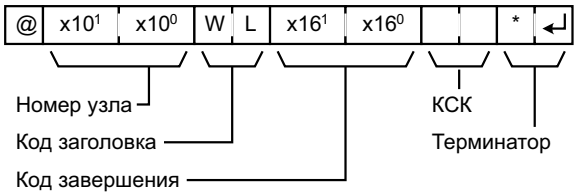


Замечание 1. Начальное слово: 0000....0252 в CQM1, 0000....0019 и 0200....0252 в CPM1.

2. При записи более 30 слов разделите команд.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Замечание Слова 0020....0199 в CPM1 нельзя задать. При попытке записать эти слова операция не выполнится, но будет нормальное завершение.

Параметры

Данные для записи (команда)

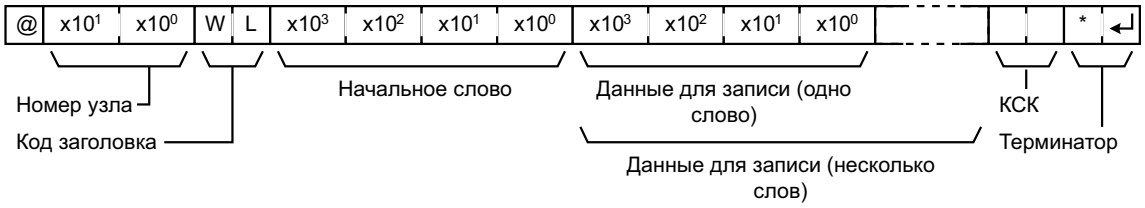
Задайте несколько слов для записи в области IR или SR в 16-ричном виде, начиная с заданного начального слова.

Замечание Если данные для записи выходят за границу допустимой зоны, генерируется ошибка и операция записи не исполняется. Если, например, в качестве начального слова для записи задано 252, и заданы для записи два слова, последним адресом для записи будет слово 253, и команда не выполнится, поскольку SR 253 находится вне допустимой зоны для записи.

6.3.9 Запись в область LR - WL

Записывает данные в область LR, начиная от указанного слова. Запись идет слово за словом.

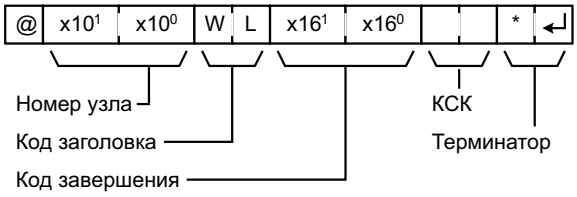
Формат команды



Замечание Начальное слово: 0000..0063 в CQM1, 0000 ....0015 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

Данные для записи (команда)

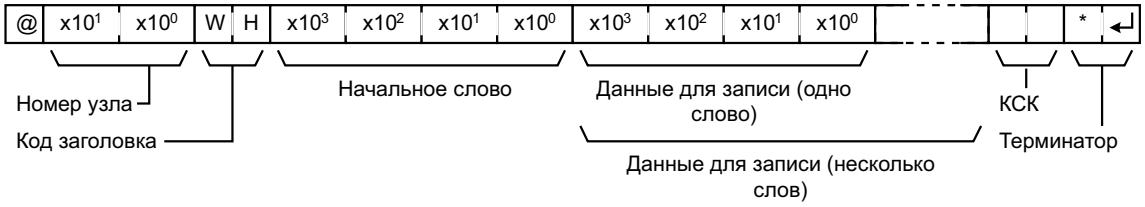
Задайте несколько слов для записи в область LR в 16-ричном виде, начиная с заданного начального слова.

Замечание Если данные для записи выходят за границу допустимой зоны, генерируется ошибка и операция записи не выполняется. Если, например, в качестве начального слова для записи задано 60, и заданы для записи пять слов, последним адресом для записи будет слово 64, и команда не выполнится, поскольку LR 64 находится вне допустимой зоны.

6.3.10 Запись в область HR - WH

Записывает данные в область HR, начиная от указанного слова. Запись идет слово за словом.

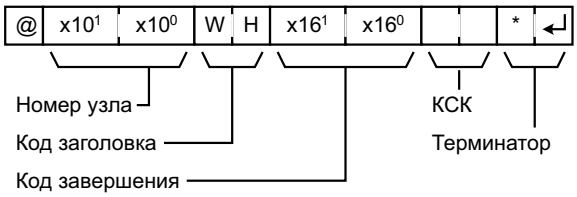
Формат команды



Замечание Начальное слово: 0000..0063 в CQM1, 0000 ....0019 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

Данные для записи (команда)

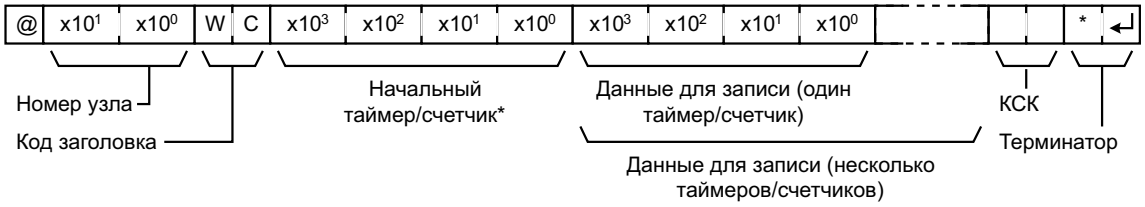
Задайте несколько слов для записи в область HR в 16-ричном виде, начиная с заданного начального слова.

Замечание Если данные для записи выходят за границу допустимой зоны, генерируется ошибка и операция записи не выполняется. Если, например, в качестве начального слова для записи задано 98, и для записи заданы три слова, последним адресом для записи будет слово 100, и команда не выполнится, поскольку HR 100 находится вне допустимой зоны.

6.3.11 Запись текущего значения - WC

Записывает текущие значения таймеров/счетчиков, начиная от указанного таймера/счетчика.

Формат команды

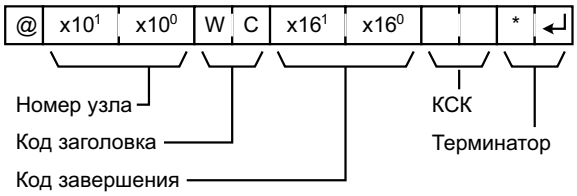


Замечание 1. Начальный таймер/счетчик: 0000..0511 в CQM1, 0000 ....01127 в CPM1.

2. При записи более 29 слов разделите команду.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

Данные для записи (команда)

Задайте в десятичных цифрах (двоично-десятичном виде) текущие значения для нескольких таймеров/счетчиков, начиная с начального таймера/счетчика.

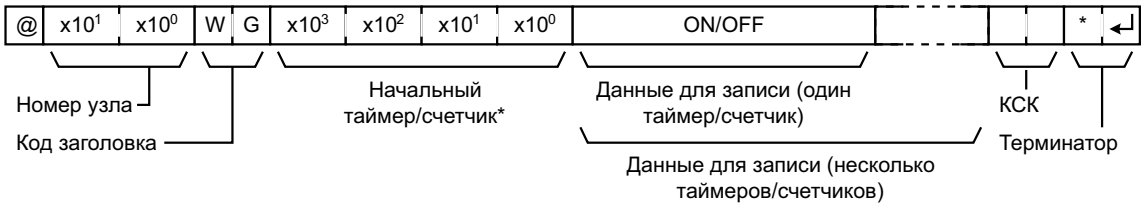
Замечание 1. Когда команда служит для записи данных в область текущих значений, флаги завершения для таймеров/счетчиков, которые пишутся, будут в состоянии 0.

2. Если данные для записи выходят за границу допустимой зоны, генерируется ошибка и операция записи не выполняется. Если, например, в качестве начального слова для записи задано 510, и для записи заданы три слова, последним адресом для записи будет слово 512, и команда не выполнится, поскольку TC 512 находится вне допустимой зоны.

6.3.12 Запись состояния T/Сч - WG

Записывает состояние флагов завершения таймеров и счетчиков, начиная от заданного таймера/счетчика. Запись производится последовательно.

Формат команды



Замечание 1. Начальный таймер/счетчик: 0000..0511 в CQM1, 0000 ....01127 в CPM1.

2. При записи состояния более 118 таймеров/счетчиков разделите команду.

Формат ответа

Код окончания 00 указывает на нормальное завершение.

Параметры

Данные для записи (команда)

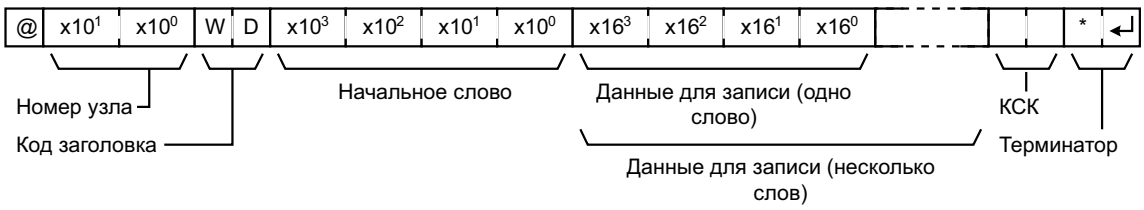
Задайте состояние флагов завершения для нескольких таймеров/счетчиков, подлежащих записи, начиная с начального слова, 1 или 0. Когда флаг завершения = 1, это указывает, что таймер или счетчик отработали задание.

Замечание Если данные для записи выходят за границу допустимой зоны, генерируется ошибка и операция записи не выполняется. Если, например, в качестве начального слова для записи задано 510, и для записи заданы три слова, последним адресом для записи будет слово 512, и команда не выполнится, поскольку TC 512 находится вне допустимой зоны.

6.3.13 Запись в область DM - WD

Записывает данные в область DM, начиная от указанного слова. Запись идет слово за словом.

Формат команды

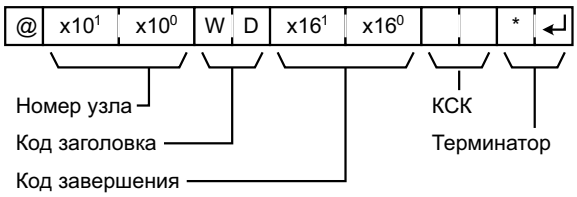


Замечание 1. Начальное слово: 0000..6143 в CQM1, 0000 ....1023 в CPM1.

2. При записи более 29 слов разделите команду.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Замечание Слова 1024....6143 в CPM1 нельзя задавать. При попытке записать эти слова операция записи не выполнится, но будет нормальное завершение.

Параметры

Данные для записи (команда)

Задайте несколько слов для записи в область DR в 16-ричном виде, начиная с заданного начального слова.

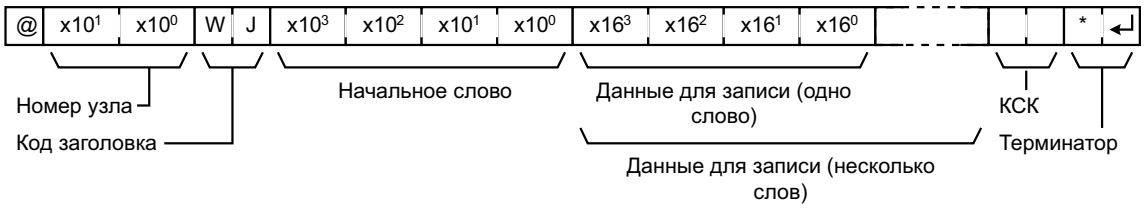
Замечание 1. Если данные для записи выходят за границу допустимой зоны, генерируется ошибка и операция записи не выполняется. Если, например, в качестве начального слова для записи задано 6142, и для записи заданы три слова, последним адресом для записи будет слово 6144, и команда не выполнится, поскольку DM 6144 находится вне допустимой зоны.

2. Будьте внимательны с конфигурацией зоны DM, поскольку она варьирует в зависимости от модели ЦУ.

6.3.14 Запись в область AR - WJ

Записывает данные в область AR, начиная от указанного слова. Запись идет слово за словом.

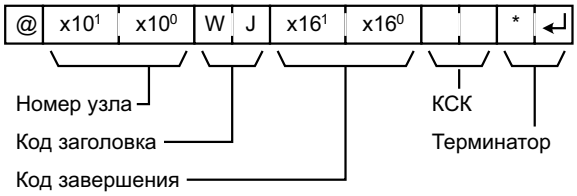
Формат команды



Замечание Начальное слово: 0000..0027 в CQM1, 0000 ....0015 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

Данные для записи (команда)

Задайте несколько слов для записи в область AR в 16-ричном виде, начиная с заданного начального слова.

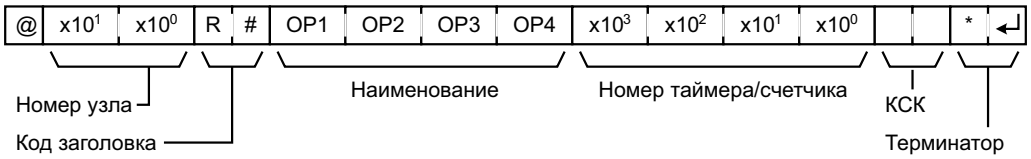
Замечание Если данные для записи выходят за границу допустимой зоны, генерируется ошибка и операция записи не выполняется. Если, например, в качестве начального слова для записи задано 26, и для записи заданы три слова,

последним адресом для записи будет слово 28, и команда не выполнится, поскольку AR 28 находится вне допустимой зоны.

6.3.15 Чтение 1 заданного значения - R#

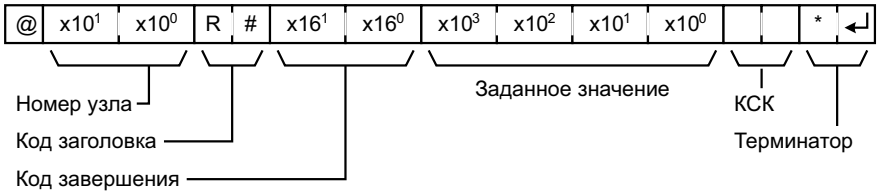
Ищет первую по порядку команду с заданным номером (из ряда TIM, TIMH(15), CNT и CNTR(12)) в программе пользователя и читает заданное значение, предполагая их константами. Считываемое задание - 4-хразрядное десятичное число. Поиск начинается с начала программы, поэтому для выдачи ответа может понадобиться до 10 с.

Формат команды



Замечание Номера Т/Сч: 0000..0511 в CQM1, 0000 ....0127 в CPM1.

Формат ответа



Параметры

Наименование, Номер Т/Сч, (команда)

Задайте команду для чтения заданного значения в ячейках “Название”. Обязательно задавайте 4 знака. В “Номер Т/Сч” задавайте номер таймера/счетчика.

Наименование команды				**Что задает**
OP1	**OP2**	**OP3**	**OP4**	
T	I	M	Пробел	Таймер
T	I	M	H	Высокоскоростной таймер
C	N	T	Пробел	Счетчик
C	N	T	R	Реверсивный счетчик

Заданное значение (ответ)

Возвращается константа “текущее значение”.

Замечание 1. Команда, заданная в ячейках “Название” должны состоять из 4 знаков.

2. Если одна и та же команда используется в программе больше одного раза, будет считано только первое значение.

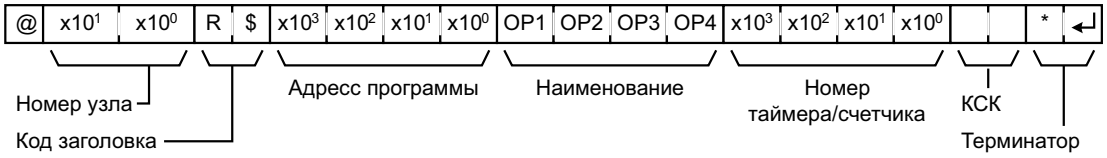
3. Используйте данную команду только когда Вы уверены, что заданное значение задано константой.

4. Код окончания ответа будет указывать на ошибку (16), если заданное значение задано не как константа.

6.3.16 Чтение 2 заданного значения - R$

Читает константу заданного значения или адрес слова, где хранится заданное значение. Считываемое задание - это 4-хразрядное десятичное число, записанное как операнд команд TIM, TIMH(15), CNT или CNTR(12) в заданном адресе программы пользователя. Данной командой можно задать адрес только программы объемом менее 10 К.

Формат команды



Замечание Номера Т/Сч: 0000..0511 в CQM1, 0000 ....0127 в CPM1.

Формат ответа

Параметры

Наименование, Номер Т/Сч, (команда)

Задайте наименование команды для чтения заданного значения в ячейках “Название”. Обязательно задавайте 4 знака. В “Номер Т/Сч” задавайте номер таймера/счетчика.

Наименование команды				**Что задает**
OP1	**OP2**	**OP3**	**OP4**	
T	I	M	Пробел	Таймер
T	I	M	H	Высокоскоростной таймер
C	N	T	Пробел	Счетчик
C	N	T	R	Реверсивный счетчик

Операнд, Заданное значение (ответ)

Имя, указывающее на тип заданного значения, возвращается в ячейках “Операнд”, а в ячейках “Задание” возвращается либо адрес заданного значения, либо константа.

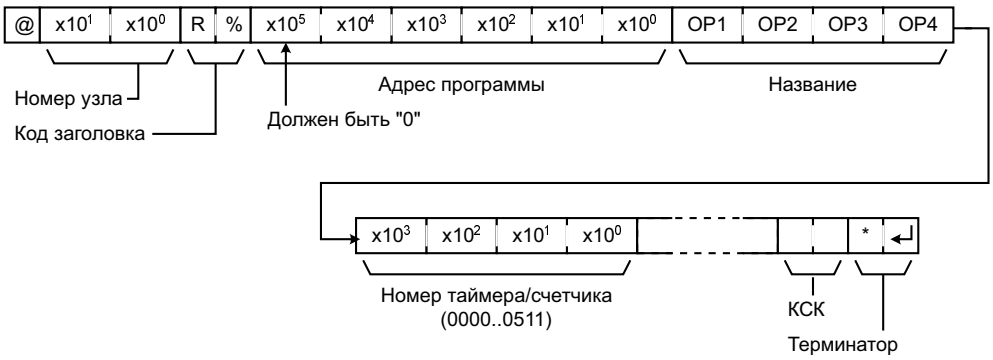
Операнд				**Что задает**	**Константа или адрес слова**	
OP1	**OP2**	**OP3**	**OP4**		**CQM1**	**CPM1**
C	I	O	Пробел	IR или SR	0000..0255	0000..0019 0200..0255
L	R	Пробел	Пробел	LR	0000..0063	0000..0015
H	R	Пробел	Пробел	HR	0000..0099	0000..0019
A	R	Пробел	Пробел	AR	0000..0027	0000..0015
D	M	Пробел	Пробел	DM	0000..6655	0000..6655
D	M	*	Пробел	DM (косвенно)	0000..6655	0000..6655
C	O	N	Пробел	Константа	0000..9999	0000..9999

Замечание Команда, заданная в ячейках “Название”, должны состоять из 4 знаков. Чтобы били заполнены 4 знака, заполняйте пустые места пробелами.

6.3.17 Чтение 3 заданного значения - R% (Только CQM1)

Читает константу заданного значения или адрес слова, в котором хранится заданное значение. Считываемое задание - это 4-хразрядное десятичное число, записанное как второе слово команд TIM, TIMH(15), CNT или CNTR(12) в заданном адресе программы пользователя. Данной командой можно задавать адрес для программ объемом 10 К и более.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение.

Параметры

Наименование, Номер Т/Сч, (команда)

Задайте наименование команды для чтения заданного значения в ячейках “Название”. Обязательно задавайте 4 знака. В “Номер Т/Сч” задавайте номер таймера/счетчика.

Наименование команды				**Что задает**	**Диапазон**
OP1	**OP2**	**OP3**	**OP4**		
T	I	M	Пробел	Таймер	0000..0511
T	I	M	H	Высокоскоростной таймер	
C	N	T	Пробел	Счетчик	
C	N	T	R	Реверсивный счетчик	

Операнд, Заданное значение (ответ)

Имя, указывающее на тип заданного значения, возвращается в ячейках “Операнд”, а в ячейках “Задание” возвращается либо адрес заданного значения, либо константа.

Операнд				**Что задает**	**Константа или адрес слова**
OP1	**OP2**	**OP3**	**OP4**		
C	I	O	Пробел	IR или SR	0000..0255
L	R	Пробел	Пробел	LR	0000..0063
H	R	Пробел	Пробел	HR	0000..0099
A	R	Пробел	Пробел	AR	0000..0027
D	M	Пробел	Пробел	DM	0000..6655
D	M	*	Пробел	DM (косвенно)	0000..6655
C	O	N	Пробел	Константа	0000..9999

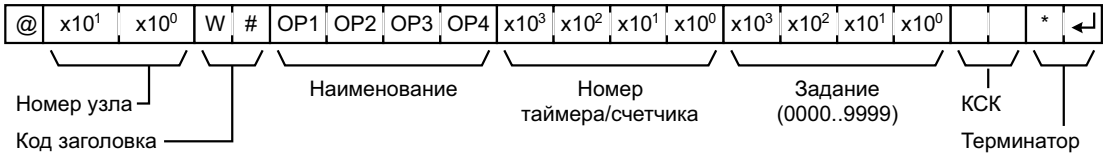
Замечание Команда, заданная в ячейках “Название”, должны состоять из 4 знаков. Чтобы били заполнены 4 знака, заполняйте пустые места пробелами.

6.3.18 Изменение заданного значения (первое)- W#

Ищет первую по порядку команду с заданным номером (из ряда TIM, TIMH(15), CNT и CNTR(12)) в программе пользователя и изменяет задание на новую константу,

указанную во втором слове этой команды. Поиск начинается с начала программы, поэтому для получения ответа может понадобиться до 10 с.

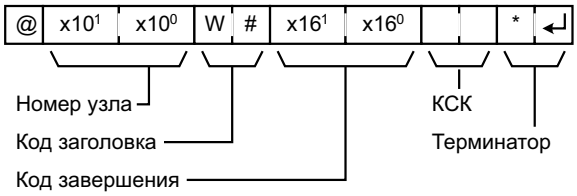
Формат команды



Замечание Номер Т/Сч: 0000..0511 в CQM1, 0000 ....0127 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

Наименование, Номер Т/Сч, (команда)

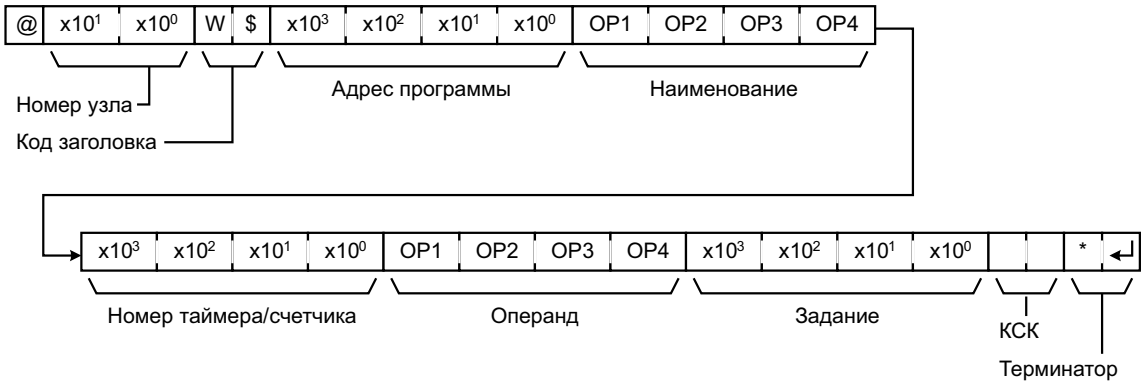
Задайте команду для чтения заданного значения в ячейках “Название”. Обязательно задавайте 4 знака. В “Номер Т/Сч” задавайте номер таймера/счетчика.

Наименование команды				**Что задает**
OP1	**OP2**	**OP3**	**OP4**	
T	I	M	Пробел	Таймер
T	I	M	H	Высокоскоростной таймер
C	N	T	Пробел	Счетчик
C	N	T	R	Реверсивный счетчик

6.3.19 Изменение 2 заданного значения - W$

Изменяет содержание второго слова команды TIM, TIMH(15), CNT или CNTR(12) в заданном адресе программы пользователя. Данной командой можно задать адрес только программы объемом менее 10 К.

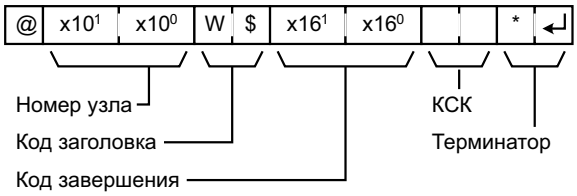
Формат команды



Замечание Номера Т/Сч: 0000..0511 в CQM1, 0000 ....0127 в CPM1.

Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

Наименование, Номер Т/Сч, (команда)

В ячейках “Название” задайте наименование команды (4 символа) для изменения заданного значения. В ячейках “Номер Т/Сч” задайте номер таймера/счетчика.

Наименование команды				**Что задает**
OP1	**OP2**	**OP3**	**OP4**	
T	I	M	Пробел	Таймер
T	I	M	H	Высокоскоростной таймер
C	N	T	Пробел	Счетчик
C	N	T	R	Реверсивный счетчик

Операнд, Заданное значение (ответ)

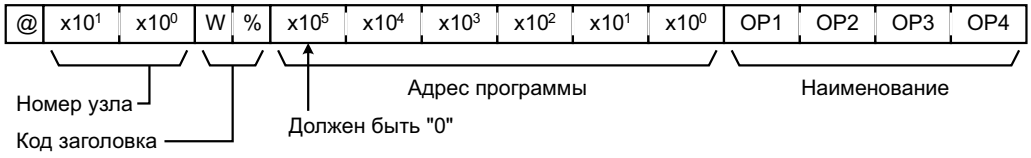
В ячейках “Операнд” задайте имя, указывающее на тип заданного значения. Задавайте имя четырьмя символами. В ячейках “Задание” укажите либо адрес слова, в котором содержится задание, либо константу.

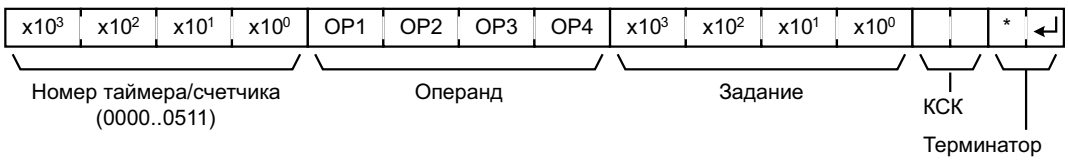
Операнд				**Что задает**	**Константа или адрес слова**	
OP1	**OP2**	**OP3**	**OP4**		**CQM1**	**CPM1**
C	I	O	Пробел	IR или SR	0000..0255	0000..0019 0200..0255
L	R	Пробел	Пробел	LR	0000..0063	0000..0015
H	R	Пробел	Пробел	HR	0000..0099	0000..0019
A	R	Пробел	Пробел	AR	0000..0027	0000..0015
D	M	Пробел	Пробел	DM	0000..6655	0000..10236144..6655
D	M		Пробел	DM (косвенно)	0000..6655	0000..10236144..6655
C	O		Пробел	Константа	0000..9999	0000..9999

6.3.20 Изменение 3 заданного значения - W%

Изменяет содержание второго слова команды TIM, TIMH(15), CNT или CNTR(12) в заданном адресе программы пользователя. Данной командой можно задать адрес программы объемом свыше 10 К.

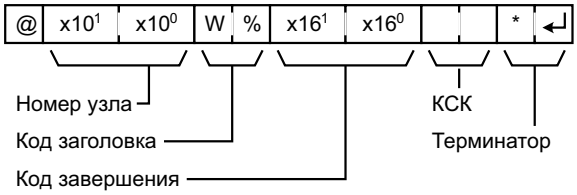
Формат команды





Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

Наименование, Номер Т/Сч, (команд)

В ячейках “Наименование” задайте наименование команды (4 символа), в которой будет изменяться заданное значения. В ячейках “Номер Т/Сч” задайте номер таймера/счетчика.

Наименование команды				**Что задает**	**Диапазон**
OP1	**OP2**	**OP3**	**OP4**		
T	I	M	Пробел	Таймер	0000..0511
T	I	M	H	Высокоскоростной таймер	
C	N	T	Пробел	Счетчик	
C	N	T	R	Реверсивный счетчик	

Операнд, Заданное значение (ответ)

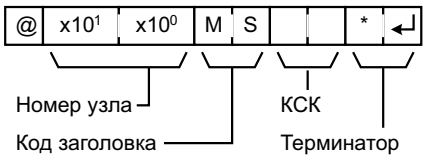
В ячейках “Операнд” задайте имя, указывающее на тип заданного значения. Задавайте имя четырьмя символами. В ячейках “Задание” укажите либо адрес слова, в котором содержится задание, либо константу.

Операнд				**Что задает**	**Константа или адрес слова**
OP1	**OP2**	**OP3**	**OP4**		
C	I	O	Пробел	IR или SR	0000..0255
L	R	Пробел	Пробел	LR	0000..0063
H	R	Пробел	Пробел	HR	0000..0099
A	R	Пробел	Пробел	AR	0000..0027
D	M	Пробел	Пробел	DM	0000..6655
D	M	*	Пробел	DM (косвенно)	0000..6655
C	O	N	Пробел	Константа	0000..9999

6.3.21 Чтение состояния - MS

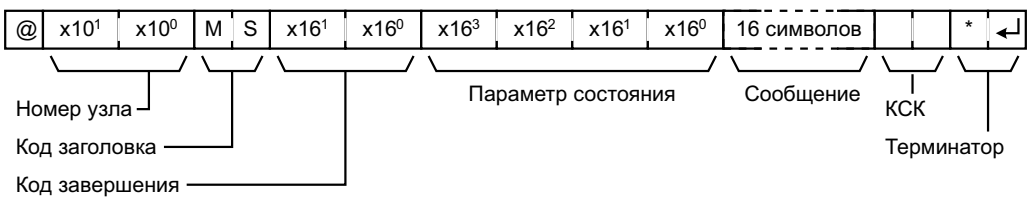
Читает состояние условий работы.

Формат команды



Формат ответа

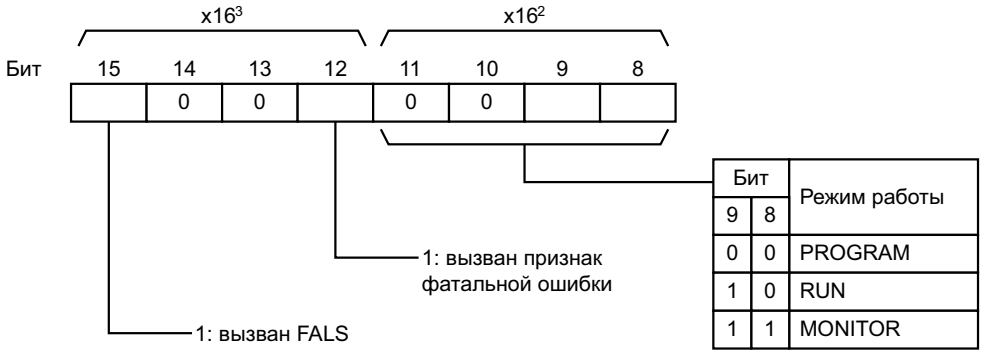
Код окончания 00 указывает на нормальное завершение.



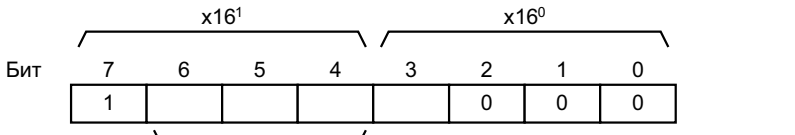
Параметры

Параметры состояния, сообщение (ответ)

“Параметр статуса” состоит из 4 цифр (двух байт) 16-ричных. Старший байт указывает режим работы ЦУ, младший - размер области программы.



Бит		Режим работы
9	8	
0	0	PROGRAM
1	0	RUN
1	1	MONITOR



Бит			Область программ
9	8	7	
0	0	0	PROGRAM
0	0	1	RUN
0	1	0	MONITOR

Защита от записи программы
0: защищена отзаписи
1: не защищена от записи

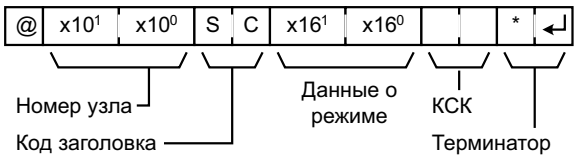
В CQM1 параметр “Сообщение” = номер FAL/FALS, который имеется при исполнении команды. Когда сообщений нет, параметр опускается.

В CPM1 параметр “Сообщение” - это 16-знаковое сообщение, которое имеется при исполнении команды. Когда сообщений нет, параметр опускается.

6.3.22 Запись состояния - SC

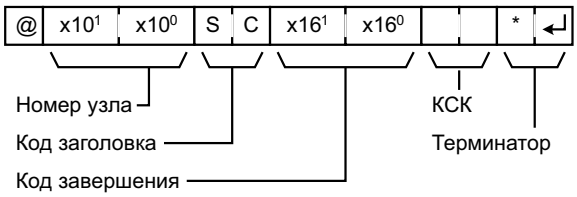
Изменяет рабочее состояние ПК.

Формат команды



Формат ответа

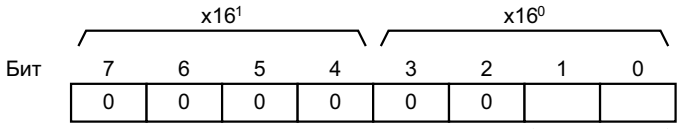
Код окончания 00 указывает на нормальное завершение.



Параметры

Параметры режима (команда)

“Параметр режима” состоит из 2 цифр (1 байт) 16-ричных. Два старших бита указывают режим работы ПК. Задайте все остальные биты в “0”.

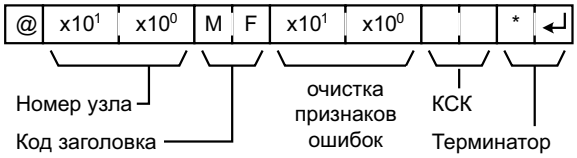


Бит		Режим работы
1	0	
0	0	PROGRAM
1	0	RUN
1	1	MONITOR

6.3.23 Чтение признаков ошибок - MF

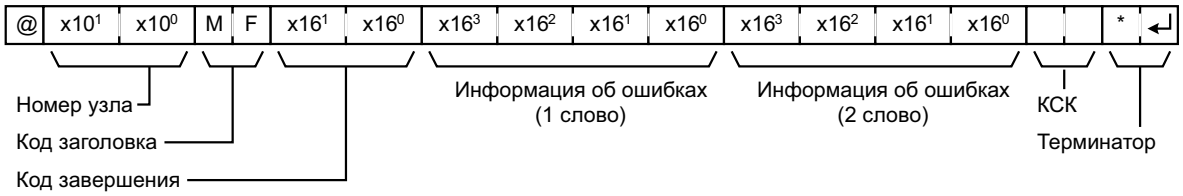
Читает и очищает признаки ошибок в ПК. Также проверяет, очищены ли признаки предыдущих ошибок.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение.



Параметры

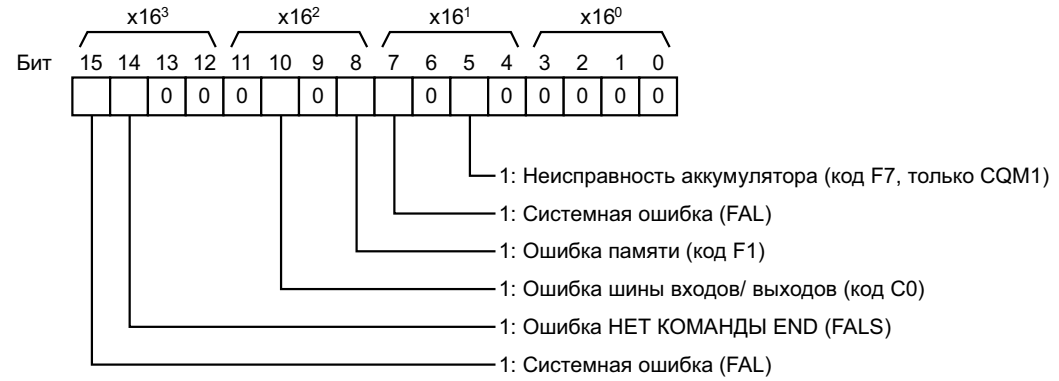
Очистка признаков ошибок (команда)

Задайте 01 для очистки признаков ошибок и 00 - не очищать (цифры двоично-десятичные). Признаки фатальных ошибок будут очищены только в режиме ПК PROGRAM.

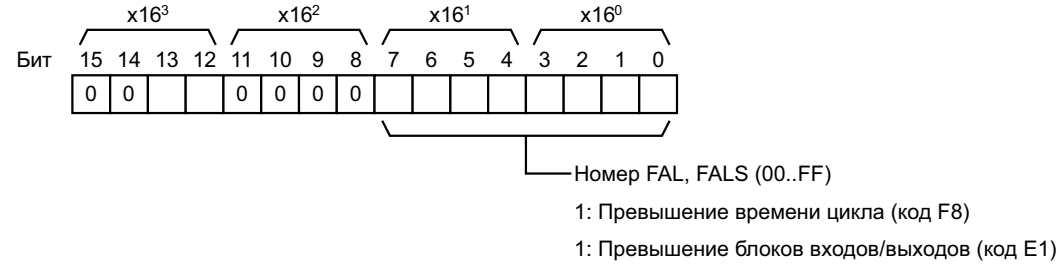
Информация об ошибках

Информация об ошибках приходит в двух словах.

1 слово



2 слово

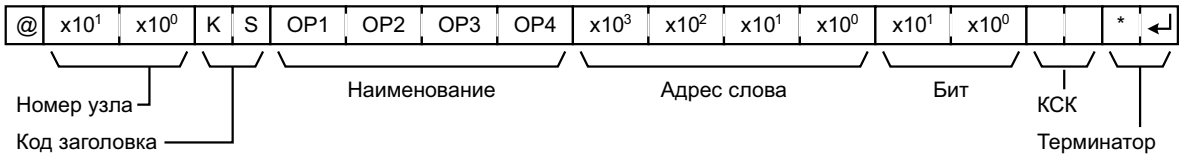


6.3.24 Принудительная установка - KS

Принудительно устанавливает бит в областях IR, SR, LR, HR, AR или TC. Одновременно может быть принудительно установлен только один бит.

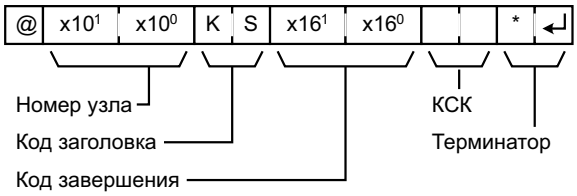
Когда бит принудительно установлен или сброшен, это состояние останется до команды ОТМЕНА ПРИНУДИТЕЛЬНОЙ УСТАНОВКИ/СБРОСА (KC) или до следующей команды ПРИНУДИТЕЛЬНАЯ УСТАНОВКА/СБРОС.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение



Параметры

Наименование, Адрес слова, Бит (команда)

В ячейках “Название” задайте область (т.е. IR, SR, LR, HR, AR или TC), в которой требуется принудительная установка. Обязательно задавайте 4 знака. В “Адресе слова” задавайте адрес слова, в “Бит” номер бита, который требует принудительной установки.

Операнд				Что задает	Константа или адрес слова		Бит
OP1	**OP2**	**OP3**	**OP4**		**CQM1**	**CPM1**	
C	I	O	Пробел	IR или SR	0000..0252	0000..0019 0200..0252	00..15 (десятичные)
L	R	Пробел	Пробел	LR	0000..0063	0000..0015	
H	R	Пробел	Пробел	HR	0000..0099	0000..0019	
A	R	Пробел	Пробел	AR	0000..0027	0000..0015	
T	I	M	Пробел	Флаг завершения (таймер)	0000..0511	0000..0127	Всегда 00
T	I	M	H	Флаг завершения (высокоскоростной таймер)			
C	N	T	Пробел	Флаг завершения (счетчик)			
C	N	T	R	Флаг завершения (реверсивный счетчик)			

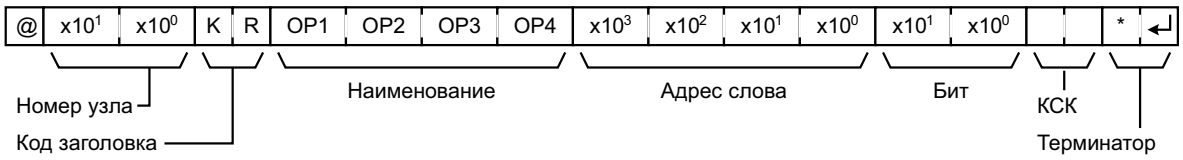
Замечание Команда, заданная в ячейках “Название”, должны состоять из 4 знаков. Чтобы били заполнены 4 знака, заполняйте пустые места пробелами.

6.3.25 Принудительный сброс - KR

Принудительно сбрасывает бит в областях IR, SR, LR, HR, AR или TC. Одновременно может быть принудительно сброшен только один бит.

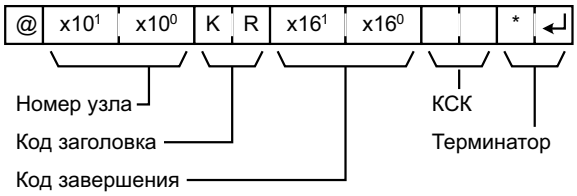
Когда бит принудительно установлен или сброшен, это состояние останется до команды ОТМЕНА ПРИНУДИТЕЛЬНОЙ УСТАНОВКИ/СБРОСА (КС) или до следующей команды ПРИНУДИТЕЛЬНАЯ УСТАНОВКА/СБРОС.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение



Параметры

Наименование, Адрес слова, Бит (Команда

В ячейках “Название” задайте область (т.е. IR, SR, LR, HR, AR или TC), в которой требуется принудительный сброс. Обязательно задавайте 4 знака. В “Адресе слова” задавайте адрес слова, в “Бит” номер бита, который требует принудительного сброса.

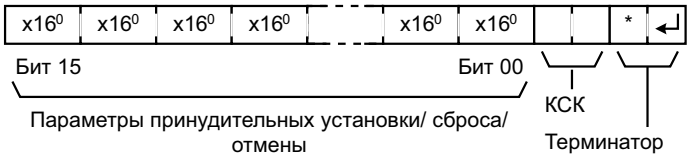
Операнд				Что задает	Константа или адрес слова		Бит
OP1	**OP2**	**OP3**	**OP4**		**CQM1**	**CPM1**	
C	I	O	Пробел	IR или SR	0000..0252	0000..0019 0200..0252	00..15 (десятичные)
L	R	Пробел	Пробел	LR	0000..0063	0000..0015	
H	R	Пробел	Пробел	HR	0000..0099	0000..0019	
A	R	Пробел	Пробел	AR	0000..0027	0000..0015	
T	I	M	Пробел	Флаг завершения (таймер)	0000..0511	0000..0127	Всегда 00
T	I	M	H	Флаг завершения (высокоскоростной таймер)			
C	N	T	Пробел	Флаг завершения (счетчик)			
C	N	T	R	Флаг завершения (реверсивный счетчик)			

Замечание Область, заданная в ячейках “Название”, должна состоять из 4 знаков. Чтобы были заполнены 4 знака, заполняйте пустые места пробелами.

6.3.26 Принудительные сброс/установка нескольких адресов - FK

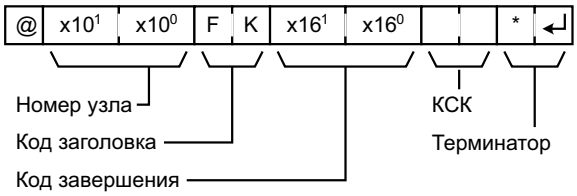
Принудительно сбрасывает, устанавливает или отменяет установку битов в областях IR, SR, LR, HR, AR или TC.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение



Параметры

Наименование, Адрес слова (Команда)

В ячейках “Наименование” задайте область (т.е. IR, SR, LR, HR, AR или TC), в которой требуется принудительные установка или сброс. Обязательно задавайте 4 знака. В “Адресе слова” задавайте адрес слова, которое требует принудительного сброса.

Операнд				Что задает	Константа или адрес слова	
OP1	**OP2**	**OP3**	**OP4**		**CQM1**	**CPM1**
C	I	O	Пробел	IR или SR	0000..0252	0000..0019 0200..0252
L	R	Пробел	Пробел	LR	0000..0063	0000..0015
H	R	Пробел	Пробел	HR	0000..0099	0000..0019
A	R	Пробел	Пробел	AR	0000..0027	0000..0015
T	I	M	Пробел	Флаг завершения (таймер)	0000..0511	0000..0127
T	I	M	H	Флаг завершения (высокоскоростной таймер)		
C	N	T	Пробел	Флаг завершения (счетчик)		
C	N	T	R	Флаг завершения (реверсивный счетчик)		

Параметры принудительных установки/сброса/отмены

Если задан флаг завершения таймера или счетчика, действует только бит 15, а все другие биты будут игнорироваться. Для таймеров/счетчиков возможны только принудительные установка или сброс.

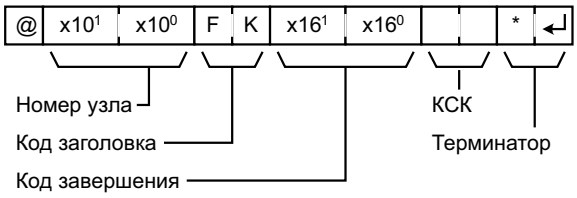
Если задан адрес слова, содержание слова задает требуемый процесс для каждого бита в заданном слове, как показано в таблице.

16-ричное задание	Процесс
0000	Нет действия (состояние бита не изменяется)
0002	Сброс
0003	Установка
0004	Принудительный сброс
0005	Принудительная установка
0008	Отмена принудительного состояния сброса/установки

Биты, которые установлены или сброшены не принудительно, могут быть изменены при выполнении программы, но биты, которые установлены/сброшены принудительно сохраняют принудительное состояние до очистки данного состояния.

Формат ответа

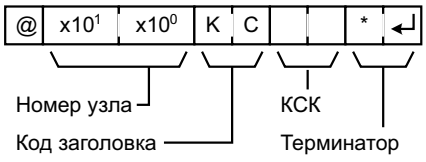
Код окончания 00 указывает на нормальное завершение.



6.3.27 Отмена принудительных установки/сброса - KC

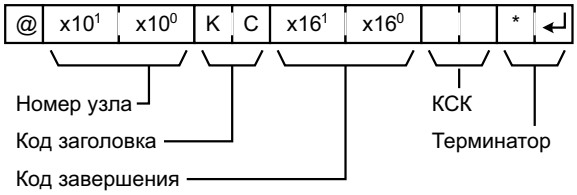
Отменяет все принудительно установленные или сброшенные биты (включая установленные командами ПРИНУДИТЕЛЬНАЯ УСТАНОВКА, ПРИНУДИТЕЛЬНЫЙ СБРОС, ПРИНУДИТЕЛЬНАЯ УСТАНОВКА/СБРОС НЕСКОЛЬКИХ БИТОВ). Если установлены несколько бит, принудительное состояние будет отменено для всех из них. Командой KC невозможно отменять состояние битов по отдельности.

Формат команды



Формат ответа

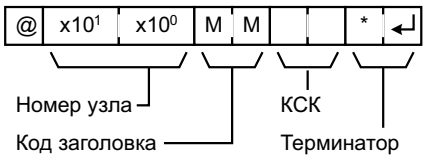
Код окончания 00 указывает на нормальное завершение



6.3.28 Читать модели ПК - MM

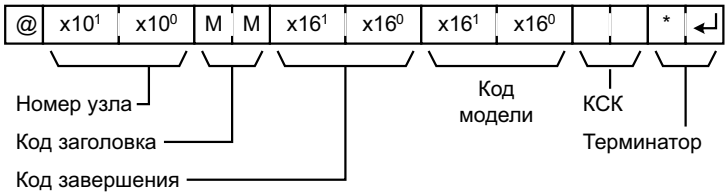
Читает модель ПК

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение



Параметры

Код модели

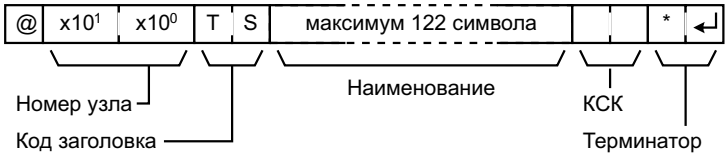
Указывает код модели ПК из двух 16-ричных чисел

Код модели	Модель
01	C250
02	C500
03	C120
0E	C2000
10	C1000H
11	C2000H/CQM1/CPM1
12	C20H/C28H/C40H/C200H/C200HS
20	CV500
21	CV1000
22	CV2000
40	CVM1-CPU01-EV2
41	CVM1-CPU11-EV2
42	CVM1-CPU21-EV2

5.3.29 Тест - TS

Возвращает, не изменяя, один блок данных, переданных с управляющего компьютера.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение

Параметры

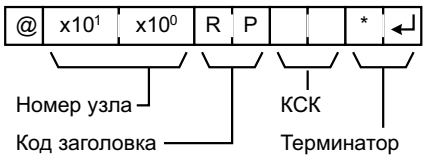
Символы (Команда, ответ)

Для команды задаются любые символы, кроме возврата каретки (CHR$(13)). Для ответа, если тест успешен, возвращаются те же символы, что и заданные в команде.

5.3.30 Чтение программы - RP

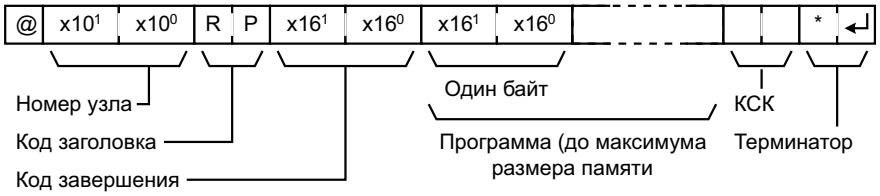
Читает содержимое области памяти программ ПК на машинном языке (в объектном коде). Содержимое читается как единый блок, с начала до конца.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение



Параметры

Программа (Ответ)

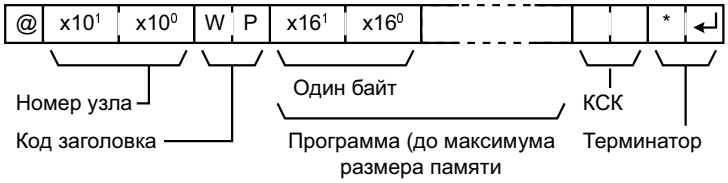
Программа читается из всей области программы.

Замечание Для остановки операции выполните команду ABORT (XZ).

6.3.31 Запись программы - WP

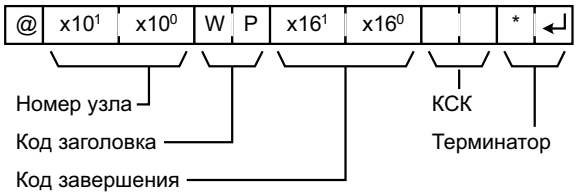
Записывает в область памяти программ ПК на машинном языке (в объектном коде) программу, переданную с управляющего компьютера. Содержимое пишется как единый блок, с начала до конца.

Формат команды



Формат ответа

Код окончания 00 указывает на нормальное завершение



Параметры

Программа (Команда)

Программа размером до максимального размера памяти.

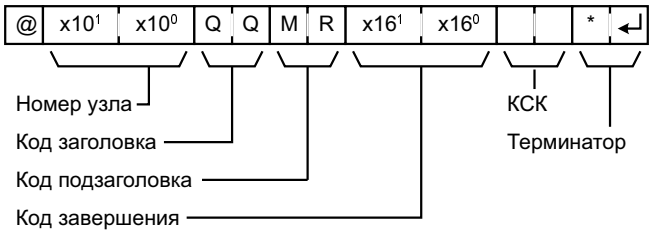
6.3.32 Составная команда - QQ

Регистрирует в ПК все биты, слова и таймеры/счетчики, которые будут считываться, и читает их состояние всех вместе.

Формат команды

Формат ответа

Код окончания 00 указывает на нормальное завершение



Параметры

Область чтения (Команда)

Задайте код из 4 знаков области, подлежащей чтению. Коды, которые можно задавать, перечислены в таблице.

Адрес слова для чтения, Формат данных (Команда)

В зависимости от области и типа данных, подлежащих чтению, считываемая информация может быть следующей. Данные для чтения заданы в 4 двоично-десятичных цифрах, формат данных задается в двух двоично-десятичных цифрах.

Области	Данные для чтения	Область для чтения	Слово для чтения		Формат данных
			CQM1	CPM1	
IR или SR	Бит	CIO (S)	0000..0255	0000..0019 0200..0255	00..15 (десятичный)
	Слово				“CH”
LR	Бит	LR (S) (S)	0000..0063	0000..0015	00..15 (десятичный)
	Слово				“CH”
HR	Бит	HR (S) (S)	0000..0099	0000..0019	00..15 (десятичный)
	Слово				“CH”
AR	Бит	AR (S) (S)	0000..0027	0000..0015	00..15 (десятичный)
	Слово				“CH”
Таймер	Флаг завершения	TIM (S)	0000..0511	0000..0127	Два символа, отличные от “CH”
	Текущее значение				“CH”
Высокоскоростной таймер	Флаг завершения	TIMH	0000..0511	0000..0127	Два символа, отличные от “CH”
	Текущее значение				“CH”
Счетчик	Флаг завершения	CNT (S)	0000..0511	0000..0127	Два символа, отличные от “CH”
					“CH”
Высокоскоростной счетчик	Флаг завершения	CNTR	0000..0511	0000..0127	Два символа, отличные от “CH”
	Текущее значение				“CH”
DM	Слово	DM (S) (S)	0000..6655	0000..1023 6144..6655	Любые два символа

(S) - пробел

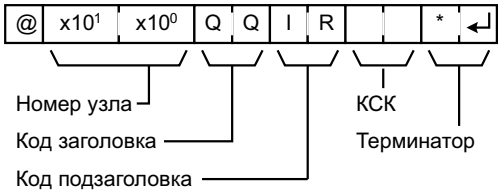
Разделитель данных (Команда)

Информация для чтения задается для одного блока за раз, блоки разделяются кодом (,). Можно задать максимальное количество блоков 128. (Когда задано текущее значение таймера/счетчика возвращается также состояние флага завершения, и ,следовательно, должно рассматриваться как 2 блока).

Чтение пакета

Бит, слово и состояние таймера/счетчика читается как пакет в соответствии с информацией, занесенной вместе с командой QQ.

Формат команды



Формат ответа

Параметры

Считываемые данные (ответ)

Считываемые данные возвращаются согласно формата данными порядку, в котором информация зарегистрировалась командой QQ.

Если задан “Флаг завершения”, возвращается бит (1 или 0);

Если задано “Слово”, возвращается слово данных;

Если для таймеров/счетчиков задано “Текущее значение”, возвращается текущее значение и флаг завершения.

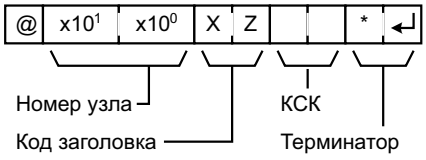
Разделитель данных (Ответ)

Код разделителя (,) при чтении возвращается между секциями.

6.3.33 Прервать - XZ

Прерывает идущую операцию связи HOST LINK, затем разрешает прием следующей команды. Команда XZ не получает ответа.

Формат команды



6.3.34 Инициировать - **

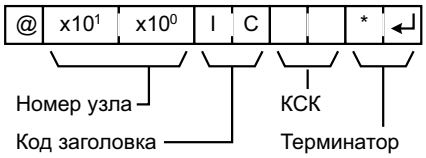
Инициирует передачу контрольной процедуры во все ПК, подключенные к управляющему компьютеру. Команда не применяет номер узла и КСК и не получает ответа.

Формат команды

6.3.35 Неопределенная команда - IC

Данный ответ возвращается, если код заголовка нельзя декодировать. Проверьте код заголовка.

Формат команды



7. Операции ПК и распределение времени

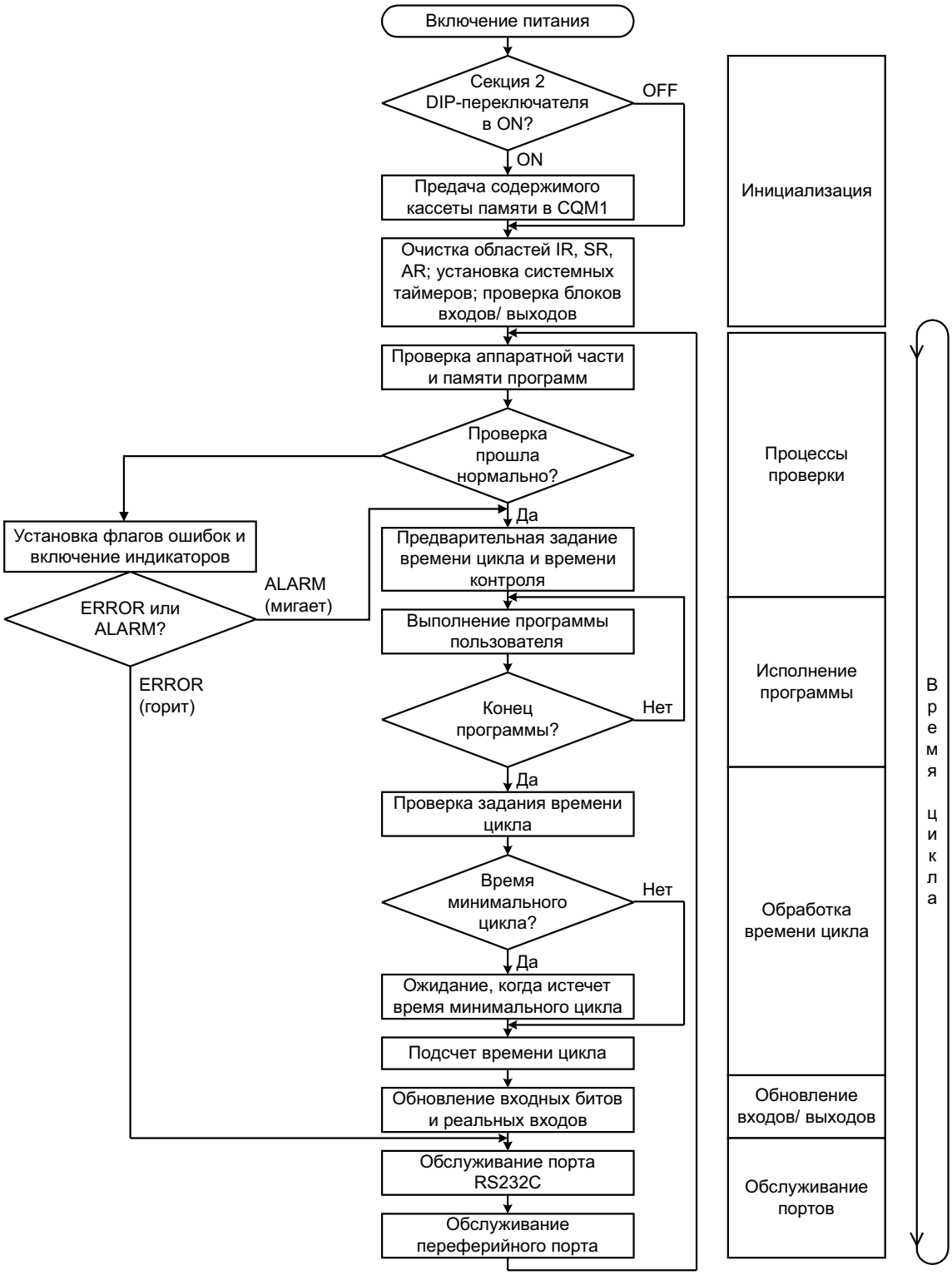
В данной главе объясняется внутренняя обработка CQM1/CPM1/CPM1A/SRM1 и время, требуемое для обработки. Смотрите данную главу для понимания точной синхронизации операций CQM1/CPM1/CPM1A/SRM1.

7.1 Время цикла CQM1 и время реакции на вход

7.1.1 Цикл CQM1

Алгоритм работы CQM1

На диаграмме показан общий алгоритм работы CQM1.



Один рабочий цикл ЦУ называется цикл. Время, требуемое для цикла, называется временем цикла.

Методы обновления входов/выходов

Операции обновления входов/выходов можно разделить на 2 типа.

- Обновление входов: чтение состояния 1 или 0 реальных входов в биты входов/выходов;
- Обновление выходов: запись состояния 1 или 0 после исполнения программы на вхоные точки.

Методы обновления входов/выходов показаны в следующей таблице.

Входы/выходы	Метод обновления	Функция
Вход	Циклическое обновление	Обновление осуществляется в отведенное время 1 раз за цикл.
	Обновление по входному прерыванию	Обновление осуществляется перед обработкой программы прерывания, если произошло какое-либо прерывание: входное прерывание, прерывание интервального таймера, прерывание высокоскоростного счетчика (Выполняется также циклическое обновление).
Выход	Циклическое обновление	Обновление осуществляется в отведенное время 1 раз за цикл.
	Прямое обновление	Реальный выход обновляется сразу же, когда есть выход от программы пользователя.

Исходное состояние обновления CQM1:

Входы: выполняется только циклическое обновление.

Выходы: выполняется только циклическое обновление.

Циклическое обновление должно выполняться и для входов и для выходов. Если обновление входов должно выполняться во время прерываний, задайте зону обновляемых входов в установочных параметрах DM 6630..DM 6638. Прекращение прямого обновления задается в DM 6639.

Кроме вышесказанных методов возможно также обновлять входы/выходы из программы командой IOFR(97).

7.1.2 Время цикла CQM1

В таблице приведены процессы, происходящие в одном цикле CQM1, и соответствующее время обработки.

Процесс	Назначение	Требуемое время
Общая диспетчеризация	Установка контрольного таймера, проверка шины входов/выходов, обновление часов, обновление бит, отведенных под новые функции и т.д.	0.8 мс (0.9 мс, если имеется кассета памяти с часами). Для CQM1-CPU4-E добавляется 0.1 мс.
Исполнение программы.	Исполнение программы пользователя.	Общее время исполнения команд. (Зависит от содержания программы).
Расчет времени цикла	Задержка, пока время минимального цикла занесется в DM 6619. Расчет времени цикла.	Почти мгновенно, за исключением времени задержки.
Обновление входов/выходов	Чтение информации с блоков входов в биты входа. Выдача информации (результата исполнения программы) на выходы блоков выхода.	Количество слов входа x 0.01 мс.
Обслуживание порта RS-232C	Обслуживается устройство, подключенное к порту RS-232C	5% или менее от времени цикла *
Обслуживание периферийного порта	Обслуживается устройство, подключенное к периферийному порту	5% или менее от времени цикла *

Замечание Процент задается в установочных параметрах DM 6616, DM 6617.

Время цикла и операции

Влияние времени цикла на операции показано в таблице.

Время цикла	Условия работы
10 мс и более	Может неточно исполняться TIMH(15), когда применяются TC 016..TC 511 (операции с TC 000..TC 015 будут исполняться нормально).[1]
20 мс и более	Может неточно исполняться программа, использующая бит часов 0.02 с (SR 25401).
100 мс и более	Может неточно исполняться программа, использующая бит часов 0.1 с (SR 25500). Генерируется признак ошибки ПРЕВЫШЕНИЕ ВРЕМЕНИ ЦИКЛА (включается в 1 SR 25309)[2]
120 мс и более	Превышено контрольное время FALS 9F. Генерируется признак ошибки системы (FALS 0F) и работа прекращается.[3]
200 мс и более	Может неточно исполняться программа, использующая бит часов 0.2 с (SR 25501).

Замечание 1. Количество таймеров, участвующих в обработке прерываний, можно задать в DM 6629. Значения по умолчанию - TC 000..TC 015.

2. Для запрещения ошибки ПРЕВЫШЕНИЕ ВРЕМЕНИ ЦИКЛА используется DM 6655.

3. Время контроля за циклом FALS 9F может изменяться параметром DM 6618.

Пример расчет времени цикла

В данном примере рассчитывается время цикла для CQM1 с 80 входами/выходами, которые распределяются следующим образом:

Входы постоянного тока: 48 (3 слова)

Выходы: 2 (2 слова)

Остальные условия работы:

Программа пользователя: 2000 команд (состоят из команд LD и OUT).

Часы: Нет

Порт RS-232C: Не используется

Время цикла: Переменное (минимум не задан)

Замечание Среднее время обработки одной команды программы пользователя принимется 0.625 мкс.

Распределение времени цикла показано в следующей таблице.

Процесс	Способ подсчета	Время с периферийным устройством	Время без периферийного устройства
Общая диспетчеризация	Фиксированное время	0.8 мс	0.8 мс
Исполнение программы.	0.625 x 2000 (мкс)	1.25 мс	1.25 мс
Расчет времени цикла	Пренебрегаем	0 мс	0 мс
Обновление входов/выходов	0.01 x 3 + 0.005 x2 (мкс)	0.04 мс	0.04 мс
Обслуживание порта RS-232C	Не используется	0 мс	0 мс
Обслуживание периферийного порта	Минимальное время	0.34 мс	0 мс
Время цикла	(1)+(2)+(3)+(4)+(5)+(6)	2.43 мс	2.09 мс

Замечание 1. Время цикла можно автоматически прочитать с ПК программирующим устройством.

2. Максимальное и текущее время цикла хранится в AR 26..AR 27.

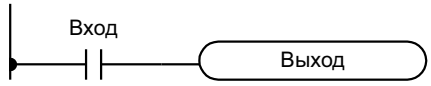
3. Время цикла может варьироваться в зависимости от условий и необязательно будет точно совпадать с вычисленным значением.

4. Минимальное время обслуживание портов RS-232C и периферийного: минимальное - 0.34 мс, максимальное - 87 мс.

7.1.3 Время реакции на вход

Время реакции на вход - это время от получения входного сигнала (т.е. после включения в 1 входного бита), включающее проверку и обработку информации, до выдачи управляющего сигнала на выходной бит. Время реакции на вход варьирует в зависимости от распределения времени и условий отработки.

Далее показано минимальное и максимальное время реакции на вход с использованием в качестве примера следующей диаграммы:



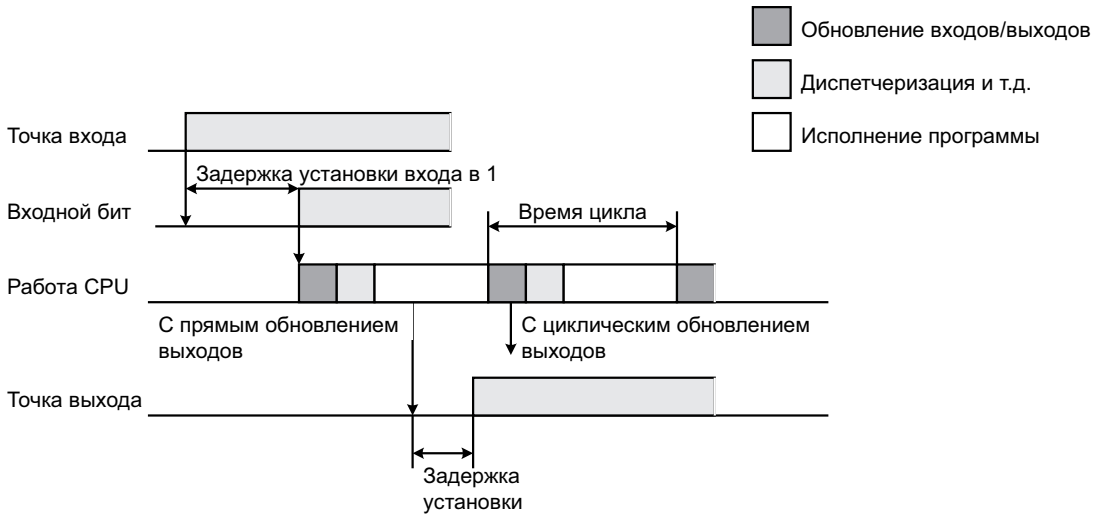
Для примера взяты следующие условия:

Задержка входа:	8 мс
Время диспетчеризации:	1 мс
Время исполнения команды:	14 мс
Задержка выхода:	10 мс
Положение команды выхода:	Начало программы
Порты:	Не используются

Замечание Задержка входа для блоков входа постоянного тока задается в установочных параметрах.

Минимальное время реакции на вход

CQM1 реагирует на входной сигнал быстрее всего, когда получает входной сигнал сразу перед фазой цикла “обновление входов”, как показано на диаграмме.



При использовании циклического обновления выходов:

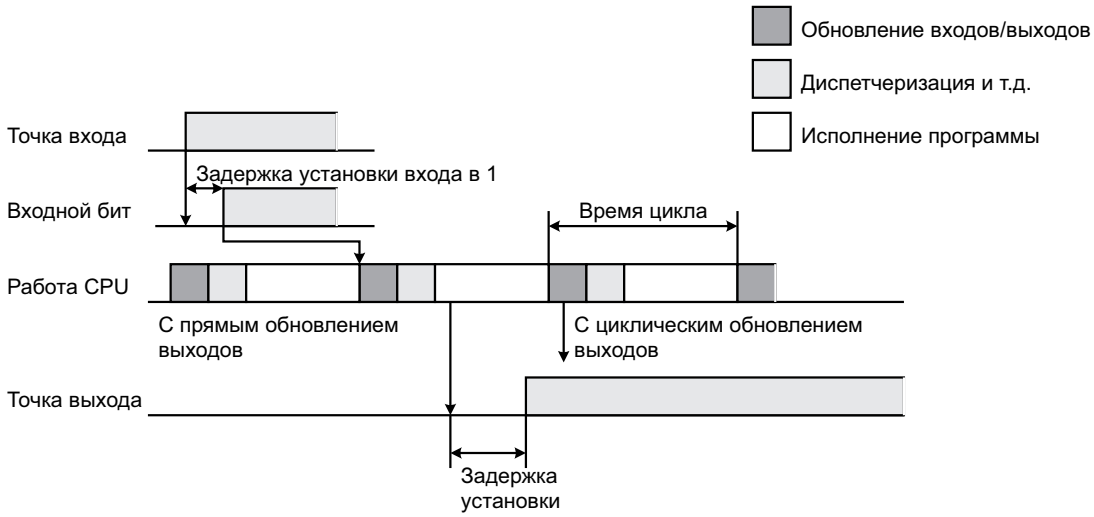
Минимальное время реакции = 8 + 15 + 10 = 33 мс

При использовании прямого обновления выходов:

Минимальное время реакции = 8 + 1 + 10 = 19 мс

Максимальное время реакции на вход

CQM1 реагирует на входной сигнал медленнее всего, когда получает входной сигнал сразу после фазы цикла “обновление входов”, как показано на диаграмме. В данном случае задержка равна примерно одному циклу.



При использовании циклического обновления выходов:

Минимальное время реакции = 8 + 15x2 + 10 = 48 мс

При использовании прямого обновления выходов:

Минимальное время реакции = 8 + 15 + 10 = 33 мс

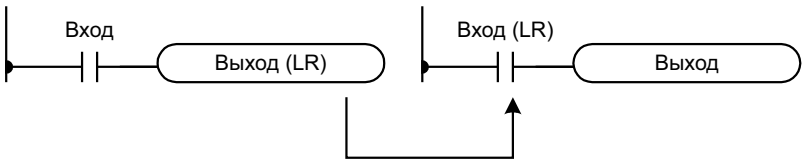
7.1.4 Время реакции на вход при связи 1+1

При связи двух CQM1 1+1 время реакции - это время, требуемое от получения входного сигнала на одном ПК до выдачи управляющего сигнала на выходной бит другого CQM1 посредством связи 1+1.

Связь 1+1 осуществляется поочередным обменом между ведущим и ведомым. Времена обмена показаны в таблице и зависят от числа используемых слов LR.

Число используемых слов LR.	**Время обмена**
64 слова (LR 00..LR 63)	39 мс
32 слова (LR 00..LR 31)	20 мс
16 слова (LR 00..LR 15)	10 мс

Далее показано минимальное и максимальное время реакции на вход с использованием в качестве примера следующей диаграммы , исполняемой ведущим и ведомым. В данном примере связь осуществляется от ведущего к ведомому.



Для примера расчета времени реакции взяты следующие условия:

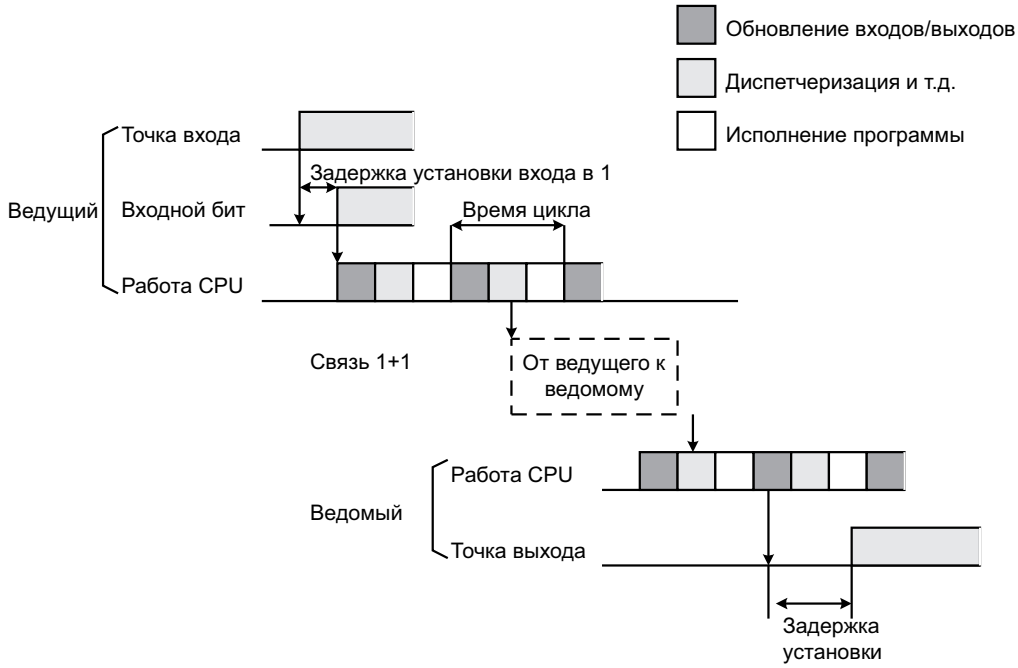
Задержка входа:	8 мс
Время цикла ведущего ПК:	10 мс
Время цикла ведомого ПК:	14 мс
Задержка выхода:	10 мс
Прямой выход:	Не используется
Число используемых слов LR:	64

Замечание Задержка входа для блоков входа постоянного тока задается в установочных параметрах.

Минимальное время реакции на вход

CQM1 реагирует на входной сигнал быстрее всего при следующих обстоятельствах:

- CQM1 получает входной сигнал сразу перед фазой цикла “обновление входов”.
- Передача от ведущего на ведомый ПК начинается немедленно.
- Ведомый ПК исполняет обслуживание связи сразу после завершения связи.



Минимальное время реакции на вход:

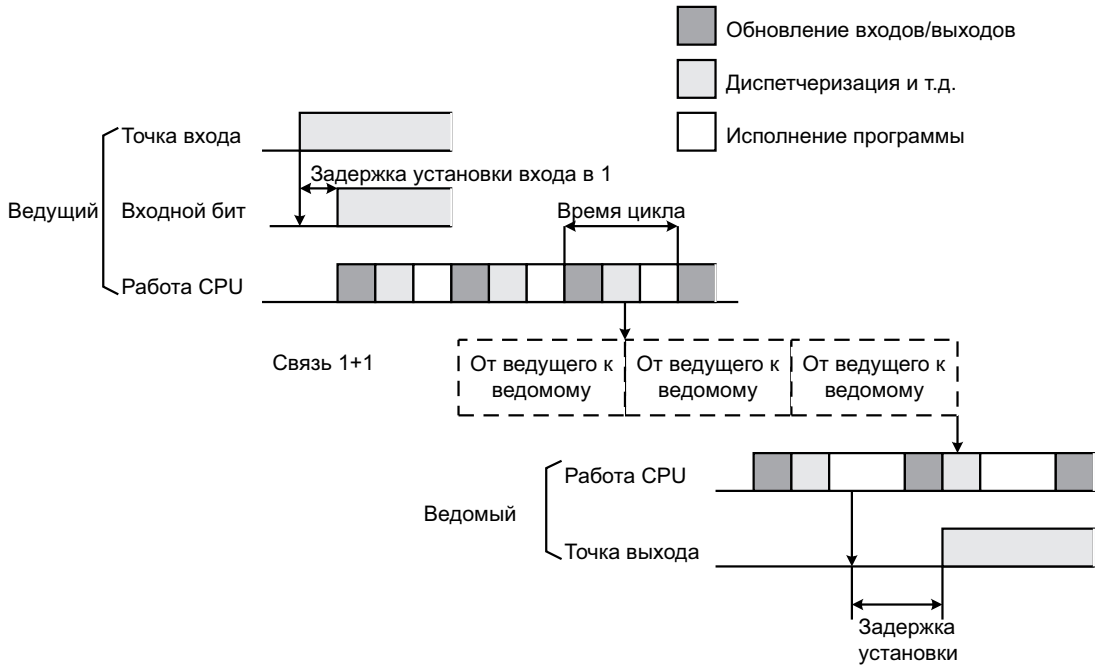
Задержка входа:	8 мс
Время цикла ведущего ПК:	10 мс
Время передачи:	39 мс
Время цикла ведомого ПК:	15 мс
Задержка выхода:	10 мс

Минимальное время реакции на вход: 82 мс

Максимальное время реакции на вход

CQM1 реагирует на входной сигнал медленнее всего при следующих обстоятельствах:

- CQM1 получает входной сигнал сразу после фазы цикла “обновление входов”.
- Передача от ведущего на ведомый ПК не начинается немедленно.
- Связь завершается сразу после обслуживании связи ведомым ПК.



Максимальное время реакции на вход:

Задержка входа:	8 мс
Время цикла ведущего ПК:	10 мс x 2
Время передачи:	39 мс x 3
Время цикла ведомого ПК:	15 мс x 2
Задержка выхода:	10 мс

Минимальное время реакции на вход: 185 мс

7.1.5 Время обработки прерываний

В данной главе описано время обработки прерываний, от момента прерывания до вызова программы прерывания и от времени завершения отработки прерывания до возвращения в исходную позицию. Описываются три типа прерываний:

- входные прерывания
- прерывания интервального таймера
- прерывания высокоскоростного счетчика.

Время обработки

В таблице приведены времена от момента прерывания до вызова программы прерывания и от времени завершения отработки прерывания до возвращения в исходную позицию.

Процесс	Содержание	Время
Время после включения в 1 входа прерывания	Это время от включения в 1 входного бита прерывания до момента, когда это изменение состояния будет принято. К другим прерываниям отношения не имеет.	50 мкс
↓ (Условия для прерывания исполнены) (.см. прим.)		
Задержка до завершения обработки маскирования прерываний	Это время, во время которого прерывания ожидают завершения отработки. Данная ситуация имеет место при исполнении обработки маскирования. Подробности см. далее.	См. ниже
↓		
Переход на обработку прерывания	Это время переключения от текущей отработки на отработку прерывания.	40 мкс

Процесс	Содержание	Время
↓		
Обновление входов во время прерывания	Это время, требуемое для обновления входов, когда задано обновление входов при вызове подпрограммы прерываний. (См. настройку установочных параметров, DM 6630..DM 6638).	10 мкс на слово
↓ (Программа прерывания исполнена)		
Возврат	Время, которое требуется для исполнения RET(93) для возврата обработки в точку, откуда был уход на отработку прерывания.	40 мкс

Замечание 1. Когда применяется высокоскоростной счетчик 0 с таблице сравнения зон, на выделенное время отработки прерывания может влиять время цикла.

2. Когда применяются высокоскоростные счетчики 1 и 2 с таблице сравнения зон (в CQM1-CPU43/44-E), выделенное время отработки прерывания можно продлить до 1 мс.

Обработка маски

Прерывания маскируются описанным далее образом. До завершения обработки все прерывания останутся замаскированными на указанное время.

Высокоскоростные таймеры:

Далее приведенное время требуется в зависимости от количества высокоскоростных таймеров, используемых с TIMH(15); количества активных высокоскоростных таймеров; (Количество высокоскоростных таймеров задается в DM 6629. Значение по умолчанию 16).

0≤Время ожидания≤50+ 3x(a+b) мкс

Может потребоваться до 50 мкс даже когда высокоскоростные таймеры не используются.

Вызов и очистка признаков нефатальных ошибок:

Когда генерируется признак нефатальной ошибки и ее содержание регистрируется ы CQM1, или во время очистки признака ошибки, прерывания будут замаскированы максимум на 100 мкс до завершения обработки.

Редактирование в режиме он-лайн

Прерывания будут замаскированы максимум на 1 с во время редактирования при работе.

На выдачу импульсов командой SPED(64) может также влиять обработка прерываний, так что распределение времени может варьироваться.

Примерный расчет

Пример показывает время реакции на прерывание (т.е. время от включения входа прерывания в 1 до начала программы отработки прерывания), когда прерывания используются при следующих условиях.

Число высокоскоростных таймеров: 0

Редактирование в режиме он-лайн: не используется

Обновление входов при прерывании: нет

Минимальное время реакции

Время задержки после включения в 1 входа прерывания: 50 мкс

Время ожидания до обработки маски прерывания: 0 мкс

Переход на обработку прерывания: 40 мкс

Минимальное время реакции: 90 мс

Максимальное время реакции

Время задержки после включения в 1 входа прерывания: 50 мкс

Время ожидания до обработки маски прерывания: 50 мкс

Переход на обработку прерывания: 40 мкс

Минимальное время реакции:140 мс

В дополнение к времени реакции, приведенном выше, требуется время для исполнения самой программы прерывания и время возврата (40 мкс) в точку, откуда был уход на прерывание.

При использовании прерываний в программе выделяйте требуемое время для обработки процесса прерывания.

Выходы из программы прерываний можно выдавать сразу, если применяется прямой выход. Прямые выходы будут использоваться и для главной программы, и для программ прерываний.

7.1.6 Время исполнения команд CQM1

В следующей таблице перечисляются команды CQM1 и время исполнения. Максимальное и минимальное время и условия, которые их вызывают, даны соответственно. Когда “слово” относится к столбцу УСЛОВИЯ, оно подразумевает содержание любого слова за исключением косвенно адресуемых слов DM. Косвенно адресуемые слова DM, которые требуют большего времени, обозначаются *DM.

Время исполнения для большинства команд зависит от того, исполняются ли они после условия 1 или 0. Исключения - команды OUT и OUT NOT, которые требуют одинакового времени, независимо от условий исполнения. Время исполнения

при условии исполнения 0 может варьировать в зависимости от обстоятельств:

если условие 0 появилось в секции INTERLOCK и при условии исполнения для IL = 0;

если условие 0 появилось между JMP(04) и JMP(05) и условия для JMP(04) = 0;

если условие 0 появилось после сброса условием исполнения 0.

На три этих случая указывают RSET, IL и JMP

Базовые команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
				RSET	IL	JMP
-	LD LD NOT AND AND NOT OR OR NOT AND LD OR LD	0.5	Любое	-		
-	OUT OUT NOT	0.75	Без прямых выходов или, если применяются прямые выходы, для операндов не IR 10000..IR 11515.	-		
-	SET	1.25	Прямые выходы	-		
-	RSET	1.25	Прямые выходы	-		
-	TIM	1.5	Константа для задания Т/Сч	1.5	1.5	1.5
			*DM для задания Т/Сч	54.1	1.5	1.5

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
				RSET	IL	JMP
-	CNT	1.5	Константа для задания Т/Сч	1.5	1.5	1.5
			DM для задания Т/Сч	51.6	1.5	1.5

Базовые команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
00	NOP	0.5	Любые	0.0		
01	END	29		0.0		
02	IL	12.3		10.9		
03	ILC	11.3		11.3		
04	JMP	18.3		11.9		
05	JME	11.0		11.0		
06	FAL	56.8		1.5		
07	FALS	4.0		1.5		
08	STEP	58.2		1.5		
09	SNXT	25.0				
10	SFT			SHIFT	IL	JMP
		44.2	С регистром сдвига в 1 слово	43.2	15.0	15.0
		77.7	С регистром сдвига в 10 слов	68.5	15.0	15.0
		415.2	С регистром сдвига в 100 слов	322.0	15.0	15.0
11	KEEP	0.75	Без прямых выходов или, если применяются прямые выходы, для операндов не IR 10000..IR 11515.			
		1.25	Прямые выходы IR 10000..IR 11515.			
12	CNTR			SHIFT	IL	JMP
		53.0	Константа для задания Т/Сч	33.1	20.7	20.7
		79.6	DM для задания Т/Сч			
13	DIFU	21.5	Любые	21.0	20.8	17.8
14	DIFD	20.8	Любые	20.8	20.6	17.6
15	TIMH	36.5	Константа для задания Т/Сч	54.7	53.0	27.7
		36.5	DM для задания Т/Сч	81.0	79.6	27.7
16	WSFT	44.7	С регистром сдвига в 1 слово	2.0		
		77.0	С регистром сдвига в 10 слов			
		2.25 мс	С регистром сдвига в 1024 слов, используя DM			
		13.05 мс	С регистром сдвига в 6144 слов, используя DM			
20	CMP	26.7	При сравнении константы и слова	2.0		
		29.5	При сравнении двух слов			
		77.3	При сравнении двух DM			
21	MOV	23.5	При посылке константы в слово	2.0		
		29.5	При посылке из одного слова в другое			
		77.3	При посылке из DM в DM			
22	MVN	23.7	При посылке константы в слово	2.0		
		26.5	При посылке из одного слова в другое			
		72.6	При посылке из DM в DM			

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
23	BIN	50.4	При преобразовании слова в слово	2.0
		96.0	При преобразовании из DM в DM	
24	BCD	47.7	При преобразовании слова в слово	2.0
		93.3	При преобразовании из DM в DM	
25	ASL	24.0	При сдвиге слова	1.5
		45.8	При сдвиге DM	
26	ASR	24.0	При сдвиге слова	1.5
		45.8	При сдвиге DM	
27	ROL	24.7	При сдвиге слова	1.5
		46.6	При сдвиге DM	
28	ROR	24.7	При сдвиге слова	1.5
		46.6	При сдвиге DM	
29	COM	25.9	При инвертировании слова	1.5
		48.3	При инвертировании DM	
30	ADD	49.9	Константа + слово слово	2.5
		53.1	Слово + слово слово	
		122.1	DM + DM DM	
31	SUB	49.9	Константа - слово слово	2.5
		53.1	Слово - слово слово	
		122.1	DM - DM DM	
32	MUL	73.7	Константа x слово слово	2.5
		77.0	Слово x слово слово	
		144.5	DM x DM DM	
33	DIV	72.7	Константа слово слово	2.5
		75.4	Слово слово слово	
		143.0	DM DM DM	
34	ANDW	41.9	Константа слово слово	2.5
		45.1	Слово слово слово	
		114.1	DM DM DM	
35	ORW	41.9	Константа V слово слово	2.5
		45.1	Слово V слово слово	
		114.1	DM V DM DM	
36	XORW	41.9	Константа слово слово	2.5
		45.2	Слово слово слово	
		114.1	DM DM DM	
37	XNRW	41.9	Константа слово слово	2.5
		45.1	Слово слово слово	
		114.1	DM DM DM	
38	INC	27.8	Инкремент слова	1.5
		50.1	Инкремент DM	
39	DEC	28.4	Декремент слова	1.5
		50.8	Декремент DM	
40	STC	12.0	Любое	1.0
41	CLC	12.0		1.0
45	TRSM	28.8		1.0

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
46	MSG	24.6	Сообщение в словах	1.5
		48.4	Сообщение DM	
50	ADB	53.4	Константа + слово слово	2.5
		56.6	Слово + слово слово	
		125.6	DM + DM DM	
51	SBB	53.4	Константа - слово слово	2.5
		56.6	Слово - слово слово	
		125.6	DM - DM DM	
52	MLB	45.7	Константа x слово слово	2.5
		48.9	Слово x слово слово	
		116.4	DM x DM DM	
53	DVB	46.7	Константа слово слово	2.5
		49.9	Слово слово слово	
		117.4	DM DM DM	
54	ADDL	59.3	Слово + слово слово	2.5
		128.9	DM + DM DM	
55	SUBL	59.3	Слово - слово слово	2.5
		128.9	DM - DM DM	
56	MULL	204.5	Слово x слово слово	2.5
		271.2	DM x DM DM	
57	DIVL	205.9	Слово слово слово	2.5
		272.6	DM DM DM	
58	BINL	76.0	Слово слово	2.0
		120.6	DM DM	
59	BCDL	60.9	Слово слово	2.0
		105.6	DM DM	
70	XFER	72.9	При пересылке константы в слово	2.5
		76.1	При пересылке слова в слово	
		2.90 мс	При пересылке 1024 слов, используя DM	
		16.66 мс	При пересылке 6144 слов, используя DM	
71	BSET	45.6	При установке константы в 1 слово	2.5
		77.9	При пересылке константы в 10 слов	
		1.93 мс	При пересылке DM в 1024 слов	
		10.95 мс	При пересылке DM в 6144 слов	
72	ROOT	63.9	Расчет из слова слово	2.0
		110.8	Расчет из DM DM	
73	XCHG	40.9	Слово слово	2.0
		85.5	DM DM	
74	SLD	41.1	Сдвиг 1 слова	2.0
		101.9	Сдвиг 10 слов	
		5.49 мс	Сдвиг 1024 слов, используя DM	
		32.58 мс	Сдвиг 6144 слов, используя DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
75	SRD	41.1	Сдвиг 1 слова	2.0
		101.9	Сдвиг 10 слов	
		5.49 мс	Сдвиг 1024 слов, используя DM	
		32.58 мс	Сдвиг 6144 слов, используя DM	
76	MLPX	59.1	Декодирование слова в слово	2.5
		136.4	Декодирование DM DM	
77	DMPX	45.1	Кодирование слова в слово	2.5
		120.6	Кодирование DM DM	
78	SDEC	60.6	Декодирование слова в слово	2.5
		138.5	Декодирование DM DM	
80	DIST	66.0	При задании константы в слово + слово	2.5
		69.3	При задании слова в слово + слово	
		144.3	При задании DM в DM + DM	
		101.0	При задании константы в стек	
		104.3	При задании слова в стек	
		177.8	При задании DM в стек	
81	COLL	65.1	При задании константа + слово в слово	2.5
		68.3	При задании слово + слово в слово	
		140.1	При задании DM + DM в DM	
		61.1	При задании слово + константа в стек FIFO	
		64.3	При задании слова + слово в стек FIFO	
		137.6	При задании DM + DM в стек FIFO посредством DM	
		60.3	При задании слово + константа в стек LIFO	
		63.6	При задании слова + слово в стек LIFO	
		136.8	При задании DM + DM в стек LIFO посредством DM	
82	MOVB	46.4	При пересылке константы в слово	2.5
		54.9	При пересылке слова в слово	
		125.2	При пересылке DM в DM	
83	MOVD	40.7	При пересылке константы в слово	2.5
		49.2	При пересылке слова в слово	
		119.4	При пересылке DM в DM	
84	SFTR	57.4	Сдвиг 1 слова	2.5
		98.4	Сдвиг 10 слов	
		2.24 мс	Сдвиг 1024 слов, используя DM	
		12.9 мс	Сдвиг 6144 слов, используя DM	
85	TCMP	95.8	Сравнение константы с таблицей, заданной словами	2.5
		98.8	Сравнение слова с таблицей, заданной словами	
		169.0	Сравнение DM с таблицей, заданной DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
86	ASC	62.5	Слово слово	2.5
		144.3	DM DM	
91	SBS	41.4	Любое	1.5
92	SBN	-		-
93	RET	39.0		1.5
97	IOFR	37.7	Обновление IR 000	2.0
		41.7	Обновление одного слова входа	
		46.9	Обновление одного слова выхода	
		112.4	Обновление 8 слов входа/выхода	
99	MCRO	140.2	С операндами входа/выхода, заданными словом	2.5
		188.1	С операндами входа/выхода, заданными DM	

Дополнительные команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
17	ASFT	62.7	Сдвиг 1 слова	2.5
		96.7	Сдвиг 10 слов	
		2.45 мс	Сдвиг 1024 слов, используя DM	
		16.33 мс	Сдвиг 6144 слов, используя DM	
18	TKY	81.1	Слово слово	2.5
		131.8	DM DM	
19	MCMP	123.9	Сравнение слов	2.5
		195.3	Сравнение DM	
47	RXD	123.1	Ввод 1 байта через слово	2.5
		843.7	Ввод 256 байт через DM	
48	TDX	105.1	Вывод 1 байта через слово (RS-232C)	2.5
		832.3	Вывод 256 байт через DM (RS-232C)	
		86.3	Вывод 1 байта через слово (по линии связи с верхним уровнем)	
		141.9	Вывод 256 байт через DM (по линии связи с верхним уровнем)	
60	CMPL	50.9	Сравнение слов	2.5
		101.0	Сравнение DM	
61	INI	Высокоскоростной счетчик 0 или выдача импульсов с бита выхода		2.5
		90.6	Пуск сравнения через слово	
		114.4	Пуск сравнения через DM	
		72.1	Останов сравнения через слово	
		83.0	Останов сравнения через DM	
		163.6	Изменение текущего значение через слово	
		182.2	Изменение текущего значение через DM	
		56.4	Останов выдачи импульсов через слово	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
		80.2	Останов выдачи импульсов через DM	
		Высокоскоростные счетчики 1 и 2 или выдача импульсов с портов 1 или 2		
		296.8	Пуск сравнения через слово	
		324.3	Пуск сравнения через DM	
		207.3	Останов сравнения через слово	
		232.8	Останов сравнения через DM	
		468.3	Изменение текущего значение через слово	
		487.8	Изменение текущего значение через DM	
		248.8	Останов выдачи импульсов через слово	
		269.8	Останов выдачи импульсов через DM	
		Абсолютные высокоскоростные счетчики 1 и 2		
		296.3	Пуск сравнения через слово	
		316.8	Пуск сравнения через DM	
		202.3	Останов сравнения через слово	
		226.3	Останов сравнения через DM	
62	PRV	Высокоскоростной счетчик 0 или выдача импульсов с бита выхода		2.5
		91.5	Задание выхода через слово	
		117.4	Задание выхода через DM	
		Высокоскоростные счетчики 1 и 2 или выдача импульсов с портов 1 или 2		
		229.3	Задание выхода через слово (чтение состояния)	
		249.3	Задание выхода через DM (чтение состояния)	
		229.8	Задание выхода через слово (чтение результатов зоны сравнения)	
		256.3	Задание выхода через DM (чтение результатов зоны сравнения)	
		Абсолютные высокоскоростные счетчики 1 и 2		
		226.3	Задание выхода через слово (чтение состояния)	
		253.3	Задание выхода через DM (чтение состояния)	
		227.8	Задание выхода через слово (чтение результатов зоны сравнения)	
		253.3	Задание выхода через DM (чтение результатов зоны сравнения)	
63	CTBL	Высокоскоростной счетчик 0 или выдача импульсов с бита выхода		2.5
		210.3	Таблица для сравнения по значению с 1 значением (слово) и пуск.	
		233.8	Таблица для сравнения по значению с 1 значением (DM) и пуск.	
		1.31 мс	Таблица для сравнения по значению с 16 значениями (слово) и пуск.	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
		1.33 мс	Таблица для сравнения по значению с 16 значениеми (DM) и пуск.	
		1.25 мс	Таблица для сравнения по зоне (слово) и пуск.	
		1.27 мс	Таблица для сравнения по зоне (DM) и пуск.	
		170.8	Таблица для сравнения по значению с 1 значением (слово).	
		194.3	Таблица для сравнения по значению с 1 значением (DM)	
		1.27 мс	Таблица для сравнения по значению с 16 значениями (слово).	
		1.30 мс	Таблица для сравнения по значению с 16 значениями (DM).	
		1.09 мс	Таблица для сравнения по зоне (слово).	
		1.11 мс	Таблица для сравнения по зоне (DM).	
		Высокоскоростные счетчики 1 и 2 или выдача импульсов с портов 1 или 2		
		692.8	Таблица для сравнения по значению с 1 значением (слово) и пуск.	
		721.8	Таблица для сравнения по значению с 1 значением (DM) и пуск.	
		2.79 мс	Таблица для сравнения по значению с 16 значениями (слово) и пуск.	
		2.81 мс	Таблица для сравнения по значению с 16 значениями (DM) и пуск.	
		2.26 мс	Таблица для сравнения по зоне (слово) и пуск.	
		2.27 мс	Таблица для сравнения по зоне (DM) и пуск.	
		488.8	Таблица для сравнения по значению с 1 значением (слово).	
		517.8	Таблица для сравнения по значению с 1 значением (DM)	
		2.57 мс	Таблица для сравнения по значению с 16 значениями (слово).	
		2.61 мс	Таблица для сравнения по значению с 16 значениями (DM).	
		2.19 мс	Таблица для сравнения по зоне (слово).	
		2.21 мс	Таблица для сравнения по зоне (DM).	
		Абсолютные высокоскоростные счетчики 1 и 2		
		600.8	Таблица для сравнения по значению с 1 значением (слово) и пуск.	
		624.8	Таблица для сравнения по значению с 1 значением (DM) и пуск.	
		2.33 мс	Таблица для сравнения по значению с 16 значениями (слово) и пуск.	
		2.34 мс	Таблица для сравнения по значению с 16 значениями (DM) и пуск.	
		1.47 мс	Таблица для сравнения по зоне (слово) и пуск.	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
		1.50 мс	Таблица для сравнения по зоне (DM) и пуск.	
		460.8	Таблица для сравнения по значению с 1 значением (слово).	
		484.8	Таблица для сравнения по значению с 1 значением (DM)	
		2.02 мс	Таблица для сравнения по значению с 16 значениями (слово).	
		2.04 мс	Таблица для сравнения по значению с 16 значениями (DM).	
		1.45 мс	Таблица для сравнения по зоне (слово).	
		1.47 мс	Таблица для сравнения по зоне (DM).	
64	SPED	Выдача импульсов с выходного бита:		2.5
		118.4	Частота задана константой	
		123.2	Частота задана словом	
		146.8	Частота задана DM	
		Выдача импульсов с портов 1 или 2		
		302.3	Частота задана константой	
		310.3	Частота задана словом	
		320.3	Частота задана DM	
65	PULS	Выдача импульсов с выходного бита:		2.5
		109.0	Число импульсов задано словом	
		137.8	Число импульсов задано DM	
		Выдача импульсов с портов 1 или 2		
		337.3	Число импульсов задано словом	
		360.3	Число импульсов задано DM	
66	SCL	105.8	Приемник - слово	2.5
		180.5	Приемник - DM	
67	BCNT	88.4	Подсчет битов в слове	2.5
		49.32 мс	Подсчет битов в 6 656 словах используя DM	
68	BCMP	140.0	Сравнение констант, результат - в слово	2.5
		143.0	Сравнение слов, результат - в слово	
		194.7	Сравнение DM , результат - в DM	
69	STIM	36.8	Пуск разового прерывания, задание - слово	2.5
		73.8	Пуск разового прерывания, задание - DM	
		37.3	Пуск прерывания по расписанию, задание - слово	
		74.3	Пуск прерывания по расписанию, задание - DM	
		66.4	Чтение таймера, задание - слово	
		113.6	Чтение таймера, задание - DM	
		35.3	Останов таймера, задание - слово	
		35.6	Останов таймера, задание - DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
87	DSW	70.3	Выдача сигнала CS, задан словом, 4 разряда	2.5
		70.3	Выдача сигнала RD, задан словом, 4 разряда	
		89.1	Ввод данных, задание - слово, 4 разряда	
		93.1	Выдача сигнала CS, задан DM , 4 разряда	2.5
		93.1	Выдача сигнала RD, задан DM , 4 разряда	
		110.3	Ввод данных, задание - DM , 4 разряда	
		74.7	Выдача сигнала CS, задан словом, 8 разрядов	2.5
		75.1	Выдача сигнала RD, задан словом, 8 разрядов	
		105.5	Ввод данных, задание - слово, 8 разрядов	
		103.5	Выдача сигнала CS, задан DM , 8 разрядов	2.5
		103.9	Выдача сигнала RD, задан DM , 8 разрядов	
		131.5	Ввод данных, задание - DM , 8 разрядов	
88	7SEG	78.7	4 разряда, приемник задан словом	2.5
		102.6	4 разряда, приемник задан DM	
		92.1	8 разрядов, приемник задан словом	
		117.6	8 разрядов, приемник задан DM	
89	INT	53.0	Задать маску с помощью слова	2.5
		80.8	Задать маску с помощью DM	
		49.9	Очистить прерывания с помощью слова	
		73.2	Очистить прерывания с помощью DM	
		50.7	Чтение состояния маски с помощью слова	
		71.9	Чтение состояния маски с помощью DM	
		64.8	Изменение задания счетчика с помощью слова	
		88.1	Изменение задания счетчика с помощью DM	
		27.5	Маскировать все прерывания с помощью слова	
		27.5	Маскировать все прерывания с помощью DM	
		28.5	Очистить все прерывания с помощью слова	
		28.5	Очистить все прерывания с помощью DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
	HKY	71.5	Выдача из слова в слово	2.5
		100.3	Выдача из DM в DM	
		81.5	Ввод из слова в слово	
		105.5	Ввод из DM в DM	
	FPD	171.6	Приемник задан словом, нет сообщений, исполнение	2.5
		279.5	Приемник задан DM , нет сообщений, исполнение	
		204.9	Приемник задан словом, нет сообщений, инициация	
		312.0	Приемник задан DM , нет сообщений, инициация	
	SRCH	62.4	Искать слово, результат - в слово	2.5
		2.64 мс	Искать 1024 слов с помощью DM, результат - в DM	
		15.11 мс	Искать 6144 слов с помощью DM, результат - в DM	
	MAX	56.1	Поиск слова, результат - в слово	2.5
		2.56 мс	Поиск 999 слов с помощью DM, результат - в DM	
	MIN	56.1	Поиск слова, результат - в слово	2.5
		2.56 мс	Поиск 999 слов с помощью DM, результат - в DM	
	APR	57.4	Подсчет синуса	2.5
		460.4	Линейная аппроксимация с таблицей из 256 значений с помощью приемника DM.	
	LINE	93.4	Слово в слово	2.5
		166.5	DM в DM	
	COLM	115.1	Слово в слово	2.5
		183.1	DM в DM	
	SEC	92.9	Слово в слово	2.5
		146.2	DM в DM	
	HMS	94.9	Слово в слово	2.5
		148.7	DM в DM	
	SUM	72.9	Добавить одно слово, результат - в слово	2.5
		6.86 мс	Сложить 999 слов, с помощью DM, результаты - в DM	
	FCS	73.6	Подсчет 1 слова, результаты - в слово	2.5
		2.33 мс	Подсчет 999 слов, с помощью DM, результаты - в DM	
	HEX	82.3	Слово в слово	2.5
		154.3	DM в DM	
	AVG	73.7	Среднее за 1 цикл для слова	2.5
		282.5	Среднее за 64 цикла для DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
	PWM	266.8	Коэффициент заполнения задается константой	2.5
		266.8	Коэффициент заполнения задается константой	
		272.8	Коэффициент заполнения задается словом	
		293.8	Коэффициент заполнения задается DM	
	PID	2.11	Слово в слово (исполнение исходное)	2.5
		2.30 мс	DM в DM (исполнение исходное)	
		607.7	Слово в слово (при выборке)	
		893.7	DM в DM (при выборке)	
	ADBL	75.4	Слово + слово слово	2.5
		152.0	DM + DM DM	
	SBBL	75.4	Слово - слово слово	2.5
		152.0	DM - DM DM	
	MBS	57.9	Константа x слово слово	2.5
		61.1	Слово x слово слово	
		135.0	DM x DM DM	
	DBS	28.3	Константа слово слово	2.5
		62.4	Слово слово слово	
		136.4	DM DM DM	
	MBSL	94.0	Слово x слово слово	2.5
		167.6	DM x DM DM	
	DBSL	86.3	Слово слово слово	2.5
		160.4	DM DM DM	
	CPS	31.0	Сравнение константы и слова	2.5
		33.7	Сравнение слов	
		82.4	Сравнение DM	
	CPSL	51.3	Сравнение слов	2.5
		102.6	Сравнение DM	
	NEG	41.3	Преобразование константа слово	2.5
		44.5	Преобразование слово слово	
		92.7	Преобразование DM DM	
	NEGL	51.1	Преобразование константа слово	2.5
		103.2	Преобразование DM DM	
	ZCP	38.2	Сравнение константы с зоной слов	2.5
		44.7	Сравнение слова с зоной слов	
		114.6	Сравнение DM с зоной DM	
	CPSL	77.7	Сравнение слова с зоной слов	2.5
		151.4	Сравнение DM с зоной DM	
	XFRB	35.3	Передача 1 бита между словами с константой для данных управления	2.5
		56.8	Передача 1 бита между словами со словом для данных управления	
		298.3	Передача 255 бит между DM с DM для данных управления	

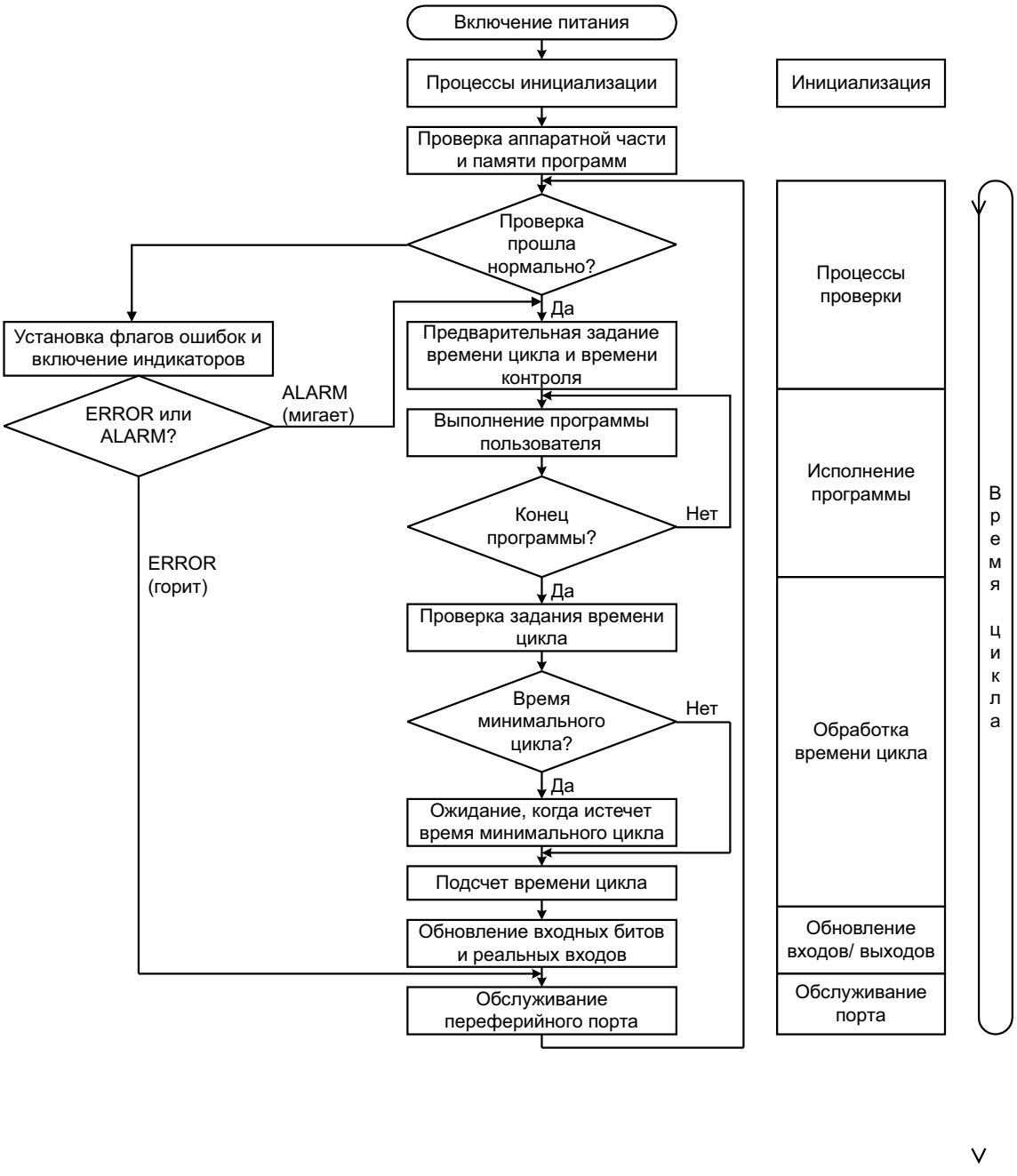
Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
	PLS2	821.7	Слова для параметров управления	2.5
		849.0	DM для параметров управления	
	ACC	547.3	Режим 0: слова для слов управления	2.5
		577.0	Режим 0: DM для слов управления	
		392.8	Режим 1: слова для слов управления	
		424.0	Режим 1: DM для слов управления	
		404.8	Режим 2: слова для слов управления	
		430.3	Режим 2: DM для слов управления	
		259.5	Режим 3: слова для слов управления	
		418.3	Режим 3: DM для слов управления	
	SCL2	105.0	Преобразование слово в слово, слова для слов параметров	2.5
		179.8	Преобразование DM в DM, DM для слов параметров	
	SCL3	112.0	Преобразование слово в слово, слова для слов параметров	2.5
		186.8	Преобразование DM в DM, DM для слов параметров	

7.2 Время цикла CPM1/CPM1A и время реакции входов/выходов

7.2.1 Цикл CPM1/CPM1A

Алгоритм работы CPM1/CPM1A:

На диаграмме показан общий алгоритм работы CPM1.



Замечание Процессы инициализации включают очистку областей IR, SR, и AR, установку системных таймеров и проверка блоков входов/выходов.

7.2.2 Время цикла CPM1/CPM1A

В таблице приведены процессы, происходящие в одном цикле CPM1, и соответствующее время обработки.

Процесс	Назначение	Требуемое время
Общая диспетчеризация	Установка контрольного таймера, проверка шины входов/выходов, проверка UM, обновление часов, обновление битов, отведенных под новые функции и т. д.	0.6 мс
Исполнение программы.	Исполнение программы пользователя.	Общее время исполнения команд. (Зависит от содержания программы).
Расчет времени цикла	Задержка, пока время минимального цикла занесется в DM 6619. Расчет времени цикла.	Почти мгновенно, за исключением времени задержки.
Обновление входов/выходов	Чтение информации с блоков входов в биты входа. Выдача информации (результата исполнения программы) на выходы блоков выхода.	ЦУ на 10 точек :0.06 мс ЦУ на 20 точек:0.06 мс ЦУ на 30 точек: 0.3 мс Блок расширения входов/выходов:0.3 мс
Обслуживание периферийного порта	Обслуживается устройство, подключенное к периферийному порту	мин. 0.34 мс, 5% или менее от времени цикла до 87 мс (см. прим.)

Замечание Процент времени цикла, выделенный на обслуживание периферийного порта задается в установочных параметрах DM 6617.

Влияние времени цикла на операции

Влияние времени цикла на операции показано в таблице. Когда долгое время цикла влияет на работу, либо сократите время цикла, либо улучшите время реакции путем программ прерываний.

Время цикла	Условия работы
10 мс и более	Может неточно исполняться TIMH(15), когда применяются TC 004..TC 127 (операции для TC 000..TC 003 будут исполняться нормально).
20 мс и более	Может неточно исполняться программа, использующая бит часов 0.02 с (SR 25401).
100 мс и более	Может неточно исполняться TIM. Может неточно исполняться программа, использующая бит часов 0.1 с (SR 25500). Генерируется признак ошибки ПРЕВЫШЕНИЕ ВРЕМЕНИ ЦИКЛА (включается в 1 SR 25309) (см. прим. 1).
120 мс и более	Превышено контрольное время FALS 9F. Генерируется признак ошибки системы (FALS 0F) и работа прекращается. (см. прим. 2).
200 мс и более	Может неточно исполняться программа, использующая бит часов 0.2 с (SR 25501).

Замечание 1. Для запрещения ошибки ПРЕВЫШЕНИЕ ВРЕМЕНИ ЦИКЛА используется DM 6655.

2. Время контроля FALS 9F за циклом изменяется параметром DM 6618.

Пример расчет времени цикла

В данном примере рассчитывается время цикла для CPM1 CPU с 20 входами/выходами, которые распределяются следующим образом:

Входы: 1 слово (00000..00011)

Выходы: 1 слово (01000..01007)

Остальные условия работы:

Программа пользователя: 500 команд (состоят из команд LD и OUT).

Время цикла: переменное (минимум не задан)

Среднее время обработки одной команды программы пользователя принимется 2.86 мкс. Распределение времени цикла показано в следующей таблице.

Процесс	Способ подсчета	Время с периферийным устройством	Время без периферийного устройства
Общая диспетчеризация	Фиксированное время	0.6 мс	0.6 мс
Исполнение программы.	2.86 x 500 (мкс)	1.43 мс	1.43 мс
Расчет времени цикла	Пренебрегаем	0 мс	0 мс
Обновление входов/выходов	0.01 x 1 + 0.005 x1 (мкс)	0.06 мс	0.06 мс
Обслуживание периферийного порта	Минимальное время	0.34 мс	0 мс
Время цикла	(1)+(2)+(3)+(4)+(5)	2.43 мс	2.09 мс

Замечание 1. Время цикла можно автоматически прочитать с ПК программирующим устройством.

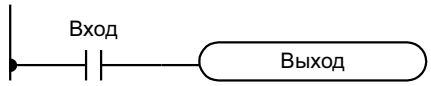
2. Максимальное и текущее время цикла хранится в AR 14..AR 15.

3. Время цикла может варьироваться в зависимости от условий и необязательно будет точно совпадать с вычисленным значением.

7.2.3 Время реакции на вход

Время реакции на вход - это время от получения входного сигнала (т.е. после включения в 1 входного бита), включающее проверку и обработку информации, до выдачи управляющего сигнала на выходной бит. Время реакции на вход варьирует в зависимости от распределения времени и условий отработки.

Далее показано минимальное и максимальное время реакции на вход с использованием в качестве примера следующей диаграммы:

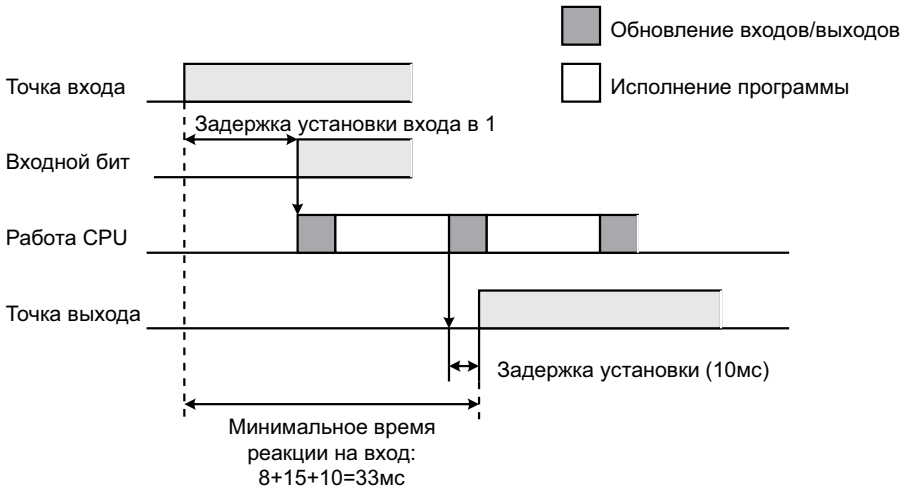


Для примера расчета времени реакции взяты следующие условия:

Задержка входа:	8 мс (задержка входа; значение по умолчанию)
Время диспетчеризации:	1 мс
Время исполнения команды:	14 мс
Задержка выхода:	10 мс
Положение команды выхода:	Начало программы
Периферийный порт:	Не используется

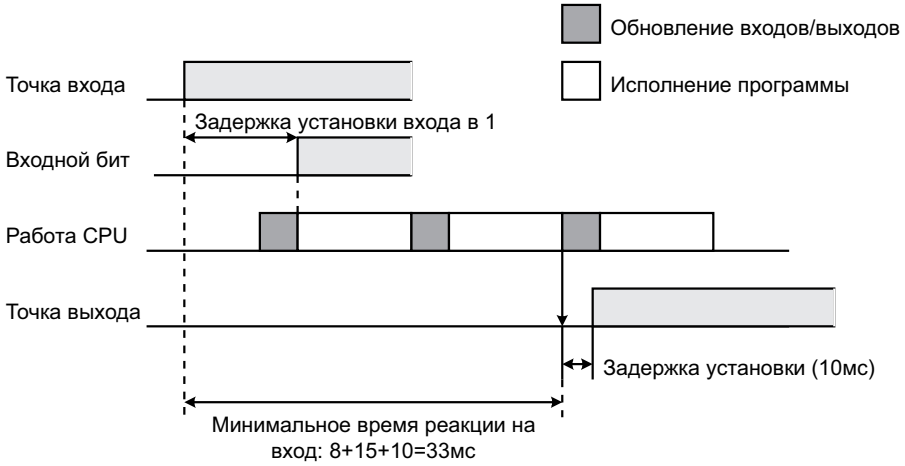
Минимальное время реакции на вход

CPM1 реагирует на входной сигнал быстрее всего, когда получает входной сигнал сразу перед фазой цикла “обновление входов”, как показано на диаграмме.



Максимальное время реакции на вход

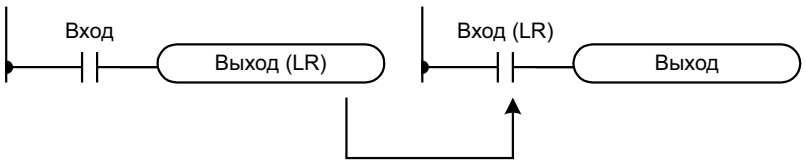
CQM1 реагирует на входной сигнал медленнее всего, когда получает входной сигнал сразу после фазы цикла “обновление входов”, как показано на диаграмме. В данном случае задержка равна примерна одному циклу.



7.2.4 Время реакции на вход при связи 1:1

При связи двух CPM1/CPM1A 1:1 время реакции - это время, от получения входного сигнала на одном ПК до выдачи управляющего сигнала на выходной бит другого CPM1 посредством связи 1:1.

Далее показано минимальное и максимальное время реакции на вход с использованием в качестве примера следующей диаграммы , исполняемой ведущим и ведомым. В данном примере связь осуществляется от ведущего к ведомому.



Для примера расчета времени реакции взяты следующие условия:

Задержка входа:	8 мс (задержка входа; значение по умолчанию)
Время цикла ведущего ПК:	10 мс
Время цикла ведомого ПК:	14 мс
Задержка выхода:	10 мс

Периферийный порт:	Не используется
Число используемых слов LR:	64

Минимальное время реакции на вход

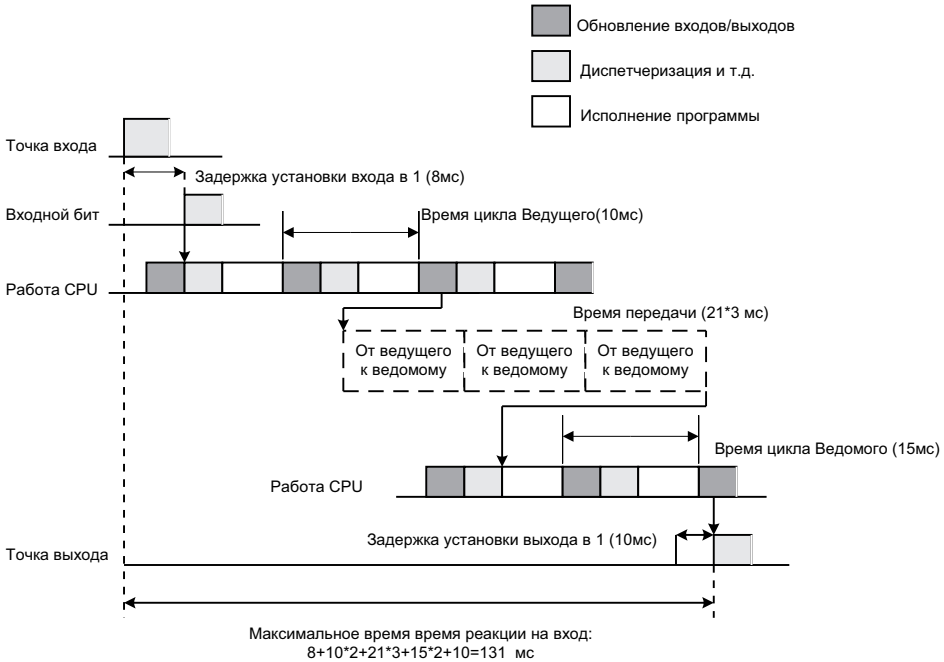
CPM1/CPM1A реагирует на входной сигнал быстрее всего при следующих обстоятельствах:

- CPM1/CPM1A получает входной сигнал сразу перед фазой цикла “обновление входов”.
- Передача от ведущего на ведомый ПК начинается немедленно.
- Ведомый ПК Начинает обслуживание связи сразу после завершения передачи.

Максимальное время реакции на вход

CPM1 реагирует на входной сигнал медленнее всего при следующих обстоятельствах:

- CPM1 получает входной сигнал сразу после фазы цикла “обновление входов”.
- Передача от ведущего на ведомый ПК не начинается немедленно.
- Связь завершается сразу после обслуживании связи ведомым ПК.



7.2.5 Время обработки прерываний

В данной главе описано время обработки прерываний, от момента прерывания до вызова программы прерывания и от времени завершения отработки прерывания до возвращения в исходную позицию. Описываются три типа прерываний:

- входные прерывания
- прерывания интервального таймера
- прерывания высокоскоростного счетчика.

1, 2, 3,... 1. Источник прерываний
2. Задержка вызова прерывания в 1
3. Ожидание завершения обработки маски прерывания
4. Переход к обработке прерывания
5. Роутинг прерываний (только CPM1A)
6. Возврат в исходную позицию

Процесс	Содержание	Время
Время после включения в 1 входа прерывания	Это время от включения в 1 входного бита прерывания до момента, когда это изменение состояния будет принято. К другим прерываниям отношения не имеет.	100 мкс
Задержка до завершения обработки маскирования прерываний	Это время, во время которого прерывания ожидают завершения обработки . Данная ситуация имеет место при исполнении обработки маскирования. Подробности см. далее.	См. ниже
Переход на обработку прерывания	Это время переключения от текущей отработки на отработку прерывания.	30 мкс
Возврат	Время, которое требуется для исполнения RET(93) для возврата обработки в точку, откуда был уход на отработку прерывания.	40 мкс

Обработка маски

Прерывания маскируются описанным далее образом. До завершения обработки все прерывания останутся замаскированными на указанное время.

Вызов и очистка признаков нефатальных ошибок:

Когда генерируется признак нефатальной ошибки и ее содержание регистрируется в CPM1, или во время очистки признака ошибки, прерывания будут замаскированы максимум на 100 мкс до завершения обработки.

Редактирование в режиме он-лайн

Прерывания будут замаскированы максимум на 600 мкс (т.е. редактирование DM 6144..DM 6655) когда во время работы идет редактирование в режиме он-лайн. Кроме того, может возникнуть задержка максимум до 170 мкс.

Пример расчета

Пример показывает время реакции на прерывание (т.е. время от включения входа прерывания в 1 до начала программы отработки прерывания), когда прерывания используются при следующих условиях.

Минимальное время реакции

Время задержки после включения в 1 входа прерывания: 100 мкс

Время ожидания до обработки маски прерывания: 0 мкс

Переход на обработку прерывания: 30 мкс

Минимальное время реакции:130 мс

Максимальное время реакции

(За исключением он-лайнового редактирования DM 6144..DM 6655)

Время задержки после включения в 1 входа прерывания: 100 мкс

Время ожидания до обработки маски прерывания: 170 мкс

Переход на обработку прерывания: 30 мкс

Минимальное время реакции:300 мс

В дополнение к времени реакции, приведенном выше, требуется время для исполнения самой программы прерывания и время возврата (30 мкс) в точку, откуда был уход на прерывание.

7.2.6 Время исполнения команд CPM1/CPM1A

В следующей таблице перечисляются команды CPM1/CPM1A и время исполнения.

Базовые команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
				RSET	IL	JMP
- -	LD LD NOT	1.72	Любое	-		
- - - -	AND AND NOT OR OR NOT	1.32				
- -	AND LD OR LD	0.72				
- -	OUT OUT NOT	4.0				
-	SET	5.8				
-	RESET	5.9				
-	TIM	10.0	Константа для задания Т/Сч	16.2	16.0	6.4
			DM для задания Т/Сч	31.4	31	6.4
-	CNT	12.5	Константа для задания Т/Сч	14.1	6.2	6.6
			DM для задания Т/Сч	29.1	6.2	6.6

Базовые команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
00	NOP	0.36	Любые			
01	END	10.8				
02	IL	4.6		2.6		
03	ILC	3.6		3.6		
04	JMP	4.3		2.4		
05	JME	4.7		4.7		
06	FAL	38.5		5.5		
07	FALS	5.0		5.4		
08	STEP	14.9		11.1		
09	SNXT	14.2		7.6		
				RSET	IL	JMP

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
10	SFT	21.9	С регистром сдвига в 1 слово	19.7	2.6	2.6
		34.1	С регистром сдвига в 10 слов	26.5	2.6	2.6
		93.9	С регистром сдвига в 100 слов	60.1	2.6	2.6
11	KEEP	6.2	Любые	6.1	3.1	3.1
12	CNTR	25.8	Константа для задания Т/Сч	16.8	12.2	12.2
		41.2	DM для задания Т/Сч			
13	DIFU	11.8	Любые	10.1	12.2	12.2
14	DIFD	11.0		10.0	9.9	2.3
15	TIMH	19.0	Регулярное исполнение, константа для задания Т/Сч	25.7	28.4	15.8
		20.2	Исполнение по прерыванию, константа для задания Т/Сч			
		19.0	Регулярное исполнение, DM для задания Т/Сч	41.2	43.6	15.8
		20.2	Исполнение по прерыванию, DM для задания Т/Сч			
16	WSFT	29.2	С регистром сдвига в 1 слово	5.6		
		40.7	С регистром сдвига в 10 слов			
		1.42 мс	С регистром сдвига в 1024 слов, используя DM			
17	ASFT	29.6	Сдвиг 1 слова	5.6		
		50.2	Сдвиг 10 слов			
		46.3	Сдвиг 1024 слов, с использованием DM			
20	CMP	15.8	При сравнении константы и слова	5.6		
		17.2	При сравнении двух слов			
		46.3	При сравнении двух DM			
21	MOV	16.3	При посылке константы в слово	5.6		
		17.7	При посылке из одного слова в другое			
		45.4	При посылке из DM в DM			
22	MVN	16.4	При посылке константы в слово	5.6		
		17.5	При посылке из одного слова в другое			
		45.7	При посылке из DM в DM			
23	BIN	31.6	При преобразовании слова в слово	5.6		
		45.7	При преобразовании из DM в DM			
24	BCD	29.5	При преобразовании слова в слово	5.6		
		57.3	При преобразовании из DM в DM			
25	ASL	17.3	При сдвиге слова	5.5		
		31.3	При сдвиге DM			
26	ASR	16.9	При сдвиге слова	5.5		
		31.1	При сдвиге DM			
27	ROL	14.5	При сдвиге слова	5.5		
		28.5	При сдвиге DM			
28	ROR	14.5	При сдвиге слова	5.5		
		28.5	При сдвиге DM			
29	COM	18.1	При инвертировании слова	5.5		
		32.1	При инвертировании DM			

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
30	ADD	29.5	Константа + слово слово	5.6
		30.9	Слово + слово слово	
		72.7	DM + DM DM	
31	SUB	29.3	Константа - слово слово	5.6
		30.5	Слово - слово слово	
		72.5	DM - DM DM	
32	MUL	49.1	Константа x слово слово	5.6
		50.5	Слово x слово слово	
		95.1	DM x DM DM	
33	DIV	47.7	Константа слово слово	5.6
		50.9	Слово слово слово	
		94.3	DM DM DM	
34	ANDW	27.1	Константа слово слово	5.6
		28.8	Слово слово слово	
		70.7	DM DM DM	
35	ORW	27.1	Константа V слово слово	5.6
		28.8	Слово V слово слово	
		70.7	DM V DM DM	
36	XORW	27.1	Константа слово слово	5.6
		28.8	Слово слово слово	
		70.7	DM DM DM	
37	XNRW	27.0	Константа слово слово	5.6
		28.6	Слово слово слово	
		70.5	DM DM DM	
38	INC	17.9	Инкремент слова	5.5
		31.9	Инкремент DM	
39	DEC	18.3	Декремент слова	5.5
		32.3	Декремент DM	
40	STC	6.3	Любое	5.5
41	CLC	6.3		5.5
46	MSG	21.5	Сообщение в словах	5.5
		35.7	Сообщение DM	
50	ADB	30.5	Константа + слово слово	5.6
		32.1	Слово + слово слово	
		73.9	DM + DM DM	
51	SBB	30.9	Константа - слово слово	5.6
		32.7	Слово - слово слово	
		74.5	DM - DM DM	
52	MLB	34.7	Константа x слово слово	5.6
		36.3	Слово x слово слово	
		80.7	DM x DM DM	
53	DVB	35.1	Константа слово слово	5.6
		36.7	Слово слово слово	
		81.1	DM DM DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
54	ADDL	48.9	Слово + слово слово	5.6
		94.7	DM + DM DM	
55	SUBL	48.9	Слово - слово слово	5.6
		94.7	DM - DM DM	
56	MULL	138.7	Слово x слово слово	5.6
		184.3	DM x DM DM	
57	DIVL	136.7	Слово слово слово	5.6
		181.3	DM DM DM	
60	CMPL	30.4	Сравнение слов	5.6
		60.8	Сравнение DM	
61	INI	112.0	Пуск сравнения с помощью слова	5.6
		126.0	Пуск сравнения с помощью DM	
		48.0	Останов сравнения с помощью слова	
		48.0	Останов сравнения с помощью DM	
		120.0	Изменение текущего значение с помощью слова	
		128.0	Изменение текущего значение с помощью DM	
		46.0	Останов выдачи импульсов с помощью слова	
		60.0	Останов выдачи импульсов с помощью DM	
62	PRV	62.2	Задание выхода с помощью слова	5.6
		78.0	Задание выхода с помощью DM	
63	CTBL	106.3	Таблица для сравнения по значению с 1 значением (слово) и пуск.	5.6
		120.3	Таблица для сравнения по значению с 1 значением (DM) и пуск.	
		775.5	Таблица для сравнения по значению с 16 значениями (слово) и пуск.	
		799.5	Таблица для сравнения по значению с 16 значениями (DM) и пуск.	
		711.5	Таблица для сравнения по зоне (слово) и пуск.	
		722.5	Таблица для сравнения по зоне (DM) и пуск.	
		91.9	Таблица для сравнения по значению с 1 значением (слово).	
		106.3	Таблица для сравнения по значению с 1 значением (DM)	
		693.5	Таблица для сравнения по значению с 16 значениями (слово).	
		709.5	Таблица для сравнения по значению с 16 значениями (DM).	
		607.5	Таблица для сравнения по зоне (слово).	
		621.5	Таблица для сравнения по зоне (DM).	
67	BCNT	52.6	Подсчет битов в слове	5.6
		4.08 мс	Подсчет битов в 6 656 словах используя DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
68	BCMP	79.6	Сравнение констант, результат - в слово	5.6
		80.8	Сравнение слов, результат - в слово	
		123.2	Сравнение DM , результат - в DM	
69	STIM	47.5	Пуск разового прерывания, задание - слово	5.6
		58.7	Пуск разового прерывания, задание - DM	
		47.9	Пуск прерывания по расписанию, задание - слово	
		59.1	Пуск прерывания по расписанию, задание - DM	
		33.5	Чтение таймера, задание - слово	
		63.5	Чтение таймера, задание - DM	
		25.7	Останов таймера, задание - слово	
		54.1	Останов таймера, задание - DM	
70	XFER	45.5	При пересылке константы в слово	5.6
		47.1	При пересылке слова в слово	
		1.78 мс	При пересылке 1024 слов, используя DM	
71	BSET	28.1	При установке константы в 1 слово	5.6
		38.3	При пересылке константы в 10 слов	
		1.12 мс	При пересылке DM в 1024 слов	
73	XCHG	30.5	Слово слово	5.6
		59.1	DM DM	
74	SLD	25.9	Сдвиг 1 слова	5.6
		51.7	Сдвиг 10 слов	
		3.02 мс	Сдвиг 1024 слов, используя DM	
75	SRD	25.9	Сдвиг 1 слова	5.6
		51.7	Сдвиг 10 слов	
		3.02 мс	Сдвиг 1024 слов, используя DM	
76	MLPX	47.7	Декодирование слова в слово	5.6
		92.7	Декодирование DM DM	
77	DMPX	59.5	Кодирование слова в слово	5.6
		95.5	Кодирование DM DM	
78	SDEC	51.1	Декодирование слова в слово	5.6
		96.3	Декодирование DM DM	
80	DIST	39.1	При задании константы в слово + слово	5.6
		40.9	При задании слова в слово + слово	
		84.7	При задании DM в DM + DM	
		63.4	При задании константы в стек	
		65.0	При задании слова в стек	
		109.6	При задании DM в стек	

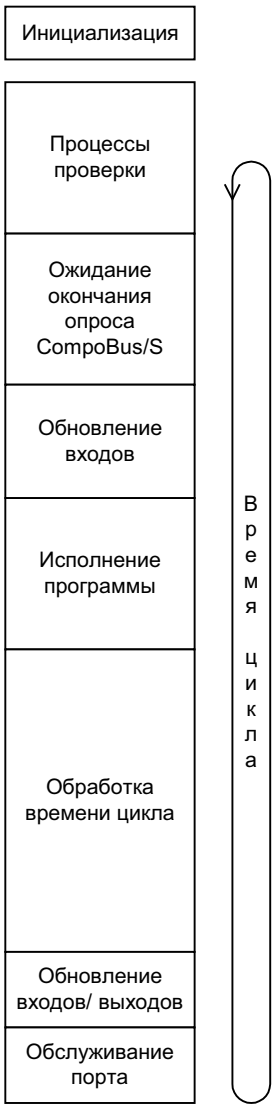
Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
81	COLL	42.6	При задании константа + слово в слово	5.6
		43.6	При задании слово + слово в слово	
		83.4	При задании DM + DM в DM	
		78.0	При задании слово + константа в стек FIFO	
		79.2	При задании слова + слово в стек FIFO	
		1.76 мс	При задании DM + DM в стек FIFO посредством DM	
		66.8	При задании слово + константа в стек LIFO	
		68.0	При задании слова + слово в стек LIFO	
		112.0	При задании DM + DM в стек LIFO посредством DM	
82	MOVB	32.5	При пересылке константы в слово	5.6
		37.5	При пересылке слова в слово	
		79.1	При пересылке DM в DM	
83	MOVD	28.3	При пересылке константы в слово	5.6
		33.3	При пересылке слова в слово	
		75.5	При пересылке DM в DM	
84	SFTR	39.3	Сдвиг 1 слова	5.6
		52.9	Сдвиг 10 слов	
		1420	Сдвиг 1024 слов, используя DM	
85	TCMP	57.7	Сравнение константы с таблицей, заданной словами	5.6
		58.9	Сравнение слова с таблицей, заданной словами	
		101.9	Сравнение DM с таблицей, заданной DM	
86	ASC	56.7	Слово слово	5.6
		103.9	DM DM	
89	INT	32.3	Задать маску с помощью слова	5.6
		46.3	Задать маску с помощью DM	
		29.1	Очистить прерывания с помощью слова	
		43.1	Очистить прерывания с помощью DM	
		27.3	Чтение состояния маски с помощью слова	
		41.5	Чтение состояния маски с помощью DM	
		29.7	Изменение задания счетчика с помощью слова	
		43.7	Изменение задания счетчика с помощью DM	
		15.3	Маскировать все прерывания с помощью слова	
		15.9	Маскировать все прерывания с помощью DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
		15.9	Очистить все прерывания с помощью слова	
91	SBS	36.6	Любое	5.5
92	SBN	1.7		1.7
93	RET	15.0		2.5
97	IOFR	40.0	Обновление IR 000	6.0
		142.6	Обновление одного слова входа	
		135.4	Обновление одного слова выхода	
99	MCRO	74.0	С операндами входа/выхода, заданными словом	5.6
		116.4	С операндами входа/выхода, заданными DM	

7.3 Время цикла SRM1 и время реакции входов/выходов

7.3.1 Цикл SRM1

На следующей диаграмме показан общий алгоритм отработки цикла SRM1.



Замечание 1. Время цикла можно прочитать с помощью периферийных устройств.

2. Максимальное время цикла и текущее хранится в AR14 и AR 15.

3. Изменения условий при отработке программы вызывают изменения времен цикла, так что расчетные и текущие значения времени цикла не всегда будут совпадать.

7.3.2 Время цикла SRM1

В таблице приведены процессы, происходящие в одном цикле SRM1, и соответствующее время обработки.

Процесс	Содержание	Требуемое время
Общая диспетчеризация	Установка контрольного таймера, проверка UM, обновление битов, отведенных под новые функции и т.д.	0.18 мс

Процесс	Содержание	Требуемое время
Ожидание окончания CompoBus/S	Ожидание окончания обработки CompoBus/S	Время реакции связи CompoBus/S - время общей диспетчеризации - время обслуживания порта RS-232C - время обслуживания периферийного порта
Обновление входов	Чтение входной информации в биты входа.	0.02 мс
Исполнение программы.	Исполнение программы пользователя. См. 7-3-6	Общее время исполнения команд. (Зависит от содержания программы пользователя).
Расчет времени цикла	Задержка до заданного значения, если время минимального цикла задано в DM 6619. Расчет времени цикла.	Почти мгновенно, за исключением времени задержки.
Обновление выходов	Выдача выходной информации (результата исполнения программы) на выходные биты. Запуск связи CompoBus/S	0.05 мс
Обслуживание порта RS-232 C	Обслуживаются устройства, подключенные к порту S-232 C	Не более 5% от времени цикла, но всегда от 0.55 до 131 мс (задается в DM 6616).
Обслуживание периферийного порта	Обслуживаются устройства, подключенные к периферийному порту	Не более 5% от времени цикла, но всегда от 0.55 до 131 мс (задается в DM 6617).

Минимальное время цикла

В SRM1 связь CompoBus/S начинается по завершении обновления выходов. В результате, когда время общей диспетчеризации + время обслуживания порта RS-232 C + время обслуживания периферийного порта меньше времени реакции связи CompoBus/S, отработка останавливается в режиме ожидания, пока не завершится связь CompoBus/S.

Таким образом, минимальное время цикла - это времени реакции связи CompoBus/S + время исполнения программы + время обновления входов + время обновления выходов. Время реакции связи CompoBus/S зависит от максимального числа подключенных узлов:

Максимальное число узлов	Время реакции CompoBus/S
32	0.8 мс
16	0.5 мс

Время цикла и отработка программы

Влияние времени цикла на операции показано в таблице. Когда долгое время цикла влияет на отработку программы, либо сократите время цикла, либо улучшите время реакции с помощью программ прерываний.

Время цикла	Условия отработки программы
10 мс и более	Может неточно исполняться TIMH(15), когда применяются TC 004..TC 127 (Отработка TC 000..TC 003 будет нормальной).
20 мс и более	Программы, использующие тактовый бит 0.02 с (SR 25401), могут работать некорректно.
100 мс и более	Может неточно работать TIM. Может неточно исполняться программа, использующая тактовый бит 0.1 с (SR 25500). Генерируется признак ошибки ПРЕВЫШЕНИЕ ВРЕМЕНИ ЦИКЛА (A CYCLE TIME OVER) (включается в 1 SR 25309) (см. прим. 1).
120 мс и более	Превышено заданное значение контрольного времени FALS 9F. Генерируется признак системной ошибки (FALS 9F) и отработка программы прекращается. (см. прим. 2).
200 мс и более	Может неточно исполняться программа, использующая тактовый бит 0.2 с (SR 25501).

Замечание 1. Для запрещения ошибки ПРЕВЫШЕНИЕ ВРЕМЕНИ ЦИКЛА (A CYCLE TIME OVER) используется DM 6655.

2. Время контроля за циклом FALS 9F можно изменять параметром DM 6618.

Пример расчет времени цикла

Далее приведен пример расчета времени цикла.

Условия отработки следующие:

Программа пользователя: 500 команд (состоят только из команд LD и OUT).

Время цикла: переменное (минимум не задан)

Порт RS-232C: не используется

Макс. Число узлов: 32 (Время реакции CompoBus/S = 0.8 мс)

Периферийный порт: 0.7 мс

Среднее время обработки одной команды программы пользователя принимается 1.66 мкс. Распределение времени цикла показано в следующей таблице.

Процесс	**Способ подсчета**	**Периферийный порт используется**	**Периферийный порт не используется**
Общая диспетчеризация	Фиксировано	0.18 мс	0.18 мс
Ожидание окончания CompoBus/S	См. Предыдущую стр.	0	0.62 мс
Обновление входов	Фиксировано	0.02 мс	0.02 мс
Исполнение программы.	1.16 x 500 (мкс)	0.8 мс	0.8 мс
Расчет времени цикла	Пренебрегаем	0 мс	0 мс
Обновление выходов	0.01 x 1 + 0.005 x1 (мкс)	0.05 мс	0.05 мс
Обслуживание порта RS-232 C	Не требуется	0	0
Обслуживание периферийного порта	5% времени цикла	0.7 мс	0
Время цикла	(1)+(2)+(3)+...+(8)	1.75 мс	1.67 мс

Замечание 1. Время цикла можно прочитать с ПК с помощью периферийного устройства.

2. Максимальное и текущее время цикла хранится в AR 14 и AR 15.

3. Время цикла может варьироваться в зависимости от условий отработки и необязательно будет точно совпадать с вычисленным значением.

4. Когда используется периферийный порт, отсутствует время ожидания окончания CompoBus/S, поскольку оно всегда 0.

5. Время ожидания окончания CompoBus/S = 0.8 - 0.18 - 0 + 0.62 (Время реакции связи CompoBus/S - диспетчеризация - время обслуживания порта RS-232 C- время обслуживания периферийного порта.

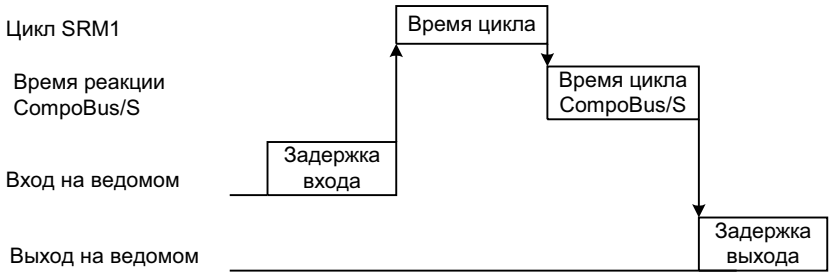
7.3.3 Время реакции вход/выход

Время реакции вход/выход - это время от получения входного сигнала (т.е. после включения в 1 входного бита), включающее проверку и обработку информации, до выдачи управляющего сигнала (т.е. выдачи результата отработки на выходной бит).

Связь CompoBus/S стартует после окончания обновления входов SRM1. Состояние 1/0 читается с клемм входов при обновлении входов и состояние 1/0 выдается на клеммы выходов при обновлении выходов. Соответственно время реакции вход/выход SRM1 варьируется в зависимости от времени цикла и состояния цикла CompoBus/S или синхронизации входов/выходов.

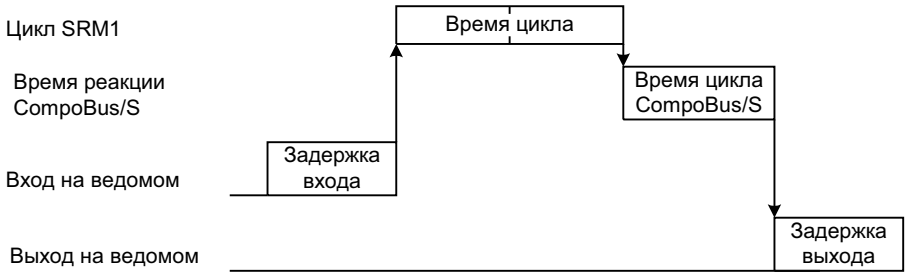
Минимальное время реакции вход/выход

Минимальное время реакции вход/выход = задержка включения входа в 1 + задержка включения выхода в 1 + время цикла CompoBus/S + время цикла SRM1.



Максимальное время реакции вход/выход

Максимальное время реакции вход/выход = задержка включения входа в 1 + задержка включения выхода в 1 + время цикла CompoBus/S + время цикла SRM1 x 2.



7.3.4 Время реакции вход/выход при связи 1:1

При связи 1+1 двух SRM1 время реакции - это время от получения входного сигнала на одном ПК до выдачи управляющего сигнала на выходной бит другого SRM1 посредством связи 1+1.

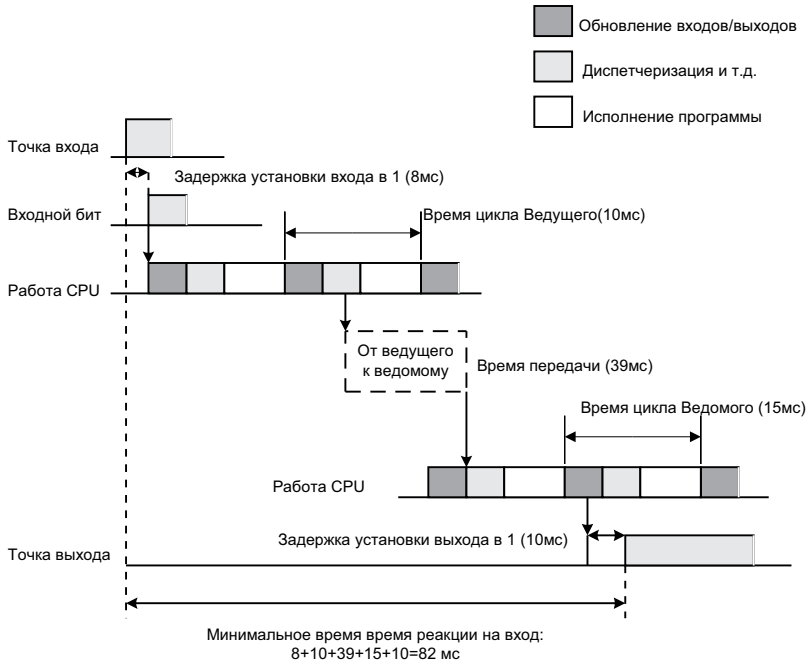
Минимальное время реакции вход/выход

SRM1 реагирует на входной сигнал быстрее всего при следующих обстоятельствах:

1, 2, 3,... 1. SRM1 получает входной сигнал сразу перед фазой цикла “обновление входов/выходов”.

2. Обслуживание связи на ведущем происходит сразу после начала передачи ведущий/ведомый.

3. Обслуживание связи на ведомом происходит сразу после завершения передачи.



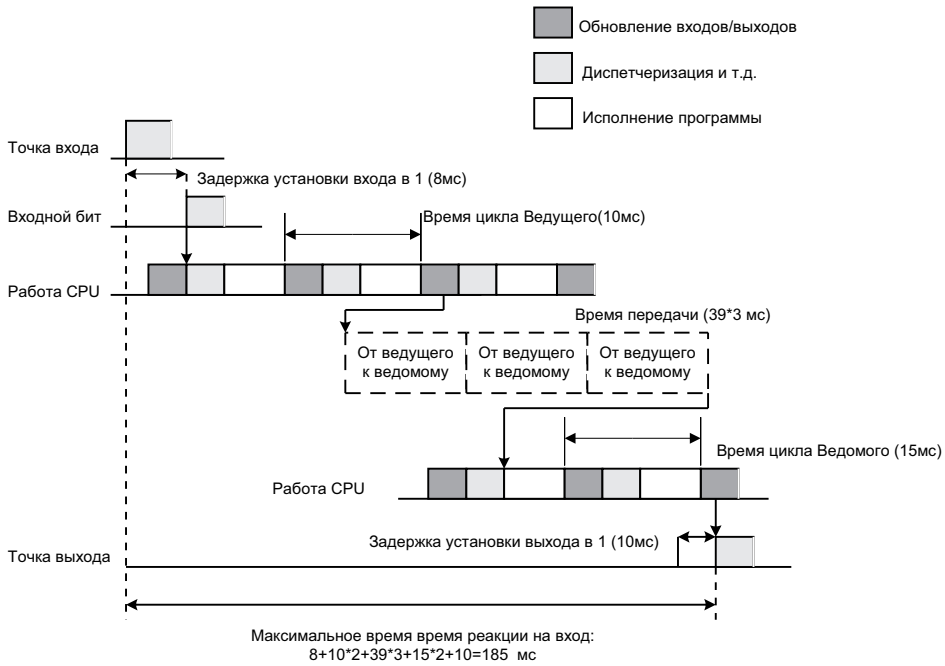
Максимальное время реакции вход/выход

SRM1 реагирует на входной сигнал быстрее всего при следующих обстоятельствах:

1, 2, 3,... 1. SRM1 получает входной сигнал сразу после фазы цикла “обновление входов/выходов”.

2. Обслуживание связи на ведущем не успевает на начало передачи ведущий/ведомый.

3. Передача завершается сразу после завершения обслуживания связи на ведомом.



7.3.5 Время обработки прерываний

В данной главе описано время обработки прерываний, от момента прерывания до вызова программы прерывания и от времени завершения отработки программы прерывания до возвращения в исходную точку. Данное описание относится к входным прерываниям и прерываниям по интервальному таймеру.

1, 2, 3,... 1. Источник прерываний
2. Ожидание завершения обработки маски прерывания
3. Переход к обработке прерывания
4. Маршрутизация прерывания (CPM1)
5. Возврат в исходную точку.

В таблице показаны времена от генерации сигнала прерывания до вызова программы прерывания и от времени завершения отработки программы прерывания до возвращения в исходную точку.

Параметр	Содержание	Время
Ожидание завершения обработки маски прерывания	Это время, во время которого прерывания ожидают завершения обработки. Данная ситуация имеет место при исполнении обработки маскирования. Подробности см. далее.	См. ниже
Переход на обработку прерывания	Это время переключения от текущей отработки на отработку прерывания.	15 мкс
Возврат	Время, которое требуется для исполнения RET(93) для возврата в программу, которая была прервана.	15 мкс

Обработка маски

Прерывания маскируются при отработке описанным далее образом. До завершения обработки все прерывания останутся замаскированными на указанное время.

Вызов и очистка признаков нефатальных ошибок:

Когда генерируется признак нефатальной ошибки и ее содержание регистрируется в SRM1 или во время очистки признака ошибки прерывания будут замаскированы максимум на 100 мкс до завершения обработки.

Редактирование в режиме он-лайн

Прерывания будут замаскированы максимум на 600 мкс (т.е. редактирование DM 6144..DM 6655) когда во время работы идет редактирование в режиме он-лайн. Кроме того, может возникнуть задержка максимум до 170 мкс.

7.3.6 Время исполнения команд SRM1

Базовые команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
				RSET	IL	JMP
	LD LD NOT	0.97				
	AND AND NOT	0.77				
	OR OR NOT	0.78				
-	AND LD OR LD	0.39	Любое	-		
-	OUT OUT NOT	2.2	Без прямых выходов или, если применяются прямые выходы, для операндов не IR 10000..IR 11515.	-		
-	SET	2.7	Прямые выходы	-		
-	RSET	2.8	Прямые выходы	-		
-	TIM	5.7	Константа для задания Т/Сч	9.3	9.1	3.5
			*DM для задания Т/Сч	17.4	17.2	3.5
-	CNT	6.6	Константа для задания Т/Сч	8.0	3.6	3.8
			DM для задания Т/Сч	16.3	3.6	3.8

Базовые команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)		
00	NOP	0.2	Любые			
01	END	4.8				
02	IL	2.5		1.4		
03	ILC	1.9		1.9		
04	JMP	2.2		1.3		
05	JME	2.5		2.5		
06	FAL	18.4		2.9		
07	FALS	3.6		2.9		
08	STEP	10.7		9.0		
09	SNXT	5.9		4.1		
10	SFT			**SHIFT**	**IL**	**JMP**
		14.5	С регистром сдвига в 1 слово	11.0	1.4	1.4
		21.0	С регистром сдвига в 10 слов	14.9	1.4	1.4
		49.1	С регистром сдвига в 100 слов	30.8	1.4	1.4
11	KEEP	3.0	Любые	3.4	1.6	1.7
12	CNTR	14.8	Константа для задания Т/Сч	9.1	6.6	6.5
		23.2	DM для задания Т/Сч			
13	DIFU	6.7	Любые	5.8	5.2	1.3
14	DIFD	6.4	Любые	5.8	5.7	1.3
15	TIMH	10.3	Нормальное выполнение, константа для задания SV	14.1	13.9	7.0
		10.9	Выполнение прерывания, константа для задания SV	15.6	15.4	8.5
		10.3	Нормальное выполнение, DM для задания SV	22.8	22.1	7.0
		10.9	Выполнение прерывания, DM для задания SV	23.9	23.6	8.5
16	WSFT	16.2	С регистром сдвига в 1 слово	2.9		
		23.0	С регистром сдвига в 10 слов			
		712.3	С регистром сдвига в 1024 слов, используя DM			
17	ASFT	18.6	Сдвиг в 1 слова	3.0		
		25.9	Сдвиг в 10 слов			
		865.7	Сдвига в 1 023 слов			
20	CMP	9.1	При сравнении константы и слова	3.0		
		9.9	При сравнении двух слов			
		25.6	При сравнении двух DM			
21	MOV	9.1	При посылке константы в слово	3.0		
		9.5	При посылке из одного слова в другое			
		24.9	При посылке из DM в DM			
22	MVN	9.3	При посылке константы в слово	3.0		
		9.8	При посылке из одного слова в другое			
		25.1	При посылке из DM в DM			
23	BIN	17.2	При преобразовании слова в слово	3.0		
		32.0	При преобразовании из DM в DM			

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
24	BCD	15.8	При преобразовании слова в слово	3.0
		30.6	При преобразовании из DM в DM	
25	ASL	9.9	При сдвиге слова	2.9
		17.3	При сдвиге DM	
26	ASR	9.7	При сдвиге слова	3.0
		17.2	При сдвиге DM	
27	ROL	8.5	При сдвиге слова	2.9
		16.1	При сдвиге DM	
28	ROR	8.5	При сдвиге слова	2.9
		16.1	При сдвиге DM	
29	COM	10.5	При инвертировании слова	3.0
		17.7	При инвертировании DM	
30	ADD	15.9	Константа + слово слово	3.1
		16.4	Слово + слово слово	
		39.5	DM + DM DM	
31	SUB	15.6	Константа - слово слово	3.0
		16.3	Слово - слово слово	
		38.6	DM - DM DM	
32	MUL	29.7	Константа x слово слово	3.0
		28.5	Слово x слово слово	
		51.6	DM x DM DM	
33	DIV	27.2	Константа слово слово	2.9
		28.5	Слово слово слово	
		53.1	DM DM DM	
34	ANDW	14.3	Константа слово слово	2.9
		15.2	Слово слово слово	
		37.3	DM DM DM	
35	ORW	14.3	Константа V слово слово	2.9
		15.2	Слово V слово слово	
		37.3	DM V DM DM	
36	XORW	14.3	Константа слово слово	2.9
		15.2	Слово слово слово	
		37.3	DM DM DM	
37	XNRW	14.3	Константа слово слово	2.9
		15.2	Слово слово слово	
		37.3	DM DM DM	
38	INC	9.9	Инкремент слова	2.9
		17.3	Инкремент DM	
39	DEC	10.2	Декремент слова	2.9
		17.4	Декремент DM	
40	STC	3.5	Любое	2.9
41	CLC	3.5		2.9
46	MSG	11.3	Сообщение в словах	2.9
		19.4	Сообщение DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
50	ADB	16.8	Константа + слово слово	3.0
		17.6	Слово + слово слово	
		39.9	DM + DM DM	
51	SBB	17.0	Константа - слово слово	3.0
		17.8	Слово - слово слово	
		40.2	DM - DM DM	
52	MLB	19.1	Константа x слово слово	3.0
		20.1	Слово x слово слово	
		43.5	DM x DM DM	
53	DVB	19.5	Константа слово слово	3.0
		20.4	Слово слово слово	
		43.7	DM DM DM	
54	ADDL	26.7	Слово + слово слово	3.0
		49.9	DM + DM DM	
55	SUBL	26.8	Слово - слово слово	3.0
		49.9	DM - DM DM	
56	MULL	81.4	Слово x слово слово	3.0
		106.2	DM x DM DM	
57	DIVL	76.9	Слово слово слово	3.0
		101.8	DM DM DM	
60	CMPL	16.9	Сравнение слов	2.9
		32.9	Сравнение DM	
67	BCNT	26.6	Подсчет битов в слове	3.0
		2.29	Подсчет битов в 6 656 словах используя DM	
68	BCMP	41.4	Сравнение констант, результат - в слово	3.0
		41.9	Сравнение слов, результат - в слово	
		64.5	Сравнение DM , результат - в DM	
69	STIM	34.7	Пуск разового прерывания, задание - слово	3.0
		49.5	Пуск разового прерывания, задание - DM	
		35.3	Пуск прерывания по расписанию, задание - слово	
		50.0	Пуск прерывания по расписанию, задание - DM	
		33.9	Чтение таймера, задание - слово	
		49.5	Чтение таймера, задание - DM	
		11.4	Останов таймера, задание - слово	
70	XFER	22.9	При пересылке константы в слово	3.0
		24.0	При пересылке слова в слово	
		902.0	При пересылке 1024 слов, используя DM	
71	BSET	15.2	При установке константы в 1 слово	3.0
		15.7	При пересылке константы в 10 слов	
		565.2	При пересылке DM в 1024 слов	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
73	XCHG	16.2	Слово слово	3.1
		31.5	DM DM	
74	SLD	13.6	Сдвиг 1 слова	3.0
		26.7	Сдвиг 10 слов	
		1540.0	Сдвиг 1024 слов, используя DM	
75	SRD	13.6	Сдвиг 1 слова	3.0
		26.6	Сдвиг 10 слов	
		1540.0	Сдвиг 1024 слов, используя DM	
76	MLPX	25.5	Декодирование слова в слово	3.0
		48.9	Декодирование DM DM	
77	DMPX	35.1	Кодирование слова в слово	3.0
		58.1	Кодирование DM DM	
78	SDEC	26.8	Декодирование слова в слово	2.9
		49.9	Декодирование DM DM	
80	DIST	21.3	При задании константы в слово + слово	3.0
		21.9	При задании слова в слово + слово	
		45.7	При задании DM в DM + DM	
		34.3	При задании константы в стек	
		35.3	При задании слова в стек	
		59.3	При задании DM в стек	
81	COLL	21.4	При задании константа + слово в слово	3.0
		21.8	При задании слово + слово в слово	
		44.9	При задании DM + DM в DM	
		34.0	При задании слово + константа в стек FIFO	
		33.9	При задании слова + слово в стек FIFO	
		892.0	При задании DM + DM в стек FIFO посредством DM	
		35.4	При задании слово + константа в стек LIFO	
		36.1	При задании слова + слово в стек LIFO	
		60.5	При задании DM + DM в стек LIFO посредством DM	
82	MOVB	18.2	При пересылке константы в слово	3.0
		19.0	При пересылке слова в слово	
		42.1	При пересылке DM в DM	
83	MOVD	16.3	При пересылке константы в слово	2.9
		17.6	При пересылке слова в слово	
		39.9	При пересылке DM в DM	
84	SFTR	21.0	Сдвиг 1 слова	3.0
		26.9	Сдвиг 10 слов	
		718.5	Сдвиг 1024 слов, используя DM	

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени, внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
85	TCMP	30.0	Сравнение константы с таблицей, заданной словами	3.0
		30.7	Сравнение слова с таблицей, заданной словами	
		53.1	Сравнение DM с таблицей, заданной DM	
86	ASC	30.0	Слово слово	3.0
		53.7	DM DM	
91	SBS	13.2	Любое	3.0
92	SBN	-		1.3
93	RET	7.8		1.3
99	MCRO	26.8	С операндами входа/выхода, заданными словом	3.0
		43.5	С операндами входа/выхода, заданными DM	

Дополнительные команды

Код	Мнемоника	Время исполнения при условии 1 (мкс)	Условия (Вверху: для мин. Времени ,внизу: для макс. времени)	Время исполнения при условии 0 (мкс)
-	HEX	51.2	DM в DM	3.0
		73.5	*DM в *DM	
-	FCS	23.4	Проверка 1 слова и передача в слово	3.0
		643.7	Проверка 999 слов и передача в *DM	
-	STUP	51.2	При пересылке константы в слово	3.0
		58.2	При пересылке слова в слово	

8. Поиск неисправностей

В данной главе описаны диагностика и исправление ошибок аппаратуры и программы, которые могут произойти во время работы ПК.

8.1 Введение

Ошибки ПК можно разделить на 4 категории:

1. Ошибки при вводе программы

   Данные ошибки появляются при вводе программы или подготовке ПК к работе.

2. Ошибки программирования

   Данные ошибки появляются при контроле программы операцией КОНТРОЛЬ ПРОГРАММЫ.

3. Ошибки, определяемые пользователем.

   Имеется 3 команды, которые пользователь может применить для создания собственных признаков ошибок или сообщений. Данные команды выполняются, когда при работе программы сложатся условия, указанные пользователем.

4. Ошибки при работе

   Данные ошибки появляются после запуска программы.

   a. Нефатальные ошибки

      При появлении одной или нескольких подобных ошибок работа ПК и отработка программы продолжается.

      b. Фатальные ошибки

      При появлении любой такой ошибки работа ПК и отработка программы прекращается, все выходы выключаются.

Индикаторы ПК будут сигнализировать об этом при появлении ошибок. Если к ПК подключен программатор или управляющий компьютер, на них индикатируются сообщение или код. Код ошибок также содержится в SR 25300..25307.

Для последних происшедших ошибок их тип и время появления заносится в область протокола ошибок. Подробности см. 8-6.

Для поиска ошибок можно также использовать флаги и другую информацию в зонах SR и AR. Подробности см. Главу 3, Области памяти.

Замечание Кроме вышеописанных ошибок могут возникнуть ошибки связи, если ПК является частью системы HOST LINK. Подробности см. 8-7.

8.2 Неисправности при работе с программатором

При работе ПК на программаторе могут появиться следующие сообщения об ошибках. Исправьте неисправности, следуя указаниям таблицы, и продолжите работу. Звездочка (*) в графе 1 таблицы находится в тех местах, где при реальной индикации будут находится числовые данные, обычно адрес..

Для работы с неисправностями, которые могут появиться при работе с SSS или устройством доступа к данным, см. инструкции:

- Инструкция по работе с программой поддержки SYSMAC SSS
- Инструкция по работе с устройством доступа к данным.

Сообщение	**Значение и действия при сообщении**
REPL ROM	Была попытка записи в память, защищенную от записи. В CQM1 установите переключатель защиты от записи в положение OFF (секция 1 переключателя DIP на ЦУ). В CPM1 установите биты 00..03 DM 6602 в 0.
PROG OVER	Команда в последнем адресе памяти не NOP (00). Сотрите ненужные команды в хвосте программы.
ADDR OVER	Задан адрес, превышающий наибольший в памяти программ. Введите меньший адрес.
SETDATA ERR	Был ввод FALS 00, а 00 не может быть введен. Снова введите данные.

Сообщение	Значение 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.

8.3 Ошибки программирования

Данные ошибки в синтаксисе программ обнаруживаются при контроле программы операцией “Контроль Программы”.

Имеется 3 уровня проверки программ. Необходимые уровень должен быть задан для указания типа ошибок, которые будут искаться. В следующей 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.

Ошибки уровня A

Сообщение	Значение сообщения и порядок действий
?????	Программа повреждена, появился несуществующий код. Введите программу снова.
CIRCUIT ERR	Число логических блоков и команд логических блоков не совпадает, т.е. LD или LD NOT использованы для начала логического блока, результаты которого не используются никакими другими командами, либо применена блоковая команда, для которой нет требуемого числа логических блоков. Проверьте программу.
OPERAND ERR	Константа, введенная для команды, лежит вне допустимых значений. Измените константу, чтобы она лежала в допустимой зоне
NO END INSTR	В программе отсутствует команда END (01). Запишите END (01) в последний адрес программы
LOCN ERR	Команда в неправильном месте программы. Изучите, требования команды и скорректируйте программу.
JME UNDEFD	Команда JME (04) отсутствует для команды JMP (05). Скорректируйте номер перехода или вставьте положенную команду JME (04).
DUPL	Дважды использован один и тот же номер перехода или подпрограммы. Скорректируйте программу, чтобы один номер использовался только для одной из них.
SBN UNDEFD	Команда SBS (91) запрограммирована для подпрограммы, с несуществующим номером. Скорректируйте номер подпрограммы или запрограммируйте требуемую подпрограмму.
STEP ERR	Некорректно использованы STEP (08) с номером секции и STEP (08) без номера секции. Проверьте требования команды и скорректируйте программу.

Ошибки уровня B

Сообщение	Значение сообщения и порядок действий
IL-ILC ERR	IL (02) и ILC (03) не использованы в паре. Скорректируйте программу, чтобы у каждой IL(02) был свой ILC (03). Хотя это сообщение об ошибке появится, если более одной IL (02) используется с одной IL (03), программа выполнится как написано. Перед отработкой убедитесь, что Ваша программа написана в соответствии с замыслом.
JMP-JME ERR	JMP (04) и JME (05) не использованы в паре. Перед отработкой убедитесь, что Ваша программа написана в соответствии замыслом.

SBN-RET ERR	Если индикатируемый адрес - адрес SBN (92), 2 разные подпрограммы определены одним именем. Измените один из номеров или удалите одну из подпрограмм. Если индикатируемый адрес - адрес RET (93), данная команда использована неправильно. Изучите требования команды RET и скорректируйте программу.

Ошибки уровня C

Сообщение	Значение сообщения и порядок действий
COIL DUPL	Один и тот же бит управляется (т.е. включается и/или выключается) более чем одной командой. (напр. OUT, OUT NOT, DIFU (13), DIFD (14), KEEP (11), SFT (10). Хотя это разрешено для некоторых команд, просмотрите требования команды и убедитесь, что программа написана правильно, или перепишите программу, чтобы каждый бит управлялся одной командой.
JMP UNDEFD	JME(05) был использован без JMP (04) с таким номером. Добавьте JMP (04) с таким же номером или удалите JME (05), который не используется.
SBS UNDEFD	Существует подпрограмма, которая не вызывается SBS(91). Запрограммируйте вызов подпрограммы в нужном месте или удалите подпрограмму, которая не используется.

Внимание! Дополнительные команды (приписанные к функциональным кодам 17,18,19, 47,48,60..69, 87, 88 и 89) не проверяются при контроле. Контроль также не проверяет DM 1024..DM 6143 для ПК, которые не поддерживают данную область DM (т.е. CQM1-CPU11-EV1 и CQM1-CPU21-E). Данные туда не будут записаны, даже если они заданы для записи, а при чтении этой области будут читаться 0000.

8.4 Ошибки, определяемые пользователем

Имеются 3 команды, с помощью которых пользователь может задать свои собственные ошибки или сообщения. Данные команды применяются для посылки на программатор, подключенный к ПК, сообщений о причине фатальной или нефатальной ошибок.

Сообщение - MSG (46) (Messa e)

Команда MSG(46) применяется для индикации сообщений на программаторе. Сообщения из максимум 16 знаков индикатируется, когда условие исполнения команды = 1. Подробности см. 5-26-2.

Нефатальная ошибка - FAL (06)

FAL(06) команда, вызывающая нефатальную ошибку. Подробности см. 5-13. При исполнении FAL (06) производятся следующие действия:

1, 2, 3,... 1. Будет мигать индикатор ERR/ALM на ЦУ. Работа ПК будет продолжаться.

2. Номер данной FAL (две десятичных цифры от 01 до 99) заносится в SR 25300..SR 25307.

3. Номер FAL заносится в область протокола ошибок. В CQM1 также записывается время появления признака ошибки, если используется кассета памяти с часами (RTC).

Номера FAL могут расставляться произвольно для индикации определенных условий. Одинаковые номера нельзя использовать для FAL и FALS.

Для стирания ошибки FAL устраните причину ошибки, выполните FAL 00, затем сотрите сообщение с помощью программатора.

Фатальная ошибка - FALS (07) (Se ere Failure Alarm)

FALS(07) - команда, вызывающая фатальную ошибку. Подробности см. 5-13. При исполнении команды FALS (07) производятся следующие действия:

1, 2, 3,... 1. Исполнение программы прерывается и выходы сбрасываются в 0.

2. Индикатор ERR/ALM на ЦУ горит.

3. Номер данной FALS из двух двоично-десятичных чисел (01-99) заносится в SR 25300..25307.

4. Номер FALS заносится в область протокола ошибок. В CQM1 также записывается время появления признака ошибки, если используется кассета памяти с часами (RTC).

Номера FALS могут расставляться произвольно для индикации определенных условий. Одинаковые номера нельзя использовать одновременно для FAL и FALS. Для удаления сообщения FALS переключите ПК в режим PROGRAM, устраните причину сообщения, сотрите признак с помощью программатора.

Поиск точки сбоя - FPD(-) (Failure Point Detect)

В CQM1 нефатальные ошибок и сообщения можно также вызывать командой FPD(-). Подробности см. 5-26-7.

8.5 Ошибки при работе

Существуют два типа ошибок при работе: нефатальные и фатальные. После нефатальных работа продолжается, после фатальных прекращается.

Внимание! Исследуйте все ошибки, и фатальные, и нефатальные. Устраните причину ошибки как можно быстрее и перезапустите ПК. Информацию об аппаратной части и работе с программатором см. Инструкцию по работе с CQM1 или Инструкцию по работе с CPM1. Операции SSS при работе с ошибками См. Инструкцию по работе с SSS.

8.5.1 Нефатальные ошибки

Работа ПК и отработка программы продолжается после появления одного или нескольких признаков таких ошибок. Хотя работа ПК продолжается, причину неисправности нужно выявить и устранить как можно быстрее.

При появлении такой неисправности индикаторы POWER (СЕТЬ) и RUN (РАБОТА) остаются включенными, индикатор ERR/ALM (Неисправность) мигает.

Нефатальные ошибки CQM1

Сообщение	FAL N	Значение сообщеия и порядок действий
SYS FAIL FAL ** (см. прим.)	01 - 99	В программе выполнилась команда FAL(06). Проверьте номер FAL для определения условий, вызвавших ее срабатывание, устраните причину и сотрите сообщение.
	9D	Произошел сбой во время обмена данных между ЦУ и кассетой памяти. Проверьте состояние флагов AR 1412..AR 1415 и исправьте неисправность: AR 1412 = 1: Включите режим PROGRAM, сотрите сообщение и возобновите передачу. AR 1413 = 1: Приемник защищен от записи Если приемник - ПК, выключите питание, установите секцию 1 переключателя DIP в положение OFF, сотрите сообщение и возобновите передачу. Если приемник - кассета памяти EEPROM, проверьте, есть ли питание, сотрите сообщение и возобновите передачу. Если адресат - кассета памяти EPROM, замените ее на кассету типа, позволяющего вести запись. AR 1414 = 1: У приемника недостаточен объем памяти. Проверьте программу источника и рассмотрите возможность использования либо другого ЦУ, либо другой кассеты памяти AR 1415 = 1: На кассете памяти отсутствует программа, или в программе есть ошибки. Проверьте кассету памяти.

Сообщение	FAL N	Значение сообщеия и порядок действий
	9C	Произошла ошибка в функции выдачи импульсов или в функции интерфейса абсолютного датчика вращения. Проверьте содержание AR 0408..AR 0415 (две двоично-десятичных цифры), и исправьте в соответствии с дальнейшими указаниями. (Данные коды ошибок приложимы только к CQM1-CPU43-E и CQM1-CPU44-E). 01,02: Произошел сбой в аппаратной части. Выключите питание, и снова включите. Если ошибка остается, замените ЦУ. 03: Некорректно заданы установочные параметры DM 6611, DM 6612, DM 6643, DM 6644. Исправьте параметры. 04: Работа CQM1 была прервана при выдаче импульсов. Поверьте, исправно ли устройство, принимающее выдаваемые импульсы.
SYS FAIL FAL ** (см. прим.)	9B	Обнаружена ошибка в установочных параметрах. Проверить флаги AR 2400..AR 2402 и исправьте ошибку согласно следующим указаниям: AR 2400 = 1: обнаружен некорректный установочный параметр (DM 6600..DM 6614) при включении питания. Исправьте установочные параметры в режиме PROGRAM и снова включите питание AR 2401 = 1: обнаружен некорректный установочный параметр (DM 6615..DM 6644) при переходе в режим RUN. Исправьте установочные параметры в режиме PROGRAM и снова включите режим RUN. AR 2402 = 1: обнаружен некорректный установочный параметр (DM 6645..DM 6655) во время работы. Исправьте установочные параметры и сотрите ошибку.
SCAN TIME OVER	F8	Время цикла превысило 100 мс. (SR 25309 будет 1). Указывает, что время программного цикла больше рекомендуемого. При возможности сократите цикл.
Аккумулятор неисправен	F7	Аккумулятор отсутствует, либо упало напряжение (SR 25308 будет 1). Проверьте аккумулятор и замените при необходимости. Проверьте установочный параметр (DM 6655), задан ли он на обнаружение сбоя работы аккумулятора.

Замечание ** может быть 01..99, 9D, 9C, или 9B.

Ошибки связи

Если произошла ошибка связи по периферийному порту или RS-232C, соответствующий индикатор (COM1 или COM2) прекратит мигать, Проверьте соединительные кабеля и программы связи на ПК и управляющем компьютере.

Сбросьте порты связи Битами Сброса Порта, SR 25208, SR 25209.

Запрет выходов

Когда горит индикатор OUT INH, то SR 25215 (бит запрета выхода) = 1, и все выходы ЦУ будут = 0. Если нет необходимости выключать все выходы, поставьте SR 25215 в состояние 0.

Нефатальные ошибки CPM1/CPM1A/SRM1

Сообщение	FAL N	Значение сообщеия и порядок действий
SYS FAIL FAL ** (см. прим.)	01 - 99	В программе выполнилась команда FAL(06). Проверьте номер FAL, определите причину, вызвавшую условие срабатывания, устраните причину и сотрите сообщение.
	9B	Обнаружена ошибка в установочных параметрах. Проверьте флаги AR 1300..AR 1302 и исправьте ошибку следующим образом: AR 1300 = 1: обнаружен некорректный установочный параметр (DM 6600..DM 6614) при включении питания. Исправьте установочные параметры в режиме PROGRAM и снова включите питание AR 1301 = 1: обнаружен некорректный установочный параметр (DM 6615..DM 6644) при переходе в режим RUN. Исправьте установочные параметры в режиме PROGRAM и снова включите режим RUN. AR 1302 = 1: обнаружен некорректный установочный параметр (DM 6645..DM 6655) во время работы. Исправьте установочные параметры и сотрите ошибку.
SCAN TIME OVER	F8	Время цикла превысило 100 мс. (SR 25309 будет = 1) Указывает, что время программного цикла больше рекомендуемого. При возможности сократите цикл. (CPM1 можно настроить так, чтобы эта ошибка не обнаруживалась).
Ошибки связи (Нет сообщения)	Нет	Если произошел сбой связи по периферийному порту, соответствующий индикатор COMM прекратит мигать. Проверьте соединительные кабеля и перезапуститесь. Проверьте, установлены ли флаги ошибок в AR 0812 =1.

Замечание ** может быть 01..99 или 9B.

8.5.2 Фатальные ошибки

Работа ПК и отработка программы прекратится и все выходы ПК выключаются в 0 при любой такой ошибке.

При прерывании питания все индикаторы на ЦУ будут выключены. При других фатальных ошибках будут гореть индикаторы POWER (ПИТАНИЕ) и ERR/ALM (АВАРИЯ). Индикатор RUN (РАБОТА) будет выключен.

Фатальные ошибки CQM1

Сообщение	FAL N	Значение сообщеия и порядок действий
Прерывание питания (нет сообщения)	нет	Было прерывание питание более чем 10 мс. Проверить напряжение и подвод питания. Снова включить питание.
MEMORY ERR (Ошибка памяти)	F1	AR 1611 = 1: Ошибка контрольной суммы в установочных параметрах (DM 6600..DM 6655). Проинициализируйте все установочные параметры и введите заново. AR 1612 = 1: Появилась ошибка контрольной суммы в программе, индикатируется некорректная команда. Проверьте программу и исправьте обнаруженные ошибки. AR 1613 = 1: Ошибка контрольной суммы в параметрах дополнительных команд. Проинициализируйте все установочные параметры дополнительных команд и введите их заново. AR 1614 = 1: Кассета памяти была установлена или снята при включенном питании. Выключите питание, вставьте кассету и снова включите питание. AR 1615 = 1: Содержимое кассеты памяти нельзя прочесть при запуске. Проверьте флаги AR 1412..AR 1415 для выяснения причины неисправности, исправьте неисправность и снова включите питание.
NO END INST	F0	В программе отсутствует команда END(01). Запишите в конце программы END(01).

Сообщение	FAL N	Значение сообщеия и порядок действий
I/O BUS ERROR	C0	Произошел сбой при передаче данных между ЦУ и блоком входов/выходов. С помощью флагов AR 2408..AR 2415 определите место неисправности, выключите питание, определите место неплотной стыковки блоков входов/выходов или оконечной крышки и снова включите питание.
I/O UNIT OVER	E1	Число входов/выходов на установленных блоках входа/выхода больше максимально допустимого. Выключите питание, перестройте систему, чтобы сократить число слов входа/выхода, и снова включите питание.
SYS FAIL FALS ** (см. прим.)	01..99	В программе выполнилась команда FALS (07). Проверьте номер FALS для определения условий, вызвавших сообщение, устраните причину и сотрите сообщение.
	9F	Время цикла превысило “Контрольное время цикла” (DM 6618), вызывающее FALS 9F. Проверьте время цикла и настройте параметр “Время контроля цикла”.

Замечание ** может быть 01-99, или 9F

Фатальные ошибки CPM1/CPM1A/SRM1

Сообщение	FAL N	Значение сообщеия и порядок действий
Прерывание питания (нет сообщения)	нет	Было прерывание питание более чем 10 мс. Проверить напряжение и подвод питания. Снова включить питание.
MEMORY ERR (Ошибка памяти)	F1	AR 1308 = 1: В программе пользователя существует незаданная битовая область. Проверьте программу и исправьте ошибку. AR 1309 = 1: Произошел сбой в памяти FLASH. Поскольку число записей в память FLASH превысило заданный уровень, замените ЦУ. AR 1301 = 1: Ошибка контрольной суммы в установочных параметрах (DM 6144..DM 6599). Проверьте и откорректируйте установочные параметры в области DM AR 1311 = 1: Ошибка контрольной суммы в установочных параметрах. Проинициализируйте все установочные параметры и введите снова. AR 1312 = 1: Ошибка контрольной суммы в программе. Проверьте программу и исправьте обнаруженные ошибки.
NO END INST	F0	В программе отсутствует команда END(01). Запишите в конце программы END(01).
I/O BUS ERROR	C0	Произошел сбой при передаче данных между ЦУ и блоком входов/выходов. Проверьте соединительный кабель к блоку входов/выходов.
I/O UNIT OVER	E1	Подключено слишком много блоков входов/выходов. Исправьте конфигурацию входов/выходов.
SYS FAIL FALS ** (см. прим.)	01 - 99	В программе выполнилась команда FALS (07). Проверьте номер FALS для определения условий, вызвавших сообщение, устраните причину и сотрите сообщение.
	9F	Время цикла превысило “Контрольное время цикла” (DM 6618), вызывающее FALS 9F. Проверьте время цикла и настройте параметр “Контрольное время цикла”.

Замечание ** может быть 01-99, или 9F

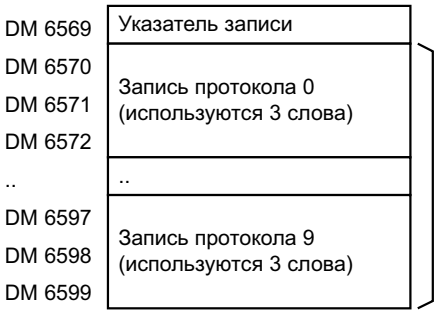
8.6 Протокол ошибок

Функция протокола ошибок регистрирует код любой фатальной и нефатальной ошибки, которые произошли в ПК. Наряду с кодом ошибки регистрируется также дата и время ошибки. Коды ошибок см. 8-5.

Область протокола ошибок CQM1

В CQM1 протокол ошибок хранится в DM 6569..DM 6599.

Если установлена кассета памяти без часов, дата и время будут установлены в 0.



Указывает, куда загружать следующую запись (0..A)

Каждая запись в протоколе имеет вид

	15 8	7 0
Первое слово	Тип ошибки	Код ошибки
Первое слово+1	Минута	Секунда
Первое слово+2	День	час

Тип ошибки:
00: нефатальная
01: фатальная

Каждый признак ошибки будет записан, даже если секция 1 переключателя DIP CQM1 включен в ON для защиты DM 6144..DM 6655.

Область протокола ошибок CPM1

В CQM1 протокол ошибок хранится в DM 1000..DM 1021.

Указывает, куда загружать следующую запись (0..6)

Каждая запись в протоколе имеет вид

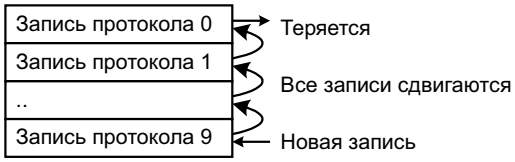
	15 8	7 0
Первое слово	Тип ошибки	Код ошибки
Первое слово+1	00	00
Первое слово+2	00	00

Тип ошибки:
00: нефатальная
01: фатальная

Методы сохранения протокола ошибок

Методы загрузки протокола признаков ошибок задаются в Установочных параметрах (DM 6655). Задайте какой-либо из методов.

1, 2, 3,... 1. Вы можете загружать самые последние 10 записей, убирая старые. Это достигается сдвигом записей, как указано на рисунке, чтобы самая последняя запись (0) терялась при появлении новой.



2. Вы можете загружать только 10 записей и игнорировать все, приходящие после этих 10.

3. Вы можете запретит запись и не будет записано ни одного признака ошибок.

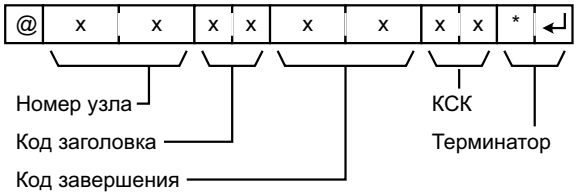
Параметры по умолчанию - первый метод. Подробности об установочных параметрах для протокола ошибок см. 1-2-10.

Очистка протокола ошибок

Для очистки протокола ошибок включите SR 25214 в 1 с периферийного устройства. (После очистки SR 25214 автоматически сбросится в 0).

8.7 Ошибки HOST LINK

Эти коды ошибок управляющий компьютер получает как код ответа (код окончания), когда команда, полученная ПК от управляющего компьютера, не может быть обработана. Формат кода ошибок показан на рисунке.



Код заголовка может варьироваться в зависимости от команды. Некоторые команды (сложные команды) содержат подзаголовок.

Код окончания	**Содержание**	**Вероятная причина**	**Методы исправления**
00	Нормальное завершение	-	-
01	Не выполнима в режиме RUN	Посланную команду нельзя выполнить в режиме ПК RUN.	Проверьте и установите соответствие команд и режима ПК.
02	Не выполнима в режиме MONITOR	Посланную команду нельзя выполнить в режиме ПК MONITOR.	
04	Превышение адреса	Превышен максимальный адрес программы пользователя	Проверьте программу.
0B	Не выполнима в режиме PROGRAM	Посланную команду нельзя выполнить в режиме ПК PROGRAM.	В настоящее время код не используется
13	Ошибка КСК (контрольной суммы кадра)	Ошибка КСК. Либо ошибка при подсчете КСК, либо воздействие помех.	Проверьте метод подсчета КСК. Если было влияние помехи, снова передайте команду.
14	Ошибка Формата	Ошибка в формате команды.	Проверьте формат и снова передайте команду.
15	Ошибка задания номера объекта	Область чтения или записи неверно задана.	Проверьте области и снова передайте команду.
16	Команда не поддерживается	Заданная команда не существует по указанному адресу.	Проверьте адреса и команды.
18	Ошибка длины кадра	Превышена максимальная длина кадра.	Разделите команду на несколько кадров.
19	Не выполнима	Объекты для чтения составной командой (QQ) не зарегистрированы.	Выполните QQ для регистрации объектов для чтения перед попыткой чтения блока.
23	Память пользователя защищена от записи	CQM1: секция 1 на DIP ЦУ установлена в 1. CPM1: Память защищена от записи в установочных параметрах.	CQM1: Установите секцию 1 на DIP ЦУ в 0. CPM1: измените задание DM 6602.

Код окончания	Содержание	Вероятная причина	Методы исправления
A3	Прервана из-за ошибки КСК при передаче данных	Ошибка возникает при исполнении команды, состоящей из нескольких кадров.*	Проверьте испорченные кадры, при необходимости исправьте и снова попытайтесь передать.
A4	Прервана из-за ошибки формата при передаче данных		
A5	Прервана из-за ошибки задания номера объекта при передаче данных		
A6	Прервана из-за ошибки длины кадра формат при передаче данных		
Другие	-	Влияние помех	Снова передайте команду

Замечание *Данные до этой точки уже записаны в соответствующую область ПК.

Прерывание питания

Следующие ответы могут быть получены от CQM1 при прерывании питания, включая мгновенные провалы,. При получении любого такого сообщения повторите команду.

Неопределенный ответ на команду

@00IC4A*↵

Отсутствие ответа

Если ответ не получен, то отмените последнюю команду и пошлите заново.

8.8 Алгоритмы поиска неисправностей

8.8.1 Алгоритмы для CPM1

См. Инструкция по работе CPM1, гл. 5.6 Алгоритмы поиска неисправностей.

8.8.1 Алгоритмы для CPM1A

См. Инструкция по работе CPM1A, гл. 5.6 Алгоритмы поиска неисправностей.

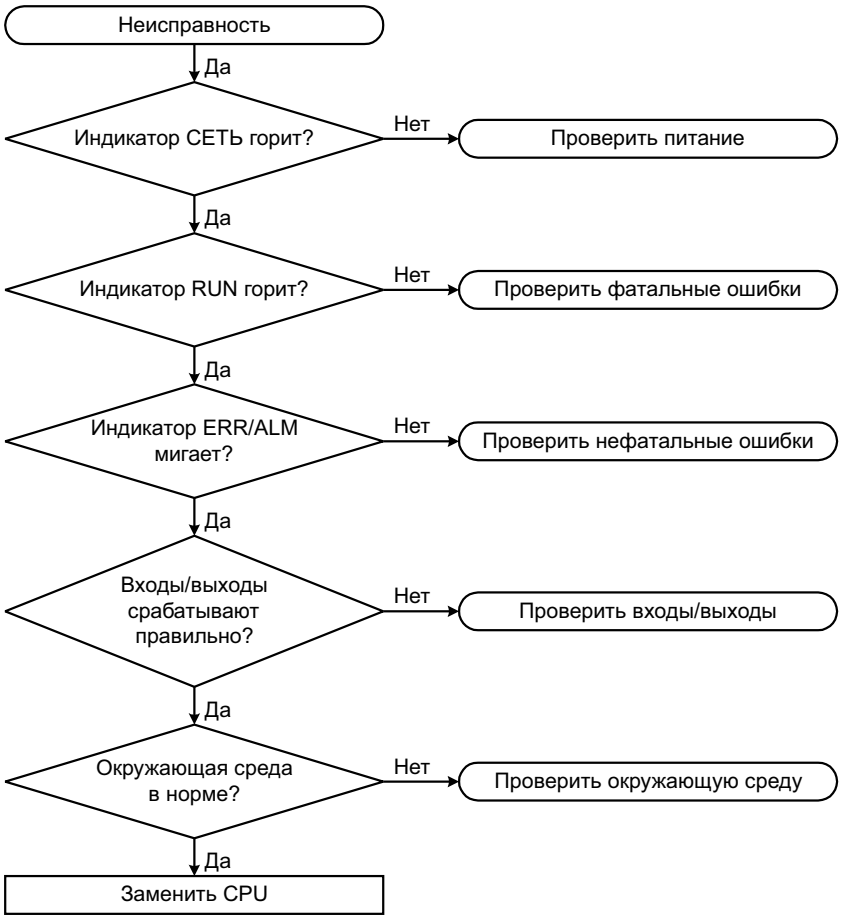
8.8.1 Алгоритмы для SRM1

См. Инструкция по работе SRM1, гл. 5.6 Алгоритмы поиска неисправностей.

8.8.2 Алгоритмы для CQM1

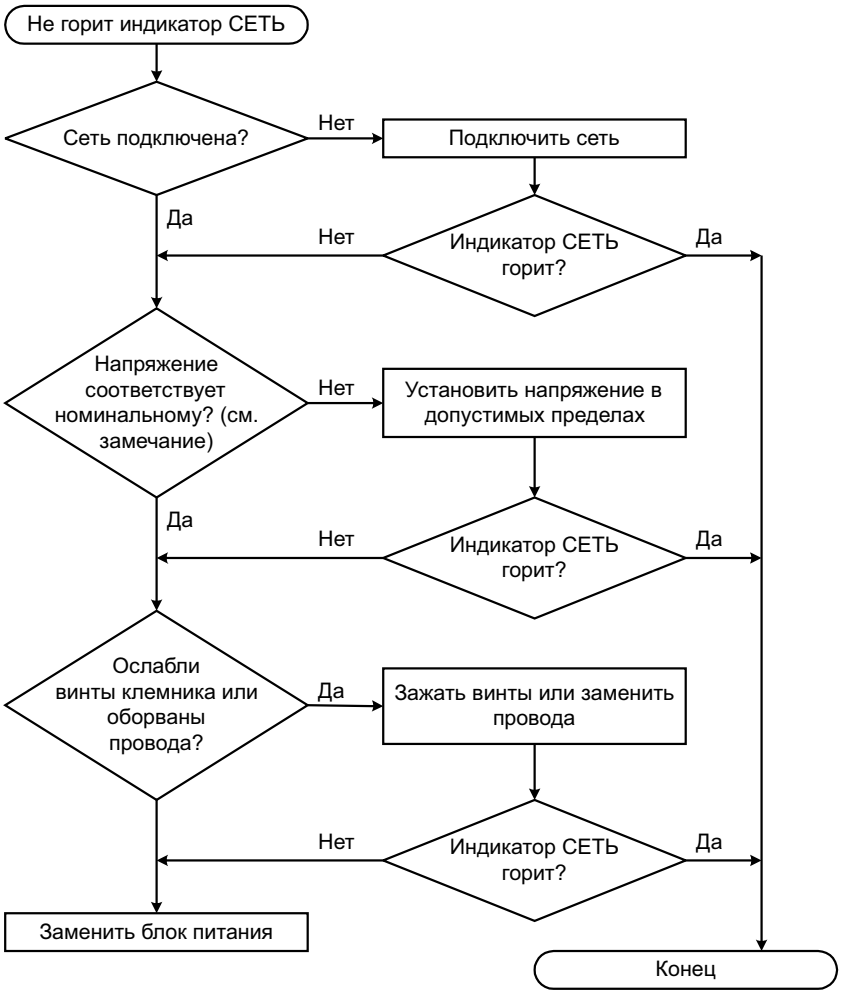
Для поиска неисправностей, произошедших при работе SQM1 пользуйтесь следующими алгоритмами.

Главный алгоритм поиска



Замечание Обязательно выключайте ПК перед заменой блоков, аккумуляторов, подключением проводов и кабелей.

Проверка питания



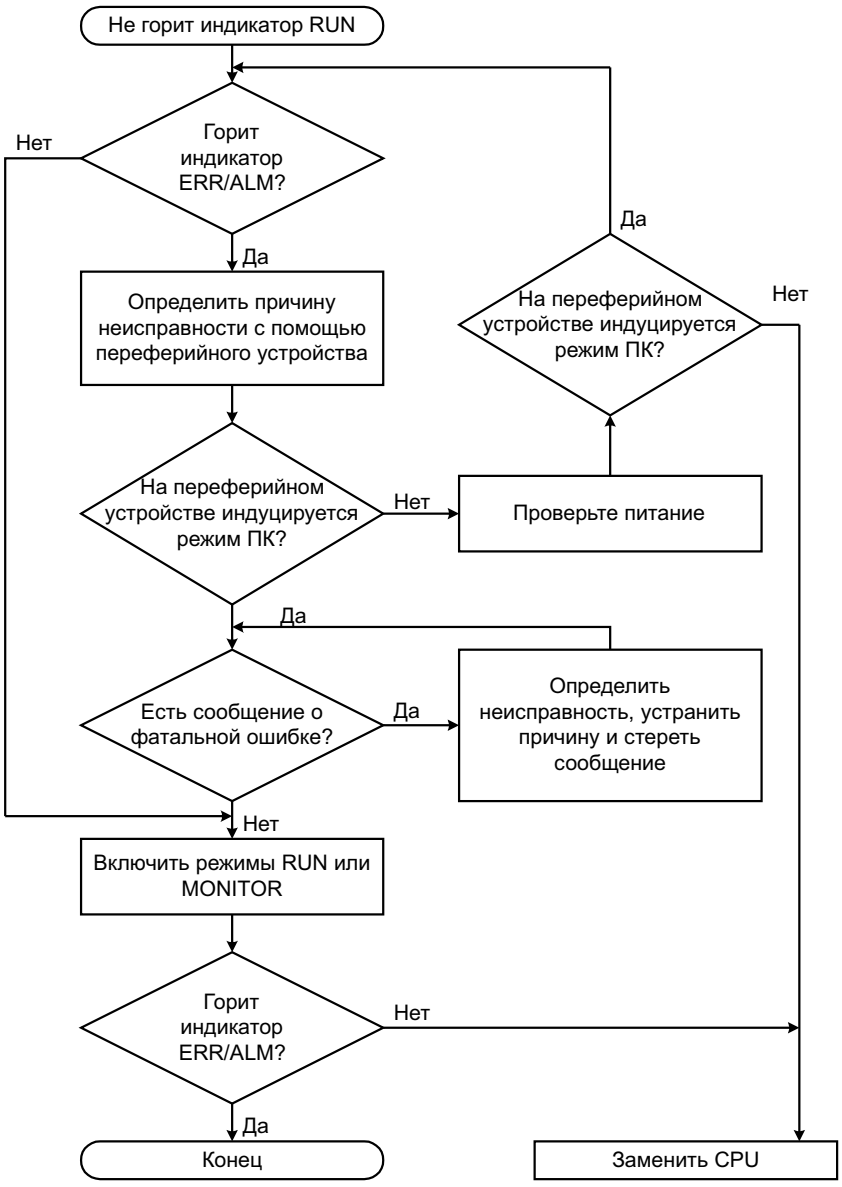
Замечание Допустимый диапазон напряжений сети:

CQM1-PA203/PA 206: 85 - 264 В пер. тока

CQM1-PD206: 20 - 28 В пост. Тока

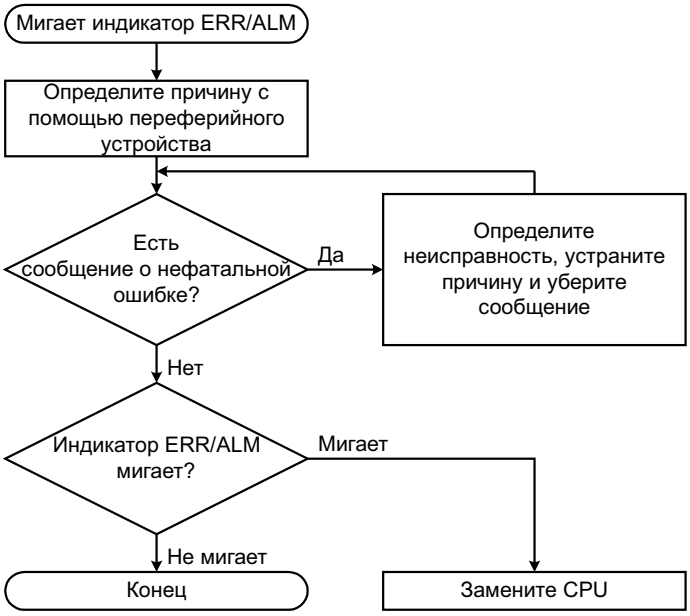
Проверка фатальных ошибок

Следующий алгоритм можно применять для поиска фатальных ошибок, когда горит индикатор СЕТЬ.



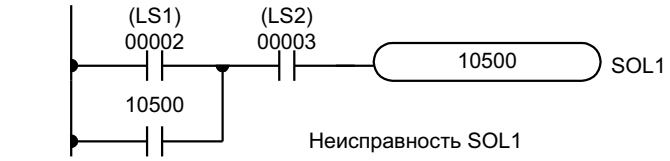
Проверка нефатальных ошибок

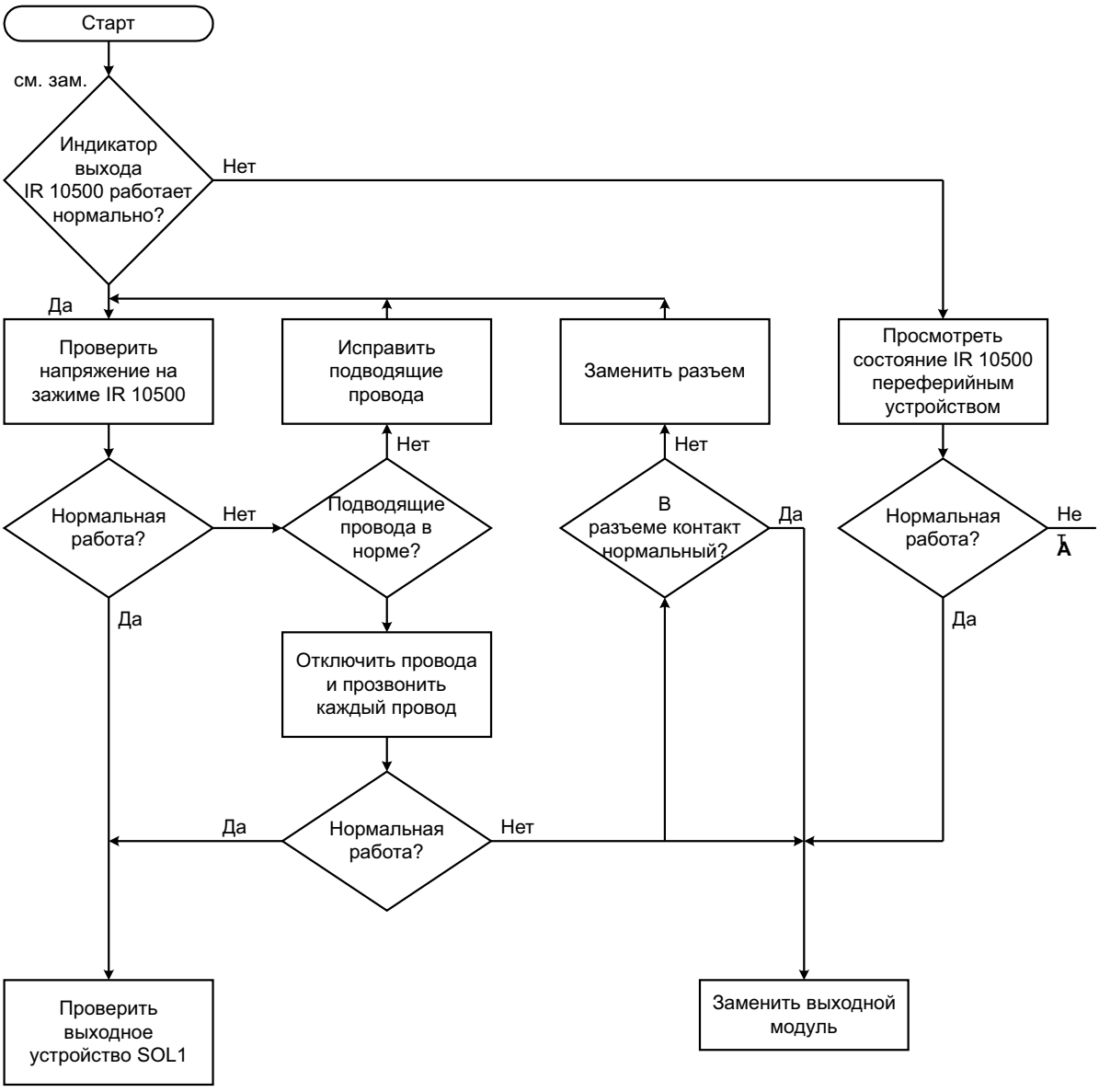
Хотя при нефатальных ошибках ПК продолжает работать, причину неисправности следует определить и устранить как можно быстрее для гарантии корректной работы. Для устранения причин нефатальных ошибок может потребоваться остановка ПК.



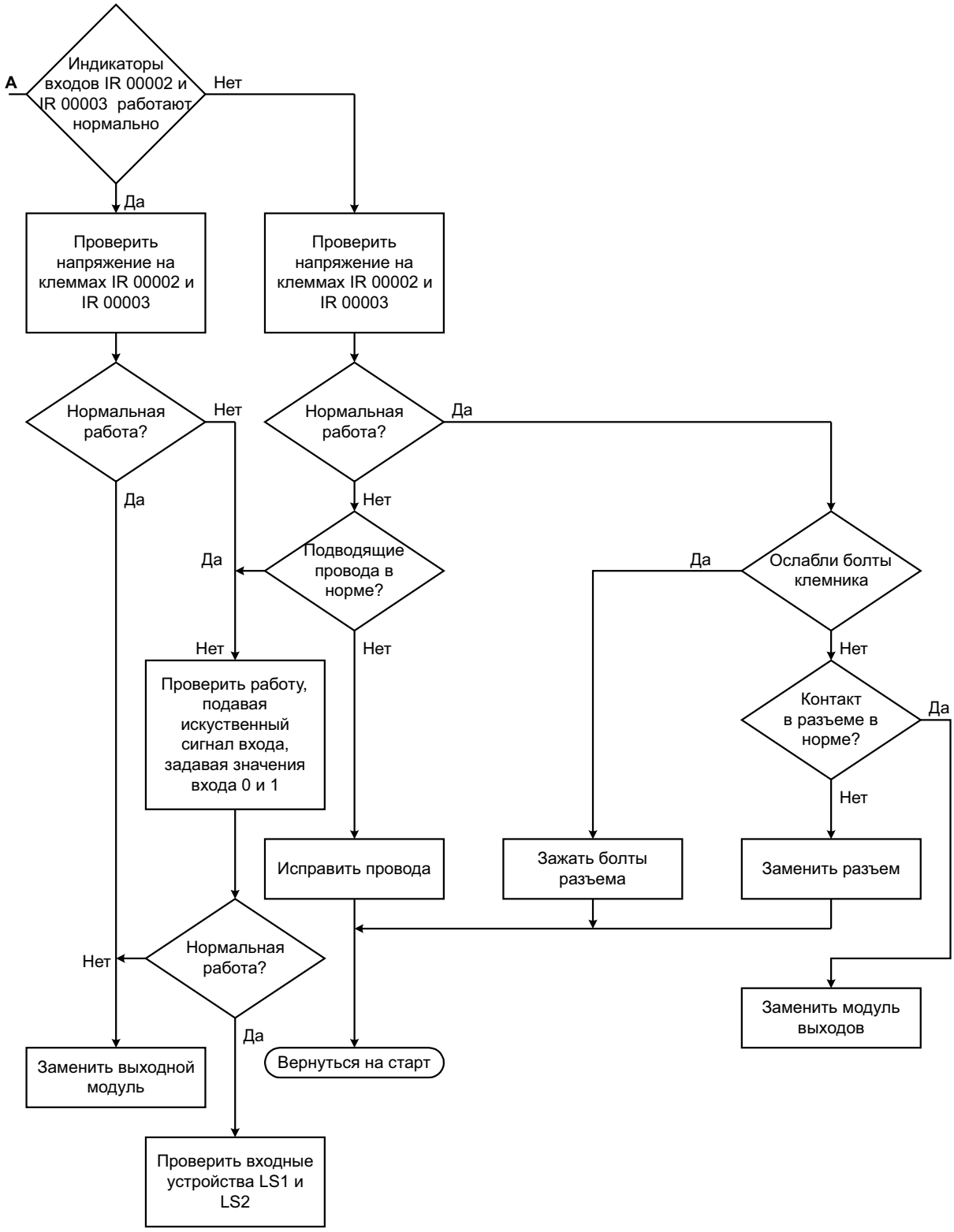
Проверка входов/выходов

Алгоритм проверки входов/выходов базируется на отработке секции релейно-контактной схемы:

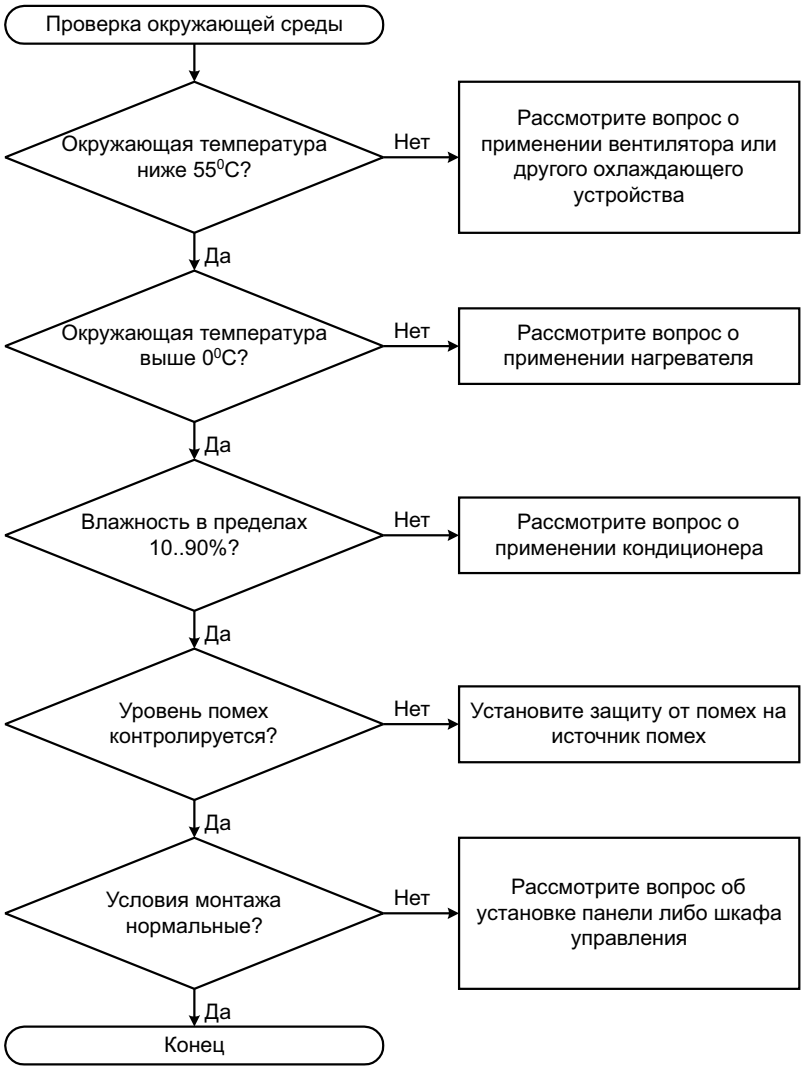




Замечание У CPM1 нет выхода IR 10500. Замените одним из IR 01000..IR 01915.


A
Индикаторы входов IR 00002 и IR 00003 работают нормально
Нет
Да
Проверить напряжение на клеммах IR 00002 и IR 00003
Проверить напряжение на клеммах IR 00002 и IR 00003
Нормальная работа?
Нет
Да
Нормальная работа?
Да
Нет
Подводящие провода в норме?
Да
Нет
Ослабли болты клемника
Да
Нет
Проверить работу, подавая искуственный сигнал входа, задавая значения входа 0 и 1
Нет
Контакт в разъеме в норме?
Да
Нет
Исправить провода
Зажать болты разъема
Заменить разъем
Нормальная работа?
Нет
Да
Заменить модуль выходов
Заменить выходной модуль
Вернуться на старт
Проверить входные устройства LS1 и LS2

Проверка окружающей среды



Приложение А

Команды программирования ЛД

Команды ПК вводятся либо нажатием соответствующей клавиши на программаторе, (например, LD, AND, OR, NOT), либо используя коды функций. 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.

Код	Мнемоника	Наименование	Функция	Пункт
-	AND	AND	Логическое И состояния указанного бита и текущего условия исполнения.	5.7.1
-	AND LD	AND LOAD	Логическое И результатов предыдущих блоков	5.7.1
-	AND NOT	AND NOT	Логическое И инверсного состояния указанного бита и текущего условия исполнения.	5.7.1
-	CNT	COUNTER	Декрементальный счетчик	5.15.2
-	LD	LOAD	Служит для начала командной линии с состояния указанного бита или для задания логического блока для использования с AND LD и OR LD.	5.7.1
-	LD NOT	LOAD NOT	Служит для начала командной линии с инверсного состояния указанного бита.	5.7.1
-	OR	OR	Логическое ИЛИ состояния указанного бита и текущего условия исполнения.	5.7.1
	OR LD	OR LOAD	Логическое ИЛИ результатов предыдущих блоков	5.7.1
-	OR NOT	OR NOT	Логическое ИЛИ инверсного состояния указанного бита и текущего условия исполнения.	5.7.1
-	OUT	OUTPUT	Включает в 1 битовый операнд при условии исполнения = 1.Включает в 0 битовый операнд при условии исполнения = 0.	5.8.1
-	OUT NOT	OUTPUT NOT	Включает в 0 битовый операнд при условии исполнения = 1.Включает в 1 битовый операнд при условии исполнения = 0.е. Операция инвертирования)	5.8.1
-	RSET	RESET	Выключает в 0 битовый операнд при условии исполнения = 1.Не влияет на состояние операнда при условии исполнения = 0.	5.8.2
-	SET	SET	Включает в 1 битовый операнд при условии исполнения = 1.Не влияет на состояние операнда при условии исполнения = 0.	5.8.2
-	TIM	TIMER	Операция задержки включения в 1 (декрементирующий таймер)	5.15.1
00	NOP	NO OPERATION	Ничего не выполняется и программа переходит к следующему адресу.	5.9
01	END	END	Требуется в конце программы	5.10
02	IL	INTERLOCK	Если условие исполнения = 0, то все выходы = 0, все текущие значения таймеров сбрасываются между IL(02) и следующим IL(03).	5.11
03	ILC	INTERLOCK CLEAR	Другие операции воспринимаются как NOP. Текущие значения счетчиков сохраняются.	5.11
04	JMP	JUMP	Если условие исполнения = 0, все команды между JMP(04) и	5.12
05	JME	JUMP END	Соответствующей JMP(05) игнорируются.	5.12
06	(@)FAL	FAILURE ALARM AND RESET	Генерирует признак нефатальной ошибки и выдает указанный номер FAL в программатор.	5.13
07	FALS	SEVERE FAILURE ALARM	Генерирует признак фатальной ошибки и выдает указанный номер FALS в программатор.	5.13

Код	Мнемоника	Наименование	Функция	Пункт
08	STEP	STEP DEFINE	При использовании с битом управления, назначает начало секции STEP и сбрасывает предыдущую секцию STEP. При использовании без N определяет окончание исполнение секции STEP.	5.14
09	SNXT	STEP START	Служит совместно с битом управления для указания окончания секции STEP и пуска следующей секции STEP.	5.14
10	SFT	SHIFT REGISTER	Создает битовый регистр сдвига.	5.16.1
11	KEEP	KEEP	Определяет бит - триггер, управляемый входами установки и сброса.	5.8.3
12	CNTR	REVERSIBLE COUNTER	Увеличивает или уменьшает текущее значение на 1, когда соответственно инкрементирующий и декрементирующий входы переключаются из 0 в 1.	5.15.3
13	DIFU	DIFFERENTIATE UP	Устанавливает в 1 указанный бит на один цикл при переднем фронте входного сигнала.	5.8.4
14	DIFD	DIFFERENTIATE DOWN	Устанавливает в 1 указанный бит на один цикл при заднем фронте входного сигнала.	5.8.4
15	TIMH	HIGH-SPEED TIMER	Высокоскоростной таймер с задержкой установки в 1 (декрементальный)	5.15.4
16	(@)WSFT	WORD SHIFT	Сдвиг данных между первым и последним словами блока, записывает нуль в первое слово.	5.16.2
17-19	Дополнительные команды			
20	CMP	COMPARE	Сравнивает содержание двух слов и выдает результат во флаги GR, EQ и LE.	5.18.1
21	(@)MOV	MOVE	Копирует данные (слово или константу) из источника в приемник.	5.17.1
22	(@)MVN	MOVE NOT	Инвертирует данные источника (слово или константу) и копирует в приемник.	5.17.2
23	(@)BIN	BCD-TO-BINARY	Преобразует 4-разрядное двоично-десятичное число источника в 16-битовое двоичное число и выдает преобразованное число в приемник.	5.19.1
24	(@)BCD	BINARY TO BCD	Преобразует двоичное число источника в двоично-десятичное и выдает преобразованное число в приемник.	5.19.2
25	(@)ASL	ARITHMETIC SHIFT LEFT	Сдвигает каждый бит одного слова на 1 бит влево, включая CY.	5.16.3
26	(@)ASR	ARITHMETIC SHIFT RIGHT	Сдвигает каждый бит одного слова на 1 бит вправо, включая CY.	5.16.4
27	(@)ROL	ROTATE LEFT	Осуществляет циклический сдвиг каждого бита одного слова на 1 бит влево, включая CY.	5.16.5
28	(@)ROR	ROTATE RIGHT	Осуществляет циклический сдвиг каждого бита одного слова на 1 бит влево, включая CY.	5.16.6
29	(@)COM	COMPLEMENT	Инвертирует состояние битов одного слова данных	5.23.1
30	(@)ADD	BCD ADD	Складывает два 4-разрядных двоично-десятичных числа и содержимое CY и выдает результат в указанное слово результата.	5.20.3
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.	5.20.4
32	(@)MUL	BCD MULTIPLY	Перемножает два 4-разрядных двоично-десятичных числа и выдает результат в указанные слова результата.	5.20.5
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.	5.20.6

Код	Мнемоника	Наименование	Функция	Пункт
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.	5.24.1
39	(@)DEC	BCD DECREMENT	Декрементирует (уменьшает на 1) 4-разрядное двоично-десятичное число.	5.24.2
40	(@)STC	SET CARRY	Установить флаг переноса (CY) в 1	5.20.1
41	(@)CLC	CLEAR CARRY	Очистить флаг переноса (CY) в 0	5.20.2
45	TRSM	TRACE MEMORY SAMPLE	Пуск трасссировки данных. Нельзя использовать с CQM1-CPU11/21-E или CPM1.	5.26.1
46	(@)MSG	MESSAGE	Отображает сообщение из 16 символов на дисплее программатора.	5.26.2
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
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.	5.20.7
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.	5.20.9
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.	5.20.10

Код	Мнемоника	Наименование	Функция	Пункт
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).	5.19.3
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).	5.19.4
60-69	См. Дополнительные команды			
70	(@)XFER	BLOCK TRANSFER	Переносит содержание нескольких последовательных слов источника в несколько последовательных слов приемника.	5.17.3
71	(@)BSET	BLOCK SET	Переносит содержание константы или одного слова в несколько последовательных слов приемника.	5.17.4
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).	5.20.11
73	(@)XCHG	DATA EXCHANGE	Обменивает содержимое двух указанных слов	5.17.5
74	(@)SLD	ONE DIGIT SHIFT LEFT	Сдвигает влево данные между первым и последним словом на о дну цифру (4 бита).	5.16.7
75	(@)SRD	ONE DIGIT SHIFT RIGHT	Сдвигает вправо данные между первым и последним словом на о дну цифру (4 бита).	5.16.8
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.	5.19.5
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.	5.19.6
78	(@)SDEC	7-SEGMENT DECODER	Преобразует 16-ричное значение из слова источника в данные для 7-сегментного индикатора.	5.19.7
80	(@)DIST	SINGLE WORD DISTRIBUTE	Посылает слово источника в слово приемника, адрес которого задается в слове базы приемника плюс смещение.	5.17.6
81	(@)COLL	DATA COLLECT	Извлекает данные из слова источника и записывает в слово приемника.	5.17.7
82	(@)MOVB	MOVE BIT	Пересылает указанный бит слова источника или константы в указанный бит слова приемника.	5.17.8
83	(@)MOVD	MOVE DIGIT	Пересылает 16-ричное значение указанной цифры (цифр) источника в указанные цифры (цифру) слова приемника (всего до 4 цифр).	5.17.9
84	(@)SFTR	REVERSIBLE SHIFT REGISTER	Сдвигает данные в указанном слове или серии слов влево либо вправо.	5.16.9
85	(@)TCMP	TABLE COMPARE	Сравнивает 4-разрядное 16-ричное значение с значениями таблицы, в которой 16 слов.	5.18.2
86	(@)ASC	ASCII CONVERT	Преобразование 16-ричное значение источника в 8-битный код ASCII, начиная от старшей либо младшей половины слова.	5.19.8
87-89	Дополнительные команды			
91	(@)SBS	SUBROUTINE ENTRY	Вызов и исполнение подпрограммы N	5.25.1
92	SBN	SUBROUTINE DEFINE	Отметка начала подпрограммы N	5.22.2

Код	Мнемоника	Наименование	Функция	Пункт
93	RET	SUBROUTINE RETURN	Отметка конца подпрограммы N	5.25.2
97	(@)IORF	I/O REFRESH	Обновляет все слова входа/выхода между начальным и конечным словом	5.26.3
99	(@)MCRO	MACRO	Вызывает и исполняет подпрограмму с заменой слов входов/выходов	5.26.4

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	Тип ЦУ	Пункт
17	(@)ASFT	ASYNCHRONOUS SHIFT REGISTER	Создает регистр сдвига, который обменивает содержимое соседних слов, когда одно из слов = 0, а другое нет.	Все	5.16.10
18	(@)TKY	TEN KEY INPUT	Вводит 8 двоично-десятичных цифр из клавиатуры на 10 клавиш.	CQM1	5.28.4
19	(@)MCMP	MULTI-WORD COMPARE	Сравнивает блок из 16 последовательных слов с другим блоком из 16 последовательных слов.	CQM1	5.18.5
47	(@)RXD	RECEIVE	Получает данные через порт связи	CQM1	5.27.1
48	(@)TXD	TRANSMIT	Посылает данные через порт связи	CQM1	5.27.2
60	CMPL	DOUBLE COMPARE	Сравнивает два 8-разрядных 16-ричных числа	Все	5.18.4
61	(@)INI	MODE CONTROL	Запускает и останавливает операцию, сравнивает и изменяет текущие значения, и останавливает выдачу импульсов.	Все	5.15.7
62	(@)PRV	HIGH-SPEED COUNTER PV READ	Читает содержимое текущего значения и состояние счетчика для высокоскоростного счетчика.	Все	5.15.8
63	(@)CTBL	COMPARISON TABLE LOAD	Сравнивает содержание текущих значений и генерирует таблицу или начинает операцию	Все	5.15.6
64	(@)SPED	SPEED OUTPUT	Выдает импульсы с указанной частотой (10 Гц - 50 кГц через 10 Гц). Выходную частоту можно изменить во время выдачи импульсов.	CQM1	5.26.10
65	(@)PULS	SET PULSES	Выдает заданное число импульсов с заданной частотой. Выдача импульсов прекратится только после выдачи заданного числа импульсов.	CQM1	5.26.9
66	(@)SCL	SCALING	Преобразование с масштабированием .	CQM1	5.19.10
67	(@)BCNT	BIT COUNTER	Подсчитывает общее число битов в состоянии 1 в указанном блоке слов.	Все	5.26.5
68	(@)BCMP	BLOCK COMPARE	Определяет, находится ли значение слова в одной из 16 зон (задаются верхней и нижней границами).	Все	5.18.3
69	(@)STIM	INTERVAL TIMER	Управляет интервальными таймерами для исполнения прерываний по расписанию.	Все	5.15.5
87	DSW	DIGITAL SWITCH INPUT	Вводит 4- или 8-разрядные двоично-десятичные данные с кодового колеса.	CQM1	5.28.2
88	7SEG	7-SEGMENT DISPLAY OUTPUT	Преобразует 4- или 8-разрядные данные в формат 7-сегментного индикатора и выводит преобразованные данные.	CQM1	5.28.1

Код	Мнемоника	Наименование	Функция	Тип ЦУ	Пункт
89	(@)INT	INTERRUPT CONTROL	Осуществляет управление прерываниями, маскирование и размаскирование входных прерываний.	Все	5.26.8
—	(@)ACC	ACCELERATION CONTROL	(Только CQM1-CPU43-E.) Совместно с PULS(-) и ACC(-) управляет ускорением и/или замедлением выдачей импульсов с портов 1 и 2.	CQM1-CPU43-E	5.26.12
—	(@)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.	CQM1	5.21.5
—	(@)APR	ARITHMETIC PROCESS	Вычисляет синус, косинус, или линейную аппроксимацию.	CQM1	5.22.5
—	AVG	AVERAGE VALUE	Складывает указанное количество 16-ричных слов и вычисляет среднее значение. Округляет до 4 цифр после запятой.	CQM1	5.22.3
—	(@)COLM	LINE TO COLUMN	Копирует 16 битов из указанного слова в столбец битов 16 последовательных слов.	CQM1	5.19.16
—	CPS	SIGNED BINARY COMPARE	Сравнивает два 16-битовых (4 цифры) двоичных чисел со знаком и выдает результат во флаги GR, EQ и LE.	CQM1	5.18.6
—	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.	CQM1	5.18.7
—	(@)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.	CQM1	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	CQM1	5.21.10
—	(@)FCS	FCS CALCULATE	Проверяет ошибки в данных, переданных в запросе при связи с верхним уровнем.	CQM1	5.26.6
—	FPD	FAILURE POINT DETECT	Ищет сбои в блоке команд.	CQM1	5.26.7
—	(@)HEX	ASCII-TO HEXADECIMAL	Преобразует данные ASCII в 16-ричные.	CQM1	5.19.9
—	HKY	HEXADECIMAL KEY INPUT	Вводит до 8 цифр 16-ричных данных с клавиатуры 16 клавиш.	CQM1	5.28.3
—	HMS	SECOND TO HOURS	Преобразует данные в секундах в данные в виде часов и минут.	CQM1	5.19.14
—	LINE	LINE	Копирует столбец битов из 16 последовательных слов в указанное слово.	CQM1	5.19.15
—	(@)MAX	FIND MAXIMUM	Находит максимальное значение в указанной области данных и выдает это значение в другое слово.	CQM1	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.	CQM1	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.	CQM1	5.21.8
—	(@)MIN	FIND MINIMUM	Находит минимальное значение в указанной области данных и выдает это значение в другое слово.	CQM1	5.22.2

Код	Мнемоника	Наименование	Функция	Тип ЦУ	Пункт
—	(@)NEG	2’S COMPLEMENT	Преобразует 4-разрядное 16-ричное содержимое источника в дополнение до 2 и выдает в результат в R.	CQM1	5.19.17
—	(@)NEGL	DOUBLE 2’S COMPLEMENT	Преобразует8-разрядное16-ричное содержимое слов источника в дополнение до 2 и выдает в результат в R и R+1.	CQM1	5.19.18
—	PID	PID CONTROL	(Только CQM1-CPU43-E). Осуществляет ПИД-регулирование, основанное на заданных параметрах.	CQM1-CPU43-E	5.26.15
—	(@)PLS2	PULSE OUTPUT	(Только CQM1-CPU4-E). Увеличивает частоту выдачи импульсов от 0 до заданного значения с заданным темпом и уменьшает с таким же темпом.	CQM1-CPU4_-E	5.26.11
—	(@)PWM	PULSE WITH VARIABLE DUTY RATIO	(Только CQM1-CPU43-E). Выдает импульсы с заданным коэффициентом заполнения (1%... 99%) с портов 1 и 2.	CQM1-CPU43-E	5.26.13
—	(@)SBBL	DOUBLE BINARY SUBTRACT	Вычитает одно 8-разрядное двоичное число (со знаком или без) из другого и выдает результат в R+1 и R.	CQM1	5.21.6
—	(@)SCL2	SIGNED BINARY TO BCD SCALING	(Только CQM1-CPU4_-E). Осуществляет линейное преобразование 4-разрядного 16-ричного числа со знаком в 4-разрядное двоично-десятичное.	CQM1-CPU4-E	5.19.11
—	(@)SCL3	BCD TO SIGNED BINARY SCALING	(Только CQM1-CPU4-E). Осуществляет линейное преобразование 4-разрядного двоично-десятичного числа со знаком в 4-разрядное 16-ричное.	CQM1-CPU4-E	5.19.12
—	SEC	HOURS TO SECONDS	Преобразует данные в форме часы и минуты в форму секунд.	CQM1	5.19.13
—	(@)SRCH	DATA SEARCH	Ищет в занузданной зоне памяти указанные данные. Выдает адрес (адреса) слов, которые содержат данные.	CQM1	5.26.14
—	(@)SUM	SUM	Подсчитывает сумму содержимого слов в указанной зоне памяти.	CQM1	5.22.4
—	(@)XFRB	TRANSFER BITS	Копирует состояние до 255 заданных бит источника в указанные биты приемника.	CQM1	5.17.10
—	ZCPL	AREA RANGE COMPARE	Сравнивает слово с зоной, заданной нижней и верхней границами и выдает результат в флаги GR, EQ и LE.	CQM1	5.18.8
—	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.	CQM1	5.18.9

Приложение B

Операции с флагами ошибок и арифметики

В таблице представлены команды, оказывающие воздействие на флаги ER, CY, GT, LT и EQ.

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 Команды.

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 команды фронта 0/1 воздействуют на флаги точно так же.

Флаги ER, CY, GT, LT и EQ сбрасываются в 0 при исполнении команды END(01), так что их состояние нельзя индикатировать программирующим устройством.

На состояние флагов ER, CY, GT, LT и EQ влияет исполнение данных команд, и их состояние меняется каждый раз при исполнении команды. Команды, срабатывающие от фронта, срабатывают только раз при изменении условий исполнения (0/1 или 1/0) и повторно срабатывают при таком же изменении условий. Таким образом, команды, срабатывающие от франта, воздействуют на флаги ER, CY, GT, LT и EQ только раз при изменения условий на входе и не воздействуют, когда нет требуемого изменения состояния на входе. Когда команда, срабатывающая от фронта, не выполняется, состояние флагов ER, CY, GT, LT и EQ не изменяется и сохраняется состояние, вызванное исполнением последней команды.

TIM	**25503(ER)**	**25504(CY)**	**25505(GR)**	**25506(EQ)**	**25507(LE)**	**Пункт**
TIM	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.15.1
CNT						5.15.2
END(01)	OFF	OFF	OFF	OFF	OFF	5.10
STEP (08)	Не влияет	Не влияет	Не влияет	Не влияет	Не влияет	5.14
SNXT(09)						5.14
CNTR(12)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.13.3
TIMH (15)						5.15.4
WSFT (16)						5.16.2
CMP(20)	⇳	Не влияет	⇳	⇳	⇳	5.18.1
MOV (21)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.17.1
MVN (22)						5.17.2
BIN (23)						5.19.1
BCD (24)						5.19.2
ASL (25)	⇳	Не влияет	⇳	Не влияет	⇳	5.16.3
ASR (26)						5.16.4
ROL (27)						5.16.5
ROR (28)						5.16.6
COM (29)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.23.1

TIM	25503(ER)	25504(CY)	25505(GR)	25506(EQ)	25507(LE)	Пункт
ADD(30)	⇳	⇳	Не влияет	⇳	Не влияет	5.20.3
SUB (31)						5.20.4
MUL (32)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.20.5
DIV (33)						5.20.6
ANDW 34)						5.23.2
ORW (35)						5.23.3
XORW (36)						5.23.4
XNRW (37)						5.23.5
INC (38)						5.24.1
DEC (39)						5.24.2
STC (40)	Не влияет	ON	Не влияет	Не влияет	Не влияет	5.20.1
CLC (41)	Не влияет	OFF	Не влияет	Не влияет	Не влияет	5.20.2
MSG (46)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.26.2
ADB(50)	⇳	⇳	Не влияет	⇳	Не влияет	5.21.1
SBB (51)						5.21.2
MLB (52)	Не влияет	⇳	Не влияет	Не влияет	⇳	5.21.3
DVB (53)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.21.4
ADDL(54)	⇳	⇳	Не влияет	⇳	Не влияет	5.20.7
SUBL (55)						5.20.8
MULL (56)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.20.9
DIVL (57)						5.20.10
BINL (58)						5.19.3
BCDL (59)						5.19.4
XFER (70)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.17.3
BSET (71)						5.17.4
ROOT (72)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.20.11
XCHG (73)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.17.5
SLD (74)						5.16.7
SRD (75)						5.16.8
MLPX (76)						5.19.5
DMPX (77)						5.19.6
SDEC (78)						5.19.7
DIST (80)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.17.6
COLL (81)						5.17.7
MOVB (82)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.17.8
MOVD (83)						5.17.9
SFTR (84)	⇳	⇳	Не влияет	Не влияет	Не влияет	5.16.9
TCMP (85)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.18.2
ASC (86)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.19.8
SBS (91)						5.25.1
MCRO (99)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.26.4

Дополнительные команды (Все CQM1/SRM1)

TIM	25503(ER)	25504(CY)	25505(GR)	25506(EQ)	25507(LE)	Пункт
ASFT (17)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.16.10
TKY (18)						5.28.4
MCMP (19)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.18.5

RXD (47)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.27.1
TXD (48)						5.27.2
CMPL(60)	⇳	Не влияет	⇳	⇳	⇳	5.18.4
INI (61)	⇳	⇳	Не влияет	Не влияет	Не влияет	5.15.7
PRV (62)						5.15.8
CTBL (63)						5.15.6
SPED (64)						5.26.10
PULS (65)						5.26.9
SCL (66)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.19.10
BCNT (67)						5.26.5
BCMP (68)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.18.3
STIM (69)						5.15.5
DSW(87)[1]						5.28.2
7SEG(88)[2]						5.28.1
INT (89)						5.26.8
HKY (-)[3]						5.28.3
FPD(-)	⇳	⇳	Не влияет	Не влияет	Не влияет	5.26.7
SRCH (-)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.26.14
MAX (-)						5.22.1
MIN (-)						5.22.2
APR (-)						5.22.5
COLM (-)						5.19.16
LINE(-)						5.19.15
HMS (-)						5.19.14
SEC(-)						5.19.13
SUM (-)						5.22.4
FCS (-)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.26.6
HEX (-)						5.19.9
AVG(-)						5.22.3

Замечание 1. SR 25410 = 1 во время выполнения DSW(87)
2. SR 25409 = 1 во время выполнения 7SEG(88)
3. SR 25408 = 1 во время выполнения HKY(-)

Дополнительные команды (Только CQM1-CPU4_-EV1)

TIM	**25503(ER)**	**25504(CY)**	**25505(GR)**	**25506(EQ)**	**25507(LE)**	**Пункт**
PWM (-)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.16.10
PID						5.28.4
ADBL(-)	⇳	Не влияет	Не влияет		Не влияет	5.18.5
SBBL (-)						5.27.1
MBS (-)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.27.2
DBS (-)						5.18.4
MBSL (-)						5.15.7
DBSL (-)						5.15.8
CPS	⇳	Не влияет	⇳	⇳	Не влияет	5.15.6
CPSL						5.26.10
NEG (-)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.26.9
NEGL (-)						5.19.10

TIM	25503(ER)	25504(CY)	25505(GR)	25506(EQ)	25507(LE)	Пункт
ZCP	⇳	Не влияет	⇳	⇳	Не влияет	5.26.5
ZCPL						5.18.3
XFRB (-)	⇳	Не влияет	Не влияет	Не влияет	Не влияет	5.15.5
PLS2 (-)						5.28.2
ACC(-)						5.28.1
SCL2 (-)	⇳	⇳	Не влияет	⇳	Не влияет	5.26.8
SCL3 (-)	⇳	Не влияет	Не влияет	⇳	Не влияет	5.28.3

Замечание В зависимости от результатов вычислений ADBL(-) и SBBL(-) могут воздействовать также на состояние флагов переполнения + и переполнения - (SR 25404 и SR 25405).

В зависимости от результатов преобразования NEG(-) и NEGL(-) могут воздействовать также на состояние флага переполнения - (SR 25405).

Приложение C

Области памяти

Функции областей памяти CQM1

Структура области памяти

CQM1 использует следующие области памяти.

Область данных		Размер	Слова	Биты	Функции
Область IR[1]	Область входов	128 или 192 бит	IR 000 - IR 011	IR 00000 - IR 01115	CQM1_CPU11/21-E: до 8 слов могут использоваться как биты входов/выходов CQM1_CPU4_-E: до 12 слов могут использоваться как биты входов/выходов
	Область выходов		IR 100 - IR 111	IR 10000 - IR 11115	
	Рабочие области	Минимум 720 бит[2]	IR 012 - IR 095	IR 01200 - IR 09515	Могут произвольно использоваться внутри программы
			IR 112 - IR 195	IR 11200- IR 19515	
			IR 216 - IR 219	IR 21600 - IR 21915	
			IR 224 - IR 229	IR 22400 - IR 22915	
Область операнда MACRO	Область входов	64 бита	IR 096 - IR 099	IR 09600 - IR 09915	Применяется прииспользовании комманды MACRO(99). Когда команда не используется, данные биты можноиспользовать в качестве рабочих бит
	Область выходов	64 бита	IR 196 - IR 199	IR 19600 - IR 19915	
Область аналоговых заданных значений[1]		64 бита	IR 220 - IR 223	IR 22000 - IR 22315	CQM1-CPU42-E: Служит для введения задания с аналоговых регуляторов (нельзя использовать в качестве рабочих бит). В других ЦУ можно использовать в качестве рабочих бит.
Текущее значение высокоскоростного счетчика 0[1]		32 бита	IR 230 - IR 231	IR 23000 - IR 23115	Служат для сохранения текущего значения высокоскоростного счетчика 0
Текущие значения импульсных выходов портов 1 и 2[1]		64 бита	IR 236 - IR 239	IR 23600 - IR 23915	CQM1-CPU43-E: Служат для сохранения текущих значений импульсных выходов портов 1 и 2 (Нельзя использовать в качестве рабочих бит) CQM1-CPU44-E: Используются системой. (Нельзя использовать в качестве рабочих бит) В других ЦУ можно использовать в качестве рабочих бит.
Текущее значение высокоскоростных счетчиков 1 и 2 (прим. 1)		64 бита	IR 232 - IR 235	IR 23200 - IR 23515	CQM1-CPU43/44-E: Служат для сохранения текущего значения высокоскоростных счетчиков 1 и 2 портов 1 и 2 (Нельзя использовать в качестве рабочих бит) В других ЦУ можно использовать в качестве рабочих бит.
Области для дополнительных команд[1]		320 бит	IR 200 - IR 215 IR 240 - IR 243	IR 20000 - IR 21515 IR 24000 - IR 24315	Служат для использования в дополнительных командах.

Область данных		Размер	Слова	Биты	Функции
Область SR		184 бита	SR 244 - SR 255	SR 24400 - SR 25515	Выполняют специальные функции, такие, как флаги и биты контроля. Можно использовать в качестве рабочих бит.
Область TR		8 бит	-	TR 0 - TR 7	Данные биты служат для временного хранения состояния 1 или 0 в ответвлениях программы.
Область HR		1600 бит	HR 00 - HR 99	HR 0000 - HR 9915	Данные биты сохраняют данные и состояние 1 или 0 при отключении питания.
Область AR		448 бит	AR 00 - AR 27	AR 0000 - AR 2715	Выполняют специальные функции, такие, как флаги и биты контроля.
Область LR[1]		1024 бита	LR 00 - LR 63	LR 0000 - LR 6315	Служат для связи 1+1 через порт RS-232C.
Область таймеров/счетчиков[3]		512 бит	TC 000 - TC 511 (номера таймеров/ счетчиков)		Одни и те же номера используются и для таймеров и для счетчиков. TC 000 - TC 002 служат для интервальных таймеров
Область DM	Чтение/ запись	1024 слов	DM 0000-DM 1023	-	Данные области DM доступны только в виде слов. Значения слов сохраняются при отключении питания
		5120 слов	DM 1024-DM 6143	-	Есть только у ЦУ CQM1-CPU4_-E
	Только чтение[5]	425 слов	DM 6144-DM 6568	-	Не могут быть изменены из программы.
	Область записи протоко-ла ошибок[5]	31 слово	DM 6569-DM 6599	-	Служат для сохранения времени и кода появления признака ошибки
	Установочные парамет ры ПК[5]	56 слов	DM 6600-DM 6655	-	Служат для сохранения параметров, управляющих работой ПК.
Область программы пользователя (область UM)		3200 или7200 слов	-		Служат для сохранения программы Сохраняется при отключении питания. CQM1-CPU11/21-E: 3 200 слов CQM1-CPU4-E: 7 200 слов

Замечание 1. Биты IR и LR, которые не используются для специальных функций, можно использовать как рабочие биты.

2. В качестве рабочих бит можно использовать как минимум 2 720 бит. Общее количество бит, которые можно использовать в качестве рабочих, зависит от конфигурации системы ПК.

3. При обращении к текущим значениям номера таймеров/ счетчиков используются как слова; при обращении к флагам завершения они используются как биты.

4. Хотя CQM1-CPU11-E и CQM1-CPU21-E не поддерживают DM 1024 - DM 6143, при адресации к данным областям признака ошибки не появляется. Любая попытка записи в данные слова не приведет в результату и любое чтение даст нуль.

5. Значения DM 6144 - DM 6655 нельзя переписать из программы

Область SR

Данные биты служат главным образом для флагов, связанных с работой команд CQM1. В таблице представлены подробности функций битов.

Слово	Бит(ы)	Функция	Пункт
SR 244	00 - 15	Задание для счетчика при использовании входного прерывания 0 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 0 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 245	00 - 15	Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 246	00 - 15	Задание для счетчика при использовании входного прерывания 2 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 247	00 - 15	Задание для счетчика при использовании входного прерывания 3 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 248	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 0 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 249	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 250	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 2 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 251	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 3 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 252	00	Бит сброса высокоскоростного счетчика 0	1.4.5
	01	CQM1-CPU43-E: Бит сброса высокоскоростного счетчика 1 Устанавливается в 1 для сброса текущего значения высокоскоростного счетчика 1 (порт 1). CQM1-CPU44-E: Бит коррекции исходного Абсолютного высокоскоростного счетчика 1 Устанавливается в 1 для установки коррекции исходного для Абсолютного высокоскоростного счетчика 1 (порт 1). Автоматически сбрасывается в 0, когда коррекция исходного задана в DM 6611.	3.1.1
	02	CQM1-CPU43-E: Бит сброса высокоскоростного счетчика 2 Устанавливается в 1 для сброса текущего значения высокоскоростного счетчика 2 (порт 2). CQM1-CPU44-E: Бит коррекции исходного Абсолютного высокоскоростного счетчика 2 Устанавливается в 1 для установки коррекции исходного для Абсолютного высокоскоростного счетчика 2 (порт 2). Автоматически сбрасывается в 0, когда коррекция исходного задана в DM 6612.	3.1.1
	03 - 07	Не используется	-
	08	Бит сброса периферийного порта Устанавливается в 1 для сброса периферийного порта (Не действует, когда периферийный порт не подключен). Автоматически сбрасывается в 0 после завершения сброса.	1.6.3
	09	Бит сброса порта RS-232C. Устанавливается в 1 для сброса порта RS-232C. Автоматически сбрасывается в 0 после завершения сброса.	1.1.1

Слово	Бит(ы)	Функция	Пункт
	10	Бит сброса установочных параметров ПК Устанавливается в 1 для инициализации установочных параметров ПК (DM 6600..DM 6655). Автоматически сбрасывается в 0 после завершения сброса. Действует только в режиме ПК PROGRAM.	1.1
	11	Бит сохранения принудительного состояния. OFF: При переключении ПК из PROGRAM в MONITOR принудительное состояние битов, ранее принудительно установленных в 0 или 1, убирается. ON: При переключении ПК из PROGRAM в MONITOR принудительное состояние битов, ранее принудительно установленных в 0 или 1, остается.	1.2.2
	12	Бит сохранения входов/выходов. OFF:При пуске и останове операций биты IR и LR сбрасываются. ON: При пуске и останове операций биты IR и LR сохраняют свое значение.	1.2.2
	13	Не используется	-
	14	Бит сброса протокола ошибок Устанавливается в 1 для очистки протокола ошибок. Автоматически сбрасывается в 0 после завершения сброса.	8.8
	15	Бит выключения выходов OFF: Нормальное состояние выходов. ON: Сбросить все выходы в состояние 0.	Прил. C
SR 253	00 - 07	Код ошибок FAL Код ошибки (2-разрядное число) заносится при появлении признака ошибки. Номер FAL Заносится при исполнения команд FAL(06) или FAL(07). Сбрасывается в 00 при исполнении FAL 00 или очистки с программирующего устройства.	5.13
	08	Флаг неисправности аккумулятора Устанавливается в 1 при падении напряжения аккумулятора.	8.5.1
	09	Флаг Превышения времени цикла Устанавливается в 1 при превышении времени цикла. (т. е. когда время цикла превышает 100 мс).	8.5.1
	10 - 12	Не используется	-
	13	Флаг всегда 1	-
	14	Флаг всегда 0	-
	15	Флаг первого цикла Включается в 1 на 1 цикл при пуске программы.	-
SR 254	00	Импульс с периодом 1 минуту (30 с = 0, 30 с = 1).	-
	01	Импульс с периодом 0.02 секунд (0.01 с = 0, 0.01 с = 1).	-
	02 - 03	Не используется	-
	04	CQM1-CPU4-E: Флаг OF (Переполнения +) Устанавливается в 1, когда результаты вычислений превышает верхнюю границу двоичных данных со знаком.	5.21.1
	05	CQM1-CPU4-E: Флаг UF (Переполнения -) Устанавливается в 1, когда результаты вычислений меньше нижней границы двоичных данных со знаком.	5.21.1
	06	Флаг завершения просмотра фронта Устанавливается в 1 после завершения просмотра фронта.	3.1.1
	07	Флаг исполнения секции STEP Устанавливается в 1 на один цикл в начале исполнения секции STEP.	5.14
	08	Флаг исполнения команды HKY(-) Устанавливается в 1 во время исполнения HKY(-).	5.28.3
	09	Флаг исполнения 7SEG(88) Устанавливается в 1 во время исполнения 7SEG(88).	5.28.1
	10	Флаг исполнения DSW(87) Устанавливается в 1 во время исполнения DSW(87).	5.28.2
	11 - 14	Не используется	-

Слово	Бит(ы)	Функция	Пункт
	15	CQM1-CPU43-E: Флаг ошибки ввода/выдачи импульсов Устанавливается в 1 при ошибке при вводе/выдаче импульсов с портов 1 и 2. CQM1-CPU44-E: Флаг ошибки Абсолютного высокоскоростного счетчика 1 (FALS: 9C). Устанавливается в 1 при ошибке при ошибке Абсолютного высокоскоростного счетчика, использующего порты 1 и 2.	8.5.1
SR 255	00	Импульс с периодом 0.1 секунды (0.05 с = 0, 0.05 с = 1).	-
	01	Импульс с периодом 0.2 секунды (0.1 с = 0, 0.1 с = 1).	-
	02	Импульс с периодом 1 секунда (0.5 с = 0, 0.5 с = 1).	-
	03	Флаг ошибки исполнения команды (ER) Устанавливается в 1, когда при исполнении команды появляется ошибка	-
	04	Флаг Переноса (CY) Устанавливается в 1, когда результат исполнения команды появляется перенос	-
	05	Флаг Больше чем (GR) Устанавливается в 1, когда результат команды сравнения “больше”	-
	06	Флаг Больше чем (GR) Устанавливается в 1, когда результат команды сравнения “равно” или когда результат исполнения команды = 0	-
	07	Флаг Меньше чем (LE) Устанавливается в 1, когда результат команды сравнения “меньше”	-
	08 - 15	Не используются	-

Замечание Запись невозможна в слова SR 248..SR 251 и SR 253..SR 255.

Пояснения к битам SR.

SR 25211 (Бит сохранения принудительного состояния)

При убирании принудительного состояния биты, которые были принудительно установлены, примут следующее состояние:

Принудительно установленные в 1: установятся в 1

Принудительно установленные в 0: установятся в 0

Все принудительные состояния убираются при переключении ПК в режим RUN (см. прим).

Бит SR 25211 устанавливается в 1 и 0 с периферийного устройства.

В установочных параметрах (DM 6601) можно задать сохранение данного бита даже при включении питания.

Замечание В установочных параметрах (DM 6601) можно задать сохранение предыдущего состояния бита SR 25211 при включении питания. Данная настройка делается для того, чтобы предотвратить сброс принудительного состояния даже при включении питания.

SR 25212 (Бит сохранения входов/выходов)

Данный бит устанавливается в 1 и 0 с периферийного устройства.

В установочных параметрах (DM 6601) можно задать сохранение данного бита даже при включении питания.

Замечание В установочных параметрах (DM 6601) можно задать сохранение предыдущего состояния бита SR 25211 при включении питания. Данная настройка делается для того, чтобы предотвратить сброс состояния сохранения даже при включении питания.

SR 25215 (Бит выключения выходов)

При установке данного бита в 1, все выходы сбрасываются в 0 и горит индикатор на ЦУ INH. Выходы остаются в состоянии 0 даже если выходные биты установлены в 1 из программы, пока данный бит снова не установится в состояние 0.

SR 25308 (Флаг неисправности аккумулятора) и SR 25309 (Флаг Превышения времени цикла)

В установочных параметрах (DM 6655) можно задать, чтобы данные признаки ошибок не выдавались.

Область AR

Данные биты служат главным образом для флагов, связанных с работой команд CQM1. В таблице представлены подробности функций битов.

За исключением AR 23 (Счетчик отключений питания), состояние всех слов AR обновляется каждый цикл. (AR обновляется только при прерываниях питания).

Слово	Бит(ы)	Функция	Пункт
AR 00 - AR 03	-	Не используются	-
AR 04	08 - 15	CQM1-CPU43/44-E: Код состояния приема/выдачи импульсов или Абсолютного высокоскоростного счетчика 00: В норме 01,02: Сбой аппаратной части 03: Ошибка в установочных параметрах 04: ПК остановился во время выдачи импульсов	1.4.7
AR 05	00 - 07	CQM1-CPU43/44-E: Флаги сравнения с зоной высокоскоростного счетчика 1: бит 00 = 1: Текущее значение вошло в зону сравнения 1 бит 01 = 1: Текущее значение вошло в зону сравнения 2 бит 02 = 1: Текущее значение вошло в зону сравнения 3 бит 03 = 1: Текущее значение вошло в зону сравнения 4 бит 04 = 1: Текущее значение вошло в зону сравнения 5 бит 05 = 1: Текущее значение вошло в зону сравнения 6 бит 06 = 1: Текущее значение вошло в зону сравнения 7 бит 07 = 1: Текущее значение вошло в зону сравнения 8	1.4.7
	08	CQM1-CPU43/44-E: Флаг сравнения высокоскоростного счетчика 1 OFF Остановлен ON Идет сравнение	1.4.7
	09	CQM1-CPU43/44-E: Флаг переполнения +/ переполнения - высокоскоростного счетчика 1 OFF В норме ON Произошло переполнение (+ или -)	1.4.7
	10 - 11	Не используются	-
	12 - 15	CQM1-CPU43/44-E: Флаг выдачи с портов 1 и 2 бит 12 = 1: Задано замедление (0: не задано) бит 13 = 1: Задано число импульсов (0: не задано) бит 14 = 1: Выдача импульсов завершена (0: не завершена) бит 15 = 1: Идет выдача импульсов (0: не идет)	5.26.11
AR 06	00 - 15	CQM1-CPU43/44-E: Флаги высокоскоростного счетчика 2/выдачи импульсов с порта 2 Аналогичны значениям битов в AR 05 (Флаги высокоскоростного счетчика 1/выдачи импульсов с порта 1)	1.4.7
AR 07	00 - 11	Не используются	-
	12	Флаг Секции 6 DIP-переключателя OFF Секция 6 переключателя DIP находится в положении OFF ON Секция 6 переключателя DIP находится в положении ON	-
	13 - 15	Не используются	-
AR 08	00 - 03	Код ошибки связи через порт RS-232C (1- разрядное число) Код будет = F, когда компьютер с SSS подключен к периферийному порту.	1.6.3
	04	Флаг ошибки RS-232C Установлен в 1 при ошибке связи по порту RS-232C.	1.6.3
	05	Флаг разрешение передачи RS-232C Действует только при использовании связи с верхним уровнем, связи по RS-232C.	1.6.3

Слово	Бит(ы)	Функция	Пункт
	06	Флаг завершения приема RS-232C Действует только при связи по RS-232C.	1.6.3
	07	Флаг Переполнение приема RS-232C Действует только при связи по порту RS-232C..	1.6.3
	08 - 11	Код ошибки связи периферийного устройства (1- разрядное число) Код будет = F, когда компьютер с SSS подключен к периферийному порту.	1.6.5
	12	Флаг ошибки периферийного устройства Установлен в 1 при ошибке связи с периферийным устройством.	1.6.5
	13	Флаг разрешение передачи на периферийное устройство Действует только при использовании связи с верхним уровнем, связи по RS-232C	1.6.5
	14	Флаг завершения приема периферийного устройства Действует только при связи по RS-232C.	1.6.5
	15	Флаг Переполнение приема периферийного устройства Действует только при связи по порту RS-232C..	1.6.5
AR 09	00 - 15	Счетчик приема RS-232C двоично-десятичных цифры; Действует только при связи по RS-232C.	1.6.5
AR 10	00 - 15	Счетчик приема периферийного устройства 4 двоично-десятичных цифры; Действует только при связи по RS-232C.	1.6.5
AR 11	00 - 07	Флаги сравнения с зоной высокоскоростного счетчика 0: бит 00 = 1: Текущее значение вошло в зону сравнения 1 бит 01 = 1: Текущее значение вошло в зону сравнения 2 бит 02 = 1: Текущее значение вошло в зону сравнения 3 бит 03 = 1: Текущее значение вошло в зону сравнения 4 бит 04 = 1: Текущее значение вошло в зону сравнения 5 бит 05 = 1: Текущее значение вошло в зону сравнения 6 бит 06 = 1: Текущее значение вошло в зону сравнения 7 бит 07 = 1: Текущее значение вошло в зону сравнения 8	1.4.5
	08 - 15	Не используются	-
AR 12	00 - 15	Не используются	-
AR 13	00	Флаг “Кассета памяти установлена” Установлен в 1, если при включении питания была установлена кассета памяти.	-
	01	Флаг “Есть часы” Установлен в 1, если установлена кассета памяти с часами.	-
	02	Флаг “Кассета защищена от записи” Установлен в 1, если установленная кассета памяти EEPROM защищена от записи или установлена кассета EPROM.	3.1.1
	03	Не используются	-
	04 - 07	Код кассеты памяти (одна цифра) 0: Кассета памяти не установлена 1: Установлена кассета памяти с EEPROM, 4 К слов 2: Установлена кассета памяти с EEPROM, 8 К слов 4: Установлена кассета памяти типа EPROM	-
	08 - 15	Не используются	-
AR 14	00	Бит “Передача с ЦУ на кассету” Установлен в 1 для передачи с ЦУ на кассету памяти. Автоматически устанавливается в 0 после завершения операции.	3.3.3
	01	Бит “Передача с кассеты на ЦУ” Установлен в 1 для передачи с кассеты памяти на ЦУ. Автоматически устанавливается в 0 после завершения операции.	3.3.4
	02	Флаг “Сравнение с кассетой памяти” Установлен в 1 при сравнении содержимого кассеты памяти и ПК. Автоматически устанавливается в 0 после завершения операции.	3.3.5
	03	Флаг “Результаты Сравнения с кассетой памяти” ON: Обнаружены различия, или сравнения невозможно OFF: Сравнение произведено и содержимое оказалось одинаковым.	3.3.5
	04 - 11	Не используются	-

Слово	Бит(ы)	Функция	Пункт
	12	Флаг “Ошибка передачи из-за режима PROGRAM” Установлен в 1, если передача невозможна из-за того, что ПК находится в режиме PROGRAM.	3.3.4
	13	Флаг “Ошибка из-за защиты записи” Установлен в 1, если передача невозможна из-за защиты от записи.	8.5.1
	14	Флаг “Мала память” Установлен в 1, если передача невозможна из-за недостаточности памяти в приемнике.	8.5.1
	15	Флаг “Нет программы” Установлен в 1, если передача невозможна из-за того, что в кассете памяти нет программы.	8.5.1
AR 15	00 - 07	Код программы кассеты памяти Код (две цифры) указывает размер программы на кассете памяти 00: Нет программы или кассета памяти не установлена 04: Программа менее 3.2 К слов 08: Программа менее 7.2 К слов	-
	08 - 15	Код программы в ЦУ Код (две цифры) указывает размер программы в ЦУ. 04: Программа менее 3.2 К слов 08: Программа менее 7.2 К слов.	-
AR 16	00 - 10	Не используются	-
	11	Флаг “Установочные параметры инициализированы” Установлен в 1 при ошибке контрольной суммы в области установочных параметров и все установки сбрасываюстя в установки по умолчанию	8.5.2
	12	Флаг “Ошибка в программе” Установлен в 1 при ошибке контрольной суммы в области программы UM, или при выполнении некорректной команды.	8.5.2
	13	Флаг “Инициализация таблицы команд” Установлен в 1 при ошибке контрольной суммы в таблице команд и все установки сбрасываются в установки по умолчанию.	8.5.2
	14	Флаг “Ошибка установки кассеты памяти” Установлен в 1, если кассета памяти устанавливается при включенном питании.	8.5.2
	15	Флаг “Ошибка передачи кассеты памяти” Установлен в 1, если невозможно осуществить передачу, когда секция 2 DIP-переключателя установлена в ON (т. е. задан на автоматическую передачу содержимого кассеты при включении питания).	8.5.2
AR 17	00 - 07	“Минуты” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
	08 - 15	“Часы” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
AR 18	00 - 07	“Секунды” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
	08 - 15	“Минуты” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
AR 19	00 - 07	“Часы” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
	08 - 15	“Дата” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
AR 20	00 - 07	“Месяц” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
	08 - 15	“Год” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1

Слово	Бит(ы)	Функция	Пункт
AR 21	00 - 07	“День недели” текущего времени, 2 двоично-десятичных цифры (Действует только при установленной кассете памяти с часами).	3.1.1
	08 - 12	Не используется	-
	13	Бит настройки по 30 с (Действует только при установленной кассете памяти с часами).	3.1.1
	14	Бит остановки часов (Действует только при установленной кассете памяти с часами).	3.1.1
	15	Бит задания часов (Действует только при установленной кассете памяти с часами).	3.1.1
AR 22	00 - 07	Количество Входных слов Количество слов для битов входа (2 двоично-десятичных цифры)	3.1.2
	08 - 15	Количество Выходных слов Количество слов для битов выхода (2 двоично-десятичных цифры)	3.1.2
AR 23	00 - 15	Счетчик выключений питания (4 двоично-десятичных цифры) Подсчитывает число выключений питания. Для очистки счетчика введите 0000 с периферийного устройства.	--
AR 24	00	Флаг ошибки в установочных параметрах, которые читаются по включению питания Устанавливается в 1 при ошибке в DM 6600 - DM 6614 (установочных параметрах, которые читаются по включению питания)	1.1.1
	01	Флаг ошибки в установочных параметрах, которые читаются при пуске операции Устанавливается в 1 при ошибке в DM 6615 - DM 6644 (установочных параметрах, которые читаются при пуске операции)	8.5.1
	02	Флаг ошибки в установочных параметрах, которые читаются постоянно Устанавливается в 1 при ошибке в DM 6645 - DM 6655 (установочных параметрах, которые читаются постоянно)	1.1.1
	03, 04	Не используется	-
	05	Флаг Большое время цикла Устанавливается в 1 если текущее время цикла больше заданного в DM 6619.	-
	06, 07	Не используется	-
	08 - 15	Код (2 двоично-десятичных цифры) показывает номер слова, у которого обнаружена ошибка шины входов/выходов. 00- 07: Соответствует входным словам 000..007 80- 87: Соответствует выходным словам 100..107 FF: Не отзывается оконечная крышка.	-
AR 25	00 - 07	Не используется	-
	08	Бит обучение FPD(-)	5.26.1
	09 - 15	Не используется	-
AR 26	00 - 15	Максимальное время цикла (4 двоично-десятичных цифры) Сохраняется наибольшее время цикла от начала операции. Очищается не в конце, а в начале операции. Могут быть следующие единицы, в зависимости от установки 9F времени контроля (параметр DM 6618). По умолчанию: 0.1 мс; задание “10 ms”: 0.1 мс; задание “100 ms”: 1 мс; задание “1 s”: 10 мс	1.2.10
AR 27	00 - 15	Текущее время цикла (4 двоично-десятичных цифры) Сохраняется самое последнее время цикла во время операции. При остановке операции не очищается. Могут быть следующие единицы, в зависимости от установки 9F времени контроля (параметр DM 6618). По умолчанию: 0.1 мс; задание “10 ms”: 0.1 мс; задание “100 ms”: 1 мс; задание “1 s”: 10 мс	1.2.10

Функции областей памяти CPM1/CPM1A

Структура области памяти

CPM1/CPM1A использует следующие области памяти.

Область данных		Размер	Слова	Биты	Функции
Область IR[1]	Область входов	160 бит	IR 000 - IR 009	IR 00000 - IR 00915	Данные биты могут быть выделены внешним входам/выходам.
	Область выходов	160 бит	IR 010 - IR 019	IR 01000 - IR 01915	
	Рабочая область	512 бит	IR 200 - IR 231	IR 20000 - IR 23115	Рабочие биты могут произвольно использоваться внутри программы
Область SR		384 бита	SR 232 - SR 255	SR 23200 - SR 25515	Выполняют специальные функции,такие, как флаги и биты контроля.
Область TR		-	-	TR 0- TR 7	Данные биты служат для временного хранения состояния 1 или 0 при ветвлениях программы.
Область HR[2]		320 бит	HR 00 - HR 19	HR 0000 - HR 1915	Данные биты сохраняют данные и сохраняют состояние 1 или 0 при отключении питания
Область AR[2]		256 бит	AR 00 - AR 15	AR 0000 - AR 1515	Выполняют специальные функции, такие, как флаги и биты контроля
Область LR[1]		256 бит	LR 00 - LR 15	LR 0000 - LR 1515	Служат для связи 1+1 с другим ПК
Область таймеров/счетчиков[2]		TC 000 - TC 127 (номера таймеров/счетчиков)[3]			Одни и те же номера используются и для таймеров и для счетчиков
Область DM	Чтение/ запись	1002 слова	DM 0000 - DM 0999 DM 1022 - DM 1023		Данные области DM доступны только в виде слов. Значения слов сохраняются при отключении питания.
	Область протокола ошибок (прим. 4)	22 слова	DM 1000 - DM 1021		Служат для сохранения времени и кода появления признака ошибки. Если функции протокола ошибок не используются, данные слова можно использовать как обычные слова чтения/записи.
	Только читать (прим. 4)	456 слов	DM 6144 - DM 6599		Не могут быть переписаны из программы.
	Установочные параметры ПК (прим. 4)	56 слов	DM 6600 - DM 6655		Служат для сохранения параметров, управляющих работой ПК.

Замечание 1. Биты IR и LR, которые не используются для специальных функций, можно использовать как рабочие биты.

2. Содержание областей HR, AR, счетчиков/таймеров и DM чтения/записи поддерживаются аккумулятором. При 25 град. аккумулятор сохраняет память 20 суток. График зависимости времени сохранения от температуры см. 2-1-2 Инструкции по работе.

3. При обращении к текущим значениям номера таймеров/ счетчиков используются как слова; при обращении к флагам завершения они используются как биты.

4. Значения DM 6144 - DM 6655 нельзя переписать из программы, но можно изменить с периферийных устройств.

Область SR

Данные биты служат главным образом для флагов, связанных с работой команд CPM1. В таблице представлены подробности функций битов.

Слово	Бит(ы)	Функция	Пункт
SR 232 - SR 235	00 - 15	Область входа функции Макро Содержит входные операнды для MCRO(99) (Если MCRO(99) не используется, могут использоваться в качестве входных битов).	2.3
SR 236 - SR 239	00 - 15	Область выхода функции Макро Содержит выходные операнды для MCRO(99) (Если MCRO(99) не используется, могут использоваться в качестве входных битов).	1.4.2
SR 240	00 - 15	Задание для счетчика при использовании входного прерывания 0 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 0 в режиме СЧЕТЧИК (4-разрядное 16-ричное число). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 241	00 - 15	Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 242	00 - 15	Задание для счетчика при использовании входного прерывания 2 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 243	00 - 15	Задание для счетчика при использовании входного прерывания 3 в режиме СЧЕТЧИК Задание для счетчика при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число). (Можно использовать в качестве рабочих бит, если входное прерывание 0 используется не в счетном режиме)	1.4.2
SR 244	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 0 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 245	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 1 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 246	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 2 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 247	00 - 15	Текущее значение счетчика минус единица при использовании входного прерывания 3 в режиме СЧЕТЧИК (4-разрядное 16-ричное число 0000 - FFFF).	1.4.2
SR 248, SR 149	00 - 15	Область текущего значения высокоскоростного счетчика (Можно использовать в качестве рабочих бит, если высокоскоростной счетчик не используется).	1.4.5
SR 250	00 - 15	Значение аналогового регулятора 0 Служит для хранения 4-разрядного двоично-десятичного задания (0000 - 0200) с аналогового регулятора 0.	2.5
SR 251	00 - 15	Значение аналогового регулятора 1 Служит для хранения 4-разрядного двоично-десятичного задания (0000 - 0200) с аналогового регулятора 1.	2.5

Слово	Бит(ы)	Функция	Пункт
SR 252	00	Бит сброса высокоскоростного счетчика 0	1.4.5
	01 - 07	Не используется	-
	08	Бит сброса периферийного порта Устанавливается в 1 для сброса периферийного порта (Не действует, когда периферийный порт не подключен). Автоматически сбрасывается в 0 после завершения сброса.	1.6.3
	09	Не используется	-
	10	Бит сброса установочных параметров ПК Устанавливается в 1 для инициализации установочных параметров ПК (DM 6600..DM 6655). Автоматически сбрасывается в 0 после завершения сброса. Действует только в режиме ПК PROGRAM.	1.1
	11	Бит сохранения принудительного состояния При переключении ПК из PROGRAM в MONITOR принудительное состояние битов, ранее принудительно установленных в 0 или 1, убирается. При переключении ПК из PROGRAM в MONITOR принудительное состояние битов, ранее принудительно установленных в 0 или 1, остается.	1.2.2
	12	Бит сохранения входов/выходов При пуске и останове операций биты IR и LR сбрасываются. При пуске и останове операций биты IR и LR сохраняют свое значение.	1.2.2
	13	Не используется	-
	14	Бит сброса протокола ошибок Устанавливается в 1 для очистки протокола ошибок. Автоматически сбрасывается в 0 после завершения сброса.	8.6
	15	Не используется	-
SR 253	00 - 07	Код ошибок FAL Код ошибки (2-разрядное число) заносится при появлении признака ошибки. Номер FAL Заносится при исполнения команд FAL(06) или FAL(07). Сбрасывается в 00 при исполнении FAL 00 или очистки с программирующего устройства.	5.13
	08	Не используется	-
	09	Флаг Превышения времени цикла Устанавливается в 1 при превышении времени цикла. (т. е. когда время цикла превышает 100 мс).	-
	10 - 12	Не используется	-
	13	Флаг всегда 1	-
	14	Флаг всегда 0	-
	15	Флаг первого цикла Включается в 1 на 1 цикл при пуске программы.	-
SR 254	00	Импульс с периодом 1 минуту (30 с = 0, 30 с = 1).	-
	01	Импульс с периодом 0.02 секунд (0.01 с = 0, 0.01 с = 1).	-
	02	Флаг Отрицательно (N)	-
	03 - 05	Не используется	-
	06	Флаг завершения просмотра фронта Устанавливается в 1 после завершения просмотра фронта.	3.1.1
	07	Флаг исполнения секции STEP(08) Устанавливается в 1 на один цикл в начале исполнения секции STEP(08).	5.14
	08 - 15	Не используется	-

Слово	Бит(ы)	Функция	Пункт
SR 255	00	Импульс с периодом 0.1 секунды (0.05 с = 0, 0.05 с = 1).	-
	01	Импульс с периодом 0.2 секунды (0.1 с = 0, 0.1 с = 1).	-
	02	Импульс с периодом 1 секунда (0.5 с = 0, 0.5 с = 1).	-
	03	Флаг ошибки исполнения команды (ER) Устанавливается в 1, когда при исполнении команды появляется ошибка.	-
	04	Флаг Переноса (CY) Устанавливается в 1, когда результат исполнения команды появляется перенос .	-
	05	Флаг Больше чем (GR) Устанавливается в 1, когда результат команды сравнения “больше” .	-
	06	Флаг Больше чем (GR) Устанавливается в 1, когда результат команды сравнения “равно” или когда результат исполнения команды = 0.	-
	07	Флаг Меньше чем (LE) Устанавливается в 1, когда результат команды сравнения “меньше” .	-
	08 - 15	Не используются	-

Область AR

Данные биты служат главным образом для флагов, связанных с работой команд CPM1. Данные биты сохраняют свое состояние даже при отключении питания CPM1 или при прерывания питания.

Слово	Бит(ы)	Функция	Пункт
AR 00, AR 01	00 - 15	Не используются	-
AR 02	00 - 07	Не используются	-
	08 - 11	Количество подключенных блоков входа/выхода	-
	12 - 15	Не используются	-
AR 03, AR 07	00 - 15	Не используются	-
AR 08	00 - 07	Не используются	-
	08 - 11	Код ошибки периферийного устройства 0: Нормальное завершение 1: Ошибка четности 2: Ошибка конверта 3: Ошибка переполнения	1.6.6
	12	Флаг ошибки периферийного устройства	1.6.6
	13 - 15	Не используются	-
AR 09	00 - 15	Не используются	-
AR 10	00 - 15	Счетчик выключений питания (4 двоично-десятичных цифры) Подсчитывает число выключений питания. Для очистки счетчика введите 0000 с периферийного устройства.	-
AR 11	00 - 07	Флаги сравнения с зоной высокоскоростного счетчика: бит 00 = 1: Текущее значение вошло в зону сравнения 1 бит 01 = 1: Текущее значение вошло в зону сравнения 2 бит 02 = 1: Текущее значение вошло в зону сравнения 3 бит 03 = 1: Текущее значение вошло в зону сравнения 4 бит 04 = 1: Текущее значение вошло в зону сравнения 5 бит 05 = 1: Текущее значение вошло в зону сравнения 6 бит 06 = 1: Текущее значение вошло в зону сравнения 7 бит 07 = 1: Текущее значение вошло в зону сравнения 8	1.4.5
	08 - 14	Не используются	-
	15	Состояние выдачи импульсов ON: Остановлено OFF: Идет выдача импульсов	-
AR 12	00 - 15	Не используются	-

Слово	Бит(ы)	Функция	Пункт
AR 13	00	Флаг ошибки в установочных параметрах, которые читаются по включению питания Устанавливается в 1 при ошибке в DM 6600 - DM 6614 (установочных параметрах, которые читаются по включению питания)	1.1.1
	01	Флаг ошибки в установочных параметрах, которые читаются при пуске операции Устанавливается в 1 при ошибке в DM 6615 - DM 6644 (установочных параметрах, которые читаются при пуске операции)	1.1.1
	02	Флаг ошибки в установочных параметрах, которые читаются постоянно Устанавливается в 1 при ошибке в DM 6645 - DM 6655 (установочных параметрах, которые читаются постоянно)	1.1.1
	03, 04	Не используется	-
	05	Флаг Большое время цикла Устанавливается в 1 если текущее время цикла больше заданного в DM 6619.	-
	06, 07	Не используется	-
	08	Флаг ошибки распределения памяти Устанавливается в 1, когда в программе есть обращение к несуществующему адресу.	-
	09	Флаг ошибки FLASH памяти Устанавливается в 1, когда произошла ошибка в памяти FLASH	-
	10	Флаг ошибки параметров DM, которые предназначены только для чтения Устанавливается в 1 при обнаружении ошибки контрольной суммы параметров, которые предназначены только для чтения (DM 6144 - DM 6599) и эта область инициализируется.	1.1.1
	11	Флаг ошибки установочных параметров Устанавливается в 1 при обнаружении ошибки контрольной суммы в установочных параметрах.	-
	12	Флаг ошибки программы Устанавливается в 1 при обнаружении ошибки контрольной суммы в памяти программы (UM) или исполнении некорректной команды.	-
	13 - 15	Не используется	-
AR 14	00 - 15	Максимальное время цикла (4 двоично-десятичных цифры) Сохраняется наибольшее время цикла от начала операции. Очищается не в конце, а в начале операции. Могут быть следующие единицы, в зависимости от установки параметра DM 6618. По умолчанию: 0.1 мс; задание “10 ms”: 0.1 мс; задание “100 ms”: 1 мс; задание “1 s”: 10 мс	1.2.10
AR 27	00 - 15	Текущее время цикла (4 двоично-десятичных цифры) Сохраняется самое последнее время цикла во время операции. При остановке операции не очищается. Могут быть следующие единицы, в зависимости от установки параметра DM 6618. По умолчанию: 0.1 мс; задание “10 ms”: 0.1 мс; задание “100 ms”: 1 мс; задание “1 s”: 10 мс	1.2.10

Функции областей памяти SRM1

Структура областей памяти

В SRM1 можно использовать следующие области памяти.

Область данных		Размер	Слова	Биты	Функция
Область IR[1]	Область входов	160 бит	IR 000.. IR 009	IR 00000.. IR 00915	Данные биты могут быть выделены внешним входам/выходам. Состояние 1 или 0 битов входов/выходов будут аналогичны состоянию на клеммах.
	Область выходов	160 бит	IR 010.. IR 019	IR 01000.. IR 01915	Все биты, которые не используются для входов/выходов можно использовать в качестве рабочих бит.
	Рабочая область	512 бит	IR 200.. IR 239	IR 20000.. IR 23915	Рабочие биты могут произвольно использоваться внутри программы. Однако биты IR 232..IR 239 использовать в качестве области входов MACRO для команды MACRO.
Область SR		248 бит	SR 240.. SR 255	SR 24000.. SR 25515	Эти биты служат в качестве ячеек сохранения флагов и заданных/текущих значений при отработке программы SRM1. См. Область AR.
Область TR		8 бит	-	TR 0..TR 7	Когда сложную РКС нельзя записать в мнемокоде эти биты служат для временного сохранения состояния 1 / 0 в ветвях программы. Эти временные биты нельзя использовать в пределах одного блока, но если блоки разные, то можно. Состояние 1/0 данных битов нельзя просмотреть с помощью функций просмотра периферийного устройства.
Область HR [2]		320 бит	HR 00.. HR 19	HR 0000.. HR 1915	Данные биты служат для хранения данных и сохраняют состояние 1/ 0 при отключении питания или запуске и останове программы. Они используются так же, как и рабочие биты.
Область AR [2]		256 бит	AR 00.. AR 15	AR 0000.. AR 1515	Эти биты обслуживают специальные функции, такие, как флаги и биты управления. Биты AR 04..07 служат ведомыми. См. Область AR.
Область LR [1]		256 бит	LR 00.. LR 15	LR 0000.. LR 1515	Служат для связи 1:1 с другими ПК SRM1, CQM1 или C200HS.
Область таймеров/счетчиков[2]		TC 000..TC 127[3]			Таймеры и счетчики используют команды TIM, TIMH(15) CNT, CNTR(12). Одинаковые номера используются и для таймеров и для счетчиков.
Область DM	Читать/ писать[2]	2000 слов	DM0000.. DM1999	-	Данные в области DM доступны только в виде слов. Значения слов сохраняются при отключении питания или при пуске / останове программы. В область чтения/записи можно читать и писать свободно в программе.
	Протокол ошибок[4]	22 слова	DM1000.. DM 1021	-	Служит для сохранения времени появления и кода произошедших ошибок. См. 5-5.
	Только читать[4]	456 слов	DM6144.. DM 6599	-	Не могут быть переписаны из программы.

Область данных		Размер	Слова	Биты	Функция
	Установочные параметры ПК[4]	56 слов	DM6600.. DM 6655	-	Служат для сохранения параметров, управляющих работой ПК.

Замечание 1. Биты IR и LR, которые не используются для отведенных им функций, можно использовать как рабочие биты.

2. Содержание областей HR, AR, счетчиков и DM чтения/записи поддерживается конденсатором. При 25 град. конденсатор сохраняет память 20 суток. График зависимости времени сохранения от температуры см. 2-1-2 Инструкции по работе с Мастер-Модулем SRM1.

3. При обращении к текущим значениям номера таймеров/ счетчиков используются как слова; при обращении к флагам завершения они используются как биты.

4. Значения DM 6144..DM 6655 нельзя переписать из программы, но можно изменить с периферийных устройств.

Область SR

Данные биты служат главным образом для флагов, связанных с работой команд CQM1. В таблице представлены подробности функций битов.

Слово	Бит(ы)	Функция	Пункт
SR 240 - SR 247	00 - 15	Не используются. Можно использовать в качестве рабочих бит	
SR 248 - SR 249	00 - 15	Зарезервировано	
SR 250 - SR 251	00 - 15	Не используются. Можно использовать в качестве рабочих бит	
SR 252	00	Не используются (используется системой)	
	01 - 07	Не используется	-
	08	Бит сброса периферийного порта Устанавливается в 1 для сброса периферийного порта (Не действует, когда периферийный порт не подключен). Автоматически сбрасывается в 0 после завершения сброса.	1.6.3
	09	Бит сброса порта RS-232C. Устанавливается в 1 для сброса порта RS-232C. Автоматически сбрасывается в 0 после завершения сброса.	1.1.1
	10	Бит сброса установочных параметров ПК Устанавливается в 1 для инициализации установочных параметров ПК (DM 6600..DM 6655). Автоматически сбрасывается в 0 после завершения сброса. Действует только в режиме ПК PROGRAM.	1.1
	11	Бит сохранения принудительного состояния. OFF: При переключении ПК из PROGRAM в MONITOR принудительное состояние битов, ранее принудительно установленных в 0 или 1, убирается. ON: При переключении ПК из PROGRAM в MONITOR принудительное состояние битов, ранее принудительно установленных в 0 или 1, остается.	1.2.2
	12	Бит сохранения входов/выходов. OFF:При пуске и останове операций биты IR и LR сбрасываются. ON: При пуске и останове операций биты IR и LR сохраняют свое значение.	1.2.2
	13	Не используется	-
	14	Бит сброса протокола ошибок Устанавливается в 1 для очистки протокола ошибок. Автоматически сбрасывается в 0 после завершения сброса.	8.8
	15	Бит выключения выходов OFF: Нормальное состояние выходов. ON: Сбросить все выходы в состояние 0.	Прил. C

Слово	Бит(ы)	Функция	Пункт
SR 253	00 - 07	Код ошибок FAL Код ошибки (2-разрядное число) заносится при появлении признака ошибки. Номер FAL Заносится при исполнения команд FAL(06) или FAL(07). Сбрасывается в 00 при исполнении FAL 00 или очистки с программирующего устройства.	5.13
	08	Флаг неисправности аккумулятора Устанавливается в 1 при падении напряжения аккумулятора.	8.5.1
	09	Флаг Превышения времени цикла Устанавливается в 1 при превышении времени цикла. (т. е. когда время цикла превышает 100 мс).	8.5.1
	10 - 12	Не используется	-
	13	Флаг всегда 1	-
	14	Флаг всегда 0	-
	15	Флаг первого цикла Включается в 1 на 1 цикл при пуске программы.	-
SR 254	00	Импульс с периодом 1 минуту (30 с = 0, 30 с = 1).	-
	01	Импульс с периодом 0.02 секунд (0.01 с = 0, 0.01 с = 1).	-
	02 - 03	Не используется	-
	04	CQM1-CPU4-E: Флаг OF (Переполнения +) Устанавливается в 1, когда результаты вычислений превышает верхнюю границу двоичных данных со знаком.	5.21.1
	05	CQM1-CPU4-E: Флаг UF (Переполнения -) Устанавливается в 1, когда результаты вычислений меньше нижней границы двоичных данных со знаком.	5.21.1
	06	Флаг завершения просмотра фронта Устанавливается в 1 после завершения просмотра фронта.	3.1.1
	07	Флаг исполнения секции STEP Устанавливается в 1 на один цикл в начале исполнения секции STEP.	5.14
SR 255	00	Импульс с периодом 0.1 секунды (0.05 с = 0, 0.05 с = 1).	-
	01	Импульс с периодом 0.2 секунды (0.1 с = 0, 0.1 с = 1).	-
	02	Импульс с периодом 1 секунда (0.5 с = 0, 0.5 с = 1).	-
	03	Флаг ошибки исполнения команды (ER) Устанавливается в 1, когда при исполнении команды появляется ошибка	-
	04	Флаг Переноса (CY) Устанавливается в 1, когда результат исполнения команды появляется перенос	-
	05	Флаг Больше чем (GR) Устанавливается в 1, когда результат команды сравнения “больше”	-
	06	Флаг Больше чем (GR) Устанавливается в 1, когда результат команды сравнения “равно” или когда результат исполнения команды = 0	-
	07	Флаг Меньше чем (LE) Устанавливается в 1, когда результат команды сравнения “меньше”	-
	08 - 15	Не используются	-

Область AR

Данные биты служат главным образом для флагов, связанных с работой команд CQM1. В таблице представлены подробности функций битов.

За исключением AR 23 (Счетчик отключений питания), состояние всех слов AR обновляется каждый цикл. (AR обновляется только при прерываниях питания).

Слово	Бит(ы)	Функция	Пункт
AR 00 - AR 03	-	Не используются	-

Слово	Бит(ы)	Функция	Пункт
AR 04 - AR 07	00 - 15	Флаг состояния ведомых	-
AR 08	00 - 03	Код ошибки связи через порт RS-232C (1- разрядное число) Код будет = F, когда компьютер с SSS подключен к периферийному порту.	1.6.3
	04	Флаг ошибки RS-232C Установлен в 1 при ошибке связи по порту RS-232C.	1.6.3
	05	Флаг разрешение передачи RS-232C Действует только при использовании связи с верхним уровнем, связи по RS-232C.	1.6.3
	06	Флаг завершения приема RS-232C Действует только при связи по RS-232C.	1.6.3
	07	Флаг Переполнение приема RS-232C Действует только при связи по порту RS-232C..	1.6.3
	08 - 11	Код ошибки связи периферийного устройства (1- разрядное число) Код будет = F, когда компьютер с SSS подключен к периферийному порту.	1.6.5
	12	Флаг ошибки периферийного устройства Установлен в 1 при ошибке связи с периферийным устройством.	1.6.5
	13	Флаг разрешение передачи на периферийное устройство Действует только при использовании связи с верхним уровнем, связи по RS-232C	1.6.5
	14	Флаг завершения приема периферийного устройства Действует только при связи по RS-232C.	1.6.5
	15	Флаг Переполнение приема периферийного устройства Действует только при связи по порту RS-232C..	1.6.5
AR 09	00 - 15	Счетчик приема RS-232C двоично-десятичных цифры; Действует только при связи по RS-232C.	1.6.5
AR 10	00 - 15	Счетчик приема периферийного устройства 4 двоично-десятичных цифры; Действует только при связи по RS-232C.	1.6.5
AR 11	00 - 15	Счетчик выключений питания (4 двоично-десятичных цифры) Подсчитывает число выключений питания. Для очистки счетчика введите 0000 с периферийного устройства.	-
AR 12	00 - 15	Не используется	
AR 13	00	Флаг ошибки в установочных параметрах, которые читаются по включению питания Устанавливается в 1 при ошибке в DM 6600 - DM 6614 (установочных параметрах, которые читаются по включению питания)	1.1.1
	01	Флаг ошибки в установочных параметрах, которые читаются при пуске операции Устанавливается в 1 при ошибке в DM 6615 - DM 6644 (установочных параметрах, которые читаются при пуске операции)	8.5.1
	02	Флаг ошибки в установочных параметрах, которые читаются постоянно Устанавливается в 1 при ошибке в DM 6645 - DM 6655 (установочных параметрах, которые читаются постоянно)	1.1.1
	03, 04	Не используется	-
	05	Флаг Большое время цикла Устанавливается в 1 если текущее время цикла больше заданного в DM 6619.	-
	06	Устанавливается в 1, когда Область памяти программ заполнена	-
	07		-
	08		
	09		
	10		
	11		
	12		
	13, 14	Не используется	

Слово	Бит(ы)	Функция	Пункт
	15	Ошибка CompoBus/S	-
AR 14	00 - 15	Максимальное время цикла (4 двоично-десятичных цифры) Сохраняется наибольшее время цикла от начала операции. Очищается не в конце, а в начале операции. Могут быть следующие единицы, в зависимости от установки 9F времени контроля (параметр DM 6618). По умолчанию: 0.1 мс; задание “10 ms”: 0.1 мс; задание “100 ms”: 1 мс; задание “1 s”: 10 мс	1.2.10
AR 15	00 - 15	Текущее время цикла (4 двоично-десятичных цифры) Сохраняется самое последнее время цикла во время операции. При остановке операции не очищается. Могут быть следующие единицы, в зависимости от установки 9F времени контроля (параметр DM 6618). По умолчанию: 0.1 мс; задание “10 ms”: 0.1 мс; задание “100 ms”: 1 мс; задание “1 s”: 10 мс	1.2.10

Приложение D

Применение функции часов

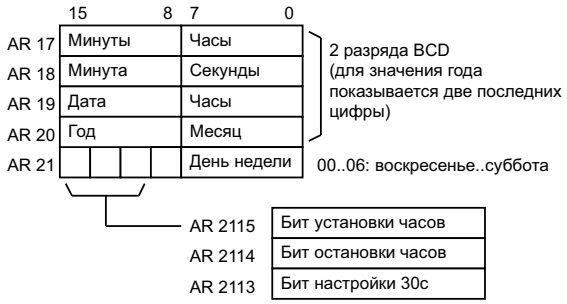
В CQM1 можно установить функцию часов, если установить кассету памяти с часами. Данное приложение описывает пользование часами.

Часами оборудованы два типа кассеты памяти:

CQM1-ME04R (EPROM) и CQM1-ME08R (EEPROM)

Слова , используемые с функцией часов

На рисунке показаны конфигурация слов AR 17..AR 21, используемые с часами. Данные слова при необходимости можно читать и использовать. AR 17 служит для быстрого получения часов и минут.



Установка времени

Для установки времени используйте периферийное устройство следующим образом:

При задании всех значений

1, 2, 3,... 1. Установите AR 2114 в 1 (бит остановки часов) для остановки часов.

2. Используя периферийное устройство, установите AR 18..AR 20 (минута/секунда, Дата/час, год/месяц) и AR 2100..AR 2107 (день недели).

3. Установите AR 2115 в 1 (бит установки часов), когда время будет таким, какое задано в шаге 2. Часы начнут идти с заданного времени, и бит останова часов автоматически сбросится. По завершении пуска часов бит AR 2115 также сбросится автоматически.

При задании только секунд

Используя AR 2114 также возможно просто задать секунды в 00 без сложной процедуры. Когда AR 2113 устанавливаются в 1, часы начинают идти следующим образом:

Если секунды заданы от 00 до 29, секунды сбросятся в 00, а значения минут останется старым.

Если секунды заданы от 30 до 59, секунды сбросятся в 00, а значения минут увеличатся на 1.

По завершении установки времени AR 2115 автоматически сбросятся в 0.

Замечание Время легко задается из меню с программатора и SSS. Подробности см. соответствующие инструкции.

для работы с программатором - Инструкцию по работе CQM1;

для работы с SSS - Инструкцию по работе с SSS: серия C.

Внимание! При удалении кассеты памяти из ПК часы остановятся и информация о текущих дате и времени будет утеряна.

Приложение E

Распределение входов/выходов

Наименование системы		Разработал	Проверил	Утвердил
Модель ПК	Лист N			

IR____	**Блок №**	**Модель**	**IR____**	**Блок №**	**Модель**
00			00		
01			01		
02			02		
03			03		
04			04		
05			05		
06			06		
07			07		
08			08		
09			09		
10			10		
11			11		
12			12		
13			13		
14			14		
15			15		

Приложение F

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 G

Мнемокод программы

Наименование системы		Разработал	Проверил	Утвердил
Модель ПК	Лист N			

No. FAL	Содержание	Меры по устранению
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		

Приложение H

Расширенные коды ASCII

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

Глоссарий

***DM**

См. Косвенный адрес (в области DM).

1:1 связь

Связь между двумя ПК, организованная таким образом, чтобы образовать общую область LR.

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.

AR область

Область памяти, выделанная для флагов и битов управления.

ASCII

American Standardcode For Information Interchahge. Служит для кодирования символов при выдачи 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.

CY

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 рабочих параметров.

DM область

Область данных (Data memery) 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 вводы 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(06).

FALS - ошибка

Признак ошибки, вызванный из программы при исполнении команды FALS(07) или вызванный системой.

FCS

См. КСК

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.

JIS

(Japanese Industrial Standards) - Японские промышленные стандарты.

LR область

Область данных, используемая при связи.

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.

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

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.

базовые команды

Основные команды лестничной диаграммы (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 имеет 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

В терминологии OMRON, сборочная единица ПК.

блок

См. Логический блок и блок команд

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 ЛД. 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 ЦУ 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 разъмам 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 (подпрогаммы)

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 ЛД 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 двоичные 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. 0.538×10^{-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.

И

Логическая операция над условиями, при которой результат является истиной, если только оба условия истины. В программировании лестничных диаграмм условия - это обычно состояния бит 1/0 или логическая комбинация таких состояний, называемая условиями исполнения.

ИЛИ

Логическая операция над условиями, при которой результат является истиной, если хотя бы одно или два условия истинны. В программировании ЛД условия - это обычно состояния битов (1 или 0) или логическая комбинация таких состояний, называемых условиями исполнения.

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 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 программировании ЛД 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 программировании ЛД 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 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 ЛД

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.

команда лестничной диаграммы

Команда, которая представляет условия на ЛД. Другие команды располагаются вдоль правой шины ЛД и называются выходными командами.

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).

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 исполнение на входе = 1.

командная линия

Группа условий, лежащих на одной горизонтальной линии ЛД. 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 (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.

ЛД

См. Лестничная диаграмма

лестничная диаграмма

Форма программирования, появившаяся из релейных систем управления, которая использует релейные символы для представления алгоритма работы программы. Внешний вид программы похож на лестницу, отсюда и названия.

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 ЛД, 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 (licker it)

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 программирования ЛД, 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 (NOT)

Логическая операция, инвертирующия 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 ЛД справа

прерывание (сигнал)

Сигнал, останавливающий нормальное исполнение программы и вызывающий подпрограмму или другие процессы отработки.

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 за груженной 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.

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 отрабатываться.

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: RPOGRAM, MONITOR, RUN.

резервная копия

Копия существующих данных, которая сохранится в случае искажения или потери исходных данных

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 лестничной диаграммы

Символ, используемый для изображения программы в виде ЛД.

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 ЦУ

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 служат 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 программы ЛД. 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 ЛД.

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 ЛД, которая управляет выходной командой. Каждому условию приписан в памяти бит, который определяет его состояние. Состояние бита определяет следующее условие исполнение. Условия соответствуют командам LOAD, LOAD NOT, AND, AND NOT, OR или OR NOT.

условие исполнения

Состояние 1 или 0, при которых исполняется команда. Условие исполнение задается логической комбинацией условий на одной командной линии лд (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, 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 ЦУ, включающий отработку ЛД, обслуживание периферийного порта, обновление входов/выходов и т. Д.

цикл исполнения

Повторяющийся цикл работы ЦУ, включающий отработку ЛД, обслуживание периферии, 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.

ЦУ

Центральное устройство контроллера. Устройство, способное хранить программу и данные, а также выполнять команды, содержащиеся в программе. В программируемом контроллере ЦУ выполняет программу, обрабатывает входные/выходные сигналы, осуществляет связь с другими устройствами и т. Д.

ЧС

Часовая стрелка (в сочетаниях: по ЧС и против ЧС)

шина

Линия связи, служащая для обмена данными между подключенными к ней блоками.

шина ЛД

Линия на лестничной диаграмме, идущая сверху вниз обычно с левой, а иногда и с правой стороны ЛД. Исполнение команд идет производится сверху вниз вдоль шины ЛД, 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.

ЭСППЗУ

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

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

Cat.No W228-E1-4 Минск, 1998