

Серия MELSEC FX

Программируемые логические контроллеры

**FX1S, FX1N,
FX2N, FX2NC,
FX3U**

К ЭТОМУ РУКОВОДСТВУ

Содержащиеся в этом руководстве тексты, изображения, диаграммы и примеры служат только для разъяснения установки, программирования и применения программируемых контроллеров MELSEC серий FX1S, FX1N, FX2N, FX2NC и FX3U.

Если возникнут вопросы относительно подключения и работы устройств, описанных в этом руководстве, сразу же вступайте в контакт с ближайшим филиалом MITSUBISHI ELECTRIC EUROPE B.V.

Без предварительного письменного разрешения, фирма MITSUBISHI ELECTRIC EUROPE B.V. не позволяет размножать никакие части этого руководства, сохранять в информационных системах или передавать другим лицам.

Фирма MITSUBISHI ELECTRIC EUROPE B.V. оставляет за собой право в любое время делать технические или иные изменения в этом руководстве без специальных указаний.

© 07/2008

УКАЗАНИЯ ПО БЕЗОПАСНОСТИ

ДОПУСК СПЕЦИАЛИСТОВ

Это руководство предназначено исключительно для знающих, имеющих специальное образование специалистов, которые знакомы со стандартами по безопасности техники автоматизации. Проектирование, подключение, ввод в эксплуатацию, обслуживание и проверка приборов должны выполняться только знающими, имеющими специальное образование специалистами, которые знакомы со стандартами по безопасности техники автоматизации.

ИСПОЛЬЗОВАНИЕ СОГЛАСНО НАЗНАЧЕНИЮ

Программируемые контроллеры серий FX1S, FX1N, FX2N, FX2NC и FX3U предназначены только для тех областей применения, которые описаны в этом руководстве. Обращайте внимание на соблюдение всех указанных в руководстве характеристик. Приборы разработаны, изготовлены, проверены и задокументированы с соблюдением норм безопасности. Неквалифицированные вмешательства в аппаратуру или программное обеспечение или несоблюдение предупреждений, содержащихся в этом руководстве или нанесенных на сам прибор, могут привести к серьезным травмам или материальному ущербу. В сочетании с программируемыми контроллерами серий FX1S, FX1N, FX2N, FX2NC и FX3U разрешается использовать только дополнительные и расширительные приборы, рекомендованные фирмой МИЦУБИСИ ЭЛЕКТРИК.

Все другие устройства, выходящие за рамки разрешенных, не должны применяться.

ПРЕДПИСАНИЯ ОТНОСЯЩИЕСЯ К БЕЗОПАСНОСТИ

При проектировании, подключении, вводе в эксплуатацию, обслуживании и проверке приборов нужно обращать внимание на предписания по безопасности и защите от несчастных случаев.

Особенно следует обращать внимание на указанные ниже предписания (не претендующие на полноту их охвата):

- VDE-предписания
 - VDE 0100
Работа в электроустановках с номинальным напряжением до 1000 В.
 - VDE 0105
Работа силовых установок.
 - VDE 0113
Электрические установки электрических аппаратов.
 - VDE 0160
Оборудование силовых установок и электрическими аппаратами.
 - VDE 0550/0551
Определения для трансформаторов.
 - VDE 0700
Безопасность электрических приборов для домашнего обихода и подобных целей.
 - VDE 0860
Определения безопасности для включаемых от сети электронных приборов и их принадлежностей для домашнего обихода и подобных целей.

- Предписания по пожарозащите
- Предписания по защите от несчастных случаев
 - VBG Nr.4
Электрические установки и аппараты

РАЗЪЯСНЕНИЯ К УКАЗАНИЯМ ПО ОПАСНОСТИ

В этом руководстве находятся указания, которые важны для надежной работы с описываемыми устройствами:



ОПАСНОСТЬ:

Означает, что имеется опасность для жизни и здоровья пользователя, если не придерживаться соответствующих мероприятий по безопасности.



ВНИМАНИЕ:

Означает предупреждение по возможному повреждению применяемых устройств или реальных ценностей, если не придерживаться соответствующих мероприятий по безопасности.

ОБЩИЕ УКАЗАНИЯ ПО ТЕХНИЧЕСКОЙ БЕЗОПАСНОСТИ И ЭКСПЛУАТАЦИИ

Следующие указания по опасности можно воспринимать как главные директивы для работы ПК совместно с другими устройствами. Эти указания при проектировании, установке и работе управляющих устройств безусловно должны приниматься во внимание.

**ОПАСНОСТЬ:**

- Необходимо учитывать предписания по безопасности и защиты от несчастных случаев, имеющих значение в специфических случаях применения устройств.
- Модули, конструктивные детали и приборы должны устанавливаться в защищенных от прикосновения корпусах с соответствующими крышками и защитными устройствами.
- Регулярно проверяйте подводящие напряжение кабели и провода, к которым присоединяются устройства на предмет повреждения изоляции или обрыва. При обнаружении повреждения в кабельных соединениях нужно немедленно обесточить устройство и заменить поврежденные кабельные соединения.
- Перед вводом в эксплуатацию проверьте соответствует ли напряжение сети допустимой области питающего напряжения.
- Предпримите необходимые меры, чтобы после отключения или пропадания напряжения прерванную программу можно было снова запустить. При этом даже кратковременно нельзя допускать опасного рабочего состояния.
- Устройства аварийного останова согласно VDE 0113 должно во всех режимах работы быть в работоспособном состоянии. Деблокировка устройства аварийного останова не может работать бесконтрольно или допускать самозапуск.
- Чтобы обрыв проводов или жил со стороны подвода сигналов не мог вызвать неопределенного состояния системы управления, должны приниматься соответствующие меры как по техническим, так и по программным средствам.

Содержание

1	ВВЕДЕНИЕ	
1.1	ОБЩИЕ ПОЛОЖЕНИЯ	1-1
1.2	ПРИМЕНЯЕМЫЕ ПРОГРАММАТОРЫ/ПРИБОРЫ ОБСЛУЖИВАНИЯ	1-2
2	ОСНОВНЫЕ ПОЛОЖЕНИЯ ПРОГРАММИРОВАНИЯ	
2.1	ОБРАБОТКА ПРОГРАММ В ПК	2-1
2.1.1	МЕТОД ОТОБРАЖЕНИЯ ПРОЦЕССА	2-2
2.1.2	РАЗЛИЧИЕ ОБРАБОТКИ УПРАВЛЯЮЩИХ СИГНАЛОВ ПО СРАВНЕНИЮ С ПРОВОДНОЙ СИСТЕМОЙ УПРАВЛЕНИЯ	2-3
2.2	УПРАВЛЯЮЩИЕ ИНСТРУКЦИИ	2-4
2.2.1	СТРУКТУРА УПРАВЛЯЮЩЕЙ ИНСТРУКЦИИ	2-4
2.2.2	ОПЕРАНДЫ	2-5
2.2.3	ВИДЫ ИЗОБРАЖЕНИЯ УПРАВЛЯЮЩИХ ИНСТРУКЦИЙ	2-6
2.2.4	ЛИСТИНГ РАСПРЕДЕЛЕНИЯ И ПОДКЛЮЧЕНИЕ ПК	2-8
3	ОПЕРАНДЫ	
3.1	ОБЗОР ОПЕРАНДОВ	3-1
3.2	ВХОДЫ И ВЫХОДЫ	3-2
3.2.1	АДРЕСАЦИЯ ВХОДОВ И ВЫХОДОВ	3-2
3.2.2	ПРОГРАММИРОВАНИЕ ВХОДОВ И ВЫХОДОВ	3-4
3.3	МЕРКЕРЫ	3-5
3.3.1	АДРЕСАЦИЯ МЕРКЕРОВ	3-5
3.3.2	ПРОГРАММИРОВАНИЕ МЕРКЕРОВ	3-6
3.4	ТАЙМЕР	3-7
3.4.1	АДРЕСАЦИЯ ТАЙМЕРА	3-7
3.4.2	ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ	3-8
3.4.3	ЗАДАНИЕ УСТАВКИ ВРЕМЕНИ С ПОМОЩЬЮ ИНТЕГРИРОВАННОГО ПОТЕНЦИОМЕТРА ДЛЯ ПК СЕРИИ FX1S И FX1N	3-9
3.4.4	ТОЧНОСТЬ ТАЙМЕРА	3-9
3.4.5	ТАЙМЕР С ПАМЯТЬЮ	3-10

3.5	СЧЕТЧИКИ	3-11
3.5.1	16-ТИ БИТНЫЙ СЧЕТЧИК	3-12
3.5.2	32-Х БИТНЫЙ СЧЕТЧИК	3-14
3.5.3	32-Х БИТНЫЙ ВЫСОКОСКОРОСТНОЙ СЧЕТЧИК	3-16
3.6	СОСТОЯНИЯ ШАГОВ	3-22
3.6.1	АДРЕСАЦИЯ ОПЕРАНДОВ СОСТОЯНИЯ ШАГОВ	3-22
3.7	КОНСТАНТЫ	3-23
3.7.1	ДЕСЯТИЧНЫЕ И ШЕСТНАДЦАТЕРИЧНЫЕ КОНСТАНТЫ	3-23
3.7.2	КОНСТАНТЫ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ	3-23
3.7.3	СТРОКОВЫЕ КОНСТАНТЫ	3-23
3.8	РЕГИСТРЫ	3-24
3.8.1	КЛАССИФИКАЦИЯ РЕГИСТРОВ	3-24
3.8.2	СТРУКТУРА РЕГИСТРА	3-25
3.8.3	АДРЕСАЦИЯ РЕГИСТРА	3-26
3.8.4	ИСПОЛЬЗОВАНИЕ СПЕЦИАЛЬНЫХ РЕГИСТРОВ	3-26
3.8.5	ВНЕШНЕ ИЗМЕНЯЕМЫЕ РЕГИСТРЫ	3-27
3.8.6	ПРИМЕНЕНИЕ ИНДЕКСНЫХ РЕГИСТРОВ	3-28
3.8.7	ПРИМЕНЕНИЕ ФАЙЛОВЫХ РЕГИСТРОВ	3-29
3.8.8	ПРЕДСТАВЛЕНИЕ ЧИСЕЛ	3-31
3.9	УКАЗАТЕЛЬ	3-37
3.9.1	АДРЕСАЦИЯ УКАЗАТЕЛЯ	3-37
3.9.2	УРОВЕНЬ ПОДПРОГРАММ (ВЛОЖЕННОСТИ)	3-37
3.10	ПРЕРЫВАНИЕ-УКАЗАТЕЛЬ	3-38
3.10.1	АДРЕСАЦИЯ ПРЕРЫВАНИЯ-УКАЗАТЕЛЯ	3-38
3.11	РАЗВЕТВЛЕНИЯ	3-43
3.11.1	АДРЕСАЦИЯ ОПЕРАНДОВ ПОДПРОГРАММ	3-43

4	НАБОР БАЗОВЫХ КОМАНД	
4.1	ОБЩИЕ УКАЗАНИЯ	4-1
4.1.1	ПОЯСНЕНИЯ К ТАБЛИЦАМ НАБОРА БАЗОВЫХ КОМАНД	4-1
4.2	НАЧАЛО ЛОГИЧЕСКИХ ЦЕПЕЙ (LD, LDI)	4-5
4.3	ВЫХОД РЕЗУЛЬТАТА ЛОГИЧЕСКИХ ЦЕПЕЙ (OUT)	4-6
4.4	ЛОГИЧЕСКАЯ СВЯЗЬ "И" (AND, ANI)	4-8
4.5	ЛОГИЧЕСКИЕ СВЯЗИ "ИЛИ" (OR, ORI)	4-10
4.6	ИМПУЛЬСНОЕ НАЧАЛО ЛОГИЧЕСКИХ СВЯЗЕЙ (LDP, LDF)	4-12
4.7	ИМПУЛЬСНЫЕ "И" ЛОГИЧЕСКИЕ СВЯЗИ (ANP, ANF)	4-14
4.8	ИМПУЛЬСНЫЕ "ИЛИ" ЛОГИЧЕСКИЕ СВЯЗИ (ORP, ORF)	4-16
4.9	ЛОГИЧЕСКИЕ СВЯЗИ "И-БЛОКИ" (ANB)	4-18
4.10	ЛОГИЧЕСКИЕ СВЯЗИ "ИЛИ-БЛОКИ" (ORB)	4-19
4.11	ОБРАБОТКА РЕЗУЛЬТАТОВ ЛОГИЧЕСКИХ СВЯЗЕЙ (MPS, MRD, MPP)	4-20
4.12	ВКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ УСЛОВИЙ КОНТРОЛЯ (MC, MCR)	4-23
4.13	ВКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ ОПЕРАНДОВ (SET, RST)	4-26
4.14	ГЕНЕРАЦИЯ ИМПУЛЬСА (PLS, PLF)	4-28
4.15	ИНВЕРСИЯ РЕЗУЛЬТАТА ОБРАБОТКИ (INV)	4-30
4.16	ПУСТАЯ СТРОКА В ПРОГРАММЕ (NOP)	4-31
4.17	КОНЕЦ ПРОГРАММЫ ПК (END)	4-32
4.18	ПРИМЕРЫ ПРОГРАММИРОВАНИЯ	4-33
4.18.1	ОПРОС ВХОДОВ	4-34
5	STL-ИНСТРУКЦИИ	
5.1	ОБЩИЕ УКАЗАНИЯ	5-1
5.1.1	ПРИМЕР ПРИМЕНЕНИЯ STL-ИНСТРУКЦИИ	5-2
5.1.2	СХЕМАТИЧНЫЙ ПРОЦЕСС ШАГОВОГО УПРАВЛЕНИЯ	5-3
5.1.3	ПРЕДСТАВЛЕНИЕ УПРАВЛЕНИЯ ПРОЦЕССОМ В ДИАГРАММЕ БЛОКОВ	5-4
5.2	ПРОГРАММИРОВАНИЕ STL-ИНСТРУКЦИИ	5-5
5.3	ИНИЦИАЛИЗАЦИЯ СОСТОЯНИЯ ШАГА	5-10
5.4	STL-РАЗВЕТВЛЕНИЯ	5-11

5.4.1	ПРОСТОЙ (ЛИНЕЙНЫЙ) ПРОЦЕСС	5-11
5.4.2	СЕЛЕКТИВНОЕ РАЗВЕТВЛЕНИЕ	5-12
5.4.3	ПАРАЛЛЕЛЬНОЕ РАЗВЕТВЛЕНИЕ	5-14
5.4.4	КОМБИНАЦИЯ ИЗ СЕЛЕКТИВНОГО И ПАРАЛЛЕЛЬНОГО РАЗВЕТВЛЕНИЯ	5-16
5.4.5	ПРОГРАММИРОВАНИЕ ПУСТОГО СОСТОЯНИЯ	5-17
5.4.6	ПЕРЕХОДНОЕ РАЗВЕТВЛЕНИЕ	5-18
5.5	ПРИМЕР КОНТРОЛЯ ЗАГРУЗКИ И РАЗГРУЗКИ	5-20
5.6	ПРИМЕР ТРАНСПОРТИРОВКИ И СОРТИРОВКИ	5-22

6 ПРИКЛАДНЫЕ ИНСТРУКЦИИ

6.1	ОБЩИЕ УКАЗАНИЯ	6-1
6.1.1	ПОЯСНЕНИЯ К ОПИСАНИЮ СЛОВНЫХ ИНСТРУКЦИЙ	6-1
6.1.2	ОПИСАНИЕ ОПЕРАНДОВ	6-2
6.1.3	ОБЪЕДИНЕНИЕ БИТОВЫХ ОПЕРАНДОВ	6-3
6.1.4	СТРУКТУРА ДАННЫХ	6-5
6.1.5	ИСПОЛНЕНИЕ СЛОВНЫХ ИНСТРУКЦИЙ	6-6
6.1.6	ПРИМЕНЕНИЕ ИНДЕКСНЫХ РЕГИСТРОВ V, Z	6-7
6.1.7	ЗНАЧЕНИЯ ФЛАГОВ	6-8
6.1.8	ОШИБКИ РАБОТЫ ПРОГРАММЫ ПРИ ВЫПОЛНЕНИИ СЛОВНЫХ ИНСТРУКЦИЙ	6-8
6.1.9	32-Х БИТНЫЕ ИНСТРУКЦИИ	6-8
6.1.10	ОБЗОР СЛОВНЫХ ИНСТРУКЦИЙ	6-9
6.2	ИНСТРУКЦИИ ПРОЦЕССА РАБОТЫ ПРОГРАММЫ	6-11
6.2.1	ПЕРЕХОД ВНУТРИ ПРОГРАММЫ (CJ)	6-12
6.2.2	ВЫЗОВ ПОДПРОГРАММ (CALL)	6-16
6.2.3	КОНЕЦ ПОДПРОГРАММЫ (SRET)	6-17
6.2.4	ПРИМЕНЕНИЕ ПРЕРЫВАНИЯ ПРОГРАММЫ (IRET, EI, DI)	6-18
6.2.5	КОНЕЦ ОБЛАСТИ ПОДПРОГРАММЫ (FEND)	6-22
6.2.6	ОБНОВЛЕНИЕ КОНТРОЛЯ ВРЕМЕНИ ЦИКЛА (WDT)	6-23
6.2.7	ПОВТОРЕНИЕ ЧАСТЕЙ ПРОГРАММЫ (FOR, NEXT)	6-25
6.3	ИНСТРУКЦИИ СРАВНЕНИЯ И ПЕРЕДАЧИ	6-27
6.3.1	СРАВНЕНИЕ ЧИСЛОВЫХ ДАННЫХ (CMP, DCMP)	6-28
6.3.2	СРАВНЕНИЕ С БЛОКОМ ЧИСЛОВЫХ ДАННЫХ (ZCP, DZCP)	6-30
6.3.3	ПЕРЕДАЧА ДАННЫХ (MOV, DMOV)	6-32
6.3.4	ПЕРЕДАЧА И СМЕЩЕНИЕ ДАННЫХ (SMOV)	6-33

6.3.5	ПЕРЕСЫЛКА С ИНВЕРТИРОВАНИЕ (CML)	6-36
6.3.6	ПЕРЕДАЧА БЛОКА (BMOV)	6-37
6.3.7	ПЕРЕДАЧА ОДИНАКОВЫХ ДАННЫХ (FMOV)	6-38
6.3.8	ОБМЕН ДАННЫМИ (XCH)	6-39
6.3.9	BСD-КОНВЕРТИРОВАНИЕ (BCD, DBCD)	6-41
6.3.10	ДВОИЧНОЕ КОНВЕРТИРОВАНИЕ (BIN, DBIN)	6-43
6.4	АРИФМЕТИЧЕСКИЕ ИНСТРУКЦИИ	6-45
6.4.1	СЛОЖЕНИЕ ЧИСЛОВЫХ ДАННЫХ (ADD, DADD)	6-45
6.4.2	ВЫЧИТАНИЕ ЧИСЛОВЫХ ДАННЫХ (SUB, DSUB)	6-47
6.4.3	УМНОЖЕНИЕ ЧИСЛОВЫХ ДАННЫХ (MUL, DMUL)	6-49
6.4.4	ДЕЛЕНИЕ ЧИСЛОВЫХ ДАННЫХ (DIV, DDIV)	6-51
6.4.5	ИНСТРУКЦИЯ ПРИРАЩЕНИЯ (INC, DINC)	6-53
6.4.6	ИНСТРУКЦИЯ УМЕНЬШЕНИЯ (DEC, DDEC)	6-54
6.4.7	ЛОГИЧЕСКАЯ СВЯЗЬ “И” ДВОИЧНЫХ ДАННЫХ (WAND, DAND)	6-55
6.4.8	ЛОГИЧЕСКАЯ СВЯЗЬ “ИЛИ” ДВОИЧНЫХ ДАННЫХ (WOR, DWOR)	6-56
6.4.9	ЛОГИЧЕСКАЯ СВЯЗЬ “ИСКЛЮЧАЮЩЕГО ИЛИ” ДВОИЧНЫХ ДАННЫХ (WXOR, DXOR)	6-57
6.4.10	ИНВЕРСИЯ ДАННЫХ (NEG)	6-58
6.5	ИНСТРУКЦИИ СДВИГОВ	6-59
6.5.1	СДВИГ ВПРАВО (ROR)	6-60
6.5.2	СДВИГ ВЛЕВО (ROL)	6-61
6.5.3	КОЛЬЦЕВОЙ СДВИГ ВПРАВО (RCR)	6-62
6.5.4	КОЛЬЦЕВОЙ СДВИГ ВЛЕВО (RCL)	6-63
6.5.5	ПОБИТНЫЙ СДВИГ ДВОИЧНЫХ ДАННЫХ (SFTR, SFTL)	6-64
6.5.6	ПОСЛОВНЫЙ СДВИГ ДАННЫХ ВПРАВО (WSFR)	6-66
6.5.7	ПОСЛОВНЫЙ СДВИГ ДАННЫХ ВЛЕВО (WSFL)	6-67
6.5.8	ЗАПИСЬ В СТЕКОВУЮ ПАМЯТЬ ТИПА FIFO (SFWR)	6-68
6.5.9	ЧТЕНИЕ ИЗ СТЕКОВОЙ ПАМЯТИ ТИПА FIFO (SFRD)	6-69
6.6	ОПЕРАЦИИ С ДАННЫМИ	6-71
6.6.1	ОТКЛЮЧИТЬ ОБЛАСТЬ ОПЕРАНДОВ (ZRST)	6-72
6.6.2	ДЕКОДИРОВАНИЕ ДАННЫХ (DECO)	6-73
6.6.3	КОДИРОВАНИЕ ДАННЫХ (ENCO)	6-75
6.6.4	СУММА АКТИВНЫХ БИТ (SUM)	6-77
6.6.5	ПРОВЕРКА БИТОВ (BON)	6-78
6.6.6	ВЫЧИСЛЕНИЕ СРЕДНЕГО ЗНАЧЕНИЯ (MEAN)	6-79
6.6.7	ЗАПУСК ИНТЕРВАЛОВ ВРЕМЕНИ (ANS)	6-80
6.6.8	ОТКЛЮЧЕНИЕ БИТА ОТОБРАЖЕНИЯ (ANR)	6-81

6.6.9	ВЫЧИСЛЕНИЕ КОРНЯ КВАДРАТНОГО (SQR)	6-82
6.6.10	ПРЕОБРАЗОВАНИЕ ФОРМАТОВ ЧИСЕЛ (FLT)	6-83
6.7	ВЫСОКОСКОРОСТНЫЕ ИНСТРУКЦИИ	6-84
6.7.1	ОБНОВЛЕНИЕ ВХОДОВ И ВЫХОДОВ (REF)	6-85
6.7.2	УСТАНОВКА ВХОДНОГО ФИЛЬТРА (REFF)	6-87
6.7.3	ЧТЕНИЕ МАТРИЦЫ (MTR)	6-89
6.7.4	СИГНАЛ ВКЛЮЧЕНИЯ И ОТКЛЮЧЕНИЯ ОТ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА (DHSCS, DHSCR)	6-92
6.7.5	СРАВНЕНИЕ ОБЛАСТЕЙ (DHSZ)	6-94
6.7.6	ОПРЕДЕЛЕНИЕ СКОРОСТИ (SPD)	6-99
6.7.7	ВЫДАЧА ОПРЕДЕЛЕННОГО ЧИСЛА ИМПУЛЬСОВ (PLSY, DPLSY)	6-101
6.7.8	ВЫДАЧА ИМПУЛЬСОВ С МОДУЛЯЦИЕЙ ШИРИНЫ ИМПУЛЬСА (PWM)	6-103
6.7.9	ВЫДАЧА ОПРЕДЕЛЕННОГО ЧИСЛА ИМПУЛЬСОВ (PLSR)	6-105
6.8	ИНСТРУКЦИИ ПОЛЬЗОВАТЕЛЯ	6-107
6.8.1	ИНИЦИАЛИЗАЦИЯ СТАТУСА ШАГА (IST)	6-108
6.8.2	ИНСТРУКЦИЯ ПОИСКА (SER)	6-114
6.8.3	АБСОЛЮТНЫЙ МНОГОУСТАВОЧНЫЙ СЧЕТЧИК (ABSD)	6-116
6.8.4	ИНКРЕМЕНТАЛЬНЫЙ МНОГОУСТАВОЧНЫЙ СЧЕТЧИК (INCD)	6-118
6.8.5	ОБУЧАЮЩИЙСЯ ТАЙМЕР (TTMR)	6-120
6.8.6	СПЕЦИАЛЬНЫЙ ТАЙМЕР (STMR)	6-121
6.8.7	ТРИГГЕРНАЯ ФУНКЦИЯ (FLIP-FLOP) (ALT)	6-122
6.8.8	ФУНКЦИЯ НАКЛОННЫХ ХАРАКТЕРИСТИК (RAMP)	6-124
6.8.9	ПОЗИЦИОНИРОВАНИЕ ПОВОРОТНОГО СТОЛА (ROTC)	6-126
6.8.10	ИНСТРУКЦИЯ СОРТИРОВКИ (SORT)	6-129

7 СПЕЦИАЛЬНЫЕ КОМАНДЫ

7.1	ОБЩИЕ УКАЗАНИЯ	7-1
7.1.1	ОБЗОР СПЕЦИАЛЬНЫХ КОМАНД	7-1
7.2	ИНСТРУКЦИИ ВВОДА/ВЫВОДА	7-5
7.2.1	ДЕСЯТИЧНАЯ КЛАВИАТУРА (ТКУ)	7-6
7.2.2	ШЕСТНАДЦАТИРИЧНАЯ КЛАВИАТУРА (НКУ)	7-8
7.2.3	ЦИФРОВОЙ ПЕРЕКЛЮЧАТЕЛЬ (DSW)	7-11
7.2.4	7-МИ СЕГМЕНТНЫЙ ЦИФРОВОЙ ИНДИКАТОР (SEGD)	7-13
7.2.5	7-МИ СЕГМЕНТНЫЙ ЦИФРОВОЙ ИНДИКАТОР С ЗАПОМИНАНИЕМ (SEGL)	7-14
7.2.6	7-МИ СЕГМЕНТНЫЙ ИНДИКАТОР С ДОПОЛНИТЕЛЬНЫМИ КЛАВИШАМИ (ARWS)	7-17

7.2.7	ASCII-КОНВЕРТИРОВАНИЕ (ASC)	7-19
7.2.8	ВЫДАЧА ДАННЫХ ПО ВЫХОДАМ (PR)	7-21
7.3	ОБМЕН ДАННЫМИ СО СПЕЦИАЛЬНЫМИ МОДУЛЯМИ	7-23
7.3.1	ЧТЕНИЕ ДАННЫХ ИЗ СПЕЦИАЛЬНОГО МОДУЛЯ (FROM)	7-25
7.3.2	ЗАПИСЬ ДАННЫХ В СПЕЦИАЛЬНЫЙ МОДУЛЬ (TO)	7-27
7.4	КОММУНИКАЦИОННЫЕ КОМАНДЫ	7-29
7.4.1	ПРИЕМ И ПЕРЕДАЧА ДАННЫХ (RS)	7-30
7.4.2	ПЕРЕНОС ВХОДОВ ИЛИ МЕРКЕРОВ (PRUN)	7-36
7.4.3	ASCII-ПРЕОБРАЗОВАНИЕ (ASCII)	7-37
7.4.4	ШЕСТНАДЦАТЕРИЧНОЕ -ПРЕОБРАЗОВАНИЕ (HEX)	7-39
7.4.5	ПРОВЕРКА НА СУММУ И ПАРИТЕТ (CCD)	7-41
7.4.6	ЧТЕНИЕ ЗНАЧЕНИЯ ИЗ FX1N-8AV-BD И FX2N-8AV-BD (VRRD)	7-43
7.4.7	ЧТЕНИЕ ПОЛОЖЕНИЯ ПЕРЕКЛЮЧАТЕЛЯ ИЗ FX1N-8AV-BD И FX2N-8AV-BD (VRSC)	7-44
7.4.8	ПОСЛЕДОВАТЕЛЬНАЯ ПЕРЕДАЧА ДАННЫХ (RS2)	7-45
7.5	ПРОГРАММИРОВАНИЕ ЗАМКНУТОГО КОНТУРА РЕГУЛИРОВАНИЯ (PID)	7-47
7.6	ОБМЕН ДАННЫХ С ИНДЕКСНЫМ РЕГИСТРОМ	7-53
7.6.1	СОХРАНЕНИЕ СОДЕРЖИМОГО ИНДЕКСНЫХ РЕГИСТРОВ (ZPUSH)	7-53
7.6.2	ВОССТАНОВЛЕНИЕ СОДЕРЖИМОГО ИНДЕКСНЫХ РЕГИСТРОВ (ZPOP)	7-56
7.7	ИНСТРУКЦИИ НАД ЧИСЛАМИ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (FX2N)	7-57
7.7.1	СРАВНЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DECMR)	7-58
7.7.2	ЗОННЫЙ КОМПАРАТОР (DEZCP)	7-59
7.7.3	ПЕРЕНОС ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEMOV)	7-61
7.7.4	ПРЕОБРАЗОВАНИЕ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В СТРОКОВУЮ ВЕЛИЧИНУ (DESTR)	7-62
7.7.5	ПРЕОБРАЗОВАНИЕ СТРОКОВОЙ ВЕЛИЧИНЫ В ЧИСЛО С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEVAL)	7-70
7.7.6	ПРЕОБРАЗОВАНИЕ ФОРМАТА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В НАУЧНЫЙ ФОРМАТ ЧИСЕЛ (DEBCD)	7-75
7.7.7	ПРЕОБРАЗОВАНИЕ ИЗ НАУЧНОГО ФОРМАТА ЧИСЕЛ В ФОРМАТ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEBIN)	7-76
7.7.8	СЛОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEADD)	7-77
7.7.9	ВЫЧИТАНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DESUB)	7-78
7.7.10	УМНОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEMUL)	7-79
7.7.11	ДЕЛЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEDIV)	7-80

7.7.12	ПРЕДСТАВЛЕНИЕ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В ВИДЕ ПОКАЗАТЕЛЯ СТЕПЕНИ ПО ОСНОВАНИЮ E (DEXP)	7-81
7.7.13	ВЫЧИСЛЕНИЕ НАТУРАЛЬНОГО ЛОГАРИФМА (DLOGE)	7-83
7.7.14	ВЫЧИСЛЕНИЕ ДЕСЯТИЧНОГО ЛОГАРИФМА (DLOG10)	7-85
7.7.15	КОРЕНЬ КВАДРАТНЫЙ ИЗ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DESQR)	7-87
7.7.16	ИЗМЕНЕНИЕ АРИФМЕТИЧЕСКОГО ЗНАКА ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DENEG)	7-88
7.7.17	ПРЕОБРАЗОВАНИЕ ФОРМАТА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В ЦЕЛОЕ ЧИСЛО (INT)	7-89
7.7.18	РАСЧЕТ СИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DSIN)	7-90
7.7.19	РАСЧЕТ КОСИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DCOS)	7-91
7.7.20	РАСЧЕТ ТАНГЕНСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DTAN)	7-92
7.7.21	ВЫЧИСЛЕНИЕ АРКСИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DASIN)	7-93
7.7.22	ВЫЧИСЛЕНИЕ АРККОСИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DACOS)	7-95
7.7.23	ВЫЧИСЛЕНИЕ АРКТАНГЕНСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DATAN)	7-97
7.7.24	ПЕРЕСЧЕТ ИЗ ГРАДУСОВ В РАДИАНЫ (DRAD)	7-99
7.7.25	ПЕРЕСЧЕТ ИЗ РАДИАН В ГРАДУСЫ (DDEG)	7-101
7.8	ИНСТРУКЦИИ ПО ОБРАБОТКЕ ДАННЫХ	7-103
7.8.1	СУММИРОВАНИЕ СОДЕРЖИМОГО СЛОВНЫХ ОПЕРАНДОВ (WSUM)	7-104
7.8.2	РАЗДЕЛЕНИЕ ДАННЫХ В СЛОВНЫХ ОПЕРАНДАХ НА БАЙТЫ (WTOB)	7-106
7.8.3	ОБРАЗОВАНИЕ СЛОВНОГО ОПЕРАНДА ИЗ ОТДЕЛЬНЫХ БАЙТОВ (VTOW)	7-108
7.8.4	ОБЪЕДИНЕНИЕ ГРУПП ПО 4 БИТА В СЛОВНЫЙ ОПЕРАНД (UNI)	7-110
7.8.5	РАЗДЕЛЕНИЕ СЛОВНОГО ОПЕРАНДА НА ГРУППЫ ПО 4 БИТА (DIS)	7-112
7.8.6	ОБМЕН СТАРШИМИ И МЛАДШИМИ БАЙТАМИ (SWAP)	7-114
7.8.7	СОРТИРОВКА ДАННЫХ В ТАБЛИЦЕ (SORT2)	7-115
7.9	КОМАНДЫ ПОЗИЦИОНИРОВАНИЯ	7-118
7.9.1	УКАЗАНИЯ ПО ПРИМЕНЕНИЮ КОМАНД ПОЗИЦИОНИРОВАНИЯ ДЛЯ КОНТРОЛЛЕРОВ FX1S И FX1N	7-118
7.9.2	ФОРМАТ ВЫХОДНЫХ ИМПУЛЬСОВ	7-119
7.9.3	ОПЕРАНДЫ ИСПОЛЬЗУЕМЫЕ ПРИ ПОЗИЦИОНИРОВАНИИ	7-119
7.9.4	ПРИМЕР ПОДКЛЮЧЕНИЯ СЕРВОПРИВОДА	7-121

7.9.5	ПРИМЕР ПРОГРАММЫ	7-122
7.9.6	ДВИЖЕНИЕ В БАЗОВУЮ ТОЧКУ С ИСПОЛЬЗОВАНИЕМ ВЫКЛЮЧАТЕЛЯ ПОЗИЦИОНИРОВАНИЯ (DSZR)	7-129
7.9.7	ПОЗИЦИОНИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ПРЕРЫВАНИЯ (DVIT)	7-130
7.9.8	ПОЗИЦИОНИРОВАНИЕ ПО ТАБЛИЦЕ ДАННЫХ (TBL)	7-131
7.9.9	ЧТЕНИЕ АБСОЛЮТНОЙ ТЕКУЩЕЙ ПОЗИЦИИ (DABS)	7-132
7.9.10	ПЕРЕМЕЩЕНИЕ К НУЛЕВОЙ (ИСХОДНОЙ) ТОЧКЕ (ZRN)	7-133
7.9.11	ВЫДАЧА ИМПУЛЬСОВ С ПЕРЕМЕННОЙ ЧАСТОТОЙ (PLSV)	7-135
7.9.12	ПОЗИЦИОНИРОВАНИЕ ПО ИНКРЕМЕНТАЛЬНЫМ ЗНАЧЕНИЯМ (DRVI)	7-137
7.9.13	ПОЗИЦИОНИРОВАНИЕ ПО АБСОЛЮТНЫМ ЗНАЧЕНИЯМ (DRVA)	7-139
7.10	ИНСТРУКЦИИ РЕАЛЬНОГО ВРЕМЕНИ	7-141
7.10.1	СРАВНЕНИЕ ДАННЫХ ВРЕМЕНИ (TCMP)	7-142
7.10.2	СРАВНЕНИЕ ДАННЫХ ВРЕМЕНИ В ОБЛАСТИ (TZCP)	7-144
7.10.3	СЛОЖЕНИЕ ДАННЫХ ВРЕМЕНИ (TADD)	7-146
7.10.4	ВЫЧИТАНИЕ ДАННЫХ ВРЕМЕНИ (TSUB)	7-148
7.10.5	ПРЕОБРАЗОВАНИЕ ЕДИНИЦЫ “ЧАСЫ” В “СЕКУНДЫ” (HTOS)	7-150
7.10.6	ПРЕОБРАЗОВАНИЕ ЕДИНИЦЫ ИЗ “СЕКУНД” В “ЧАСЫ” (STON)	7-152
7.10.7	ЧТЕНИЕ ДАННЫХ ВРЕМЕНИ (TRD)	7-154
7.10.8	ЗАПИСЬ ДАННЫХ ВРЕМЕНИ (TWR)	7-155
7.10.9	СЧЕТ РАБОЧИХ ЧАСОВ (HOUR)	7-156
7.11	ИНСТРУКЦИИ КОДА ГРЕЯ	7-158
7.11.1	ПРЕОБРАЗОВАНИЕ ЦЕЛОГО ЧИСЛА В КОД ГРЕЯ (GRY)	7-158
7.11.2	ПРЕОБРАЗОВАНИЕ КОДА ГРЕЯ В ЦЕЛОЕ ЧИСЛО (GBIN)	7-159
7.12	ОБМЕН ДАННЫМИ С АНАЛОГОВЫМИ МОДУЛЯМИ	7-160
7.12.1	ЧТЕНИЕ АНАЛОГОВЫХ ВХОДНЫХ ЗНАЧЕНИЙ (RD3A)	7-160
7.12.2	ЗАПИСЬ АНАЛОГОВЫХ ВЫХОДНЫХ ЗНАЧЕНИЙ (WR3A)	7-161
7.13	ВНЕШНЯЯ КАССЕТА ПАМЯТИ	7-162
7.13.1	ФУНКЦИИ ВНЕШНЕЙ ROM КАССЕТЫ (EXTR)	7-162
7.14	РАЗНЫЕ КОМАНДЫ	7-183
7.14.1	СЧИТЫВАНИЕ КОММЕНТАРИЯ К ОПЕРАНДУ (COMRD)	7-183
7.14.2	ГЕНЕРИРОВАНИЕ СЛУЧАЙНОГО ЧИСЛА (RND)	7-186
7.14.3	ТАКТОВЫЙ ГЕНЕРАТОР С РЕГУЛИРУЕМОЙ СКВАЖНОСТЬЮ ИМПУЛЬСОВ (DUTY)	7-187
7.14.4	ПРОВЕРКА ДАННЫХ (CRC)	7-188
7.14.5	ПЕРЕДАЧА ФАКТИЧЕСКОГО ЗНАЧЕНИЯ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА (DHCMOV)	7-191

7.15	КОМАНДЫ ДЛЯ БЛОКОВ ДАННЫХ	7-195
7.15.1	СЛОЖЕНИЕ ДАННЫХ ДВУХ БЛОКОВ (BK+)	7-196
7.15.2	ВЫЧИСЛЕНИЕ РАЗНОСТИ ДАННЫХ ДВУХ БЛОКОВ (BK-)	7-199
7.15.3	СРАВНЕНИЕ ДАННЫХ В БЛОКАХ ДАННЫХ (BKCMP■)	7-202
7.16	КОМАНДЫ ДЛЯ РАБОТЫ СО СТРОКОВЫМИ ВЕЛИЧИНАМИ	7-207
7.16.1	ПРЕОБРАЗОВАНИЕ ДВОИЧНЫХ ДАННЫХ В СТРОКОВЫЕ (STR)	7-208
7.16.2	ПРЕОБРАЗОВАНИЕ СТРОКОВЫХ ДАННЫХ В ДВОИЧНЫЕ (VAL)	7-213
7.16.3	ОБЪЕДИНЕНИЕ СТРОКОВЫХ ВЕЛИЧИН (\$+)	7-218
7.16.4	ОПРЕДЕЛЕНИЕ ДЛИНЫ СТРОКОВОЙ ВЕЛИЧИНЫ (LEN)	7-220
7.16.5	ФРАГМЕНТ СТРОКОВОЙ ВЕЛИЧИНЫ СПРАВА (RIGHT)	7-222
7.16.6	ФРАГМЕНТ СТРОКОВОЙ ВЕЛИЧИНЫ СЛЕВА (LEFT)	7-225
7.16.7	КОПИРОВАНИЕ ЗНАКОВ ИЗ СТРОКОВОЙ ВЕЛИЧИНЫ (MIDR)	7-228
7.16.8	ЗАМЕНА ЦЕПОЧКИ СИМВОЛОВ (MIDW)	7-231
7.16.9	ПОИСК ЦЕПОЧКИ СИМВОЛОВ (INSTR)	7-235
7.16.10	ПЕРЕДАЧА ЦЕПОЧКИ СИМВОЛОВ (\$MOV)	7-237
7.17	КОМАНДЫ ДЛЯ РАБОТЫ С ПЕРЕЧНЯМИ ДАННЫХ	7-239
7.17.1	СТИРАНИЕ ДАННЫХ ИЗ ПЕРЕЧНЯ ДАННЫХ (FDEL)	7-239
7.17.2	ВВОД ДАННЫХ В ПЕРЕЧЕНЬ ДАННЫХ (FINS)	7-241
7.17.3	СЧИТЫВАНИЕ ДАННЫХ, ВВЕДЕННЫХ В ПЕРЕЧЕНЬ ПОСЛЕДНИМИ (POP)	7-243
7.17.4	ПЕРЕМЕЩЕНИЕ 16-БИТНОГО СЛОВА ДАННЫХ ВПРАВО (SFR)	7-246
7.17.5	ПЕРЕМЕЩЕНИЕ 16-БИТНОГО СЛОВА ДАННЫХ ВЛЕВО (SFL)	7-248
7.18	ИНСТРУКЦИИ СРАВНЕНИЯ II	7-250
7.18.1	ЗАГРУЗКА СРАВНЕНИЯ (LD□)	7-251
7.18.2	ЗАГРУЗКА СРАВНЕНИЯ (ANDI)	7-252
7.18.3	ЗАГРУЗКА СРАВНЕНИЯ (ORI)	7-254
7.19	ОБЗОР КОМАНД КОНТРОЛЯ ДАННЫХ	7-255
7.19.1	ОГРАНИЧЕНИЕ ДИАПАЗОНА ВЫВОДА ЗНАЧЕНИЙ (LIMIT)	7-256
7.19.2	УСТАНОВКА СМЕЩЕНИЯ ВХОДА (BAND)	7-259
7.19.3	УСТАНОВКА СМЕЩЕНИЯ ВЫХОДА (ZONE)	7-262
7.19.4	МАСШТАБИРОВАНИЕ ЗНАЧЕНИЙ (SCL)	7-265
7.19.5	ПРЕОБРАЗОВАНИЕ ЧИСЛА В КОДИРОВКЕ ASCII В ДВОИЧНОЕ ЗНАЧЕНИЕ (DABIN)	7-269
7.19.6	ПРЕОБРАЗОВАНИЕ ДВОИЧНОГО ЗНАЧЕНИЯ В КОД ASCII (BINDA)	7-272
7.19.7	МАСШТАБИРОВАНИЕ ЗНАЧЕНИЙ (SCL2)	7-275

7.20	ОБМЕН ДАННЫМИ С ПРЕОБРАЗОВАТЕЛЯМИ ЧАСТОТЫ	7-279
7.20.1	ПРОВЕРКА СОСТОЯНИЯ ПРЕОБРАЗОВАТЕЛЯ ЧАСТОТЫ (IVCK)	7-280
7.20.2	УПРАВЛЕНИЕ ПРЕОБРАЗОВАТЕЛЕМ ЧАСТОТЫ (IVDR)	7-281
7.20.3	СЧИТЫВАНИЕ ПАРАМЕТРОВ ПРЕОБРАЗОВАТЕЛЯ ЧАСТОТЫ (IVRD)	7-282
7.20.4	ЗАПИСЬ ПАРАМЕТРОВ В ПРЕОБРАЗОВАТЕЛЬ ЧАСТОТЫ (IVWR)	7-282
7.20.5	ПОБЛОЧНАЯ ЗАПИСЬ ПАРАМЕТРОВ В ПРЕОБРАЗОВАТЕЛЬ ЧАСТОТЫ (IVBWR)	7-283
7.21	ОБМЕН ДАННЫМИ СО СПЕЦИАЛЬНЫМИ МОДУЛЯМИ	7-284
7.21.1	СЧИТЫВАНИЕ ИЗ БУФЕРНОЙ ПАМЯТИ СПЕЦИАЛЬНЫХ МОДУЛЕЙ (RVFM)	7-284
7.21.2	ЗАПИСЬ В БУФЕРНУЮ ПАМЯТЬ СПЕЦИАЛЬНЫХ МОДУЛЕЙ (WBFM)	7-287
7.21.3	УКАЗАНИЯ ПО ПРИМЕНЕНИЮ КОМАНД RVFM И WBFM	7-288
7.21.4	ПРИМЕР ПРОГРАММЫ ДЛЯ КОМАНД RVFM И WBFM	7-290
7.22	КОМАНДА ДЛЯ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА	7-291
7.23	КОМАНДЫ ДЛЯ РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ	7-295
7.23.1	СЧИТЫВАНИЕ ДАННЫХ ИЗ РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ (LOADR)	7-295
7.23.2	ЗАПИСЬ ДАННЫХ В РАСШИРЕННЫЕ ФАЙЛОВЫЕ РЕГИСТРЫ (SAVER)	7-297
7.23.3	ИНИЦИАЛИЗАЦИЯ РАСШИРЕННЫХ РЕГИСТРОВ И РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ (INITR)	7-302
7.23.4	СОХРАНЕНИЕ ЗНАЧЕНИЙ ОПЕРАНДОВ В РАСШИРЕННЫХ РЕГИСТРАХ И РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРАХ (LOGR)	7-305
7.23.5	ПЕРЕДАЧА ДАННЫХ ИЗ РАСШИРЕННЫХ РЕГИСТРОВ В РАСШИРЕННЫЕ ФАЙЛОВЫЕ РЕГИСТРЫ (RWER)	7-310
7.23.6	ИНИЦИАЛИЗАЦИЯ РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ (INITER)	7-313

8 СПЕЦИАЛЬНЫЕ ФУНКЦИИ

8.1	СОХРАНЕНИЕ ДАННЫХ В РЕЖИМЕ “СТОП”	8-2
8.2	РАБОТА С ПОСТОЯННЫМ ВРЕМЕНЕМ ЦИКЛА ПРОГРАММЫ (СКАНОМ)	8-3
8.3	ФУНКЦИИ КОДА РАЗРЕШЕНИЯ РАБОТЫ	8-4
8.4	ФУНКЦИИ ФИКСАЦИИ ИМПУЛЬСОВ	8-5
8.5	НАСТРОЙКА ВХОДНОГО ФИЛЬТРА	8-7
8.5.1	СЕРИИ FX1S И FX1N	8-7
8.5.2	СЕРИИ FX2N, FX2NC И FX3U	8-7
8.6	ИНТЕГРИРОВАННЫЕ ПОТЕНЦИОМЕТРЫ В FX1S И FX1N	8-8

8.7	ФУНКЦИИ РЕАЛЬНОГО ВРЕМЕНИ	8-9
8.8	ФАЙЛОВЫЕ РЕГИСТРЫ	8-10
8.9	ПЕРЕКЛЮЧЕНИЕ РЕЖИМА РАБОТА/СТОП (RUN/STOP)	8-11
8.10	БАЗОВЫЕ АППАРАТЫ FX2N С ПИТАНИЕМ 24 В ПОСТОЯННОГО ТОКА	8-11
8.11	МОДУЛЬ ОТОБРАЖЕНИЯ FX1N-5DM ДЛЯ FX1S И FX1N	8-12
8.11.1	ФУНКЦИИ	8-12
8.11.2	ОПЕРАНДЫ ДЛЯ УПРАВЛЕНИЯ МОДУЛЕМ ОТОБРАЖЕНИЯ	8-13
8.11.3	ОПРЕДЕЛЕНИЕ ОТОБРАЖАЕМЫХ ОПЕРАНДОВ	8-14
8.11.4	ЗАЩИТА ОТ НЕСАНКЦИОНИРОВАННОГО “ДОСТУПА”	8-15
8.11.5	ИЗМЕНЕНИЕ СОСТОЯНИЯ ИЛИ ЗНАЧЕНИЯ ОПЕРАНДОВ	8-15
8.11.6	АВТОМАТИЧЕСКОЕ ОТКЛЮЧЕНИЕ ПОДСВЕТКИ ОТОБРАЖЕНИЙ	8-16
8.11.7	ДЕБЛОКИРОВКА И БЛОКИРОВКА ОТОБРАЖЕНИЯ ОШИБОК	8-16

9 СПЕЦИАЛЬНЫЕ МЕРКЕРЫ, СПЕЦИАЛЬНЫЕ РЕГИСТРЫ

9.1	СПЕЦИАЛЬНЫЕ МЕРКЕРЫ (M8000...M8467)	9-1
9.1.1	СОСТОЯНИЕ ПК (M8000–M8009)	9-2
9.1.2	ВРЕМЕННЫЕ ТАКТЫ (M8011...M8019)	9-3
9.1.3	ФЛАГИ (M8020...M8029)	9-4
9.1.4	РЕЖИМЫ ПК (M8030...M8039)	9-5
9.1.5	STL-СОСТОЯНИЯ (M8040...M8049)	9-6
9.1.6	ПРОГРАММА ПРЕРЫВАНИЯ (M8050...M8059)	9-7
9.1.7	СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ СООБЩЕНИЙ О НЕИСПРАВНОСТИ (M8060-M8069)	9-8
9.1.8	ФУНКЦИЯ ПЕРЕХВАТА ИМПУЛЬСОВ	9-9
9.1.9	ФУНКЦИЯ СВЯЗИ И СПЕЦИАЛЬНЫЕ МЕРКЕРЫ (M8070–M8191)	9-10
9.1.10	СЧЕТЧИКИ СЧЕТА ВВЕРХ-ВНИЗ (M8200...M8254)	9-13
9.1.11	СПЕЦИАЛЬНЫЕ АНАЛОГОВЫЕ МОДУЛИ (M8260–M8299)	9-15
9.1.12	СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ДИАГНОСТИКИ НЕИСПРАВНОСТЕЙ (M8316-M8329)	9-16
9.1.13	СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ВЫВОДА ИМПУЛЬСОВ И ПОЗИЦИОНИРОВАНИЯ (M8330-M8467)	9-17
9.1.14	СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА (M8380-M8392)	9-19
9.1.15	СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ПРОГРАММ ПРЕРЫВАНИЯ (M8393 И M8394)	9-19
9.1.16	СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ КОЛЬЦЕВОГО СЧЕТЧИКА (M8398)	9-20
9.1.17	СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ КОММУНИКАЦИЙ (M8401-M8449)	9-20

9.2	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ (D8000–D8511)	9-21
9.2.1	СОСТОЯНИЕ ПК (D8000–D8009)	9-21
9.2.2	ТАКТЫ ВРЕМЕНИ (D8010...D8019).....	9-22
9.2.3	ФЛАГИ (D8020–D8029)	9-22
9.2.4	РЕЖИМЫ РАБОТЫ ПК (D8030 – D8039).....	9-23
9.2.5	СОСТОЯНИЕ STL (D8040–D8049)	9-23
9.2.6	РЕГИСТРЫ СВЯЗИ И СПЕЦИАЛЬНЫЕ (D8070 – D8099)	9-24
9.2.7	ПРОЧИЕ РЕГИСТРЫ (D8101 – D8114).....	9-24
9.2.8	РЕГИСТРЫ КОММУТАЦИОННЫХ АДАПТЕРОВ (D8120 – D8129).....	9-25
9.2.9	РЕГИСТРЫ ИСПОЛНЕНИЯ ДЛЯ HSZ- И PLSY-ИНСТРУКЦИИ (D8130 – D8148)	9-26
9.2.10	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ КОММУНИКАЦИЙ С ПРЕОБРАЗОВАТЕЛЯМИ ЧАСТОТЫ.....	9-27
9.2.11	РАЗЛИЧНЫЕ ФУНКЦИИ (D8158 – D8164	9-28
9.2.12	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ СЕТИ ТИПА N:N (D8173 - D8180, D820)	9-29
9.2.13	ИНДЕКС-РЕГИСТРЫ (D8182 – D8195).....	9-30
9.2.14	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ АНАЛОГОВЫХ СПЕЦИАЛЬНЫХ МОДУЛЕЙ (D8260 - D8299)	9-30
9.2.15	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ МОДУЛЯ ИНДИКАЦИИ FX3U-7DM (D8300 - D8303).....	9-31
9.2.16	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ КОМАНДЫ RND (D8310 И D8311).....	9-31
9.2.17	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ ДИАГНОСТИКИ ОШИБОК (D8312 - D8319)	9-32
9.2.18	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ ВЫВОДА ИМПУЛЬСОВ И ПОЗИЦИОНИРОВАНИЯ	9-32
9.2.19	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ ПРОГРАММ ПРЕРЫВАНИЯ (D8393)	9-34
9.2.20	КОЛЬЦЕВОЙ СЧЕТЧИК (D8398 И D8399	9-34
9.2.21	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛx КОММУНИКАЦИЙ (D8400 - D8437)....	9-35

10 ОШИБКИ ПРОГРАММЫ

10.1	ОПОЗНАНИЕ ОШИБКИ	10-1
10.1.1	СПЕЦИАЛЬНЫЕ МЕРКЕРЫ (M8060...M8069)	10-1
10.1.2	СПЕЦИАЛЬНЫЕ РЕГИСТРЫ (D8060–D8069).....	10-2
10.2	КОДЫ ОШИБОК.....	10-3
10.2.1	КОДЫ ОШИБОК (3801-3820)	10-3
10.2.2	КОДЫ ОШИБОК (6101–6409)	10-3
10.2.3	КОДЫ ОШИБОК (6501–6511)	10-5

10.2.4 КОДЫ ОШИБОК (6601–6609)	10-6
10.2.5 КОДЫ ОШИБОК (6610–6632)	10-7
10.2.6 КОДЫ ОШИБОК (6701–6710)	10-8
10.2.7 КОДЫ ОШИБОК (6730–6773)	10-9
10.2.8 КОДЫ ОШИБОК ПРИ НЕИСПРАВНОСТЯХ В СПЕЦИАЛЬНЫХ МОДУЛЯХ	10-11

A ТЕХНИЧЕСКИЕ ДАННЫЕ

A.1 ОБЗОР БАЗОВЫХ КОМАНД	A-1
A.2 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELESEC FX1S	A-4
A.3 ОПЕРАНДЫ MELESEC FX1S	A-4
A.4 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELESEC FX1N	A-6
A.5 ОПЕРАНДЫ MELESEC FX1N	A-6
A.6 ИСПОЛЬЗУЕМЫЕ ИНСТРУКЦИИ MELSEC FX1S/FX1N	A-8
A.7 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELSEC FX2N/FX2NC	A-10
A.8 ОПЕРАНДЫ MELSEC FX2N/FX2NC	A-10
A.9 ИСПОЛЬЗУЕМЫЕ ИНСТРУКЦИИ MELSEC FX2N/FX2NC	A-12
A.10 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELSEC FX3U	A-14
A.11 ОПЕРАНДЫ MELSEC FX3U	A-14
A.12 ПРИКЛАДНЫЕ КОМАНДЫ MELSEC FX3U	A-16

B ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ

B.1 ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ ДЛЯ ПК СЕРИЙ FX1S/FX1N	B-1
B.1.1 БАЗОВЫЕ КОМАНДЫ И ИНСТРУКЦИИ ШАГОВЫХ СОСТОЯНИЙ	B-1
B.1.2 ИНСТРУКЦИИ РАЗВЕТВЛЕНИЯ	B-3
B.1.3 ИНСТРУКЦИИ СРАВНЕНИЯ И ПЕРЕНОСА	B-3
B.1.4 АРИФМЕТИЧЕСКИЕ ИНСТРУКЦИИ	B-4
B.1.5 ИНСТРУКЦИИ РОТАЦИИ И СДВИГА	B-4
B.1.6 ОПЕРАЦИИ С ДАННЫМИ	B-5
B.1.7 ВЫСОКОСКОРОСТНЫЕ ИНСТРУКЦИИ	B-5
B.1.8 ИНСТРУКЦИЙ ОРИЕНТИРОВАННЫЕ НА ПОЛЬЗОВАТЕЛЯ	B-6
B.1.9 СПЕЦИАЛЬНЫЕ FNC-ИНСТРУКЦИИ	B-6
B.1.10 ИНСТРУКЦИИ ПОЗИЦИОНИРОВАНИЯ	B-7

В.1.11	ИНСТРУКЦИИ РЕАЛЬНОГО ВРЕМЕНИ	В-7
В.1.12	ОБМЕН ДАННЫМИ С ВХОДНЫМИ АНАЛОГОВЫМИ МОДУЛЯМИ FXON-3A	В-7
В.1.13	ИНСТРУКЦИИ СРАВНЕНИЯ	В-8
В.2	ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ ДЛЯ ПК СЕРИИ FX2N/FX2NC	В-9
В.2.1	БАЗОВЫЕ КОМАНДЫ И ИНСТРУКЦИИ ШАГОВЫХ СОСТОЯНИЙ	В-9
В.2.2	ИНСТРУКЦИИ РАЗВЕТВЛЕНИЯ	В-11
В.2.3	ИНСТРУКЦИИ СРАВНЕНИЯ И ПЕРЕНОСА	В-11
В.2.4	АРИФМЕТИЧЕСКИЕ ИНСТРУКЦИИ	В-12
В.2.5	ИНСТРУКЦИИ РОТАЦИИ И СДВИГА	В-13
В.2.6	ОПЕРАЦИИ С ДАННЫМИ	В-14
В.2.7	ВЫСОКОСКОРОСТНЫЕ ИНСТРУКЦИИ	В-14
В.2.8	ИНСТРУКЦИЙ ОРИЕНТИРОВАННЫЕ НА ПОЛЬЗОВАТЕЛЯ	В-15
В.2.9	СПЕЦИАЛЬНЫЕ FNC-ИНСТРУКЦИИ	В-16
В.3	ВРЕМЯ ВЫПОЛНЕНИЯ В СЕРИИ FX3U	В-19
В.3.1	БАЗОВЫЕ КОМАНДЫ И КОМАНДЫ СОСТОЯНИЯ ШАГА	В-19
В.3.2	2 КОМАНДЫ РАЗВЕТВЛЕНИЯ ПРОГРАММЫ	В-21
В.3.3	КОМАНДЫ СРАВНЕНИЯ И ПЕРЕДАЧИ	В-22
В.3.4	АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ	В-23
В.3.5	КОМАНДЫ ВРАЩЕНИЯ И СДВИГА	В-24
В.3.6	ОПЕРАЦИИ С ДАННЫМИ	В-24
В.3.7	ВЫСОКОСКОРОСТНЫЕ КОМАНДЫ	В-25
В.3.8	ПРИКЛАДНЫЕ КОМАНДЫ	В-25
В.3.9	СПЕЦИАЛЬНЫЕ КОМАНДЫ	В-26
В.3.10	ВЛИЯНИЕ НА ВРЕМЯ ВЫПОЛНЕНИЯ В СЛУЧАЕ ИМПУЛЬСНОЙ ВЕРСИИ КОМАНДЫ (FX3U)	В-32
В.4	КОДИРОВКА ASCII	В-33

1 ВВЕДЕНИЕ

1.1 ОБЩИЕ ПОЛОЖЕНИЯ

Области применения

В данном руководстве изложены основы программирования контроллеров MELSEC серий FX1S, FX1N, FX2N, FX2NC и FX3U.

Информация по установке вводу в эксплуатацию, обслуживанию и устранению ошибок прибора имеется в соответствующих руководствах по техническим средствам.

УКАЗАНИЕ

Если нет особых указаний, то все общие определения всегда относятся ко всем названным типам устройств. Применяемое во всем руководстве обозначение "FX-семейство" в основе своей относится ко всем типам устройств:

- FX1S
- FX1N
- FX2N
- FX2NC
- FX3U

Указания и особенности, которые всегда касаются только определенного типа устройства управления, соответственно поясняются.

1.2 ПРИМЕНЯЕМЫЕ ПРОГРАММАТОРЫ/ПРИБОРЫ ОБСЛУЖИВАНИЯ

В таблице приведены программаторы, которые непосредственно могут работать с инструкциями FX2-семейства.

Описание	Модель	Новые версии, полностью совместимые
Ручной программатор	FX-10P-E	FX2N с V 3.00 FX1N/FX1S с V 4.00
Ручной программатор	FX-20P-MFXA-E	FX2N с V 3.00 FX1N/FX1S с V 5.00
Устройство отображения данных	FX-10DU-E	V 4.00
	FX-20DU-E	Поддерживает только операнды FX-серии
	FX-30DU-E	V 3.00
	FX-40DU-E(S)	Поддерживает только операнды FX-серии
	FX-40DU-TK-ES	V 3.00
	FX-50DU-TK(S)-E	V 2.10

Кроме того могут применяться панели оператора серий F-GOT, A-GOT, MAC E.

Программировать контроллеры MELSEC FX особенно удобно с помощью персонального компьютера, на котором установлена среда программирования GX Developer FX, GX Developer или GX IEC Developer.

2 ОСНОВНЫЕ ПОЛОЖЕНИЯ ПРОГРАММИРОВАНИЯ

2.1 ОБРАБОТКА ПРОГРАММ В ПК

Принцип функционирования

На входы ПК поступают аналоговые или двоичные сигналы, они обрабатываются в ПК и передаются на последующие выходы.

Принцип работы

ПК работает по предварительно заданной программе. Такая программа заносится в ПК и хранится в памяти программ.

Программа состоит из последовательности отдельных управляющих инструкций, которые определяют функции управления. ПК обрабатывает инструкции управления в соответствии с запрограммированной последовательностью одну за другой - т.е. последовательно. Поэтому для изготовления управляющей программы собственно управляющий процесс должен быть расчленен на отдельные инструкции.

Общий проход программы непрерывно повторяется, это определяет, таким образом, циклический проход программы. Время, необходимое для прохода программы обозначается как время цикла программы. Проход программы, обычно называют сканированием, а один цикл программы - сканом.

Для выполнения отдельных команд контроллеру нужно время (см. приложение В). Суммарное время выполнения команд плюс время, затрачиваемое на внутрисистемные процессы, дает время цикла программы контроллера. Это - время, необходимое для прогона программы. Время цикла зависит от типа контроллера, а также от количества и типа управляющих команд.

2.1.1 МЕТОД ОТОБРАЖЕНИЯ ПРОЦЕССА

Программа пользователя обрабатывается в ПК по так называемому методу отображения процесса (здесь процесс является отображением состояния - включен/отключен - входов, выходов и различных внутренних элементов ПК):

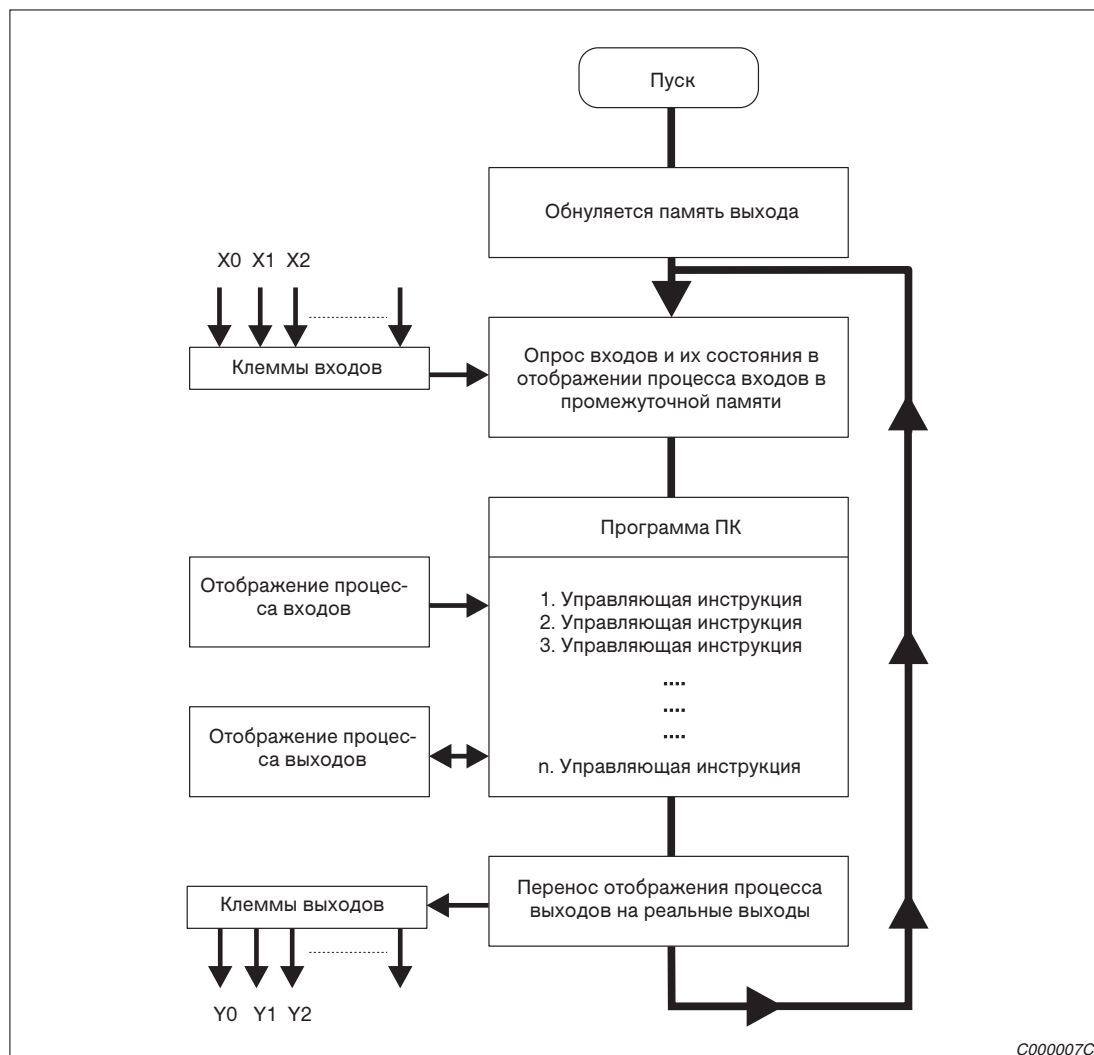


Рис. 2-1. Обработка программы по методу отображения процесса

Отображение процесса входов

Выполняется опрос реальных входов и запись их текущего состояния в память. Тем самым создается процесс отображения входов.

Обход программы

Во время подключения обхода программы микропроцессор обращается к записанному в отображении процесса входов состоянию входа. Каждая инструкция управления в соответствии с ее последовательностью связывается с операндом. Результат связи сохраняется в промежуточной памяти. Т.е. изменение сигнала на входе опознается лишь на следующем цикле программы.

Отображение процесса выходов

выходов. Лишь после обхода программы пользователя промежуточные результаты передаются к реальным выходам. В памяти промежуточных выходов всегда изменяется согласно результатам логических связей состояние выходов и хранится в процессе отображения выходов. При обходе программы никогда не выполняется непосредственный доступ к входам/выходам, а только к их процессу отображения. После окончания присвоения значений состояния реальным выходам цикл программы повторяется.

2.1.2 РАЗЛИЧИЕ ОБРАБОТКИ УПРАВЛЯЮЩИХ СИГНАЛОВ ПО СРАВНЕНИЮ С ПРОВОДНОЙ СИСТЕМОЙ УПРАВЛЕНИЯ

При проводной системе управления все задаваемые управляющие процессы выполняются одновременно (параллельно). Каждое изменение состояния входных сигналов сейчас же действует на изменение состояния выходных сигналов.

УКАЗАНИЕ

При управлении от ПК изменение состояния входных сигналов, произошедшее во время текущего прохода программы, опознается только на следующем цикле программы. Этот недостаток ПК значительно сглаживается только благодаря чрезвычайно короткому времени цикла.

Время цикла обхода программы зависит от количества управляющих инструкций в программе и от типа используемых инструкций.

2.2 УПРАВЛЯЮЩИЕ ИНСТРУКЦИИ

Программа ПК состоит из последовательности логических связей, которые определяют функции системы управления. Для создания программы необходимо разложить задачу управления на отдельные управляющие инструкции. Управляющая инструкция является самым малым элементом программы.

2.2.1 СТРУКТУРА УПРАВЛЯЮЩЕЙ ИНСТРУКЦИИ

Управляющая инструкция состоит из номера шага (строки памяти программы), инструкции (команды) и операнда.

Управляющая инструкция			
Номер шага	Инструкция	Операнд	
		Признак типа операнда	Адрес операнда
"015"	"AND"	"γ"	"003"

Табл.2-1.
Структура управляющей инструкции

- Инструкции обрабатываются в определенной последовательности, которая определяется указанием номера шага.
- Инструкция (команда) описывает выполняемую функцию, т.е. вид логической связи.
- Операнд указывает, с кем логическая связь (инструкция) должна выполняться. Операнд может быть, например, входной клеммой, выходной клеммой или внутренним счетчиком.

УКАЗАНИЕ

Для некоторых управляющих инструкций (команд) могут отсутствовать указание операнда и/или адреса операнда.

2.2.2 ОПЕРАНДЫ

Операнды состоят из

- Признака типа операнда и
- адреса операнда.

Признак типа операнда определяет вид операнда.

Указание адреса операнда позволяет

- Отличать элементы при многократном использовании одинаковых признаках типа операнда или
- определять значение числа, например, для констант.

В таблице дан обзор всех программируемых операндов и соответствующих признаков типов операндов.

Операнд	Признак типа операнда	Значение
Вход	X	Входные клеммы ПК
Выход	Y	Выходные клеммы ПК
Меркер	M	Вспомогательные реле. Память для двоичных промежуточных результатов
Таймер	T	Реле времени. Память для реализации временных параметров
Счетчик	C	Счетчик. Память для реализации счета
Состояние шага	S	Определение шагов. Программирование управляющего процесса
Десятичная константа	K	Определение десятичного значения числа
шестнадцатеричного константа	H	Определение шестнадцатеричного значения числа
Регистр данных	D	Память данных. 16-ти битный, 32-х битный формат
Регистр индексации	V, Z	Память данных для промежуточных результатов, индексации 16-ти битный формат
Указатель	P	Адрес для перехода. Маркирование перехода программы
Указатель прерывания	I	Прерывание программы. Переход к программе прерывания
Вложимость (подпрограммы)	N	Разветвление программ. Вложимость (подпрограммы)

Табл. 2-2. Операнды и соответствующие признаки типов операндов

2.2.3 ВИДЫ ИЗОБРАЖЕНИЯ УПРАВЛЯЮЩИХ ИНСТРУКЦИЙ

пользователя:

- Листинга инструкций (AWL)
- Пошаговая схема (STL)
- Схемы релейно-контактной логики (КОР)

УКАЗАНИЕ

В зависимости от используемой системы программирования не могут применяться все три системы изображения.

Программирование на видах изображения, определенных IEC 1131.3, также возможно.

ЛИСТИНГ ИНСТРУКЦИЙ

Листинг инструкций представляет программу как последовательность управляющих инструкций на листе.

Номер шага	Инструкция	Признак типа операнда	Адрес операнда
000	LD	X	000
001	ORI	X	001
002	OUT	Y	000
003	END	–	–

Табл.2-3. Пример листинга инструкций

Функциональная схема

функциональная схема представляет программу как последовательность связей, причем управляющие команды в пределах цепи представлены как функциональные блоки.

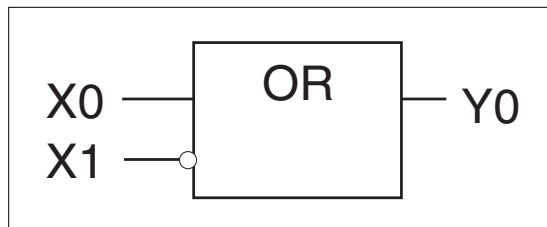


Рис.2-2.
Пример функционального блок

000381C

КОНТАКТНАЯ СХЕМА

Контактная схема является производной от релейно-контактной принципиальной электросхемы в упрощенном представлении. В отличие от обычного для принципиальной схемы вертикального расположения цепей тока в контактной схеме цепи тока изображаются горизонтально и размещаются одна под другой.

В основном применяются следующие базовые символы.


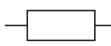
Символ	Значение
	Символ для входного сигнала (аргумента) с опросом на состояние сигнала "1"
	Символ для входного сигнала (аргумента) с опросом на состояние сигнала "0"
	Символ для выходного сигнала (функции). При управлении с сигналом "1" сигнал "1" присваивается соответствующему операнду
	Символ для специальных функций

Табл.2-4.
Символика контактной схемы

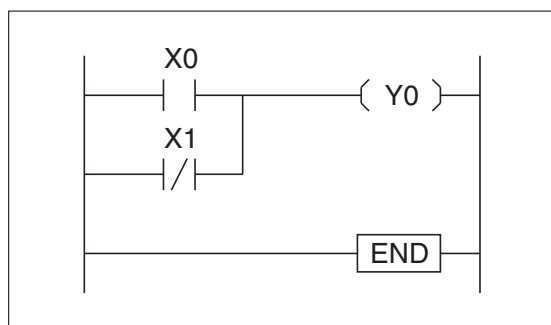


Рис.2-3.
Пример контактной схемы

C000004C

2.2.4 ЛИСТИНГ РАСПРЕДЕЛЕНИЯ И ПОДКЛЮЧЕНИЕ ПК

Наряду с описанием программы пользователя ПК на практике важны листинг распределения и подключение к ПК.

ЛИСТИНГ РАСПРЕДЕЛЕНИЯ

По листингу распределения определяются

- Какие аппараты присоединяются к входам/выходам
- Какие функции, имеющиеся в ПК (счетчики, меркеры и др.), применяются в управляемом процессе.

Назначение	Позиционное обозначение аппарата	Признак типа операнда	Адрес операнда
Замыкающий контакт "ВКЛ"	S1	X	000
Размыкающий контакт "ОТКЛ"	S2	X	001
Сигнализатор 1	H1	Y	000
Сигнализатор 2	H2	Y	001
Таймер (100 мс)	–	T	003

Табл.2-5. Пример листинга распределения

ПОДКЛЮЧЕНИЕ К ПК

Подключение к ПК представляет связи между ПК и присоединяемыми аппаратами входов и выходов.

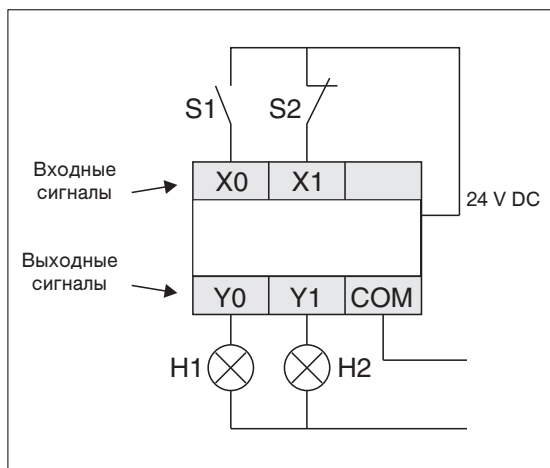


Рис.2-4.
Пример присоединения к ПК

C000005C

3 ОПЕРАНДЫ

3.1 ОБЗОР ОПЕРАНДОВ

Эта глава описывает все имеющиеся операнды и возможности их применения в программе пользователя ПК. При указании операнда определяется с какой операцией (инструкцией) производится работа.

Операнд состоит из

- Признака типа операнда и
- адреса операнда

MELSEC-Операнды		Мах.число операндов				
		FX1S	FX1N	FX2N	FX2NC	FX3U
Вход	X	Максимально 16, зависит от типа системы управления ПК	Сумма до 128	Сумма до 256		
Выход	Y	Максимально 14, зависит от типа системы управления ПК				
Меркер	M	384	384	3072		8192
Таймер	T	64	256	256		512
Счетчик	C	53	256	256		235
Высокоскоростной счетчик (используются одновременно)	C	21	21	21		8
Состояние шага	S	128	1000	1000		4096
Десятичная константа	K	16/32бита	16/32бита	16/32бита		16/32 бита
шестнадцатеричного константа	H	16/32бита	16/32бита	16/32бита		16/32 бита
Константа с плавающей запятой	E	–	–	–		32 бита
Строковая величина	" "	–	–	–		макс. 32 знака
Регистр данных	D	256	8000	8000		8000
Регистр файлов	D	–	7000 (частично)	7000 (частично)		7000 (по частям)
Регистр индексации	V, Z	16	16	16		16
Указатель	P	64	128	128		4096
Указатель прерывания	I	6	6	6 входов 3 таймера		6 входов, 3 таймера, 3 счетчика
Разветвление программы	N	8	8	8		8

Табл.3-1. Операнды и соответствующие признаки типов операндов

Подробный обзор операндов и адресов операндов для для каждого типа ПК находится в приложении этого руководства.

3.2 ВХОДЫ И ВЫХОДЫ

Входы и выходы в программе пользователя ПК представляются операндами. Посредством указания дополнительно адреса операнда можно точно обращаться при программировании к отдельным входам и выходам.

3.2.1 АДРЕСАЦИЯ ВХОДОВ И ВЫХОДОВ

Адресация входов и выходов должна выполняться в восьмеричной системы, т.е. имеет место пропуск позиций уже после 8-ми цифр (0,1,2,3,4,5,6,7,10,11,...16,17).

Тип ПК	Вид операнда	Признак типа операнда, адрес операнда	Количество адресов
FX1S	Входы	X0 ... X17	6 – 16
	Выходы	Y0 ... Y15	4 – 14
FX1N	Входы	X0 ... X157	max. 128 ^①
	Выходы	Y0 ... Y157	max. 128 ^①
FX2NFX2NC	Входы	X0 ... X317	max. 256 ^②
	Выходы	Y0 ... Y317	max. 256 ^②
FX3U	Входы	X0 ... X367	max. 256 ^②
	Выходы	Y0 ... Y367	max. 256 ^②

Табл.3-2. Количество имеющихся входов и выходов с относящимся к ним адресам операндов

- ① Сумма входов/выходов (технических средств) равна 128. Посредством программных средств можно адресовать 128 входов и 128 выходов.
- ② Сумма входов/выходов (технических средств) равна 256. Посредством программных средств можно адресовать 256 входов и 256 выходов.

Обработка входных сигналов при коротком временном импульсе

Очень короткий импульс на входе не воспринимается. Сигналы включенного или отключенного состояния входов должны сохраняться во время всего цикла программы (скана).

При времени скана примерно в 10 мс и замедлении их восприятия в 10 мс включенное и отключенное состояние входов должно быть больше 20 мс.

На этом примере входные сигналы, частота включения которых составляет более 25 Гц, не могут обрабатываться непосредственно. Обработка программы этих сигналов возможна с помощью применяемых инструкций (согласно циклам обхода программы - сканам).

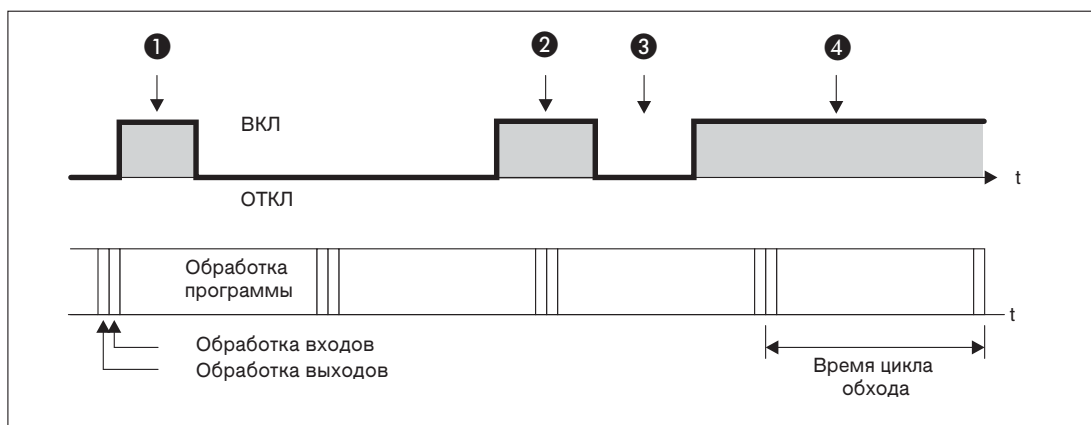


Рис.3-1. Обработка входных сигналов с коротким временем импульса

- ① и ③: это состояние входа не опознается;
- ②: это состояние входа случайно опознались;
- ④: это состояние входа всегда корректно опознается.

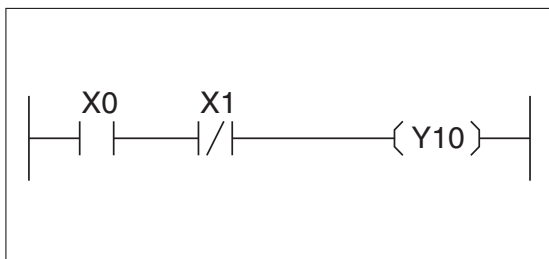
3.2.2 ПРОГРАММИРОВАНИЕ ВХОДОВ И ВЫХОДОВ

Состояние сигналов входов и выходов может опрашиваться в программе с помощью различных инструкций.

По выходам результаты логических связей могут выдаваться. Дополнительно сигналы состояния выходов при обходе программы определяются непосредственно (включаются или отключаются в отображении выходов).

ПРИМЕР ▾

Использование входов и выходов

**Рис.3-2.**

Пример программирования с использованием входов и выходов

C000122C

Выход Y10 определяет состояние сигнала "1", если выполняются следующие условия:

- Вход X0 определяет состояние сигнала "1", т.е. на клемму входа подано напряжение
- Вход X1 определяет состояние сигнала "0", т.е. на клемму входа не подано напряжение.

△

УКАЗАНИЕ

После обработки программы ПК на последнем этапе цикла ПК идет перезапись отображения процесса выходов на реальные клеммы выходов, т.е. выполняется включение или отключение выходов согласно логике обработки программы.

3.3 МЕРКЕРЫ

Для запоминания двоичных результатов логических связей (состояний сигналов "0" или "1") внутри программы применяется промежуточная память (меркер). Эти меркеры соответствуют промежуточным реле в системах управления на релейной логике.

Семейство FX наряду с "нормальными меркерами" (отключающимися при прекращении питания) представляет в распоряжение так называемые меркеры с памятью подачи и специальные меркеры.

- Меркеры с памятью сохраняют свою информацию при отключении напряжения. Информация записывается в память, сохраняющуюся при отключении напряжения.
- Специальные меркеры представляют в распоряжение специальные особые функции (см. раздел 10.1).

3.3.1 АДРЕСАЦИЯ МЕРКЕРОВ

Адресация меркеров и меркеров с памятью выполняется в десятичной системе.

Тип ПК	Операнд	Сохраняемое содержание памяти в EEPROM	Признак типа операнда, адрес операнда	Количество адресов
FX1S	Меркер	–	M0 – M383	384
	Меркер с памятью	●	M384 – M511	128
	Специальный меркер	●	M8000 – M8255	256
FX1N	Меркер	–	M0 – M383	384
	Меркер с памятью	●	M384 – M1535	1152
	Специальный меркер	●	M8000 – M8255	256
FX2N FX2NC	Меркер ^①	–	M0 – M3071	3072
	Меркер с памятью	–	M500 – M3071	2572 (частично)
	Специальный меркер	–	M8000 – M8255	256
FX3U	Маркеры	–	M0 – M499 ^②	500
	Фиксируемые маркеры	●	M500 – M1023 ^③	524
			M1024 – M7679	6656
Специальные маркеры	●	M8000 – M8511	512	

Табл.3-3. Меркеры и адресация их операндов

- ① Меркеры с адресами от M2800 до M3071 могут применяться в сочетании инструкциями формирования импульсов по переднему или заднему фронту (LDP, LDF и т.д.).
- ② В параметрах контроллера этим маркерам можно присвоить функцию фиксируемых маркеров.
- ③ В параметрах контроллера этим маркерам можно присвоить функцию небуферизованных маркеров.

3.3.2 ПРОГРАММИРОВАНИЕ МЕРКЕРОВ

Меркеры программируются как выходы. Однако отсутствует возможность присоединить к этим меркерам аппараты вне ПК, поэтому меркер представляет только место в рабочей памяти ПК.

ПРИМЕР ▾

Применения меркера

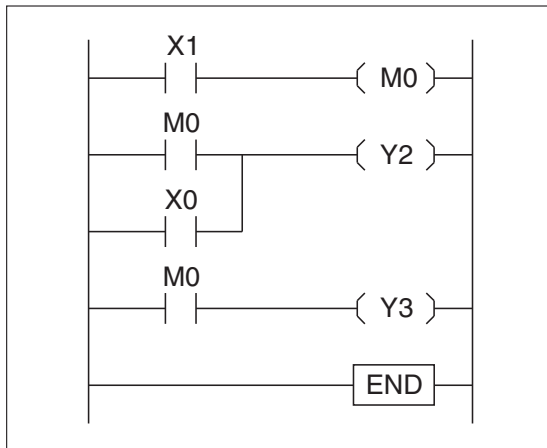


Рис.3-3.

Пример применения меркера

C000024C

Если вход X1 имеет состояние сигнала "1", то меркер M0 включается в состояние "1". Меркер M0 переключает затем выходы Y2 и Y3 в состояние "1". Если вход X0 имеет состояние сигнала "1", то выход Y2 включается в состояние "1" вне зависимости от состояния меркера M0. △

3.4 ТАЙМЕР

В релейной технике применяются реле времени с задержкой на включение или отключение. Техника ПК применяет внутренние элементы памяти, характеристики которых могут определяться программой.

Имеется различие между аналоговыми и цифровыми временными элементами (таймерами). Для включения и отключения таймер программируется как выход. Контакты таймера внутри программы могут опрашиваться как угодно часто.

3.4.1 АДРЕСАЦИЯ ТАЙМЕРА

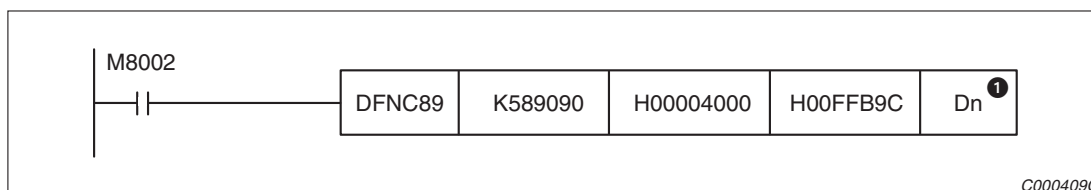
Таймер адресуется десятичными цифрами.

Тип ПК	Признак типа операнда	Количество адресов	Шаг времени	Диапазон времени	Специальный мернер M8028
FX1S	T0 – T62	63	100 ms	0,1 – 3276,7 s	ОТКЛ
	T0 – T31	32	100 ms		ВКЛ
	T32 – T62	31	10 ms	0,01 – 327,67 s	ВКЛ
	T63	1	1 ms	0,001 – 32,767 s	–
FX1N	T0 – T199	200	100 ms	0,1 – 3276,7 s	–
	T200 – T245	46	10 ms	0,01 – 327,67 s	–
	T246 – T249	4	1 ms	0,001 – 32,767 s	–
	T250 – T255 (с памятью)	6	100 ms	0,1 – 3276,7 s	–
FX2N FX2NC	T0 – T199	200	100 ms	0,1 – 3276,7 s	–
	T200 – T245	46	10 ms	0,01 – 327,67 s	–
	T246 – T249	4	1 ms	0,001 – 32,767 s	–
	T250 – 255 (с памятью)	6	100 ms	0,1 – 3276,7 s	–
FX3U	T0 – T199	200	100 ms	0,1 – 3276,7 s	–
	T200 – T245	46	10 ms	0,01 – 327,67 s	–
	T246 – T249 (с памятью)	3	1 ms	0,001 – 32,767 s	–
	T250 – T255 (с памятью)	6	100 ms	0,1 – 3276,7 s	–
	T256 – T511	256	1 ms	0,001 – 32,767 s	–
FX2N	T0 – T199	200	100 ms	0 – 3276,7 s	–
	T200 – T245	46	10 ms	0 – 327,67 s	–
	T246 – T249	4	1 ms	0 – 32,767 s	–
	T250 – T255	6	100 ms	0 – 3,2767 s	–

Табл.3-4. Временная область таймеров и относящиеся к ним адреса операндов

УКАЗАНИЕ

Если у ПК серии FX1S с номером версии от V1.00 до V1.30 в программе применяется таймер T63 (шаг времени – дискрета равен 1 мс), то программа должна быть дополнена приведенной ниже инструкцией. Инструкцию нужно вставлять только один раз, перед записью T63. Для ПК, начиная с версии V1.40 эта инструкция не нужна.



C000409C

Рис. 3-4: Добавление программы при применении таймера T63 для ПК FX1S версии от V1.00 до V1.30

① Номер регистра определяется пользователем. Согласно инструкции записываются два следующих один за другим регистра.

3.4.2 ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ

Требуемая уставка времени определяется с помощью десятичной константы К, которая указывает количество отсчитываемых шагов времени (дискрет).

ПРИМЕР ▾

Для 100 мс таймера, у которого десятичная константа определена как $K = 5$, значение уставки времени равно $5 \times 100 = 500$ мс.

△

Таймер работает с задержкой на включение. Таймер при управлении им активизируется состоянием сигнала "1". После отсчета установленного значения времени таймер принимает значение "1". Таймер возвращается в отключенное состояние, как только его вход (цепь катушки таймера) изменится состояние "1" на "0".

УКАЗАНИЕ

Для 100 мс таймера, у которого десятичная константа определена как $K = 5$, значение уставки времени равно $5 \times 100 = 500$ мс.

ПРИМЕР ▾

Следующий пример контактной схемы показывает использование таймера при косвенном задании уставки времени.

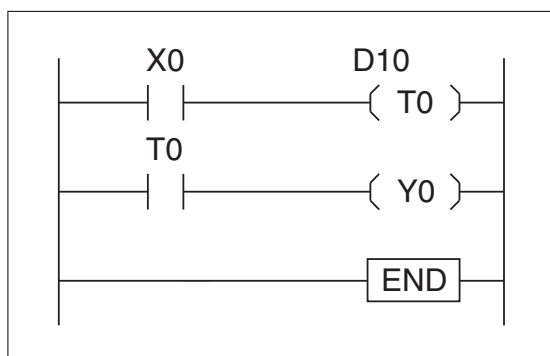


Рис.3-5.

Пример программирования с использованием таймера

C000201C

Уставка времени записана в регистр данных D10.

△

ПРИМЕР ▾

Следующий пример контактной схемы показывает использование таймера при непосредственном задании уставки времени.

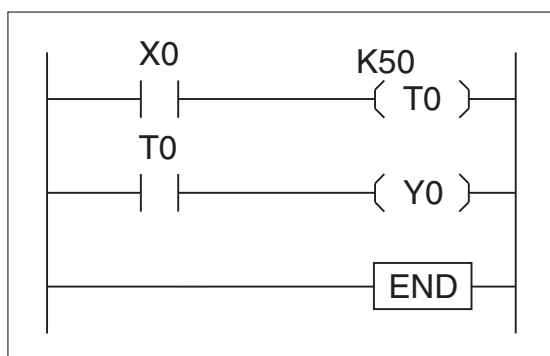


Рис. 3-6:

Указание временной константы с помощью константы

C000027C

Если вход X0 принимает состояние "1", начинается отсчет заданного времени. После окончания отсчета запрограммированного времени $t = 5$ с выход Y0 примет состояние сигнала "1". Таймер отключится, как только вход X0 примет значение сигнала "0".

△

3.4.3 ЗАДАНИЕ УСТАВКИ ВРЕМЕНИ С ПОМОЩЬЮ ИНТЕГРИРОВАННОГО ПОТЕНЦИОМЕТРА ДЛЯ ПК СЕРИИ FX1S И FX1N

С помощью двух потенциометров можно заносить значения данных в два специальных регистра, изменяя значения от 0 до 255. Значение верхнего потенциометра VR1 можно прочесть из специального регистра D8030, нижний потенциометр VR2 действует на содержание регистра данных D8031. Содержание регистров данных можно затем применить в программе как уставку времени для таймера, эти же процедуры могут быть использованы и для счетчиков.

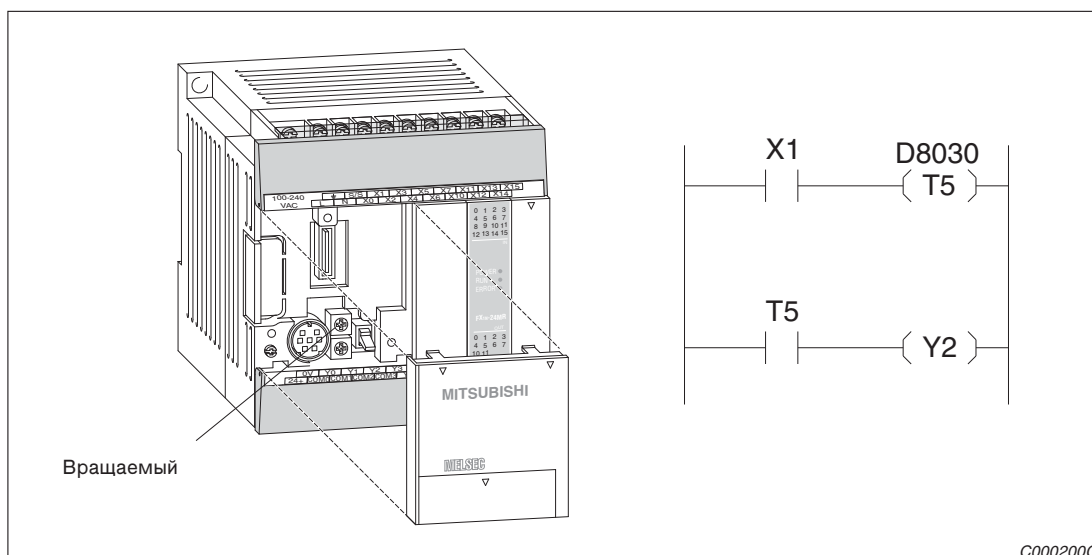


Рис.3-7. Пример настройки задания уставки времени в регистр данных D8030 с помощью потенциометра

Для ПК серии FX2N этот способ указания уставки возможен как опция.

Тип ПК	Количество потенциометров	Соответствующие регистры данных
FX1S FX1N	2	D8030 (VR1) D8031 (VR2)
FX2N FX2NC	опционально 8	Смотри используемую инструкцию VRSC

Таб.3-5. Характеристики потенциометров

3.4.4 ТОЧНОСТЬ ТАЙМЕРА

Работа таймера начинается, как только срабатывает входная цепь.

Точность отсчета таймера составляет: $(T - \alpha) \leq T \leq (T + T_0)$

T: Задаваемое значение времени

T₀: Время цикла программы

α : Шаг времени таймера (100 мс, 10 мс, 1 мс)

Если исполняемая инструкция рабочего контакта таймера находится в программе перед записью катушки таймера, то ошибочная задержка в максимуме может составлять (+2 T₀), т.к. $(T + T_0) + T_0 = T + 2T_0$.

Если уставка времени T = 0, то рабочий контакт таймера срабатывает, как только в программе пользователя начнет обрабатываться инструкция содержащая этот контакт.

3.4.5 ТАЙМЕР С ПАМЯТЬЮ

ПК серии FX1N, FX2N и FX2NC наряду с уже описанными видами таймеров имеют также таймеры с памятью (T246 до T255), которые после отключения управляющей логической связи сохраняют уже накопленное значение времени.

Действительное значение времени в таймере записывается в память, содержимое которой сохраняется и при отключении напряжения.

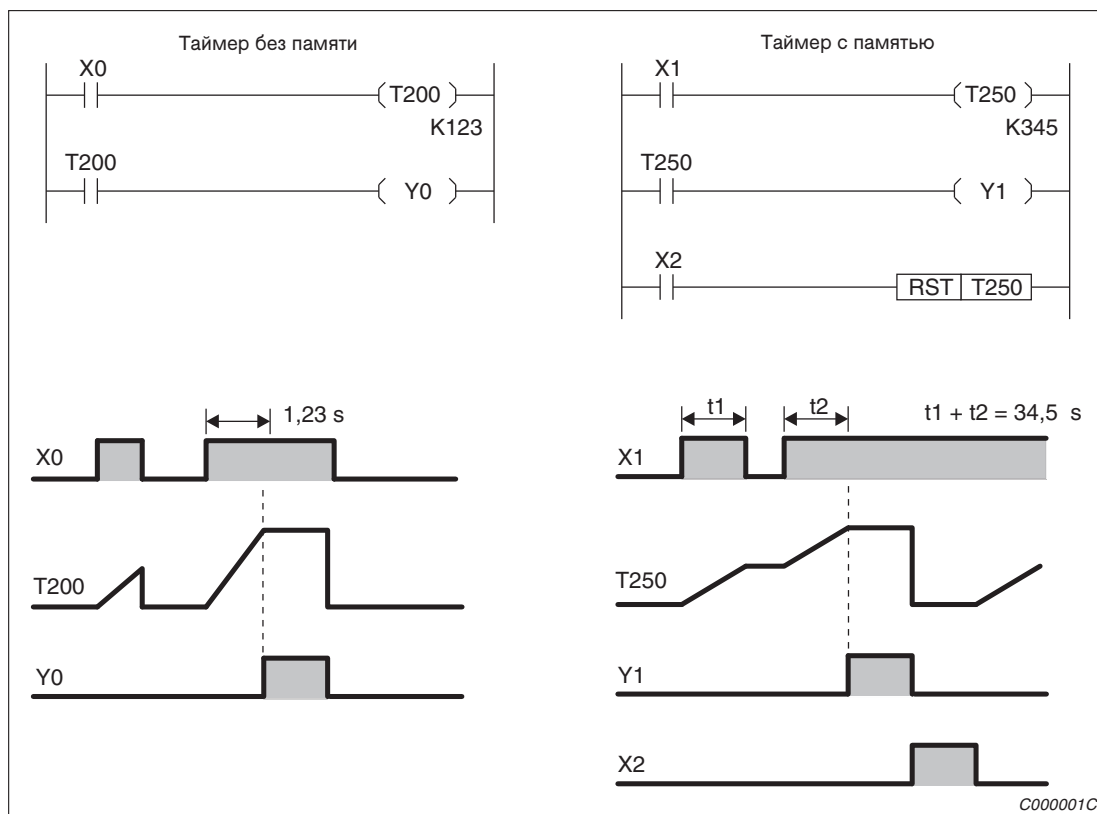


Рис.3-8. Принцип работы таймера с памятью

3.5 СЧЕТЧИКИ

Чтобы можно было программировать процессы счета, ПК семейства FX представляет в распоряжение пользователя несколько видов внутренних счетчиков.

Счетчики можно подразделить на следующие группы:

- 16-ти битный счетчик, счет вверх
Он обрабатывает внутренне программируемые сигналы счета. Подсчитанное значение счета не сохраняется в памяти ПК при отключении напряжения.
- 16-ти битный счетчик, счет вверх
Он обрабатывает внутренне программируемые сигналы счета. Подсчитанное значение счета хранится в памяти, независимой от отключения напряжения, и сохраняется в ПК при отключении напряжения.
- 32-х битный счетчик, счет вверх/вниз
Он обрабатывает внутренне программируемые сигналы счета. Направление счета задается посредством состояния специальных меркеров. Подсчитанное значение счета не сохраняется в памяти ПК при отключении напряжения.
- 32-х битный счетчик, счет вверх/вниз
Он обрабатывает внутренне программируемые сигналы счета. Направление счета задается посредством состояния специальных меркеров. Подсчитанное значение счета хранится в памяти, независимой от отключения напряжения, и сохраняется в ПК при отключении напряжения.
- 32-х битный высокоскоростной счетчик (быстрый счетчик), счет вверх/вниз
Высокоскоростной счетчик обрабатывает очень быстрые один за другим следующие внешние сигналы счета независимо от имеющегося времени цикла программы.

3.5.1 16-ТИ БИТНЫЙ СЧЕТЧИК

АДРЕСАЦИЯ 16-ТИ БИТНОГО СЧЕТЧИКА

Адресация 16-ти битного счетчика - десятичная.

Тип ПК	Адреса операндов	Количество	Сохранение при отключении напряжения
FX1S	C0 - C15	16	—
	C16 - C31	16	●
FX1N	C0 - C15	16	—
	C16 - C199	184	●
FX2N	C0 - C99	100	В параметрах контроллера можно установить, должны ли фактические значения этих счетчиков сохраняться при выключении напряжения питания.
	C100 - C199	100	
FX2NC	C0 - C99	100	
	C100 - C199	100	
FX3U	C0 - C99	100	
	C100 - C199	100	

Табл.3-6. 16-ти битный счетчик и присвоенные ему адреса операндов

ПРОГРАММИРОВАНИЕ 16-ТИ БИТНОГО СЧЕТЧИКА

Выбранное задаваемое значение счета определяется дополнительной десятичной константой. Для десятичной константы К могут применяться значения чисел в диапазоне от +1 до +32 767.

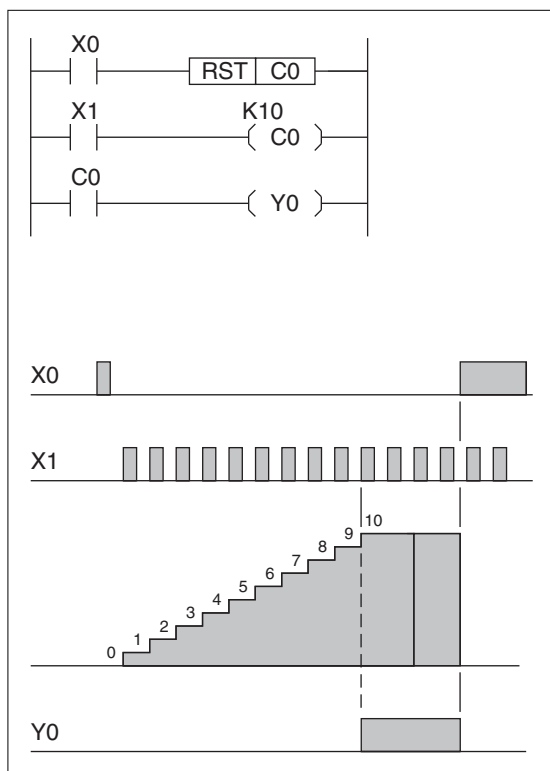
Процесс счета активизируется посредством соответствующего управляющего сигнала "1". Имеющееся значение счета повышается при этом на единицу (счет вверх). После отсчета числа импульсов равного предварительно заданному числу счетчик приобретает значение сигнала "1".

УКАЗАНИЕ

Ввод задаваемого числа счета может выполняться также косвенно через регистр данных, в который заносится десятичное число.

ПРИМЕР ▾

Применение 16-ти битного счетчика с непосредственно предварительно задаваемым числом счета

**Рис.3-9.**

Пример программирования с использованием 16-ти битного счетчика с непосредственно предварительно задаваемым числом счета

C000008C

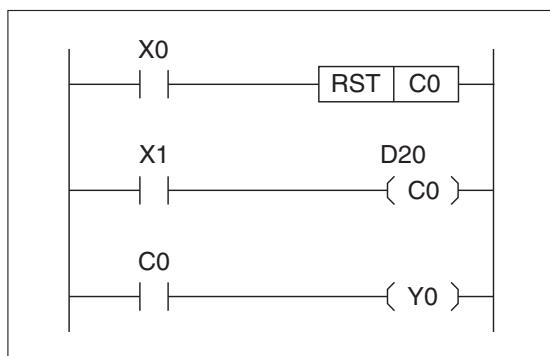
При каждом поступлении сигнала "1" на входе X1 счетчик C0 ведет счет вверх на одну единицу. Выход Y0 включается после отсчета 10 счетных сигналов на входе X1 (значение заданного числа K = 10).

После достижения заданного числа K10 счетчик останавливает счет и не реагирует на последующие импульсы поступающие со входа X1.

По входу X0 счетчик с помощью инструкции RST отключается. При этом счет сбрасывается на 0 и счетчик принимает значение сигнала "0". Выход Y0 отключается. ▴

ПРИМЕР ▾

Применение 16-ти битного счетчика с косвенно предварительно задаваемым числом счета

**Рис.3-10.**

Пример программирования с использованием 16-ти битного счетчика с косвенно предварительно задаваемым числом счета

C000028C

Задаваемое значение счета устанавливается косвенно посредством регистра данных D20. ▴

3.5.2 32-Х БИТНЫЙ СЧЕТЧИК

АДРЕСАЦИЯ 32-Х БИТНОГО СЧЕТЧИКА

Тип ПК	Адреса операндов	Количество	Сохранение при отключении напряжения
FX1N	C200 – C219	20	–
	C220 – C234	15	●
FX2N	C200 – C219	20	В параметрах контроллера можно установить, должны ли фактические значения этих счетчиков сохраняться при выключении напряжения питания.
	C220 – C234	15	
FX2NC	C200 – C219	20	
	C220 – C234	15	
FX3U	C200 – C219	20	
	C220 – C234	15	

Табл.3-7. 32-х битный счетчик и присвоенные ему адреса операндов

Счетчики C200 до C234 считают вверх и вниз; направление счета задается предварительно посредством записи предписанного специального меркера. Направление счета может изменяться во время процесса счета.

ПРЕДПИСАННЫЕ СПЕЦИАЛЬНЫЕ СЧЕТЧИКИ

Адрес операнда	C200	C201	C202	C203	C204	C205	C206
Специальный меркер	M8200	M8201	M8202	M8203	M8204	M8205	M8206
Адрес операнда	C207	C208	C209	C210	C211	C212	C213
Специальный меркер	M8207	M8208	M8209	M8210	M8211	M8212	M8213
Адрес операнда	C214	C215	C216	C217	C218	C219	C220
Специальный меркер	M8214	M8215	M8216	M8217	M8218	M8219	M8220
Адрес операнда	C221	C222	C223	C224	C225	C226	C227
Специальный меркер	M8221	M8222	M8223	M8224	M8225	M8226	M8227
Адрес операнда	C228	C229	C230	C231	C232	C233	C234
Специальный меркер	M8228	M8229	M8230	M8231	M8232	M8233	M8234

Табл.3-8. Распределение специальных меркеров

меркеров.

- Специальный меркер включен: счет вниз
- Специальный меркер выключен: счет вверх

Принцип работы соответствует 16-ти битному счетчику.

УКАЗАНИЕ

При косвенной адресации необходимы два регистра данных. Для присвоения требуемого значения счета должны применяться 32-х битные инструкции.

ПРИМЕР ▾

32-х битного счетчика с прямым заданием числа счета

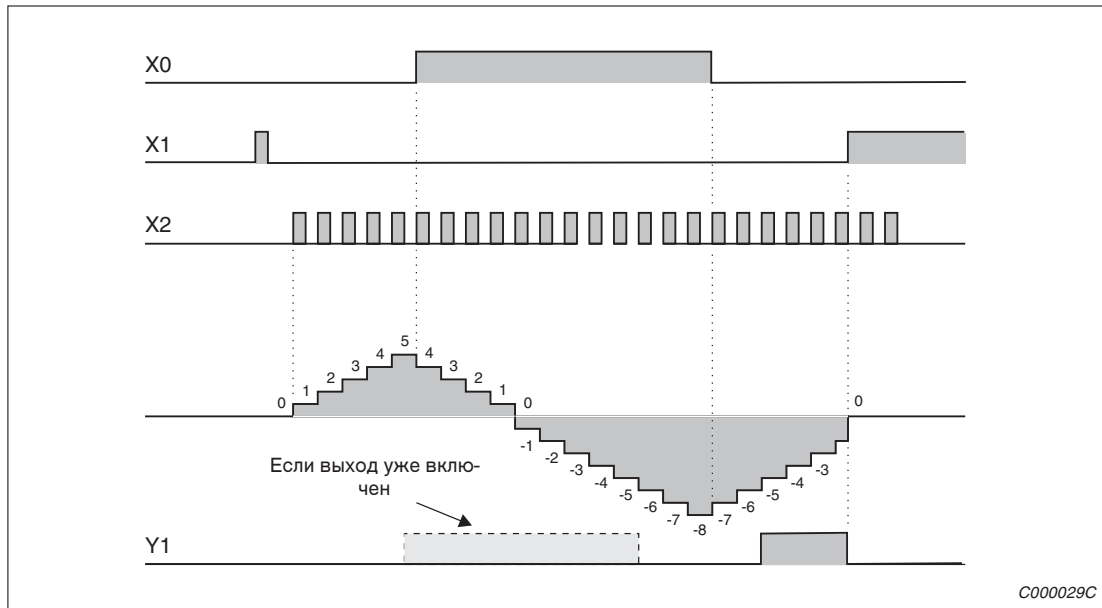


Рис.3-11. Пример программирования с использованием 32-х битного счетчика с прямым предварительно задаваемым числом счета

Как только вход X2 включится, начнется процесс счета. Счетчик C200 считает импульсы включения X2.

Выход Y1 включится, если значения в счетчике перейдет с -6 на -5. Y1 отключится, если значение в счетчике перейдет с -5 на -6.

Процесс счета (вверх и вниз) независим от текущего состояния выхода. Однако, если счетчик срабатывает выше числа +2147483647, автоматически примется значение -2147483648. Если считать ниже -2147483648, то установится значение +2147483647.

Этот счетчик называется “Кольцевым счетчиком”.

По входу X1 выполняется инструкция RST. Значение счетчика сбрасывается на 0. Выход Y1 отключается. ▴

ПРИМЕР ▾

Использование 32-х битного счетчика с косвенным заданием числа счета

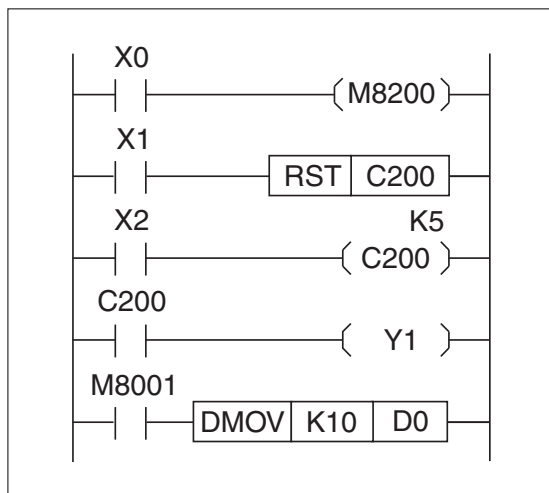


Рис.3-12.

Пример программирования с использованием 32-х битного счетчика с косвенным предварительно задаваемым числом счета

C000030C

▴

3.5.3 32-Х БИТНЫЙ ВЫСОКОСКОРОСТНОЙ СЧЕТЧИК

Высокоскоростные счетчики представляют собой 32-битовые счетчики, обрабатывающие быстрые внешние сигналы. В качестве счетных входов применяются входы с X0 по X7. Входы X6 и X7 используются только для пусковых сигналов (кроме контроллера FX3U). Они не могут быть использованы для высокоскоростного процесса счета. Так как двойное присвоение входов не разрешено, в каждой программе контроллера возможны максимум шесть высокоскоростных счетчиков (у FX3U - 8 счетчиков).

Высокоскоростным счетчикам и различным входам присвоены жестко заданные функции.

Высокоскоростной счетчик работает по принципу прерывания. Преимуществом этого является то, что сигналы счета по времени независимы от времени обработки программы (скана).

РАСПРЕДЕЛЕНИЕ ВХОДОВ

X	1-фазный счетчик без пуска и сброса						1-фазный счетчик без пуска и сброса					2-фазный счетчик, двунаправленный				А/В-фазный счетчик									
	C 235	C 236	C 237	C 238	C 239	C 240	C 241	C 242	C 243	C244	C245	C 246	C 247	C248	C 249	C 250	C 251	C 252	C253	C 254	C 255				
X0	U/D						U/D			U/D				U	U			U		A	A			A	
X1		U/D					R			R				D	D			D		B	B			B	
X2			U/D					U/D			U/D				R			R			R			R	
X3				U/D				R			R			U	U*		U			A	A*			A	
X4					U/D				U/D					D	D*		D			B	B*			B	
X5						U/D			R					R			R			R				R	
X6										S	U/D*						S							S	
X7											S	U/D*					S							S	

Табл.3-12. Высокоскоростной счетчик и его входы счета

* У контроллера FX3U входы счетчиков C244, C245, C248 и C253 можно переключать с помощью специальных маркеров M8090...M8092. (M8092 влияет на C248 и C253.)

U: Вход счета вверх

D: Вход счета вниз

A: Вход А-фазы

B: Вход В-фазы

R: Вход сброса

S: Вход пуска

Входы X6 и X7 работают только как пусковые сигналы. Они не могут применяться для высокоскоростного счета.

В программе ПК могут одновременно использоваться несколько высокоскоростных счетчиков.

При применении разных высокоскоростных счетчиков нужно обращать внимание на то, не был применен ни один счетчик, вход которого уже использован для другого счетчика. Двойное присвоение входов не разрешено.

МАКСИМАЛЬНАЯ ЧАСТОТА СЧЕТА И СКОРОСТЬ СЧЕТА

Максимальная частота или скорость поступления сигналов счета, которые еще могут обрабатываться, составляет при использовании только одного высокоскоростного счетчика в одной программе ПК:

- 1- и 2-фазный счетчик: макс. 10 кГц (у FX3U - 100 кГц)
- Сумма всех частот счета в контроллерах серий FX1S или FX1N не должна превышать 60 кГц, в контроллерах серий FX2N или FX2NC - 30 кГц, а в контроллерах серии FX3U - 80 кГц. При этом следует иметь в виду, что частота счетчиков фаз "А", "В" в этом расчете учитывается вдвойне.

Благодаря особой конструкции входов X0 и X1 в контроллерах FX2N и FX3U, эти входы способны считать очень высокие частоты. 1-фазный счетчик C235, C236 или C246: 60 кГц (FX2N), 100 кГц (FX3U)

2-фазный счетчик C251: 30 кГц (FX2N), 50 кГц (FX3U)

Если в программе применяются операции пуска или останова высокоскоростного счетчика (инструкции DHSCS, FNC 53 и DHSCR, FNC 54) или сравнения областей (DHSZ, FNC 55), то должны использоваться входы X0 и X1. В этом случае используется следующая таблица:

Тип ПК	Инструкция	Сумма всех частот счета
FX1S FX1N	FNC53 или FNC54	30 кГц
FX2N FX2NC	FNC53 или FNC54	11 кГц
	FNC55	5,5 кГц

Табл.3-13.

Ограничения при применении инструкций FNC53, FNC54, FNC55

Накопленное значение всех высокоскоростных счетчиков записано в независимую от напряжения память.

УКАЗАНИЕ

Входы счета X0...X5 не могут программироваться в качестве условий включения для высокоскоростных счетчиков.

Высокоскоростные счетчики не могут применяться совместно с 16-ти битными инструкциями

Должна соблюдаться максимальная частота счета на входах контроллера. (Счетчики фаз "А", "В" следует засчитывать вдвойне).

Инструкция SPD (FNC 56) имеет характеристику счетчика и прерывателя высокоскоростного счетчика. Поэтому в SPD-инструкции должны применяться входы X0...X5. Также и для этих входов действительно правило, что они не могут применяться одновременно другими высокоскоростными счетчиками.

1-ФАЗНЫЙ СЧЕТЧИК СО СЧЕТНЫМ ВХОДОМ

1-фазные счетчики являются высокоскоростными счетчиками только со счетным входом.

1-фазные счетчики можно подразделить на три группы:

- без пускового и сбросового входа (C235 ... C240)
- с входом сброса (C241 ... C243)
- с входами пуска и сброса (C244 ... C245)

Направление счета (счет вверх или вниз) определяется подключением специального меркера.

Специальный меркер включен: счет вверх

Специальный меркер выключен: счет вниз

1-фазные счетчики	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
меркеры	M8235	M8236	M8237	M8238	M8239	M8240	M8241	M8242	M8243	M8244	M8245

Табл.3-11. 1-фазные счетчики и присвоенные им специальные меркеры

Область счета равна от -2 147 483 648 до 2 147 483 647.

ПРИМЕР ▾

Применение 1-фазного счетчика с входами пуска и сброса (C245)

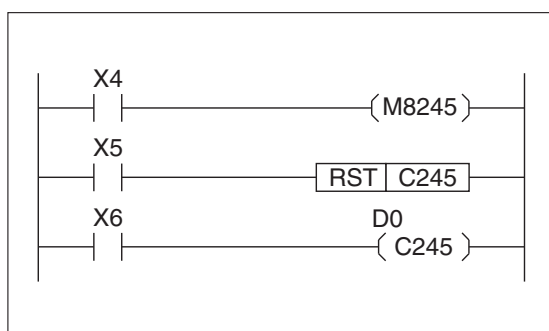


Рис. 3-14:

Пример программирования с использованием 1-фазного счетчика с входами пуска и сброса (C245)

C000123C

Если включен специальный меркер M8245, то счетчик C245 считает вверх. Если специальный меркер M8245 не включен, то счетчик C245 считает вниз. С помощью выключателя X5 накопленное значение счетчика сбрасывается на 0. Это же можно получить также при автоматическом присвоении входа сброса X3 (см.табл.3-12).

При включении X6 и автоматическому присвоению входа пуска X7 счетчик C245 активизируется и считает импульсы по входу счета X2. Так как здесь работают с 32-х битным счетчиком, то используются регистры данных D0 и D1 для записи задаваемого значения.

ПРИМЕР ▾

По сравнению с входами X5 и X6, применяемыми в программе, использование входов X7 и X3 дает то преимущество, что обработка внешних пусковой и отключающей команд являются независимыми от времени цикла программы (от скана).

△

2-Х ФАЗНЫЙ СЧЕТЧИК С ДВУМЯ СЧЕТНЫМИ ВХОДАМИ

2-х фазный счетчик имеет для каждого счетного входа возможность счета вверх и вниз.

2-фазные счетчики можно подразделить на три группы:

- без пускового и сбросового входа (C246)
- с входом сброса (C247, C248)
- с входами пуска и сброса (C249, C250)

ПК автоматически включает специальный меркер, который определяет текущее направление счета 2-х фазного счетчика:

Специальный меркер включен: счет вверх

Специальный меркер выключен: счет вниз

2-х фазные счетчики	C246	C247	C248	C249	C250
Специальные меркеры	M8246	M8247	M8248	M8249	M8250

Табл.3-12.

2-х фазные счетчики и присвоенные им специальные меркеры

Область счета равна от -2 147 483 648 до 2 147 483 647.

ПРИМЕР ▾

| Применение 2-х фазного счетчика без входов пуска и сброса (C246)

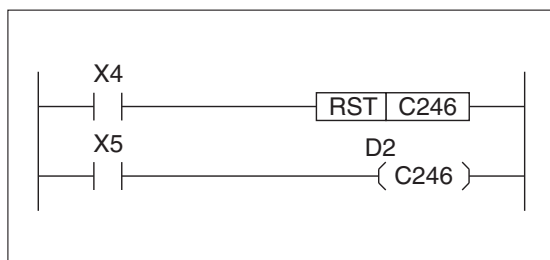


Рис.3-15.

Пример программирования с использованием 2-х фазного счетчика без входов пуска и сброса (C246)

C000124C

Если включен вход X4, счетчик C246 сбрасывается. Процесс счета запускается, если включен вход X5. Для счетчика C246 зарезервированы входы счета X0 и X1 для счета вверх и вниз.

При импульсных сигналах на входе X0 счетчик считает вверх, а при импульсных сигналах на входе X1 счетчик считает вниз. △

АВ-ФАЗНЫЕ СЧЕТЧИКИ С ДВУМЯ СЧЕТНЫМИ ВХОДАМИ

АВ-фазные счетчики имеют для каждого счетного входа фазы А и В. По сигналу на входах фаз А и В определяется должны ли счетчики считать вверх или вниз.

- Счет вверх
А-фаза-вход: "1"-сигнал
В-фаза-вход: возрастающий фронт сигнала (смена сигнала с "0" на "1")
- Счет вниз
А-фаза-вход: "1"-сигнал
В-фаза-вход: падающий фронт сигнала (смена сигнала с "1" на "0")

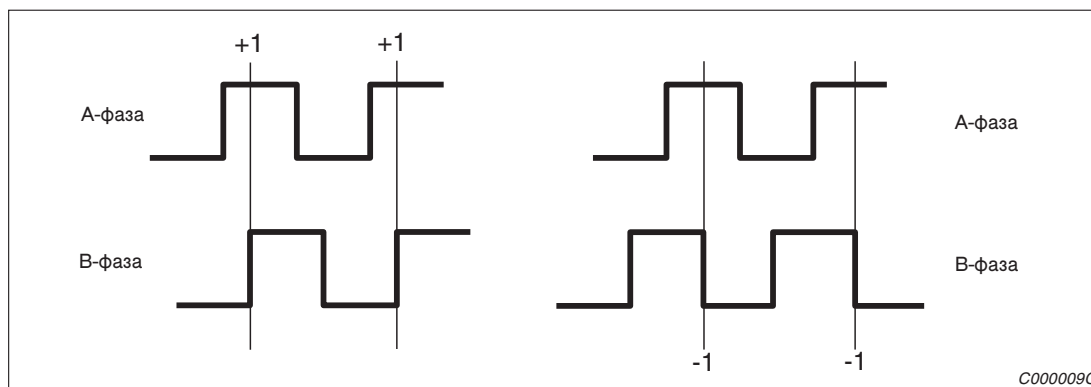


Рис.3-16. АВ-фазный счетчик

считать по единице вверх и каждая смена сигнала с падающим фронтом на единицу вниз. Во время смены сигнала на входе фазы А должен иметься сигнал "1".

АВ-фазные счетчики можно подразделить на три группы:

- без пускового и сбросового входа (C251)
- с входом сброса (C252, C253)
- с входами пуска и сброса (C254, C255)

ПК автоматически включает специальный меркер, который определяет текущее направление счета АВ-фазного счетчика:

Специальный меркер включен: счет вниз

Специальный меркер выключен: счет вверх

АВ-фазные счетчики	C251	C252	C253	C254	C255
Специальные меркеры	M8251	M8252	M8253	M8254	M8255

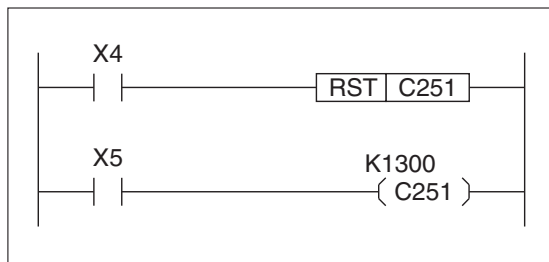
Табл.3-13.

АВ-фазные счетчики и присвоенные им специальные меркеры

Область счета равна от -2 147 483 648 до 2 147 483 647.

ПРИМЕР ▾

Применение АВ-фазного счетчика без входов пуска и сброса (C251)

**Рис.3-17.**

Пример программирования с использованием АВ-фазного счетчика без входов пуска и сброса (C251)

C000126C

Если включен вход X5, счетчик C251 считает сигналы на счетных входах X0 (А-фазный вход) и X1 (В-фазный вход). △

3.6 СОСТОЯНИЯ ШАГОВ

Операнды состояния шагов применяются для управления процессом работы (STL-инструкции). С помощью операндов состояния шагов определяются отдельные шаги управления процессом работы.

3.6.1 АДРЕСАЦИЯ ОПЕРАНДОВ СОСТОЯНИЯ ШАГОВ

Имеется до 1000 операндов состояния шагов в области от S0 до S999.

Операнды состояния шагов S можно подразделить на две группы:

- От S0 до S9 (10 адресов): инициализация состояния шагов
- От S10 до S999: свободно выбираемая область применения

Steuerung	Operanden	Davon gepuffert
FX1S	S0 - S127 (128)	S0 - S127 (128)
FX1N	S0 - S999 (1000)	S0 - S999 (1000)
FX2N	S0 - S999 (1000)	S500 - S999 (500)
FX2NC		
FX3U	S0 - S4095 (4096)	S500 - S4095 (3596)

Табл.3-14.
Обзор операндов состояний шагов

Более детальная информация для использования STL-инструкций и операндов состояния шагов S содержится в разделе 5.1.

Если в программе не применяется шаговое управление, то операнды состояния шагов S могут применяться как меркеры.

ПРИМЕР ▾

Применение операндов состояния шагов

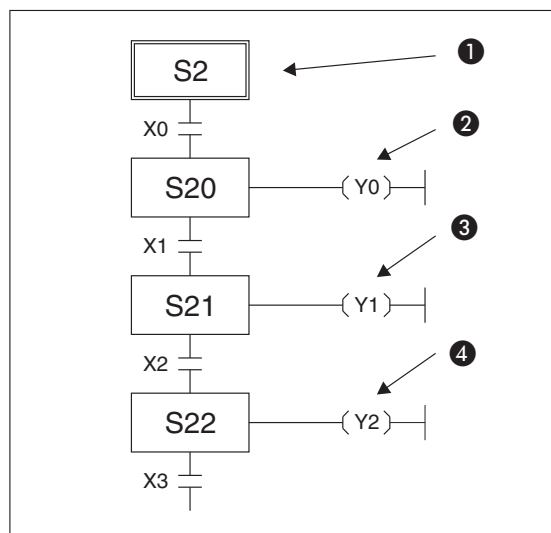


Рис 3-18.

Пример программирования для применения операндов состояния шагов

- ① Инициализируемое состояние
- ② Опускание
- ③ Захват
- ④ Подъем

C000127C

Операнд состояния шага S20 активизируется, если включается вход X0. После чего включается выход Y0 (② опускание).

После достижения нижнего состояния - срабатывает путевой выключатель X1 - и активизируется операнд состояния шага S21, после чего включается выход Y1 (③ захват).

Если включится путевой выключатель X2 (контроль окончания захвата), активизируется шаг S22. Включится выход Y2 (④ подъем). ▴

3.7 КОНСТАНТЫ

3.7.1 ДЕСЯТИЧНЫЕ И ШЕСТНАДЦАТЕРИЧНЫЕ КОНСТАНТЫ

С помощью десятичных и шестнадцатеричных констант (K, H) можно задавать числовые значения внутри программы ПК (например, задаваемые значения уставок времени или счета). Числовые значения кодируются внутри ПК в двоичном счислении.

В разделе 3.8.7 подробно описаны все важнейшие системы счисления и их перекодирование одну из другой.

Константы	16бита	32бита
Десятичные K	-32 768 – +32 767	-2 147 483 648 – +2 147 483 647
Шестнадцатеричные H	0 – FFFF	0 – FFFFFFFF

Табл.3-15. Области числовых значений десятичных и шестнадцатеричных констант

Обозначения констант в программе начинаются с букв „K“ или „H“. Примеры: K100 (десятичное число 100), H64 (шестнадцатеричное число 64)

3.7.2 КОНСТАНТЫ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Десятичные константы - это все числа без дробной части. В отличие от них, числа с плавающей запятой могут иметь разряды перед и после десятичной запятой, что дает преимущества при арифметических операциях.

Константы, представляющие собой числа с плавающей запятой, в программе обозначаются буквой "E" (например, E1.234 или E1.234 + 3). Числа с плавающей запятой можно задать двумя различными способами:

– Указание константы без показателя степени

Значение указывается как обычно. Однако запятую требуется заменить точкой. Например, значение "10,2345" в программе можно передать в виде "E10.2345".

– Указание константы с показателем степени

Значение указывается с основанием и показателем степени. Показатель степени имеет основание 10 (10^n). Например, значение "1234" можно также отобразить в виде "1,234 x 1000" или - в степенном представлении - в виде "1,234 x 10³". В программе контроллера FX3U это число указывается в виде "E1.234+3" (где "+3" соответствует "10³").

Числа с плавающей запятой можно указывать в диапазоне

$-1,0 \times 2^{128}$ до $-1,0 \times 2^{-126}$,

0

и от $1,0 \times 2^{-126}$ до $-1,0 \times 2^{+128}$.

3.7.3 СТРОКОВЫЕ КОНСТАНТЫ

Если знаки в программе указываются в кавычках, они интерпретируются в качестве знаков ASCII (например, "MOTOR12"). Один знак занимает 1 байт. Строковая величина может содержать до 32 знаков.

3.8 РЕГИСТРЫ

Регистры представляют память данных внутри ПК. В регистре можно собирать числовые значения и следующую друг за другом двоичную информацию. Для этого возможно, например, состояние сигналов нескольких входов запомнить вместе и в программе обработать.

Данные сохраняются в 16-ти битном регистре. Благодаря совместного включения двух 16-ти битных регистров можно образовать 32-х битный “Двойной регистр”.

3.8.1 КЛАССИФИКАЦИЯ РЕГИСТРОВ

Имеются следующие типы регистров:

- **РЕГИСТР ДАННЫХ (не буферизован)**
Регистр без сохранения данных при отключении напряжения ПК
- **РЕГИСТР ДАННЫХ (буферизован)**
Регистр с сохранением данных при отключении напряжения ПК. Данные хранятся в энерго независимой памяти.
- **ИНДЕКСНЫЙ РЕГИСТР**
Этот регистр служит для запоминания промежуточных результатов и для индицирования операндов. Более подробные данные см.в разделе 5.8.5
- **СПЕЦИАЛЬНЫЙ РЕГИСТР**
Для определенных контрольных и проверочных функций предусмотрен ряд специальных регистров. Подробнее см.раздел 3.8.4.
- **РЕГИСТР ФАЙЛОВ**
Для сохранения, например, параметров или рецептов, требуются регистры файлов. В контроллерах серий FX1N, FX2N, FX2NC и FX3U области памяти для этих регистров устанавливаются пользователем. Эти регистры файлов являются частью фиксируемых регистров файлов.

3.8.2 СТРУКТУРА РЕГИСТРА

Каждый регистр состоит из бита знака числа и нескольких битов данных.

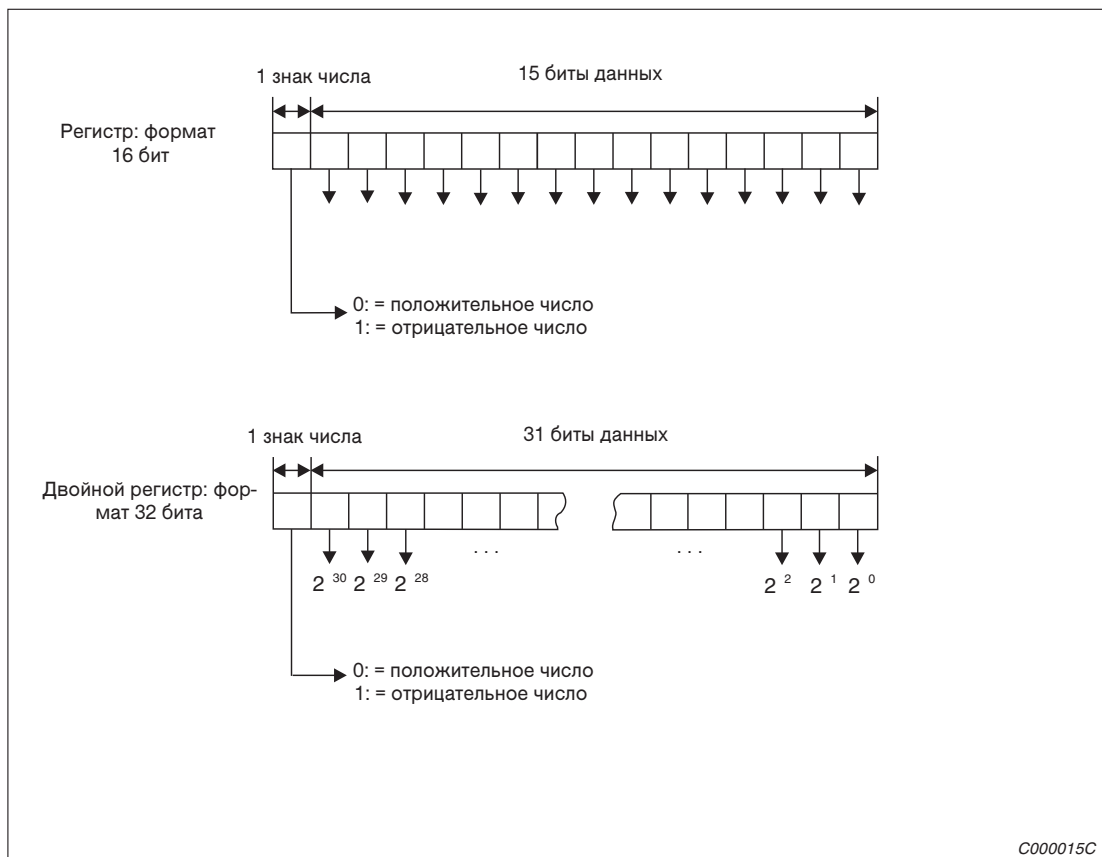


Рис.3-19. Структура регистра (16 бит) и двойного регистра (32 бита)

3.8.3 АДРЕСАЦИЯ РЕГИСТРА

Адресация регистров данных выполняется десятичной. Для двойного регистра адресация начинается с младшего 16-ти битового регистра.

Тип ПК	Регистр	Адреса	Количество	Из них буферизированно (с памятью)	Количество
FX1S	Регистр данных	D0 – D255	256	D128 – D255	128
	Индексный регистр	V, Z	16	–	–
	Специальный регистр	D8000 – D8255	256	D8000 – D8255	256
	Внешний изменяемый регистр ①	D8030 und D8031, Значения от 0 до 255 настраиваются посредством интегрированного потенциометра	2	–	–
	Регистр файлов	D1000 – D2499	1500	D1000 – D2499	1500
FX1N	Регистр данных	D0 – D7999	8000	D128 – D7999	7872
	Индексный регистр	V, Z	16	–	–
	Специальный регистр	D8000 – D8255	256	D8000 – D8255	256
	Внешний изменяемый регистр ①	D8030 und D8031, Значения от 0 до 255 настраиваются посредством интегрированного потенциометра	2	–	–
	Регистр файлов	D1000 – D7999	7000 (частично)	D1000 – D7999	7000 (частично)
FX2N FX2NC	Регистр данных	D0 – D7999	8000	D200 – D7999*	7800
	Индексный регистр	V0 – V7, Z0 – Z7	16	–	–
	Специальный регистр	D8000 – D8255	256	D8000 – D8255	256
	Регистр файлов	D1000 – D7999	7000 (частично)	D1000 – D7999	7000 (частично)
FX3U	Регистры данных	D0 – D7999	7000 (по частям)	D200 – D7999 ②	7800
	Индексные регистры	V0 – V7, Z0 – Z7	16	–	–
	Специальные регистры	D8000 – D8511	512	D8000 – D8511	512
	Регистры файлов	D1000 – D7999	7000 (по частям)	D1000 – D7999	7000 (по частям)

Табл.3-16. Регистры и присваиваемые им адреса операндов

- ① Оба этих регистра находятся также в специальных регистрах.
- ② Из указанной области регистров данных в контроллерах серий FX2N, FX2NC и FX3U пользователь только для регистров D200...D511 может устанавливать, должны ли они быть буферизованными или нет. Начиная с D512, регистры данных в этих контроллерах всегда буферизованы. Регистрам с D0 по D199 в параметрах контроллера также можно присвоить функцию буферизованных регистров.

3.8.4 ИСПОЛЬЗОВАНИЕ СПЕЦИАЛЬНЫХ РЕГИСТРОВ

Специальные регистры D8000 до D8255 внутренне жестко приписаны контрольным и проверочным функциям (функциям монитора).

При включении управления программным обеспечением автоматически записываются стандартные значения в специальные регистры. Так например, значение данных контрольного времени (Watch Dog Timers) автоматически записывается в специальный регистр D8000. Если необходимо это значение данных изменить, то нужно переписать старые данные с помощью MOV-инструкции (см.указание в р.6.3.3).

УКАЗАНИЕ

Данные не теряются при переключении управления в режим СТОП. Однако при отключении напряжения данные теряются.

Должен применяться только разрешенный для записи регистр данных. В главе 9 приведены все специальные регистры и их функции.

3.8.5 ВНЕШНЕ ИЗМЕНЯЕМЫЕ РЕГИСТРЫ

В ПК серии FX1S и FX1N интегрированы два потенциометра, с помощью которых можно изменять содержимое двух специальных регистров от 0 до 255. Для ПК серии FX2N имеется адаптер задания требуемого значения FX2N-8AV-BD, у которого возможно задание 8 аналоговых требуемых значений. Чтение требуемых значений, а также распределение областей потенциометров выполняется посредством используемых инструкций VRRD (FNC85) и VRSC (FNC86).

С помощью потенциометра могут, например, изменяться задаваемые значения для таймеров и счетчиков без подключения программатора.

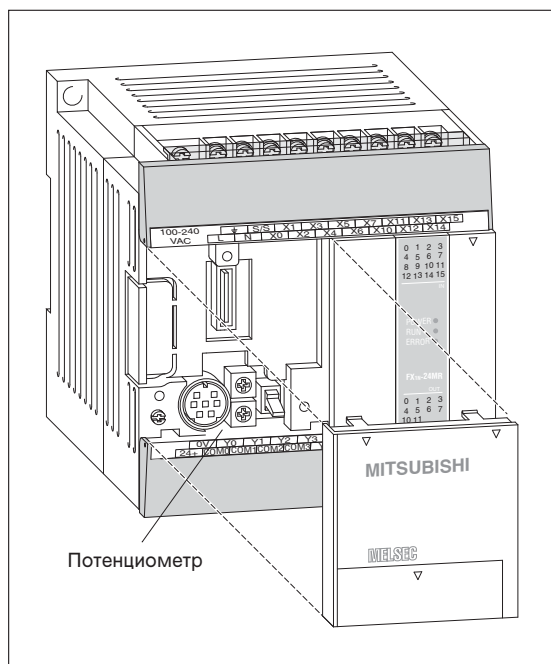


Рис.3-20.
Потенциометры на ПК серии FX1S/FX1N

C000413C

	FX1S	FX1N	FX2N
Количество потенциометров	Два интегрированных потенциометра (VR1 и VR2), при применении адаптера задания требуемого значения FX2N-8AV-BD можно использовать 8 дополнительных потенциометров.		При применении адаптера задания требуемого значения FX2N-8AV-BD можно использовать 8 потенциометров.
Внешне изменяемые регистры	D8030 действует с VR1, D8031 действует с VR2. Дополнительные 8 регистров при применении адаптера задания требуемого значения FX2N-8AV-BD определяет пользователь.		8 регистров определяются пользователем с инструкциями VRRD и VRSC.

Табл.3-17. Внешне изменяемые регистры

3.8.6 ПРИМЕНЕНИЕ ИНДЕКСНЫХ РЕГИСТРОВ

Индексные регистры применяются для того, чтобы для инструкций передачи и сравнения к адресам операндов добавить значение индекса.

Индексный регистр является 16-ти битовым регистром.

В 32-х битовых инструкциях индексные регистры V (V0...V7) и Z (Z0...Z7) применяются комбинированно. Z содержит 16 младших бит, V запоминает 16 старших бит. В качестве адреса назначения указывается индексный регистр Z. Индексный регистр не может самостоятельно индцироваться.

ПРИМЕР ▾

Передача данных от регистра данных D5V к регистру данных D10Z

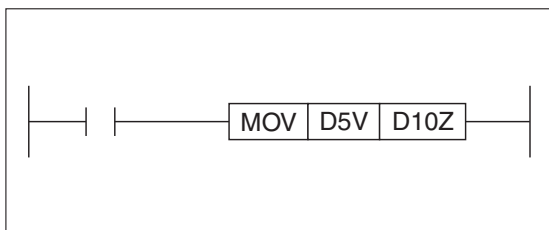


Рис.3-21.
Пример программирования передачи данных от регистра данных D5V к регистру данных D10Z

C000044C

Расчет адреса выхода D5V:

$$V = 8$$

$$5 + 8 = 13 \rightarrow D13$$

Расчет адреса пересылки D10Z:

$$Z = 14$$

$$10 + 14 = 24 \rightarrow D24$$

И следовательно, имеет место передача данных от регистра данных D13 к регистру данных D24.

△

3.8.7 ПРИМЕНЕНИЕ ФАЙЛОВЫХ РЕГИСТРОВ

Файловые регистры записываются блоками по 500 адресов в программной области памяти (EPROM или EEPROM) ПК. Количество блоков устанавливается в параметрах. Доступ к файловым регистрам возможен через программаторы и терминалы обслуживания.

УКАЗАНИЕ

Если используются файловые регистры, то для программы ПК сокращается полезная область памяти. Для каждого блока по 500 файловых регистров сокращается количество полезных программных шагов примерно на 500 шагов. Количество файловых регистров варьируется в зависимости от типа ПК.

При использовании файловых регистров в контроллерах серий FX1N, FX2N, FX2NC и FX3U необходимо учитывать, что область памяти для этих регистров перекрывается с буферизованной областью памяти.

ЧТЕНИЕ ФАЙЛОВЫХ РЕГИСТРОВ

Во время работы ПК могут читаться данные файловых регистров посредством MOV-инструкций.

ЗАПИСЬ ФАЙЛОВЫХ РЕГИСТРОВ

В контроллерах серии FX1S запись в файловых регистр возможна только с помощью программаторов или персональных компьютеров с соответствующим программным обеспечением. В контроллерах серий FX1N, FX2N, FX2NC и FX3U содержимое файловых регистров может быть изменено и из программы контроллера с помощью команды MOV.

Для более подробной информации используйте, пожалуйста, руководства соответствующей системы программирования.

У контроллера FX1S нельзя изменять данные в режиме работы (RUN-РЕЖИМЕ).

УКАЗАНИЕ

Изменение данных файлового регистра возможно в RUN-РЕЖИМЕ только для RAM-регистров или для файловых регистров во внутренней памяти.

Файловые регистры, которые находятся в RAM, во внутренней памяти или на EEPROM кассете памяти, могут изменяться в STOP-РЕЖИМЕ.

Файловые регистры, которые находятся на EPROM кассете памяти, не могут изменяться.

ОБЛАСТИ ЗНАЧЕНИЙ ЧИСЕЛ РЕГИСТРОВ ДАННЫХ

Если в регистре запоминаются числа в двоичном коде, то область значений чисел ограничивается на базе граничных величин регистров.

- Десятичные числа
16 бит: -32 768 ... +32 767 32 бит: -2 147 483 648 ... +2 147 483 647
- Шестнадцатеричные числа
16 бит: 0 ... FFFF 32 бит: 0 ... FFFFFFFF

ПРЕДСТАВЛЕНИЕ ОТРИЦАТЕЛЬНЫХ ЧИСЕЛ

Отрицательные числа представляются как дополнительный код.

При образовании 2-го дополнения инвертируется двоичное число (1-ое дополнение) и прибавляется двоичное значение числа 1.

ПРИМЕР ▾

0101101 (двоичное) → +45 (десятичное)
 1010010 (двоичное) → инверсия
 1010011 (двоичное) → дополнение
 1010011 (двоичное) → -45 (десятичное)

△

Регистр данных заносит значение как отрицательное, если в высшем разряде бита (знак числа) стоит число 1.

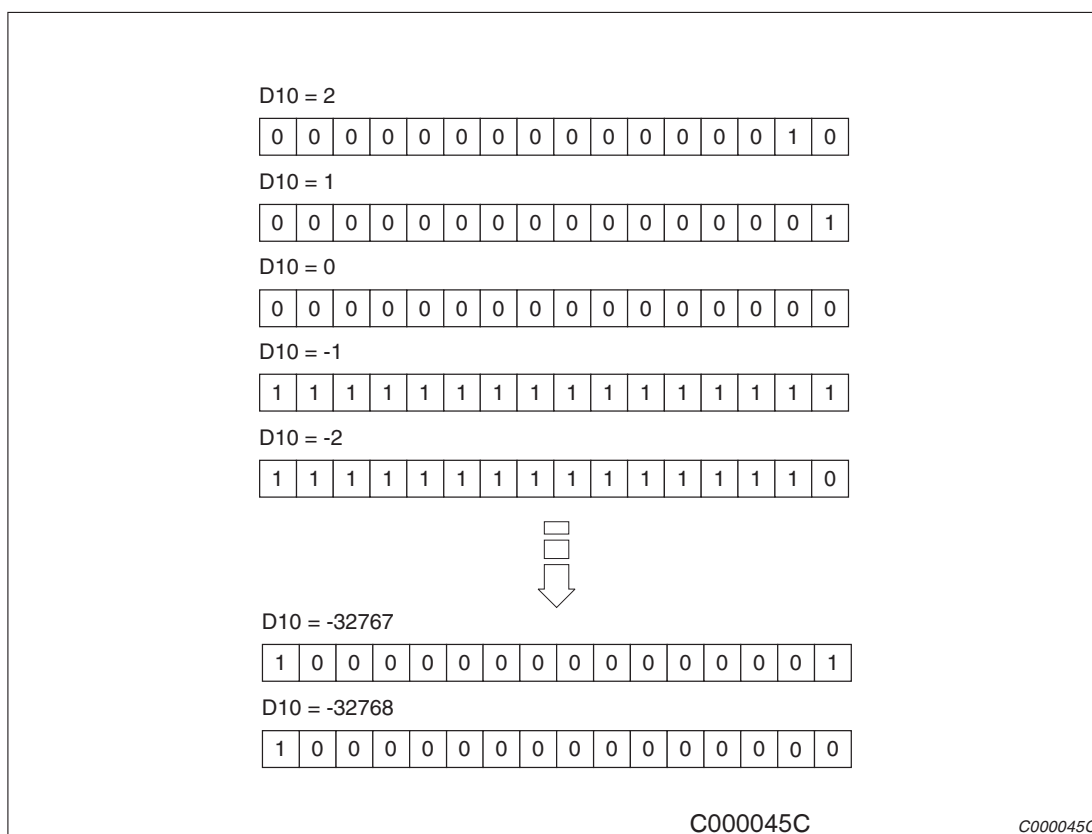


Рис.3-22. Примеры представления отрицательных чисел

△

3.8.8 ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

ПК семейства FX дают возможность работать со значениями чисел в следующих представлениях:

- десятичные числа
- Числа в научном формате
- Числа с плавающей запятой
- Двоичные числа
- Шестнадцатеричные числа
- Формат BCD (числа в двоично-десятичном коде)

Битовый пример

ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ПК

Микропроцессор (центральный процессор) ПК в основном обрабатывает только двоичную информацию. Самый малый элемент двоичной информации называется битом. Бит может представляться двумя состояниями сигнала "0" и "1"

Все числовые значения, которых нет в двоичном формате, преобразовываются поэтому в ПК в двоичный формат (кодируются).

УКАЗАНИЕ

ПК внутри представляет все числа как 16-ти или 32-х разрядные двоичные числа или как битовый пример (16-ти и 32-х битовые форматы).

В следующих разделах будут рассмотрены различные системы чисел и преобразования значений чисел (кодирование) между этими числовыми системами.

База: 10

Цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Область значений:

- 16-ти битный формат: -32 768 ... +32 767
- 32-х битный формат: -2 147 483 648 ... +2 147 483 647

ПРИМЕР ▾

351 (десятичное) = 3 x 10² + 5 x 10¹ + 1 x 10⁰

ЧИСЛА В НАУЧНОМ ФОРМАТЕ

Этот формат рассчитан на научное представление особенно больших и особенно малых чисел. Представление выполняется в 32-х битном формате с плавающей запятой.

Формат: Мантисса $\times 10^{\text{Экспонента}}$

Область значений:

- Мантиссы: W1000 ... 9999, или 0
- Экспонента: -41 ... +35

ПРИМЕР ▾

Скорость света:

- как десятичное число: 299 792 458 м/с
- в научном формате: 2998×10^5 м/с

Здесь 2998 является мантиссой и 5 - экспонентой. В регистре данных число сохраняется, например, в форме $D120 \times 10D^{121}$.

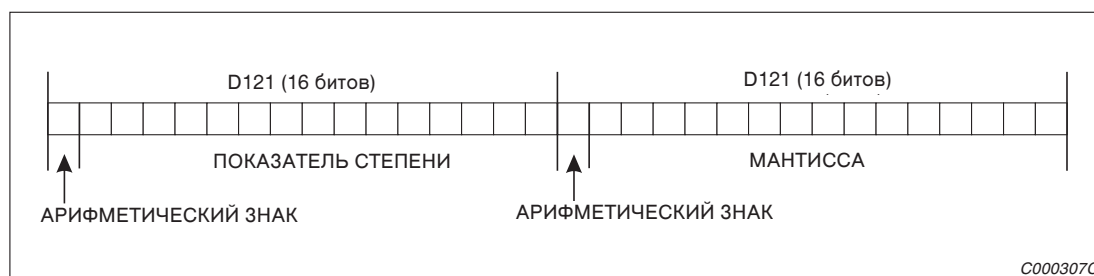


Рис.3-23. Расположение в регистре данных

△

СИСТЕМА ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Операции с числами очень быстро превышают допустимые значения областей, серия FX предлагает дополнительное представление очень больших и очень малых чисел в формате с плавающей запятой, как это применяется в персональных и микро- компьютерах.

Формат системы чисел с плавающей запятой запоминает мантиссу и экспоненту как двоичные числа в 32-х битовых двойных словах, где мантисса имеет 23 бита, а экспонента 8 бит.

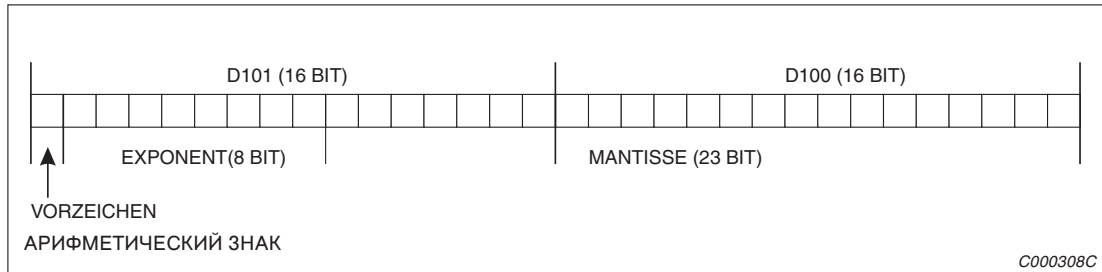


Рис.3-24. Размещение бит в регистре данных

Формат: + - Мантисса x 2^{Экспонента}

Область значений:

Мантиссы: $1 \times 2^0 + A_{22} \times 2^{-1} + A_{21} \times 2^{-2} + \dots + A_0 \times 2^{-23}$

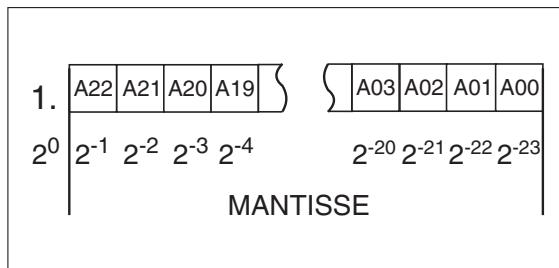


Рис.3-25.
Мантисса

C000309C

Экспонента: $(E_7 \times 2^7 + E_6 \times 2^6 + \dots + E_0 \times 2^0) - 127$, что составляет от -126 до +127

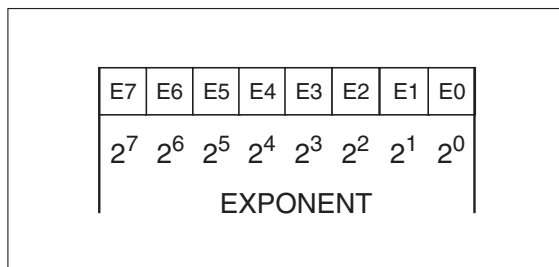
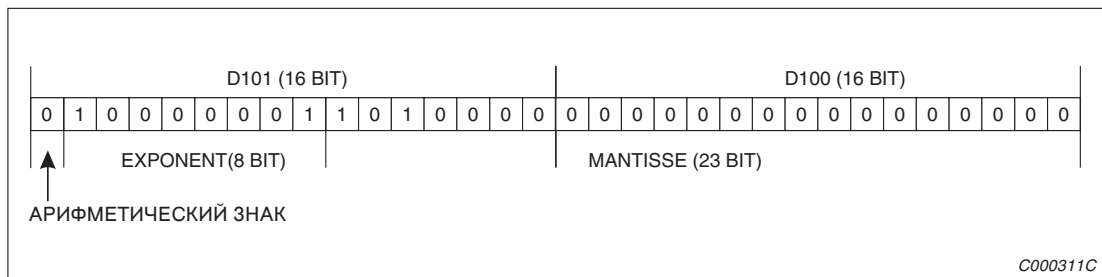


Рис.3-26.
Экспонента

C000310C

ПРИМЕР ▾D101 = 16592 = 40D0_{HEX}D100 = 0 = 0000_{HEX}**Рис.3-27.** Размещение бит в регистре данных

Бит знака числа равен 0 - положительное значение

Экспонента записана числом 10000001, это соответствует

$$\begin{aligned} & (1 \times 2^7 + 0 \times 2^6 + \dots + 1 \times 2^0) - 127 \\ & = (128 + 0 + \dots + 1) - 127 \\ & = 2 \end{aligned}$$

Мантисса ли

1×2^0 + представлена числом 101000000000000000000000 это соответствует $1,101_{\text{BIN}}$ и $1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + \dots + 0 \times 2^{-23}$

$$= 1,625$$

В результате число равно $+ 1,625 \times 2^2 = 6,5$.

△

ДВОИЧНАЯ СИСТЕМА ЧИСЕЛ

База: 2

Цифры: 0, 1

ПРИМЕР ▾

11001 (двоичное число)

$$11001 \text{ (двоичное число)} = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$11001 \text{ (двоичное число)} = 16 + 8 + 1$$

$$1001 \text{ (двоичное число)} = 25 \text{ (десятичное)}$$

△

ДВОИЧНОЕ КОДИРОВАНИЕ: ДЕСЯТИЧНОЕ ЧИСЛО → ДВОИЧНОЕ ЧИСЛО**ПРИМЕР** ▾

30 (десятичное)

$$30 : 2 = 15 \text{ остаток } 0$$

$$15 : 2 = 7 \text{ остаток } 1$$

$$7 : 2 = 3 \text{ остаток } 1$$

$$3 : 2 = 1 \text{ остаток } 1$$

$$1 : 2 = 0 \text{ остаток } 1$$

$$30 \text{ (десятичное)} = 11110 \text{ (двоичное)}$$

△

КОДИРОВАНИЕ: ДВОИЧНОЕ ЧИСЛО → ДЕСЯТИЧНОЕ ЧИСЛО**ПРИМЕР** ▾

111000 (двоичное)

$$111000 \text{ (двоичное)} = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$111000 \text{ (двоичное)} = 32 + 6 + 8$$

$$111000 \text{ (двоичное)} = 56 \text{ (десятичное)}$$

△

ВОСЬМЕРИЧНАЯ СИСТЕМА ЧИСЕЛ

База: 8

Цифры: 0, 1, 2, 3, 4, 5, 6, 7

ПРИМЕР ▾

245 (осьмеричное)

$$245 \text{ (осьмеричное)} = 2 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$$

$$245 \text{ (осьмеричное)} = 128 + 32 + 5$$

$$245 \text{ (осьмеричное)} = 165 \text{ (десятичное)}$$

△

КОДИРОВАНИЕ: ДЕСЯТИЧНОЕ ЧИСЛО → ВОСЬМЕРИЧНОЕ ЧИСЛО**ПРИМЕР** ▾

30 (десятичное)

$$30 : 8 = 3 \text{ остаток } 6$$

$$3 : 8 = 0 \text{ остаток } 3$$

$$30 \text{ (десятичное)} = 36 \text{ (осьмеричное)}$$

△

КОДИРОВАНИЕ: ВОСЬМЕРИЧНОЕ ЧИСЛО → ДЕСЯТИЧНОЕ ЧИСЛО**ПРИМЕР** ▾

374 (осьмеричное)

$$374 \text{ (осьмеричное)} = 3 \times 8^2 + 7 \times 8^1 + 4 \times 8^0$$

$$374 \text{ (осьмеричное)} = 192 + 56 + 4$$

$$374 \text{ (осьмеричное)} = 252 \text{ (десятичное)}$$

△

ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА ЧИСЕЛ

База: 16

Цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

(A=10, B=11, C=12, D=13, E=14, F=15)

ПРИМЕР ▾ 1E (шестнадцатеричное)1E (шестнадцатеричное) = $1 \times 16^1 + 14 \times 16^0$

1E (шестнадцатеричное) = 16 + 14

1E (шестнадцатеричное) = 30 (десятичное) △**КОДИРОВАНИЕ: ДЕСЯТИЧНОЕ ЧИСЛО → ШЕСТНАДЦАТЕРИЧНОЕ ЧИСЛО****ПРИМЕР** ▾ 63 (десятичное)

63 : 16 = 3 остаток 15 R F (шестнадцатеричное)

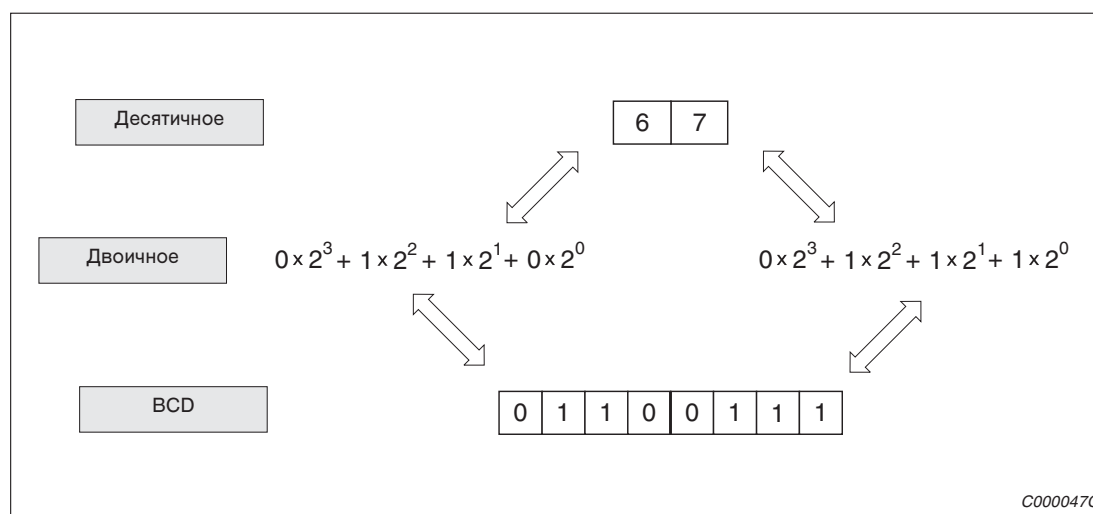
3 : 16 = 0 остаток 3 R 3 (шестнадцатеричное)

63 (десятичное) = 3F (шестнадцатеричное) △**КОДИРОВАНИЕ: ШЕСТНАДЦАТЕРИЧНОЕ ЧИСЛО → ДЕСЯТИЧНОЕ ЧИСЛО****ПРИМЕР** ▾ 7A (шестнадцатеричное)7A (шестнадцатеричное) = $7 \times 16^1 + 10 \times 16^0$

7A (шестнадцатеричное) = 112 + 10

7A (шестнадцатеричное) = 122 (десятичное) △**BCD - ФОРМАТ (ДВОИЧНО-ДЕСЯТИЧНЫЙ КОД)**

В BCD-формате (двоично-десятичном коде) каждая цифра десятичного числа представляется четырехбитным двоичным числом. При четырехбитном представлении имеется возможность двоично кодировать десятичные цифры от 0 до 15. Однако в BCD-формате допустимо кодирование только десятичных цифр от 0 до 9.

КОДИРОВАНИЕ: ДЕСЯТИЧНОЕ ЧИСЛО → BCD - ФОРМАТ**ПРИМЕР** ▾ 67 (десятичное) <=> Цифры: 6, 7**Рис.3-28.** Кодирование десятичного числа в BCD-формат и наоборот

3.9 УКАЗАТЕЛЬ

Указатель программируется вместе с инструкциями CJ-перехода или CALL-вызова.

Указатель является адресом места перехода, в котором маркируется место перехода или подпрограмма в программе (Указатель-Маркировка).

3.9.1 АДРЕСАЦИЯ УКАЗАТЕЛЯ

В контроллерах серии FX1S имеются метки указателя P0...P63 (64 адреса). В серии FX1N, FX2N и FX2NC диапазон указателей охватывает 128 адресов (от P0 до P127), а в контроллере FX3U - 4096 адресов (от P0 до P4095). С помощью команды CJ и адреса указателя P63 во всех контроллерах можно перейти в конец программы. В этом случае команду END не требуется снабжать меткой указателя.

УКАЗАНИЕ

Одинаковые указатели-маркировки не могут использоваться несколько раз в одной программе ПК.

3.9.2 УРОВЕНЬ ПОДПРОГРАММ (ВЛОЖЕННОСТИ)

Во время исполнения прерывания все другие прерывания не активны. Чтобы сохранить стековые запоминания прерываний должны быть запрограммированы инструкции EI-DI внутри программы прерывания. Прерывания могут накапливаться в двух уровнях подпрограмм.

3.10 ПРЕРЫВАНИЕ-УКАЗАТЕЛЬ

С помощью прерывания-указателя можно выполнить переход внутри программы ПК к прерывателю-программы. (см.также раздел 6.2.4)

3.10.1 АДРЕСАЦИЯ ПРЕРЫВАНИЯ-УКАЗАТЕЛЯ

FX1S/FX1N**MELSEC FX1S и FX1N:**

Имеется 6 прерывателей-указателей. Адресация прерывателей-указателей должна выполняться согласно следующим указаниям:

Прерыватель-указатель: I ① 0 ②

- ① Адреса от 0 до 3; соответственно входы от X0 до X3
- ② 0: = прерывание при падающем фронте входного сигнала
1: = прерывание при возрастающем фронте входного сигнала

ПРИМЕР ▾

Прерыватель-указатель: I201

Программа-прерывания, вызываемая прерывателем-указателем выполняется при возрастающем фронте сигнала на входе X2.

Возврат в главную программу происходит после того, как будет выполнена инструкция IRET.



FX1S

Для ПК серии FX1S с номеров версий от 1.00 до 1.30 должна расширяться программа, если применяется прерыватель-указатель.

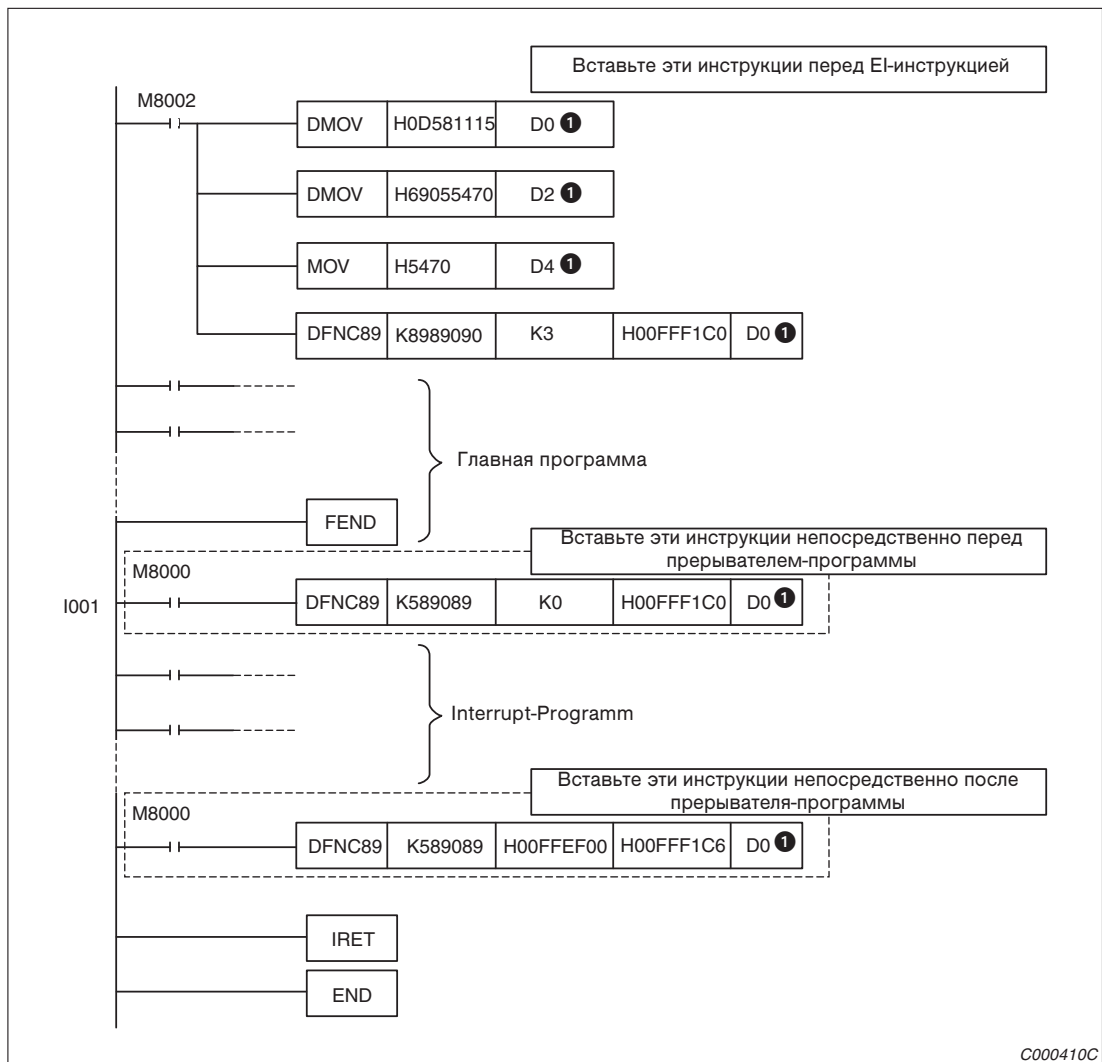


Рис.3-29. Необходимое расширение программы для FX1S ((V1.00 до V1.30) при применении прерывателя-указателя

- ❶ Регистры от D0 до D4 записываются посредством этих частей программ и не могут применяться в программе ПК.

УКАЗАНИЕ

У ПК FX1S, начиная с номера версии 1.40 эти расширения программы не нужны.

FX1N

Для ПК серии FX1N версии 1.00 при применении прерывателя-указателя программа должна расширяться с помощью приведенных ниже инструкций.

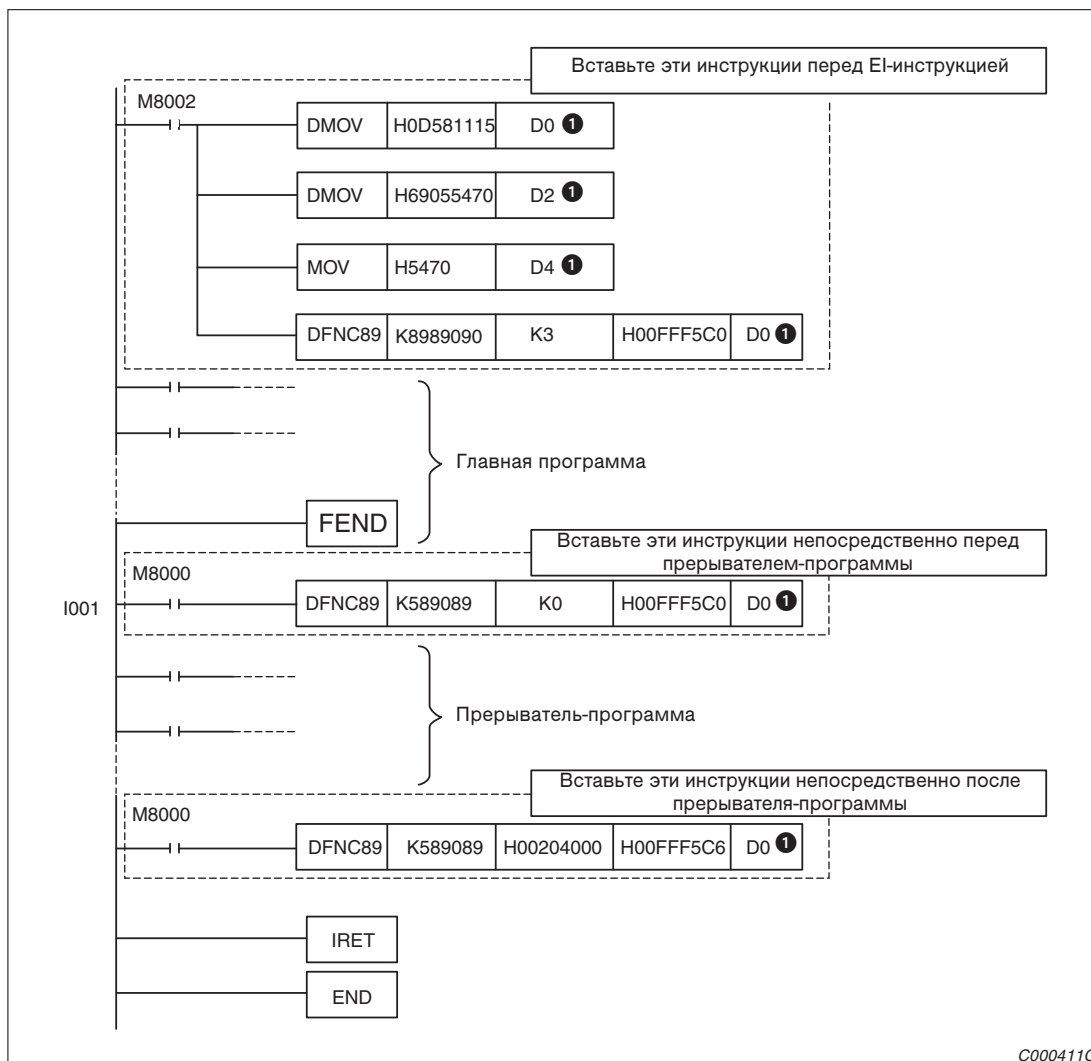


Рис.3-30. Необходимое расширение программы для FX1N (V1.00) при применении прерывателя-указателя

① Регистры от D0 до D4 записываются посредством этих частей программ и не могут применяться в программе ПК.

УКАЗАНИЕ

У ПК FX1N, начиная с номера версии 1.10 эти расширения программы не нужны.

FX2N/2NC/3U MELSEC FX2N и FX2NC:

В контроллерах FX2N, FX2NC и FX3U имеются 15 указателей прерывания.

При адресации указателя прерывания следует различать три группы:

- прерывания по входу (6 указателей прерывания)
Эти прерывания активируются в результате перемены сигнала на входе контроллера.
Формат указателя прерывания по входу: I ① 0 ②

- ① Адреса от 0 до 5
Каждый адрес может использоваться только один раз.
- ② 0: прерывание при падающем фронте
1: прерывание при возрастающем фронте

ПРИМЕР ▾

Прерыватель-указатель: I001

Программа-прерывания, вызываемая прерывателем-указателем, выполняется при возрастающем фронте сигнала на входе X0.

Возврат в главную программу происходит после того, как будет выполнена инструкция IRET.

- прерывания по таймеру (3 указателя прерывания)
Прерывание активируется с равными интервалами времени (от 1 до 99 мс).
Формат указателя для прерывания по таймеру: I ① ②

△

- ① Адреса от 6 до 8
Каждый адрес может использоваться только один раз.
- ② От 10 до 99 мс

ПРИМЕР ▾

Прерыватель-указатель: I610

Программа-прерывания, вызываемая прерывателем-указателем, выполняется в интервалах по 10 мс.

Возврат в главную программу происходит после того, как будет выполнена инструкция IRET.

△

УКАЗАНИЕ

Прерыватель-указатель программируется после инструкции FTND. При этом может выполняться не более 9 прерывателей-указателей. Больше двух уровней разветвления не допустимы.

- прерывания по счетчику (6 указателей прерывания)
Прерывание активируется, если состояние счетчика соответствует значению, указанному в команде сравнения.
Формат указателя прерывания по счетчику: I 0 ① 0

- ① Адреса от 1 до 6
Прерыватель-счетчик может применяться как операнд для включения (HSCS, FNC 53) или отключения (HSCR, FNC 54) посредством высокоскоростного счетчика. Для выключения прерывателя-счетчика включается специальный меркер M8059.

ПРИМЕР ▾

Прерыватель-указатель: I030

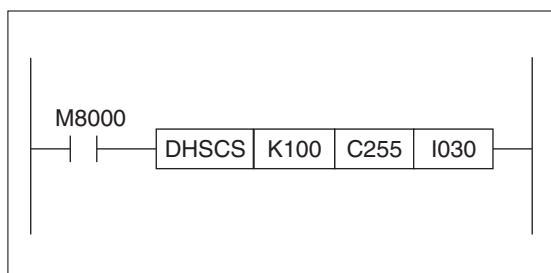


Рис.3-31.
Пример программирования при использовании прерывателя-счетчика

C000333C

Программа-прерывания, вызываемая прерывателем-указателем I030, выполняется при достижении высокоскоростным счетчиком C255 значения, заданного в K100. △

УКАЗАНИЕ

Пожалуйста, обратите внимание на раздел 6.7.4 с более подробной информацией для использования команд включения и отключения посредством высокоскоростных счетчиков.

ВЫКЛЮЧЕНИЕ ЛЮБЫХ ПРЕРЫВАНИЙ

Любые прерывания с помощью включения соответствующих специальных меркеров можно выключать в зависимости от времени или во время непрерывного процесса управления. Соответствующие специальные меркеры приведены в главе 6. Для всех моделей ПК есть первый специальный меркер M8050, который выключает прерывания I0①②.

УКАЗАНИЕ

Никогда не включайте специальный меркер, не зная с уверенностью его функции. Не все ПК всегда работают с одинаковыми специальными функциями.

Прерывания высокоскоростных счетчиков всегда могут отключаться только как отдельные группы с помощью специального меркера M8059.

3.11 РАЗВЕТВЛЕНИЯ

С помощью операций подпрограмм могут реализовываться разветвления уровней внутри программы. Операции подпрограмм применяются вместе с инструкциями MC и MCR.

Точное применение операций подпрограмм описано в разделе 4.9 в описании контролируемых условий (MC, MCR).

3.11.1 АДРЕСАЦИЯ ОПЕРАНДОВ ПОДПРОГРАММ

Имеется восемь операндов подпрограмм от N0 до N7.

4 НАБОР БАЗОВЫХ КОМАНД

4.1 ОБЩИЕ УКАЗАНИЯ

Эта глава описывает набор базовых команд семейства ПК FX. С помощью инструкций из набора базовых команд могут программироваться все логические базовые связи. Инструкции из набора базовых команд могут срабатывать только по адресам операндов.

4.1.1 ПОЯСНЕНИЯ К ТАБЛИЦАМ НАБОРА БАЗОВЫХ КОМАНД

Все базовые команды представлены в табличной форме на следующих двух страницах. Этот раздел кратко поясняет структуру обзорных таблиц.

②		①					
		SET					
		③ Включение; Включение операнда					
④		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●	
⑤		⑥					
Операнды		Шаги программы			Примечание		
Y, M, S		SET (Операнд Y, M)			1		
		SET (Специальный меркер S)			2		

① ИНСТРУКЦИЯ

На этом месте указывается имя инструкции, которое используется при программировании на языке Листинга Инструкций - (AWL).

② СИМВОЛЫ КОНТАКТНОЙ СХЕМЫ

Символы контактной схемы применяются при программировании на языке (KOP). Символы контактной схемы состоят из инструкции и используемого операнда.

③ ЗНАЧЕНИЕ

Здесь находится краткое описание значения инструкции.

④ ЦЕНТРАЛЬНЫЙ ПРЦЕССОР

Здесь знаком 2 характеризуется серия ПК MELSEC, в которой эта инструкция выполняется.

⑤ ОПЕРАНД

На этом поле указывается операнд, примененный вместе с инструкцией.

⑥ ШАГИ ПРОГРАММЫ

Указывается число шага программы, которое требуется до полного выполнения инструкции.

ОБЗОР БАЗОВЫХ КОМАНД

Инструкция	Символы Контактной Схемы	Значение	Операнды	Шагов Программы	Ссылка на Раздел
LD		ЗАГРУЗКА; Начало логических связей (начало логического выражения) с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.2
LDI		НЕТ ЗАГРУЗКИ; Начало логических связей (начало логического выражения) с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.2
OUT		ВЫХОД; Выход (выдача), присвоение выводу результата логического выражения.	Y, M, S, T, C	Y, M: 1 S, Спец. меркер: 2 T: 3, C: 3 C (32бита): 5	Разд. 4.3
AND		И; И - логическая связь с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.4
ANI		НЕ И; И - логическая связь с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.4
OR		ИЛИ; ИЛИ - логическая связь с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.5
ORI		НЕ ИЛИ; ИЛИ - логическая связь с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.5
LDP		ЗАГРУЗКА; (импульсно) Начало логических связей с опросом по возрастающему (переднему) фронту сигнала	X, Y, M, S, T, G	2	Разд. 4.6
LDF		ЗАГРУЗКА; (импульсно) Нач-ло логических связей с опросом по падающему (заднему) фронту сигнала	X, Y, M, S, T, G	2	Разд. 4.6
ANP		И; (импульсно) И - логическая связь с опросом по возрастающему (переднему) фронту сигнала	X, Y, M, S, T, G	2	Разд. 4.7
ANF		И; (импульсно) И - логическая связь с опросом по падающему (заднему) фронту сигнала	X, Y, M, S, T, G	2	Разд. 4.7
ORP		ИЛИ; (импульсно) ИЛИ - логическая связь с опросом по возрастающему (переднему) фронту сигнала	X, Y, M, S, T, G	2	Abs. 4.8

Табл.4-1. Обзор базовых команд (часть 1)


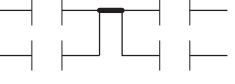
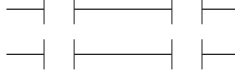
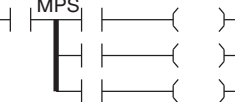
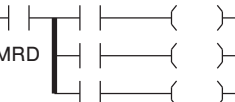
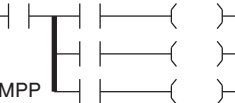
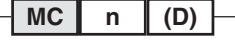

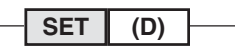

Инструкция	Символы Контактной Схемы	Значение	Операнды	Шагов Программы	Ссылка на Раздел
ORF		ИЛИ; (импульсно) ИЛИ - логическая связь с опросом по падающему (заднему) фронту сигнала	X, Y, M, S, T, G	2	Разд. 4.8
ANB		И-БЛОК; Команда связи: последовательное включение параллельных связей	–	1	Разд. 4.9
ORB		ИЛИ-БЛОК; Команда связи: параллельное включение последовательных связей	–	1	Разд. 4.10
MPS		ДВИЖЕНИЕ ВНИЗ ПО СТЕКУ; Накопление результатов логических связей	–	1	Разд. 4.11
MRD		ЧТЕНИЕ ВНИЗ ПО СТЕКУ; Чтение результатов логических связей	–	1	Разд. 4.11
MPP		ВСПЛЫВАЮЩИЙ СТЕК; Чтение и стирание накопленных результатов логических связей	–	1	Разд. 4.11
MC		МАСТЕР КОНТРОЛЬ; Включение (запуск) условий контроля	Y, M, Нет спец. меркера	3	Разд. 4.12
MCR		МАСТЕР КОНТРОЛЬ СБРОС; Отключение (сброс) условий контроля	N	2	Разд. 4.12
SET		ВКЛЮЧЕНИЕ; Включение операнда	Y, M, S	Y, M: 1 S, Спец. меркер: 2	Разд. 4.13
RST		ОТКЛЮЧЕНИЕ; Отключение операнда	Y, M, S, D V, Z, T, C	Y, M: 1 D, V, Z, Спец. меркер: 3 T, C: 2	Разд. 4.13

Табл.4-2. Обзор базовых команд (часть 2)

Инструкция	Символы Контактной Схемы	Значение	Операнды	Шагов Программы	Ссылка на Раздел
PLS		ГЕНЕРАЦИЯ ИМПУЛЬСОВ; Формирование одноразового импульса при возрастающем фронте	Y, M	2	Разд. 4.14
PLF		ГЕНЕРАЦИЯ ИМПУЛЬСОВ; Формирование одноразового импульса при падающем фронте	Y, M	2	Разд. 4.14
INV		ИНВЕРСИЯ; Замена результата логических связей на противоположный сигнал	–	1	Разд. 4.15
NOP	–	ПУСТАЯ СТРОКА; Пустая строка без функций	–	1	Разд. 4.16
END		Ende; Конец программы ПК	–	1	Разд. 4.17

Табл.4-3. Обзор базовых команд (часть 3)

4.2 НАЧАЛО ЛОГИЧЕСКИХ ЦЕПЕЙ (LD, LDI)

	LD					
	ЗАГРУЗКА; Начало логических связей с опросом на состояние сигнала "1"					
	LDI					
	НЕТ ЗАГРУЗКИ; Начало логических связей с опросом на состояние сигнала "0"					
	CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●
		●	●	●	●	●
Операнды	Шаги программы			Примечание		
X, Y, M, S, T, C	LD			1		
	LDI			1		

ФУНКЦИЯ

Программирование начала логических связей (начало логического выражения)

ОПИСАНИЕ

- Начало логически связей программируется инструкциями LD или LDI.
- Программирование цепи тока всегда начинается с инструкций LD или LDI.
- LD или LDI инструкции используются также совместно с инструкциями ANB или ORB для пуска разветвления (см. также разделы 4.6 и 4.7).

ПРИМЕР ▾

Применение инструкций LD, LDI

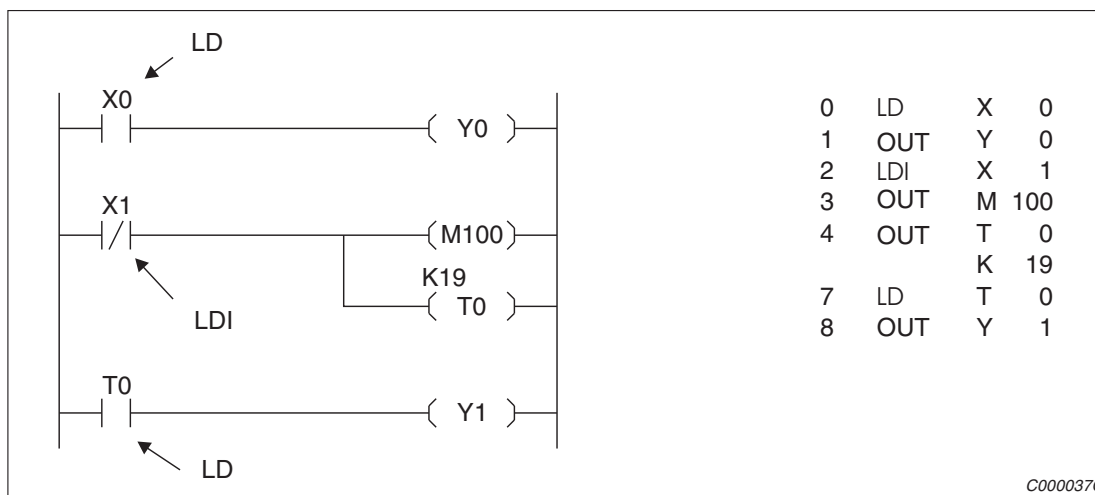


Рис.4-1. Пример программирования с использованием инструкций LD, LDI

Вход X0 опрашивается на состояние сигнала "1". Выход Y0 включается на состояние сигнала "1", как только вход X0 получит значение "1". Вход X1 опрашивается на состояние сигнала "0". Меркер M100 получит состояние сигнала "1", и установленное время таймера T0 начнет отсчитываться, как только вход X1 приобретет значение "1". После отсчета настроенного значения времени (19 x 100 мс = 1,9 с) тайме T0 включит выход Y1 на состояние сигнала "1".

△

4.3 ВЫХОД РЕЗУЛЬТАТА ЛОГИЧЕСКИХ ЦЕПЕЙ (OUT)

		OUT				
		ВЫХОД; Выход, присвоение результата логических связей				
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●
Operanden	Шаги программы				Примечание	
Y, M, S, T, C	Y, M	1	T, C (16бита)	3		
	S,	2	C (32бита)	5		

ФУНКЦИЯ

Присвоение состояния сигнала (включения или отключения выхода) в зависимости от результата логических связей (результата обработки центральным процессором логического выражения).

ОПИСАНИЕ

- С помощью инструкции OUT можно завершить программирование цепи тока (логического выражения).
- Программирование нескольких инструкций OUT как результат обработки логического выражения также возможно.
- Результат логических связей, представленный посредством инструкции OUT, может применяться в следующих шагах программы как состояние входного сигнала, т.е. может многократно опрашиваться во многих логических выражениях.
- Результат логических связей, представленный OUT инструкцией, активен (включен) до тех пор, пока действуют условия его включения.

ПРИМЕР ▾

Использование инструкции OUT

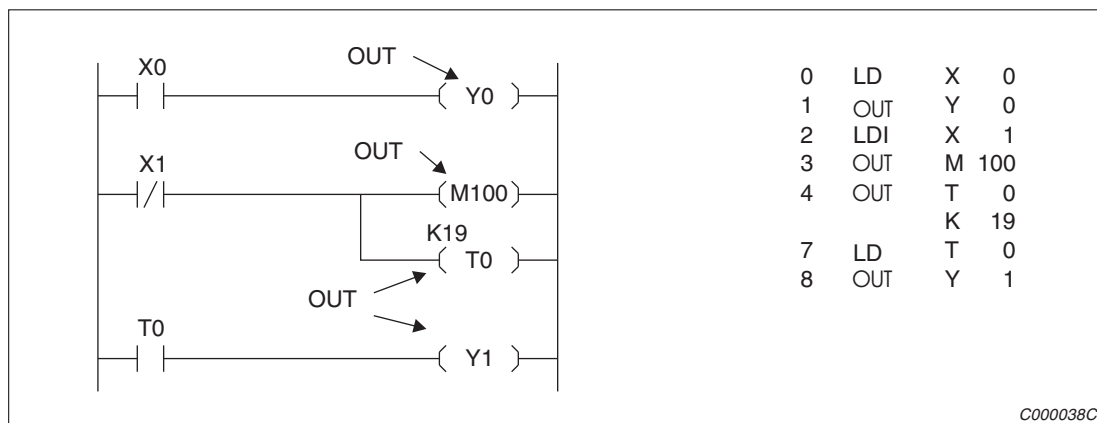


Рис.4-2. Пример программирования с применением инструкции OUT

Вход X0 опрашивается на состояние сигнала "1". Выход Y0 включается на состояние сигнала "1", как только вход X0 получит значение "1".

Вход X1 опрашивается на состояние сигнала "0". Меркер M100 и таймер T0 получит состояние сигнала "1", как только вход X1 приобретет значение "0".

После отсчета настроенного значения времени (19 x 100 мс = 1,9 с) тайме T0 включит выход Y1 на состояние сигнала "1".

КОЛИЧЕСТВО ПРОГРАММНЫХ ШАГОВ ПРИ ПРИМЕНЕНИИ ТАЙМЕРОВ И СЧЕТЧИКОВ

Для инструкций OUT, которые программируются для таймеров и счетчиков, необходимо обращать внимание на то, что в них действуют двухшаговые инструкции. Во втором шаге программы выполнена установка значения времени или счета. Это происходит заданием десятичной константы К.

Программирование таймеров и счетчиков подробно описано в разделах 3.4 и 3.5.

ДВОЙНАЯ ЗАПИСЬ ВЫХОДОВ

При программировании двойной записи одинаковых выходов (их адресов) могут возникнуть проблемы при отработке программы. Следующий пример поясняет эту проблему.

ПРИМЕР ▾

Двойная запись выхода

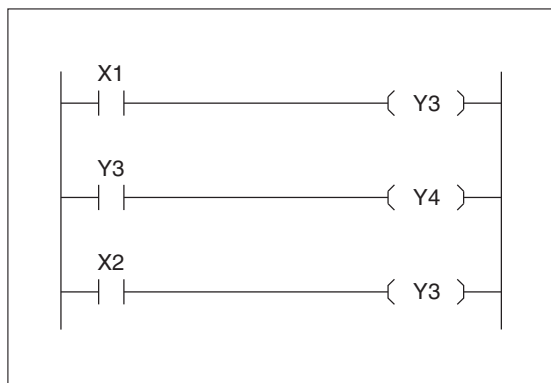


Рис.4-3.
Пример программирования двойной записи выхода

C000050C

Исходим из того, что вход X1 включен (сигнал "1"), а вход X2 отключен (сигнал "0").

Первая запись выхода Y3 активизируется включенным входом X1. В отображении процесса выходов Y3 включен. Соответственно активизируется также выход Y4.

Эта программная последовательность имеет следствием то, что Y3 от X2 отключается, а Y4 остается включенным.

В разделе 2.1 отработка программы ПК детально описана. △

УКАЗАНИЕ

Избегайте двойной записи выходов, так как может привести к помехам при отработке программы.

ПРИМЕР ▾

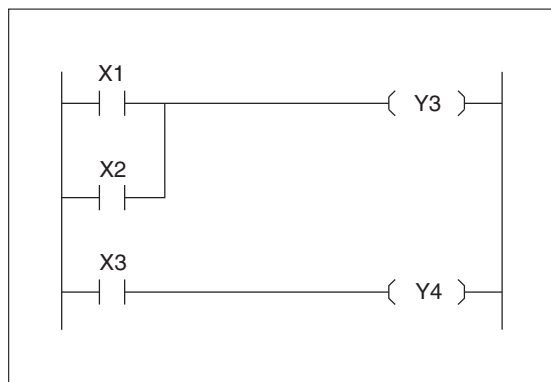




Рис.4-4.
Пример программирования

C000051C

△

4.4 ЛОГИЧЕСКАЯ СВЯЗЬ "И" (AND, ANI)

	AND				
	И; Логическая связь И с опросом на состояние сигнала "1"				
	ANI				
	И НЕ; Логическая связь И с опросом на состояние сигнала "0"				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
X, Y, M, S, T, C	AND-инструкция		1		
	ANI-инструкция		1		

ФУНКЦИЯ

Программирование логической связи "И"

ОПИСАНИЕ

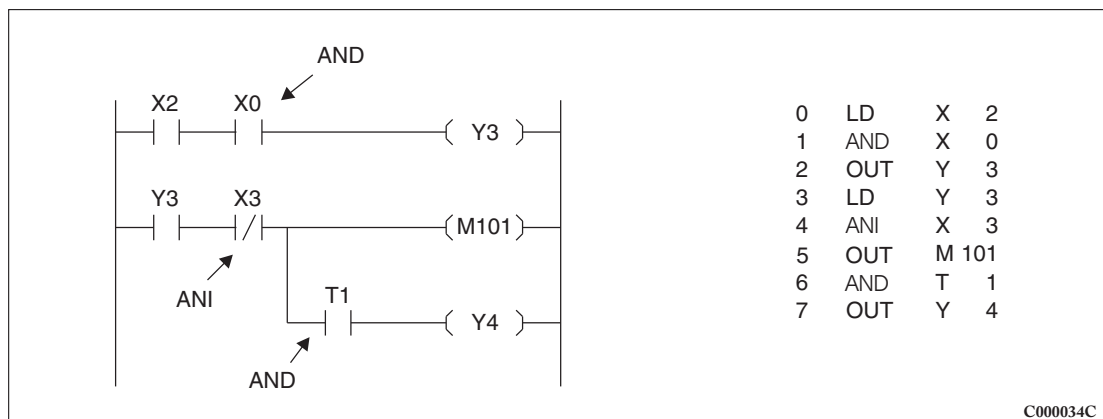
- Инструкции AND и ANI применяются для последовательного включения контактов (логическая связь (операция) "И").
- Обе инструкции представляют логические операции и поэтому не могут программироваться в начале цепи тока. В начале логического выражения программируются инструкции LD или LDI.
- Если желательно включить последовательно несколько один за другим следующих схемных блоков, можно применить также ANB-инструкцию.

УКАЗАНИЕ

В одной последовательной цепи тока может программироваться 10 контактов и максимум 24 цепи (параллельных цепей) на одну катушку, т.е. логическое выражение может иметь 10 x 24 = 240 контактов (аргументов функции).

ПРИМЕР ▾

Применение инструкций AND, ANI

**Рис.4-5.** Пример программирования с использованием инструкций AND и ANI

Выход Y3 приобретет состояние сигнала "1", если выполняются следующие условия:

- Вход X2 приобретет состояние сигнала "1"
- И
- Вход X0 приобретет состояние сигнала "1"

Меркер M101 приобретет состояние сигнала "1", если выполняются следующие условия:



- Выход Y3 приобретет состояние сигнала "1"
- И
- Вход X3 приобретет состояние сигнала "0"

Выход Y4 приобретет состояние сигнала "1", если выполняются следующие условия:

- Выход Y3 приобретет состояние сигнала "1"
- И
- Вход X3 приобретет состояние сигнала "0"
- И
- Контакт таймера T1 приобретет состояние сигнала "0"

△

4.5 ЛОГИЧЕСКИЕ СВЯЗИ "ИЛИ" (OR, ORI)

	OR				
	ЛИ; ИЛИ - логическая связь с опросом на состояние сигнала "1"				
	ORI				
	ИЛИ НЕ; ИЛИ - логическая связь с опросом на состояние сигнала "0"				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
X, Y, M, S, T, C	OR-инструкция		1		
	ORI-инструкция		1		

ФУНКЦИЯ

Программирование логической связи "ИЛИ"

ОПИСАНИЕ

- Инструкции OR и ORI применяются для параллельного соединения контактов (логическая связь (операция) "ИЛИ").
- Обе инструкции представляют логические операции и поэтому не могут программироваться в начале цепи тока. В начале логического выражения программируются инструкции LD или LDI (см. раздел 4.2).
- Если желательно включить параллельно несколько один за другим следующих схемных блоков, можно применить также ORB-инструкцию.

УКАЗАНИЕ

В одном логическом выражении нельзя программировать больше 24 параллельных цепей тока.

ПРИМЕР ▾

Применение инструкций OR, ORI

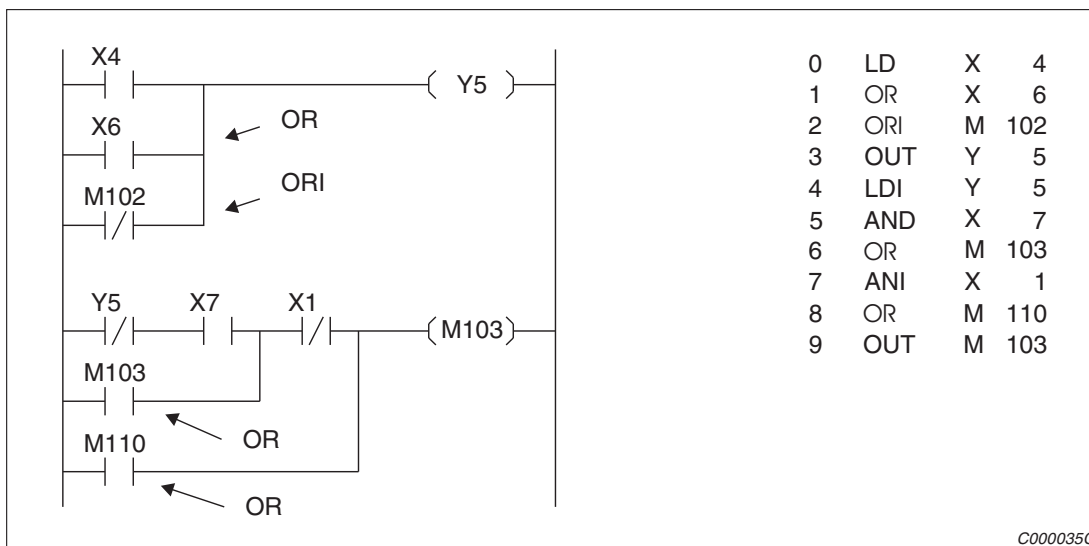


Рис.4-6. Пример программирования с использованием инструкций OR и ORI

Выход Y5 приобретет состояние сигнала "1", если выполняются следующие условия:



- Вход X4 приобретет состояние сигнала "1"
- ИЛИ
- Вход X6 приобретет состояние сигнала "1"
- ИЛИ
- Меркер M102 имеет состояние сигнала "0"

Меркер M103 приобретет состояние сигнала "1", если выполняются следующие условия:

- Выход Y5 приобретет состояние сигнала "0"
- И
- Вход X7 приобретет состояние сигнала "1"
- И
- Вход X1 приобретет состояние сигнала "0"
- ИЛИ
- Вход X1 приобретет состояние сигнала "0"
- И
- Контакт меркера M103 имеет состояние сигнала "1", (меркер своим контактом становится на самопитание – приобретает состояние "1"
- ИЛИ
- Контакт меркера M110 приобретет состояние сигнала "1"



4.6 ИМПУЛЬСНОЕ НАЧАЛО ЛОГИЧЕСКИХ СВЯЗЕЙ (LDP, LDF)

	LDP				
	ЗАГРУЗКА (импульсно); Начало логических связей с опросом по возрастающему (переднему) фронту сигнала				
	LDF				
	ЗАГРУЗКА (импульсно); Начало логических связей с опросом по падающему (заднему) фронту сигнала				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
X, Y, M, S, T, C	LDP			2	
	LDF			2	

ФУНКЦИЯ

Программирование импульсного начала логической связи

ОПИСАНИЕ

- Импульсное начало логических связей программируется инструкциями LDP (возрастающий фронт) или LDF (падающий фронт).
- Обе инструкции должны программироваться в начале цепи .
- Обе инструкции используются также вместе с инструкциями ANB и ORB для запуска разветвлений (см.разделы 4.9 и 4.10).
- LDP-инструкция после положительного фронта сохраняется на время цикла программы (скана).
- LDF-инструкция после отрицательного фронта сохраняется на время цикла программы (скана).

УКАЗАНИЯ

Если вместе с LDP и LDF инструкциями применяется импульсные меркеры от M2800 до M3071, то при многократном программировании каждого импульсного меркера в программе обрабатывается только первый меркер. Это свойство используется вместе с STL- программированием (см.главу 5).

Функции LD-,AND-,OR-,и т.д. инструкций сохраняются.

ПРИМЕР ▾

Применение инструкций LDP, LDF

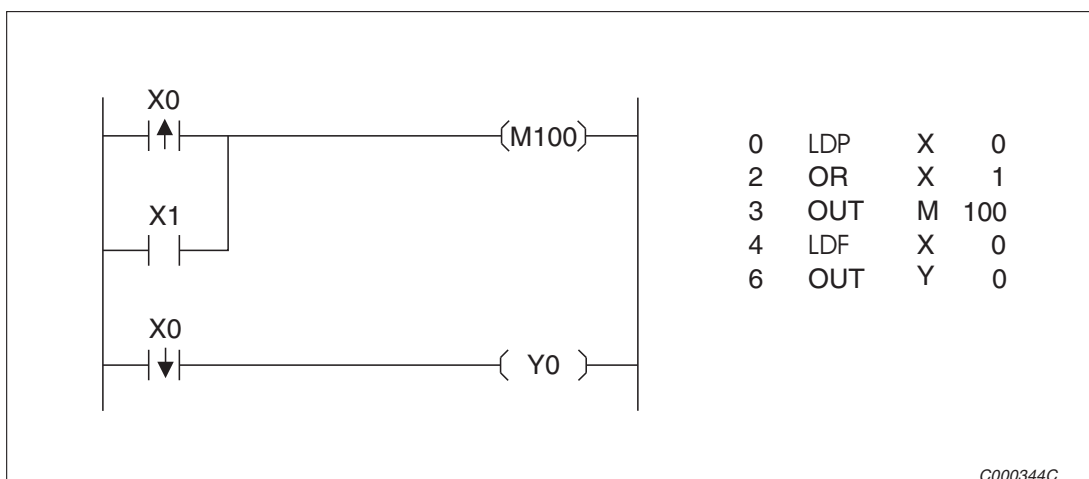




Рис.4-7. Пример программирования с использованием инструкций LDP и LDF

Меркер M100 включается на время включения X1 или при положительном фронте X0 (момента его включения).

Выход Y0 включается при отрицательном фронте X0 (момента его отключения).

△

4.7 ИМПУЛЬСНЫЕ "И" ЛОГИЧЕСКИЕ СВЯЗИ (ANP, ANF)

	ANP				
	И (импульсно), И - логическая связь с опросом по возрастающему (переднему) фронту сигнала				
	ANF				
	И (импульсно), И - логическая связь с опросом по падающему (заднему) фронту сигнала				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
X, Y, M, S, T, C	ANP		2		
	ANF		2		

ФУНКЦИЯ

Программирование импульсной "И" логической связи.

ОПИСАНИЕ

- Импульсная "И" логическая связь программируется инструкциями ANP (возрастающий фронт) или ANF (падающий фронт).
- Обе инструкции могут применяться как AND и ANI инструкции.
- ANP-инструкция обрабатывается положительным фронтом.
- ANF-инструкция обрабатывается отрицательным фронтом.

УКАЗАНИЯ

Если вместе с ANP и ANF инструкциями применяется импульсные меркеры от M2800 до M3071, то при многократном программировании каждого импульсного меркера в программе обрабатывается только первый меркер. Это свойство используется вместе с STL- программированием (см. главу 5).

Функции LD-,AND-,OR-, и.т.д инструкций сохраняются.

ПРИМЕР ▾

Применение инструкций ANP, ANF

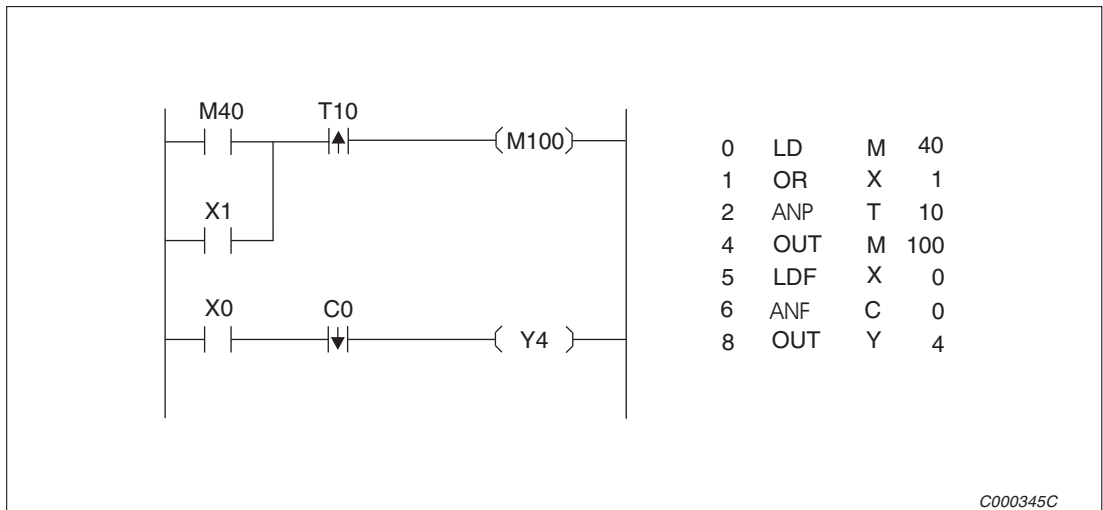




Рис.4-8. Пример программирования с использованием инструкций ANP, ANF

Меркер M100 включается при включенном меркере M40 или включенном входе X1 и возрастающем фронте контакта таймера T10.

Выход Y4 включается при включенном входе X0 и отрицательном фронте контакта счетчика C0.



4.8 ИМПУЛЬСНЫЕ "ИЛИ" ЛОГИЧЕСКИЕ СВЯЗИ (ORP, ORF)

	ORP				
	ИЛИ (импульсно); ИЛИ - логическая связь с опросом по возрастающему (переднему) фронту сигнала				
	ORF				
	ИЛИ (импульсно); ИЛИ - логическая связь с опросом по падающему (заднему) фронту сигнала				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
X, Y, M, S, T, C	ORP		2		
	ORF		2		

ФУНКЦИЯ

Программирование импульсной "ИЛИ" логической связи

ОПИСАНИЕ

- Импульсная "ИЛИ" логическая связь программируется инструкциями ORP (возрастающий фронт) или ORF (падающий фронт).
- Обе инструкции могут применяться как OR и ORI инструкции.
- ORP-инструкция обрабатывается положительным фронтом.
- ORF-инструкция обрабатывается отрицательным фронтом.

УКАЗАНИЯ

Если вместе с ORP и ORF инструкциями применяются импульсные меркеры от M2800 до M3071, то при многократном программировании каждого импульсного меркера в программе обрабатывается только первый меркер. Это свойство используется вместе с STL- программированием (см. главу 5).

Функции LD-, AND-, OR-, и т. д. инструкций сохраняются.

ПРИМЕР ▾

Применение инструкций ORP, ORF

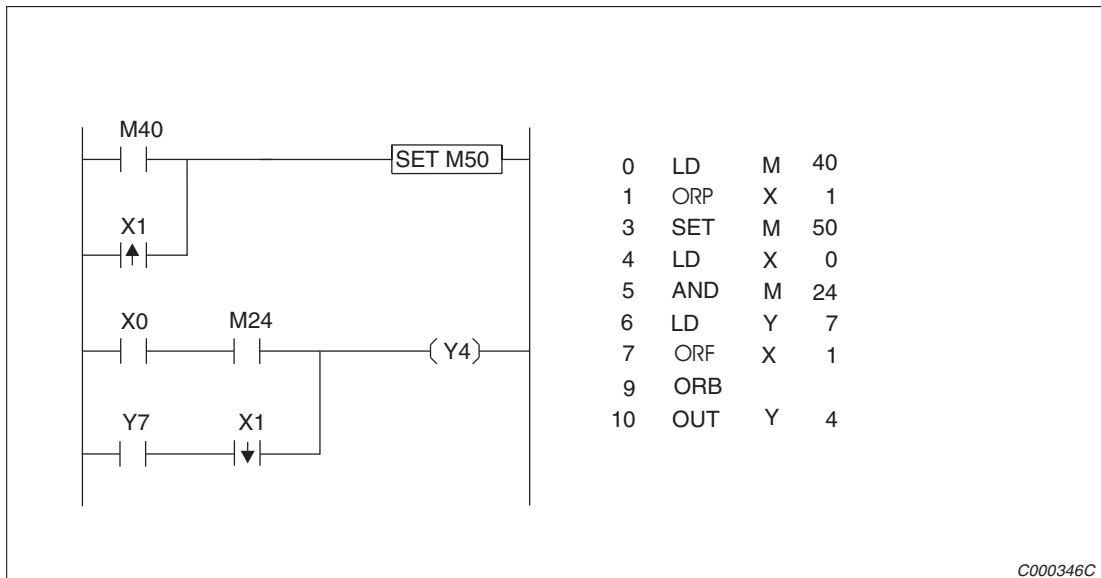


Рис.4-9. Пример программирования с использованием инструкций ORP, ORF

Меркер M50 включается по инструкции SET при включенном меркере M40 или возрастающем фронте входа X1.

Выход Y4 включается при включенном входе X0 и включенном меркере M24 или включенном выходе Y7 и отрицательном (падающем) фронте X1.

△

4.9 ЛОГИЧЕСКИЕ СВЯЗИ "И-БЛОКИ" (ANB)

		ANB				
		И-блок; Команда связи: последовательное включение параллельных связей				
Операнды	CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●
Шаги программы		Примечание				
—		ANB-инструкция			1	

ФУНКЦИЯ

Последовательное включение параллельных блоков логических связей

ОПИСАНИЕ

- Отдельные блоки, параллельно включенных элементов, заносятся в программу отдельно. Чтобы эти блоки соединить последовательно после каждого блока программируется ANB инструкция.
- Начало разветвления программируется с помощью инструкций LD или LDI (см.раздел 4.2).
- ANB-инструкция является независимой и не требует ввода никаких операндов.
- ANB-инструкция внутри всей программы может программироваться многократно.
- В контактной схеме ANB-инструкция изображается как последовательное соединение. ANB-инструкция, имеющаяся на языке листинга инструкций (AWL), при конвертировании в контактную схему появляется автоматически и изображается как перемычка.

УКАЗАНИЕ

Если программируется несколько отдельных блоков непосредственно один за другим, то нужно ограничить число LD и LDI инструкций и, соответственно, также число ANB-инструкций до 8.

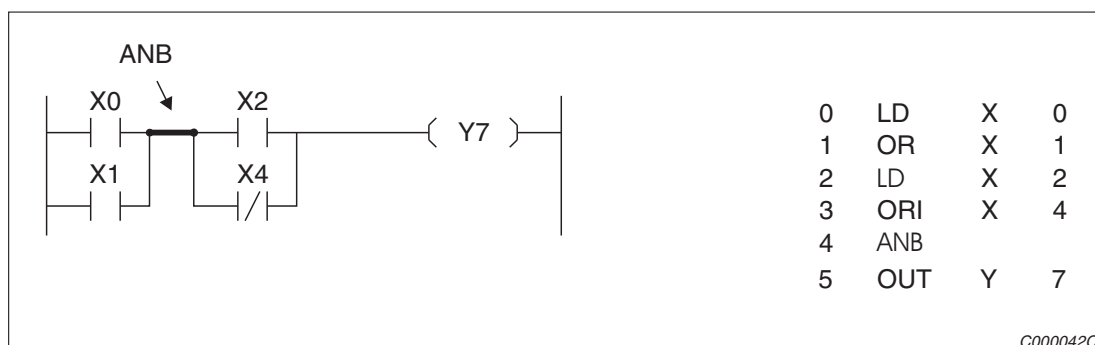


Рис.4-10. Пример программирования с использованием инструкции ANB

4.10 ЛОГИЧЕСКИЕ СВЯЗИ "ИЛИ-БЛОКИ" (ORB)

	ORB				
	ИЛИ-блок; Команда связи: параллельное включение последовательных связей				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
—	ORB-инструкция			1	

ФУНКЦИЯ

Параллельное включение последовательно включенных блоков логических связей

ОПИСАНИЕ

- Если несколько последовательных блоков включаются параллельно, то нужно после программирования каждого отдельного блока вводить ORB инструкцию.
- Начало разветвления программируется с помощью инструкций LD или LDI (см. раздел 4.2).
- ORB-инструкция является независимой и не требует ввода никаких операндов.
- ORB-инструкция внутри всей программы может программироваться многократно.
- В контактной схеме ORB-инструкция изображается как параллельное соединение. ORB-инструкция, имеющаяся на языке листинга инструкций (AWL), при конвертировании в контактную схему появляется автоматически и изображается как перемычка.

УКАЗАНИЕ

Если программируется несколько отдельных блоков непосредственно один за другим, то нужно ограничить число LD и LDI инструкций и, соответственно, также число ORB-инструкций до 8.

	Рекомендуемое программирование	Неблагоприятное программирование
	0 LD X 0	0 LD X 0
	1 AND X 1	1 AND X 1
	2 LD X 2	2 LD X 2
	3 AND X 3	3 AND X 3
	4 ORB	4 LDI X 4
	5 LDI X 4	5 AND X 5
	6 AND X 5	6 ORB
	7 ORB	7 ORB
8 OUT Y 6	8 OUT Y 6	

C000043C

Рис.4-11. Пример программирования с использованием инструкции ORB

4.11 ОБРАБОТКА РЕЗУЛЬТАТОВ ЛОГИЧЕСКИХ СВЯЗЕЙ (MPS, MRD, MPP)

	MPS				
	Движение вниз по стеку; Накопление результатов логических связей				
	MRD				
	Чтение вниз по стеку; Чтение результатов логических связей				
	MPP				
	Всплывающий стек; Чтение и стирание накопленных результатов логических связей				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
●	●	●	●	●	●
MPP					
Всплывающий стек; Чтение и стирание накопленных результатов логических связей					
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
●	●	●	●	●	●
Операнды	Шаги программы			Примечание	
—	MPS-инструкция		1		
	MRD-инструкция		1		
	MPP-инструкция		1		

ФУНКЦИЯ

Инструкции MPS, MRD, MPP служат для того, чтобы создавать уровни логических связей (например, после одного начального логического выражения создать несколько логических выражений на выходе, т.е. включать несколько выходов - катушек)

ОПИСАНИЕ

- С помощью инструкции MPS запоминается предыдущий результат логических связей (обработки логического выражения).
- С помощью инструкции MRD возможно прочтение нескольких частных разветвлений между началом (MPS) и концом (MPP) разветвления, учитывающих на каждом разветвлении результат обработки логического выражения до MPS.
- Последнее частное разветвление создается MPP инструкцией.
- Открывшееся с помощью MPS инструкции разветвление всегда должно быть закрыто MPP инструкцией.
- Все три инструкции не требуют никаких операндов.
- В контактной схеме эти инструкции не изображаются. Если программирование выполняется в контактной схеме разветвления используются как обычно. MPS-, MRD-, и MPP-инструкции на языке листинга инструкций (AWL) появляются автоматически, после того как программа конвертируется в контактную схему.

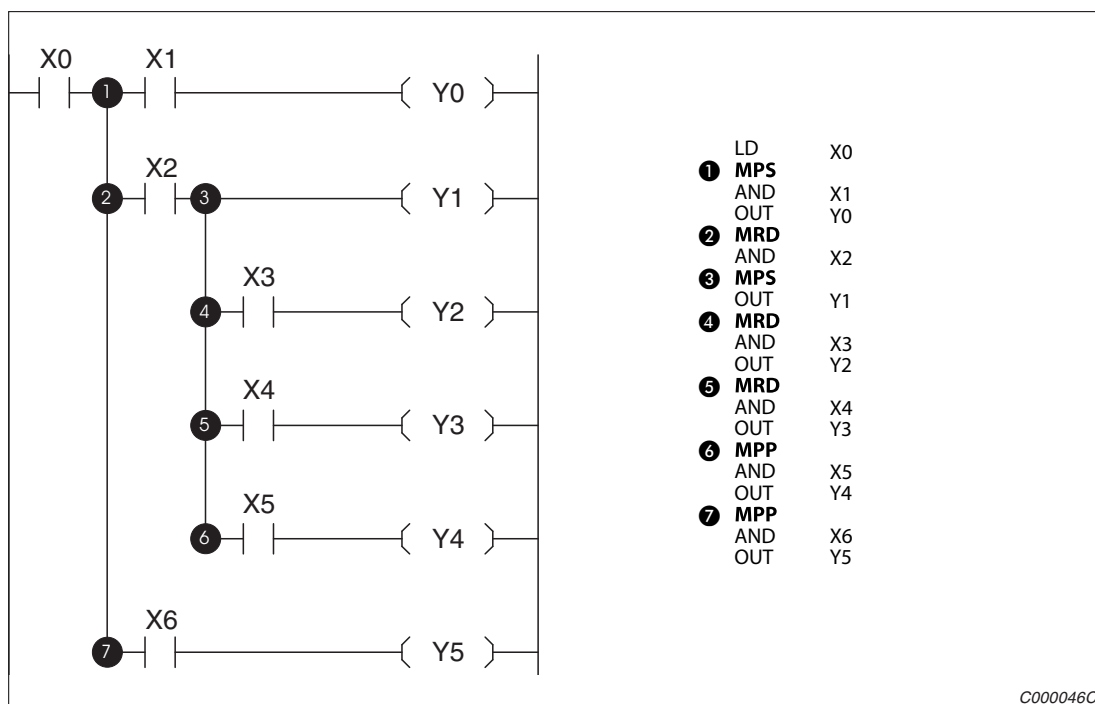
УКАЗАНИЕ

Максимум допускается 11 уровней логических связей (параллельно присоединенных катушек к одному логическому выражению).

Детальное описание трех инструкций представляется на основании следующих примеров программирования.

ПРИМЕР ▾

Использование инструкций MPS, MRD, MPP



C000046C

Рис.4-12. Пример программирования инструкций MPS, MRD, MPP

- 1 **MPS**
Промежуточный результат (здесь X0) на 1-ом уровне логических связей занесен на 1-ое место в стековую память промежуточных связей.
- 2 **MRD**
Перед выполнением следующей инструкции опрашивается промежуточный результат на 1-ом месте памяти логических связей
- 3 **MPS**
Промежуточный результат на 2-ом месте уровня логических связей записывается на 1-ое место в память логических связей. Уже имеющееся значение на 1-ом месте смещается на 2-ое место.
- 4 **MRD**
Перед выполнением следующей инструкции опрашивается промежуточный результат на 1-ом месте памяти логических связей
- 5 **MRD**
Перед выполнением следующей инструкции опрашивается промежуточный результат на 1-ом месте памяти логических связей
- 6 **MPP**
Перед выполнением следующей инструкции опрашивается промежуточный результат на 1-ом месте памяти логических связей. Операция на 2-ом уровне промежуточных результатов завершена. Значение на 1-ом месте в памяти логических связей стирается. Значение 2-ого места смещается назад на 1-е место.
- 7 **MPP**
Перед выполнением следующей инструкции опрашивается промежуточный результат на 1-ом месте памяти логических связей. Операция на 1-ом уровне промежуточных результатов завершена, и память логических связей стирается.

△

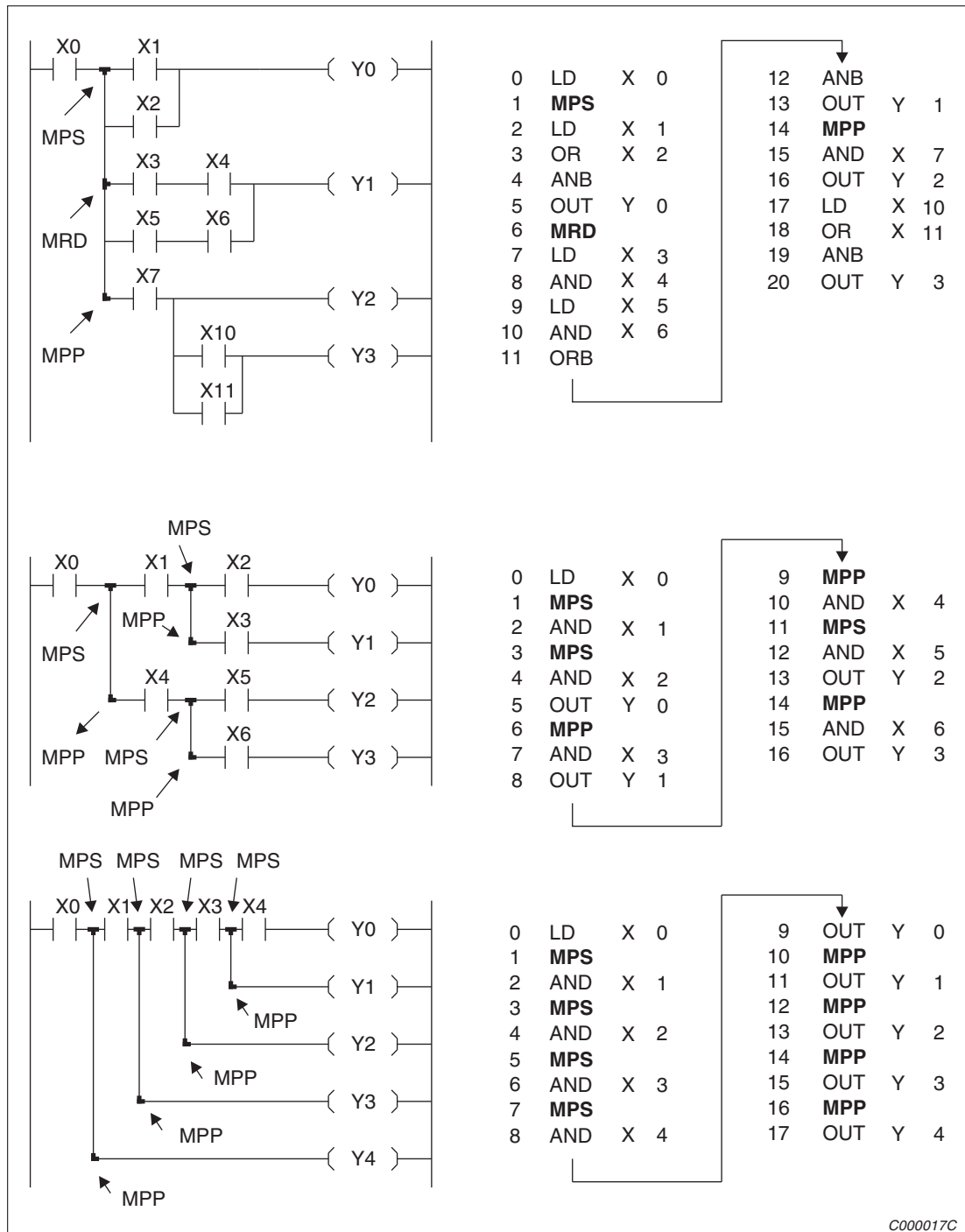


Рис.4-13. Примеры программирования инструкций MPS, MRD и MPP

C000017C

4.12 ВКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ УСЛОВИЙ КОНТРОЛЯ (MC, MCR)

	MC				
	Верхний уровень управления; Включение (запуск) условий контроля				
	MCR				
	Верхний уровень управления сбросом; Отключение (сброс) условий контроля				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
MC: Y, M, нет специального меркера	MC-инструкция			3	
MCR: N	MCR-инструкция			2	

ФУНКЦИЯ

Благодаря включению (MC) или отключению (MCR) условий контроля можно активизировать или деактивизировать отдельные области программ. Функция работает как главный контакт левой сборной (питающей) шины (программирование в контактной схеме).

ОПИСАНИЕ

- С помощью MC-инструкции включаются условия контроля для активизации определенной области программы.
 - какая область программы должна активизироваться, определяется указанием адреса программирования разветвления n: определяется от N0 до N7 (адрес разветвления).
 - задание операнда Y или M определяет контакт включения. Этот контакт активизирует область программы n, как только выполняться входные условия для MC-инструкции.
- После программирования MC-инструкции должны всегда программироваться LD или LDI инструкции (см.раздел 4.2).
- MCR-инструкция отключает MC-контакт и представляет тем самым конец области программирования n.
- Если входные условия не выполняются состояния операндов изменяются между MC и MCR как показано ниже:
 - для счетчика с жесткой памятью и операндов, которые программируются по SET и RST инструкциям, состояние сохраняется (см.раздел 4.10)
 - все таймеры и операнды, которые программировались по OUT инструкции, отключаются
- Внутри программы ПК могут быть созданы до 8 уровней разветвления. Уровень разветвления характеризуется параметром “n”.
- То, на что нужно обращать внимание при использовании нескольких MC и MCR инструкций внутри одной программы, пояснено на следующих примерах.

УКАЗАНИЕ

Благодаря применению MC и MCR инструкций не сокращается время цикла программы.

ПРИМЕР ▾

Применение инструкций МС и МСR

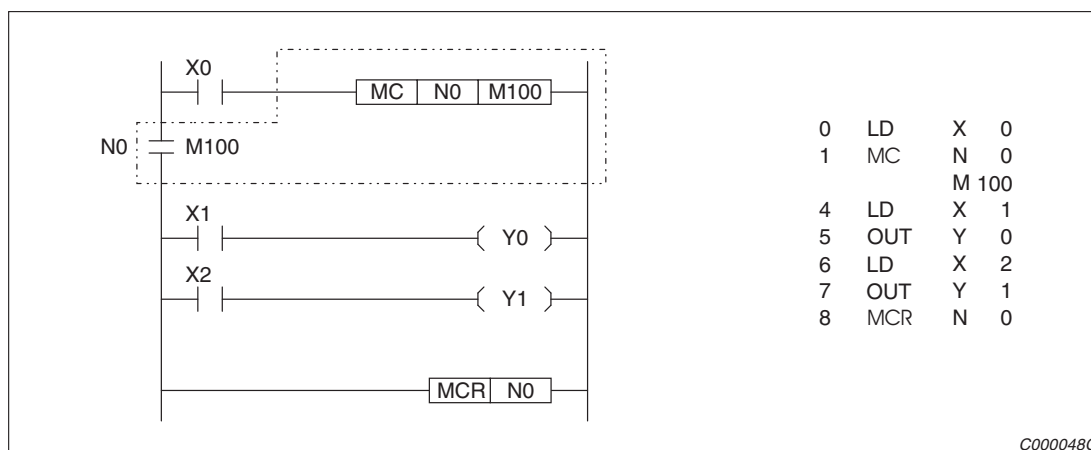


Рис.4-14. Пример программирования инструкций МС и МСR

Дублирование операндов внутри или вне Главной контрольной области (Master-Control-Область) также при неактивной Главной контрольной области приводит к проблемам, описанным в разделе 4.3 (дублирование записи выходов).

Как только выполняются входные условия для МС инструкции, включается контакт меркера М100 (адрес программы разветвления N0). Все цепи между инструкциями МС и МСR активизируются. Состояние сигнала Y0 или Y1 зависит тогда только от сигналов состояния X1 и X2. △

Использование нескольких mc и mcg инструкций внутри одной программы

При программировании нескольких MC и MCR инструкций внутри программы нужно обращать внимание на следующее:

- чтобы **1-АЯ MC**-инструкция начиналась с самого **низкого** адреса N разветвления программы
- чтобы **1-АЯ MCR**-инструкция начиналась с самого **высокого** адреса N разветвления программы.

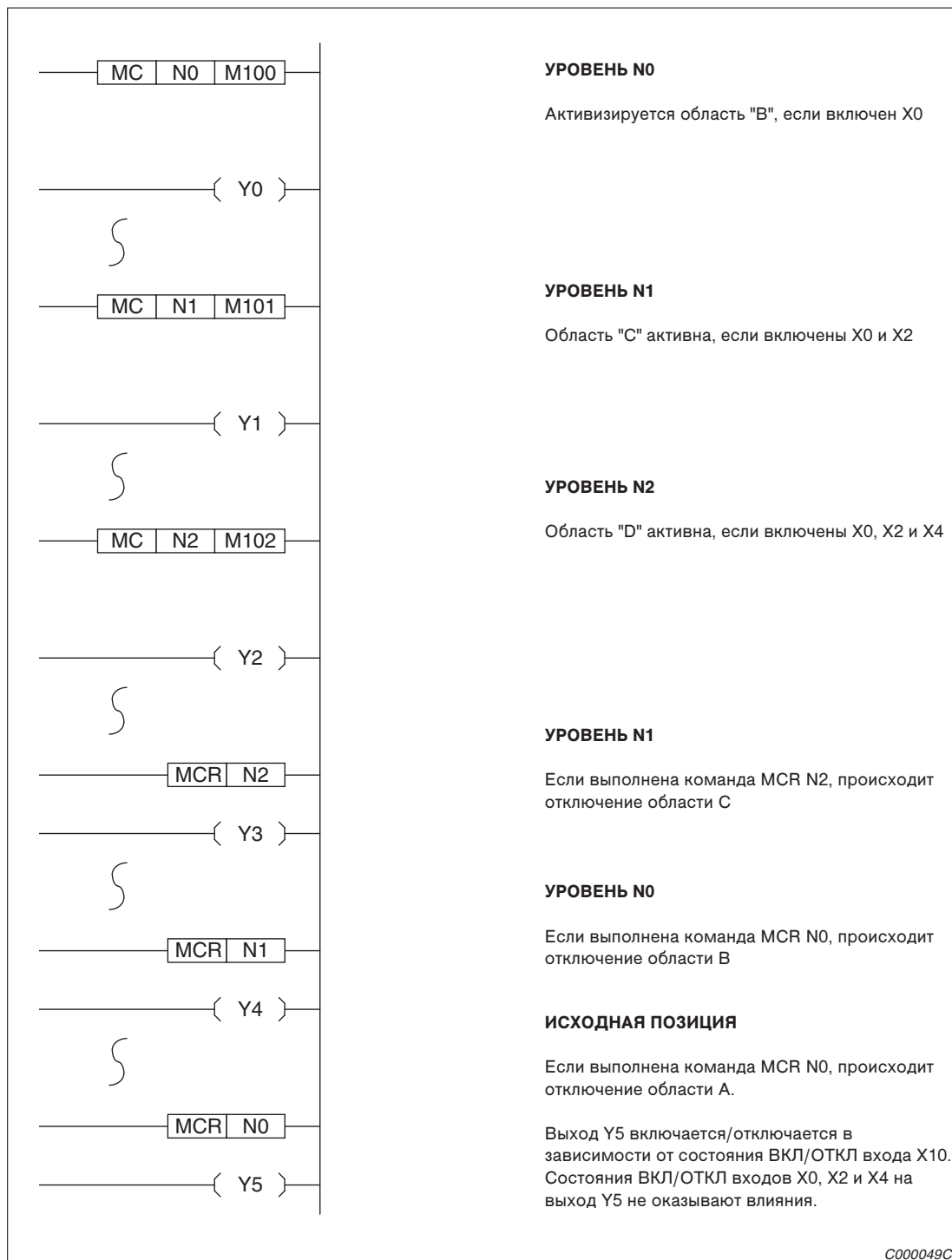
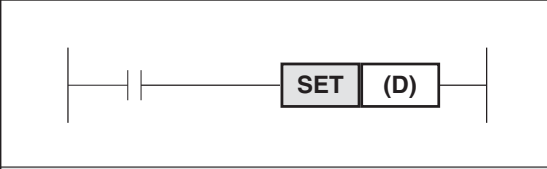


Рис.4-15. Пример программирования нескольких MC и MCR инструкций внутри программы

4.13 ВКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ ОПЕРАНДОВ (SET, RST)

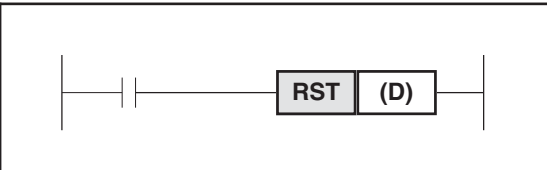
	SET				
	Включение; Включение операнда				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
Y, M, S	Y, M	1			
	S, специальный меркер	2			

ФУНКЦИЯ

Состояние сигнала операнда с помощью SET инструкции (включение) может устанавливаться непосредственно.

ОПИСАНИЕ

- С помощью SET могут устанавливаться в "1" (включаться) операнды Y, M или S.
 - как только входное условие установится для SET инструкции (сигнал "1"), включится соответствующий операнд.
 - в том случае, если входные условия для SET инструкции больше не будут выполняться, соответствующий операнд останется включенным.

	RST				
	Отключение; Отключение операнда				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды	Шаги программы			Примечание	
Y, M, S, D, V, Z, T, C	Y, M	1	S, T, C	2	
	D, V, Z, специальный регистр	3			

ФУНКЦИЯ

Состояние сигнала операнда с помощью RST инструкции (сброс) может устанавливаться непосредственно.

- С помощью RST-инструкции могут отключаться соответствующие операнды. Это означает:
 - выходы Y, меркеры M и операнды состояния шагов S выключаются (состояние сигнала "0").
 - действительное значение таймера и счетчика, а также содержание регистров D, V и Z сбрасываются на "0".
 - как только входное условие установится для RST инструкции (сигнал "1"), выключится соответствующий операнд.
 - в том случае, если входные условия для RST инструкции больше не будут выполняться, соответствующий операнд останется выключенным.

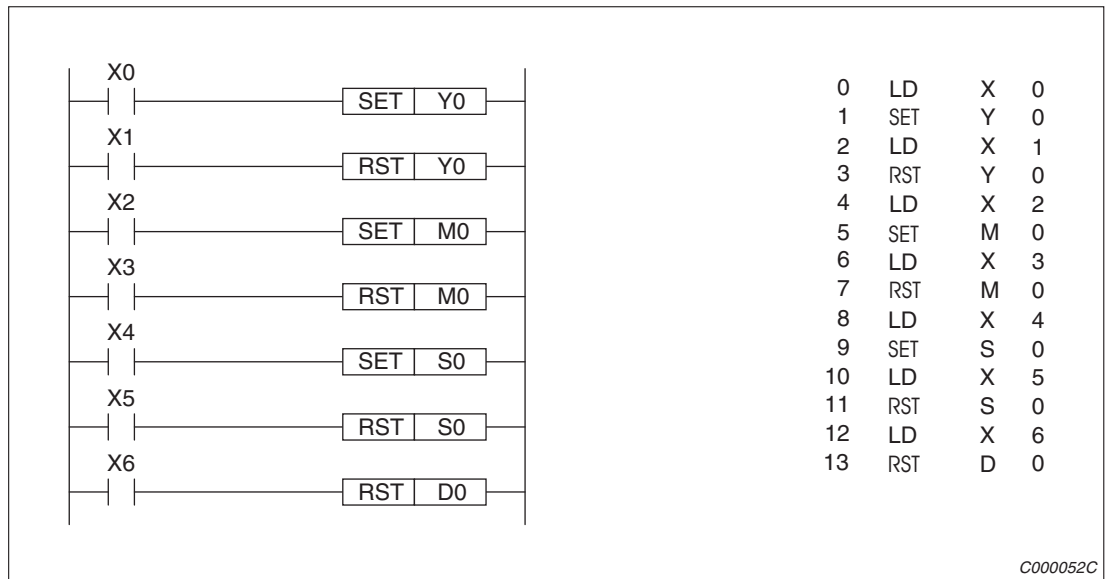


Рис.4-16. Пример программирования инструкций SET и RST

ПРИМЕР ▾

Сброс (отключение) 16-ти битного счетчика с помощью RST инструкции

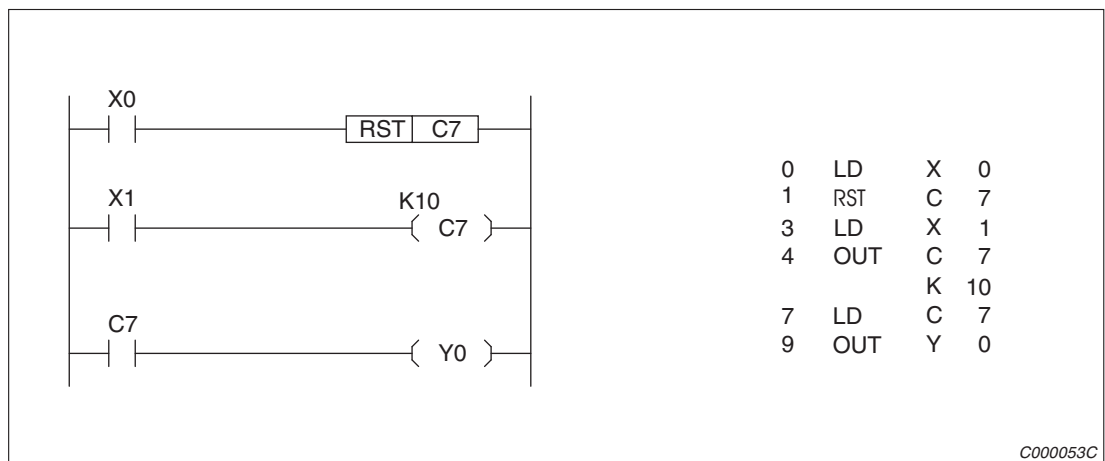
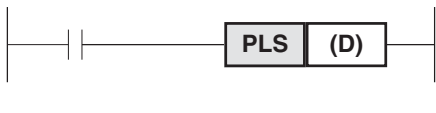



Рис.4-17. Пример программирования отключения 16-ти битного счетчика с помощью RST инструкции

Выходной контакт Y0 активизирован, если достигнуто настроенное значение K10. Как только вход X0 включится, выход Y0 отключится и действительное (накопленное) значение счетчика C7 установится на "0". △

4.14 ГЕНЕРАЦИЯ ИМПУЛЬСА (PLS, PLF)

	PLS				
	Генерация импульсов; Формирование одноразового импульса при возрастающем фронте				
	PLF				
	Генерация импульсов; Формирование одноразового импульса при падающем фронте				
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
	●	●	●	●	●
Операнды					
Y, M	Шаги программы		Примечание		
	PLS-инструкция		2		
PLF-инструкция		2			

ФУНКЦИЯ

Генерация одного импульса - опознание фронта сигнала - независимо от продолжительности входного сигнала для включения соответствующего операнда. Операнд остается во включенном состоянии на протяжении одного цикла программы (скана).

ОПИСАНИЕ

- PLS и PLF инструкции могут использоваться совместно с меркером M и цифровым выходом Y. Инструкции генерируют одинаковые импульсы независимо от продолжительности входного сигнала.
- После исполнения PLS или PLF сигнал соответствующего операнда (Y или M) удерживается в "1" на протяжении одного скана.
- PLS-инструкция генерирует одиночный импульс по возрастающему фронту входного сигнала.
- PLF-инструкция генерирует одиночный импульс по падающему фронту входного сигнала.

УКАЗАНИЕ

Специальные меркеры не должны активизироваться с PLS и PLF инструкциями.

ПРИМЕР ▾

Применение однократного импульса PLS, PLF

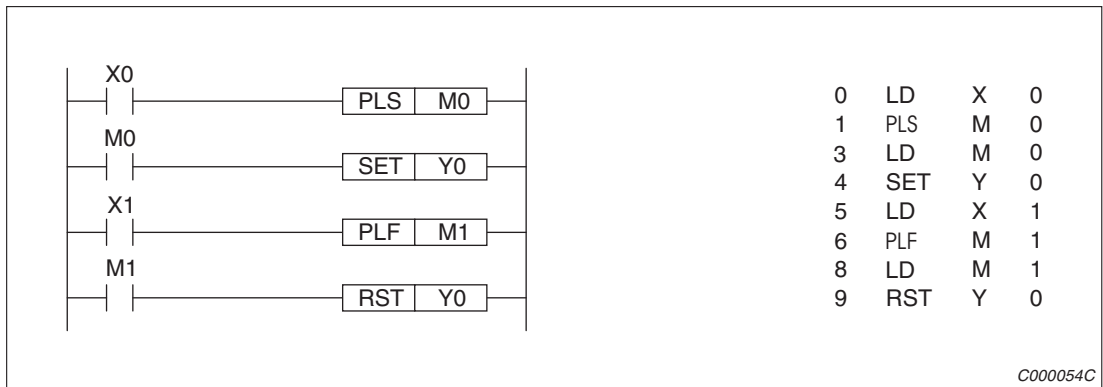


Рис.4-18. Пример программирования инструкций PLS и PLF

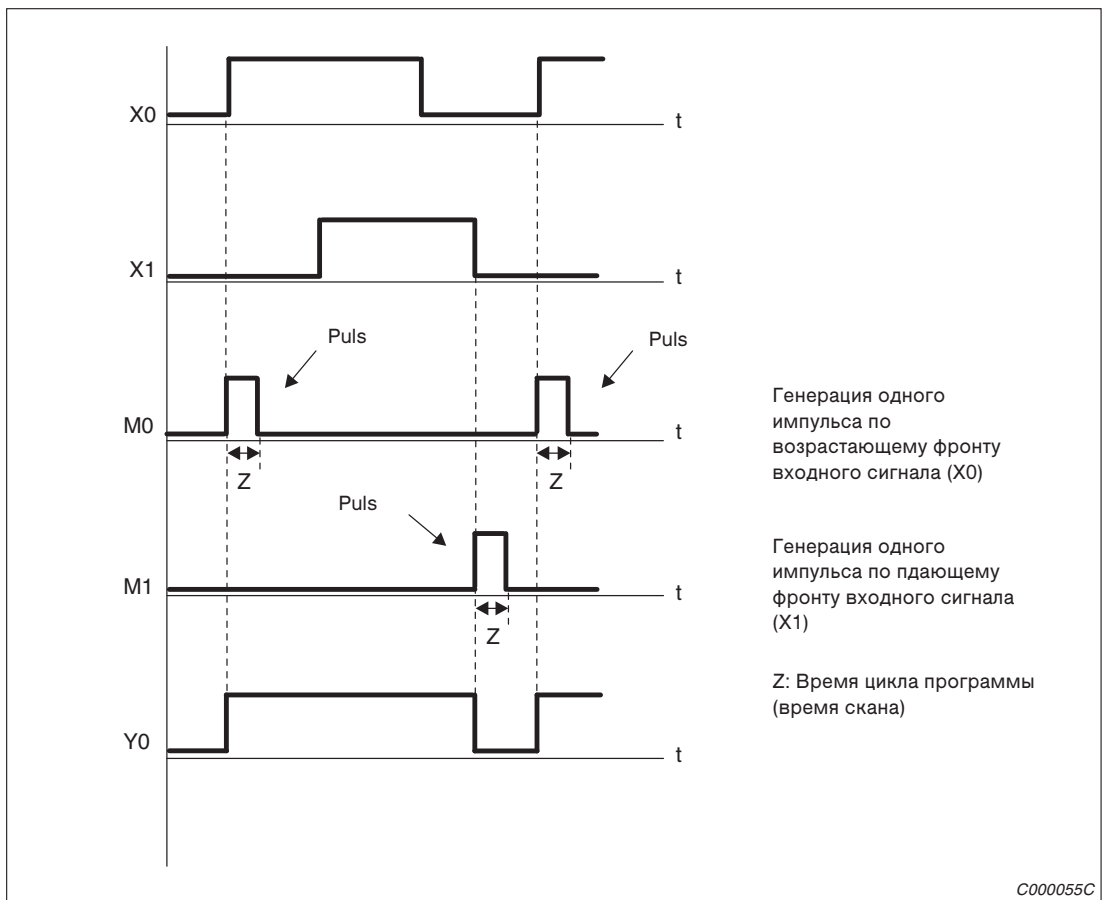


Рис.4-19. Пример программирования, изображение обработки входного сигнала и генерации импульса

При возрастании входного сигнала на входе X0 с "0" до "1" (возрастающий фронт) меркер M0 благодаря PLS-инструкции получает импульс (включается на время одного скана). С помощью этого импульса по контакту меркера M0 включается выход Y0. Лишь только, когда на входе X1 произойдет смена сигнала с "1" на "0" (падающий фронт), выход Y0 снова отключится. △

4.15 ИНВЕРСИЯ РЕЗУЛЬТАТА ОБРАБОТКИ (INV)

		INV				
		Инверсия; Инверсия результата обработки				
		CPU	FX1S	FX1N	FX2N	FX2NC
		●	●	●	●	●
Операнды		Шаги программы			Примечание	
—		INV-инструкция			1	

ФУНКЦИЯ

Реверсирование состояния сигнала результата обработки

ОПИСАНИЕ

INV-инструкция инвертирует состояние сигнала результата стоящей впереди инструкции.

- Полученная согласно обработки "1", после инверсии становится "0".
- Полученный согласно обработки "0", после инверсии становится "1".
- INV-инструкция может применяться как AND и ANI инструкции.

УКАЗАНИЕ

INV-инструкция может применяться для реверсирования сигнала результата комплексной схемы.

INV-инструкция может применяться для реверса сигнала результата импульсных инструкций LDP, LDF, ANP и т. д.

ПРИМЕР ▾

Применение INV-инструкции

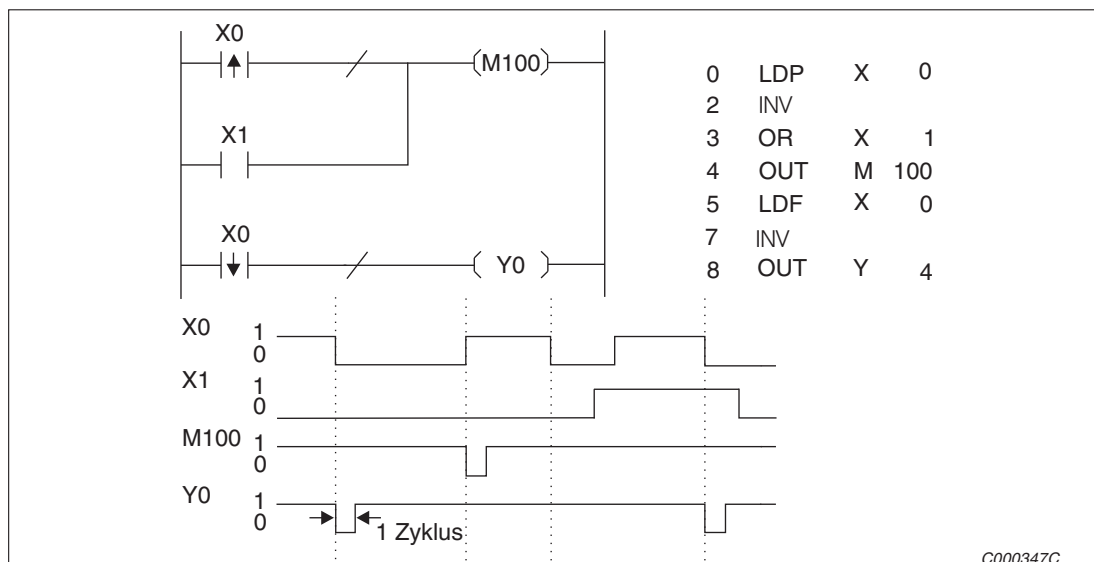


Рис.4-20. Пример программирования инструкции INV

Меркер M100 отключается положительным фронтом X0.

Выход Y0 отключается падающим фронтом X0.



4.16 ПУСТАЯ СТРОКА В ПРОГРАММЕ (NOP)

	NOP				
	Пустая строка; Пустая строка без функций				
	CPU	FX1S	FX1N	FX2N	FX2NC
	●	●	●	●	●
Операнды	Шаги программы				Примечание
—	NOP-инструкция			1	

ФУНКЦИЯ

Можно создать пустую строку без логических функций, которая позднее может быть использована для каких-либо инструкций, например, при окончательном изготовлении программы, при отладке оборудования

ОПИСАНИЕ

- После успешного завершения программы NOP-команды должны быть удалены, так как в противном случае они бесполезно удлиняют время цикла программы.
- Количество NOP-команд не ограничено.
- При стирании всей программы все инструкции переписываются на NOP-инструкции.
- Дополнительный ввод NOP-инструкций с помощью ручного программатора может выполняться функцией ВСТАВИТЬ (INSERT).

УКАЗАНИЕ

Если инструкции LD, LDI, ANB или ORB заменяются NOP-инструкцией, можно значительно изменить логическую конструкцию схемы. Это может привести к ошибочному исполнению программы.

ПРИМЕР ▾

Применение NOP-инструкции

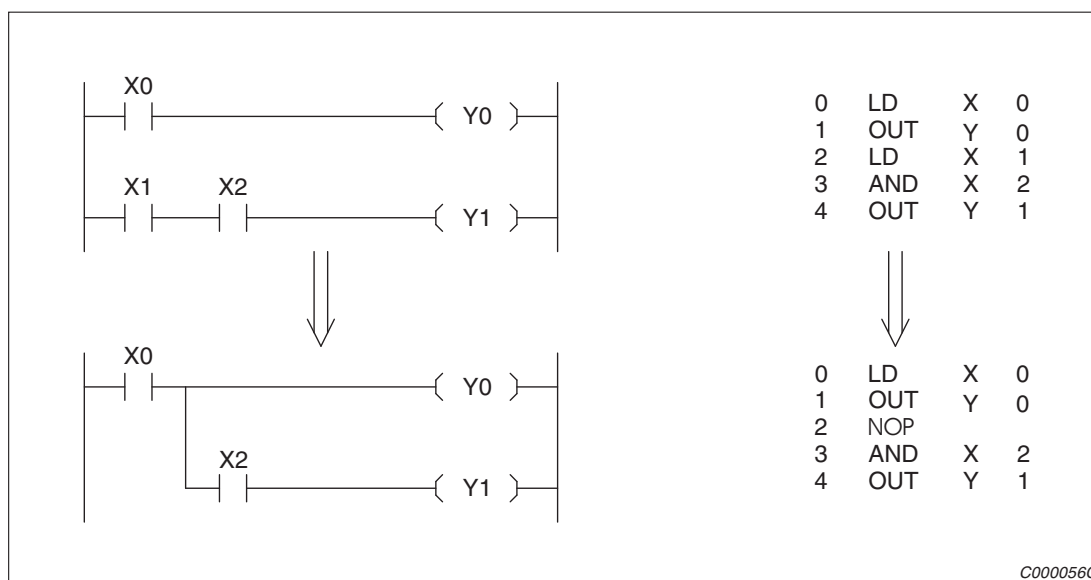


Рис. 4-21: Пример программирования инструкции NOP

Вход X1 заменяется NOP-инструкцией. Из-за этого изменяется логическая конструкция схемы ▽

4.17 КОНЕЦ ПРОГРАММЫ ПК (END)

	END					
	КОНЕЦ; Конец программы ПК					
	CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●
Операнды	Шаги программы			Примечание		
—	END-инструкция			1		

ФУНКЦИЯ

Окончание программы ПК и переход к началу программы (шаг 0)

ОПИСАНИЕ

- Каждая программа ПК должна завершаться инструкцией END.
- Если программируется END-инструкция, то на этом месте оканчивается обработка программы. Последующие области программы не принимаются больше во внимание. После отработки END-инструкции выполняется обработка выходов. Для этого обработка программы перескакивает к началу программы (шаг 0).
- Чтобы организовать отдельные участки программы для пошаговой проверки программы, можно вводить END-инструкцию также внутри программы. Инструкции после END-инструкции при проверке не учитываются. Эта дополнительная END-инструкция должна после окончания проверки удаляться.

УКАЗАНИЕ

После выполнения END-инструкции имеет место обновление контрольного времени цикла (Watch-Dog-Timers) и регистра отображения (Image Register).

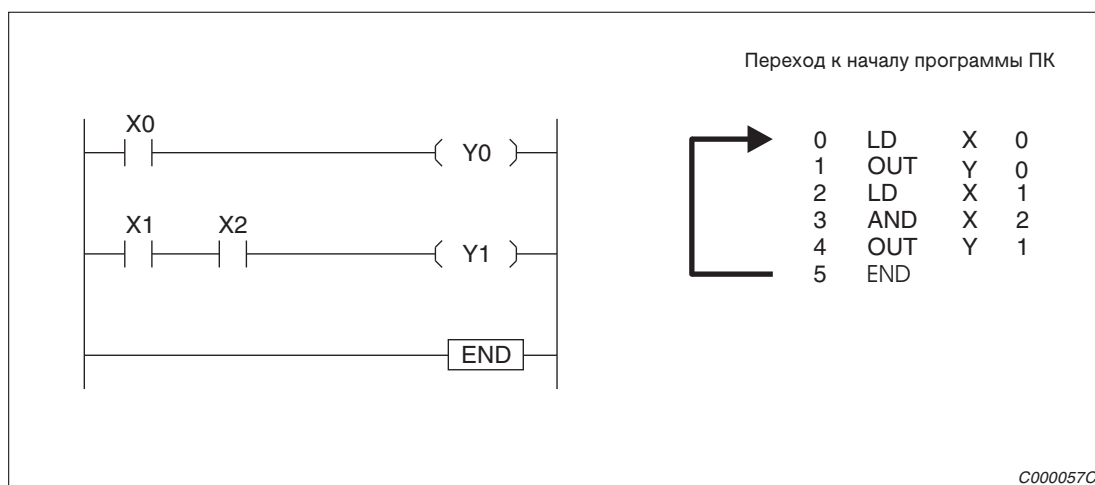


Рис.4-22. Пример программирования END-инструкции

4.18 ПРИМЕРЫ ПРОГРАММИРОВАНИЯ

Следующий раздел показывает несколько простых примеров использования набора базовых команд. Примеры могут непосредственно программироваться и исполняться и служить для углубления знаний, приобретенных в первой части этого руководства.

- Опрос входов (размыкающих и замыкающих).
- Последовательное включение входов.
- Параллельное включение входов.
- Самопитание выхода.
- Замедление включения.
- Замедление отключения.
- Счет вверх.

4.18.1 ОПРОС ВХОДОВ

ЗАДЕЙСТВОВАННЫЙ ЗАМЫКАЮЩИЙ КОНТАКТ

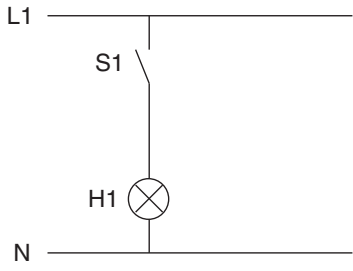
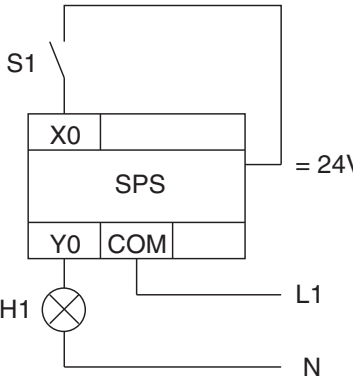
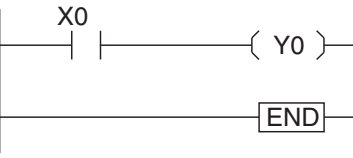
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При замыкании замыкающего контакта S1 должна засветиться лампа H1.</p>	 <p style="text-align: right;"><i>C000007G</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Сигнальная лампа: H1 Y0</p>	 <p style="text-align: right;"><i>C000008G</i></p>
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (КОР)
<pre> 0 LD X0 1 OUT Y0 2 END </pre>	 <p style="text-align: right;"><i>C000009G</i></p>
ПРИМЕЧАНИЕ	
<p>Выход Y0 выполняет сигнал "1", если на входе X0 есть сигнал "1".</p>	

Табл.4-4. Пример для задействованного замыкающего контакта



УКАЗАНИЕ

Для включения используйте замыкающий контакт, чтобы в случае поломки не было самопроизвольного включения

ЗАДЕЙСТВОВАННЫЙ РАЗМЫКАЮЩИЙ КОНТАКТ

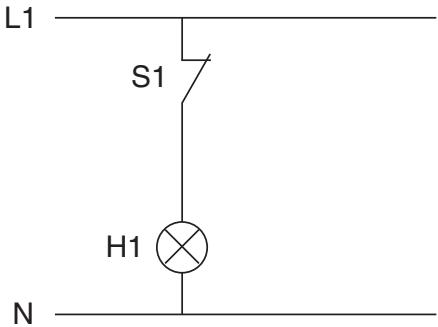
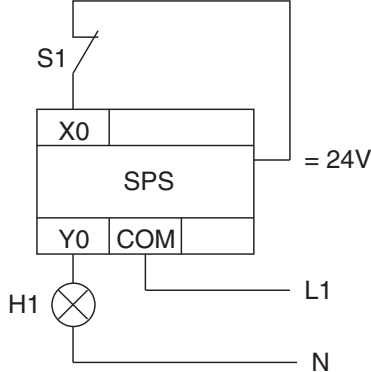
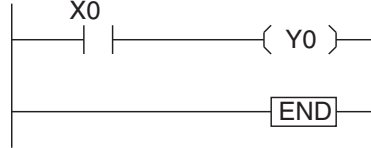
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При задействованном размыкающем контакте S1 должна светиться лампа H1.</p>	 <p style="text-align: right;"><i>C000010G</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Размыкающий контакт: S1 X0</p> <p>Сигнальная лампа: H1 Y0</p>	 <p style="text-align: right;"><i>C000011G</i></p>
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (КОР)
<pre> 0 LD X0 1 OUT Y0 2 END </pre>	 <p style="text-align: right;"><i>C000012G</i></p>
ПРИМЕЧАНИЕ	
<p>Выход Y0 выполняет сигнал "1", если на входе X0 есть сигнал "1". Размыкающий контакт S1 опрашивается в программе на состояние сигнала "1", чтобы выход Y0 при задействовании (срабатывании) размыкающего контакта S1 отразил сигнал "0".</p>	

Табл. 4-5. Пример задействованного размыкающего контакта



УКАЗАНИЕ

Для выключения используйте размыкающий контакт, чтобы при поломке было гарантированное выключение

ПОСЛЕДОВАТЕЛЬНОЕ СОЕДИНЕНИЕ

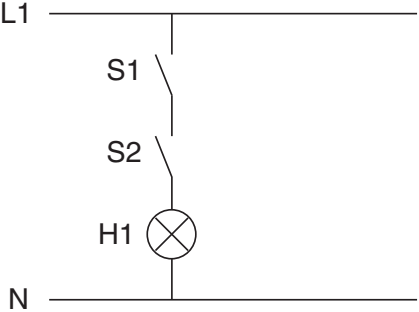
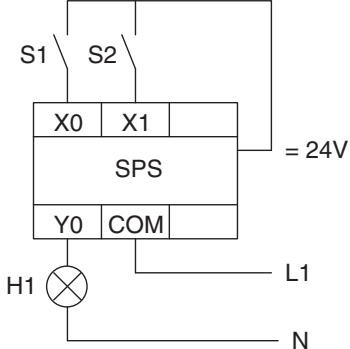
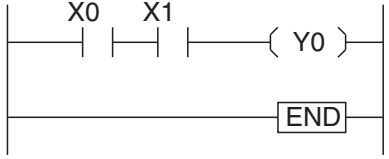
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При задействованных замыкающем контакте S1 "И" замыкающем контакте S2 должна светиться лампа H1.</p>	 <p style="text-align: right;"><i>C00001</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Замыкающий контакт: S2 X1 Сигнальная лампа: H1 Y0</p>	 <p style="text-align: right;"><i>C000017G</i></p>
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (КОР)
<pre> 0 LD X0 1 AND X1 2 OUT Y0 3 END </pre>	 <p style="text-align: right;"><i>C000018G</i></p>
ПРИМЕЧАНИЕ	
<p>Выход Y0 выполняет сигнал "1", если на входе X0 и входе X1 есть сигнал "1".</p>	

Табл.4-6. Пример логической связи "И"

ПАРАЛЛЕЛЬНОЕ СОЕДИНЕНИЕ

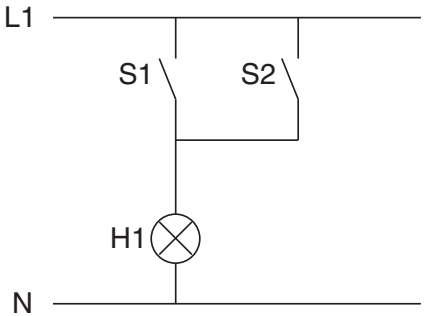
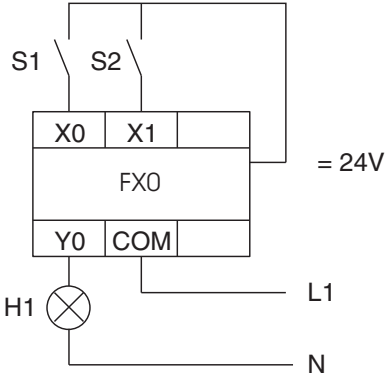
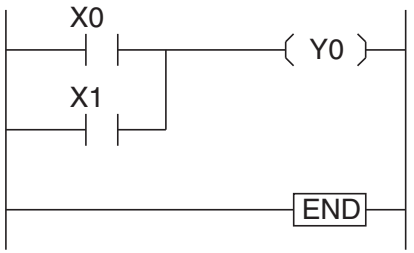
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При задействованном замыкающем контакте S1 "ИЛИ" замыкающем контакте S2 должна светиться лампа Н1.</p>	 <p style="text-align: right;"><i>C000019G</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Замыкающий контакт: S2 X1 Сигнальная лампа: Н1 Y0</p>	 <p style="text-align: right;"><i>C000020G</i></p>
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (КОР)
<pre> 0 LD X0 1 OR X1 2 OUT Y0 3 END </pre>	 <p style="text-align: right;"><i>C000021G</i></p>
ПРИМЕЧАНИЕ	
<p>Выход Y0 выполняет сигнал "1", если на входе X0 или ходе X1 есть сигнал "1".</p>	

Табл. 4-7: Пример логической связи "И"

САМОПИТАНИЕ (I)

Включение выхода с самопитанием

<p>ПРИМЕР</p> <p>При включении замыкающего контакта S1 должна загореться лампа H1, даже если замыкающий контакт S1 больше не включен. После кратковременного воздействия на размыкающий контакт S2, лампа H1 перестает гореть.</p>	<p>ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА</p> <p style="text-align: right;">C000033G</p>
<p>ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК</p> <p>Замыкающий контакт: S1 X0</p> <p>Размыкающий контакт: S2 X1</p> <p>Сигнальная лампа: H1 Y0</p>	<p>СХЕМА ПОДКЛЮЧЕНИЯ К ПК</p> <p style="text-align: right;">C000034G</p>
<p>ЛИСТИНГ ИНСТРУКЦИЙ (AWL)</p> <pre> 0 LD X0 1 OR Y0 2 AND X1 3 OUT Y0 4 END </pre>	<p>КОНТАКТНАЯ СХЕМА (КОР)</p> <p style="text-align: right;">C000036G</p>
<p>ПРИМЕЧАНИЕ</p> <p>Выход Y0 включается (состояние сигнала "1"), если вход X0 кратковременно замкнется (активизируется замыкающий контакт S1). Выход Y0 отключится (состояние сигнала "0"), если кратковременно задействуется вход X1 (активизируется размыкающий контакт S2).</p>	

Табл.4-8. Пример включения и отключения выхода с самопитанием

САМОПИТАНИЕ (II)

Включение и отключение выхода с помощью SET-/RST-инструкций

<p>ПРИМЕР</p> <p>При включении замыкающего контакта S1 должна загореться лампа H1, даже если замыкающий контакт S1 больше не включен. После кратковременного воздействия на размыкающий контакт S2, лампа H1 перестает гореть.</p>	<p>ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА</p> <p style="text-align: right;">C000033G</p>
<p>ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК</p> <p>Замыкающий контакт: S1 X0</p> <p>Размыкающий контакт: S2 X1</p> <p>Сигнальная лампа: H1 Y0</p>	<p>СХЕМА ПОДКЛЮЧЕНИЯ К ПК</p> <p style="text-align: right;">C000034G</p>
<p>ЛИСТИНГ ИНСТРУКЦИЙ (AWL)</p> <pre> 0 LD X0 1 SET Y0 2 LDI X1 3 RST Y0 4 END </pre>	<p>КОНТАКТНАЯ СХЕМА (КОР)</p> <p style="text-align: right;">C000035G</p>
<p>ПРИМЕЧАНИЕ</p> <p>Выход Y0 включается (состояние сигнала "1"), если вход X0 кратковременно замкнется (активизируется замыкающий контакт S1). Выход Y0 отключится (состояние сигнала "0"), если кратковременно задействуется вход X1 (активизируется размыкающий контакт S2).</p>	

Табл.4-9. Пример включения (SET) и отключения (RST) выхода

ПРИМЕНЕНИЕ ТАЙМЕРА ДЛЯ ЗАДЕРЖКИ ВКЛЮЧЕНИЯ

ПРИМЕР	ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК
<p>При задействованном замыкающем контакте S1 лампа H1 не должна загораться до отсчета 5 с.</p>	<p>Замыкающий контакт: S1 X0 Сигнальная лампа: H1 Y0 Таймер : T0 100 мс</p>
ВРЕМЕННАЯ ДИАГРАММА	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (НОР)
<pre> 0 LD X0 1 OUT T0 K50 4 LD T0 5 OUT Y0 6 END </pre>	
ПРИМЕЧАНИЕ	
<p>Если вход X0 примет состояние сигнала "1", начнется отсчет времени. После отсчета запрограммированного времени $t = 5$ с выход Y0 примет состояние сигнала "1". Таймер T0 вернется в отключенное состояние "0", как только вход X0 примет состояние сигнала "0" (по окончании 5 с при включенном X0 таймер остается включенным, но отсчет времени на нем прекращается).</p>	

Табл.4-10. Пример применения таймера для замедления включения

ПРИМЕНЕНИЕ ТАЙМЕРА ДЛЯ ЗАДЕРЖКИ ОТКЛЮЧЕНИЯ

ПРИМЕР	ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК
<p>При задействованном замыкающем контакте S1 лампа Н1 сейчас же загорается. Однако лампа Н1 должна светить на 5 с больше, чем отключится замыкающий контакт S1.</p>	<p>Замыкающий контакт: S1 X0 Сигнальная лампа: Н1 Y0 Таймер : T0 100 мс</p>
ВРЕМЕННАЯ ДИАГРАММА	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (КОР)
<pre> 0 LD X0 1 OR Y0 2 ANI T0 3 OUT Y0 4 LD Y0 5 ANI X0 5 OUT T0 K50 9 END </pre>	
ПРИМЕЧАНИЕ	
<p>Если вход X0 примет состояние сигнала "1", включится и станет на самопитание выход Y0. Как только вход X0 снова примет состояние "0" начнется отсчет запрограммированного времени. После отсчета настроенного времени t = 5 с выход Y0 отключится (примет состояние сигнала "0").</p>	

Табл.4-11. Пример применения таймера для замедления отключения

ПРИМЕНЕНИЕ СЧЕТЧИКА СО СЧЕТОМ ВВЕРХ

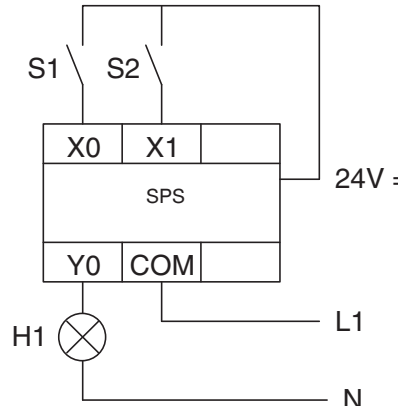
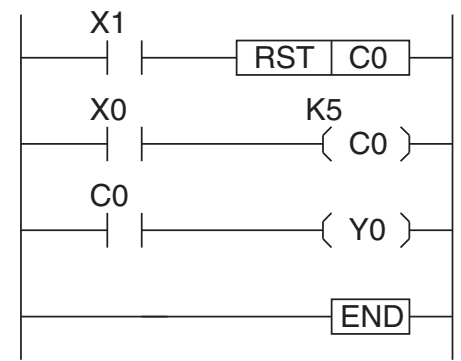
ПРИМЕР	-
<p>После 5-ти срабатываний замыкающего контакта S1 должна включиться лампа H1. Контактom S2 счетчик C0 должен снова вернуться в исходное состояние (сбросить накопленный счет) и погасить лампу H1.</p>	-
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Замыкающий контакт: S2 X1 Сигнальная лампа: H1 Y0 Счетчик: C0</p>	
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (КОР)
<pre> 0 LD X1 1 RST C0 3 LD X0 4 OUT C0 K5 7 LD C0 8 OUT Y0 9 END </pre>	
ПРИМЕЧАНИЕ	
<p>При каждом включении счетчика C0 в состояние "1" действительное накопленное в нем значение повышается на 1. После того как действительное значение счетчика достигнет установленного значения, счетчик C0 включит выход Y0 в состояние "1". Сигнал "1" на входе X1 вернет счетчик в состояние "0".</p>	

Табл.4-12. Пример применения счета вверх

5 STL-ИНСТРУКЦИИ

5.1 ОБЩИЕ УКАЗАНИЯ

STL-инструкция является элементарной инструкцией ПК для единого программирования процесса управления. STL-инструкция применяется совместно со статусом шага и обеспечивает комфортное программирование шагового управления.

В случае простых пуско-/стоповых-последовательностях работы, даже начинающий специалист может эффективно использовать систему управления и тем самым значительно ограничить затраты на программирование для таких последовательностей.

STL-инструкция программируется совместно с операндами шаговых состояний S. В зависимости от применяемого типа центрального процессора CPU имеется до 1000 шаговых операндов в области от S0 до S999, при этом операндам S0...S9 присвоены жесткие функции.

ЗНАЧЕНИЯ ШАГОВЫХ ОПЕРАНДОВ S0...S9

Шаговые операнды S0...S9 являются инициализирующими операндами, с помощью которых могут выполняться различные шаговые процессы внутри STL-программы, чтобы, например, реализовать различные рабочие процессы (Наладочные и автоматические режимы работы, режим подхода к нулевой точке и т.д.). Это приобретает особое значение при использовании IST-инструкции.

Если не предусматривается никаких специальных процессов для ручного или автоматического режима работы и подхода к нулевой точке, то операнды S0...S9 могут использоваться как “нормальные” операнды без специальных функций.

5.1.1 ПРИМЕР ПРИМЕНЕНИЯ STL-ИНСТРУКЦИИ

Традиционный метод проектирования посредством контактной схемы состоит в том, что выход срабатывает от определенного, ему предписанного входного контакта (например, внешнего механического выключателя) и этот выходной контакт соответствует параллельному или последовательному действию относительно задания по управлению. Для обеспечения надежности работы относительно нежелательных управляющих процессов и зависимых от них ошибочных функций такие программы должны содержать многочисленные блокировочные мероприятия.

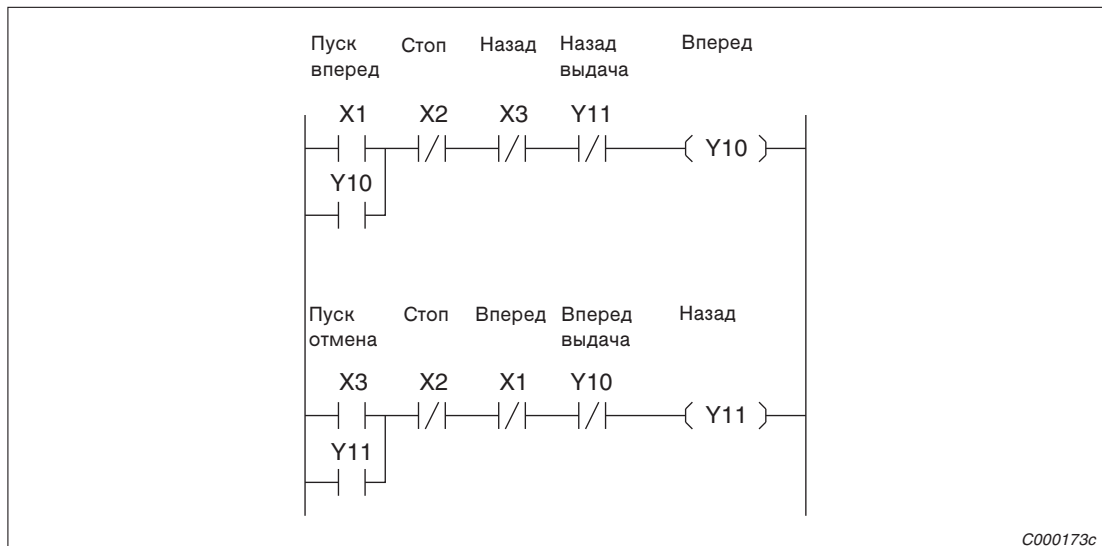


Рис.5-1. Пример применения блокировочных контактов

При применении шаговых управляющих инструкций показанные блокировочные контакты могут не потребоваться, так как сигналы управления, как например, “вверх”, “вниз” и т.д., выполняются по программе принимая во внимание определенные граничные значения.

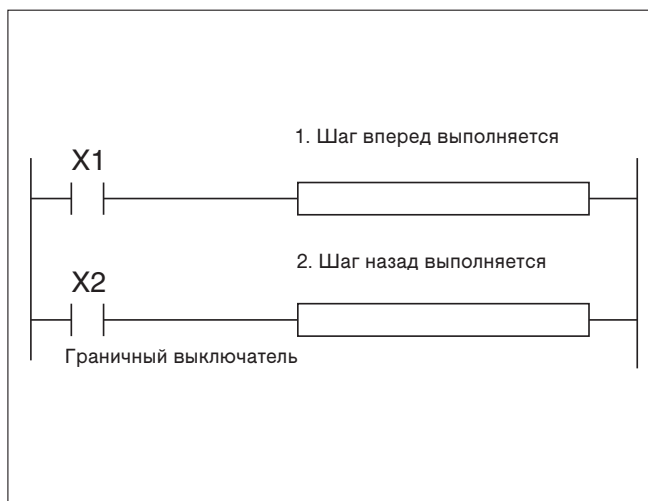


Рис.5-2. Пример схемы без блокировочных контактов

C000129C

5.1.2 СХЕМАТИЧНЫЙ ПРОЦЕСС ШАГОВОГО УПРАВЛЕНИЯ

На основании краткой последовательности процесса в качестве примера описано шаговое управление с четырьмя рабочими шагами. Четвертый шаг заканчивает шаговое управление.

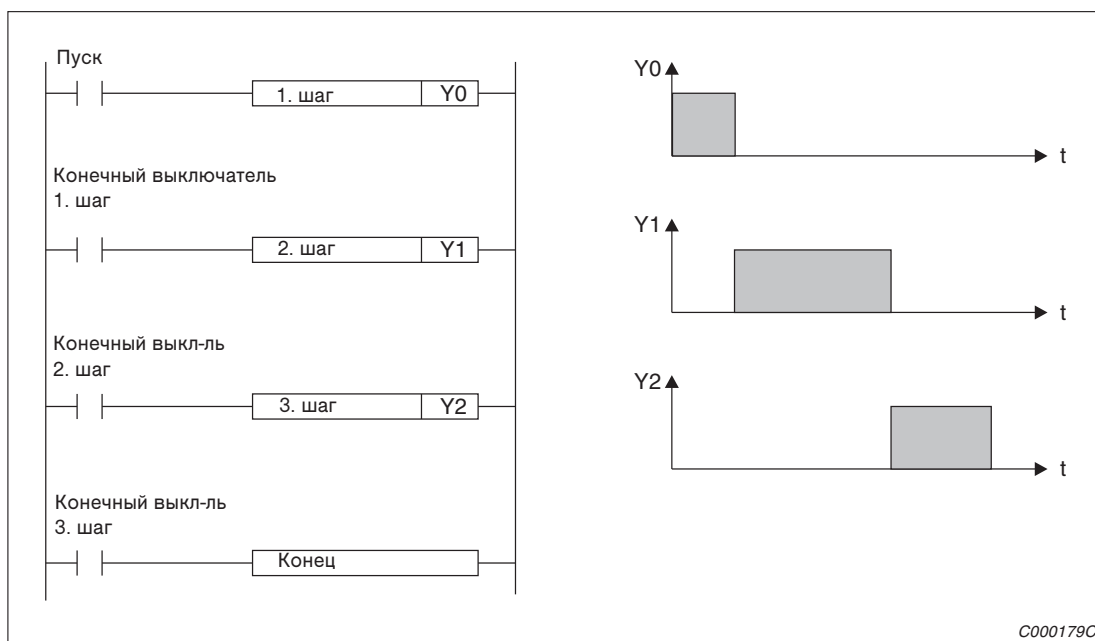


Рис.5-3. Схематичный процесс шагового управления

По рисунку 5-3 видно, что 2-ой рабочий шаг включается, как только заканчивается 1-ый шаг и включается соответствующий ему конечный выключатель. Это означает, что все состояния операндов внутри первого шага отключились.

Конец 2-го шага означает одновременно пуск 3-его шага. С помощью включения 3-его конечного выключателя достигается окончание шаговой последовательности (4-ый рабочий шаг)

5.1.3 ПРЕДСТАВЛЕНИЕ УПРАВЛЕНИЯ ПРОЦЕССОМ В ДИАГРАММЕ БЛОКОВ

На следующем рисунке представлена линейное управление процессом в диаграмме блоков (по IEC-стандарту). На диаграмме представлено управление процессом упрощенно независимо от позднейшей реализации в программе ПК.

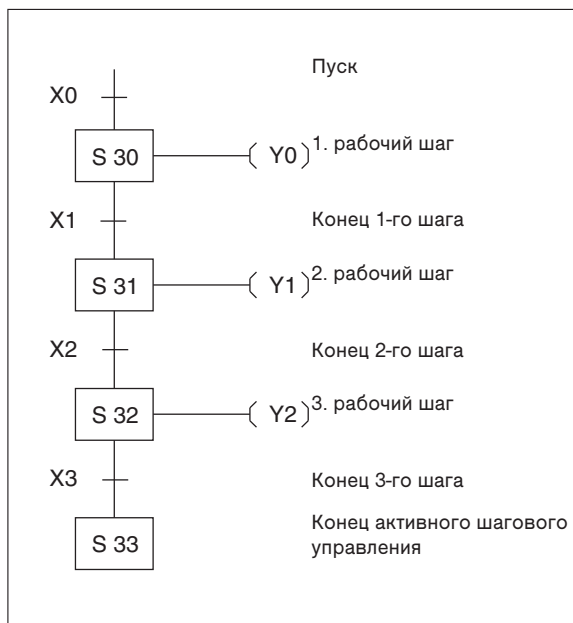


Рис.5-4.
Диаграмма блоков примера
пользователя

C000148C

5.2 ПРОГРАММИРОВАНИЕ STL-ИНСТРУКЦИИ

	STL		RET				
	Состояние шага активное/не активное						
CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
	●	●	●	●	●		
Операнд	Имп. инструкция (P)		Обработка		Шагов программы		
S0...S999; Область адресов зависит от примененного типа ПК MELSEC (см.табл.5-1)			16-бита	32-бита	STL	1	
					RET	1	

Тип ПК	Количество операндов	Область адресов
FX1S	128	S0 – S127
FX1N	1000	S0 – S999
FX2N		
FX2NC	4096	S0 – S4095
FX3U		

Табл.5-1. Область адресов

ФУНКЦИЯ

Программирование управления процессом

ОПИСАНИЕ

- STL-инструкция применяется совместно с операндом шагов S. Операнд шагов S может программироваться со следующими инструкциями набора базовых команд: LD, LDI, AND, ANI, OR, ORI, OUT, SET, RST.
- У ПК серии FX2N и FX2NC возможно применение импульсных инструкций (LDP, LDF и т.д.) и импульсных меркеров (M2800...M3071).
- В программе без шагового управления шаговые операнды S могут применяться также как обычные меркеры.
- В начале области программы (состояния шага) каждый отдельный операнд состояния шага должен применяться с SET-инструкцией.
- Внутри контактной схемы STL-контакт появляется на левой сборной (“питающей”) шине и может поэтому рассматриваться как “Главный контакт”.
- STL-инструкция следующей цепи может обрабатываться лишь тогда, когда применен STL-контакт.
- Как только STL-контакт отключается, следующая цепь не может больше обрабатываться.
- С помощью RET-инструкции заканчивается вся область STL-программы (состояния шагов).

УКАЗАНИЕ

Операнд состояния шага может программироваться только один раз в программе с помощью STL-инструкции.

STL-инструкция не может применяться в программе прерывания.

Не применяйте никаких инструкций переходов внутри шаговых состояний.

Каждый шаг управления должен заканчиваться RET-инструкцией

Активизированный последним операнд состояния шага должен отключаться с помощью RST-инструкции, или он должен выполнять дальнейшее включение (разрешение) обратно к началу шаговой цепи.

ДОПУСТИМЫЕ ИНСТРУКЦИИ ВНУТРИ СОСТОЯНИЙ ШАГОВ

В следующей таблице приведены инструкции набора базовых команд, которые могут применяться между STL-инструкциями или между STL- и RET-инструкциями.

СОСТОЯНИЕ	ИНСТРУКЦИИ		
	LD, LDI, OUT, NOP, AND, ANI, SET, RST, OR, ORI, PLS, PLF	ANB, ORB, MPS, MRD, MPP	MC, MCR
Инициализирующее состояние	допустимо	допустимо	не допустимо
Разветвление программы	Выходы	допустимо	не допустимо
	Условия дальнейшего включения	допустимо	не допустимо

Табл.5-2. Допустимые инструкции внутри состояний шагов

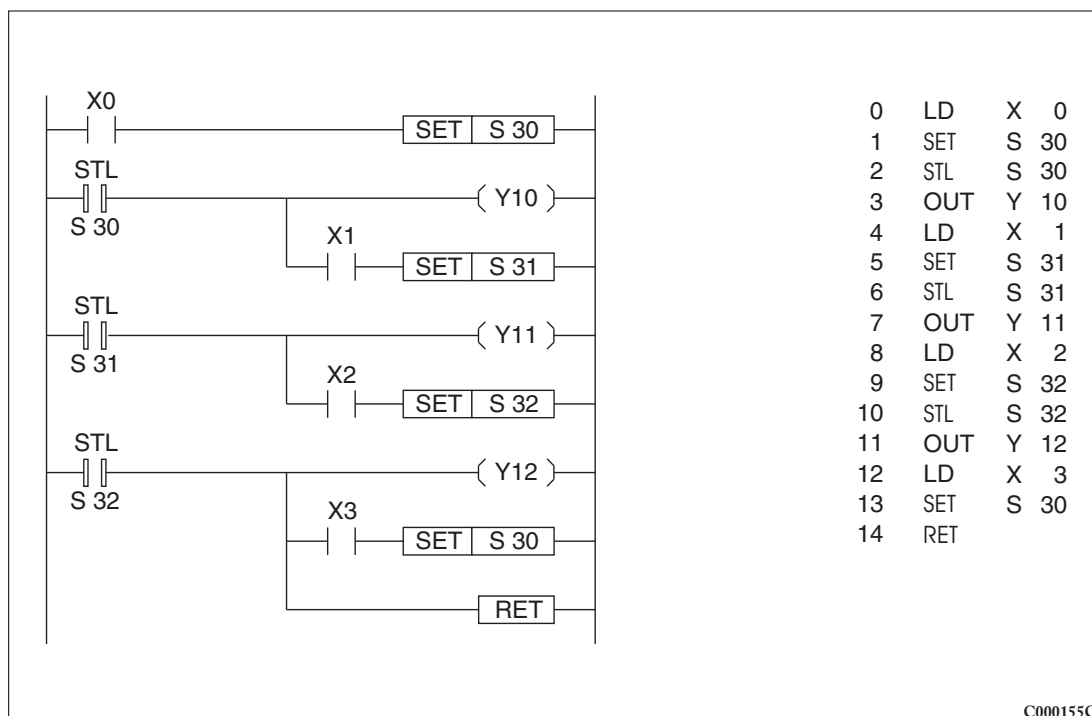


Рис.5-5. Пример программирования STL-, RET-инструкций

ЗАПИСЬ ВЫХОДОВ НЕСКОЛЬКО РАЗ

Одинаковые выходы могут опрашиваться с разными STL-инструкциями или операндами состояния шага.

ПРИМЕР ▾

Запись выходов несколько раз

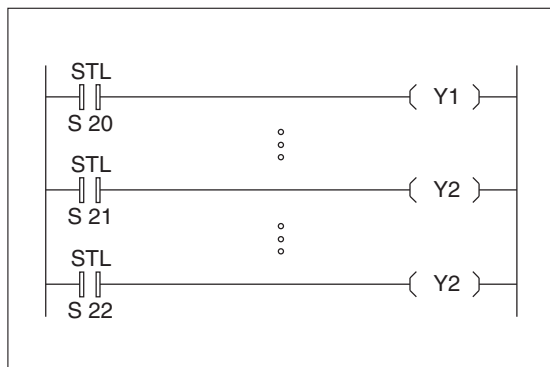


Рис.5-6.

Запись выходов несколько раз

C000138C

На приведенном участке программы одинаковый выход (Y2) опрашивается через разные STL-инструкции или операнды состояния шага (S21 или S22).

Y2 включается, если активен S21 или S22. Y2 выключается, если не активны S21 и S22. В этом случае двойная запись не создает проблем, так как шаги 21 и 22 не могут быть активными одновременно. ▴

ФУНКЦИЯ ОТКЛЮЧЕНИЯ УСЛОВИЯ ДАЛЬНЕЙШЕЙ РАБОТЫ

Как только статус S через STL-инструкцию включится, условия продолжения работы предыдущего статуса отключится. Это означает, что в цикле программы текущий, а также последующий статус на очень короткое время могут включиться одновременно.

УКАЗАНИЕ

Если один за другим следующие операнды не должны быть активны одновременно, то рекомендуется защищать операнды блокировкой.

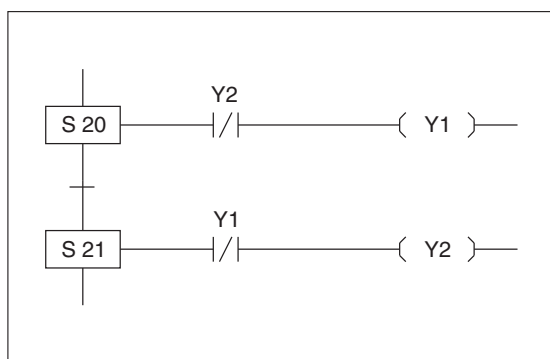


Рис.5-7.

Блокировочные мероприятия, чтобы исключить одновременные состояния включения

C000139C

МНОГОКРАТНАЯ ЗАПИСЬ ТАЙМЕРА

В одной программе таймер благодаря применению операнда шагового состояния (статуса) может записываться многократно. Однако одинаковый таймер не может применяться в двух один за другим следующих шагах.

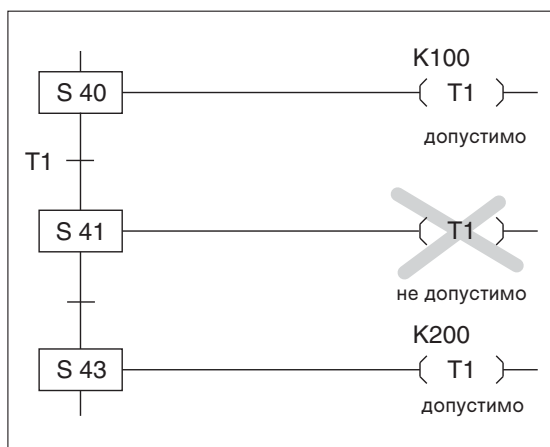


Рис.5-8.
Многократная запись таймеров

C000040C

УСЛОВИЯ ПОСЛЕДУЮЩИХ ВКЛЮЧЕНИЙ ПОСРЕДСТВОМ ИМПУЛЬСНОГО СИГНАЛА

Один за другим следующие шаги могут активизироваться по одинаковым условиям разрешения последующих шагов. Для этого требуется применение импульсной инструкции (PLS-инструкции).

ПРИМЕР ▾

Условие последующих включений через импульсный сигнал

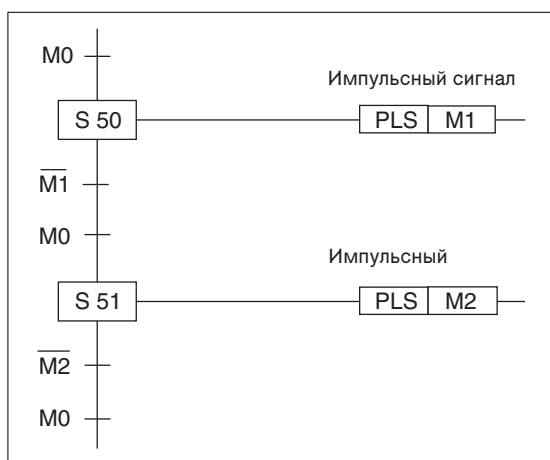


Рис.5-9.
Условие последующего включения по импульсному сигналу

C000041C

Первый сигнал M0 активизирует шаг состояния S50 и включает M1. M1 препятствует непосредственной активизации следующего шага. S51 станет активным лишь тогда, если будет введен следующий сигнал M0. △

Условие дальнейшего переключения на основе импульсных команд (FX2N, FX2NC и FX3U)

В контроллерах серий FX2N, FX2NC и FX3U условия дальнейшего переключения можно реализовать с использованием импульсных команд (LDP, LDF, ANP и т. п.) и импульсных маркеров M2800...M3071.

ПРИМЕР ▾

Условие последующих включений при применении импульсного маркера M2800:

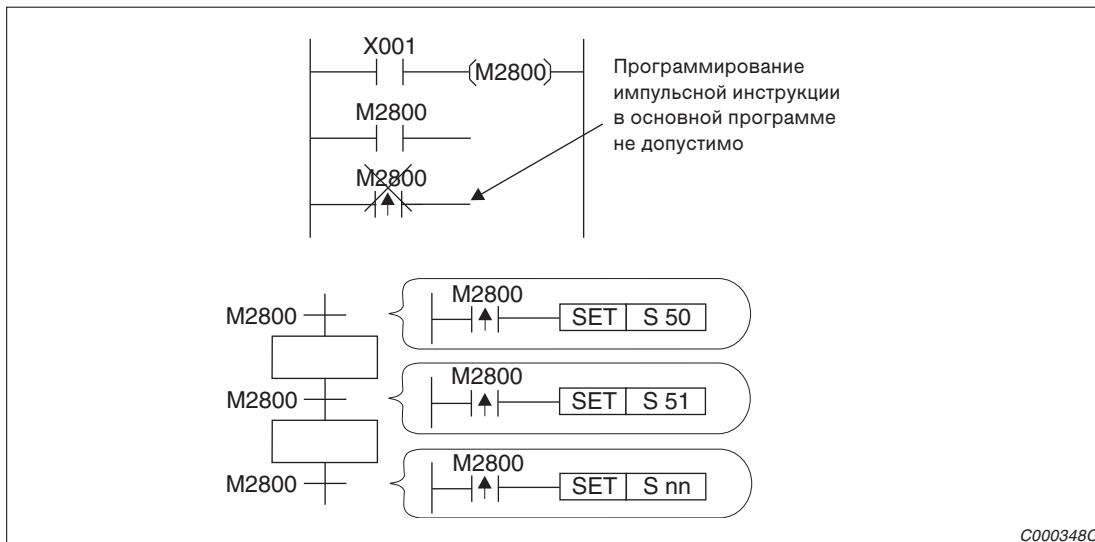


Рис.5-10. Условие последующего включения по импульсному маркеру M2800

Посредством включения маркера M2800 с X001 активизируется шаг S51. Одновременная активизация шага Snn невозможна, так как M2800 (импульсный) программировался бы 2-ой раз. При следующем включении M2800 через X001 шаг Snn активизируется, так как шаг S50 не активен, а импульсный маркер M2800 может быть активным только один раз.

△

5.3 ИНИЦИАЛИЗАЦИЯ СОСТОЯНИЯ ШАГА

Каждое состояние шага требует инициализации. Для этого имеются, например, инициализирующие операнды S0...S9. С помощью инициализирующих операндов можно выполнить различные шаги процессов внутри STL-программы, чтобы реализовать, например, разные процессы работы (наладочный и автоматический режимы, подход к нулевой точке и т.д.).

ПРИМЕР ▾

Инициализация шаговых состояний

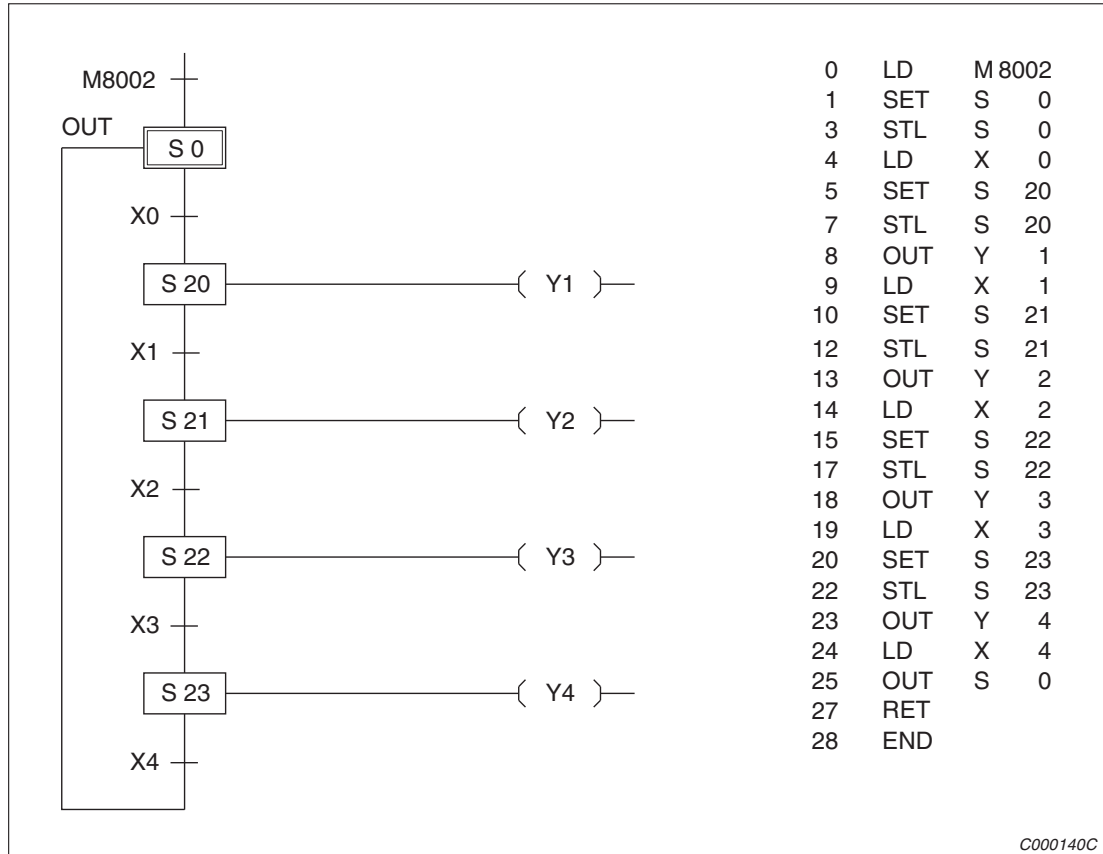


Рис.5-11. Пример программирования инициализации шаговых состояний

Меркер M8002 задействует при включении ПК определенное системное состояние (см.также раздел 10.1.1). Инициализация шаговой цепи определится включением S0.

Условия шагов для каждого последующего шага выполняются уже описанным способом.

Чтобы осуществить новый пуск или повторение шаговой цепи, снова должен включиться S0△

5.4 STL-РАЗВЕТВЛЕНИЯ

Программируемое управление на ПК семейства FX могут обрабатывать различные, друг от друга независимые процессы состояний и разветвления. Нужно различать процессы:

- Простой (линейный) процесс
- Селективное разветвление
- Параллельное (одновременное) разветвление
- Переходное разветвление

5.4.1 ПРОСТОЙ (ЛИНЕЙНЫЙ) ПРОЦЕСС

При простом процессе шаговые состояния обрабатываются последовательно (один за другим). Последовательность обработки определяется только положением шагового состояния в простом процессе и благодаря независимости от адреса шагового состояния.

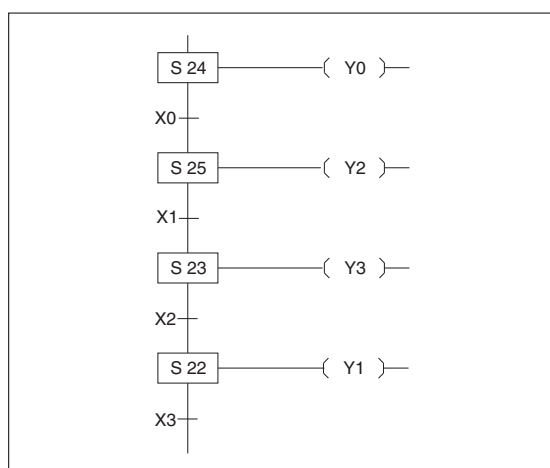
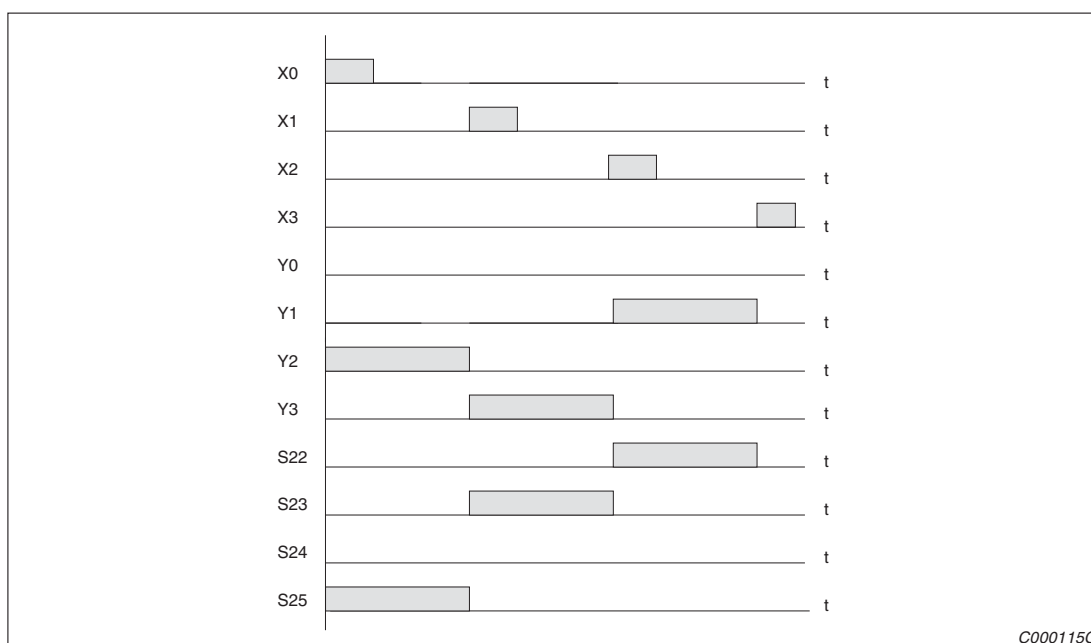


Рис.5.12.
Пример простого процесса

C000149C



C000115C

Рис.5.13. Временная диаграмма простого процесса

5.4.2 СЕЛЕКТИВНОЕ РАЗВЕТВЛЕНИЕ

При селективном разветвлении имеется возможность произвести в этой операции выбор среди двух или более процессов состояний.

Из одного шагового состояния разветвление может создавать несколько (максимум 8) процессов состояний.

В зависимости от соответственно примененных входных условий производится выбор, какой процесс состояний должен активизироваться в программе.

УКАЗАНИЕ

Может программироваться максимум 8 разветвлений, выходящих из одного шагового операнда. Общее количество всех селективных разветвлений не должно превышать 16.

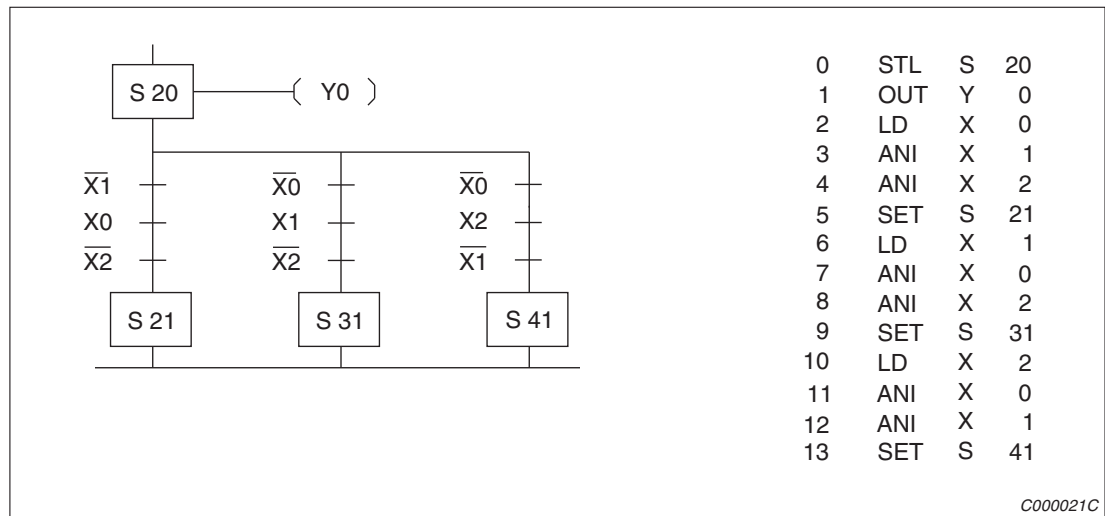


Рис.5-14. Пуск (начало) селективного разветвления

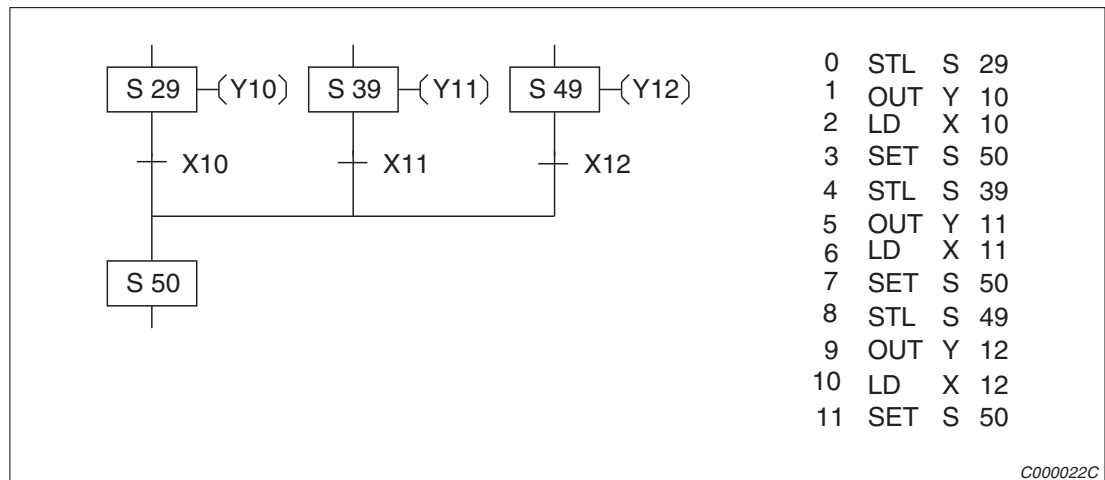


Рис.5-15. Сборка (окончание) селективного разветвления

ПРИМЕР ▾

Блочная диаграмма, контактная схема (КОР) и листинг инструкций (AWL) селективного разветвления

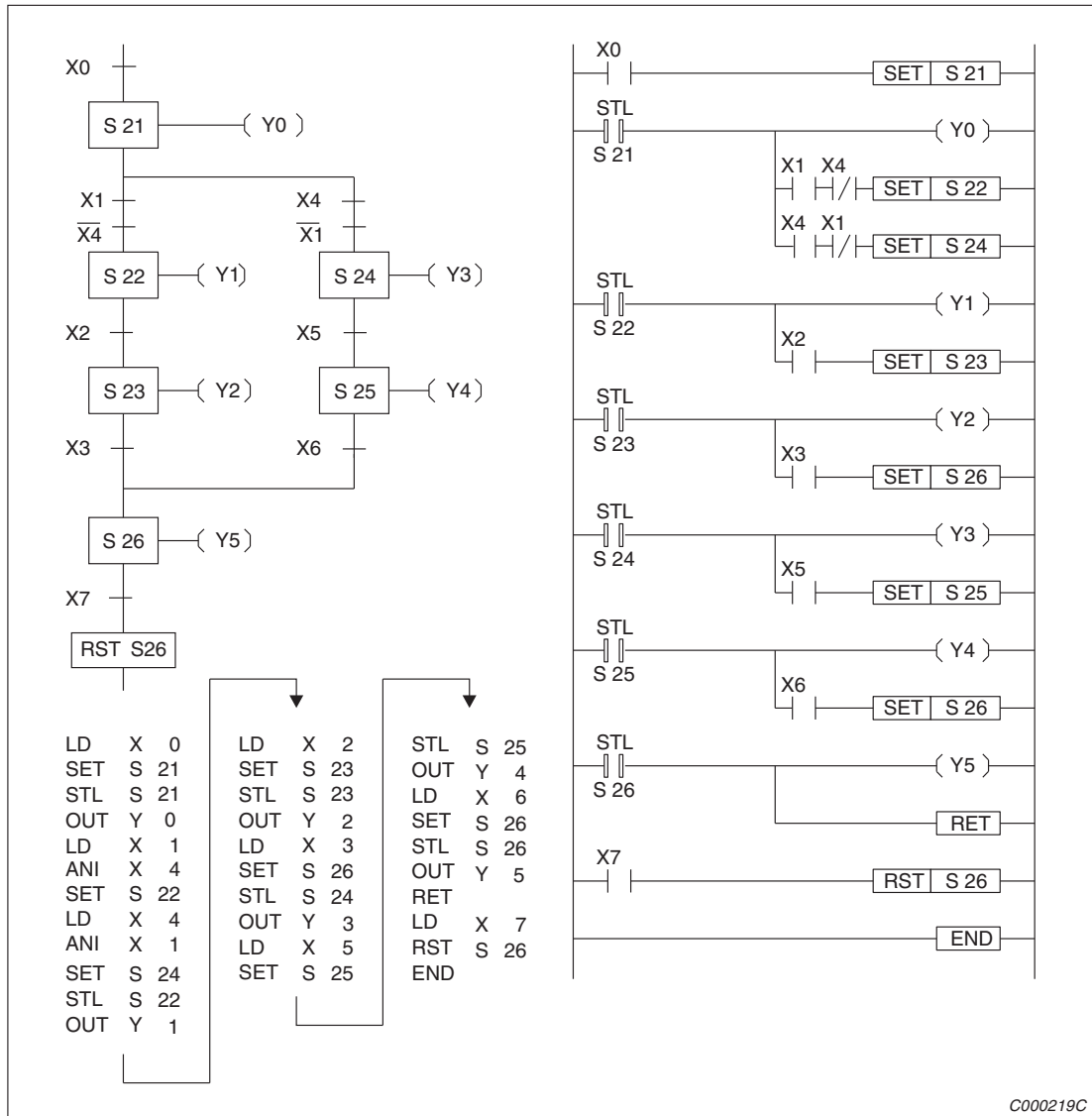


Рис.5-16. Пример программирования селективного разветвления

Здесь всегда можно выполнять только одну функцию. Это определяется тем, что S21 автоматически выключится, если не включится ни S22 ни S24.

S26 включится по шагам S23 и S25. Соответственно этому при включении S26 отключается или S23 или S25. △

5.4.3 ПАРАЛЛЕЛЬНОЕ РАЗВЕТВЛЕНИЕ

При параллельном разветвлении два или несколько процессов состояний обрабатываются одновременно. Из одного состояния разветвление может создавать несколько (максимум 8) процессов состояний.

УКАЗАНИЕ

Количество всех разветвлений не должно превышать 16.

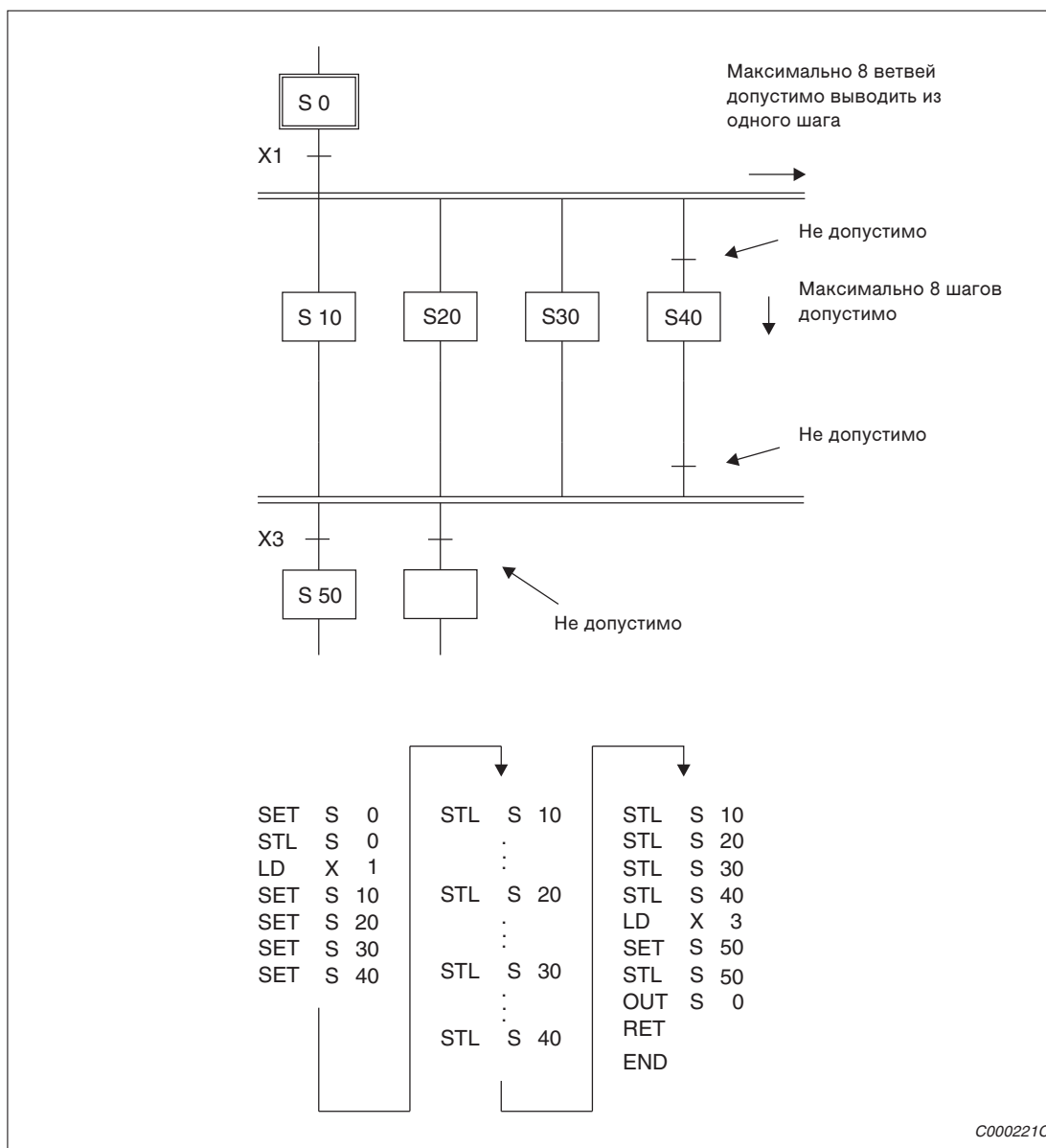


Рис.5-17. Допустимые параллельные разветвления

В зависимости от соответственно примененных входных условий выполняется разветвление на отдельные ветви. В противоположность к селективному разветвлению при параллельном разветвлении могут одновременно обрабатываться несколько процессов состояний.

Включенные операнды параллельных шагов отключаются лишь тогда, когда обработаются шаги, лежащие после объединения параллелей.

УКАЗАНИЕ

После разветвления (начало) и перед объединением (концом) не допустимы никакие логические связи.

Параллельное разветвление может максимум содержать 8 параллельных ветвей, из которых каждая ветвь может содержать максимум 8 один за другим следующих (последовательных) шагов.

Внутри параллельного разветвления нельзя программировать никаких селективных разветвлений.

ПРИМЕР ▾

Блочная диаграмма, контактная схема (KOP) и листинг инструкций (AWL) параллельного разветвления

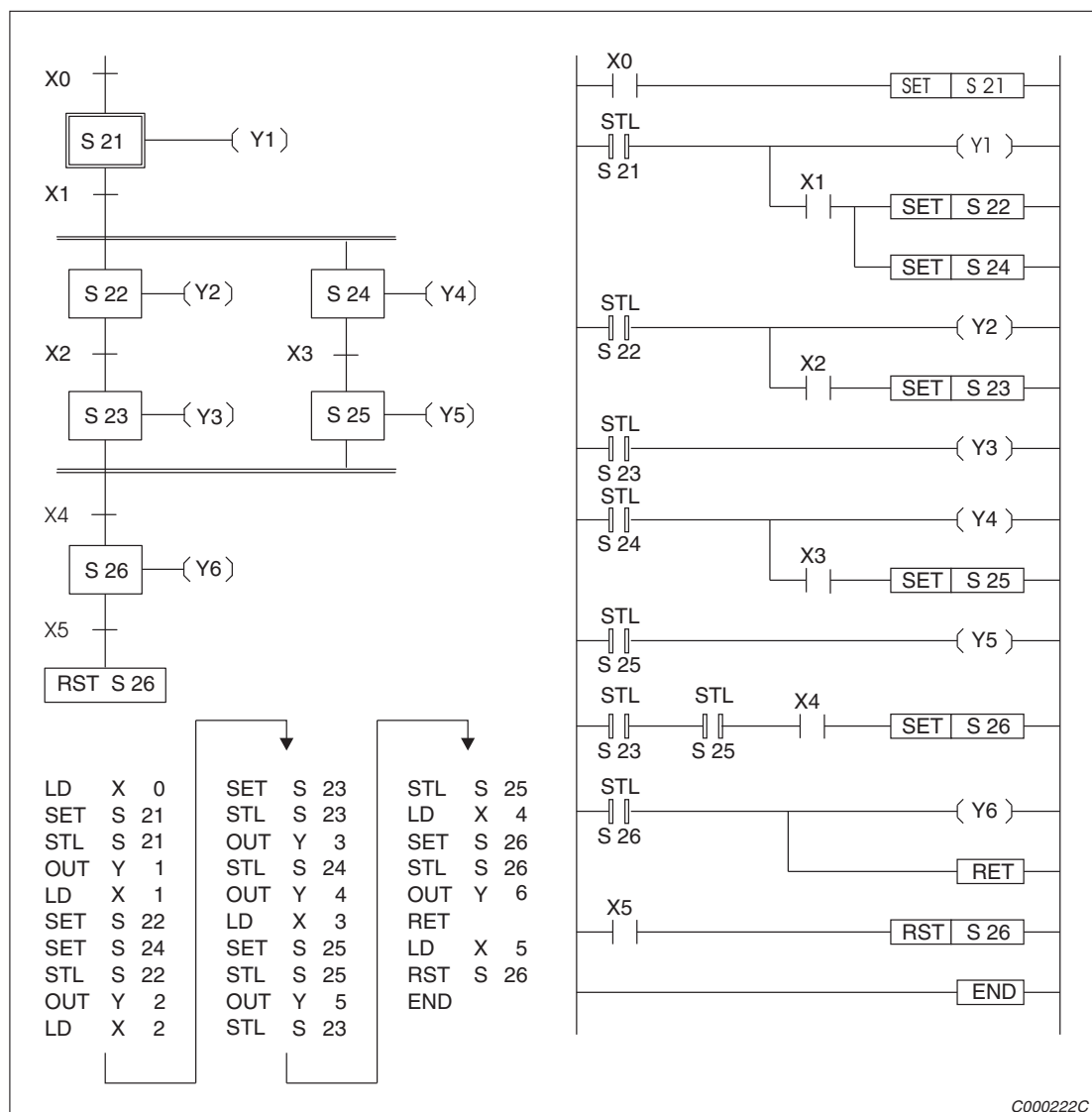


Рис.5-18. Параллельное разветвление

Шаг S26 выполняется в зависимости от X4 лишь после выполнения шагов S23 и S25. △

5.4.4 КОМБИНАЦИЯ ИЗ СЕЛЕКТИВНОГО И ПАРАЛЛЕЛЬНОГО РАЗВЕТВЛЕНИЯ

Селективное и параллельное разветвление могут комбинироваться в одной программе STL.

ПРИМЕР ▾

Комбинация из селективного и параллельного разветвления

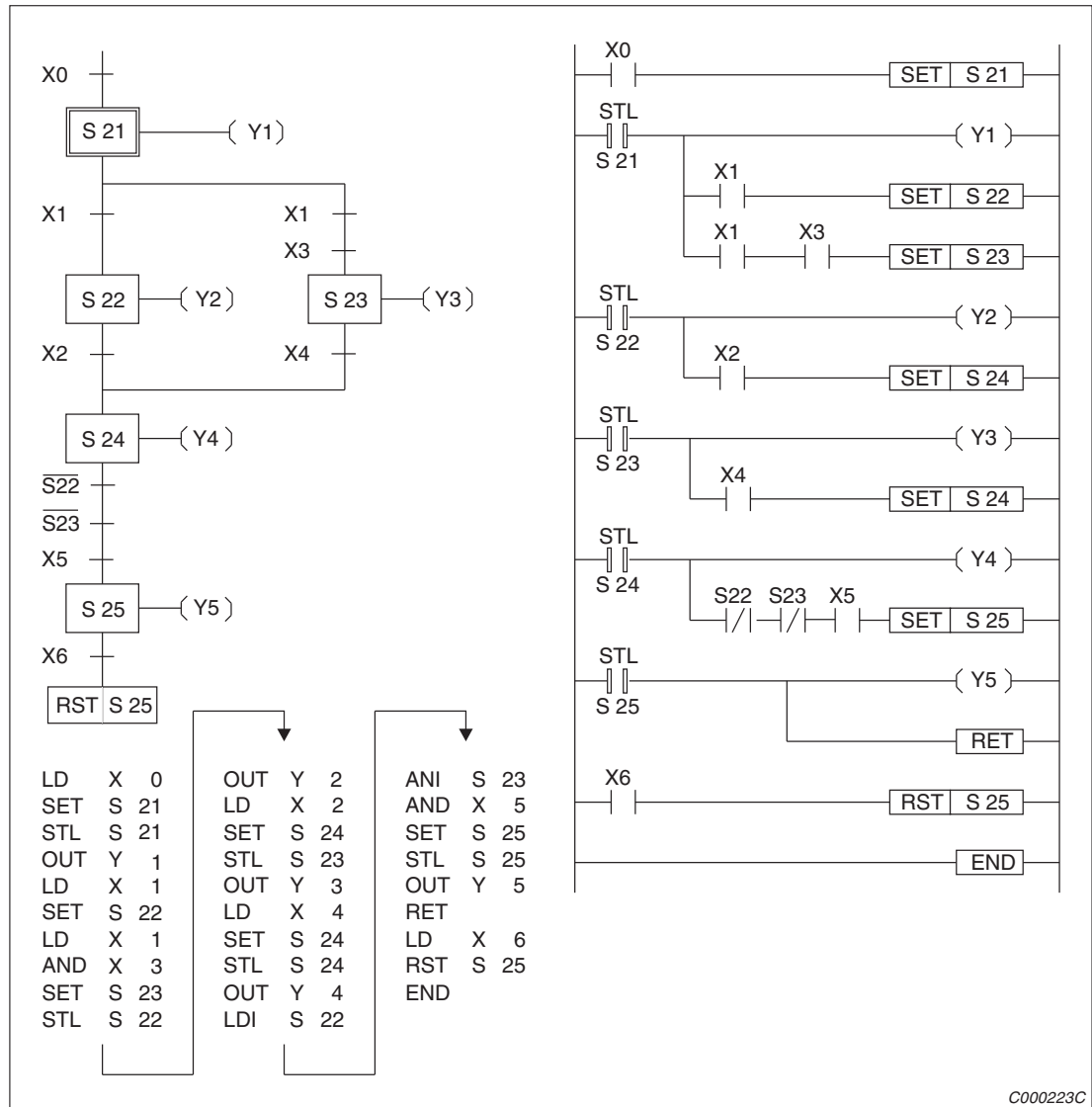


Рис.5-19. Комбинация из селективного и параллельного разветвления

Если в примере X3 включится, выполняться условия для параллельного разветвления. Если X3 не включится, выполнится селективная программа обработки, т.е. S24 сможет включиться только через S22.

S24 включится только тогда, если S22 или S23 отключится

S25 включится только тогда, если S22 и S23 отключатся

△

5.4.5 ПРОГРАММИРОВАНИЕ ПУСТОГО СОСТОЯНИЯ

Для реализации нескольких шаговых процессов необходимо программирование пустого состояния. Эта возможность создает лучший обзор процесса программы, а также экономию шагов программы.

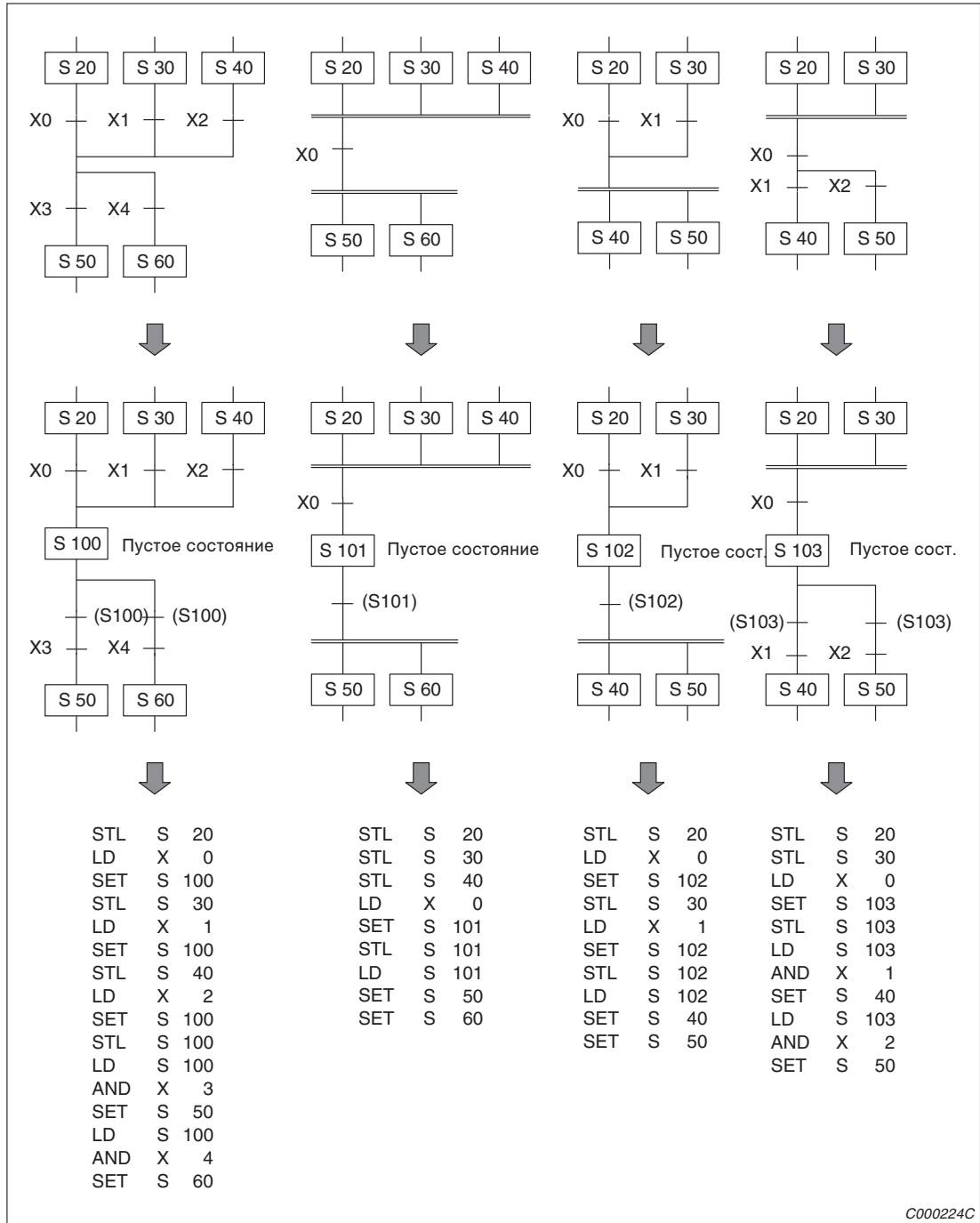


Рис.5-20. Программирование пустого состояния

5.4.6 ПЕРЕХОДНОЕ РАЗВЕТВЛЕНИЕ

Имеется возможность перескочить через часть области (схемы) последовательности состояния или многократно выполнить петлю программы.

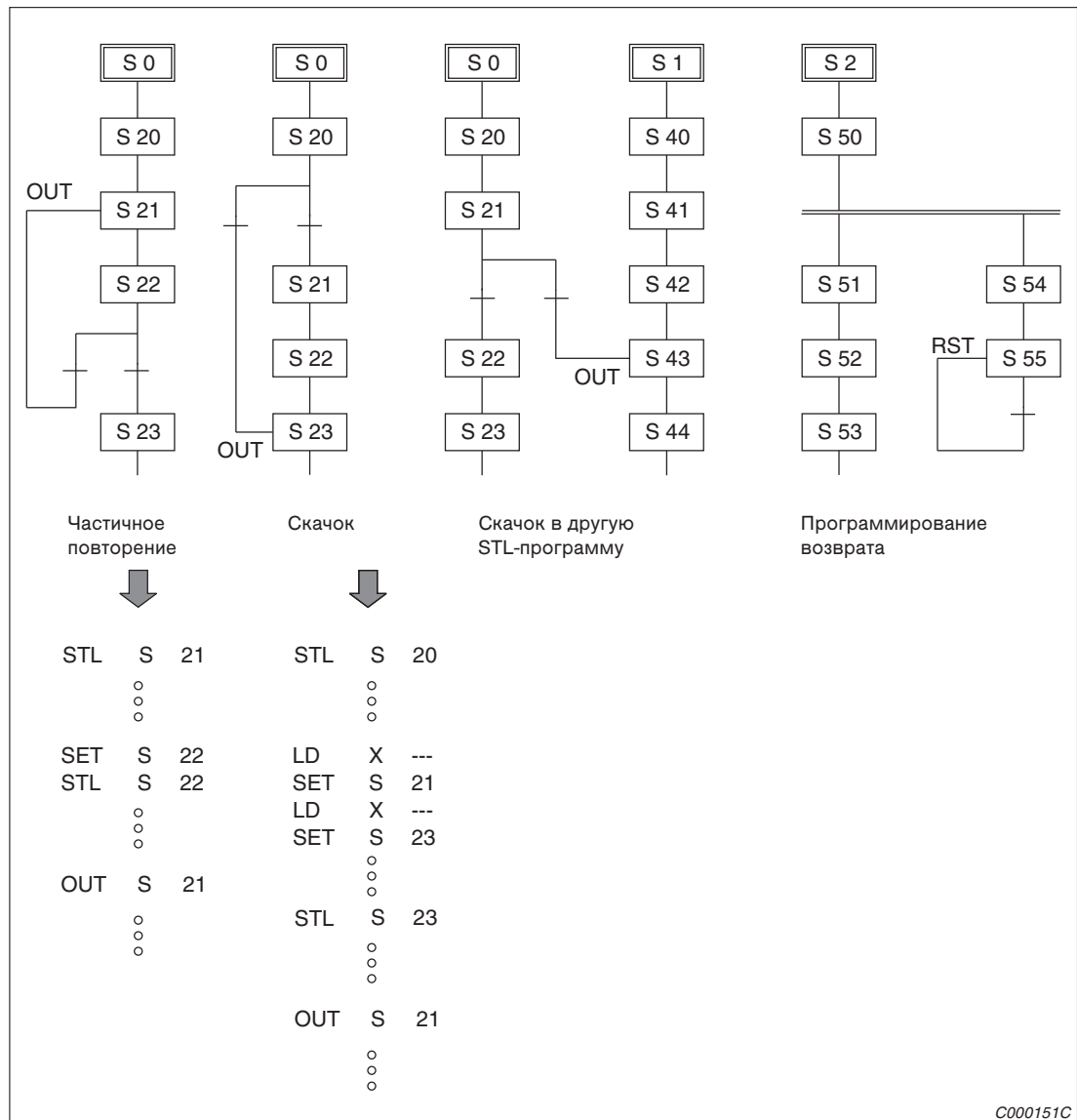


Рис.5-21. Пример программирования различных возможностей переходного разветвления

УСЛОВИЕ ПОСЛЕДУЮЩЕГО ВКЛЮЧЕНИЯ В ДРУГИЕ ШАГОВЫЕ ЦЕПИ

Для условий включения из одной шаговой цепи в другую шаговую цепь может вместо SET-инструкции программироваться также OUT-инструкция (см. OUT S31 в примере к шаговой цепи I). Эта альтернатива не влияет на внутреннюю обработку программы управления.

ПРИМЕР ▾

Условие последующего включения в другую шаговую цепь

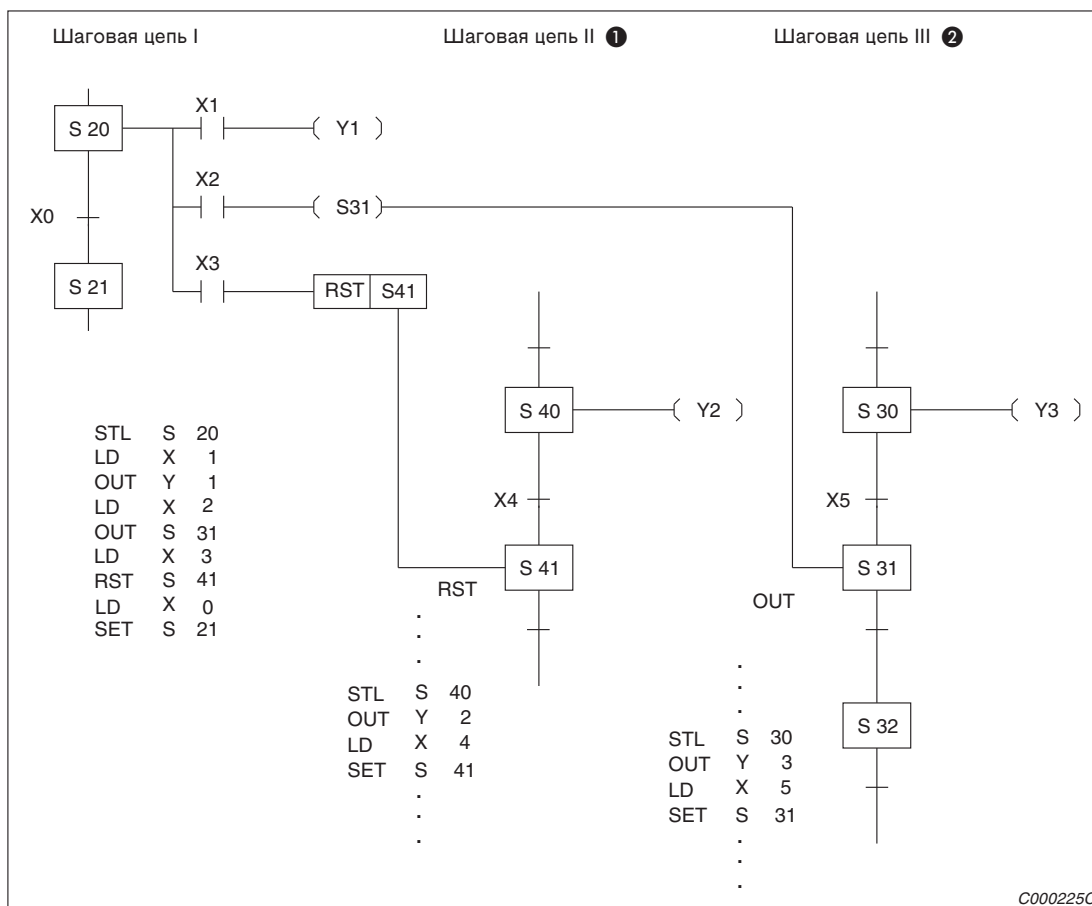


Рис.5-22. Условие последующего включения в другую шаговую цепь

- ❶ Состояние шага S41 в шаговой цепи II включается по S40 и по входу X4. Состояние шага S41 отключается, если S20 и вход X3 в шаговой цепи I включены.

Если процесс отключения завершен, то шаговая цепь и дальше находится в зависимости от состояния шага S20, на который S41 не оказывает влияния.

- ❷ Состояние шага S31 в шаговой цепи III включается, если S20 и вход X2 в шаговой цепи I включены. S31 отключается после того, как к S32 передается условие включения. Состояние шага S20 отключится, если к состоянию шага S31 передается условие включения.

△

5.5 ПРИМЕР КОНТРОЛЯ ЗАГРУЗКИ И РАЗГРУЗКИ

ПРИМЕР ▾

В этом примере передвижной контейнер для транспортировки сыпучих грузов перемещается по жестко заданному отрезку и на определенных местах загружается и разгружается.

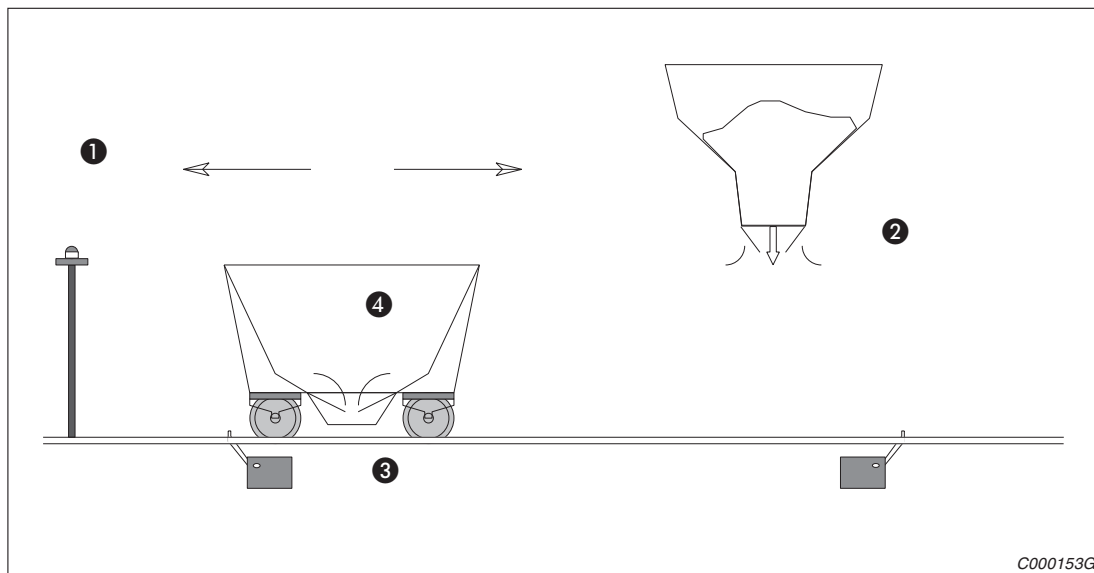


Рис.5-23 Пример контроля загрузки и разгрузки контейнера

- ① После воздействия на пусковую кнопку X0 тележка перемещается в направлении места загрузки и останавливается на конечном выключателе X1.
- ② Загрузчик силоса открывается на 7 секунд (Y1).
- ③ Тележка едет назад и останавливается на конечном выключателе X2 в месте разгрузки.
- ④ Клапан разгрузки тележки открывается на 5 секунд (Y3).

ПРИМЕР ▾

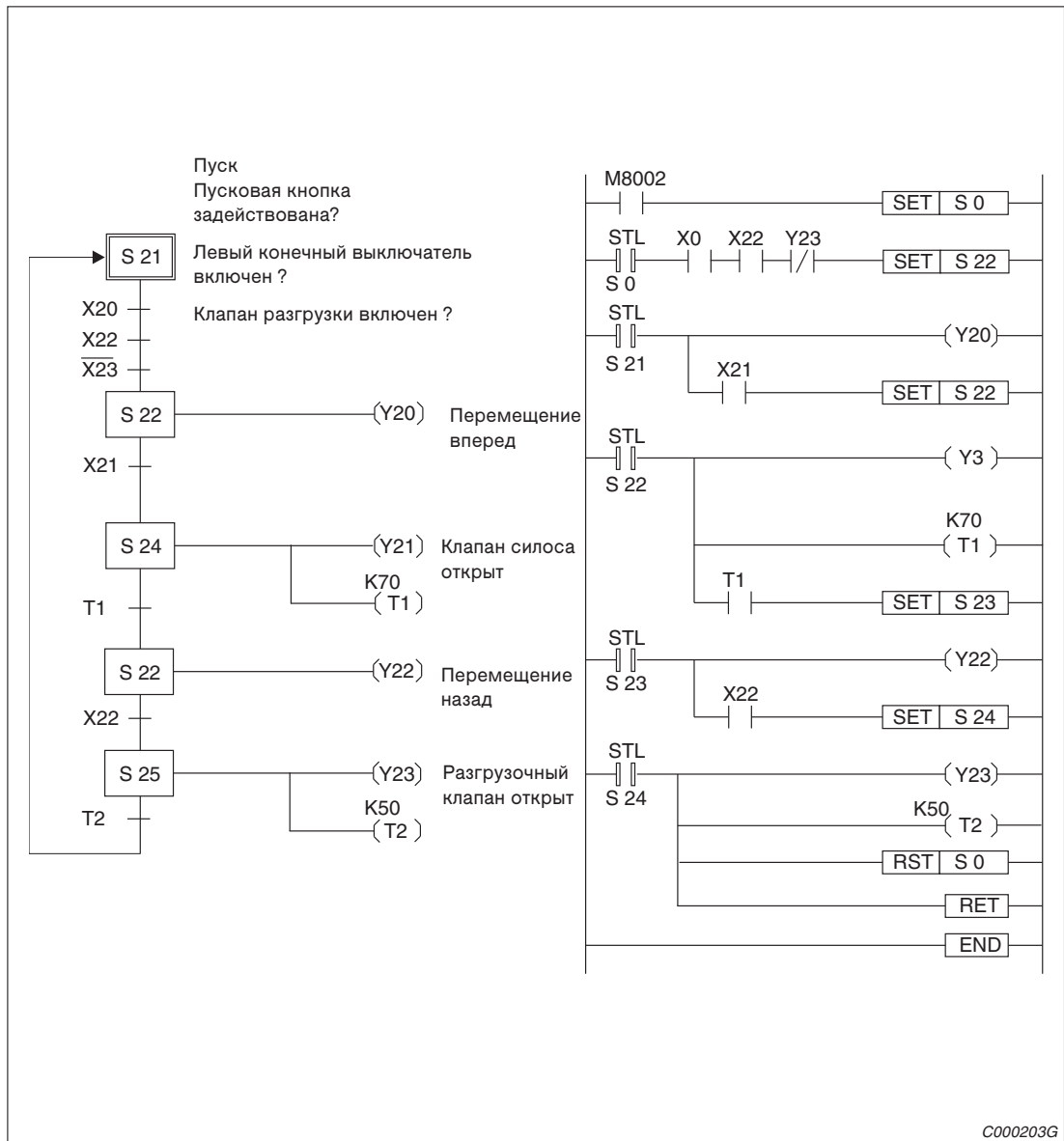
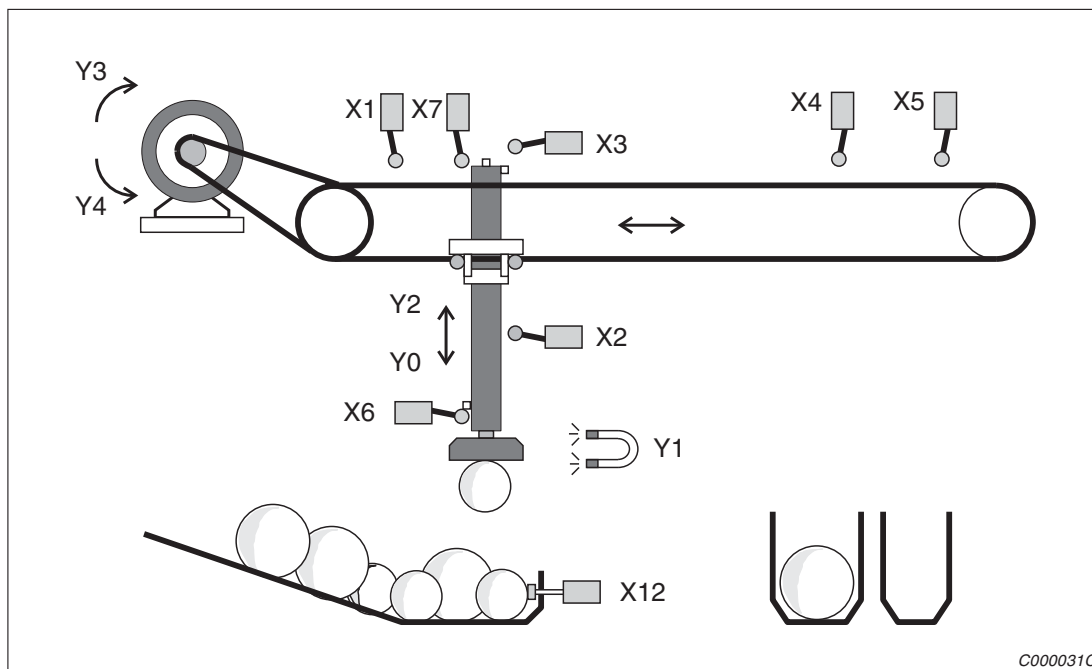


Рис.5-24. Программа процесса контроля загрузки и разгрузки контейнера



5.6 ПРИМЕР ТРАНСПОРТИРОВКИ И СОРТИРОВКИ

Этот пример отражает механизм управления, в котором разные по величине стальные шары поднимаются из одного лотка и транспортируются по транспортеру. В конце транспортного пути шары сортируются в соответствующие сосуды в зависимости от их величины.



C000031C

Рис.5-25. Устройство сортировки шаров

- ① Гидравлическая рука опускается из исходного положения (Y0=вкл)
- ② Если нижняя граница не достигнута, то в шахте забора шара находится большой шар (X2=выкл, X6=вкл). Контакт X2 замкнут, если в шахте забора не лежит малый шар.
- ③ Включается электромагнит (Y1=вкл), и шар захватывается.
- ④ Гидрорука поднимается (Y2=вкл). Рука останавливается при достижении верхней границы.
- ⑤ Гидрорука перемещается вправо (Y3=вкл)
- ⑥ Если был взят малый шар, двигатель останавливается при достижении конечного выключателя X5. Если был взят большой шар, останов мотора выполняется при достижении конечного выключателя X4.
- ⑦ Гидрорука опускается (Y0=вкл).
- ⑧ После достижения нижнего положения (X6) магнит выключается (Y1=выкл).
- ⑨ Гидравлическая рука поднимается до верхней границы (X3) (Y2=выкл).
- ⑩ Гидрорука перемещается в исходную позицию (Y4=вкл).
- ⑪ Достигнута исходная позиция (X7=вкл).

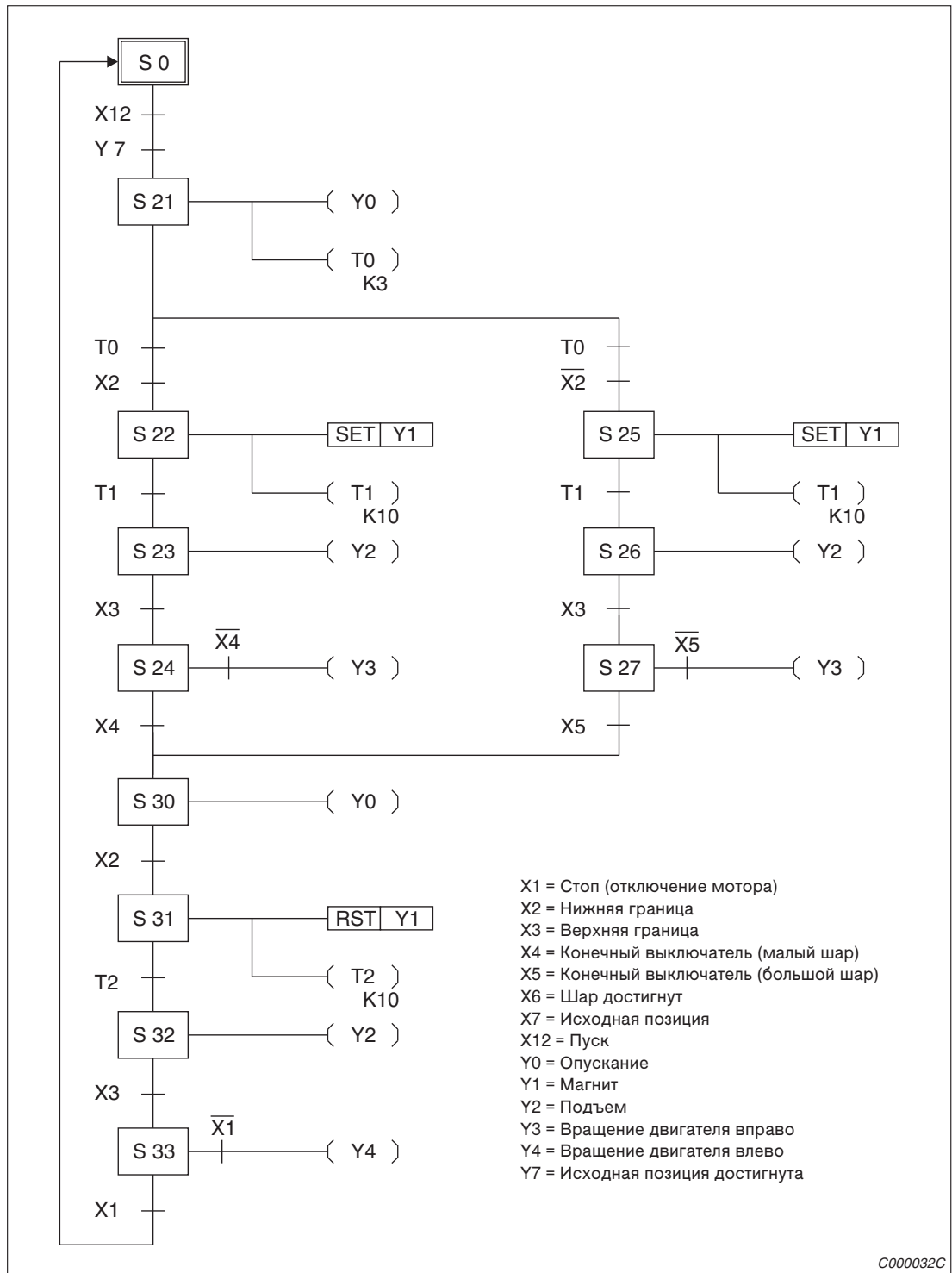


Рис.5-26. Пример программирования приведенного впереди устройства сортировки

6 ПРИКЛАДНЫЕ ИНСТРУКЦИИ

6.1 ОБЩИЕ УКАЗАНИЯ

Эта глава описывает прикладные инструкции семейства FX. С помощью этих инструкций можно реализовывать специальные функции (например, арифметические функции). Описание этих инструкций начинается с обзора в форме таблиц, в которых имеется вся важная информация для работы с ними.

6.1.1 ПОЯСНЕНИЯ К ОПИСАНИЮ СЛОВНЫХ ИНСТРУКЦИЙ

Этот раздел дает пояснения по структуре таблиц прикладных инструкций, которые приведены для каждой инструкции в начале соответствующего раздела.

②			①					
			CMP		FNC 10			
④			③ Назначение инструкции					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●	●
Операнд	S1+, S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y, M, S	●	16бит а	32бит а	CMP	7	
⑤			⑥	⑦				

① ИНСТРУКЦИЯ

В верхней строке указывается наименование инструкции и относящийся к ней FNC-номер (FNC - номер функции), которое применяется при программировании на языке Листинга инструкций (AWL). В зависимости от системы программирования можно применять или имя инструкции или ее номер.

② СИМВОЛ КОНТАКТНОЙ СХЕМЫ

Символ контактной схемы применяется при программировании на языке контактной схемы. Символ контактной схемы состоит из инструкции и примененного операнда.

③ ЗНАЧЕНИЕ

Здесь приводится краткое описание назначения инструкции.

④ CPU (ТИП ПК)

На этом месте знаком ● показана серия ПК MELSEC, в которой данная инструкция выполняется.

⑤ ОПЕРАНД

Все операнды, применяемые с этой инструкцией, перечисляются в этом поле. Подробное описание операндов и структуры данных содержится в разделах 6.1.2...6.14.

6 ИМПУЛЬСНАЯ ИНСТРУКЦИЯ

Согласно наличию знака “●” инструкция может выполняться также при поднимающемся фронте входной команды (см. также раздел 6.1.5). В этом случае после инструкции должен приводиться знак “P”.

7 ОБРАБОТКА

Здесь указывается, с какой по объему инструкцией работают - 16-ти или 32-ух разрядной инструкцией.

8 ШАГИ ПРОГРАММЫ

Здесь указывается количество шагов программы, которые требуются для полного выполнения инструкции.

6.1.2 ОПИСАНИЕ ОПЕРАНДОВ**БИТОВЫЙ ОПЕРАНД**

Битовый операнд может принимать два состояния сигнала (“0” и “1”). Его состояние сигнала может определяться тем самым с помощью бита (0 и 1)..

Битовый операнд	Код операнда
Вход	X
Выход	Y
Меркер	M
Состояние шага	S

Табл. 6-1.
Битовый операнд

ПОСЛОВНЫЙ ОПЕРАНД

Пословный операнд может принимать информационное состояние, которое состоит из нескольких бит (числовое значение данных). При этом 8 бит составляют байт и 2 байта - слово данных.

Пословный операнд	Код операнда
Таймер	T
Счетчик	C
Регистр данных	D
Индексный регистр	V, Z

Табл. 6-2.
Пословный операнд

6.1.3 ОБЪЕДИНЕНИЕ БИТОВЫХ ОПЕРАНДОВ

Несколько, друг за другом следующих битовых операндов, могут собираться в слово данных. Благодаря этому имеется, например, возможность сразу обрабатывать состояние сигналов нескольких входов.

Количество адресов битовых операндов, которые должны сработать от пословной инструкции, определяются указанием константы К. При 16-ти битовой инструкции может задаваться до 16, а при 32-х битовой инструкции до 32 адресов операндов в наборах по 4 операнда. Количество совмещаемых адресов операндов определяется длиной блока.

Для 16-ти битовых инструкций длина блоков лежит в области от К1 до К4.

Длина Блока	Количество Адресов
К1	4
К2	8
К3	12
К4	16

Табл. 6-3.

Длина блоков при 16-ти битовой инструкции

Для 32-х битовых инструкций длина блоков лежит в области от К1 до К8.

Длина Блока	Количество Адресов
К1	4
К2	8
К3	12
К4	16
К5	20
К6	24
К7	28
К8	32

Табл. 6-4.

Длина блоков при 32-х битовой инструкции

ЗАДАНИЕ НАЧАЛЬНОГО (ПУСКОВОГО) АДРЕСА БЛОКА

Задание начального адреса определяет начало блока. При указании начального адреса может применяться любое число.

УКАЗАНИЕ

При определении входов X или выходов Y применяйте по возможности только начальные адреса кратные 10 (например, X0, X10 и т.д.).

При определении операндов M и S нужно по возможности указывать начальные адреса кратные 8.

ПРИМЕР ▾ Установка длины блока и начального адреса.

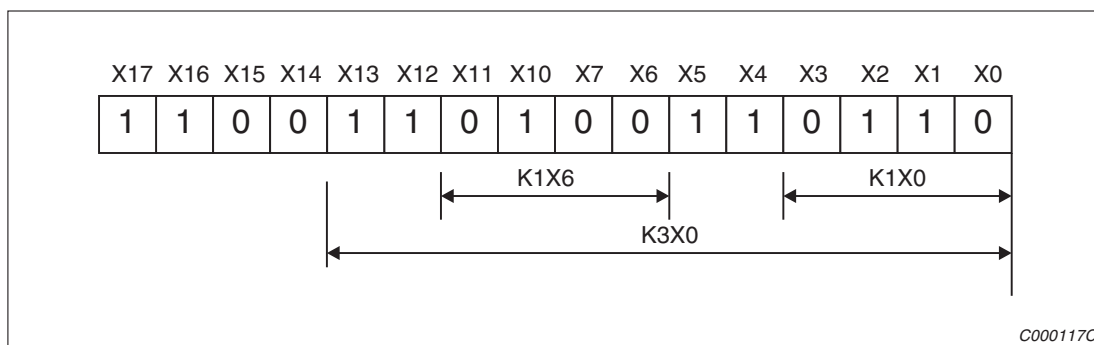


Рис. 6-1. Пример установки длины блока и начального адреса

K1X0: X0...X3 → 4 входа, начальный адрес X0

K1X6: X6...X11 → 4 входа, начальный адрес X6

K3X0: X0...X13 → 12 входов, начальный адрес X0

△

6.1.4 СТРУКТУРА ДАННЫХ

ИСТОЧНИК ДАННЫХ (S)

Источники данных являются данными, которые должны обрабатываться с помощью словных инструкций. Источники данных содержат один или несколько адресов операндов и могут состоять из констант и/или битовых или словных операндов.

Константы являются числовыми значениями, которые предварительно задаются для выполнения определенных операций. Значение константы определяется при настройке программирования и не может больше изменяться во время обработки программы.

С помощью битового или словного операнда определяется адрес операнда, по которому хранятся данные, которые должны обрабатываться. Изменение данных возможно в любое время во время обработки программы.

Группа Данных		Кодовое обозначение
Русское обозначение	Английское обозначение	
Источник данных	Source	(S)
Источник данных 1	Source 1	(S1)
Источник данных 2	Source 2	(S2)

Табл. 6-5.
Характеристика (код) источника данных

КОНЕЧНЫЕ ДАННЫЕ (D)

Конечными являются данные, которые содержат результат операции после исполнения словной инструкции. Конечные данные также состоят из одного или нескольких адресов операндов и могут состоять из битовых или словных операндов. Возможная область адресов конечных данных определяется константами и должна соответствовать величине области адресов источников данных.

Группа Данных		Кодовое обозначение
Русское обозначение	Английское обозначение	
Конечные данные	Destination	(D)
Конечные данные 1	Destination 1	(D1)
Конечные данные 2	Destination 2	(D2)

Табл. 6-6.
Характеристика (код) конечных данных

6.1.5 ИСПОЛНЕНИЕ СЛОВНЫХ ИНСТРУКЦИЙ

Имеется две возможности исполнения этих инструкций:

- Исполняемый сигнал может быть статическим. При включении сигнала словная инструкция выполняется.
- Исполняемый сигнал может, кроме того, формироваться как импульс при поднимающемся или падающем фронте. Словная инструкция при этом может только тогда выполняться, когда ее входной сигнал изменяется с "0" на "1" или же с "1" на "0". Это может реализовываться, например, подключаемой впереди функцией импульса (PLS-, PLF-инструкцией).

ПРИМЕР ▾

Выполнение с помощью меркера, который включается на время цикла программы (скана)

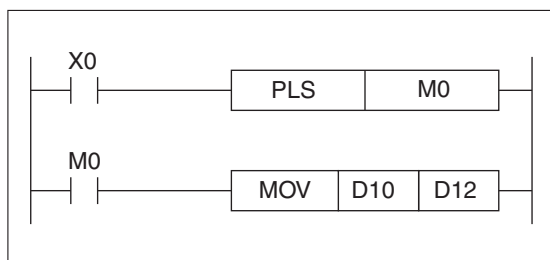


Рис. 6-2.

Выполнение с помощью меркера, который включается на время скана

C000118C



ПРИМЕР ▾

Выполнение по импульсному сигналу

Дополнительно к уже показанной возможности MELSEC-управление располагает командным параметром "P", который позволяет опознавать изменение управляющего сигнала при возрастающем фронте.

Функция этого параметра соответствует инструкции "PLS" базового набора команд.

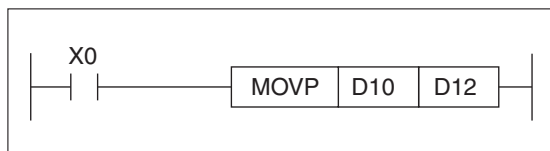


Рис. 6-3.

Выполнение словной инструкции по импульсному сигналу

C000003C

Функция этого примера идентична примеру на рис. 6-3.

Инструкция MOV выполняется, если на входе X0 происходит смена сигнала с "0" на "1". Инструкция выполняется только один раз. Только при повторной смене сигнала с "0" на "1" инструкция выполнится снова.

Инструкция не выполняется, если X0 отключается.



ПРИМЕР ▾

Выполнение словной инструкции по статическому сигналу

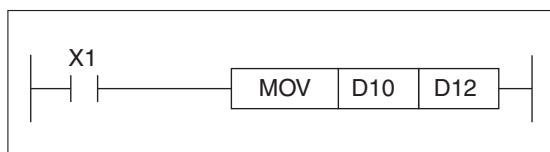


Рис. 6-4.

Выполнение словной инструкции по статическому сигналу

C000121C

Инструкция MOV выполняется, если на входе X1 есть сигнал "1". Инструкция выполняется каждый цикл программы до тех пор пока имеется сигнал "1".

Инструкция не выполняется, если X1 отключен.



6.1.6 ПРИМЕНЕНИЕ ИНДЕКСНЫХ РЕГИСТРОВ V, Z

Индексные регистры V и Z применяются, чтобы при инструкциях перехода или сравнения суммировать по адресу операнда индексируемые значения.

Индексные регистры V и Z являются 16-ти битовыми регистрами.

В 32-х битовых инструкциях оба индексных регистра V и Z должны применяться комбинированно. Z запоминает младшие 16 бит, V запоминает старшие 16 бит. В качестве целевого (конечного) адреса указывается индексный регистр Z.

УКАЗАНИЕ

Операции, при которых может применяться индцирование отмечаются знаком плюс "+": например, (S+) и (D+) (здесь S и D исходные и конечные данные)

ПРИМЕР ▾

Применение индексных регистров V, Z

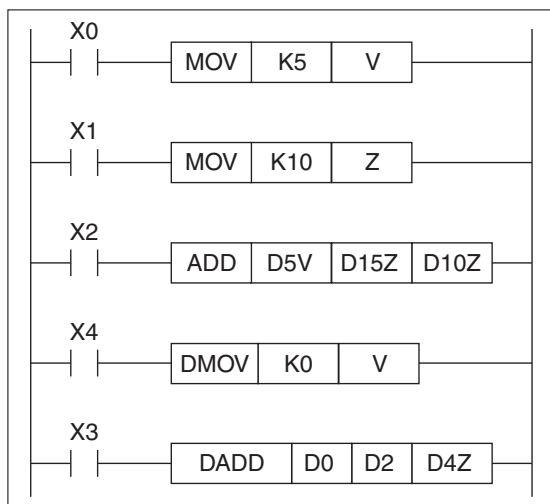


Рис. 6-5.
Пример программирования индексных регистров V, Z

C000120C

- Значение 5 (K5) передается с помощью MOV-инструкции в индексный регистр V.
- Значение 10 (K10) передается с помощью MOV-инструкции в индексный регистр Z.
- D5V должно суммироваться с D15Z. Результат должен быть записан в регистр данных D10Z.
- Расчет регистра данных:

$$V = 5 (K5)$$

$$Z = 10 (K10)$$

$$D5V = D10 (D5 + V = D5 + 5 = D10)$$

$$D15Z = D25 (D15 + Z = D15 + 10 = D25)$$

$$D10Z = D20 (D10 + Z = D10 + 10 = D20)$$

- Значение 0 (K0) передается с помощью MOV-инструкции в индексный регистр V. Имеется 32-х битная операция.
- D0 и D1 должны суммироваться с D2 и D3. Результат должен быть сохранен в регистрах данных D14 и D15.

△

6.1.7 ЗНАЧЕНИЯ ФЛАГОВ

При обработке некоторых словных инструкций автоматически из ПК включаются или отключаются различные флаги (специальные меркеры). Используемый флаг показывает определенное состояние программы (например, превышение допустимой числовой области данных при выполнении словной инструкции). Этот флаг каждый раз включается или отключается, если в программе активизируется соответствующая инструкция. Однако включение или отключение флага не произойдет, что должно было бы отразиться в следующем скане, если не выполнится инструкция, флаг которой изменяет свое состояние.

Обзор всех флагов и их значение находятся в разделе 10.1.3.

6.1.8 ОШИБКИ РАБОТЫ ПРОГРАММЫ ПРИ ВЫПОЛНЕНИИ СЛОВНЫХ ИНСТРУКЦИЙ

При ошибочном программировании словных инструкций или адресов операндов запоминаются сообщения об ошибках в регистре данных ошибок. Глава описания ошибок содержит детальный обзор всех регистров данных ошибок и сообщений об ошибках.

6.1.9 32-Х БИТНЫЕ ИНСТРУКЦИИ

Если инструкция должна выполняться как 32-х битная, то инструкция в своем обозначении приобретает параметр "D". Может ли инструкция выполняться как 32-х битная, можно увидеть на основании обзорной таблицы, с которой начинается описание каждой словной инструкции. При обработке словной инструкции со словным операндом нужно следить за тем, чтобы два словных операнда составлялись в 32-х битное слово (двойное слово), но всегда адресуется байт младшего значения (младших 16 бит).

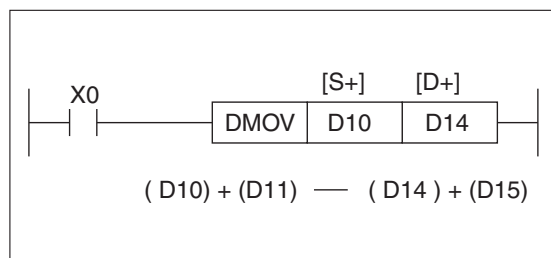


Рис. 6-6.

Пример программирования 32-х битной инструкции DMOV

C000209C

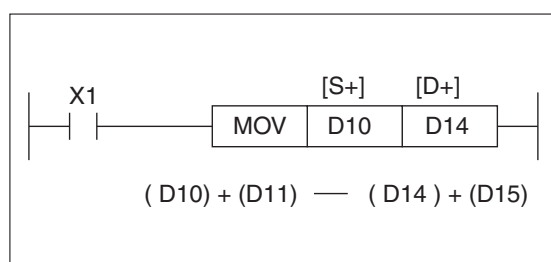


Рис. 6-7.

Пример программирования 16-ти битной инструкции MOV

C000210C

УКАЗАНИЕ

При использовании индексного регистра совместно с 32-х битной инструкцией должен адресоваться только индексный регистр Z.

6.1.10 ОБЗОР СЛОВНЫХ ИНСТРУКЦИЙ

Классификация	Инструкция	FNC (№ функции)	Назначение	Указание раздела	Серия типов ПК				
					FX1S	FX1N	FX2N	FX2NC	FX3U
Инструкции процесса работы программы	CJ	00	Переход внутри программы	6.2.1	●	●	●	●	●
	CALL	01	Вызов подпрограммы	6.2.2	●	●	●	●	●
	SRET	02	Конец подпрограммы	6.2.3	●	●	●	●	●
	IRET	03	Закончить прерывание программы	6.2.4	●	●	●	●	●
	EI	04	Активизировать прерывание программы	6.2.4	●	●	●	●	●
	DI	05	Деактивизировать прерывание программы	6.2.4	●	●	●	●	●
	FEND	06	Конец области программы	6.2.5	●	●	●	●	●
	WDT	07	Обновить значение таймера контроля скана	6.2.6	●	●	●	●	●
	FOR	08	Начало повторения программы	6.2.7	●	●	●	●	●
	NEXT	09	Конец повторения программы	6.2.7	●	●	●	●	●
Инструкции сравнения и передачи	CMP	10	Сравнение числовых данных	6.3.1	●	●	●	●	●
	ZCP	11	Сравнение областей числовых данных	6.3.2	●	●	●	●	●
	MOV	12	Передача данных	6.3.3	●	●	●	●	●
	SMOV	13	Передача со сдвигом	6.3.4			●	●	●
	CML	14	Копирование и инвертирование	6.3.5			●	●	●
	BMOV	15	Передача блока	6.3.6	●	●	●	●	●
	FMOV	16	Передача одинаковых данных	6.3.7			●	●	●
	XCH	17	Обмен данными	6.3.8			●	●	●
	BCD	18	BCD-конвертирование	6.3.9	●	●	●	●	●
	BIN	19	Двоичное конвертирование	6.3.10	●	●	●	●	●
Арифметические инструкции	ADD	20	Суммирование числовых данных	6.4.1	●	●	●	●	●
	SUB	21	Вычитание числовых данных	6.4.2	●	●	●	●	●
	MUL	22	Умножение числовых данных	6.4.3	●	●	●	●	●
	DIV	23	Деление числовых данных	6.4.4	●	●	●	●	●
	INC	24	Приращение	6.4.5	●	●	●	●	●
	DEC	25	Уменьшение	6.4.6	●	●	●	●	●
	AND	26	Логическая И-связь	6.4.7	●	●	●	●	●
	OR	27	Логическая ИЛИ-связь	6.4.8	●	●	●	●	●
	XOR	28	Логическая исключающая ИЛИ связь	6.4.9	●	●	●	●	●
	NEG	29	Инверсия данных	6.4.10			●	●	●
Инструкции сдвигов	ROR	30	Кольцевой сдвиг вправо	6.5.1			●	●	●
	ROL	31	Кольцевой сдвиг влево	6.5.2			●	●	●
	RCR	32	Вращение по битам вправо	6.5.3			●	●	●
	RCL	33	Вращение по битам влево	6.5.4			●	●	●
	SFTR	34	Двоичные данные сдвиг по битам вправо	6.5.5	●	●	●	●	●
	SFTL	35	Двоичные данные сдвиг по битам влево	6.5.5	●	●	●	●	●
	WSFR	36	Слова данных сдвиг вправо	6.5.6			●	●	●
	WSFL	37	Слова данных сдвиг влево	6.5.7			●	●	●
	SFWR	38	Запись в FIFO-память	6.5.8	●	●	●	●	●
	SFRD	39	Чтение из FIFO-памяти	6.5.9	●	●	●	●	●

Табл. 6-7. Обзор словных инструкций (1)

Классификация	Инструкция	FNC (№ функции)	Назначение	Указание раздела	Серия типов ПК				
					FX1S	FX1N	FX2N	FX2NC	FX3U
Операции с данными	ZRST	40	Область операндов отключить	6.6.1	●	●	●	●	●
	DECO	41	Декодирование данных	6.6.2	●	●	●	●	●
	ENCO	42	Кодирование данных	6.6.3	●	●	●	●	●
	SUM	43	Определение включенного бита	6.6.4			●	●	●
	BON	44	Проверка бита	6.6.5			●	●	●
	MEAN	45	Определение среднего значения	6.6.6			●	●	●
	ANS	46	Запуск интервала времени	6.6.7			●	●	●
	ANR	47	Отключение указания бита	6.6.8			●	●	●
	SQR	48	Определение квадратного корня	6.6.9			●	●	●
	FLT	49	Преобразование формата чисел	6.6.10			●	●	●
Высокоскоростные инструкции	REF	50	Обновление входов и выходов	6.7.1	●	●	●	●	●
	REFF	51	Установка входного фильтра	6.7.2			●	●	●
	MTR	52	Чтение матрицы	6.7.3			●	●	●
	DHSCS	53	Включение с помощью высокоскоростного счетчика	6.7.4	●	●	●	●	●
	DHSCR	54	Отключение с помощью высокоскоростного счетчика	6.7.4	●	●	●	●	●
	DHSZ	55	Сравнение областей	6.7.5			●	●	●
	SPD	56	Определение скорости	6.7.6	●	●	●	●	●
	PLSY	57	Выдача определенного числа импульсов	6.7.7	●	●	●	●	●
	PWM	58	Выдача импульсов с модуляцией ширины импульса	6.7.8	●	●	●	●	●
	PLSR	59	Выдача определенного числа импульсов	6.7.9	●	●	●	●	●
Инструкции пользования	IST	60	Инициализация статуса шага	6.8.1	●	●	●	●	●
	SER	61	Инструкция поиска	6.8.2			●	●	●
	ABSD	62	Абсолютный многоуставочный счетчик	6.8.3	●	●	●	●	●
	INCD	63	Инкрементальный многоуставочный счетчик	6.8.4	●	●	●	●	●
	TTMR	64	Обучающийся таймер	6.8.5			●	●	●
	STMR	65	Специальный таймер	6.8.6			●	●	●
	ALT	66	Функция пульс-пары	6.8.7	●	●	●	●	●
	RAMP	67	Функция наклонных характеристик	6.8.8	●	●	●	●	●
	ROTC	68	Позиционирование поворотного стола	6.8.9			●	●	●
	SORT	69	Инструкция сортировки	6.8.10			●	●	●

Табл. 6-8. Обзор словных инструкций (2) (продолжение)

УКАЗАНИЕ

| Словные инструкции FNC 70...98 описаны в главе 7

6.2 ИНСТРУКЦИИ ПРОЦЕССА РАБОТЫ ПРОГРАММЫ

ОБЗОР ИНСТРУКЦИЙ FNC 00...09

Символ	FNC	Назначение	Раздел
CJ	00	Переход внутри программы	6.2.1
CALL	01	Вызов подпрограммы	6.2.2
SRET	02	Конец подпрограммы	6.2.3
IRET	03	Закончить прерывание программы	6.2.4
EI	04	Активизировать прерывание программы	6.2.4
DI	05	Деактивизировать прерывание программы	6.2.4
FEND	06	Конец области программы	6.2.5
WDT	07	Обновить значение таймера контроля скана	6.2.6
FOR	08	Начало повторения программы	6.2.7
NEXT	09	Конец повторения программы	6.2.7

Табл. 6-9. Обзор инструкций FNC 00...09

6.2.1 ПЕРЕХОД ВНУТРИ ПРОГРАММЫ (CJ)

	CJ		FNC 00			
	Переход внутри программы					
	CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●
Операнды	Имп. инструкция (P)	Обработка		Шаги программы		
Точки P0...P63 (индексирование-отражение допустимо) P63 означает переход к концу инструкции.	●	16бита	32бита	CJ / CJP	3	
		●		Точка P**	1	

ФУНКЦИЯ

С помощью CJ-инструкции может пропускаться часть программы. При применении этой инструкции время программы может уменьшаться.

ОПИСАНИЕ

- Цель (конец) перехода определяется установкой маркировки (маркировка точки) в программе.
- Указание адреса конца перехода (Адреса точки) определяет, к какой маркировке точки должен выполняться переход.
- Если внутри подпрограммы перехода программируется инструкция сброса (отключения) для счетчика с запоминанием, то процесс сброса (стирание накопленного значения) имеет место тогда, когда перепрыгивается цепь схемы катушки счетчика.
- Имеется возможность дублирования записи выхода.

УКАЗАНИЕ

При дублировании записи выходов следите за тем, чтобы оба выхода никогда не были активными в одно и тоже время. Это может привести к ошибочной отработке программы.

ВЫПОЛНЕНИЕ МАРКИРОВКИ ТОЧКИ В ПРОГРАММЕ

- Маркировка точки выполняется при программировании на языке AWL (Листинг инструкций) непосредственно перед цепью схемы (перед инструкциями LD и LD1).
- При программировании на языке контактной схемы маркировка точки указывается слева от цепи схемы.

ПРИМЕР ▾

Применение CJ-инструкции

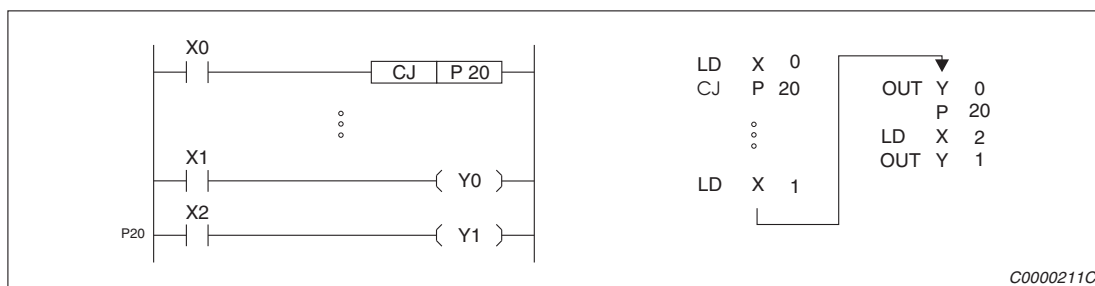


Рис. 6-8. Пример программирования CJ-инструкции

Если включается X0, то выполняется переход к маркировке точки P20.



ДВУХРАЗОВАЯ ВСТАВКА В ПРОГРАММЕ АДРЕСА ЦЕЛИ ПЕРЕХОДА (АДРЕСА ТОЧКИ)

ПРИМЕР ▾

Двухразовая вставка в программу адреса точки P9

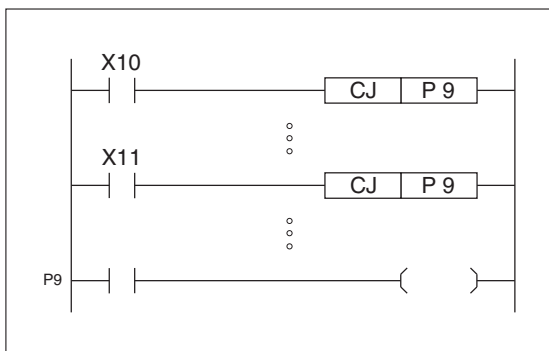


Рис. 6-9.
Пример программирования для двухразовой вставки в программу адреса точки P9

C000212C

Если X10 включен, то выполняется переход к промаркированной точке P9. Если X10 выключен, а X11 включен, то все равно произойдет переход к точке P9. ▴

УКАЗАНИЕ

Однако одинаковая маркировка точек не должна многократно использоваться в программе. Может создаться ошибка в работе программы.

УСТАНОВКА МАРКИРОВКИ ТОЧКИ ПЕРЕД CJ-ИНСТРУКЦИЕЙ ПЕРЕХОДА

Обратный переход (вверх программы) также может выполняться внутри программы.

УКАЗАНИЕ

Если входной сигнал для CJ-инструкции держится больше 200 мс, то появляется ошибка времени работы (Watch-Dog-Timer).

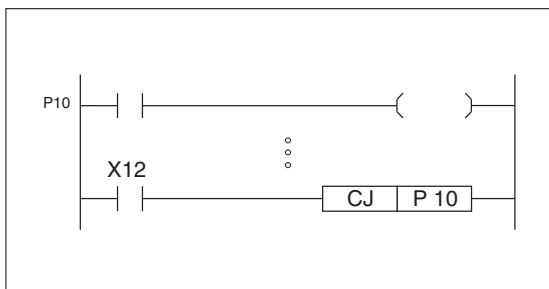


Рис. 6-10.
Пример программирования по установке точки маркировки перед CJ-инструкцией

C000213C

ПЕРЕХОДЫ В ОБЛАСТИ ГЛАВНОГО УПРАВЛЕНИЯ (Master-Control)

Процесс программирования при использовании CJ-инструкции вместе с MC- и MCR-инструкциями показан на следующем примере:

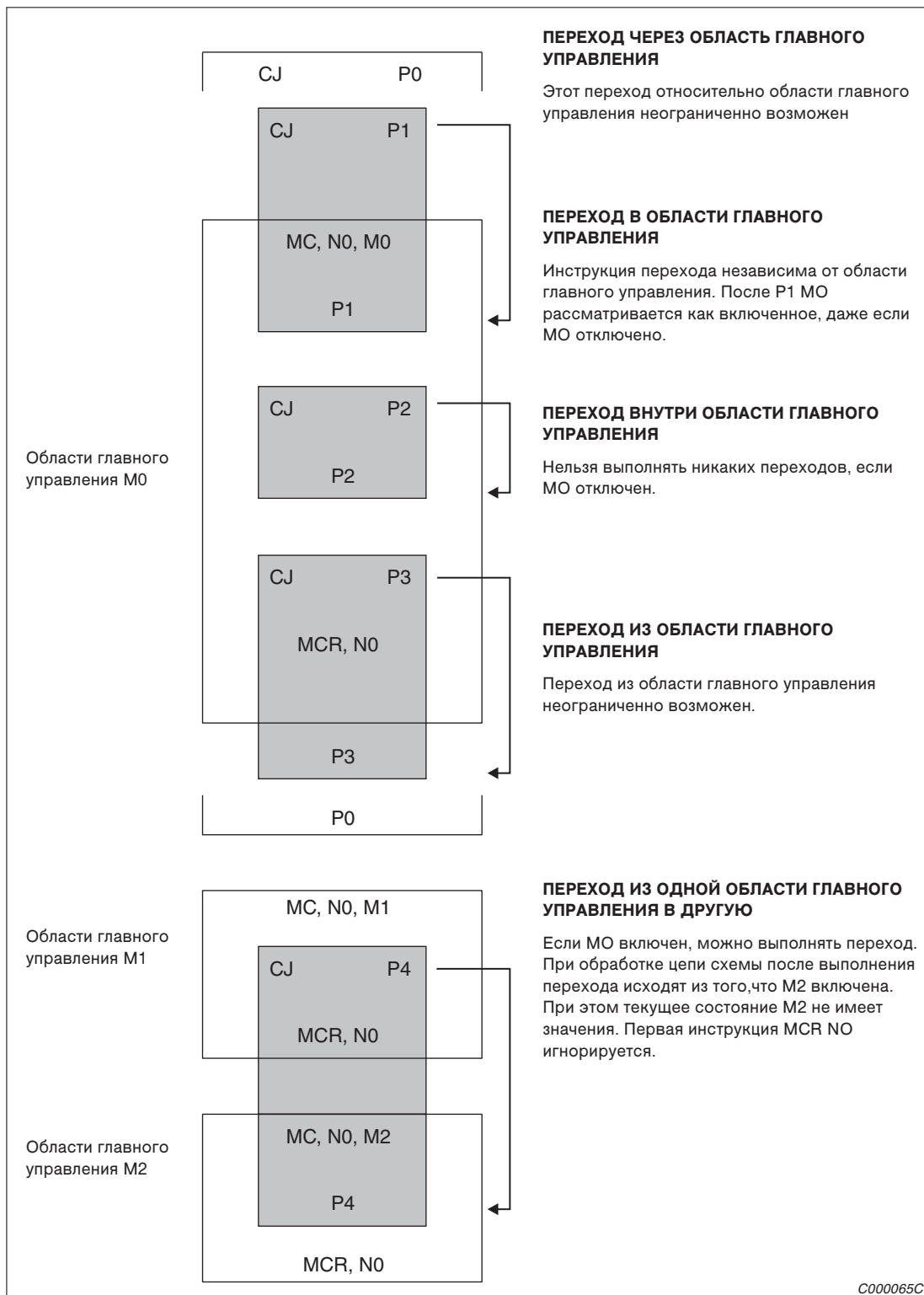


Рис. 6-11. Примеры переходов в области главного управления

СОСТОЯНИЕ КОНТАКТОВ И КАТУШЕК ВНУТРИ ПРОПУЩЕННОЙ ЧАСТИ ПРОГРАММЫ

Операнды	Состояние контактов и катушек перед переходом	Состояние контактов и катушек после перехода	Примечания
Выходы Y	ВКЛ	ВКЛ	—
Меркеры M	ВКЛ	ВКЛ	—
Состояния шагов S	ВКЛ	ВКЛ	—
Таймеры T	ВКЛ	ВКЛ	Отсчет времени остановлен. Накопленное значение времени сохраняется. Если условия перехода больше не выполняются, то продолжается отсчет времени.
Счетчики C	ВКЛ	ВКЛ	Счет остановлен. Накопленное значение счета сохраняется. Если условия перехода больше не выполняются, то продолжается счет сигналов.
Словные инструкции	—	—	Выполнение остановлено. Инструкции, как например, RAMP, INC или DEC сохраняют однако их текущее значение данных.

Табл. 6-10. Состояние контактов и катушек при отработке инструкции перехода

6.2.2 ВЫЗОВ ПОДПРОГРАММ (CALL)

		CALL		FNC 01			
		Вызов подпрограммы					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D	Имп. инструкция (P)		Обработка		Шаги программы	
	Точки P0...P62 (индицирование-отражение допустимо)	●	16-бита	32-бита	CALL/CALLP	3	
			●		Точка P**	1	

ФУНКЦИЯ

С помощью CALL-инструкции вызывается подпрограмма

ОПИСАНИЕ

- Подпрограмма маркируется с помощью точек (P0...P62) и вызывается CALL-инструкцией.
- В конце подпрограммы должна находиться SRET-инструкция.
- Подпрограмма программируется после FEND-инструкции и перед END-инструкцией.
- Если активируется CALL-инструкция, то выполняется переход к указанной точке маркировки. После отработки SRET-инструкции выполняется обратный переход к инструкции, переходящей на CALL-инструкцию.
- Активированные в подпрограмме операнды остаются активированными после отработки подпрограммы до новой обработки подпрограммы.
- В подпрограмме должны использоваться таймеры T192...T199 и T246...T249.
- Те же точки могут использоваться с любым числом CALL-инструкций.

УКАЗАНИЕ

Внутри подпрограммы могут вызываться другие подпрограммы. Возможно максимум 4 уровня разветвлений.

6.2.3 КОНЕЦ ПОДПРОГРАММЫ (SRET)

		SRET					FNC 02						
		Конец подпрограммы											
Операнды		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	Имп. инструкция (P)		Обработка		Шаги программы	
		●	●	●	●	●		●	16бита	32бита	SRET	1	

ФУНКЦИЯ

С помощью SRET-инструкции определяется конец подпрограммы.

ОПИСАНИЕ

- Подпрограмма помечается указателем (Pn) и вызывается командой CALL.
- В конце подпрограммы должна стоять SRET-инструкция.
- Подпрограмма программируется после FEND-инструкции и перед END-инструкцией.
- После обработки SRET-инструкции осуществляется переход обратно к инструкции следующей за CALL-инструкцией.

УКАЗАНИЕ

| SRET-инструкция может программироваться только вместе с CALL-инструкцией.

ПРИМЕР ▾

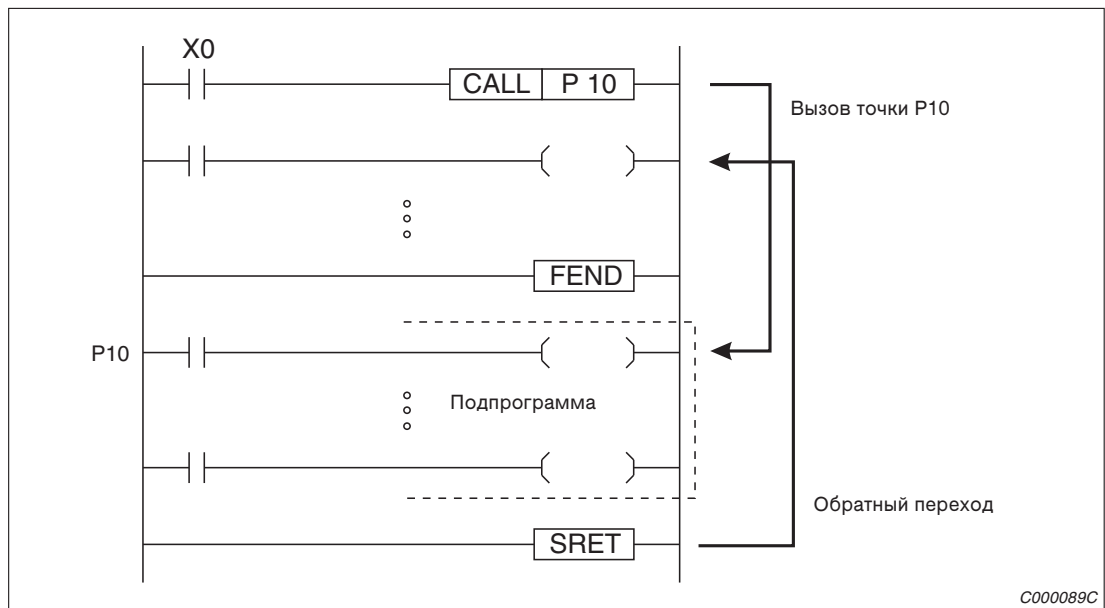


Рис. 6-12. Пример программирования с применением CALL- и SRET- инструкций



6.2.4 ПРИМЕНЕНИЕ ПРЕРЫВАНИЯ ПРОГРАММЫ (IRET, EI, DI)

		IRET		FNC 03			
		Окончание программы прерывания					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D	Имп. инструкция (P)		Обработка		Шаги программы	
	—		16бита	32бита	IRET	1	Точка I***

		EI		FNC 04			
		Импульсный запуск/активизация программы прерывания					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D	Имп. инструкция (P)		Обработка		Шаги программы	
	—		16бита	32бита	EI	1	Точка I***

		DI		FNC 05			
		Деактивизация программы прерывания					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D	Имп. инструкция (P)		Обработка		Шаги программы	
	—		16бита	32бита	DI	1	Точка I***

ПРИНЦИП ФУНКЦИОНИРОВАНИЯ

Вызов, деблокировка и блокировка программ прерывания

ВЫЗОВ ПРОГРАММЫ ПРЕРЫВАНИЯ

- При вызове программы прерывания оставляется программа ПК и выполняется переход к программе прерывания. После окончания программы прерывания выполняется возврат к программе ПК.
- Начало программы прерывания определяется установкой маркировки (точки прерывания).
- Конец программы прерывания определяется IRET-инструкцией.
- Входы X0...X5 образуют входы прерывания.
- Сигналы прерывания должны иметь ширину импульса минимум в 200 мкс.
- Программа прерывания должна программироваться в конце программы ПК за последней FEND-инструкцией и перед END-инструкцией.

УКАЗАНИЕ

В контроллерах FX2N, FX2NC и FX3U возможны также прерывания по таймеру и счетчику.

Входы X0...X5 не могут применяться одновременно для обработки сигналов прерывания и для обработки сигналов высокоскоростного счетчика.

АДРЕСАЦИЯ ТОЧЕК ПРЕРЫВАНИЯ

- Адресация точек прерывания должна выполняться следующим образом:

Точка прерывания: I ◆ 0 ◇

◆: адрес 0...5, соответствующий входам X0...X5

◇: 0:= прерыванию при падающем фронте входного сигнала

1:= прерыванию при возрастающем фронте входного сигнала

ПРЕРЫВАНИЕ ПО ВРЕМЕНИ (только для FX2N, FX2NC и FX3U)

- Программа прерывания обрабатывается соответственно после завершения предварительно заданного времени.

Точка прерывания: I ◆◇◇

◇◇: время интервала 10...99 мс

◆: адрес 6...8 (T6...T8)

ПРЕРЫВАНИЕ ПО СЧЕТУ (только для FX2N, FX2NC и FX3U)

- Программа прерывания обрабатывается после достижения предварительно заданного значения счета.

Точка прерывания: I 0 ◇ 0

◇: номер прерывания 1...6

УКАЗАНИЕ

Адрес прерывания может использоваться только один раз

ПРИМЕНЕНИЕ EI- И DI-ИНСТРУКЦИЙ

- С помощью EI-инструкции могут активироваться инструкции прерывания. Это означает, что после отработки EI-инструкции, смена сигнала, которая появляется на одном из входов X0...X5, обрабатывается как сигнал прерывания в программе.
- С помощью DI-инструкции могут деактивироваться инструкции прерывания. Это означает, что после отработки DI-инструкции, смена сигнала, которая появляется на одном из входов X0...X5, не обрабатывается больше как сигнал прерывания в программе.

УКАЗАНИЕ

Если ни одна из обеих инструкций EI или DI не программируется, режим прерывания не активизируется, т.е. тогда не может обрабатываться никакой сигнал прерывания.

ОТРАБОТКА ПРОГРАММЫ ПРЕРЫВАНИЯ

- Программа прерывания, которая вызывается в области между DI- и Во время исполнения программы прерывания не может вызываться никакая другая программа прерывания. Однако может программироваться два уровня разветвления.
- Несколько, одна за другой следующие, программы прерывания обрабатываются в последовательности их вызова.
- Если одновременно вызываются несколько программ прерывания, то вначале обрабатывается программа прерывания с более низким адресом точки.

ВЫКЛЮЧЕНИЕ ЛЮБОГО ПРЕРЫВАНИЯ

- Любое прерывание может временно или постоянно выключаться посредством включения соответствующего специального меркера. Соответствующий специальный меркер указан в главе 6. Для всех ПК первым специальным меркером является M8050, который выключает прерывание I0①②.

УКАЗАНИЕ

Никогда не вводите специального меркера, не зная точно его функции. Не все ПК всегда работают с одними и теми же меркерами.

Прерывание высокоскоростного счетчика всегда может выключаться только как отдельная группа специальным меркером M8059.

Может программироваться максимум два уровня разветвления.

Программа прерывания не выполняется, если активирован относящийся к ней специальный меркер. Так, не выполняется программа прерывания I □ **, если активированы специальные меркеры M805□ (□: 1, 2, 3, 4, 5).

ЗАПОМИНАНИЕ СМЕНЫ СИГНАЛА ВХОДОВ ПРЕРЫВАНИЯ

Эта функция (импульсный запуск функции - Pulse-Catch-Funrtion) позволяет запоминать смену сигнала на входах прерывания X0...X5 в специальных меркерах M8170...M8175. Эта функция запоминания может выполняться одновременно только один раз для одного входа. Функция импульсного запуска активизируется с помощью EI-инструкции.

ПРИМЕР ▾

Адресация точки прерывания

Точка: I001

Пояснение: вход прерывания X0, прерывание при возрастающем фронте входного сигнала (смена сигнала с "0" на "1") △

ПРИМЕР ▾

Применение инструкций EI, DI и IRET

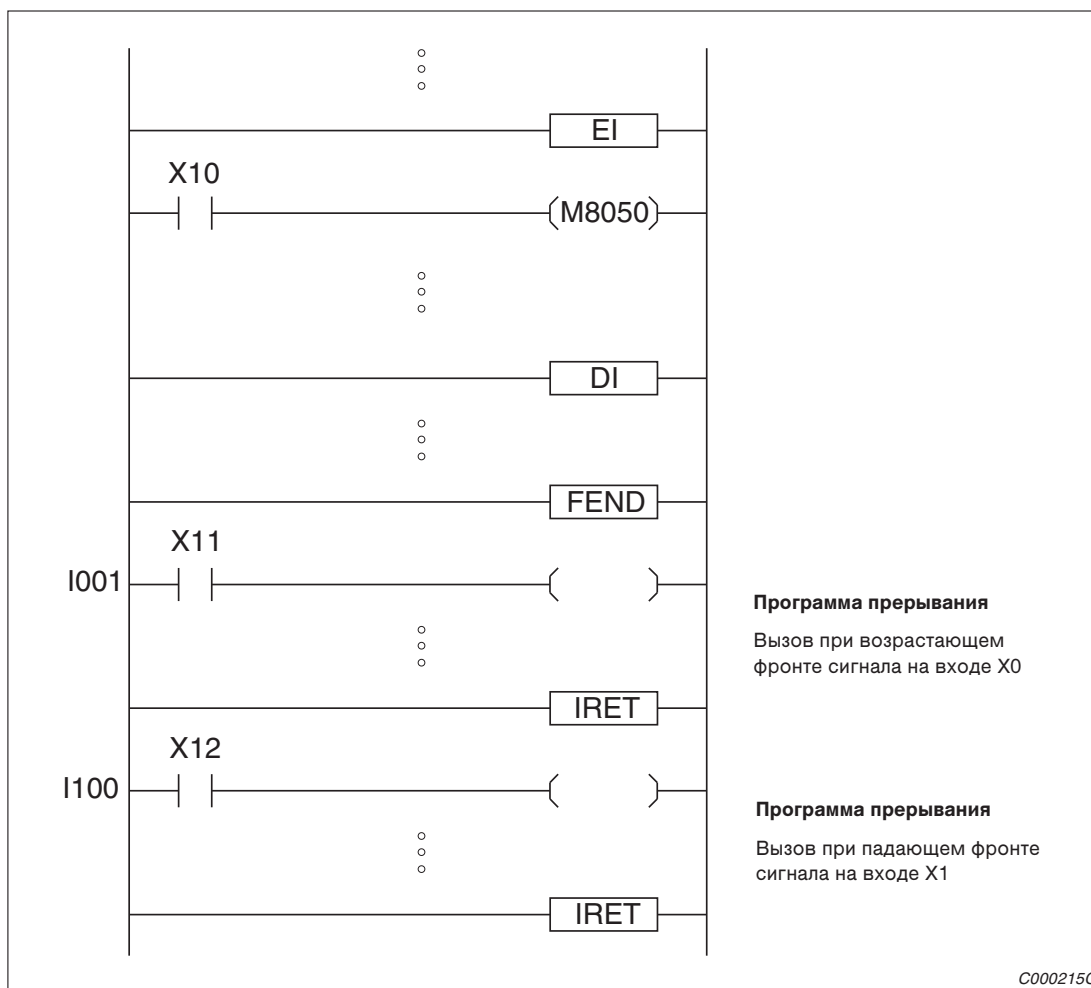


Рис. 6-13. Пример программирования при использовании инструкций EI, DI и IRET

- Если вход X0 устанавливает сигнал прерывания во время выполнения шага программы внутри области от EI-инструкции до DI-инструкции, то имеет место переход к программе прерывания I001. Программа прерывания выполняется и происходит возврат в программу ПК.
- Программа прерывания I001 не выполняется, если активизирован специальный меркер M8050 (вход X10 включен).
- Если вход X1 устанавливает сигнал прерывания во время выполнения шага программы внутри области от EI-инструкции до DI-инструкции, то имеет место переход к программе прерывания I100. Программа прерывания выполняется и происходит возврат в программу ПК.
- Если появляются одновременно сигналы X0 и X1, то вначале обрабатывается программа прерывания I001, а затем программа прерывания I100.



6.2.5 КОНЕЦ ОБЛАСТИ ПОДПРОГРАММЫ (FEND)

		FEND		FNC 06		
		Конец области программы				
Операнды	D	Имп. инструкция (P)	Обработка		Шаги программы	
			16бита	32бита	FEND	1
	—					

ФУНКЦИЯ

Окончание отдельной области подпрограмм внутри программы ПК

ОПИСАНИЕ

- С помощью FEND-инструкции определяется конец области программы. Можно применять несколько FEND-инструкций внутри программы ПК.
- После отработки FEND-инструкции выполняется обработка выходов. Затем выполняется возврат к программному шагу 0. Обновляется обработка входов и время уставки контроля цикла программы.

УКАЗАНИЕ

Программируйте программу прерывания между FEND-инструкцией и END-инструкцией.

Не путайте FEND-инструкцию с END-инструкцией. С помощью END-инструкции завершается вся программа ПК.

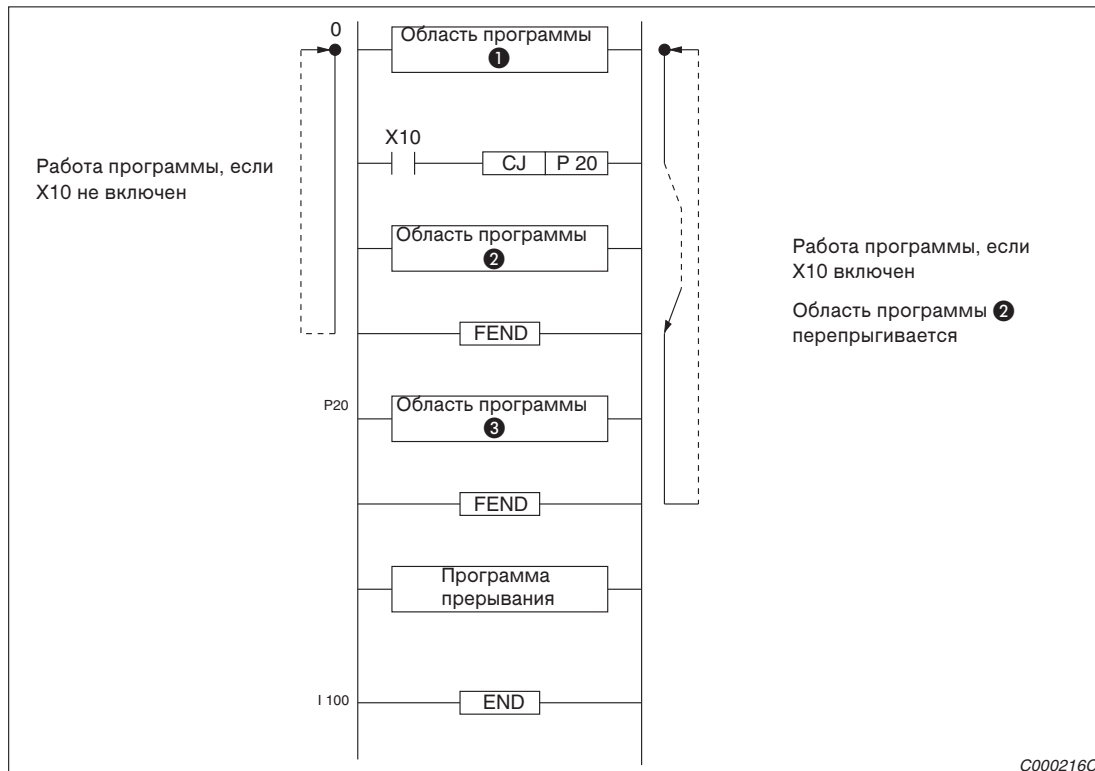


Рис. 6-14. Пример программирования при использовании FEND-инструкции

△

6.2.6 ОБНОВЛЕНИЕ КОНТРОЛЯ ВРЕМЕНИ ЦИКЛА (WDT)

		WDT		FNC 07			
		Обновление контроля времени цикла					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D	Имп. инструкция (P)		Обработка		Шаги программы	
	—	●	16бита	32бита	WDT	1	
			●		WDTP	1	

ФУНКЦИЯ

С помощью WDT-инструкции можно длинные программы разделить на отдельные отрезки программ. Время цикла программы (скана) определяется для каждого отдельного отрезка программы самим ПК (WDT обновляется после каждого отрезка программы). С помощью WDT-инструкции можно обрабатывать программу, время цикла которой превышает 200 мс.

ОПИСАНИЕ

- WDT-инструкция должна применяться, если время цикла программы от 0-го шага программы до END- или FEND-инструкций превышает значение **200 мс**.
- И далее WDT-инструкция может программироваться после точки маркировки, если она находится в программе перед относящейся к ней инструкции перехода (CJ-инструкции, см.раздел 6.2.1)
- WDT-инструкция может применяться также внутри FOR-NEXT-петли (см.раздел 6.2.7).

УКАЗАНИЕ

Время контроля цикла обновляется при каждом выполнении инструкций END, FEND или WDT.

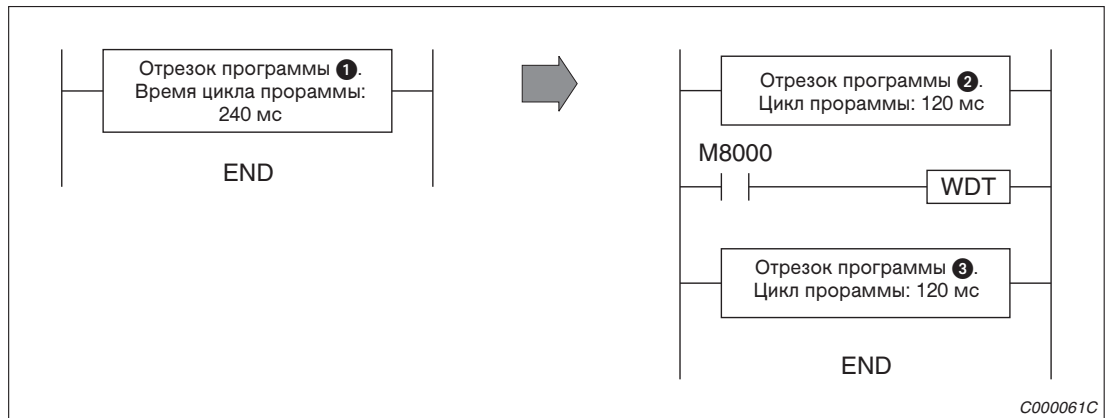
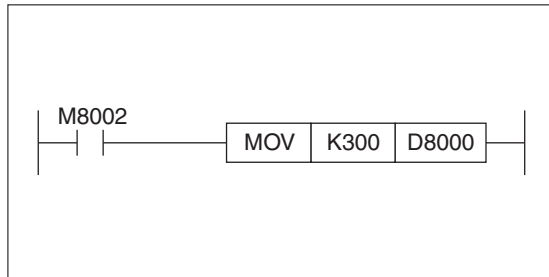


Рис. 6-15. Пример программирования при использовании WDT-инструкции

Время обработки для отрезка программы 1 превысил значение 200 мс. Поэтому отрезок программы 1 был разделен благодаря WDT-инструкции на два отрезка программ (2, 3). Отрезки программ 2 и 3 требуют соответственно только по 120 мс времени цикла. △

ЗНАЧЕНИЕ ВРЕМЕНИ ЦИКЛА ПРОГРАММЫ ИЗМЕНЯЕТСЯ В СПЕЦИАЛЬНОМ РЕГИСТРЕ D8000

Если время цикла программы постоянно превышает значение 200 мс, можно изменить значение максимально допустимого времени цикла в специальном регистре D8000.

**Рис. 6-16.**

Пример программирования для установки максимально допустимого времени цикла программы в регистре данных D8000 на значение 300 мс.

C000070C

6.2.7 ПОВТОРЕНИЕ ЧАСТЕЙ ПРОГРАММЫ (FOR, NEXT)

		FOR		FNC 08				
		Начало повторения программы						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●	●	
Операнды	S	Имп. инструкция (P)	Обработка		Шаги программы			
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		16бита	32бита	FOR	3		
			●					

		NEXT		FNC 09				
		Конец повторения программы						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●	●	
Операнды	S	Имп. инструкция (P)	Обработка		Шаги программы			
	—		16бита	32бита	NEXT	1		
			●					

ФУНКЦИЯ

Программирование повторений программы (петля программы)

ОПИСАНИЕ

- Часть программы между FOR- и NEXT-инструкциями повторяется “n” раз. После завершения FOR- выполняется шаг программы после NEXT-инструкции.
- Значение “n” может находиться внутри следующей области: “n”: от +1 до +32 767. Если для “n” указано значение между 0 и -32 767, то петля FOR-NEXT обрабатывается только один раз.
- Можно программировать до пяти FOR-NEXT-уровней разветвления.

УКАЗАНИЕ

FOR- и NEXT-инструкции могут программироваться только попарно. К каждой инструкции FOR должна программироваться соответственно NEXT-инструкция.

ИСТОЧНИКИ ОШИБОК

В следующих случаях появляются ошибки в работе программы:

- NEXT-инструкция запрограммирована перед FOR-инструкцией.
- NEXT-инструкция запрограммирована после FEND-инструкции или END-инструкции.
- Количество NEXT-инструкций не соответствует количеству FOR-инструкций.

ПРИМЕР ▾

| Применение FOR- и NEXT-инструкций

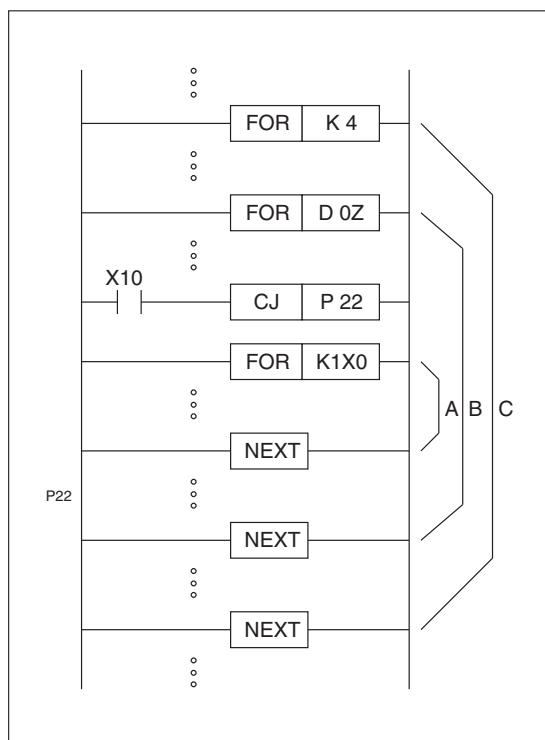


Рис. 6-17.

Пример программирования при использовании FOR- и NEXT-инструкций

C000018C

В примере запрограммированы три входящие друг в друга FOR- и NEXT-уровня разветвления.

- Отрезок программы С обрабатывается четыре раза (здесь K4 константа 4). В конце обработки последний программный шаг выполняется после третьей NEXT-инструкции.
- При каждом исполнении отрезка С отрезок программы В обрабатывается шесть раз, если в регистре данных D0Z записано число 6.
- Поэтому отрезок В обрабатывается $6 \times 4 = 24$ раза.
- Если вход X10 включен, то FOR-NEXT-петля (отрезок программы) пропускается (не обрабатывается) с помощью CJ-инструкции.
- Если вход X10 выключен и содержание K1X0 (блок K1 - первых 4 бита - в слове X0) равно 7, то при каждом выполнении отрезка В отрезок программы обрабатывается семь раз.
- Поэтому отрезок А обрабатывается $168 (6 \times 4 \times 7)$ раз.

△

6.3 ИНСТРУКЦИИ СРАВНЕНИЯ И ПЕРЕДАЧИ

ОБЗОР ИНСТРУКЦИЙ FNC 10...19

Символ	FNC	Назначение	Раздел
CMP	10	Сравнение числовых данных	6.3.1
ZCP	11	Сравнение числовых областей данных	6.3.2
MOV	12	Передача данных	6.3.3
SMOV	13	Передача и сдвиг данных	6.3.4
CML	14	Копирование и инвертирование	6.3.5
BMOV	15	Передача блоков	6.3.6
FMOV	16	Передача одинаковых данных	6.3.7
XCH	17	Обмен данными	6.3.8
BOD	18	BOD-конвертирование	6.3.9
BIN	19	BIN-конвертирование	6.3.10

Табл. 6-11. Обзор инструкций FNC 10...19

6.3.1 СРАВНЕНИЕ ЧИСЛОВЫХ ДАННЫХ (CMP, DCMP)

		CMP			FNC 10		
		Сравнение числовых данных					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S+, S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y, M, S	●	16бита	32бита	CMP/CMPP	7
				●	●	DCMP/DCMPP	13

ФУНКЦИЯ

Сравнение между двумя числовыми значениями данных (больше, меньше, равно)

ОПИСАНИЕ

- Данные в обоих источниках (S1+) и (S2+) сравниваются друг с другом.
- Результат сравнения (больше, меньше, равно) отображается (индицируется) благодаря задействованию меркера M, операнда состояния шага S или выхода Y. Определение, какой из этих операндов должен задействоваться, выполняется по адресу цели (D+).

$(S1+) > (S2+) \rightarrow (D+)$

$(S1+) = (S2+) \rightarrow ((D+)+1)$

$(S1+) < (S2+) \rightarrow ((D+)+2)$

- Данные в S1+ и S2+ обрабатываются как двоичные данные.

ИСТОЧНИКИ ОШИБОК

- CMP-инструкция требует записи трех операндов. Если указывается меньшее число операндов, появляется ошибка - код 6503. Выполнение инструкции прерывается.
- Если программируется недопустимый операнд, выдается ошибка по коду 6705. В качестве адреса цели не могут, например, использоваться входы X, регистры данных D, таймеры или счетчики (см.операнд D+ в структурной таблице).
- Если указанный операнд превышает область, допустимую для него, то выдается ошибка по коду 6706. Ошибка такого рода появляется, например, при очень большом значении числа вместе с адресацией индекса.

Детальное описание кодов ошибок содержится в главе 11.

ПРИМЕР ▾

Применение CMP-инструкции

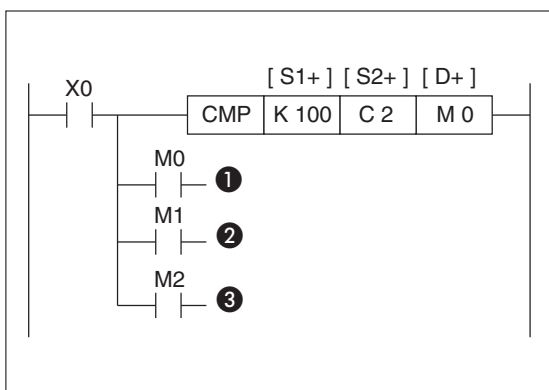


Рис. 6-18.
Пример программирования с использованием CMP-инструкции

C000071C

В адресе цели (D+) в этом примере указан меркер M0. Соответствующие результаты сравнения автоматически присваиваются приращиваемым на 1 последующих адресов меркеров M0, M1, M2 и имеют следующие значения:

- ① M0: включен, если $K100 >$ накопленного в счетчике C2 значения
 - ② M1: включен, если $K100 =$ накопленному в счетчике C2 значению
 - ③ M2: включен, если $K100 <$ накопленного в счетчике C2 значения
- M0, M1, M2 не изменяются, если входное условие X0 выключено.

△

6.3.2 СРАВНЕНИЕ С БЛОКОМ ЧИСЛОВЫХ ДАННЫХ (ZCP, DZCP)

		ZCP		FNC 11			
		Сравнение с блоком числовых данных					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S1+, S2+, S3+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y, M, S	●	16бита	32бита	ZCP/ZCPP	9
				●	●	DZCP/DZCPP	17

ФУНКЦИЯ

Сравнение числовых значений данных с числовыми областями данных (больше, меньше, равно)

ОПИСАНИЕ

- Данные в источнике (S3+) сравниваются с данными обоих источников (S1+) и (S2+).
- Результат сравнения (больше, меньше, равно) отображается (индицируется) благодаря задействованию меркера M, операнда состояния шага S или выхода Y. Определение, какой из этих операндов должен задействоваться, выполняется в регистре данных (по адресу цели) - (D+).

$$(S1+) > (S3+) \rightarrow (D+)$$

$$(S1+) < (S3+) < (S2+) \rightarrow ((D+)+1)$$

$$(S2+) < (S3+) \rightarrow ((D+)+2)$$

- Данные в (S1+) не должны быть больше, чем данные в (S2+).

ПРИМЕР ▾

Если в (S1+) находится значение K100, а в (S2+) - K90, то при исполнении ZCP-инструкции нужно исходить из того, что в (S2+) также находится числ K100.

△

ПРИМЕР ▾

Применение ZCP-инструкции

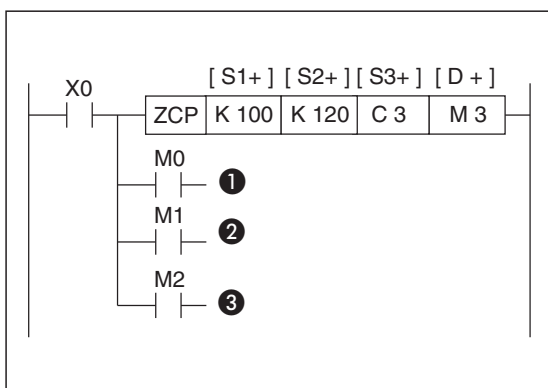


Рис. 6-19.

Пример программирования с использованием ZCP-инструкции

C000072C

В адресе цели (D+) в этом примере указан меркер M3. Соответствующие результаты сравнения автоматически присваиваются приращиваемым на 1 последующим адресам меркеров M3, M4, M5 и имеют следующие значения:

- ① M3: включен, если $K100 >$ накопленного в счетчике C3 значения
- ② M4: включен, если $K120 \leq$ накопленного в счетчике C3 значения
- ③ M5: включен, если накопленное в счетчике C3 значение $>$ K120

M3, M4, M5 не изменяются, если входное условие X0 выключено.

Если накопленное значение счетчика C3 находится в области от 100 до 120, то меркер M4 включен.

△

6.3.3 ПЕРЕДАЧА ДАННЫХ (MOV, DMOV)

		MOV		FNC 12				
		Передача данных						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●	●	
Операнды	S+	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	MOV/MOVP	5	
				●	●	DMOV/DMOVP	9	

ФУНКЦИЯ

Передача данных от источника данных (S+) к данным цели (D+)

ОПИСАНИЕ

- Инструкция служит для передачи данных от источника данных (S+) к данным цели (D+).
- Данные в источнике данных (S+) при выполнении MOV-инструкции автоматически интерпретируются как двоичные значения.

УКАЗАНИЕ

Инструкции выполняются в каждом цикле программы. Этого можно избежать благодаря использованию вставленной впереди импульсной функции (PLS- или PLF-инструкции или же параметра "P").

ПРИМЕР ▾

Применение MOV-инструкции

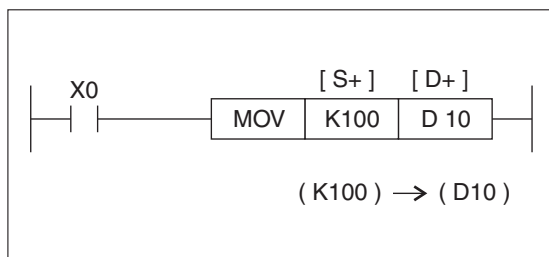


Рис. 6-20.

Пример программирования с использованием MOV-инструкции

C000073C

Если входное условие X0 включено, то выполняется переход данных от (S+) к (D+). Если X0 выключен, не выполняется никакой переход.

Константа K100 при выполнении MOV-инструкции автоматически интерпретируется двоичным значением. △

6.3.4 ПЕРЕДАЧА И СМЕЩЕНИЕ ДАННЫХ (SMOV)

				SMOV		FNC 13					
				Передача и смещение данных							
Операнды				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
							●	●	●		
S+		D+		n, m1, m2		Имп. инструкция (P)		Обработка		Шаги программы	
K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z 0 - 9999		K, H, KnY, KnM, KnS, T, C, D, V, Z		K, H 0 - 4		●		16бита	32бита	SMOV/ SMOVP	11
								●			

ФУНКЦИЯ

Передача данных и изменение их значимости

ОПИСАНИЕ

- Инструкция обрабатывается в 5-ти логических шагах:
 - 1.) Чтение двоичных данных из (S+) (исходных данных)
 - 2.) Преобразование данных в BCD-формат
 - 3.) Сдвиг BCD-мест
 - 4.) Преобразование данных в BIN-формат
 - 5.) Запись двоичных данных в (D+) (цель данных)
- n, m1, m2 определяют вид (параметры) сдвига BCD-мест.
 - m1 = 1. первое место (в слове), которое должно быть смещено
 - m2 = число мест, которые должны быть смещены
 - n = 1. первый адрес в цели (место BCD, куда смещать)

ПРИМЕР ▾

SMOV-инструкции без специального меркера

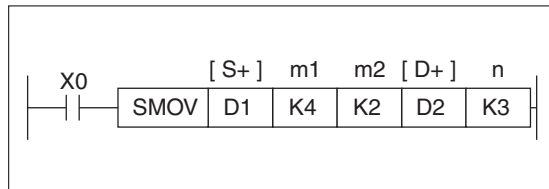


Рис. 6-21.

Пример программирования с использованием SMOV-инструкции без специального меркера

C000125C

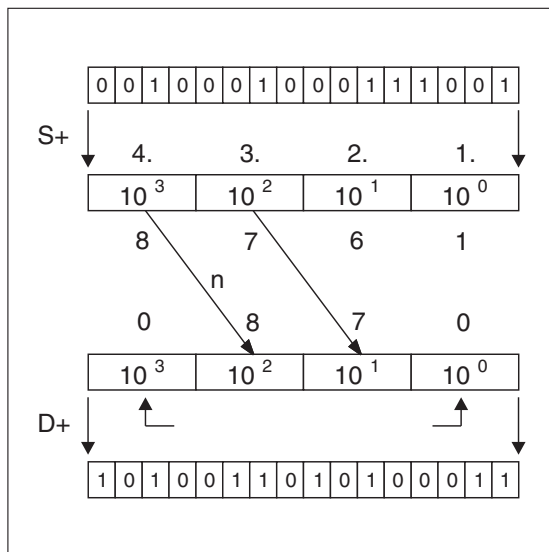


Рис. 6-22.

Конвертирование и передача

C000128C

ПРИМЕР ▾

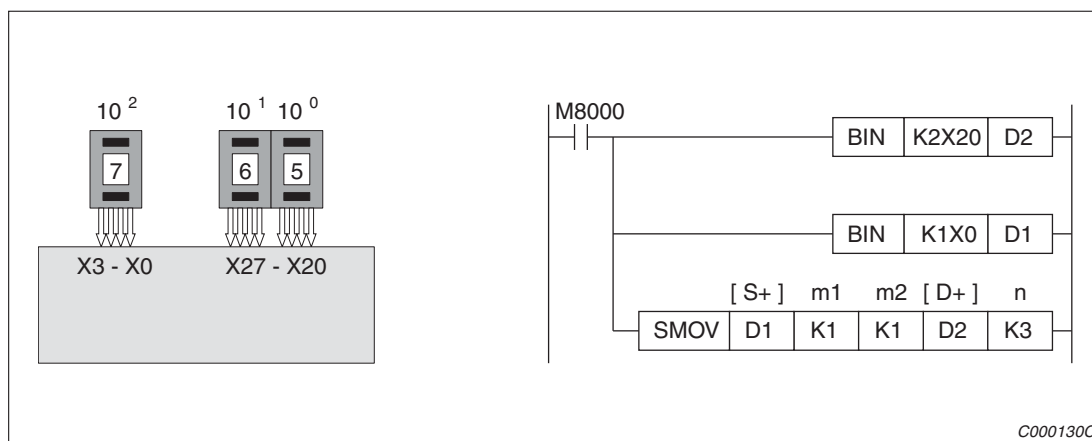


Рис. 6-23. Пример программирования для двоичных данных

Входные данные поставляются от трех BCD-переключателей, распределенных по соответствующим входам ПК.

Вначале конвертируются BCD-данные входов X20...X27 (K2X20 - две тетрады (8 бит) K2 начиная с бита X20) в двоичные данные и заносятся в регистр данных D2.

Аналогично преобразовываются в двоичные данные BCD-данные входов X0...X3 (K1X0).

BCD-значение регистра данных D1 (первая тетрада m1 = K1 и одна перемещаемая тетрада - m2 = K1) записываются на третью позицию регистра цели D2. В заключение BCD-данные снова преобразовываются в двоичные данные.

В примере числовые входные данные снимаются с трех BCD-переключателей и запоминаются как двоичные данные в регистре данных D2.

△

ПРИНЦИП ФУНКЦИОНИРОВАНИЯ СО СПЕЦИАЛЬНЫМ МЕРКЕРОМ M8168

Передача данных в шестнадцатиричном формате и изменение их значимости

ОПИСАНИЕ

- SMOV-инструкция должна комбинироваться с OUT-инструкцией.
- Инструкция обрабатывается в 3-х логических шагах:
 - 1.) Чтение шестнадцатиричных данных из (S+), макс.4 разряда, макс. FFFF_h
 - 2.) Сдвиг BCD-мест
 - 3.) Запись данных в (D+)
- n, m1, m2 определяют вид (параметры) сдвига мест.
 - m1 = 1. первое место (в слове), которое должно быть смещено
 - m2 = число мест, которые должны быть смещены
 - n = 1. первый адрес в цели (куда смещать)

ПРИМЕР ▾

Применение SMOV-инструкции со специальным меркером

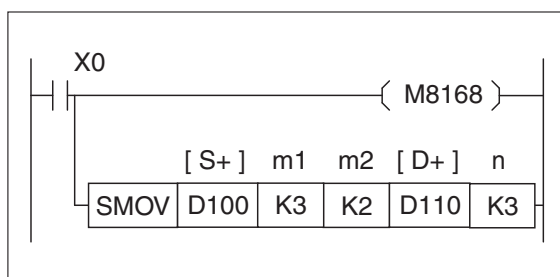


Рис. 6-24.
Пример программирования с использованием SMOV-инструкции

C000312C

В регистр данных D100 записано число FFE2_h, в регистр данных D110 записано число 2CD9_h. Если X0 включен SMOV-инструкция выполняется.

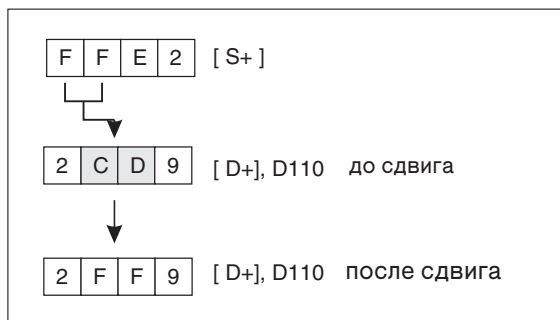


Рис. 6-25.
Сдвиг в регистре данных (D110 перед выполнением / D110 после выполнения)

C000313C

Цифры FF_h, два разряда (m2) начиная с третьего (m1), копируются и смещаются в D110. Тем самым цифры CD_h в D110 переписываются на цифры FF_h.



6.3.5 ПЕРЕСЫЛКА С ИНВЕРТИРОВАНИЕ (CML)

		CML		FNC 14						
		Копирование и инвертирование								
Операнды		S+		D+		Имп. инструкция (P)	Обработка		Шаги программы	
		K, H, KnX, KnY, KnM, KnS, T, C, D, V Z		KnY, KnM, KnS, T, C, D, V, Z			●	●	●	●
						●	16бита	32бита	CML, CMLP	5
						●	●	●	DCML, DCMLP	9

ФУНКЦИЯ

Образование 1-го дополнения двоичного числа

ОПИСАНИЕ

Двоичное значение числа в (S+) преобразовывается в свое 1-ое дополнение и записывается в данные цели (D+).

УКАЗАНИЕ

Если адрес цели располагает большим числом бит, чем адрес источника, то все неиспользуемые биты включаются.

ПРИМЕР ▾

CML-инструкции

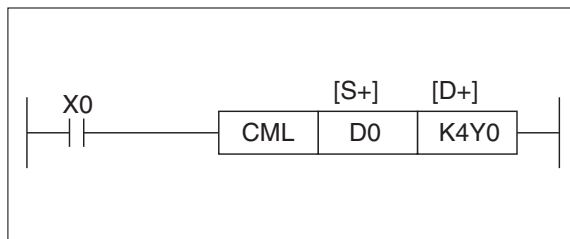
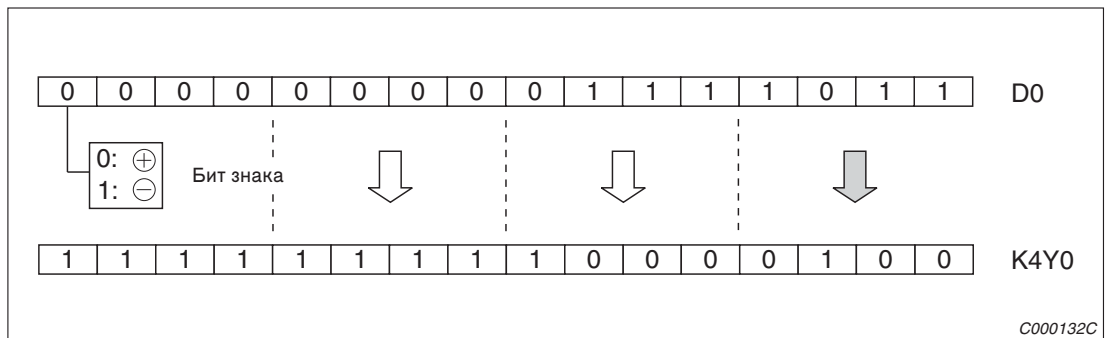


Рис. 6-26. Пример программирования для CML-инструкции

C000131C



C000132C

Рис. 6-27. Инвертирование и передача (перенос)



6.3.6 ПЕРЕДАЧА БЛОКА (BMOV)

				BMOV		FNC 15				
				Передача блока						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	●	●	
Операнды	S+		D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	KnX,KnY,KnM,KnS, T,C,D,V,Z, File-Register		KnY,KnM,KnS, T,C,D,V,Z, File-Registe	K, H		●	16бита	32бита	BMOV/ BMOVP	7

ФУНКЦИЯ

Перенос пакета данных

ОПИСАНИЕ

- Переносится (передается) предварительно заданное количество словных операндов.
- Для передачи предварительно задаются стартовый адрес (S+), адрес цели (D+) и число переносимых слов (n).
- С помощью BMOV-инструкции могут читаться или записываться файловые регистры семейства MELSEC FX.

УКАЗАНИЕ

Если величина пакета данных превышает величину областей цели или источника, то передаются только слова, которые могут поместиться в области.

Во всех контроллерах, кроме серии FX1S, направление переноса для команды BMOV можно изменить с помощью специального маркера M8024.

ПРИМЕР ▾

BMOV-инструкции

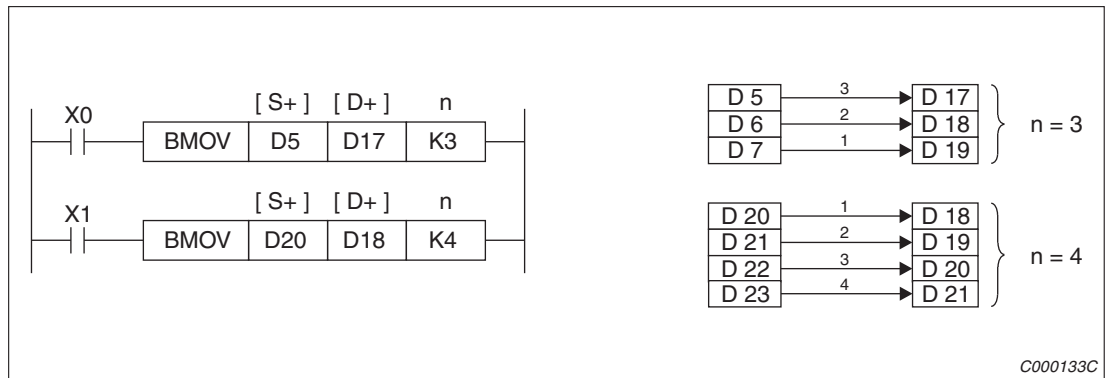


Рис. 6-28. Передача блока



6.3.7 ПЕРЕДАЧА ОДИНАКОВЫХ ДАННЫХ (FMOV)

				FMOV		FNC 16			
				Передача одинаковых данных					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	KnX,KnY,KnM,KnS, T,C,D,V,Z	KnY,KnM,KnS, T,C,D,V,Z	K, H		●	16бита	32бита	FMOV/ FMOV P	7
						●	●	DFMOV/ DFMOV P	13

ФУНКЦИЯ

Перенос значения данных в несколько операндов

ОПИСАНИЕ

- Содержание данных из (S+) переносится (передается) в несколько операндов одинакового типа.
- Первый операнд цели определяется в (D+).
- Исходя от операнда цели (D+) значение данных передается из (S+) в “n” операндов.

УКАЗАНИЕ

Если “n” больше число используемых операндов, то выполняется передача только до последнего имеющегося операнда.

ПРИМЕР ▾

FMOV-инструкции

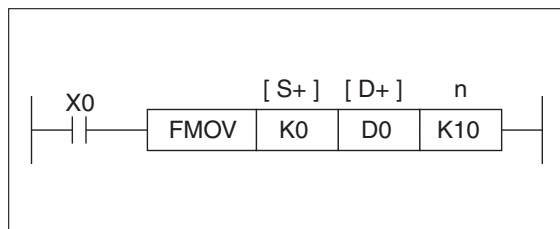


Рис. 6-29.

Пример программирования FMOV-инструкции

C000134C

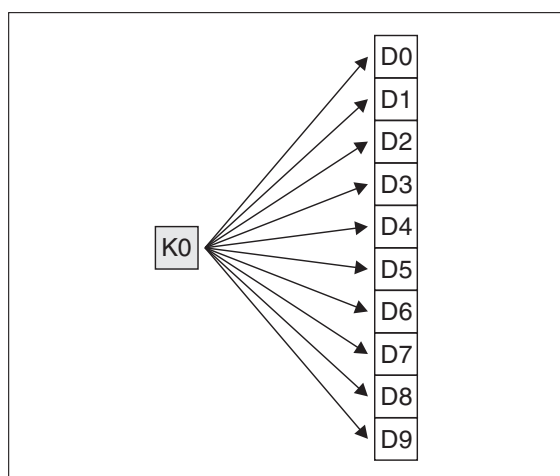


Рис. 6-30.

Передача (перенос) значения “0” в регистры данных D0...D9

C000119C



6.3.8 ОБМЕН ДАННЫМИ (XCH)

		XCH		FNC 17			
		Обмен данными					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	D1+, D2+	Имп. инструкция (P)	Обработка		Шаги программы		
	KnY, KnM, KnS T, C, D, V, Z		●	16бита	32бита	XCH/XCHP	5
			●	●	DXCH/DXCHP	9	

ФУНКЦИЯ

Обмен данными между двумя операндами

ОПИСАНИЕ

Обмениваются данные из (D1+) и (D2+).

УКАЗАНИЕ

Процесс обмена выполняется в каждом цикле, если не программируются никакое управление по фронту.

ПРИМЕР ▾

XCH-инструкции без специального меркера M8160

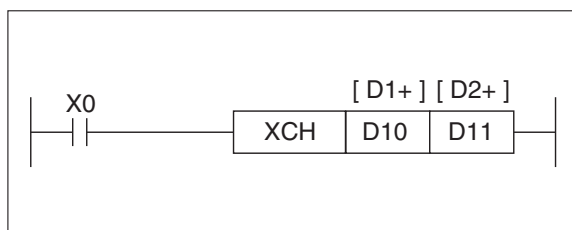


Рис. 6-31.

Пример программирования XCH-инструкции без специального меркера

C000135C

Значения перед выполнением инструкции: D10 = 5, D11 = 7

Значения после выполнения инструкции: D10 = 7, D11 = 5

△

ПРИНЦИП ФУНКЦИОНИРОВАНИЯ СО СПЕЦИАЛЬНЫМ МЕРКЕРОМ M8160

При введении специального меркера M8160 выполняется обмен старших и младших байтов в (D1+) и (D2+).

ОПИСАНИЕ

После введения специального меркера M8160 в (D1+) и (D2+) обмениваются старший и младший байты. После XCH-инструкция специальный меркер M8160 должен снова отключиться.

УКАЗАНИЕ

При применении XCH-инструкции со специальным меркером M8160 (D1+) и (D2+) должны иметь одинаковые типы регистров данных, в противном случае может появиться ошибка с сообщением флага ошибки M8067.

Процесс обмена выполняется в каждом цикле, если не программируется никакого управления по фронту.

ПРИМЕР ▾

DXCH-инструкция со специальным меркером M8160

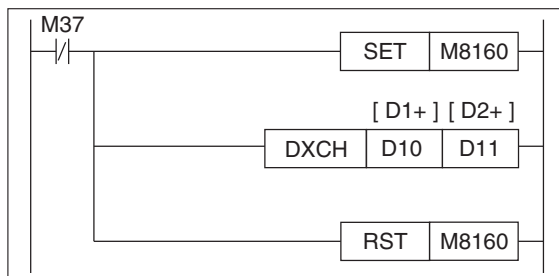


Рис. 6-32.

Пример программирования с DXCH-инструкцией со специальным меркером M8160

C000314C

Процесс обмена может быть представлен следующим образом:

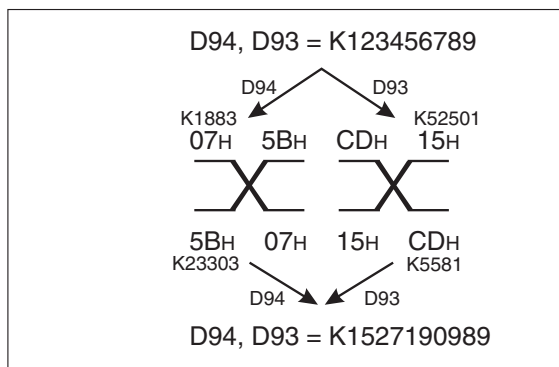


Рис. 6-33.

Представление процесса обмена (DXCH-инструкция обмена)

C000315C



УКАЗАНИЕ

При применении 32-х битных инструкций DXCH/DXCHP происходит обмен верхних и нижних байтов независимо в каждом отдельном слове (16 бит).

6.3.9 BCD-КОНВЕРТИРОВАНИЕ (BCD, DBCD)

		BCD		FNC 18			
		BCD-конвертирование					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	BCD/BCDP	5
				●	●	DBCD/DBCDP	9

ФУНКЦИЯ

Конвертирование двоичных данных в BCD-формат

ОПИСАНИЕ

Внутри ПК обрабатываются только двоичные данные. Благодаря применению BCD-инструкции могут выдаваться данные также в BCD-формате (например, для управления семисегментным устройством отображения).

- Двоичные данные источника (S+) конвертируются в BCD-данные и передаются по адресу цели (D+).
- Результат BCD-конвертирования должен находиться внутри допустимой области:
 16-ти-битовая-инструкция: от 0 до +9 999
 32-х-битовая-инструкция: от 0 до +99 999 999

ИСТОЧНИК ОШИБКИ

Если результат BCD-конвертирования находится вне допустимой области, то появляется ошибка обработки программы и инструкция не выполняется.

ПРИМЕР

BCD-инструкция может применяться, например, чтобы прочесть двоичное число из ПК и изобразить его на семисегментном указателе.

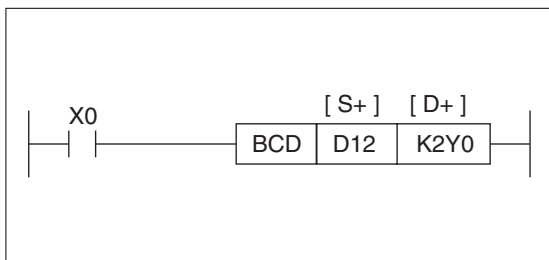
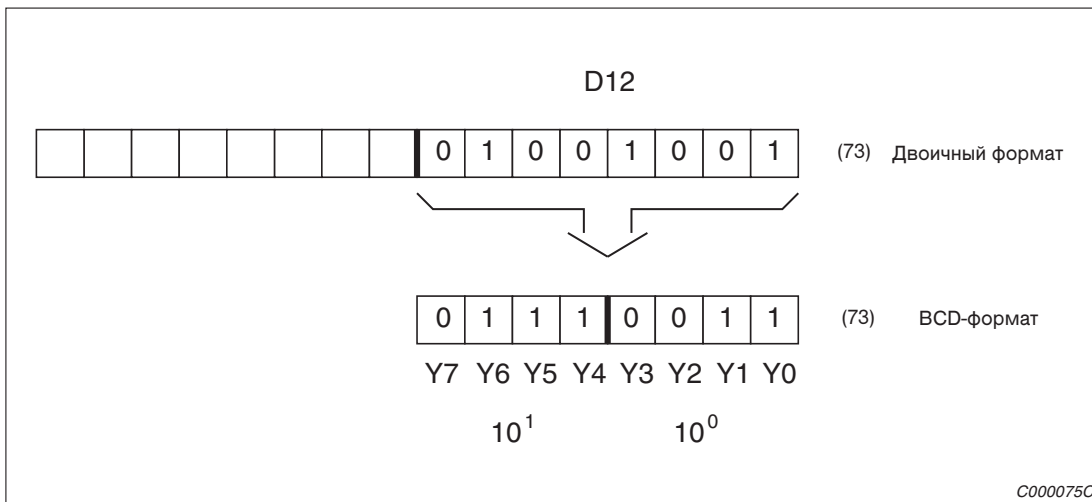


Рис. 6-34.
Пример программирования BCD-инструкции

C000074C



C000075C

Рис. 6-35. Пример программирования конвертирования двоичных данных в BCD-формат

Двоичные данные из регистра данных D12 конвертируются в BCD-формат и затем выдаются по выходам Y0...Y7. В этом примере: 73 (десятичные). △

6.3.10 ДВОИЧНОЕ КОНВЕРТИРОВАНИЕ (BIN, DBIN)

		BIN		FNC 19			
		Двоичное конвертирование					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	BIN/BINP	5
				●	●	DBIN/DBINP	9

ФУНКЦИЯ

Конвертирование из BCD-данных в двоичный формат

ОПИСАНИЕ

Внутри ПК обрабатываются только двоичные данные. Благодаря применению BIN-инструкции могут по входам считываться данные также в BCD-формате.

- BCD-данные источника (S+) конвертируются в двоичные данные и передаются по адресу цели (D+).
- Данные в (S+) должны находиться внутри допустимой области:
 16-ти-битовая-инструкция: от 0 до +9 999
 32-х-битовая-инструкция: от 0 до +99 999 999

ИСТОЧНИК ОШИБКИ

Если данные в (S+) не в BCD-формате, появляется ошибка. Ошибка отображается благодаря включившемуся специальному меркеру M8067. Специальный меркер M8068 не включается (см. также раздел 11.1.1).

ПРИМЕР ▽

VIN-инструкция может применяться, например, чтобы передать BCD-данные от цифрового переключателя в ПК.

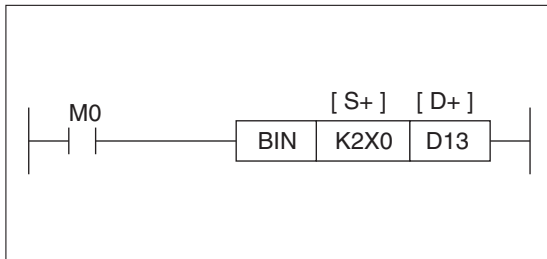
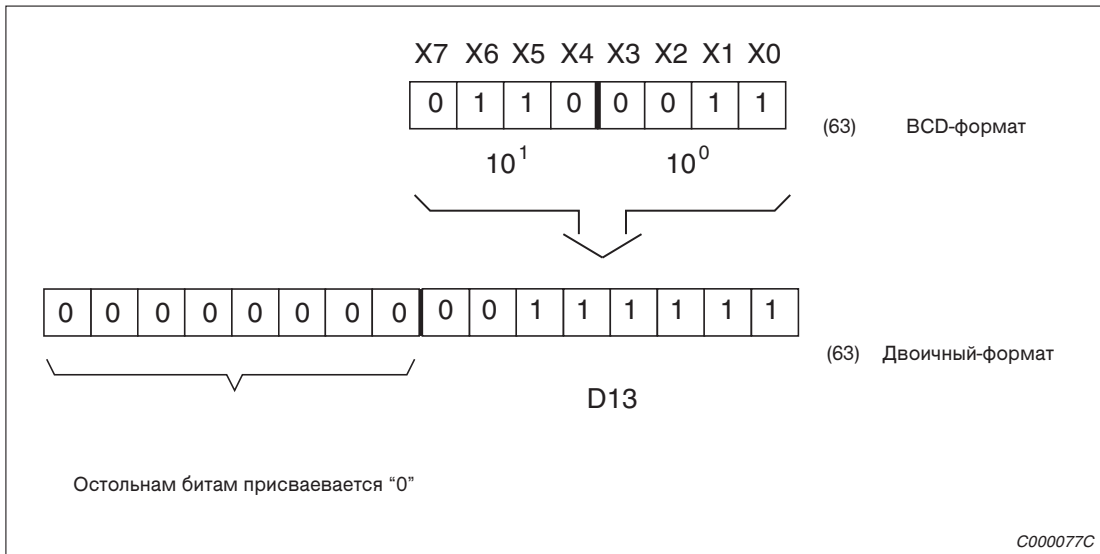


Рис. 6-36.
Пример программирования
BIN-инструкции

C000076C



C000077C

Рис. 6-37. Пример программирования конвертирования данных в BCD-формате в двоичную форму данных

B CD-данные на входах X0...X7 конвертируются в формат двоичных данных по адресу цели D13 и затем выдаются на выходы Y0...Y7. В этом примере: 73 (десятичное число). △

6.4 АРИФМЕТИЧЕСКИЕ ИНСТРУКЦИИ

ОБЗОР ИНСТРУКЦИЙ FNC 20...29

Символ	FNC	Назначение	Раздел
ADD	20	Сложение числовых данных	6.4.1
SUB	21	Вычитание числовых данных	6.4.2
MUL	22	Умножение числовых данных	6.4.3
DIV	23	Деление числовых данных	6.4.4
INC	24	Инкрементация (приращение)	6.4.5
DEC	25	Декрементация (уменьшение)	6.4.6
WAND	26	Логическая связь "И"	6.4.7
WOR	27	Логическая связь "ИЛИ"	6.4.8
WXOR	28	Логическая связь "НЕ ИЛИ"	6.4.9
NEG	29	Отрицание	6.4.10

Табл. 6-12. Обзор инструкций FNC 20...29

6.4.1 СЛОЖЕНИЕ ЧИСЛОВЫХ ДАННЫХ (ADD, DADD)

		ADD		FNC 20			
		Сложение числовых данных					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S+, S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	ADD/ADDP	7
				●	●	DADD/DADDP	13

ФУНКЦИЯ

Сложение двух числовых данных. Результат сложения хранится по адресу цели.

ОПИСАНИЕ

- Двоичные данные в адресах источников (S1+) и (S2+) суммируются. Результат суммирования запоминается в адресе цели (D+).

$$(S1+) + (S2+) = (D+)$$

- В старшем бите запоминается знак числа суммирования:

0: знак положительного числа

1: знак отрицательного числа

- При выполнении 32-х битной инструкции в ней указывается операнд слова младших 16 бит. Следующий за ней операнд является операндом слова старших 16 бит. Рекомендуется при задании адреса применять четные числа, чтобы не запрограммировать по ошибке наложение адресов.
- В адресе источника (S+) и адресе цели (D+) нужно также указывать одинаковые типы операндов.

УКАЗАНИЕ

При определенных результатах счета после исполнения инструкции включается специальный меркер (флаг).

ФЛАГ НУЛЯ M8020

Если результатом сложения является 0, включается флаг нуля

ФЛАГ ЗАИМСТВОВАНИЯ M8021

Если результатом сложения явилось число меньше -32 767 (16-ти битовая операция) или же -2 147 483 648 (32-х битовая операция), включается флаг заимствования.

ФЛАГ ПЕРЕНОСА M8022

Если результатом сложения явилось число выше +32 767 (16-ти битовая операция) или же +2 147 483 647 (32-х битовая операция), включается флаг переноса.

ПРИМЕР ▾

Применение ADD-инструкции

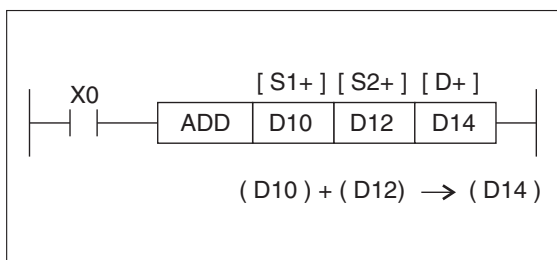


Рис. 6-38.
Пример программирования
ADD-инструкции

C000078C

Если включен X0, то суммируются значения данных в регистрах D10 и D12. Результат суммирования запоминается в регистре данных D14.

Применение DADD-инструкции

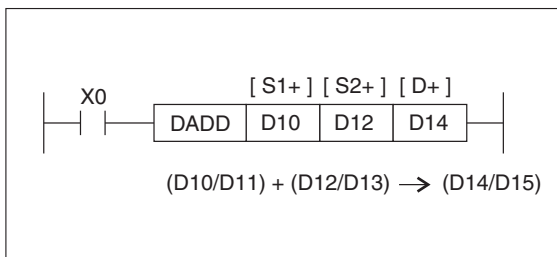


Рис. 6-39.
Пример программирования
DADD-инструкции

C000069C



6.4.2 ВЫЧИТАНИЕ ЧИСЛОВЫХ ДАННЫХ (SUB, DSUB)

		SUB		FNC 21				
		Вычитание числовых данных						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●	●	
Операнды	S+, S2+		D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	SUB/SUBP	7
					●	●	DSUB/DSUBP	13

ФУНКЦИЯ

Вычитание двух числовых данных. Результат вычитания хранится по адресу цели.

ОПИСАНИЕ

- Значение данных в (S2+) вычитается из значения данных (S1+). Результат вычитания запоминается в адресе цели (D+).
 $(S1+) - (S2+) = (D+)$
- В старшем бите запоминается знак числа вычитания:
 0: знак положительного числа
 1: знак отрицательного числа
- При выполнении 32-х битной инструкции в ней указывается операнд слова младших 16 бит. Следующий за ней операнд является операндом слова старших 16 бит. Рекомендуется при задании адреса применять четные числа, чтобы не запрограммировать по ошибке наложение адресов.
- В адресе источника (S+) и адресе цели (D+) нужно также указывать одинаковые типы операндов.

УКАЗАНИЕ

При определенных результатах счета после исполнения инструкции включается специальный меркер (флаг).

ФЛАГ НУЛЯ M8020

Если результатом вычитания является 0, включается флаг нуля.

ФЛАГ ЗАИМСТВОВАНИЯ M8021

Если результатом вычитания явилось число меньше -32 767 (16-ти битовая операция) или же -2 147 483 648 (32-х битовая операция), включается флаг заимствования.

ФЛАГ ПЕРЕНОСА M8022

Если результатом сложения явилось число выше +32 767 (16-ти битовая операция) или же +2 147 483 647 (32-х битовая операция), включается флаг переноса.

ПРИМЕР ▾

Применение SUB-инструкции

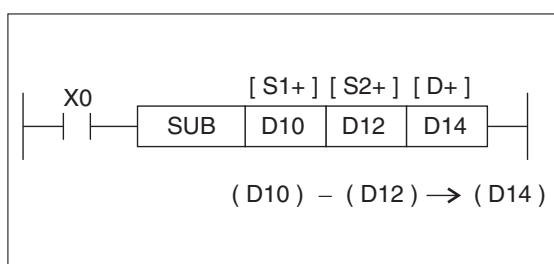


Рис. 6-40.

Пример программирования SUB-инструкции

C000079C

Если включен X0, то значения данных в регистре данных D12 вычитается из значения данных в регистре данных D10. Результат вычитания запоминается в регистре данных D14.

Применение DSUB-инструкции

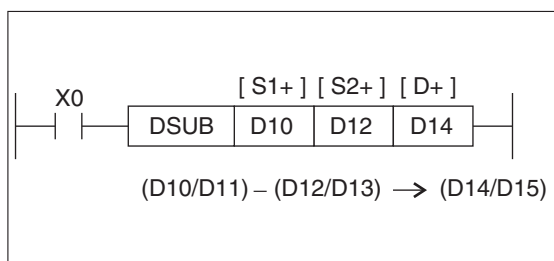


Рис. 6-41.

Пример программирования DSUB-инструкции

C000067C



6.4.3 УМНОЖЕНИЕ ЧИСЛОВЫХ ДАННЫХ (MUL, DMUL)

		MUL		FNC 22				
		Умножение числовых данных						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●	●	
Операнды	S+, S2+		D+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, Z (V)	●	16бита	32бита	MUL/MULP	7
					●	●	DMUL/DMULP	13

ФУНКЦИЯ

Умножение двух числовых данных. Результат умножения хранится по адресу цели.

ОПИСАНИЕ

- Данные в S1+ и S2+ перемножаются между собой. Результат умножения запоминается по адресу операнда указанного в D+ и в следующим за ним адресе операнда.
 $(S1+) \times (S2+) = (D+)$
- В старшем бите запоминается знак результата перемножаемых чисел:
 0: знак положительного числа
 1: знак отрицательного числа
- При выполнении 16-ти битной операции результат заносится в 32-х битное число в (D+) и ((D+)+1). При выполнении 32-х битной операции результат заносится в 64-х битное число в (D+) и в три следующих за ним операнда.
- При 32-х битном операнде для (D+) нельзя применять Z(V).

УМНОЖЕНИЕ 16-ТИ БИТНЫХ ДАННЫХ (MUL-ИНСТРУКЦИЯ)

Результат 16-ти битного умножения оказывается 32-х битным числом. Это число запоминается как 32-х битное значение. Младшие 16 бит записываются по адресу операнда, заданному в (D+). Старшие 16 бит записываются по следующему за ним адресу операнда.

ПРИМЕР ▾

Применение MUL-инструкции

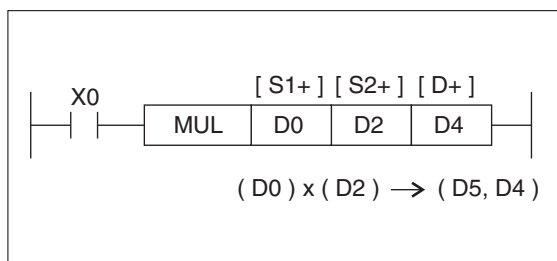


Рис. 6-42.
Пример программирования MUL-инструкции

C000080C

Результат умножения записывается как 32-х битное значение данных в регистры данных D4 и D5. В D4 стоят младшие 16 бит, а в D5 - старшие 16 бит. Знак числа стоит в 15-ом бите D5.

△

УМНОЖЕНИЕ 32-Х БИТНЫХ ДАННЫХ (DMUL-ИНСТРУКЦИЯ)

Результат 32-х битного умножения запоминается 64-х значением данных. Младшие 16 бит запоминаются по адресу операнда, заданному в (D+). Старшие биты записываются по следующему за ним адресу операнда.

ПРИМЕР ▾

Применение DMUL-инструкции

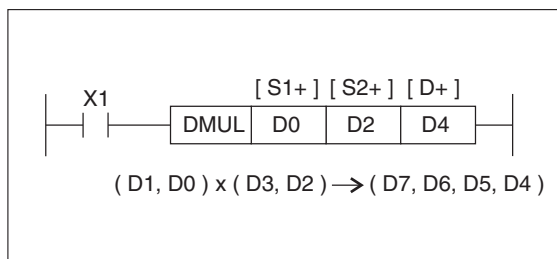


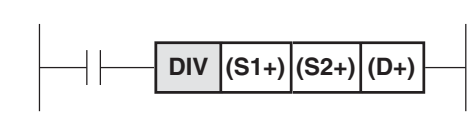
Рис. 6-43.
Пример программирования DMUL-инструкции

C000081C

Результат умножения записывается как 64-х битное значение данных в регистры данных D4, D5, D6 и D7. В D4 стоят младшие 16 бит, а в D5, D6 и D7 - старшие биты.

△

6.4.4 ДЕЛЕНИЕ ЧИСЛОВЫХ ДАННЫХ (DIV, DDIV)

		DIV		FNC 23						
		Деление числовых данных								
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U			
			●	●	●	●	●			
Операнды	S1+, S2+		D+		Имп. инструкция (P)		Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, Z (V)		●		16бита	32бита	DIV/DIVP	7
							●	●	DDIV/DDIVP	13

ФУНКЦИЯ

Деление двух числовых данных. Результат деления хранится по адресу цели.

ОПИСАНИЕ

- Выполняется деление данных в (S1+) и (S2+). Данные в (S1+) соответствуют делимому, в (S2+) - делителю. Результат деления запоминается по адресу операнда указанного в (D+) и в следующем адресе операнда. Остаток делимого запоминается в следующем адресе операнда.

$$(S1+) : (S2+) = (D+)$$

- В старшем бите запоминается знак числа результата деления.

0: знак положительного числа
1: знак отрицательного числа

Знак результата деления зависит от знаков чисел делимого и делителя.

- При 32-х битном операнде для (D+) нельзя применять Z(V).

Делимое	Делитель	Результат	Остаток
+	+	+	+
+	-	-	+
-	+	-	-
-	-	+	-

Табл. 6-13.

Знак числа результата деления

УКАЗАНИЕ

| Если адресом цели является бит-операнд, то ПК не может определить остатка от деления

ИСТОЧНИК ОШИБКИ

При работе программы появляется ошибка, если значение делителя равно 0. Инструкция не обрабатывается.

ДЕЛЕНИЕ 16-ТИ БИТНЫХ ДАННЫХ (DIV-ИНСТРУКЦИЯ)

Результат 16-ти битного деления запоминается по адресу операнда заданном D+. Остаток от деления запоминается в следующем за ним адресе операнда.

ПРИМЕР ▾

Применение DIV-инструкции

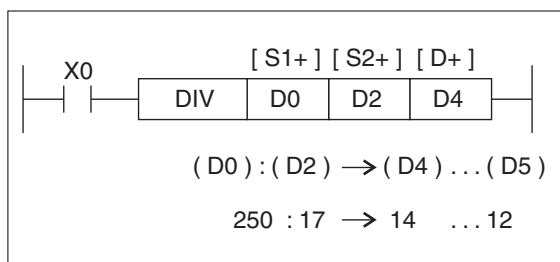


Рис. 6-44.
Пример программирования
DIV-инструкции

C000082C

Результат деления 14 записывается в регистр данных D4. Остаток от деления 12 записывается в следующий регистр данных D5.

△

ДЕЛЕНИЕ 32-Х БИТНЫХ ДАННЫХ (DDIV-ИНСТРУКЦИЯ)

При делении 32-х битных данных для делимого, делителя, результата и остатка от деления имеется соответственно по два следующих друг за другом регистра данных. В DDIV-инструкции должны указываться соответственно регистры данных с нижними адресами операндов.

ПРИМЕР ▾

Применение DDIV-инструкции

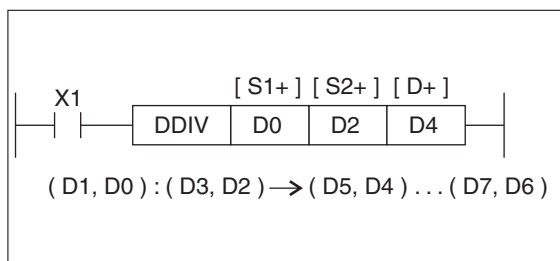


Рис. 6-45.
Пример программирования
DDIV-инструкции

C000083C

Результат деления записывается в регистры данных D4 и D5. Остаток от деления записывается в следующие регистры данных D6 и D7.

△

6.4.5 ИНСТРУКЦИЯ ПРИРАЩЕНИЯ (INC, DINC)

		INC		FNC 24		
		Приращение				
Операнды	D+	Имп. инструкция (P)	Обработка		Шаги программы	
		KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	INC/INCP
			●	●	DINC/DINCP	5

ФУНКЦИЯ

К числовому значению данных прибавляется число 1 (приращивается).

УКАЗАНИЕ

Эта команда выполняется в каждом программном цикле. Вы можете воспрепятствовать этому, вставив перед ней импульсную функцию (команду PLS или PLF) или применив командный параметр P.

При приращении к значению числа, имеющемуся в D+, прибавляется число 1, как только выполнится входное условие.

- **16-ТИ БИТОВЫЙ ОПЕРАНД (INC-ИНСТРУКЦИЯ)**
Если при 16-ти битовом операнде значение 1 добавится к + 32 767, то запишется значение -32 768. Не появляется никакого флага.
- **32-Х БИТОВЫЙ ОПЕРАНД (DINC-ИНСТРУКЦИЯ)**
Если при 32-х битовом операнде значение 1 добавится к числу +2 147 483 647, то запишется значение -2 147 483 648. Не появляется никакого флага.

ПРИМЕР ▾

Применение INC-инструкции

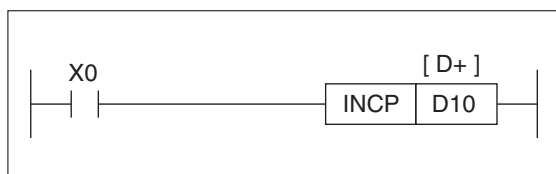


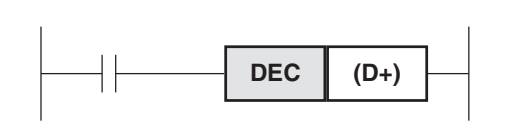
Рис. 6-46.
Пример программирования INCP-инструкции

C000136C

Значение данных в регистре данных D10 при наличии входного сигнала X0 повысится на число 1.

Инструкция активизируется благодаря подключенной впереди функции импульса. Это важно, чтобы процесс суммирования не выполнялся в каждом цикле программы. △

6.4.6 ИНСТРУКЦИЯ УМЕНЬШЕНИЯ (DEC, DDEC)

		DEC		FNC 25			
		Уменьшение					
Операнды	D+	Имп. инструкция (P)	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	Шаги программы		
			●	●	DEC/DECP	3	DDEC/DDECP

ФУНКЦИЯ

От числового значения данных вычитается число 1 (уменьшается).

УКАЗАНИЕ

Инструкция выполняется в каждом цикле программы. Этого можно избежать благодаря введению впереди функций импульса (PLS- или PLF-инструкции) или применив командный параметр P (для MELSEC FX/FX2N).

УМЕНЬШЕНИЕ

При уменьшении от значения числа, имеющегося в D+, вычитается число 1, как только выполнится входное условие.

- **16-ТИ БИТОВЫЙ ОПЕРАНД (DEC-ИНСТРУКЦИЯ)**
Если при 16-ти битовом операнде значение 1 отнимется от числа -32 768, то запишется значение +32 768. Не появляется никакого флага.
- **32-Х БИТОВЫЙ ОПЕРАНД (DDEC-ИНСТРУКЦИЯ)**
Если при 32-х битовом операнде значение 1 отнимется от числа -2 147 483 648, то запишется значение +2 147 483 647. Не появляется никакого флага.

ПРИМЕР ▽

Применение DEC-инструкции

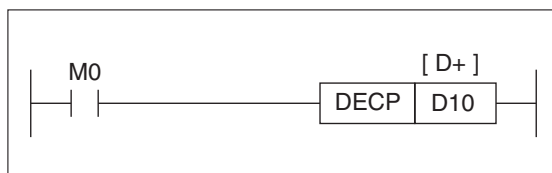


Рис. 6-47.
Пример программирования DECP-инструкции

C000066C

Значение данных в регистре данных D10 при каждом включении входного сигнала X0 уменьшится на число 1.

Инструкция активизируется благодаря подключенной впереди функции импульса. Это важно, чтобы процесс вычитания не выполнялся в каждом цикле программы. △

6.4.7 ЛОГИЧЕСКАЯ СВЯЗЬ “И” ДВОИЧНЫХ ДАННЫХ (WAND, DAND)

		WAND		FNC 26		
		Логическая связь “И”				
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●
Операнды	S1+, S2+	D+	Имп. инструкция (P)	Обработка	Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита 32бита	WAND/ WANDP	7

		DAND		FNC 26		
		Логическая связь “И”				
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U
		●	●	●	●	●
Операнды	S1+, S2+	D+	Имп. инструкция (P)	Обработка	Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита 32бита	DAND/ DANDP	13

ФУНКЦИЯ

Логическая связь “И” двоичных данных.

ОПИСАНИЕ

- Логическая связь “И” двоичных данных выполняется по отдельным битам.
- Данные в (S1+) и (S2+) побитно логически связываются друг с другом. Результат связи сохраняется в (D+).

(S1+)	(S2+)	(D+)
1	1	1
1	0	0
0	1	0
0	0	0

Табл. 6-14.
Таблица истинности “И”

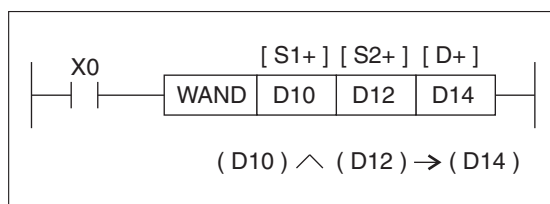


Рис. 6-48.
Пример программирования “И”-связи (выражения)

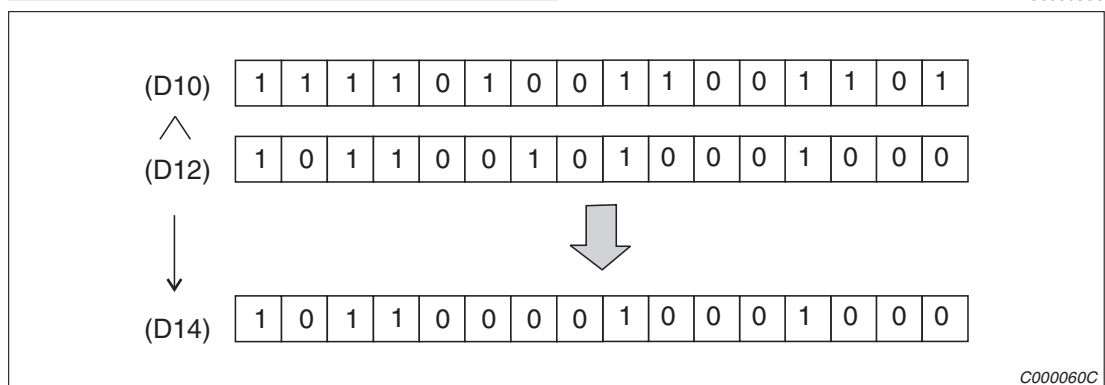


Рис. 6-49. Применение WAND-инструкции

6.4.8 ЛОГИЧЕСКАЯ СВЯЗЬ “ИЛИ” ДВОИЧНЫХ ДАННЫХ (WOR, DWOR)

		WOR		FNC 27			
		Логическая связь “ИЛИ”					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S1+, S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	WOR/ WORP	7

		DOR		FNC 27			
		Логическая связь “ИЛИ”					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S1+, S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	DOR/ DORP	13

ФУНКЦИЯ

Логическая связь “ИЛИ” двоичных данных.

ОПИСАНИЕ

- Логическая связь “ИЛИ” двоичных данных выполняется по отдельным битам.
- Данные в (S1+) и (S2+) побитно логически связываются друг с другом. Результат связи сохраняется в (D+).

(S1+)	(S2+)	(D+)
1	1	1
1	0	1
0	1	1
0	0	0

Табл. 6-15.
Таблица истинности “ИЛИ”

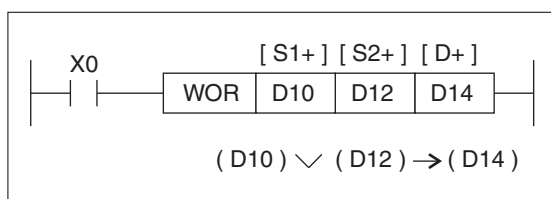


Рис. 6-50.
Пример программирования WOR-инструкции (“ИЛИ”-связи)

C000087C

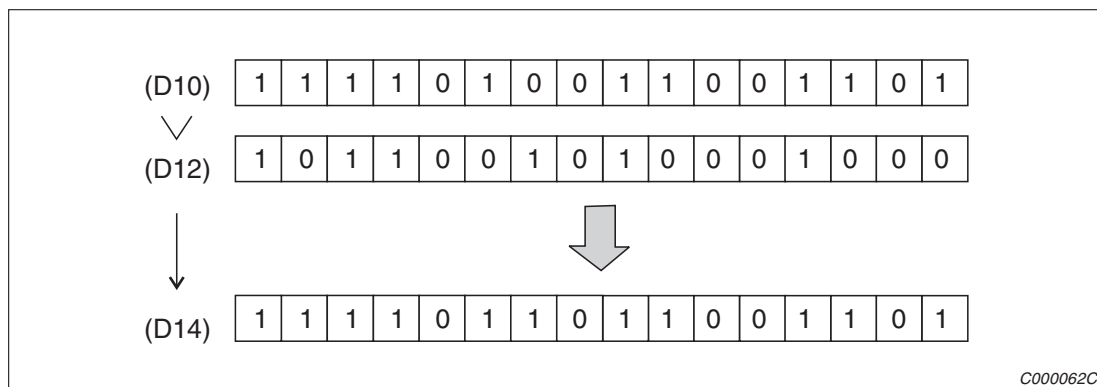


Рис. 6-51. Применение WOR-инструкции

6.4.9 ЛОГИЧЕСКАЯ СВЯЗЬ “ИСКЛЮЧАЮЩЕГО ИЛИ” ДВОИЧНЫХ ДАННЫХ (WXOR, DXOR)

		WXOR		FNC 28		
		Логическая связь “Исключающего ИЛИ”				
Операнды	S1+, S2+ K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	D+ KnY, KnM, KnS, T, C, D, Z (V)	Имп. инструкция (P) ●	Обработка	Шаги программы	
				16бита ●	32бита	WXOR/ WXORP

		DXOR		FNC 29		
		Логическая связь “Исключающего ИЛИ”				
Операнды	S1+, S2+ K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	D+ KnY, KnM, KnS, T, C, D, Z (V)	Имп. инструкция (P) ●	Обработка	Шаги программы	
				16бита	32бита ●	DXOR/ DXORP

ФУНКЦИЯ

Логическая связь “Исключающего ИЛИ” двоичных данных.

ОПИСАНИЕ

- Логическая связь “Исключающего ИЛИ” двоичных данных выполняется по отдельным битам.
- Данные в (S1+) и (S2+) побитно логически связываются друг с другом. Результат связи сохраняется в (D+).

(S1+)	(S2+)	(D+)
1	1	0
1	0	1
0	1	1
0	0	0

Табл. 6-16.

Таблица истинности “Исключающего ИЛИ”

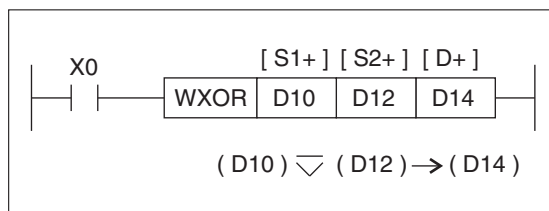


Рис. 6-52.

Пример программирования WXOR-инструкции (“Исключающего ИЛИ”-связи)

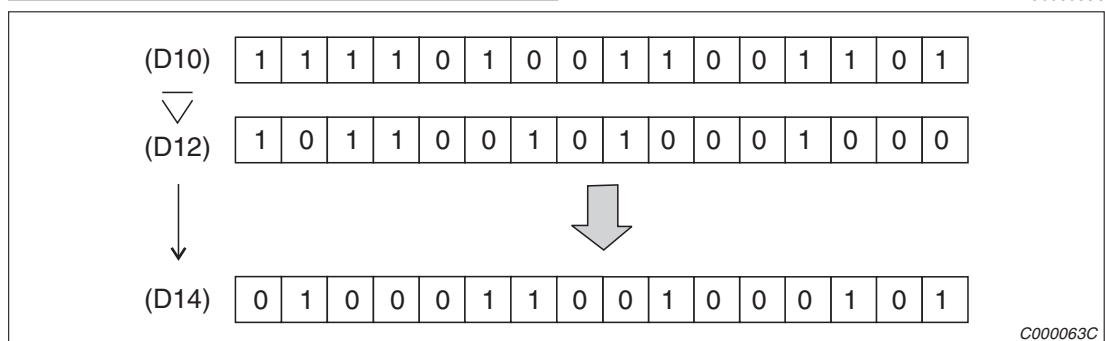


Рис. 6-53 Пример применение WXOR-инструкции

6.4.10 ИНВЕРСИЯ ДАННЫХ (NEG)

		NEG		FNC 29			
		инверсия данных					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	D	Имп. инструкция(P)	Обработка		Шаги программы		
	KnY, KnM, KnS T, C, D, V, Z		●	16бита	32бита	NEG/NEGP	3
				●	●	DNEG/DNEGP	5

ФУНКЦИЯ

Образование двоичного дополнительного кода

ОПИСАНИЕ

- С помощью инструкции NEG образуется второе дополнение значения данных, записанных в (D+), и оно сохраняется в том же (D+).

УКАЗАНИЕ

Если не программируется никакого опознания фронта, то образование дополнения повторяется в каждом цикле программы.

ПРИМЕР ▽

NEG-инструкция

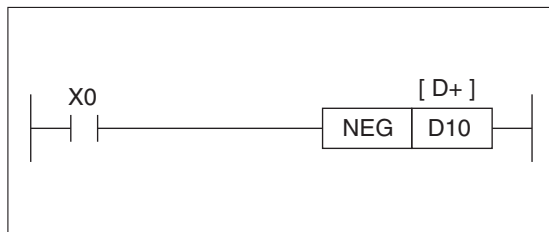
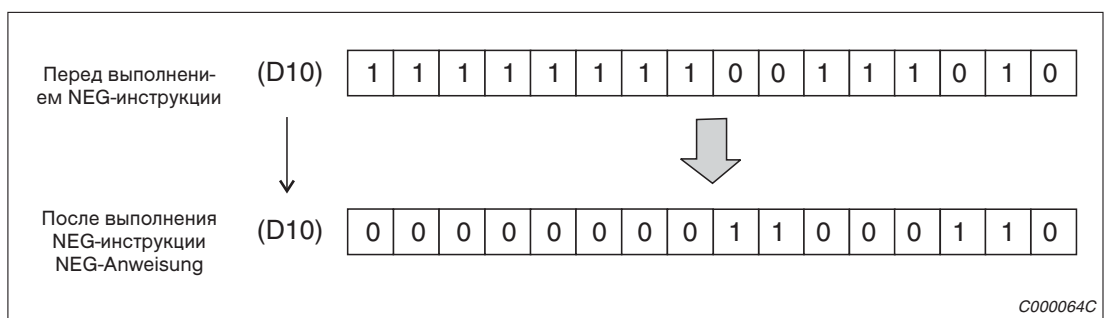


Рис. 6-54.
Пример программирования NEG-инструкции

C000137C

ФУНКЦИЯ

Двоичное: $D10 + 1 \rightarrow D10$



C000064C

Рис. 6-55. Функция NEG-инструкции

△

6.5 ИНСТРУКЦИИ СДВИГОВ

ОБЗОР ИНСТРУКЦИЙ FNC 30...39

Символ	FNC	Назначение	Раздел
ROR	30	Ротация вправо	6.5.1
ROL	31	Ротация влево	6.5.2
RCR	32	Ротация побитно вправо	6.5.3
RCL	33	Ротация побитно влево	6.5.4
SFTR	34	Сдвиг побитно двоичных данных вправо	6.5.5
SFTL	35	Сдвиг побитно двоичных данных влево	6.5.5
WSFR	36	Сдвиг данных пословно вправо	6.5.6
WSFL	37	Сдвиг данных пословно влево	6.5.7
SFWR	38	Запись в накопитель типа FIFO	6.5.8
SFRD	39	Чтение из накопителя типа FIFO	6.5.9

Табл. 6-17. Обзор инструкций FNC 30...39

6.5.1 СДВИГ ВПРАВО (ROR)

		ROR		FNC 30			
		Ротация вправо					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●	16бита	32бита	ROR, RORP	5
				●	●	DROR, DRORP	9

① Kn = K4 (16-тFacility Brochureи битный операнд), Kn = K8 (32-х битный операнд)

② n ≤ 16 (16-ти битный операнд), n ≤ 32 (32-х битный операнд)

ФУНКЦИЯ

Ротация бит по (n) мест вправо

ОПИСАНИЕ

- Битовое отображение в (D+) сдвигается вправо на n мест при каждом исполнении ROR
- Состояние последнего сдвигаемого бита копируются в M8022 (флаг передачи - Carry)

УКАЗАНИЕ

Если не программируется никакого опознавания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

ПРИМЕР ▾

ROR-инструкция

Битовые данные в регистре данных D0 каждый раз сдвигаются вправо на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ. Значение последнего сдвигаемого бита запоминается в флаге передачи.

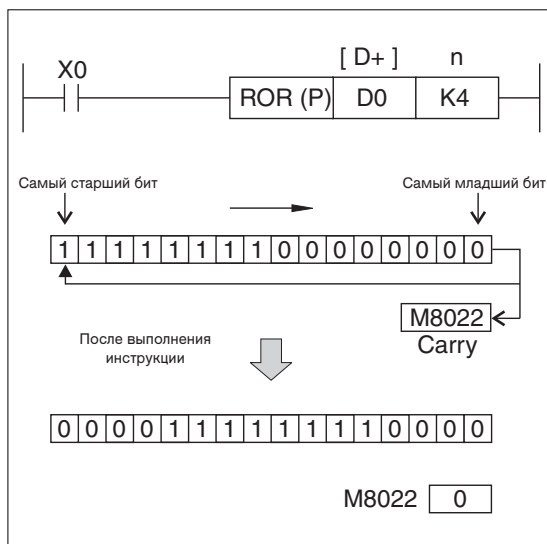


Рис. 6-56.

Пример программирования ротации вправо

C000091C



6.5.2 СДВИГ ВЛЕВО (ROL)

		ROL		FNC 31				
		Ротация влево						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	
Операнды	D+	n	Имп. инструкция (P)		Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●		16бита	32бита	ROL, ROLP	5
					●	●	DROL, DROLP	9

① Kn = K4 (16-ти битный операнд), Kn = K8 (32-х битный операнд)

② n ≤ 16 (16-ти битный операнд), n ≤ 32 (32-х битный операнд)

ФУНКЦИЯ

Ротация бит по (n) мест влево

ОПИСАНИЕ

- Битовое отображение в (D+) сдвигается влево на n мест при каждом исполнении ROR.
- Состояние последнего сдвигаемого бита копируются в M8022 (флаг передачи - Carry).

УКАЗАНИЕ

Если не программируется никакого опознавания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

ПРИМЕР ▾

ROL-инструкция

Битовые данные в регистре данных D0 каждый раз сдвигаются влево на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ. Значение последнего сдвигаемого бита запоминается в флаге передачи.

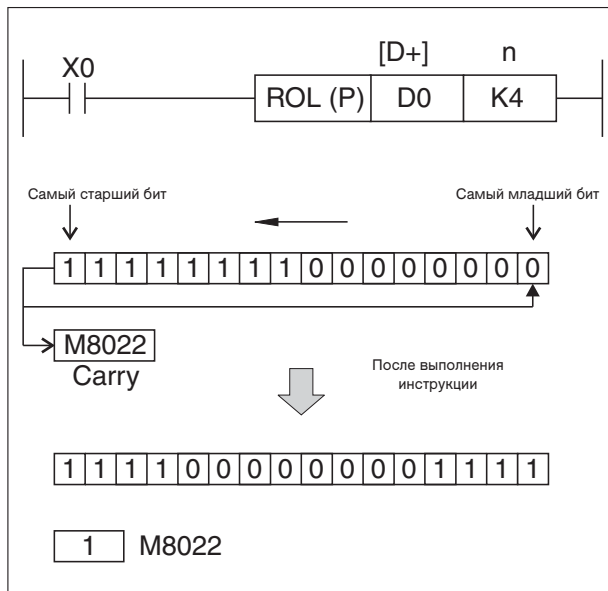


Рис. 6-57.

Пример программирования ротации влево

C000092C

6.5.3 КОЛЬЦЕВОЙ СДВИГ ВПРАВО (RCR)

		RCR		FNC 32			
		Ротация бит вправо					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	D+	n	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●	16бита	32бита	RCR, RCRP	5
				●	●	DRCR, DRCRP	9

① Kn = K4 (16-ти битный операнд), Kn = K8 (32-х битный операнд)

② n ≤ 16 (16-ти битный операнд), n ≤ 32 (32-х битный операнд)

ФУНКЦИЯ:

Ротация бит по (n) мест вправо с использованием флага передачи (Carry) M8022

ОПИСАНИЕ:

- Битовое отображение в (D+) сдвигается вправо на n мест.
- Флаг передачи вводится в петлю смещения.
- Если в (D+) должен быть использован операнд, составленный из битов, то для их составления используются только константы K4 (16-ти битный операнд) и Kn = K8 (32-х битный операнд).

УКАЗАНИЕ

Если не программируется никакого опознавания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

Состояние бита-передачи при включении инструкции воспринимается вместе со сдвигаемым битовым отображением.

ПРИМЕР ▾

RCR-инструкция

Битовые данные в регистре данных D0 каждый раз сдвигаются вправо на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ.

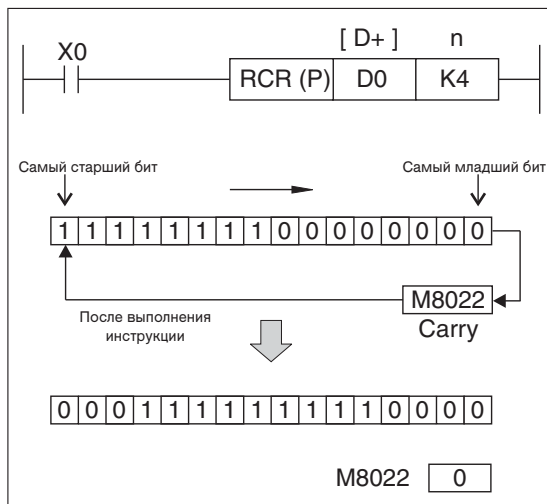


Рис. 6-58.
Пример программирования ротации вправо

C000093C



6.5.4 КОЛЬЦЕВОЙ СДВИГ ВЛЕВО (RCL)

		RCL		FNC 33			
		Ротация бит влево					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	D+	n	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●	16бита	32бита	RCL, RCLP	5
				●	●	DRCL, DRCLP	9

① Kn = K4 (16-ти битный операнд), Kn = K8 (32-х битный операнд)

② n ≤ 16 (16-ти битный операнд), n ≤ 32 (32-х битный операнд)

ФУНКЦИЯ

Ротация бит по (n) мест влево с использованием флага передачи (Carry-Flags) M8022

ОПИСАНИЕ

- Битовое отображение в (D+) сдвигается влево на n мест.
- Флаг передачи вводится в петлю смещения.
- Если в (D+) должен быть использован операнд, составленный из битов, то для их составления используются только константы K4 (16-ти битный операнд) и Kn = K8 (32-х битный операнд).

УКАЗАНИЕ

Если не программируется никакого опознавания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

Состояние бита-передачи при включении инструкции воспринимается вместе со сдвигаемым битовым отображением.

ПРИМЕР ▽

RCL-инструкция

Битовые данные в регистре данных D0 каждый раз сдвигаются влево на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ.

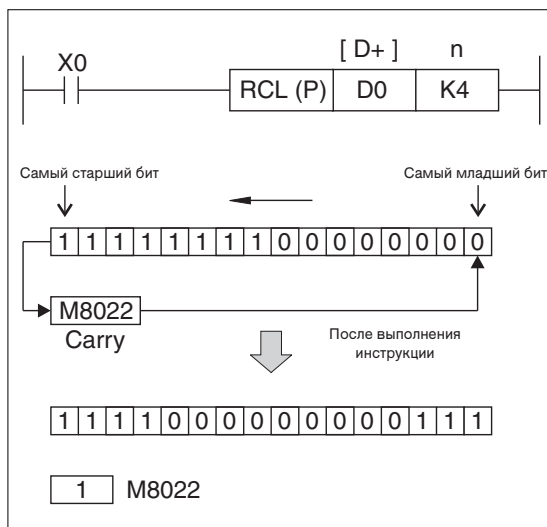


Рис. 6-59.

Пример программирования ротации влево

C000094C



6.5.5 ПОБИТНЫЙ СДВИГ ДВОИЧНЫХ ДАННЫХ (SFTR, SFTL)

				SFTR		FNC 34				
				Побитный сдвиг двоичных данных вправо						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
				●	●	●	●	●	●	
Операнды	S+	D+	n1, n2	Имп. инструкция (P)	Обработка		Шаги программы			
	X, Y, M, S	Y, M, S	K, H		●	16бита	32бита	SFTR/ SFTRP	9	

				SFTL		FNC 35				
				Побитный сдвиг двоичных данных влево						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
				●	●	●	●	●	●	
Операнды	S+	D+	n1, n2	Имп. инструкция (P)	Обработка		Шаги программы			
	X, Y, M, S	Y, M, S	K, H		●	16бита	32бита	SFTL/ SFTLP	9	

ФУНКЦИЯ:

Двоичные данные побитно сдвигаются вправо или влево

ОПИСАНИЕ:

- Эти инструкции касаются битовых операндов, которые охватываются словом данных. Ширина слова определяется числом n1.
- При исполнении инструкции биты в (D+) сдвигаются на n2 мест, а исходя из значений эти n2 бит в (S+) вводятся в начальные или конечные биты (D+) в зависимости от направления сдвига.
- n1: число адреса цели, начиная со стартового адреса в (D+)
n2: число смещаемых бит.
($n2 \leq n1 \leq 512$) и ($n1 \leq$ максимально возможному адресу указанному в (D+) операнду)
- С помощью SFTR-инструкции можно побитно сдвигать данные вправо
- С помощью SFTL-инструкции можно побитно сдвигать данные влево

УКАЗАНИЕ

Инструкция выполняется в каждом цикле программы. Этого можно избежать введением впереди функции импульса (PLS- или PLF- инструкции) или применением командного параметра P.

ПРИМЕР ▾ Применение SFTR-инструкции

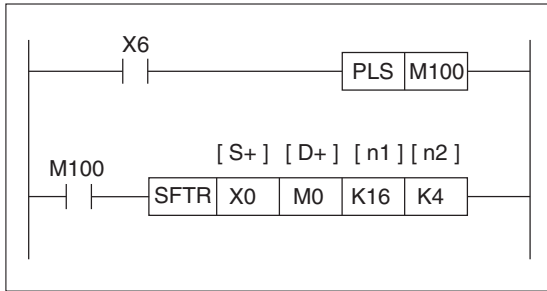
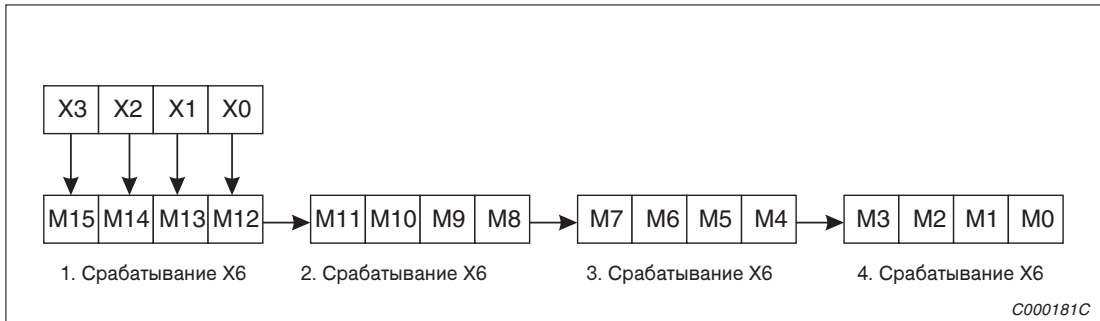


Рис. 6-60.
Пример программирования SFTR-инструкции

C000090C



C000181C

Рис. 6-61. Пример побитного сдвига вправо

При срабатывании X6 двоичные сигналы, находящиеся на входах X0...X3 побитно считываются в указанную область меркера и соответственно сдвигаются вправо. ▴

ПРИМЕР ▾ Применение SFTL-инструкции

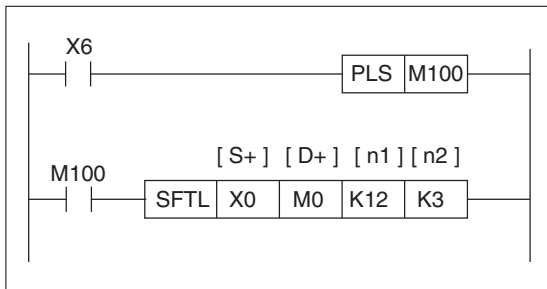
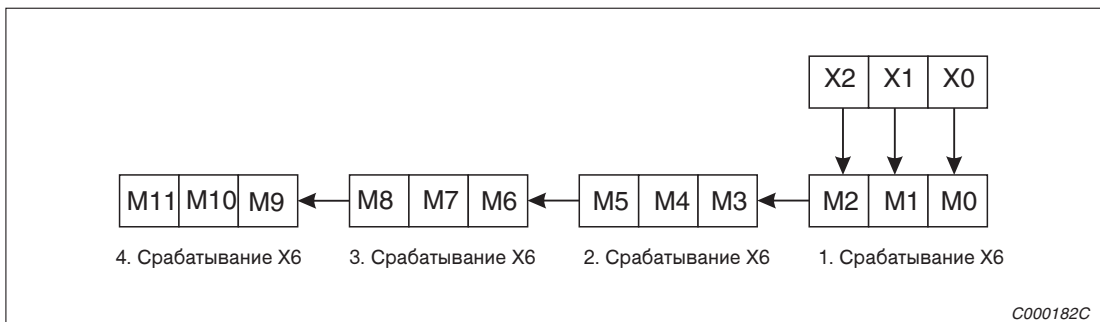


Рис. 6-62.
Пример программирования SFTL-инструкции

C000116C




C000182C

Рис. 6-63. Пример побитного сдвига влево

Если срабатывает, то двоичные сигналы на входах X0...X2 побитно считываются в указанную область меркера и сдвигаются влево. ▴

6.5.6 ПОСЛОВНЫЙ СДВИГ ДАННЫХ ВПРАВО (WSFR)

					WSFR		FNC 36		
					Пословный сдвиг данных вправо				
					CPU	FX1S	FX1N	FX2N	FX2NC
							●	●	●
Операнды	S+	D+	n1, n2		Имп. инструкция (P)	Обработка		Шаги программы	
	KnX,KnY,KnM,KnS,T,C,D	KnY,KnM,KnS,T,C,D	K, H n2Jn1J512		●	16бита	32бита	WSFR/ WSFRP	9

ФУНКЦИЯ

Данные пословно сдвигаются вправо

ОПИСАНИЕ

- Данные источника (S+) записываются и сдвигаются в стековом накопителе (D+). Глубина стека составляет n1 слов.
- При каждом исполнении инструкции считывается n2 слов и сдвигается содержание стека.

УКАЗАНИЕ

При применении объединенных битовых операндов необходимо следить за тем, чтобы (S+) и (D+) располагали одинаковым числом бит.

ПРИМЕР ▾

WSFR-инструкции

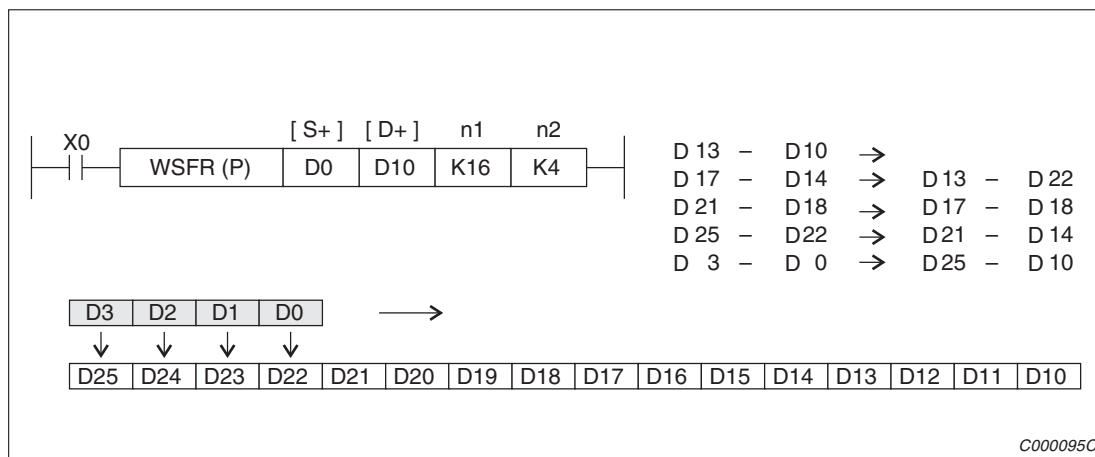


Рис. 6-64. Пример программирования сдвига вправо



6.5.7 ПОСЛОВНЫЙ СДВИГ ДАННЫХ ВЛЕВО (WSFL)

				WSFL		FNC 37			
				Пословный сдвиг данных влево					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S+	D+	n1, n2	Имп. инструкция (P)	Обработка		Шаги программы		
	KnX,KnY,KnM,KnS,T,C,D	KnY,KnM,KnS,T,C,D	K, H n2mn1m512	●	16бита	32бита	WSFL/ WSFLP	9	

ФУНКЦИЯ

Данные пословно сдвигаются влево

ОПИСАНИЕ

- Данные источника (S+) записываются и сдвигаются в стековом накопителе (D+). Глубина стека составляет n1 слов.
- При каждом исполнении инструкции считывается n2 слов и сдвигается содержание стека.

УКАЗАНИЕ

При применении объединенных битовых операндов необходимо следить за тем, чтобы (S+) и (D+) располагали одинаковым числом бит.

ПРИМЕР ▾

WSFL-инструкции

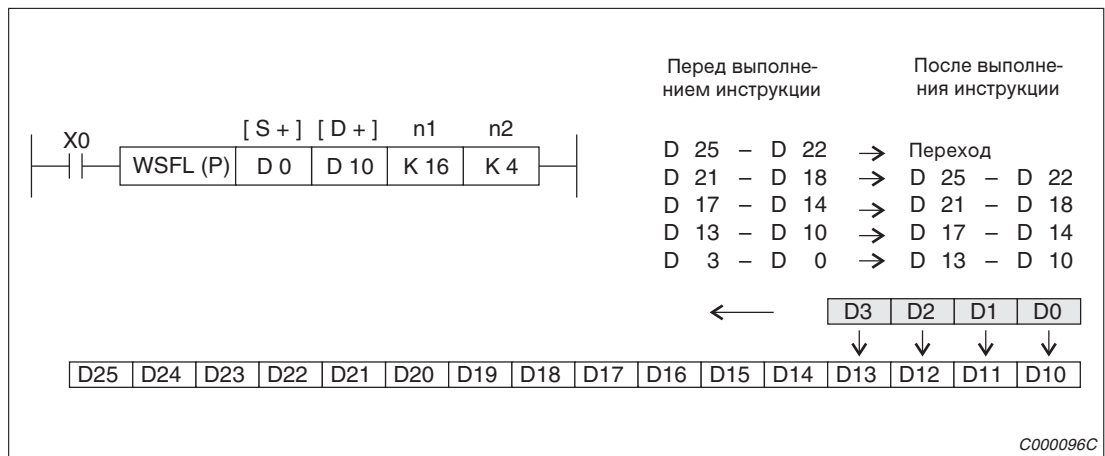


Рис. 6-65. Пример программирования сдвига влево



6.5.8 ЗАПИСЬ В СТЕКОВУЮ ПАМЯТЬ ТИПА FIFO (SFWR)

				SFWR		FNC 38			
				Запись в стековую память типа FIFO					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	KnX,KnY,KnM,KnS,T,C,D,V,Z	KnY,KnM,KnS,T,C,D	K, H 2mm512	●	16бита	32бита	SFWR/SFWRP	7	

ФУНКЦИЯ

Чтение слов в FIFO-стековой памяти, определенной по инструкции.

ОПИСАНИЕ

- Данные источника (S+) переписываются в стековый накопитель.
- Первым адресом стека является (D+).
- Глубина стека составляет n слов.
- В стеке может записываться максимум (n-1) слов, так как (D+) применяется как указатель для накопителя. (D+) перед первой инструкцией должен сбрасываться на ноль.
- Если (n-1) слов записывается в стек, не читая других слов, нельзя записывать следующие слова. Это состояние указывается включением битом переноса (Carry (M8022)).
- При каждом исполнении инструкции выполняется приращение указателя (D+).
- Инструкция применяется совместно с инструкцией SFRD; параметр (n) должен быть одинаковым в обеих инструкциях.

ПРИМЕР ▾

SFWR-инструкции

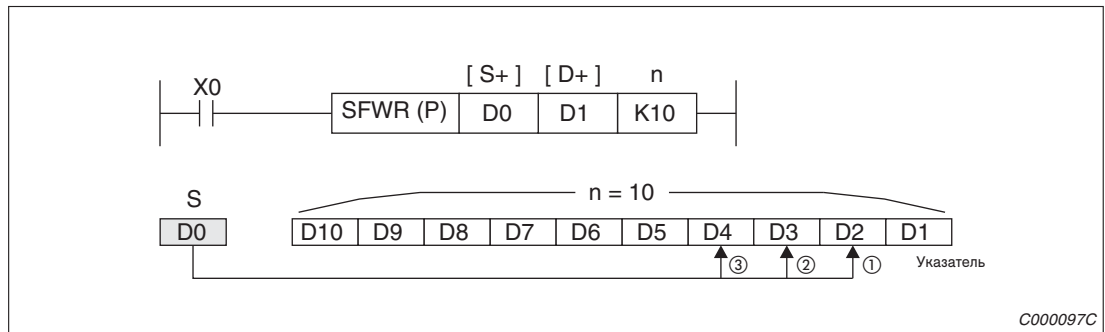


Рис. 6-66. Пример программирования записи в стек FIFO

Пример использования см. страницу 6-70.



6.5.9 ЧТЕНИЕ ИЗ СТЕКОВОЙ ПАМЯТИ ТИПА FIFO (SFRD)

				SFRD		FNC 39			
				Чтение из стековой памяти типа FIFO					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	KnX,KnY,KnM,KnS,T,C,D,V,Z	KnY,KnM,KnS,T,C,D	K, H 2mnm512	●	16бита	32бита	SFRD/ SFRDP	7	

ФУНКЦИЯ

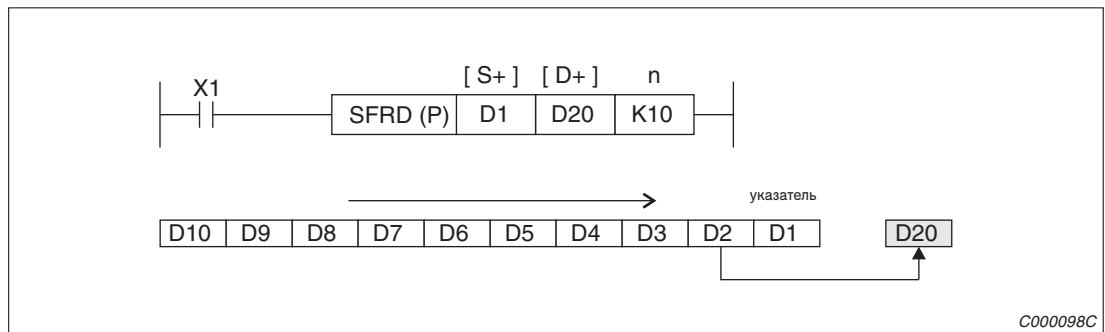
Чтение слов из FIFO-стековой памяти.

ОПИСАНИЕ

- Из стекового накопителя, начинающегося с (S+), считывается содержание по ((S+)+1) в (D+).
- Указатель стека (S+) уменьшается при каждом исполнении SFRD.
- Значения ((S+)+2 до ((S+)+n) перемещаются по одной позиции вверх.
- Если (S+) принимает значение нуль, то стековый накопитель пустой. Это указывается битом M8020.
- Инструкция SFRD работает совместно с инструкцией SFWR. Параметр (n) должен быть одинаковым в обеих инструкциях.

ПРИМЕР ▾

SFRD-инструкции



C000098C

Рис. 6-67. Пример программирования чтения из стека FIFO

Пример использования см. страницу 6-70.



ПРИМЕР ▾

Программирование стекового накопителя FIFO

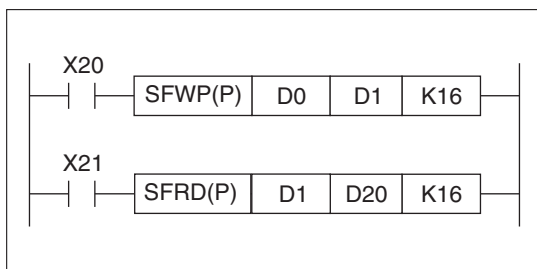


Рис. 6-68.
Пример программирования записи и чтения стека FIFO

C000146C

Определяется стековый накопитель с 15 адресами и указателем.

Приводится в действие от:		/	X20	X20	X20	X21	X20	X21
Указатель	D20	0	0	0	0	55	55	66
	D0	0	55	66	77	77	88	88
	D1	0	1	2	3	2	3	2
	D2	0	55	55	55	66	66	77
	D3	0	0	66	66	77	77	88
	D4	0	0	0	77	0	88	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	D15	0	0	0	0	0	0	0
	D16	0	0	0	0	0	0	0

Рис. 6-69. Регистр стекового накопителя FIFO

При каждом срабатывании X20 числовое значение D0 записывается в первый еще не занятый адрес внутри стекового накопителя.

При каждом срабатывании X21 содержание от D2 до D20 прочитывается, а содержание других адресов внутри стекового накопителя сдвигается на одну позицию. △

6.6 ОПЕРАЦИИ С ДАННЫМИ

ОБЗОР ИНСТРУКЦИЙ FNC 40...49

Символ	FNC	Назначение	Раздел
ZRST	40	Отключить область операндов	6.6.1
DECO	41	Декодирование данных	6.6.2
ENCO	42	Кодирование данных	6.6.3
SUM	43	Определение использованного бита	6.6.4
BON	44	Проверка бита	6.6.5
MEAN	45	Расчет среднего значения числа	6.6.6
ANS	46	Запуск интервала времени	6.6.7
ANR	47	Отключение бита отображения	6.6.8
SQR	48	Вычисление корня квадратного	6.6.9
FLT	49	Преобразование формата числа	6.6.10

Табл. 6-18. Обзор инструкций FNC 40...49

6.6.1 ОТКЛЮЧИТЬ ОБЛАСТЬ ОПЕРАНДОВ (ZRST)

		ZRST		FNC 40			
		Отключить область операндов					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D1+, D2+	Имп. инструкция (P)	Обработка		Шаги программы		
	Y, M, S, T, C, D	●	16бита	32бита	ZRST/ ZRSTP	5	

ФУНКЦИЯ

Несколько следующих друг за другом операндов (область операндов) может быть отключена только одной ZRST-инструкцией в состоянии сигнала или действительное значение "0".

ОПИСАНИЕ

- В (D1+) и (D2+) определяется область операндов, которые могут быть отключены.
- В (D1+) и (D2+) нужно указать одинаковые типы операндов.

(D1+): адрес первого операнда
(D2+): адрес последнего операнда

Должно соблюдаться: $(D1+) \leq (D2+)$
Если $(D1+) > (D2+)$, отключается только операнд, указанный в (D1+).

УКАЗАНИЕ

Хотя здесь предполагаются 16-ти битные операнды, в обеих адресах цели могут применяться также 32-х битные счетчики. Однако комбинированное применение 16-ти и 32-х битных счетчиков не допустимо. Так, например, не разрешено в (D1+) задать 16-ти битный счетчик, а в (D2+) 32-х битный.

Отдельные операнды могут отключаться с помощью RST-инструкции (см. раздел 4.10).

ПРИМЕР ▾

ZRST-инструкции

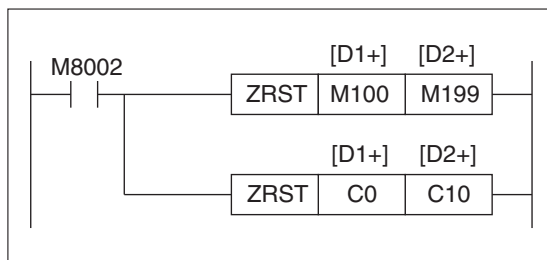


Рис. 6-70.
Пример программирования ZRST-инструкции

C000100C

Битовые операнды M100...M199 отключаются до состояния сигнала "0". Словные операнды C0...C10 отключаются до состояния действительного значения "0". Выключаются соответствующие катушки и контакты. △

6.6.2 ДЕКОДИРОВАНИЕ ДАННЫХ (DECO)

				DECO		FNC 41					
				Декодирование данных							
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
					●	●	●	●	●		
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы				
	K, H, X, Y, M, S, T, C, D, V, Z	Y, M, S, T, C, D	K, H	●	16бита	32бита	DECO/DECOP	7			

ФУНКЦИЯ

Декодирование данных

ОПИСАНИЕ

Данные в n операндов, начиная со стартового адреса, указанного в (S+), декодируются. В (D+) определяется стартовый адрес операнда цели, куда записывается результат декодировки.

- n: Число операндов, данные которых должны декодироваться.
 При указании битового операнда в D+ должно соблюдаться: $(1 \leq n \leq 8)$.
 При указании словного операнда в D+ должно соблюдаться: $(1 \leq n \leq 4)$.
- (S+): Стартовый адрес операндов, данные которых должны декодироваться
- 2ⁿ: Количество операндов цели
- (D+): Стартовый адрес операнда цели

УКАЗАНИЕ

- | Инструкция не выполняется, если n = 0.
- | Инструкция выполняется только в том случае, если включены начальные условия. Соответственно выход остается активным, если входные условия в конце действия снова отключаются.

ИСТОЧНИКИ ОШИБОК

- Ошибка при отработке программы появляется, если n находится не в области от 0 до 8.
- Ошибка при отработке программы появляется, если все биты выходных операндов имеют значение "0".

ПРИМЕР ▾

Применение DECO-инструкции с указанием битовых операндов в D+ (1J nJ 8)

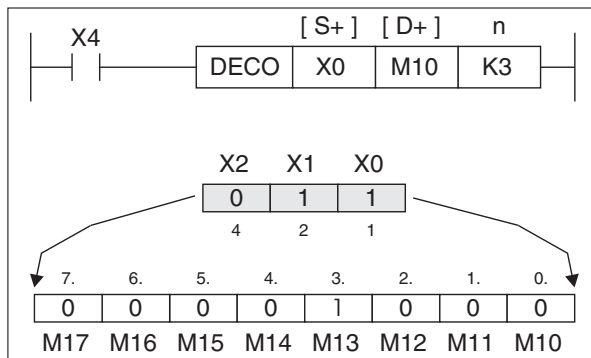


Рис. 6-71.

Пример программирования DECO-инструкции с указанием битовых операндов в D+

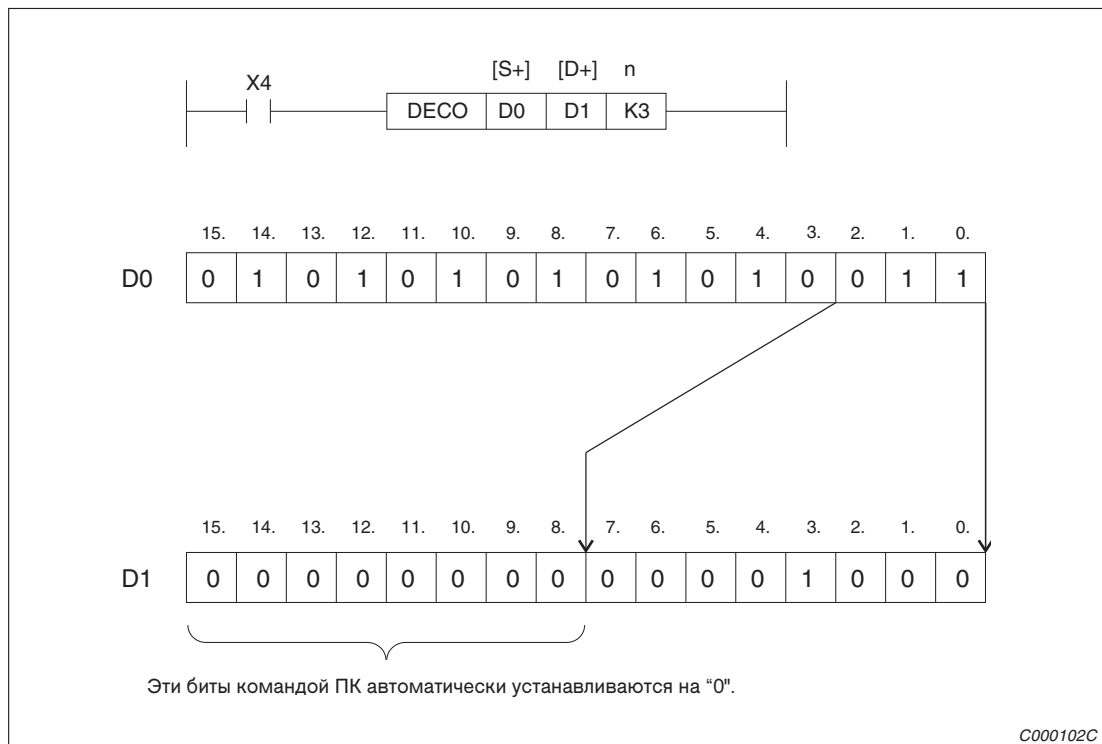
C000101C

Если $n = 3$, обрабатываются входные операнды X0, X1 и X2. Потому что $2^n = 2^3 = 8$ представлены в качестве адресов цели меркеров M10...M17.

Значения входных операндов равны $1 + 2 = 3$. Соответственно третий бит адреса цели, т.е. меркер M13, включается. Если обрабатывается значение входного операнда "0", то включается меркер M10. ▴

ПРИМЕР ▾

Применение DECO-инструкции с указанием словных операндов в D+ ($1 \leq n \leq 4$)



C000102C

Рис. 6-72. Пример программирования DECO-инструкции с указанием словных операндов в D+

Младшие 3 бита из регистра данных D0 декодируются. Результат декодирования $1 + 2 = 3$ передается в регистр данных D1. В этом регистре данных включается 3 бита.

Если значение для $n \leq 3$, то все ненужные биты более высокого номера в адресах цели устанавливаются на ноль.

6.6.3 КОДИРОВАНИЕ ДАННЫХ (ENCO)

				ENCO		FNC 42			
				Кодирование данных					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	X, Y, M, S, T, C, D, V, Z	T, C, D, V, Z	K, H	●	16бита	32бита	ENCO/ ENCOP		7

ФУНКЦИЯ

Определение, какие биты установлены

ОПИСАНИЕ

Данные в 2^n операндов, начиная со стартового адреса, указанного в (S+), кодируются. В (D+) определяется операнд цели, куда записывается результат кодировки.

2^n : Количество операндов, данные которых должны кодироваться.

n: Число операндов цели

При указании битового операнда в S+ должно соблюдаться: $(1 \leq n \leq 8)$

При указании словного операнда в S+ должно соблюдаться: $(1 \leq n \leq 8)$

(S+): Стартовый адрес операндов, данные должны кодироваться.

(D+): Операнд цели

УКАЗАНИЕ

Если несколько операндов, указанных в (S+), имеют значение 1, то обрабатывается только старший бит.

Инструкция не выполняется, если $n = 0$.

Инструкция выполняется только в том случае, если включены начальные условия. Соответственно выход остается активным, если входные условия в конце действия снова отключаются.

ИСТОЧНИКИ ОШИБОК

- Ошибка при отработке программы появляется, если n находится не в области от 0 до 8.
- Ошибка при отработке программы появляется, если все биты выходных операндов имеют значение "0".

ПРИМЕР ▾

Применение ENCO-инструкции с указанием битовых операндов в (S+) ($1 \leq n \leq 8$)

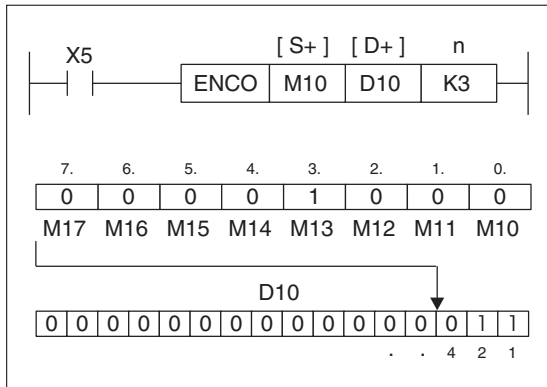


Рис. 6-73. Пример программирования ENCO-инструкции с указанием битовых операндов в (S+)

Если $2^n = 2^3 = 8$, то в качестве адресов выходов имеются меркеры M10...M17. Поскольку у операндов выхода 3-ий операнд, т.е. меркер M13 включается, в регистр данных D10 записывается значение 3. ▴

C000103C

ПРИМЕР ▾

Применение ENCO-инструкции с указанием словных операндов в S+ ($1 \leq n \leq 4$)

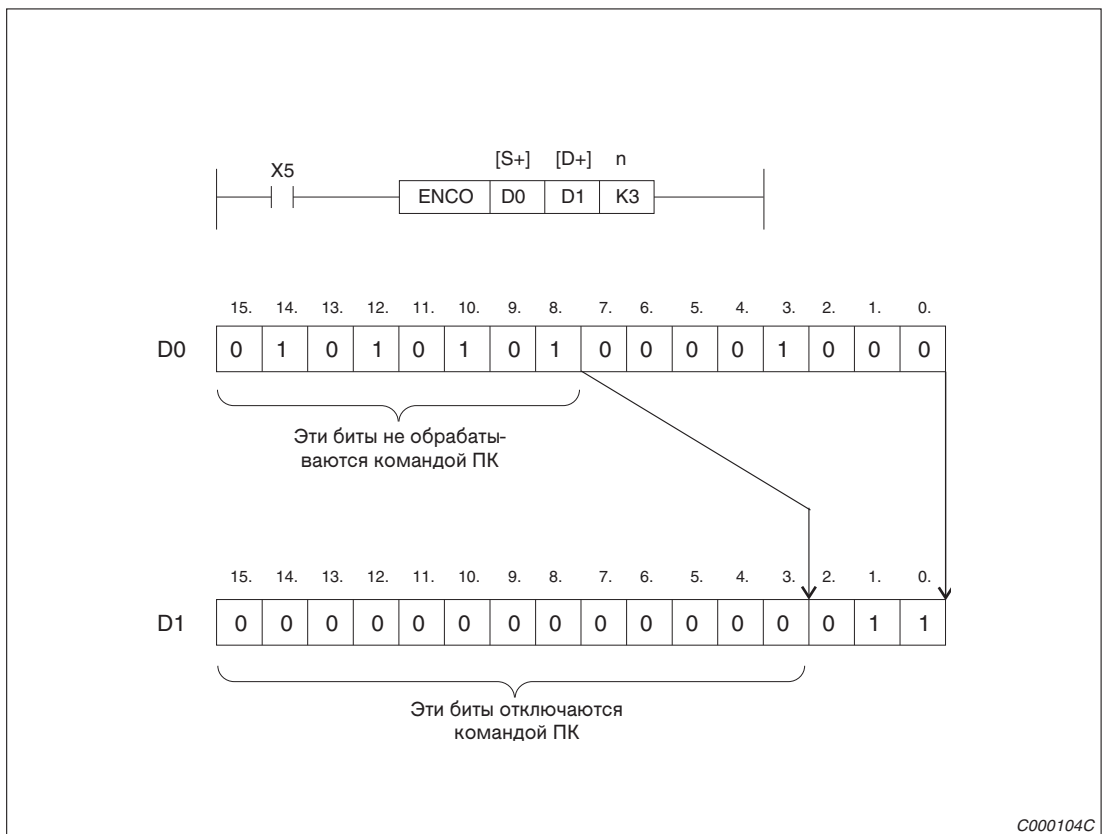
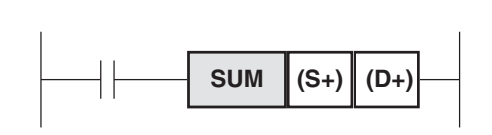


Рис. 6-74. Пример программирования ENCO-инструкции с указанием словных операндов в (S+)

В регистре данных D0 включается 3-ий бит. Тем самым значение числа 3 кодируется и сохраняется в регистре данных D1. ▴

C000104C

6.6.4 СУММА АКТИВНЫХ БИТ (SUM)

		SUM		FNC 43			
		Определение используемых битов					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	S+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	SUM, SUMP	7
				●	●	DSUM, DSUMP	9

ФУНКЦИЯ

Определение количества используемых битов в слове данных.

ОПИСАНИЕ

- Определяется количество включенных битов в (S+).
- Определенное значение заносится в (D+).

УКАЗАНИЕ

Если обрабатывается 32-х битная операция, то в старшие 16 битов ((D+) + 1) операндов цели (D+) устанавливаются в ноль, так как максимальное число включенных битов в (S+) составляет 32.

ПРИМЕР ▾

SUM-инструкция

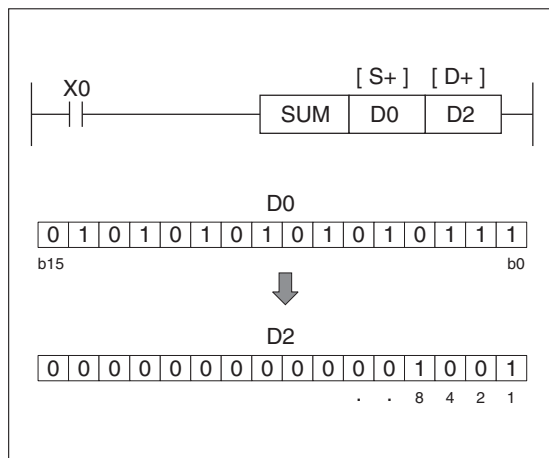


Рис. 6-75.

Пример программирования для определения включенных бит в (S+)

C000141C



6.6.5 ПРОВЕРКА БИТОВ (BON)

				BON		FNC 44				
				Проверка битов						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	●	●	
Операнды	S+		D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, S, T, C, D, V, Z		Y, M, S	K, H ^①		●	16бита	32бита	BON/BONP	7
							●	●	DBON/DBONP	13

① от= 0 до 15 (для 16-ти битной операции); n= от 0 до 31 (для 32-х битной операции)

ФУНКЦИЯ

Проверяется отдельный бит внутри слова данных.

ОПИСАНИЕ

- Если бит по номеру n включен внутри (S+), то включается соответствующий бит в (D+).

ПРИМЕР ▾

BON-инструкция

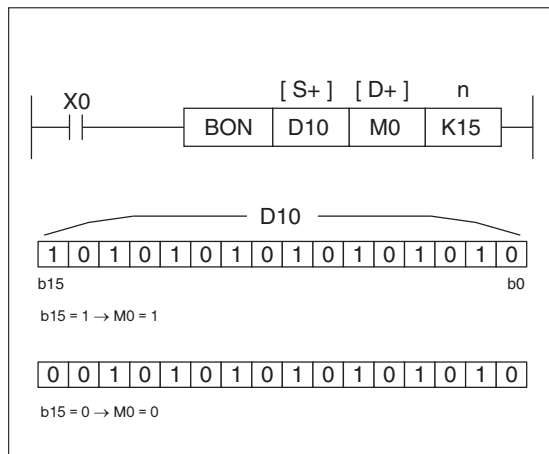


Рис. 6-76.

Пример программирования для проверки бита

C000142C



6.6.6 ВЫЧИСЛЕНИЕ СРЕДНЕГО ЗНАЧЕНИЯ (MEAN)

				MEAN		FNC 45			
				Вычисление среднего значения					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D	KnY, KnM, KnS, T, C, D, V, Z	K, H (n = 1 ... 64)	●	16бита	32бита	MEAN/ MEANP	7	
					●	●	DMEAN/ DMEANP	13	

ФУНКЦИЯ

Образуется арифметическое среднее значение по нескольким словам данных.

ОПИСАНИЕ

Начиная с (S+) суммируются n слов данных и делятся на n. Целое число результата заносится в (D+).

УКАЗАНИЕ

Если выбранное n больше, чем имеющаяся область операндов, начиная с (S+), то n автоматически согласовывается с имеющимся количеством операндов.

ИСТОЧНИК ОШИБКИ

Ошибка при отработке программы появляется, если n больше 64

ПРИМЕР ▾

MEAN-инструкция

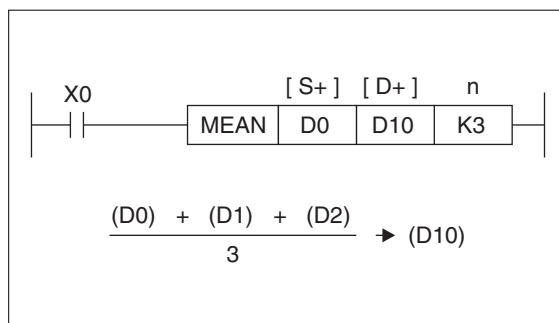


Рис. 6-77.

Пример программирования для вычисления среднего значения

C000143C



6.6.7 ЗАПУСК ИНТЕРВАЛОВ ВРЕМЕНИ (ANS)

				ANS		FNC 46			
				Запуск интервалов времени					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S+	D+	m	Имп. инструкция (P)	Обработка		Шаги программы		
	T T0 ... T199	S S900 ... S999	K (1 ... 32767)		16бита	32бита	ANS		7
					●				

ФУНКЦИЯ

Запуск интервалов времени и включение бита отображения

ОПИСАНИЕ

- Запуск интервалов времени и включение бита отображения.
- При выполнении этой инструкции запускается время m x 100 мс.
- После отсчета времени включается бит отображения (D+).
- В (S+) предварительно задается таймер, который определяет интервал времени.

УКАЗАНИЕ

| Примененный таймер не должен больше использоваться в последующей программе.

ПРИМЕР ▾

Программирование ANS-инструкции

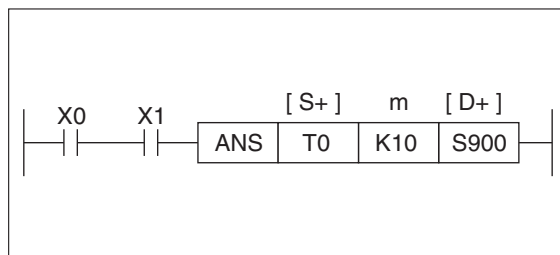


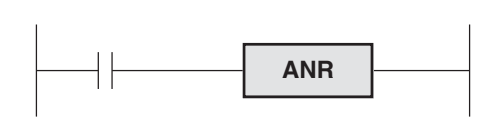
Рис. 6-78.

Пример программирования для запуска временного интервала

C000144C



6.6.8 ОТКЛЮЧЕНИЕ БИТА ОТОБРАЖЕНИЯ (ANR)

		ANR		FNC 47			
		Отключение бита отображения					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	—	Имп. инструкция (P)		Обработка		Шаги программы	
	—	●	16бита	32бита	ANR/ANRP		1

ФУНКЦИЯ

Отключение бита отображения

ОПИСАНИЕ

Если инструкция активна, то отключается активный бит отображения с самого младшего адреса.

УКАЗАНИЕ | Инструкция должна выполняться с опцией "P".

ПРИМЕР ▾ Программирование ANR-инструкции

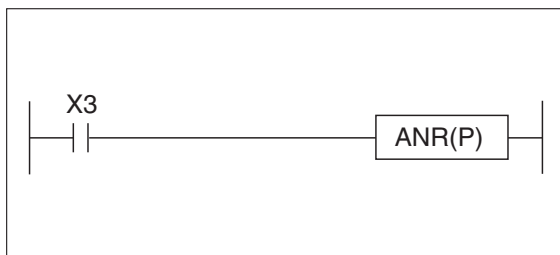


Рис. 6-79.
Пример программирования отключения бита отображения

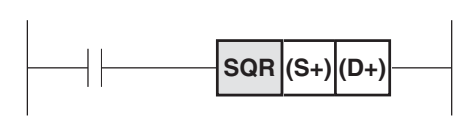
C000145C

Если включен X3, отключается используемый бит отображения между S900 и S999.

Если использовано несколько битов отображения, отключается бит отображения с самым низким адресом.

Другие включенные биты отображения отключаются при повторном включении X3 в возрастающей последовательности их адресов. ▴

6.6.9 ВЫЧИСЛЕНИЕ КОРНЯ КВАДРАТНОГО (SQR)

			SQR		FNC 48			
			Вычисление корня квадратного					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
						●	●	●
Операнды	S+	D+	Имп. инструкция (P)		Обработка		Шаги программы	
	K, H, D	D	●		16бита	32бита	SQR/SQRP	5
					●	●	DSQR/DSQRP	9

ФУНКЦИЯ

Вычисление корня квадратного, $(D+) = \sqrt{S+}$

ОПИСАНИЕ

Исходя из данных в (S+), выполняется вычисление корня квадратного и с округлением до целого числа результат заносится в (D+).

ПРИМЕР ▾

Программирование SQR-инструкции без флага

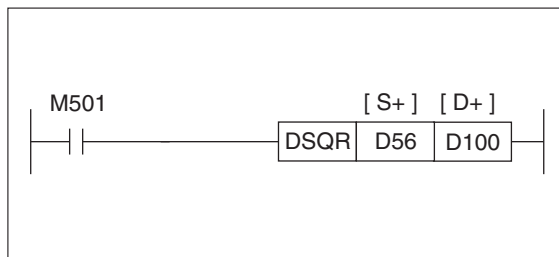


Рис. 6-80.

Пример программирования расчета корня квадратного

C000302C

Если включен меркер M501, рассчитывается значение корня квадратного в регистре данных D56 и результат как округленное целое число записывается в регистр данных D100.

УКАЗАНИЕ

Корень квадратный из отрицательного числа всегда приводит к ошибке и включается меркер ошибки M8067.

В следующей таблице приведено несколько результатов примеров по расчету корня квадратного SQR.

(S+)	Результат	(D+)
25	5,0	5
60	7,746	7
-236	15,36 i	ERROR
147	12,124	12

Табл. 6-19.

Результаты примеров расчета корня квадратного

6.6.10 ПРЕОБРАЗОВАНИЕ ФОРМАТОВ ЧИСЕЛ (FLT)

			FLT		FNC 49				
			Преобразование форматов чисел						
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
						●	●	●	
Операнды	S+	D+	Имп. инструкция (P)	Обработка		Шаги программы			
	D	D	●	●	●	16бита	32бита	FLT/FLTP	5
				●	●			DFLT/DFLTP	9

ФУНКЦИЯ

Преобразование чисел формата целого числа в формат числа с плавающей запятой

ОПИСАНИЕ

Исходя из данных в (S+), целое число преобразовывается в число с плавающей запятой и заносится в (D+).

УКАЗАНИЕ

Результат преобразования чисел всегда записывается в 32-х битный регистр данных.

ПРИМЕР ▾

Программирование FLT-инструкции

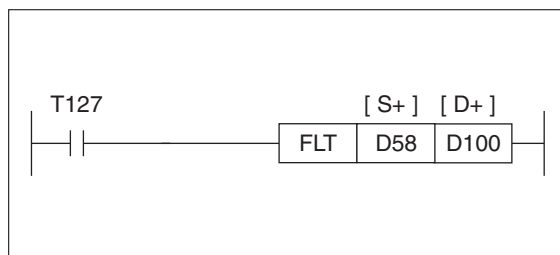


Рис. 6-81.

Пример программирования преобразования в формат с плавающей запятой

C000304C

Как только таймер T127 включится, значение в регистре данных D58 преобразуется в число с плавающей запятой и его значение запишется в регистр данных D100.

6.7 ВЫСОКОСКОРОСТНЫЕ ИНСТРУКЦИИ

ОБЗОР ИНСТРУКЦИЙ FNC 50...59

СИМВОЛ	FNC	НАЗНАЧЕНИЕ	РАЗД.
REF	50	Обновление входов и выходов	6.7.1
REFF	51	Установка входного фильтра	6.7.2
MTR	52	Чтение матрицы	6.7.3
DHSCS	53	Включение по высокоскоростному счетчику	6.7.4
DHSCR	54	Выключение по высокоскоростному счетчику	6.7.4
DHSZ	55	Сравнение областей	6.7.5
SPD	56	Определение скорости	6.7.6
PLSY	57	Выдача заданного числа импульсов	6.7.7
PWM	58	Выдача импульсов с модуляцией ширины	6.7.8
PLSR	59	Выдача определенного числа импульсов	6.7.9

Табл. 6-20. Обзор инструкций FNC 50...59

6.7.1 ОБНОВЛЕНИЕ ВХОДОВ И ВЫХОДОВ (REF)

			REF		FNC 50			
			Обновление входов и выходов					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●	●	●	●
Операнды	D	n	Имп. инструкция (P)	Обработка		Шаги программы		
	X, Y ^①	K, H ^②	●	16бита	32бита	REF, REFP		5

① Операнд должен быть кратным 10: X0, X10, X20, и т. д.

② n должно быть кратным 8: 8, 16, 24, etc.

ФУНКЦИЯ

Обновление входов и выходов

ОПИСАНИЕ

- Обработка программы у ПК серии FX выполняется по методу отображения процесса управления. Перед обработкой программы CPU ПК считывает состояние сигналов входов и сохраняет их в специальной области памяти - регистрах отображения входов. Таким образом обрабатываются не реальные входы, а регистр отображения входов.
После обработки программы считываются данные (записанные при обработке программы) из регистров отображения выходов и передаются на реальные выходы.
- С помощью REF-инструкции входы во время цикла работы программы могут опрашиваться и обновляться содержание регистра отображения.
- Можно применять REF-инструкцию, чтобы прочесть последнюю информацию входов, во время выполнения операции.
- И далее с помощью REF-инструкции можно выдавать результаты операции непосредственно после ее выполнения.
- REF-инструкция может, например, вводиться в FOR-NEXT-инструкцию или между SJ-инструкцией (наивысший номер шага) и относящейся к ней точкой маркировки (наименьший номер шага).

УКАЗАНИЕ

| Состояние входов и выходов обновляется в каждом цикле программы.

ПРИМЕР ▾ Применение REF-инструкции, обновление входов

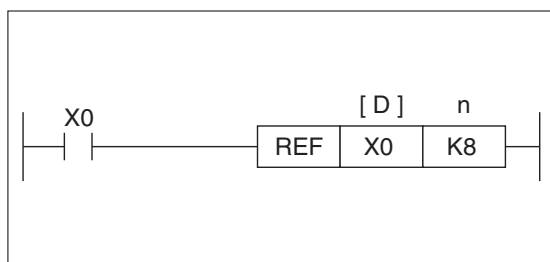


Рис. 6-82.
Пример программирования REF-инструкции; обновление входов

C000106C

Обновляется 8 адресов, т.е. входов X0...X7.

Если входы активированы примерно за 10 мс (время задержки) перед обработкой REF-инструкции, то входной регистр отображения активизируется, если выполнится RTF-инструкция. ▴

УКАЗАНИЕ

Время задержки сигналов на входах X0...X7 (X0...X17 у серии FX2N и FX3U, кроме FX3U-16M) можно регулировать с помощью команды REFF (см. раздел 6.7.2) или непосредственно в специальном регистре D8020.

ПРИМЕР ▾ Применение REF-инструкции, обновление входов

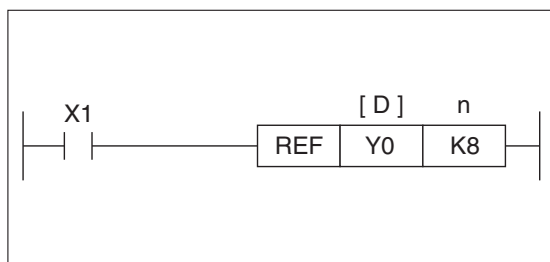


Рис. 6-83.
Пример программирования REF-инструкции; обновление входов

C000107C

Обновляется 8 адресов, т.е. выходов Y0...Y7.

Если выходы включены, то включатся принадлежащие к ним регистры отображения выходов после выполнения REF-инструкции. Зажим реального выхода активизируется по истечению времени ответа. Время ответа является физически обусловленным временем включения активизированного выхода. ▴

6.7.2 УСТАНОВКА ВХОДНОГО ФИЛЬТРА (REFF)

		REFF		FNC 51			
		Ein- und Ausgdnge auffrischen					
		CPU	FX1S	FX1N	FX2N	FX2NC	F X3U
			●	●	●	●	●
Операнды	D	n	Имп. инструкция (P)		Обработка		Шаги программы
	X, Y a	K, H	●		16бита	32бита	REF, REFP

ФУНКЦИЯ

Настройка времени фильтра для входов X0...X7 (X0...X17 у серии FX3U, кроме FX3U-16M) в специальном регистре D8020.

ОПИСАНИЕ

- В результате выполнения команды REFF состояния входов X0...X7 (X0...X17 у серий FX2N и FX3U, кроме FX3U-16M) перенимаются в область отображения и входные фильтры настраиваются на (n) мс.
- (n) может принимать значения между 0 и 60 [мс]. Настройка на 0 означает следующее время фильтра:
 - серии FX1S, FX1N, FX2N и FX2NC: 50 мкс
 - серии FX3U
 - X0...X5: 5 мкс
 - X6 и X7: 50 мкс
 - X10...X17: 200 мкс (в контроллере FX3U-16M эти входы жестко настроены на 10 мс)

УКАЗАНИЕ

Инструкция должна быть включена в каждом цикле, иначе уставка фильтра устанавливается на значение по умолчанию 10ms.

В контроллерах серии FX3U, независимо от настройки с помощью команды REFF, при следующих функциях для входов X0...X5 активируется время фильтра 5 мкс, а для входов X6 и X7 - время фильтра 50 мкс:

- применение входа для активирования прерывания
- применение входа для высокоскоростных счетчиков
- выполнение команды SPD (см. раздел 6.7.6)

При других функциях действительно время, установленное в специальном регистре D8020. Входные сигналы, длительность которых меньше времени фильтра, не регистрируются.

ПРИМЕР ▾

Если активен вход X10, то уставка фильтра примет значение 1ms, при отключенном - 10ms. По команде "REFF K20 " устанавливается задержка включения 20ms.

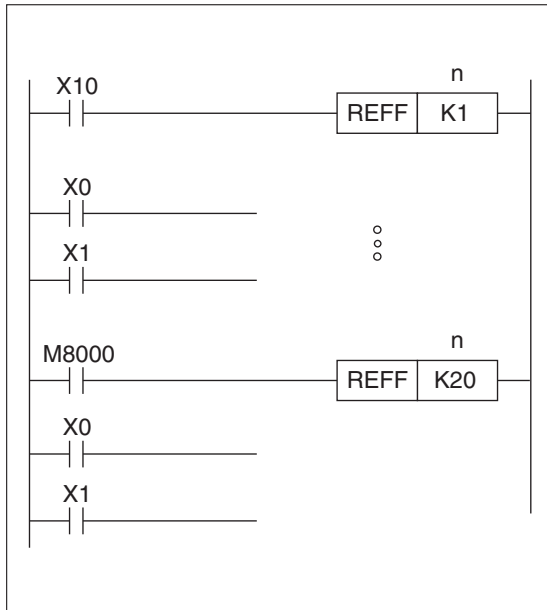


Рис. 6-84.
Пример использования REFF-инструкции.

C000150C



6.7.3 ЧТЕНИЕ МАТРИЦЫ (MTR)

				MTR		FNC 52			
				Чтение матрицы					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●	●	●	●	●
Операнды	S+, D1+	D2+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	H ^①	Y, M, S	K, H n = 2 ... 8		16бита	32бита	MTR	9	
					●				

① Операнд должен быть кратным 10: X0, X10, X20, и т. д.

ФУНКЦИЯ

Чтение в ПК 8 x n-матрицы

ОПИСАНИЕ

- Переключатель 8 x n-матрицы считывается по Multiplex-методу (методу умножения) по 8-ми входам и n выходам.
- Входы от (S+) до ((S+) +8) распределяются по каждому из n рядов меркеров. Меркеры для первого ряда начинаются с (D2+).
- Каждый ряд соответствует выходу; Первый ряд срабатывает на выходе (D1+).

УКАЗАНИЕ

- | Для выполнения инструкции должен использоваться ПК с транзисторными выходами.
- | Инструкция может использоваться в программе только один раз.

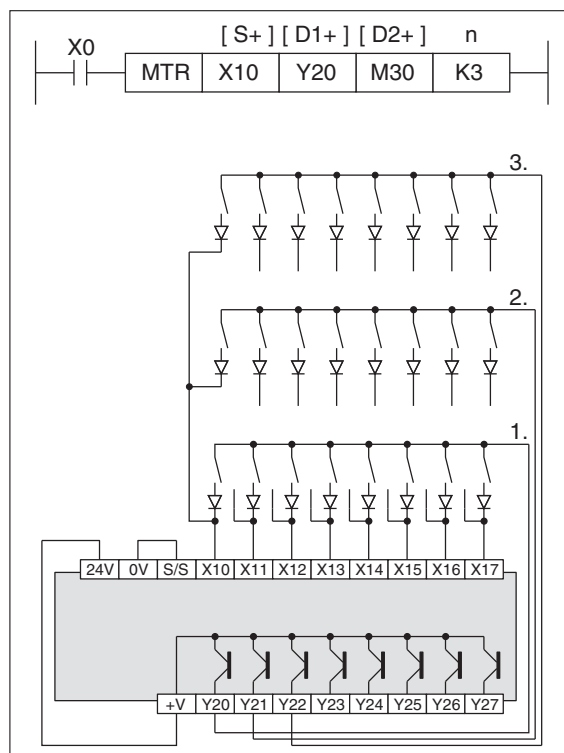


Рис. 6-85.

Пример программирования и применения MTR-инструкции

C000154C

ПРИМЕР ▾

На следующем примере видно, что 3 выхода Y20, Y21 и Y22 включаются один за другим. Этот процесс повторяется постоянно. Данные, воспринимаемые в первой, второй и третьей строках, непрерывно передаются к M30...M37, M40...M47 и M50...M57 и запоминаются в них.

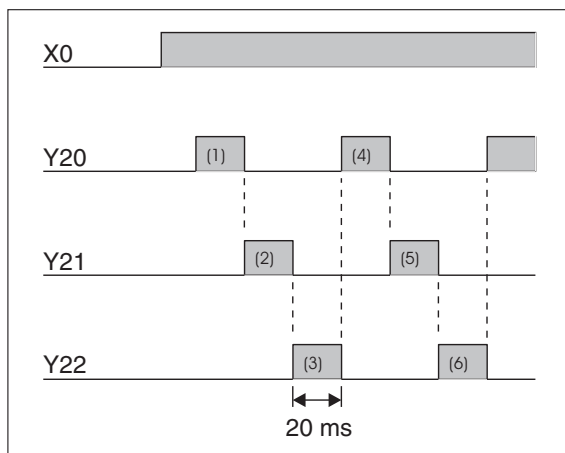


Рис. 6-86.
Пример программирования включения выходов

C000153C

Обработка входов/выходов для каждого выхода выполняется в режиме прерывания с интервалами в 20 мс, причем принимается во внимание время замедления входного фильтра 10 мс.

С помощью MTR-инструкции могут восприниматься 64 входных состояния при применении 8 входов и 8 транзисторных выходов. Все входные данные считываются в пределах 160 мс (20 x 8). Входные данные могут восприниматься в течении 80 мс, если применить входы X0...X7, так как для чтения данных каждой строки требует только 10 мс.

M30...M37 остаются неизменными до тех пор, пока не включатся входные условия. Специальный меркер M8029 включится, как только выполнится матрица. M8029 отключится, если выключатся входные условия.

△

УКАЗАНИЕ

Во избежание программных конфликтов, в качестве входов по возможности не следует использовать адреса X0...X7 (X0...X17 у FX3U).

все же эти адреса применяются, то для каждого выхода согласно Рис. 6-90 последовательно включаются дополнительные резисторы.

ПРИМЕР ▾

Промежуточное включение дополнительных резисторов при использовании входов X0...X7.

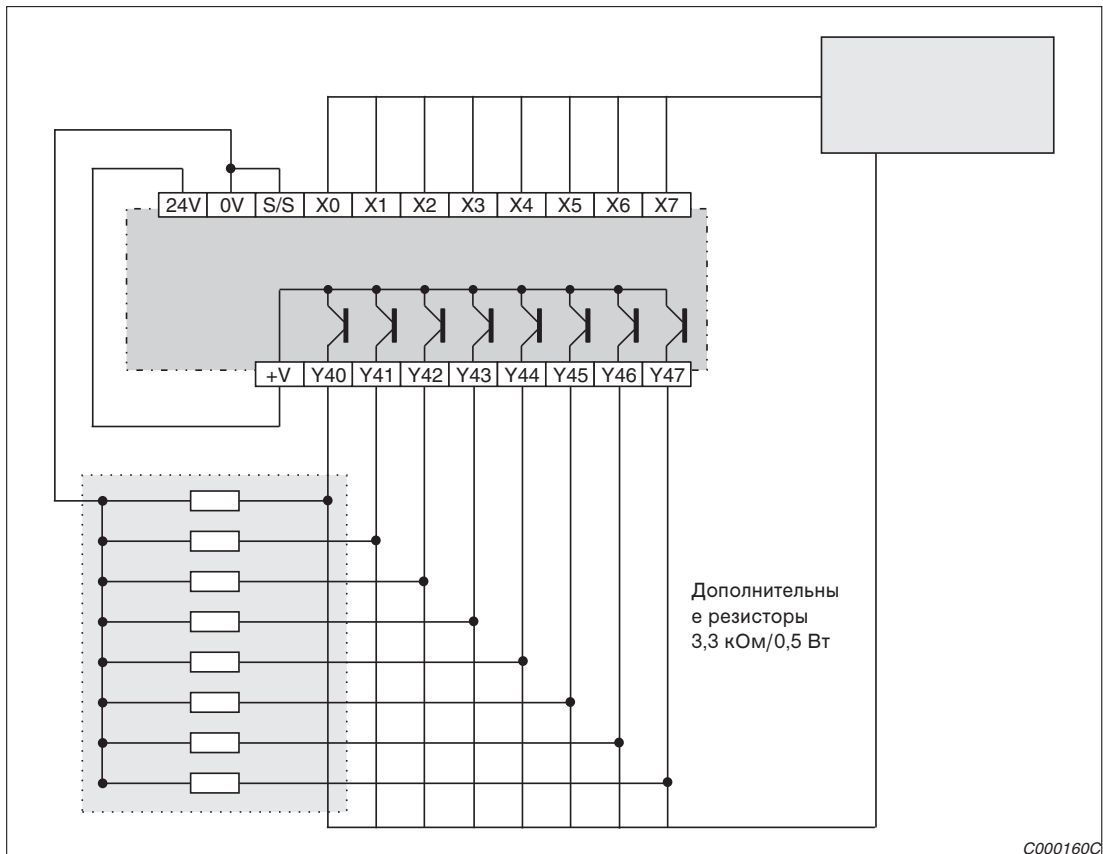


Рис. 6-87. Пример применения Pull-Down-резисторов



6.7.4 СИГНАЛ ВКЛЮЧЕНИЯ И ОТКЛЮЧЕНИЯ ОТ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА (DHSCS, DHSCR)

				DHSCS		FNC 53			
				Включение от высокоскоростного счетчика					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●
Операнды	S1+	S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	C235 – C254	Y, M, S			16бита	32бита	DHSCS	13

				DHSCR		FNC 54			
				Отключение от высокоскоростного счетчика					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●
Операнды	S1+	S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	C235 – C254	Y, M, S, C235 – C254			16бита	32бита	DHSCR	13

ФУНКЦИЯ

Включение и отключение операндов от команды высокоскоростного счетчика. Операнды включаются или выключаются сразу по окончании выполнения инструкции до окончания цикла программы.

ОПИСАНИЕ

- Высокоскоростной счетчик считает изменение состояния на входах счетчика в режиме прерывания. Каждому высокоскоростному счетчику определены жесткие входы счета с жестко определенными функциями.
- В разделе 9.1.9 находится подробное описание работы всех имеющихся высокоскоростных счетчиков и возможности их встройки в программу работы.
- С помощью DHSCS-инструкции могут включаться операнды от команды высокоскоростного счетчика. Операнд, записанный в D+, включается, как только будет достигнуто установленное значение счета.
- С помощью DHSCR-инструкции могут выключаться операнды от команды высокоскоростного счетчика. Операнд, записанный в D+, выключается, как только будет достигнуто установленное значение счета.
- Операнды включаются до окончания цикла программы непосредственно после выполнения инструкции.
- Инструкция выполняется, если данные в S1+ согласованы с данными в S2+. При этом активизация должна выполняться или по импульсу на счетном входе или на входе сброса. Если активизация должна выполняться по входу сброса, то должен быть включен меркер M8025.
- Инструкция не выполняется, если согласование данных между S1+ и S2+ произведено посредством косвенного изменения данных в S1+. Если, например, в S1+ находился регистр данных D0 и значения данных в D0 были изменены инструкцией MOV, то высокоскоростная инструкция не выполнится.

УКАЗАНИЕ

| В программе ПК не может использоваться больше 6 DHSCS и DHSCR инструкций

ПРИМЕР ▾ Применение DHSCS-, DHSCR-инструкций

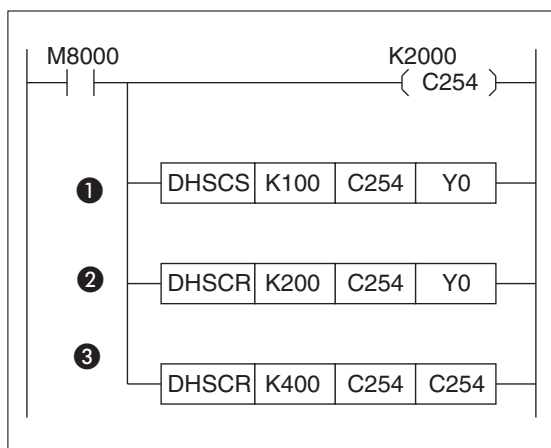


Рис. 6-88.
Пример программирования DHSCS-, DHSCR-инструкций

C000111C

Счетным входом высокоскоростного счетчика C254 является X0 (А-фаза) и X1 (В-фаза). Входом сброса является X2, а входом запуска X3.

- ① Если накопленное значение счетчика C254 изменяется с 99 на 100 или со 101 на 100, то сразу включится Y0.
- ② Если накопленное значение счетчика C254 изменяется с 199 на 200 или с 201 на 200, то сразу отключится Y0.
- ③ Если накопленное значение счетчика C254 изменяется с 399 на 400 или со 401 на 400, то сразу отключится счетчик C254.

△

УКАЗАНИЕ

Выход изменится согласно его физическому времени включения. Внутренние операнды заносятся в соответствующий регистр отображения после выполнения инструкции.

ПРИМЕНЕНИЕ ТОЧКИ ПРЕРЫВАНИЯ СЧЕТЧИКА

- Счетчик-прерывания: I 0 ① 0
- ① Адрес от 1 до 6
Счетчик-прерывания может применяться как операнд для включения (HSCS, FNC 53) или отключения (HSCR, FNC 54). Для отключения Счетчика-прерывания нужно включать меркер M8059.

ПРИМЕР ▾ Точка-прерывания: I030

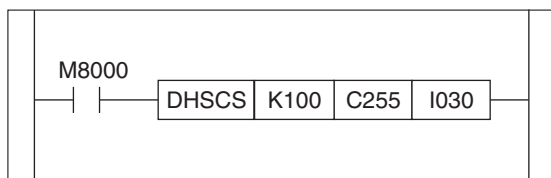


Рис. 6-89.
Пример программирования Счетчика-прерывания

C000333C

Программа-прерывания, вызванная Точкой-прерывания I030, выполняется как только значение высокоскоростного счетчика C255 достигнет заданного значения по константе K100.

△

УКАЗАНИЕ

Обратите внимание на подробную информацию по применению команд включения и отключения от высокоскоростного счетчика.

6.7.5 СРАВНЕНИЕ ОБЛАСТЕЙ (DHSZ)

				DHSZ		FNC 55			
				Сравнение областей					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S1+, S2+		S+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		C235 ... C255	Y, M, S			16бита	32бита	DHSZ
							●		

ПРИНЦИП ФУНКЦИОНИРОВАНИЯ БЕЗ СПЕЦИАЛЬНОГО МЕРКЕРА

Сравнение областей для высокоскоростных счетчиков с областями, ограниченными значениями указанными в (S1+/S2+)

ОПИСАНИЕ

- DHSZ-инструкция сравнивает в режиме прерывания накопленное значение высокоскоростного счетчика с областями, указанными в (S1+/S2+).
- Сравнение выполняется по каждому импульсу счета в указанном в (S+) счетчике.
- Результат сравнения представляется в операндах (D+), ((D+)+1), ((D+)+2).
 - (D+) = S+ < S1+ / S2+; текущее значение меньше области
 - ((D+)+1) = S+ = S1+ / S2+; текущее значение внутри области
 - ((D+)+2) = S+ > S1+ / S2+; текущее значение больше области
- Так как здесь рассматривается высокоскоростная инструкция, то выходы, которые указываются в (D+), сразу выдаются физически.

УКАЗАНИЕ

В программе ПК серии FX могут быть активными одновременно 6 высокоскоростных инструкций типа DHSCS, DHSCR и DHSZ.

ПРИМЕР ▾

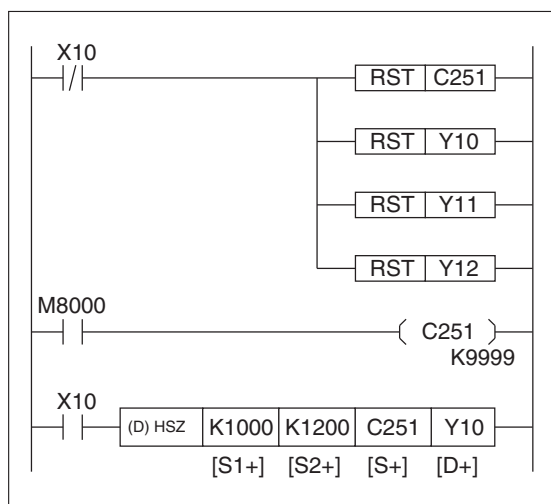


Рис. 6-90. Пример программирования DHSZ-инструкции

- (S1+): окончание быстрого хода (пуск медленного хода)
- (S2+): конец медленного хода (введение торможения)
- (S+): определение высокоскоростного счетчика
- (D+): Y10 -> быстрый ход
Y11 -> медленный ход
Y12 -> торможение

Процессы счета и сравнения, а также внешних выходов выполняются в режиме прерывания.

C000156C

ПРИМЕР ▾

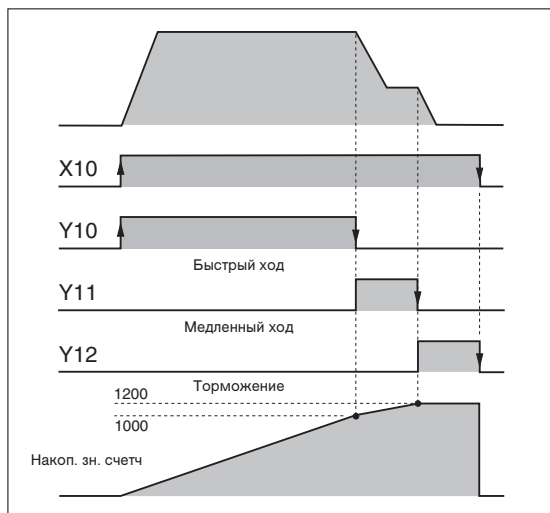


Рис. 6-91.
Временные характеристики включения выходов Y10,Y11,Y12

C000110C

Согласно накопленному в этом примере значению счетчика C251 включаются выходы Y10,Y11,Y12. Если вход X10 выключен, то выходы Y10,Y11,Y12 также выключаются (согласно RST-функции). Если, например, с помощью MOV-инструкции передается новое накопленное значение счетчика, то состояния выходов остаются неизменными до выполнения следующей операции счета.

△

Принцип работы специального маркера M8130

Сравнение таблиц высокоскоростных счетчиков в областях от S1+ и следующих n1

ОПИСАНИЕ

- DHSZ-инструкция со специальным маркером M8130 сравнивает в режиме прерывания накопленное значение высокоскоростного счетчика со значениями, указанными в табличных областях.
- Операнд (D+) для этой специальной функции задается по специальному маркеру M8130.
- Длина таблицы указывается значением констант (K, H). Максимальная длина составляет 128 записей. На каждую запись таблицы дается 4 регистра данных. В каждой записи должны запоминаться следующие данные:
 - сравниваемые значения,
 - адресуемый выход (шестнадцатеричный),
 - инструкция включения или отключения.
- Сравнение выполняется по каждому импульсу счета на указанном счетчике (S+).

УКАЗАНИЕ

В программе ПК серии FX могут одновременно быть активными только 6 высокоскоростных инструкции типа DHSCS и DHSCR или DHSZ.

Всегда может быть применена только одна DHSZ-инструкция специального маркера M8130.

Должна быть корректно введена HSC-инструкция.

DHSZ-инструкция со специальным маркером M8130 первый раз выполняется по первой END-инструкции. ПК позволяет сделать это, поскольку он внутренне создает таблицу сравнения.

Сравнение в таблице всегда происходит последовательно. По этой причине сравниваемые значения всегда должны быть отсортированы в возрастающей или уменьшающейся последовательности.

ПРИМЕР ▾

DHSZ-инструкции со специальным меркером M8130

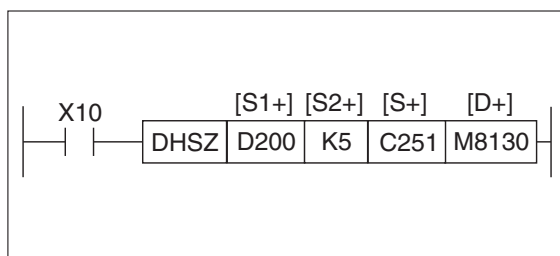


Рис. 6-92.
Пример программирования DHSZ-инструкции со специальным меркером M8130

C000322C

После срабатывания входа X10 выполняется сравнение таблицы от регистра данных D200 в 5-ти записях таблицы данных со значением в счетчике C251.

Таблица имеет следующую форму (K1 = включение, K0 = отключение):

Номер записи	Сравниваемые значения		Выход-ная инструкция	Включение/отключение
	нижнее	верхнее		
0	D200	D201	D202	D203
	K123		H10 (=Y10)	K1
1	D204	D205	D206	D207
	K234		H10	K0
2	D208	D209	D210	D211
	K345		H23 (=Y23)	K1
3	D212	D213	D214	D215
	K456		H23	K0
4	D216	D217	D218	D219
	K567		H23	K1

Табл. 6-21.
Таблица сравнения, стартовый адрес D200, длина K5

Значения сравнения запоминаются в 32-х битном формате (двойное слово). Выход выдается в шестнадцатиричном формате.

ПРИМЕР ▾

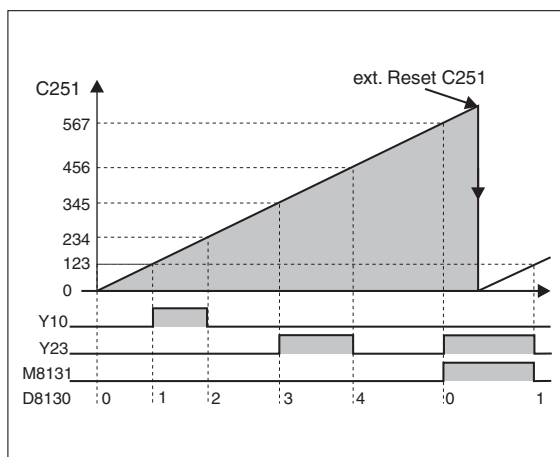


Рис. 6-93.
Временные процессы для включения выходов Y10, Y23

C000323C

Если DHSZ-инструкция применяется со специальным меркером M8130, то регистр данных 8130 определяется как счетчик номера записи. После каждого сравнения D8130 переходит к следующему номеру записи.

Если все записи таблицы обработаны, то включается флаг конца процесса M8131 и D8130 выключается внешним импульсом или импульсом из программы. D8130 стартует вновь согласно счету, если отключается флаг M8131.

Принцип работы специального маркера M8132

Сравнение таблиц высокоскоростных счетчиков в областях от S1+ и следующих n1 с зависимым от результата управлением частотой в DPLSY-инструкции.

ОПИСАНИЕ

- DHSZ-инструкция со специальным меркером M8132 сравнивает в режиме прерывания накопленное значение высокоскоростного счетчика со значениями, указанными в табличных областях. При согласовании выдается значение, указанное в таблице, по следующей DPLSY-инструкции (FNC 57) управления частотой.
- Операнд (D+) для этой специальной функции задается по специальному меркеру M8132.
- Длина таблицы указывается значением констант (K,H). Максимальная длина составляет 128 записей. На каждую запись таблицы дается 4 регистра данных. В каждой записи должны запоминаться следующие данные:
 - сравниваемые значения (32-х битный формат),
 - адресуемый выход (32-х битный формат).
- Сравнение выполняется по каждому импульсу счета на указанном счетчике (S+).

УКАЗАНИЕ

- | В программе ПК серии FX могут одновременно быть активными только 6 высокоскоростных инструкции типа DHSCS и DHSCR или DHSZ.
- | Всегда может быть применена только одна DHSZ-инструкция специального меркера M8132.
- | Должна быть корректно введена HSC-инструкция.
- | DHSZ-инструкция со специальным меркером M8132 первый раз выполняется по первой END-инструкции. ПК позволяет сделать это, поскольку он внутренне создает таблицу сравнения.
- | Сравнение в таблице всегда происходит последовательно. По этой причине сравниваемые значения всегда должны быть отсортированы в возрастающей или уменьшающейся последовательности.
- | Последняя запись в таблице должна выполняться на (K0, K0), чтобы гарантировать, что остановлена выдача импульсов и D8131 не перескочит к началу таблицы. Регистры D8134 и D8135 удерживают значение K0 и указывают конец таблицы.

ПРИМЕР ▾

Применение DHSZ-инструкции со специальным меркером M8132

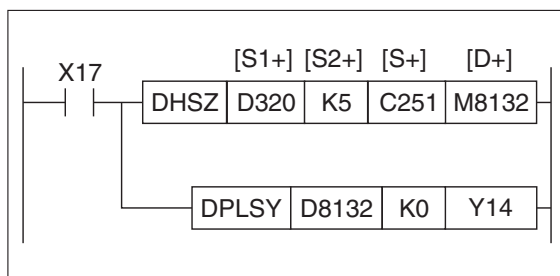


Рис. 6-94.
Пример программирования DHSZ-инструкции со специальным меркером M8132

C000324C

После срабатывания входа X10 выполняется сравнение таблицы от регистра данных D200 в 5-ти записях таблицы данных со значением в счетчике C251.

Таблица имеет следующую форму:

Номер записи	Сравниваемые значения		Выходная частота	
	нижнее	верхнее	нижнее	верхнее
0	D320	D321	D322	D323
	K20		K300	
1	D324	D325	D326	D327
	K600		K500	
2	D328	D329	D330	D331
	K700		K200	
3	D332	D333	D334	D335
	K800		K100	
4	D336	D337	D338	D339
	K0		K0	

Табл. 6-22.

Таблица сравнения, стартовый адрес D320, длина K5

Значения сравнения запоминаются в 32-х битном формате (двойное слово). Выходное значение выдает частоту, которая сохраняется до тех пор, пока не будут согласованы результаты сравнения.

ПРИМЕР ▾

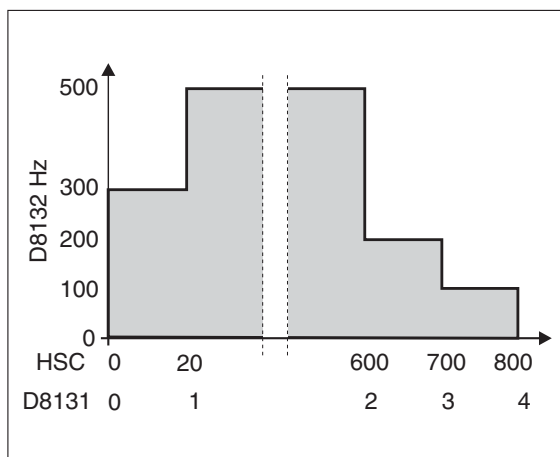


Рис. 6-95.

Изменение частоты при применении DHSZ-инструкции со специальным меркером M8132

C000325C

Если DHSZ-инструкция применяется со специальным меркером M8132, то регистр данных 8131 определяется как счетчик номера записи. После каждого сравнения D8131 переходит к следующему номеру записи.

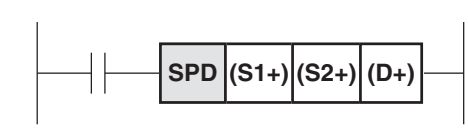
D8132 содержит соответствующее значение частоты записей таблицы для применения в PLSY-инструкции. Для обработки регистры данных D8134 и D8135 содержат сравниваемые значения.

Если все записи таблицы обработаны, то включается флаг конца процесса M8133 и D8131 выключается внешним импульсом или импульсом из программы. D8131 стартует вновь согласно счету, если отключается флаг M8133.

При отключении DHSZ-инструкции отключаются все значения, включая выдаваемую частоту.



6.7.6 ОПРЕДЕЛЕНИЕ СКОРОСТИ (SPD)

				SPD		FNC 56			
				Определение скорости					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
		●	●	●	●	●			
Операнды	S1+	S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	X0 ... X5	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	T, C, D		16бита	32бита	SPD	7	
					●				

ФУНКЦИЯ

Фиксирование числа импульсов в течении заданного времени

ОПИСАНИЕ

- Импульсы на (S1+) подсчитываются за время в (S2+) в мс и результат записывается в (D+).
- Задействуются операнды (D+), ((D+)+1), ((D+)+2).
 - (D+): сумма импульсов после отсчета времени
 - ((D+)+1): текущее значение времени внутри интервала времени
 - ((D+)+2): остающееся отсчитываемое время

УКАЗАНИЕ

После отсчета времени содержание ((D+)+1) передается в (D+), а само ((D+)+1) отключается.

Входы высокоскоростного счетчика, используемые в инструкции, не могут применяться в других высокоскоростных операциях.

Для каждого высокоскоростного входа можно задать максимум одну SPD-инструкцию.

ПРИМЕР ▾

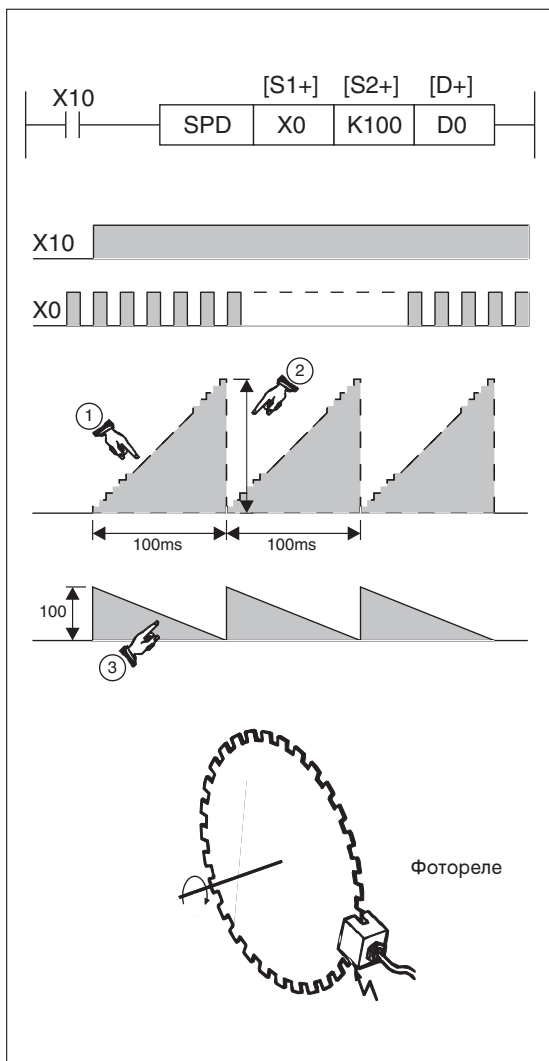


Рис. 6-96.
Пример программирования SPD-инструкции

- ①: действительное текущее набираемое значение импульсов (D1)
- ②: накопленное значение счетчика за установленный интервал времени (D0)
- ③: остающееся в (D2) время до передачи данных из (D1) в (D0)

В примере D1 считает количество включений X0. После 100 мс результат счета сохраняется в D0.

D1 отключается и начинает вновь счет включений X0.

В D2 соответственно измеряется оставшееся время.

С помощью этого значения можно определить число оборотов привода.

$$N = \frac{60 \times D0}{n \times t} \times 10^3 \text{ (об/мин.)}$$

n: импульсов на оборот

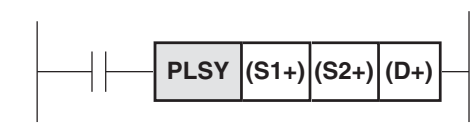
N: скорость

t: интервал времени (мс), который указывается в S2+



C000158C

6.7.7 ВЫДАЧА ОПРЕДЕЛЕННОГО ЧИСЛА ИМПУЛЬСОВ (PLSY, DPLSY)

		PLSY		FNC 57			
		Выдача определенного числа импульсов					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S1+, S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Только Y0 или Y1		16бита	32бита	PLSY	7
				●	●	DPLSY	13

ФУНКЦИЯ

Определенное число импульсов с жестко заданной частотой и жестким соотношением ширины импульса 50:50 выдаваемое на выход

ОПИСАНИЕ

- Инструкция формирует определенное число импульсов.
- В (S1+) устанавливается частота
 - команда PLSY: 1...32767 Гц (FX1S/FX1N/FX2N/FX2NC/FX3U)
 - команда DPLSY: 1...100 кГц (FX1S/FX1N), 2...20 кГц (FX2N/FX2NC), 1...200 кГц (FX3U)
- В (S2+) указывается число создаваемых импульсов. При этом не должны превышать следующие значения областей.
 - 16-ти битовые инструкции: от 1 до 32767 импульсов
 - 32-х битовые инструкции: от 1 до 2 147 483 647 импульсов

Если указано значение 0, то создается последовательный ряд импульсов.
- В (D+) определяется адрес выхода.
- Соотношение между включенным и отключенным состоянием: 50% ВКЛ, 50% ОТКЛ. Включенное и отключенное состояния выдаются непосредственно в режиме прерывания.
- При применении DPLSY-инструкции число импульсов задается в двух следующих друг за другом регистров данных.
- Если желаемое число импульсов создано, то включается специальный меркер M8029 (инструкция полностью отработана). M8029 отключается, если деактивируется DPLSY-инструкция.

Данные в (S1+) (частота) могут изменяться во время выполнения инструкции. Однако изменение данных в (S2+) (число импульсов) может применяться лишь тогда, если инструкция уже отработана.

УКАЗАНИЕ

- | В программе допускается использование не более 2 инструкций PLSY.
- | Возможно применение одновременно инструкций PLSY и PLSR (FNC59), если данные инструкций подключены к разным выходам.
- | Импульсы могут выдаваться только на выходах Y0 и Y1.
- | Возможно использование в подпрограммах нескольких инструкций PLSY, тем не менее должна остановиться запущенная команда, прежде чем в подпрограмме начинается следующая команда PLSY.

| ПК должно иметь транзисторные выходы.

| Для нормального функционирования контроллера на максимальной частоте, ток нагрузки на выходе контроллеров серий FX2N и FX3U должен быть не меньше 200 мА. У контроллеров серий FX1S и FX1N ток должен находиться в диапазоне между 10 и 100 мА.

| Применение DHSZ-инструкции смотри в разделе Рис. 6-97.

ПРИМЕР ▾

Применение PLSY-инструкции формирования группы импульсов

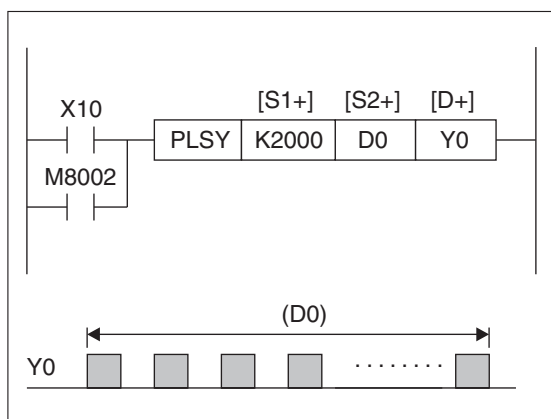


Рис. 6-97.
Пример программирования PLSY-инструкции

C000105C

Если включено X10, то создаются импульсы с частотой 2000 Гц. Всегда создается столько импульсов, сколько указано в регистре данных D0 (S2+).

Создание импульсов прекращается, если выключается X10. Если X10 включается снова, то операция начинается вновь. Если X10 не включено, то выключается Y0.

△

УКАЗАНИЕ

| У ПК серии FX2N двойное слово в D8140 и D8141 показывают текущие импульсы Y0. Двойное слово D8142 и D8143 показывает импульсы Y1.

| В регистры D8136 и D8137 серий FX2N, FX2NC и FX3U записывается сумма импульсов на Y0 и Y1.

6.7.8 ВЫДАЧА ИМПУЛЬСОВ С МОДУЛЯЦИЕЙ ШИРИНЫ ИМПУЛЬСА (PWM)

		PWM		FNC 58			
		Выдача импульсов с модуляцией ширины импульса					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S1+, S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z (S1 ≤ S2)	Только Y0 или Y1		16бита	32бита	PWM	7
				●			

ФУНКЦИЯ

Последовательный ряд импульсов с жестко заданной шириной импульса и продолжительностью периода выдается на выход

ОПИСАНИЕ

- Инструкция формирует последовательный ряд импульсов. Соотношение между шириной импульса t и длительностью периода $T0$ задается:
 t : ширина импульса (мс)
 $T0$: продолжительность периода (мс)
 Частота f : $1/T0$ (кГц)
- В (S1+) устанавливается ширина импульса в области t : от 1 до 32767 мс. Ширина импульса должна лежать в пределах $1 \leq t \leq T0$.
- В (S2+) устанавливается длительность периода $T0$: от 1 до 32767 мс.
- В (D+) указывается адрес выхода.

УКАЗАНИЕ

- | Инструкция может использоваться в программе только один раз.
- | Могут выдаваться только выходы Y0 и Y1.
- | ПК должно иметь транзисторные выходы, чтобы избежать износа контактов.
- | Для нормальной работы контроллера при минимальной длительности периода $T0$ (1 мс), ток нагрузки выходов у контроллеров серий FX2N и FX3U должен быть не меньше 200 мА. У контроллеров серий FX1S и FX1N ток должен находиться в диапазоне между 10 и 100 мА.

ИСТОЧНИК ОШИБОК

Если установленное значение ширины импульса в (S1+) больше, чем жестко определенное значение для $T0$ в (S2+), то появляется ошибка в работе программы.

ПРИМЕР ▾

Применение PWM-инструкции моделирования импульсов

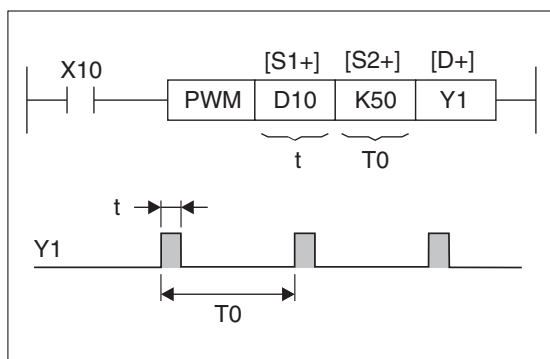


Рис. 6-98.
Пример программирования
PWM-инструкции

C000113C

Благодаря изменению данных в регистре D10 в области от 0 до 50 можно варьировать относительную ширину импульса T0 от 0 до 100%. Если значение D10 устанавливается на 0, то не выдается никаких импульсов. Если значение D10 изменяется на 50, то Y1 вводится для всего цикла.

Y1 выключается, если выключается X10.

△

6.7.9 ВЫДАЧА ОПРЕДЕЛЕННОГО ЧИСЛА ИМПУЛЬСОВ (PLSR)

		PLSR		FNC 59				
		Выдача определенного числа импульсов						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●	●	
Операнды	S1+, S2+, S3+	D+	Имп. инструкция (P)		Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z (S1 ≤ S2)	Только Y0 или Y1			16бита	32бита	PLSR	9
					●	●	DPLSR	17

ФУНКЦИЯ

Жестко заданное число импульсов с жестко заданной частотой выдается на выход.

ОПИСАНИЕ

- PLSR-инструкция создает на выходе заданное число импульсов (S2+) с заданной частотой (S1+). Частота по десять шагов изменяется вверх в начале работы инструкции (разгон) и вниз в конце за заданное время (S3+).
- Контроллеры серий FX2N и FX2NC могут выдавать частоту между 10 и 20.000 Гц. Контроллеры серий FX1S и FX1N могут выдавать частоты от 10 до 100.000 Гц. У контроллеров серии FX3U возможны частоты от 10 до 200.000 Гц. Указанная частота должна быть кратной 10. Если указанная частота не кратна 10, частота округляется вверх до кратного значения.
- Ширина шага наклонной составляет 1/10 указанной выходной частоты (при применении шагового двигателя это нужно учитывать).
- Максимальное количество выходных импульсов у серии FX3U и FX2N, а также серии FX2NC, начиная с версии 3.00:
16-битная команда: 1...32.767 импульсов
32-битная команда: 1...2.147.483.647 импульсов
- Время подъема ramпы должно соответствовать ниже описанным граничным значениям.
- В качестве выходов могут программироваться только Y0 и Y1.

УКАЗАНИЕ

В одной программе в то же самое время могут применяться две PLSR-инструкции с передачей импульсов на Y0 и Y1. Возможно также применение PLSY-инструкции (FNC57) и PLSR-инструкции (FNC59) в одном цикле с передачей импульсов на Y0 и Y1. Многократное применение может реализовываться по подпрограмме или подобных методах

Если число указанных импульсов недостаточно, чтобы получить указанную частоту, частота срезается.

Специальный меркер M8029 включается после выдачи указанного числа импульсов. Отключение меркера выполняется при отключении условий выполнения PLSR-инструкции.

ОГРАНИЧЕНИЕ ВРЕМЕНИ НАКЛОНА

Время наклона (S3+) ограничивается 5 000 мс. Граничное значение времени наклона в зависимости от частоты и числа выходных импульсов рассчитывается следующим образом:

- Значение в (S3+) должно быть минимум в 10 раз больше, чем время цикла программы (D8012).
- Минимальное значение для (S3+) рассчитывается по уравнению:
 $(S3+) \leq (9.000 / (S1+)) \times 5$
- Максимальное значение для (S3+) рассчитывается по уравнению:
 $(S3+) \leq ((S2+) / (S1+)) \times 818$
- Если параметр выходит за расчетные границы, то значение (S1+) (частота) уменьшается.
- Подъем выходной частоты происходит в 10 шагов.

УКАЗАНИЕ

Максимальная выходная частота и ширина шага частоты ограничиваются внутри области 2...20 000 Гц.

После отключения условий выполнения PLSR-инструкции выключаются сработавшие выходы. При новом включении условий выполнения снова начинается обработка инструкции.

Если во время обработки изменяется операнд, профиль (характер) выхода сохраняется. Изменения операнда начинают действовать при следующей обработке инструкции.

ПРИМЕР ▾

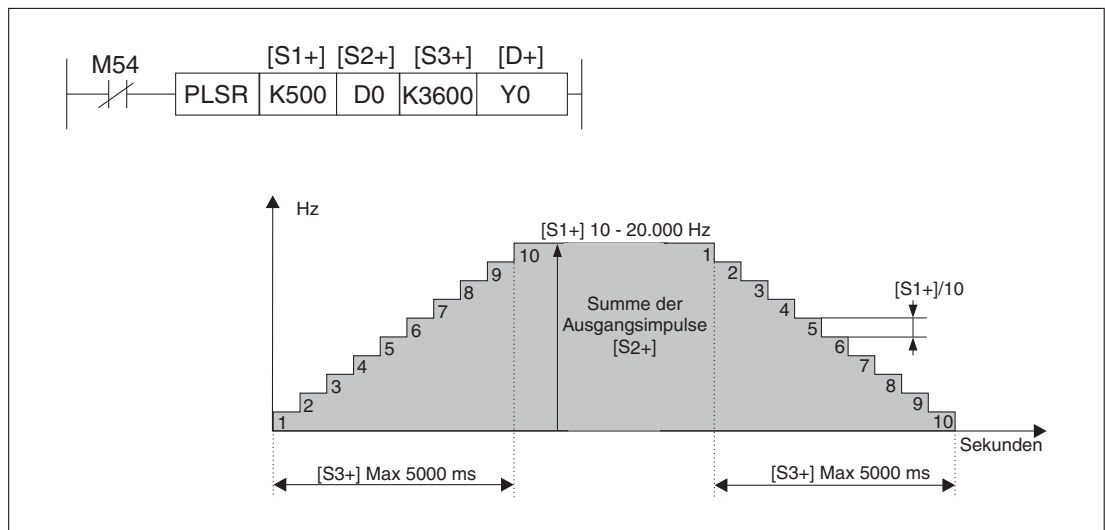


Рис. 6-99. Пример программирования PLSR-инструкции

При отключении меркера M54 число импульсов, указанное в D0 (S2+) выдается на Y0 (D+).

Выходная частота составляет 500 Гц (S1+).

Возрастание частоты до 500 Гц (S1+) и снижение частоты до 0 выполняется соответственно за 3600 мс (S3+) шагами по 50 Гц (S1+ / 10).



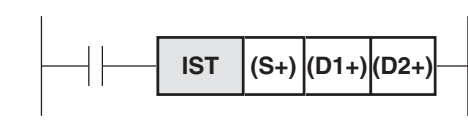
6.8 ИНСТРУКЦИИ ПОЛЬЗОВАТЕЛЯ

ОБЗОР ИНСТРУКЦИЙ FNC 60...69

СИМВОЛ	FNC	НАЗНАЧЕНИЕ	РАЗД.
IST	60	Инициализация статуса шага	6.8.1
SER	61	Инструкция поиска	6.8.2
ABSD	62	Сравнение абсолютного счетчика	6.8.3
INCD	63	Сравнение инкрементального счетчика	6.8.4
TTMR	64	Таймер обучения	6.8.5
STMR	65	Специальный таймер	6.8.6
ALT	66	Функция пульс-пары (Flip-Flop)	6.8.7
RAMP	67	Функция наклонных характеристик	6.8.8
ROTC	68	Позиционирование поворотного стола	6.8.9
SORT	69	Инструкция сортировки	6.8.10

Табл. 6-23. Обзор инструкций FNC 60...69

6.8.1 ИНИЦИАЛИЗАЦИЯ СТАТУСА ШАГА (IST)

		IST		FNC 60			
		Инициализация шаговых состояний					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	S+	D1+, D2+	Имп. инструкция (P)		Обработка		Шаги программы
	X, Y, M, S	FX1S: S20-S127; FX1N: S20-S999 FX2N: S20-S999 (D1+ < D2+)		16бита	32бита	IST	7
				●			

ФУНКЦИЯ

Включение специальных функций и резервирование операндов шаговых состояний для шагового управления. С помощью IST-инструкции могут связываться различные шаговые цепи на пульте управления. Так, могут инициализироваться шаговые цепи для автоматического и наладочного режимов работы и режима возврата в нулевую точку (в исходное положение).

ОПИСАНИЕ

- Операнды шаговых состояний S0...S2 предназначены (резервируются) для инициализации шаговых цепей
 - наладочный режим работы,
 - автоматический режим и
 - перемещение в нулевую точку.
- Операндам шаговых состояний S0...S2 не нужны SET-инструкции.
- Операнды шаговых состояний S3...S9 остаются свободными.
- Операнды шаговых состояний S10...S19 резервируются для возврата в нулевую точку.
- Для программирования остальных шаговых цепей в распоряжении имеются оставшиеся операнды шаговых состояний S20...S127 (S999).
- Специальные маркеры M8040...M8043, M8045 и M8047 управляют шаговыми цепями.
- В (D1+) и (D2+) определяется область операндов шаговых состояний для шаговых цепей автоматического режима работы. При этом должно выполняться условие: (D1+) < (D2+).
- В (S+) определяется область управляющих входов. В качестве управляющих входов могут применяться операнды X, Y или M. Они указывают стартовые адреса областей операндов.

УКАЗАНИЕ

| IST-инструкция может применяться в программе только один раз.

ПРИМЕР ▾

Применение IST-инструкции

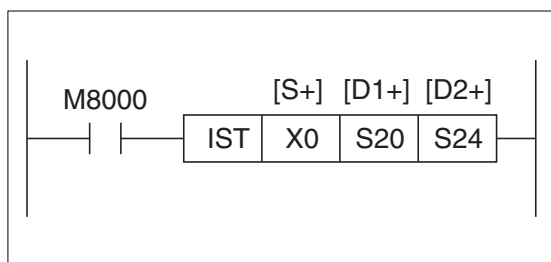


Рис. 6-100.

Пример программирования IST-инструкции

C000157C

В этом примере входы содержат следующие специальные функции:

- X0: наладочный режим работы
- X1: возврат в исходную позицию
- X2: шаговый режим работы
- X3: режим работы отдельными операциями
- X4: автоматический режим работы
- X5: кнопка возврата в исходную позицию
- X6: кнопка запуска автоматического режима работы
- X7: кнопка останова автоматического режима работы



УКАЗАНИЕ

Эти управляющие входы служат для непосредственного управления процессом работы. Ими выбирается нужный режим работы.

Следующие специальные меркеры благодаря IST-инструкции влияют или управляют обработкой шаговых цепей:

- M8040: дальнейшее переключение в другой шаг возможно
- M8041: начало дальнейшего переключения (сбрасывается при STOP контроллера)
- M8042: пусковой импульс (сбрасывается при STOP контроллера)
- M8043: достигнута нулевая точка
- M8045: заблокировать сброс всех выходов
- M8047: показать состояние STL (устанавливается при обработке команды END)

ПРИМЕР ▾

Передвижной контейнер для транспортировки сыпучих материалов в автоматическом режиме должен непрерывно загружаться и разгружаться.

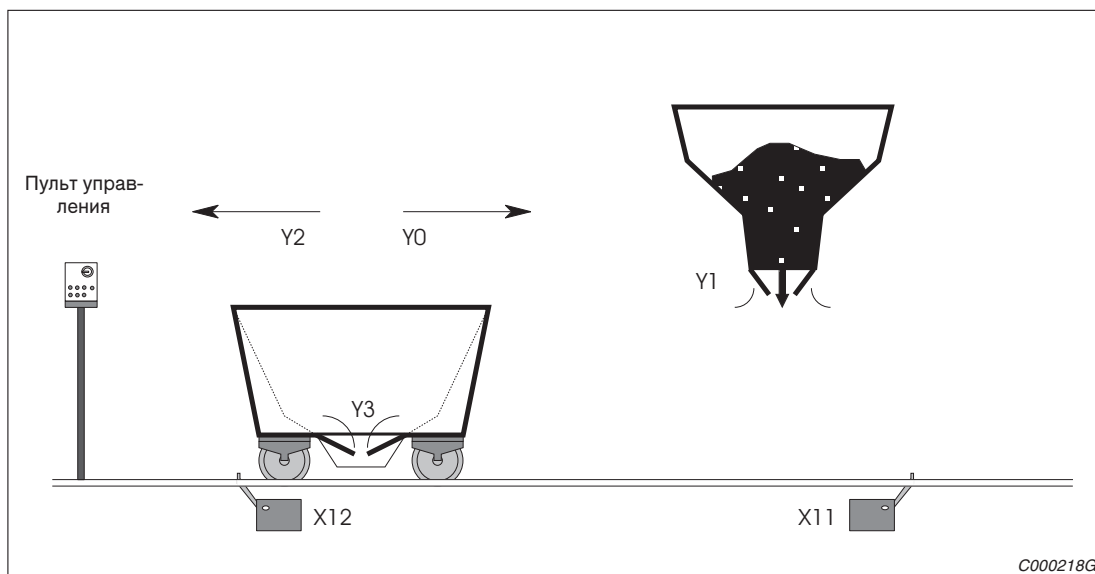


Рис. 6-101. Пример, контроль загрузки и разгрузки контейнера с помощью IST-инструкции

ОПИСАНИЕ ДАТЧИКОВ СИГНАЛОВ И ИСПОЛНИТЕЛЬНЫХ АППАРАТОВ

- Конечные выключатели
слева: X12
справа: X11
- Направление движения контейнера (электромагниты)
слева: Y2
справа: Y0
- Открылась загрузка материала на 7 секунд: Y1
- Открылись разгрузочные дверцы контейнера на 5 секунд: Y3

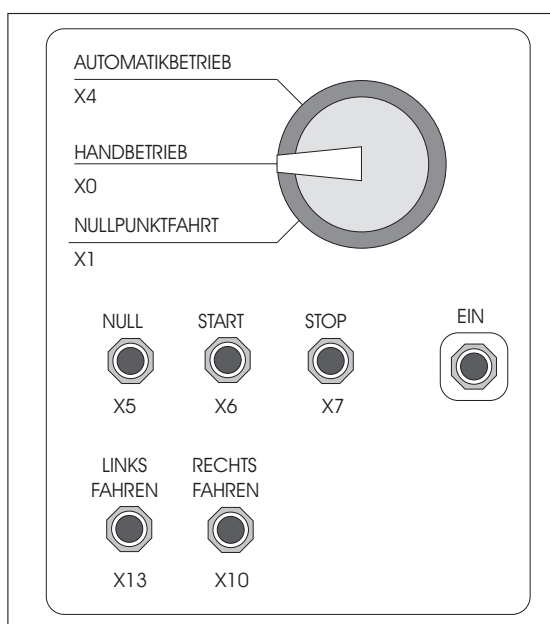


Рис. 6-102.
Пример пульта управления

C000217G

ПРИМЕР ▾

При применении IST-инструкции действуют предварительно задаваемые специальные функции:

- X0: наладочный режим работы
- X1: возврат в исходную позицию
- X2: шаговый режим работы
- X3: режим работы отдельными операциями
- X4: автоматический режим работы
- X5: кнопка возврата в исходную позицию
- X6: кнопка запуска автоматического режима работы
- X7: кнопка останова автоматического режима работы

ОПИСАНИЕ ФУНКЦИЙ

В режиме автоматики перемещаемый контейнер должен непрерывно перемещаться между загрузочной и разгрузочной станциями. Автоматический режим запускается, если включается X4 и выполняется воздействие на X6. В качестве других условий включения нужно, чтобы контейнер находился в исходном положении (нулевая точка) (M8040 и M8044 активны). Автоматический режим прекращается с помощью X7.

В наладочном режиме контейнер может перемещаться в обоих направлениях. Для этого должен быть включен X0. Контейнер при воздействии на кнопки X13 или X10 перемещается влево или вправо.

Перемещение в нулевую точку запускается включением X1 и воздействием на X5. Благодаря этому контейнер из любой позиции возвращается в исходную позицию.

Исходная позиция достигается, если контейнер находится на позиции конечного выключателя X12 и опорожнен .

△

ПРИМЕР ▾

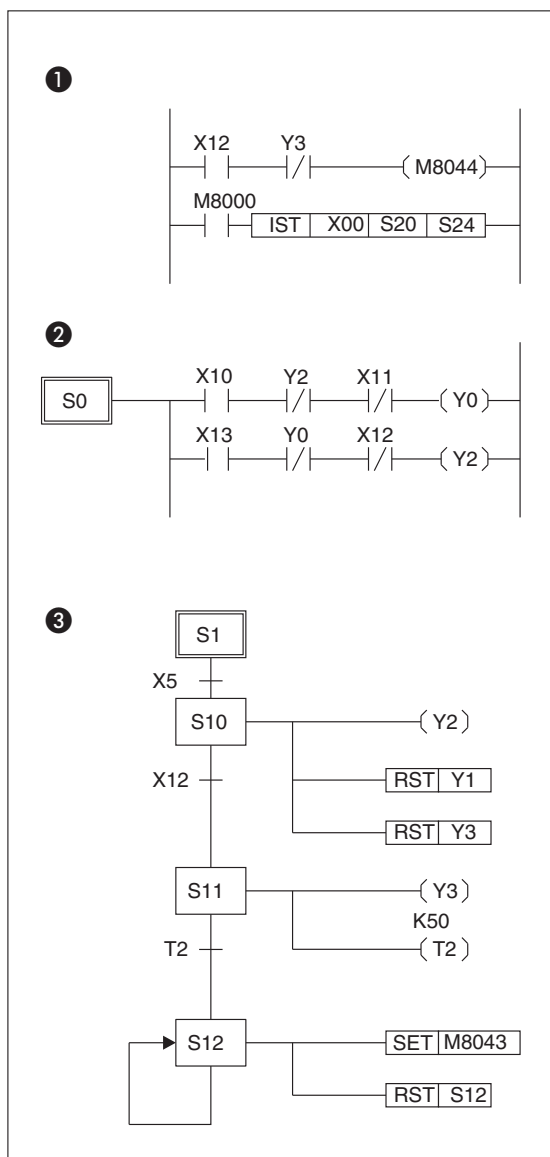


Рис. 6-103.

Пример программирования контроля загрузки и разгрузки перемещаемого контейнера по IST-инструкции

Требуются следующие три главные области работы:

- ① Инициализация
- ② Наладочный режим работы
- ③ Возврат в нулевую точку

Шаговая цепь для наладочного режима работы позволяет вручную задавать команды перемещения подвижного контейнера.

Показанная шаговая цепь режима перемещения в нулевую точку возвращает контейнер из любого положения в исходную позицию. Там контейнер опорожняется.

C000213G

ПРИМЕР ▾

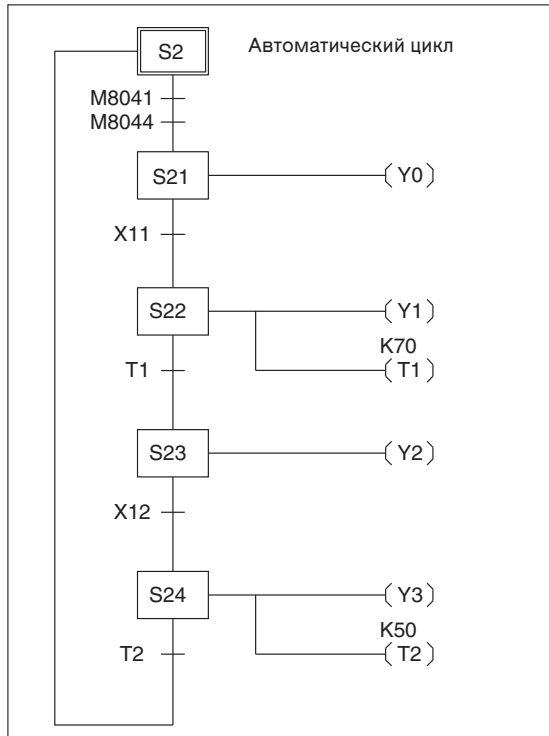


Рис. 6-104.

Пример программирования контроля загрузки и разгрузки перемещаемого контейнера по IST-инструкции (автоматика)

C000214G

Инициализация, условия получения нулевой точки	Подход к нул. позиции	Автоматич. режим	Наладочный режим работы
LD X 12	STL S 1	STL S 2	STL S 0
ANI Y 3	LD X 5	LD M8041	LD X 10
OUT M8044	SET S 10	AND M8044	ANI Y 2
LD M8000	STL S 10	SET S 21	OUT Y 0
IST	RST Y 1	STL S 21	LD X 13
X 0	RST Y 3	OUT Y 0	ANI Y 0
S 20	OUT Y 2	LD X 11	OUT Y 2
S 24	LD X 12	SET S 22	RET
	SET S 11	STL S 22	END
	STL S 11	OUT Y 1	
	OUT Y 3	OUT T 1	
	OUT T 2	K 70	
	K 50	LD T 1	
	LD T 2	SET S 23	
	SET S 12	STL S 23	
	STL S 12	OUT Y 2	
	SET M8043	LD X 12	
	RST S 12	SET S 24	
	RET	STL S 24	
		OUT Y 3	
		OUT T 2	
		K 50	
		LD T 2	
		OUT S 2	
		RET	

C000215G

Рис. 6-105. Пример программирования на языке листинга инструкций для контроля загрузки и разгрузки перемещаемого контейнера по IST-инструкции



6.8.2 ИНСТРУКЦИЯ ПОИСКА (SER)

					SER		FNC 61			
					Инструкция поиска					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●	●	●
Операнды	S1+	S2+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D	K, H, D		●	16бита	32бита	SER/SERP	9
							●	●	DSER/DSERP	17

ФУНКЦИЯ

Сквозной поиск областей данных по значению поиска

ОПИСАНИЕ

- Области данных от (S1+) до ((S1+)+n) просматриваются для поиска ключа указанного в (S2+) и результат поиска заносится в регистры данных, начиная с (D+). Одновременно определяются и запоминаются самое малое и самое большое значения в области поиска.
- Длина области поиска n для данных в 16-ти битовом формате определена по максимуму в 256, а для данных в 32-х битовом формате определена по максимуму в 128.
- Результат поиска заносится в 5 регистров данных, при 32-х битовом формате в 10 регистров данных. Они содержат:
 - количество значений, согласованных с ключом поиска, в области поиска (0 при отсутствии согласования с ключом поиска)
 - позиция первого согласованного значения (0 при несогласованности)
 - последнего согласованного значения (0 при несогласованности)
 - позиция самого малого встретившегося в области поиска значения. Если это значение появляется многократно, то запоминается последняя позиция.
 - позиция самого большого встретившегося в области поиска значения. Если это значение появляется многократно, то запоминается последняя позиция.

ПРИМЕР ▾

Применение SER-инструкции

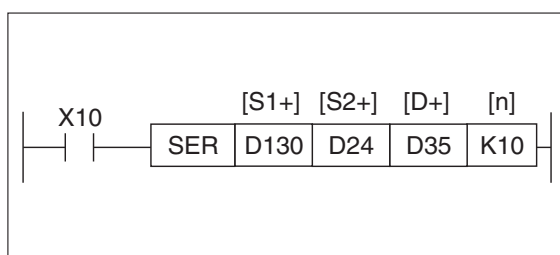


Рис. 6-106.
Пример программирования SER-инструкции

C000326C

В примере сверху показан поиск по коду в D24 = K100, начиная с D130 при длине K10. Результат заносится в D35 до D39 (5 регистров при 16-ти битовом формате).

Область поиска может отображаться следующим образом:

Позиция	Перечень найденного	Равные результаты поиска	Максимум	Минимум
0	D130 = K100	●		
1	D131 = K111			
2	D132 = K100	●		
3	D133 = K98			
4	D134 = K123			
5	D135 = K66			●
6	D136 = K100	●		
7	D137 = K95			
8	D138 = K78			
9	D139 = K210		●	

Табл. 6-24.
Область поиска

Результаты поиска могут представляться следующим образом:

Перечень результатов	Содержание	Значение
D35	3	Количество результатов поиска "="
D36	0	Первая позиция согласована
D37	6	Последняя позиция согласована
D38	5	Позиция наименьшего значения
D39	9	Позиция наибольшего значения

Табл. 6-25.
Результат поиска



6.8.3 АБСОЛЮТНЫЙ МНОГОУСТАВОЧНЫЙ СЧЕТЧИК (ABSD)

					ABSD		FNC 62		
					Сравнение абсолютного счетчика				
CPU					FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●
Операнды	S1+	S2+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS ^① , T, C ^② , D	C ^②	Y, M, S	K, H n ≤ 64		16бита	32бита	ABSD	9
						●	●	DABSD	17

- ① X, Y, M, S должны быть кратны 8.
- ② При 32-битной команде (DABSD) высокоскоростные счетчики не допускаются.

ФУНКЦИЯ

Включение бита в зависимости от состояния счетчика

ОПИСАНИЕ

- Накопленное (действительное) значение счетчика (S2+) сравнивается с таблицей значений входов/выходов.
- Таблица образуется из словных операндов. Первым операндом является (S1+). Таблица содержит n строк.

Значение включения	Значение отключения	Включаемый операнд
(S1+)	(S1+) +1	D
(S1+) +2	(S1+) +3	D + 1
(S1+) +4	(S1+) +5	D + 2
⋮	⋮	⋮
(S1+) + (2n+1)	(S1+) + 2n	D + 3

Рис. 6-107.

Таблица словных операндов для 16-битной команды ABSD

- Если счетчик (S2+) получил накопленное значение, занесенное в таблицу, включается относящийся к нему операнд.
- Значения таблицы при 16-битной команде должны находиться в диапазоне 0...32.767, а при 32-битной команде (DABSD) - в диапазоне 0...2.147.483.647.
- Значение может, например, быть записанным с помощью MOV-инструкции.

УКАЗАНИЕ

Всегда занимают 2 счетчика, (S2+) и ((S2+)+1). В (S1+) должны использоваться четные адреса операндов. Если в (S1+) указываются битовые операнды, адрес операнда должен быть кратен 16 (16, 32, 64 и т. п.) или равен 0. В команде ABSD в этом случае допускается только коэффициент K4, а в команде DABSD - только коэффициент K8.

ABSD-инструкция может использоваться в программе только один раз.

ПРИМЕР ▾

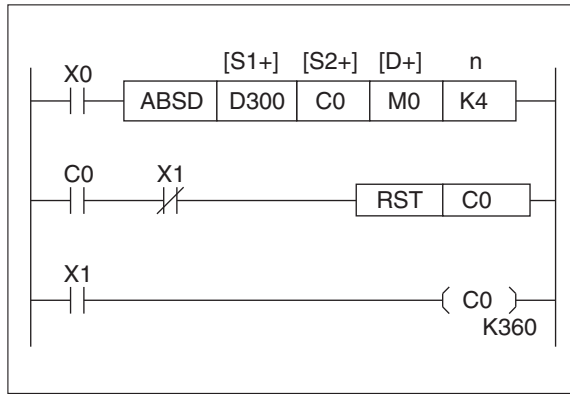


Рис. 6-108.
Пример программирования ABSD-инструкции

C000163C

В примере программирования на рисунке контролируется состояние ВКЛ./ОТКЛ. меркеров от M0 до M3 при вращении поворотного стола (см. также раздел 6.8.9).

В (S1+) адрес операндов X, Y, M и S должен быть кратным 8. В (S2+) указывается счетчик (C0...C199). Число n определяет количество включаемых и отключаемых операндов цели (D+) и соответственно число операндов (S1+), в которых записаны значения включения и отключения.

Так как n = 4, то для процессов включения и отключения имеются меркеры M0...M3.

Четыре включающих значения записываются в регистры данных D300, D302, D304 и D306. Четыре отключающих значения записываются в регистры данных D301, D303, D305 и D307.

Для значения включения применяются операнды с четными адресами. Значения отключения записываются в операнды с нечетными адресами. Значения включения/отключения записываются в регистры данных D300...D307 с помощью MOV-инструкции.

ЗНАЧЕНИЕ ВКЛЮЧЕНИЯ	ЗНАЧЕНИЕ ОТКЛЮЧЕНИЯ	ВЫДАЧА
D300 = 40	D301 = 140	M0
D302 = 100	D303 = 200	M1
D304 = 160	D305 = 60	M2
D306 = 240	D307 = 280	M3

Табл. 6-26.
Значения включения/отключения

Если X0 включен, маркеры M0...M3 переключаются в соответствии со следующей иллюстрацией. Если X0 выключен, маркеры не переключаются.

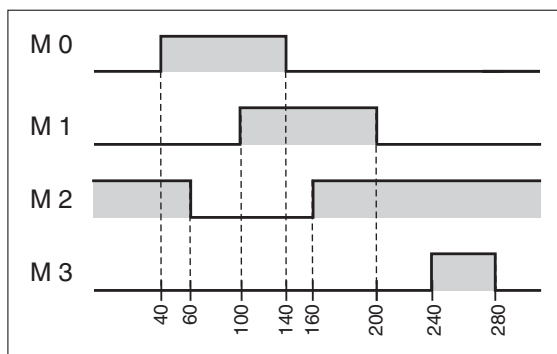


Рис. 6-109.
Временная характеристика включения/отключения меркеров

C000202C



6.8.4 ИНКРЕМЕНТАЛЬНЫЙ МНОГОУСТАВОЧНЫЙ СЧЕТЧИК (INCD)

					INCD		FNC 63			
					Инкрементальный многоуставочный счетчик					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●	●
Операнды	S1+	S2+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS ^① , T, C ^② , D	C ^②	X, Y, M, S	K, H n ≤ 64			16бита	32бита	INCD	9
						●				

- ① X, Y, M, S должны быть кратными 8.
 ② Недопустимы высокоскоростные счетчики

ФУНКЦИЯ

Включение битов в зависимости от состояния счетчика

ОПИСАНИЕ

- n битовые операнды, определяемые в (D+), включаются в зависимости от счетчиков (S2+) и ((S2+)+1).
- В (S1+) предварительно задается требуемое значение для точки включения битовых операндов (D+).
- Счетчик (S2+) должен программироваться с ПК; требуемое (задаваемое) значение должно быть больше, чем самое большое значение включения в (S1+).
- Счетчик (S2+) считает процессы отключения на (S2+).
- Благодаря отключению условий включения инструкции отключаются счетчики (S2+) и ((S2+)+1), а также n битовые операнды (D+).
- После включения n битовых операндов, ((S1+)+1) отключается и включается M8029.

УКАЗАНИЕ

INCD-инструкция может использоваться в программе только один раз.

Если в (S1+) указываются битовые операнды, адрес операнда должен быть кратен 16 (16, 32, 64 и т. п.) или равен 0.

ПРИМЕР ▾

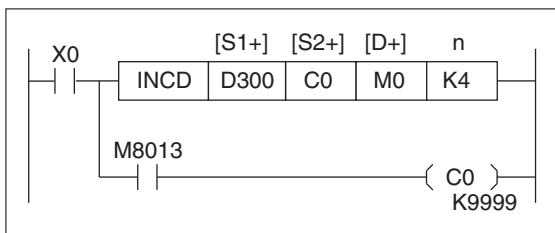


Рис. 6-110.
Пример программирования INCD-инструкции

C000165C

Регистр	Уставка/ число C0	Число C1
D300	20	0
D301	30	1
D302	10	2
D303	40	3

Табл. 6-27.
Пример вводимых значений

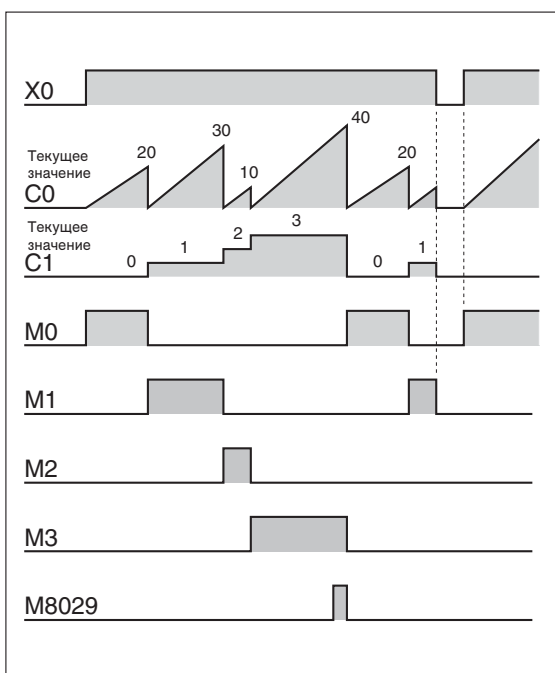


Рис. 6-111.
Временная характеристика соответствует рассмотренному выше примеру включения и отключения меркеров

C000164C

Счетчик C0 отключается автоматически, если будут достигнуты значения, занесенные в D300 ... D303.

Счетчик C1 подсчитывает количество процессов отключения на C0.

Меркеры M0...M3 включают соответствующие значения счетчика C1.

Флаг M8029 включается, если выполняется последний, т.е. n-ый процесс сета. В конце действий этот процесс повторяется.

Счетчики C0 и C1 выключаются, если выключается X0; меркеры M0...M3 также выключаются. Если X0 снова включается, процесс начинается вновь.

△

6.8.5 ОБУЧАЮЩИЙСЯ ТАЙМЕР (TTMR)

			TTMR		FNC 64			
			Обучающийся таймер					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
						●	●	●
Операнды	D	n	Имп. инструкция (P)	Обработка		Шаги программы		
	D	K, H n = 0 - 2		16бита	32бита	TTMR		5
				●				

ФУНКЦИЯ

Предварительное задание настраиваемого значения таймера выполняется воздействием на кнопку

ОПИСАНИЕ

- Измеряется продолжительность включения инструкции (в секундах) (время нажима кнопки), умножается и записывается в регистр данных (D+)+1).
- Число n определяет множитель времени.

$$\begin{aligned}
 n = 0 \quad R \quad D+ &= [(D+) + 1] \times 1 \\
 n = 1 \quad R \quad D+ &= [(D+) + 1] \times 10 \\
 n = 2 \quad R \quad D+ &= [(D+) + 1] \times 100
 \end{aligned}$$

УКАЗАНИЕ

- | С помощью TTMR-инструкции создаются 2 регистра данных (D+) и ((D+)+1).
- | (D+) содержит умноженное время воздействия на кнопку (s). ((D+)+1) содержит время воздействия на кнопку (s).
- | Если условие для выполнения команды более не выполняется, содержимое ((D+)+1) стирается. После этого содержимое (D+) более не изменяется.

ПРИМЕР ▽

Пример формирования времен

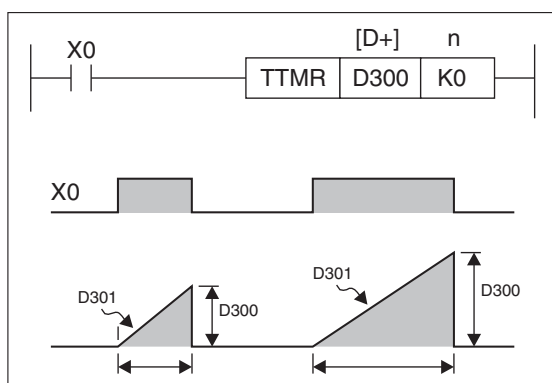


Рис. 6-112.
Пример программирования TTMR-инструкции

C000167C

При включении X0 измеряется продолжительность воздействия.
 D300: Время воздействия в секундах, умноженное на 1
 D301: Время воздействия в секундах



6.8.6 СПЕЦИАЛЬНЫЙ ТАЙМЕР (STMR)

				STMR		FNC 65			
				Специальный таймер					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	T0 - T199	Y, M, S	K, H n=1 - 32767		16бита	32бита	STMR		7
					●				

ФУНКЦИЯ

Генерация функции специального таймера

ОПИСАНИЕ

- С помощью STMR-инструкции могут управляться функции специального таймера
 - замедление отключения
 - Однократный импульс (выводится после отключения условия входа)
 - датчик мигания

- Таймер в (S+) управляет битовым операндом (D+) следующим образом: (D+): задержка выключения
 ((D+)+1): однократный импульс
 ((D+)+2)/((D+)+3): такт мигания

Начиная с (D+), занимают четыре следующих друг за другом бита

УКАЗАНИЕ

| Таймер, используемый в этой инструкции, не может применяться еще раз в другом месте.

ПРИМЕР ▾

Применение STMR-инструкции

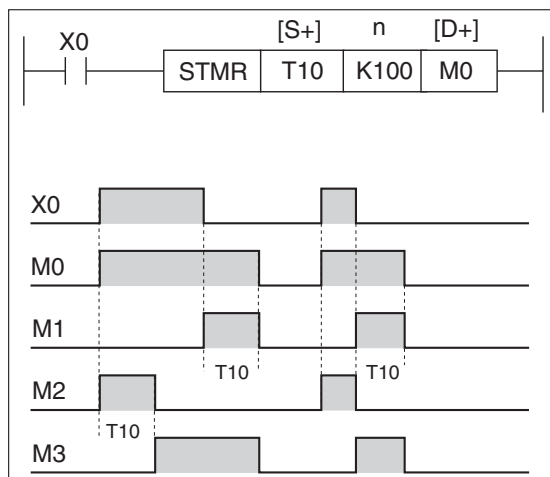


Рис. 6-113.

Пример программы для генерации задержки выключения (M0) и импульса (M1)

C000161C



ПРИМЕР ▾

Пример применения команды STMR (2)

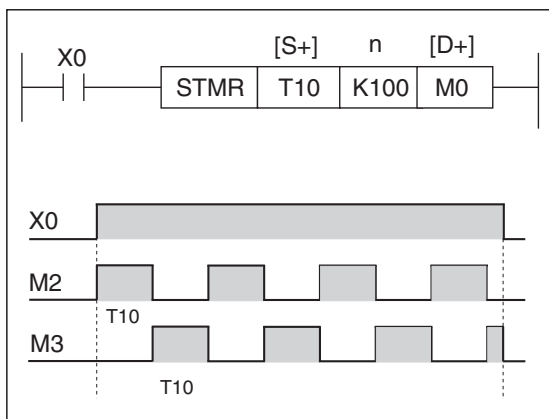


Рис. 6-114:

Если входное условие непрерывно выполняется, M2 и M3 можно применять в качестве такта мигания.

6.8.7

ТРИГГЕРНАЯ ФУНКЦИЯ (FLIP-FLOP) (ALT)

		ALT		FNC 66			
		Триггерная функция					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	Y, M, S	●	16бита	32бита	ALT/ALTP	3	

ФУНКЦИЯ

Реализация триггерной функции

ОПИСАНИЕ

- С помощью команды ALT состояние битового операнда попеременно изменяется.
- ALT-инструкция активизируется по управлению сигналом "1" и записывается в операнд, указанный в D+.
- следующем выполнении команды ALT операнд сбрасывается.

УКАЗАНИЕ

Инструкция выполняется в каждом цикле программы. Этого можно избежать последовательным включением функции импульса (PLS-инструкцией) или параметром "P".

ПРИМЕР ▾

Применение ALT-инструкции. Инвертирование статуса операнда

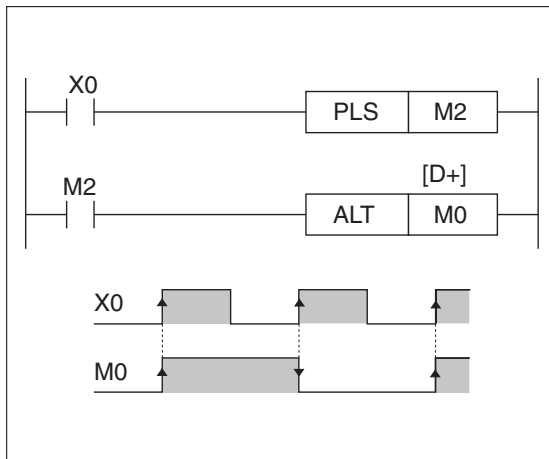


Рис. 6-114.

Пример программирования ALT-инструкции (инвертирование статуса операнда)

C000108C

Состояние маркера M0 инвертируется при каждом включении входа X0.



ПРИМЕР ▾

Применение ALT-инструкции. Инвертирование статуса операнда

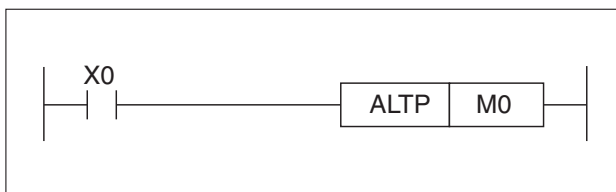


Рис. 6-115.

В этом примере программы для управления маркером M0 применяется команда ALTP.

Функция идентична, приведенной на рисунке 6-114.



ПРИМЕР ▾

Применение ALT-инструкции. Старт-стоп-функция

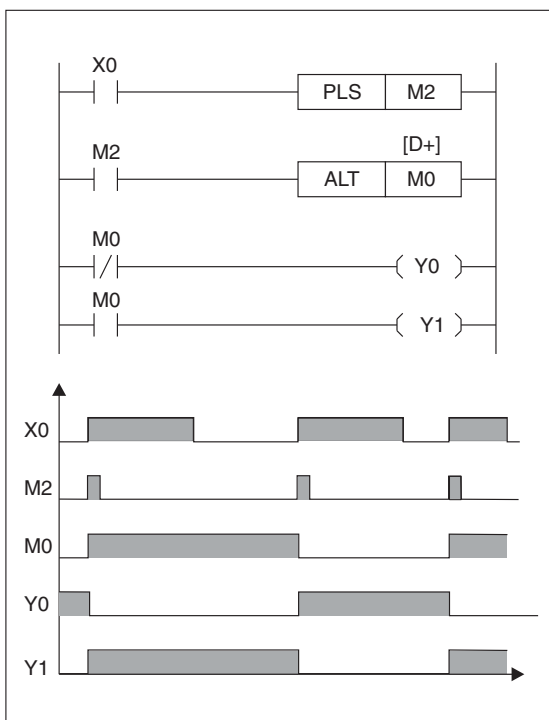


Рис. 6-116.

Пример программирования ALT-инструкции (старт-стоп- функция)

C000114C

Старт-выход Y1 активизируется нажатием кнопки X0. Стоп-выход Y0 активизируется повторным нажатием кнопки X0.

6.8.8 ФУНКЦИЯ НАКЛОННЫХ ХАРАКТЕРИСТИК (RAMP)

		RAMP		FNC 67					
		Рамповая функция							
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
			●	●	●	●	●		
Операнды	S1+, S2+, D+	n	Имп. инструкция (P)	Обработка		Шаги программы			
	D	K, H n=1 ... +32 767		16бита	32бита	RAMP		9	
				●					

ФУНКЦИЯ

Зависимое от времени изменение значения данных, начиная со стартового значения и кончая значением цели. (RAMP функция - это функция наклонного подъема управляющего параметра при разгоне или наклонного опускания при останове).

ОПИСАНИЕ

- Регистр данных, указанный в (S1+), определяет исходное значение.
- Регистр данных, указанный в (S2+), определяет целевое значение.
- Регистр данных, указанный в (D+), запоминает текущее значение функции. Как стартовое значение в (D+) запоминается исходное значение. Рамповое значение функции изменяется n раз до достижения целевого значения.

(S1+): Исходное значение

(S2+): Целевое значение

(D+): Текущее значение рамповой функции

n: Количество циклов операций

- Необходимое время выполнения инструкции T составляет: $T = (n \times \text{время цикла программы})$
- Если в (D+) достигнуто целевое значение, то включается специальный меркер (флаг) M8029. Целевое значение в дальнейшем сохраняется в (D+).
- Количество циклов операций после обработки инструкции запоминается в регистре данных, следующим за (D+).

УКАЗАНИЕ

Время исполнения RAMP-инструкции зависит от времени цикла программы. Поэтому ПК должно работать с постоянным временем цикла программы, чтобы гарантировать определенное соотношение функции.

В центральных процессорах FX2N и FX3U с помощью специального маркера M8026 можно влиять на характеристику функции RAMP. Если M8026 не установлен, происходит постоянное повторение линейно нарастающей функции. Это означает, что если фактическое значение D равно значению в S2, линейно нарастающая функция автоматически сбрасывается и снова запускается. Если специальный маркер M8026 установлен, линейно нарастающая функция удерживается. Это значит, что как только фактическое значение D соответствует значению S2, линейно нарастающая функция сохраняет свое мгновенное состояние. В этом случае M8029 остается установленным до тех пор, пока линейно нарастающая функция активна. Значение в D не сбрасывается до тех пор, пока команда не будет инициализирована повторно.

При FX1S- и FX1N-CPU нельзя влиять на процесс RAMP-функции. Здесь RAMP-функция ведет себя так, как будто включен M8026, т.е. RAMP-функция сохраняет мгновенное состояние.

Если RAMP-функция прерывается перед окончанием, позиция до временной точки прерывания сохраняется столько времени, пока снова не появится сигнал запуска. Если снова появится сигнал рампы, регистры D и D1 снова отключатся и начнется новый цикл.

ПРИМЕР ▾

Применение RAMP-инструкции

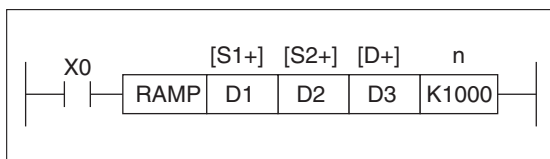
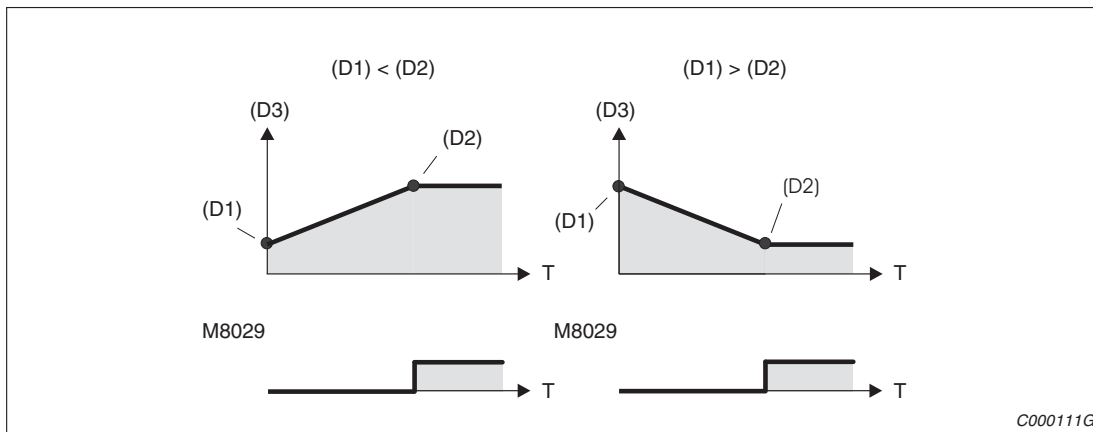


Рис. 6-117.
Пример программирования RAMP-инструкции

C000099C



C000111G

Рис. 6-118. Временные процессы этой схемы

- После включения X0 D3 (D+) вначале определяет исходное значение. Это значение, последовательно возрастая (1000 раз - "n"), изменяется до тех пор, пока в не будет достигнуто целевое значение, записанное в D2 (S2+).
- Время T, необходимое для этого процесса: $T = (n \times \text{время цикла программы})$.
- Количество циклов операций n записано в D4 [(D+)+1].
- Если после определения времени цикла программы (оно несколько больше, чем текущее время цикла программы) в регистре данных D8039 активизируется меркер M8039, то ПК работает с постоянным временем цикла программы.
Если, например, определенное в D8039 значение составляет 20 мс, то в регистре данных D3 потребуются изменения от исходного значения до целевого значения $T = 1000 \times 20 = 20 \text{ с}$.
- Если X0 выключено, прерывается выполнение рамповой функции. Если затем X0 снова включится, выполнение рамповой функции начнется вновь с исходного состояния.
- Если выполнение рамповой функции окончено, включается специальный меркер (флаг) 8029, и D3 (D+) принимает из D1 (S1+) исходное значение.
- Убедитесь, что D4 пустой, если ПК после останова снова включен в режим "Работа" и X0 еще включен.

△

УКАЗАНИЕ

Рамповая функция может применяться также со специальными меркерами M8193 и M8194, чтобы стимулировать функции SER- и RS-инструкций (FNC61 и FNC80). Это в первую очередь необходимо для старых программаторов.

6.8.9 ПОЗИЦИОНИРОВАНИЕ ПОВОРОТНОГО СТОЛА (ROTC)

				ROTC		FNC 68			
				Позиционирование поворотного стола					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S+	m1 / m2	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	D ①	K, H ②	Y, M, S ③	●	16бита	32бита	ROTC		9
					●				

① 3 следующих друг за другом списка данных ((S+1) ≤ m1)

② m1 = от 2 до 32 767; m2 = от 0 до 32 767 (m1 ≥ m2)

③ 8 следующих друг за другом битов

ФУНКЦИЯ

Управление поворотным столом

ОПИСАНИЕ

- Позиции поворотного стола контролируются 2-х фазным датчиком.
- Все позиции цели на столе указываются относительно нулевой позиции.
- Позиция цели всегда определяется по кратчайшему пути.

УКАЗАНИЕ

Эта команда переключает или опрашивает следующие операнды:

- ((D+)+0) сигнал фазы А счетчика
- ((D+)+1) сигнал фазы В счетчика
- ((D+)+2) распознавание нулевой точки
- ((D+)+3) вывод: быстрое вращение стола вперед
- ((D+)+4) вывод: ползучее движение вперед
- ((D+)+5) выход „Стоп“
- ((D+)+6) вывод: ползучее движение назад
- ((D+)+7) вывод: быстрое вращение стола назад
- m1 количество счетных импульсов на каждый оборот стола
- m2 количество счетных импульсов для пути, проходимого на ползучей скорости
- ((S+)+0) фактическое положение (возможно только считывание)
- ((S+)+1) целевое положение
- ((S+)+2) распознавание перемещаемой детали

Если во время выполнения команды ROTC распознана нулевая точка (((D+)+2), содержимые операндов с ((S+)+0) по ((S+)+2) стираются.

ПРИМЕР ▾

Прием (фиксация) импульсов счета

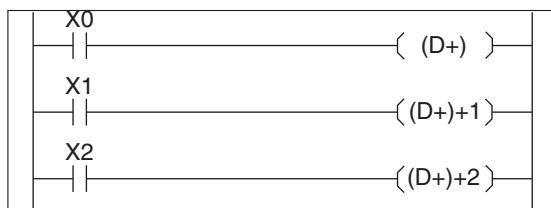


Рис. 6-119.

Пример программирования для определения сигналов

C000238C

Так как энкодер и выключатель нулевой точки контролируются через входы, эти входы необходимо записывать в операнды (D+).



УКАЗАНИЕ

- | Параметры ((S+)+1) и ((S+)+2) должны записываться перед включением.
- | Перед первым вводом в эксплуатацию стол должен быть установлен в исходную позицию.
- | Инструкция может быть применена только один раз в программе.
- | Так как движения поворота стола фиксируются по нормальным входам, то нельзя выдавать больше, чем примерно 25 импульсов в секунду.

ПРИМЕР ▾

Поворотному столу для одного поворота необходимо 500 импульсов. Стол имеет десять позиций; поэтому между двумя позициями выдается 50 импульсов. Нулевая позиция воспринимается как команда “Нулевая позиция”.

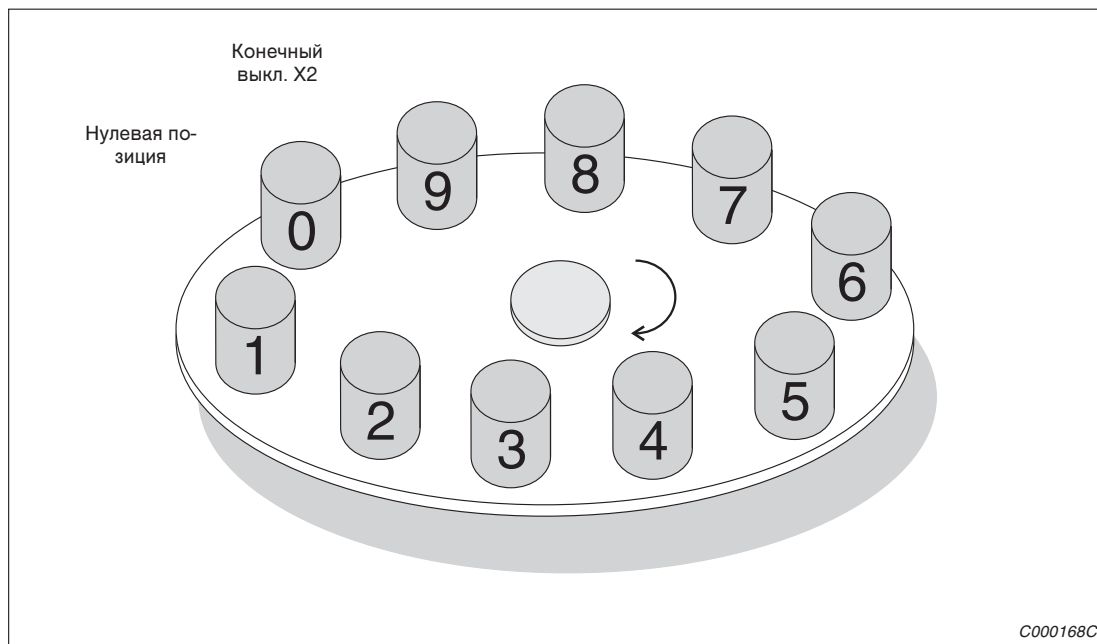


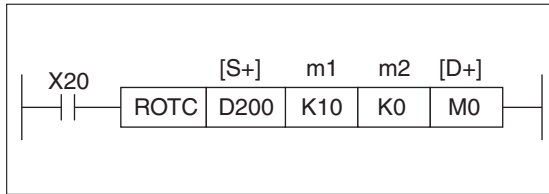
Рис. 6-120. Пример позиционирования поворотного стола

Если деталь, находящаяся на 7-ой позиции стола, должна переместиться на 3-ю позицию, ROTC-инструкция должна выглядеть следующим образом:

- (S+) = любой регистр данных, например, D200
- ((S+)+1) = 3 x 50 = 150: удаление позиции в импульсах от позиции 0
Значение должно быть записано в D201 перед выполнением инструкции.
- ((S+)+2) = 7 x 50 = 350: удаление позиции в импульсах от позиции 0
Значение должно быть записано в D202 перед выполнением инструкции.
- m1 = 500

Если необходимо перемещаться на двух скоростях, то путь, который нужно проходить назад на медленной скорости, указывается импульсами в (m2).

- m2 = 0 (нет медленной скорости)
- (D+) = любые биты M, Y, S, которые управляют двигателем стола

ПРИМЕР ▾**Рис. 6-121.**

Пример программирования
ROTC-инструкции

C000240C

Если включен X20, то стол вращается по часовой стрелке на 4 позиции.

Двигатель управляется меркером M3. Если позиция цели достигнута, то включается меркер M5.

Меркеры M0, M1 и M2 контролируют поворот стола. △

6.8.10 ИНСТРУКЦИЯ СОРТИРОВКИ (SORT)

					SORT		FNC 69			
					Инструкция сортировки					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●	●	●
Операнды	S+	n1 / n2	D+	m	Имп. инструкция (P)	Обработка		Шаги программы		
	D ①	K, H ②	D	K, H, D		16бита	32бита	SORT		11
						●				

① (N1) и (n2) следующие друг за другом списки данных

② n1 = от 1 до 32; n2 = от 1 до 6

ФУНКЦИЯ

Сортировка матрицы по значениям колонок (столбцов)

ОПИСАНИЕ

При вызове команды SORT внутренняя матрица данных (таблица), заданная начальным регистром данных (S+) и размером (n1 строк и n2 столбцов), сортируется по значениям столбца „m“ и заново записывается начиная с регистра данных (D+).

УКАЗАНИЕ

| SORT-инструкция может применяться в программе только один раз.

| При выполнении SORT-инструкции каждая запись в соответствии с данными в выбранном поле сортировки m сортируется в возрастающей последовательности.

| (S+) и (D+) могут указываться в том же регистре данных, так как накопленные значения не изменяются.

| Если области данных, в которых сохраняются матрицы, при отличающихся регистрах данных (S+) и (D+) перекрещиваются, то это может привести к потере данных.

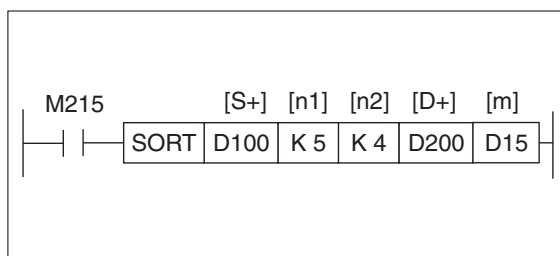
| После исполнения SORT-инструкции с помощью M8029 включается опознание. Процесс сортировки заканчивается только тогда, когда в n1 будет получено заданное число.

| Во время процесса сортировки данные в сортируемых матрицах не могут изменяться, так как в этом случае могут быть записаны ошибочные данные.

| С помощью команды SORT данные можно сортировать только в возрастающей последовательности. С помощью команды SORT2 данные в таблице можно дополнительно сортировать и в убывающей последовательности (раздел 6.8.10).

ПРИМЕР ▾

Применение SORT-инструкции на матрице

**Рис. 6-122.**

Пример программирования SORT-инструкции

C000327C

Матрицы данных могут быть представлены в следующей форме:

Строка №.	1	2	3	4
Столбец №.	ID- размер	Высота	Вес	Возраст
1	D100	D105	D110	D115
	1	150	45	20
2	D101	D106	D111	D116
	2	180	50	40
3	D102	D107	D112	D117
	3	160	70	30
4	D103	D108	D113	D118
	4	100	20	8
5	D104	D109	D114	D119
	5	150	50	45

Табл. 6-28.

Неотсортированная матрица

Сортировка по столбцу (m) = K2:

Строка №.	1	2	3	4
Столбец №.	ID- размер	Высота	Вес	Возраст
1	D200	D205	D210	D215
	4	100	20	8
2	D201	D206	D211	D216
	1	150	45	20
3	D202	D207	D212	D217
	5	150	50	45
4	D203	D208	D213	D218
	3	160	70	30
5	D204	D209	D214	D219
	2	180	50	40

Табл. 6-29.

Отсортированная матрица по примененной SORT-инструкции

7 СПЕЦИАЛЬНЫЕ КОМАНДЫ

7.1 ОБЩИЕ УКАЗАНИЯ

В этом разделе описаны специальные команды контроллеров семейства FX для особых применений. С помощью этих команд можно реализовать функции ввода и вывода данных, коммуникаций между модулями и управления специальными модулями.

Подробное пояснение структуры таблиц инструкций содержится в разделе 6.1.1.

7.1.1 ОБЗОР СПЕЦИАЛЬНЫХ КОМАНД

Классификация	Инструкция	FNC	Назначение	Раздел	Серия типов ПК				
					FX1S	FX1N	FX2N	FX2NC	FX3U
Инструкции ввода/вывода	TKY	70	Десятичная клавиатура	7.2.1			●	●	●
	HKY	71	Десятичная клавиатура	7.2.2			●	●	●
	DSW	72	Числовой переключатель	7.2.3	●	●	●	●	●
	SEGD	73	7-ми сегментный индикатор	7.2.4			●	●	●
	SEGL	74	7-ми сегментный индикатор с запоминанием	7.2.5	●	●	●	●	●
	ARWS	75	7-ми сегментный индикатор с доп. кнопками	7.2.6			●	●	●
	ASC	76	ASCII-конвертирование	7.2.7			●	●	●
	PR	77	Выдача данных по выходам	7.2.8			●	●	●
Обмен данными со специальными модулями	FROM	78	Чтение данных из специального модуля	7.3.1		●	●	●	●
	TO	79	Запись данных в специальные модули	7.3.2		●	●	●	●
Инструкции последовательной коммуникации	RS	80	Запись данных в специальный модуль	7.4.1	●	●	●	●	●
	PRUN	81	Перезапись входов или меркеров	7.4.2	●	●	●	●	●
	ASCI	82	Перезапись входов или меркеров	7.4.3	●	●	●	●	●
	HEX	83	Преобразование в шестнадцатеричное значение	7.4.4	●	●	●	●	●
	CCD	84	Проверка на сумму и паритет	7.4.5	●	●	●	●	●
	VRRD	85	Чтение заданного значения из FX1N-8AV-BD и FX2N-8AV-BD	7.4.6	●	●	●	●	
	VRSC	86	Чтение положения переключателя FX1N-8AV-BD и FX2N-8AV-BD	7.4.7	●	●	●	●	
	RS2	87	Последовательная передача данных	7.4.8					●
Команда регулирования	PID	88	Программирование контуров регулирования	7.5	●	●	●	●	●
Содержимого индексных фиксации/восстановление	ZPUSH	102	Фиксация содержимого индексных регистров	7.5.1					●
	ZPOP	103	Восстановление содержимого индексных регистров	7.5.2					●

Табл. 7-1: Обзор команд для обмена данными с периферийными модулями связи регулирования и работы с индексными регистрами

Классификация	Инструкция	FNC	Назначение	Раздел	Серия типов ПК				
					FX1S	FX1N	FX2N	FX2NC	FX3U
Инструкции чисел с плавающей запятой	DECMP	110	Сравнение чисел с плавающей запятой	7.7.1			●	●	●
	DEZCP	111	Сравнение чисел с плавающей запятой в области	7.7.2			●	●	●
	DEMOV	112	Перенос чисел с плавающей запятой	7.7.3					●
	DESTR	116	Преобразование числа с плавающей запятой в строковую величину	7.7.4					●
	DEVAL	117	Преобразование строковой величины в число с плавающей запятой	7.7.5					●
	DEBCD	118	Преобразование формата с плавающей запятой в научный формат чисел	7.7.6			●	●	●
	DEBIN	119	Преобразование научного формата чисел в формат с плавающей запятой	7.7.7			●	●	●
	DEADD	120	Сложение чисел с плавающей запятой	7.7.8			●	●	●
	DESUB	121	Вычитание чисел с плавающей запятой	7.7.9			●	●	●
	DEMUL	122	Умножение чисел с плавающей запятой	7.7.10			●	●	●
	DEDIV	123	Деление чисел с плавающей запятой	7.7.11			●	●	●
	DEXP	124	Представление числа с плавающей запятой в виде показателя степени по основанию e	7.7.12					●
	DLOGE	125	Вычисление натурального логарифма	7.7.13					●
	DLOG10	126	Вычисление десятичного логарифма	7.7.14					●
	DESQR	127	Квадратный корень из числа с плавающей запятой	7.7.15			●	●	●
	DENEG	128	Изменение арифметического знака числа с плавающей запятой	7.7.16					●
	INT	129	Преобразование формата с плавающей запятой в десятичный формат	7.7.17			●	●	●
	DSIN	130	Расчет синуса числа с плавающей запятой	7.7.18			●	●	●
	DCOS	131	Расчет косинуса числа с плавающей запятой	7.7.19			●	●	●
	DTAN	132	Расчет тангенса числа с плавающей запятой	7.7.20			●	●	●
DASIN	133	Вычисление арксинуса	7.7.21					●	
DACOS	134	Вычисление арккосинуса	7.7.22					●	
DATAN	135	Вычисление арктангенса	7.7.23					●	
DRAD	136	Пересчет из градусов в радианы	7.7.24					●	
DDEG	137	Пересчет из радиан в градусы	7.7.25					●	
Инструкции обработки данных	WSUM	140	Суммирование содержимого словных операндов	7.8.1					●
	WTOB	141	Разделение данных в словных операндах на байты	7.8.2					●
	BTOW	142	Образование словного операнда из отдельных байтов	7.8.3					●
	UNI	143	Объединение групп по 4 бита в словный операнд	7.8.4					●
	DIS	144	Разделение словного операнда на группы по 4 бита	7.8.5					●
	SWAP	147	Обмен старшим и младшим байтами	7.8.6			●	●	●
	SORT2	149	Сортировка данных в таблице	7.8.7					●
Позиционирование с помощью сервоусилителя	DSZR	150	Движение в базовую точку (с использованием выключателя приближения)	7.9.6					●
	DVIT	151	Позиционирование с использованием прерывания	7.9.7					●
	TBL	152	Позиционирование по таблице данных	7.9.8					●
	ABS	155	Чтение абсолютного действительного значения позиции	7.9.9	●	●			●
	ZRN	156	Перемещение в исходную точку	7.9.10	●	●			●
	PLSV	157	Выдача импульсов с переменной частотой	7.9.11	●	●			●
	DRVI	158	Позиционирование по приращиваемому значению	7.9.12	●	●			●
	DRVA	159	Позиционирование по абсолютному значению	7.9.13	●	●			●

Табл. 7-2: Обзор специальных команд для обработки чисел с плавающей запятой и дат, а также для позиционирования

Классификация	Инструкция	FNC	Назначение	Раздел	Серия типов ПК				
					FX1S	FX1N	FX2N	FX2NC	FX3U
Инструкции реального времени	TCMP	160	Сравнение данных часов	7.10.1	●	●	●	●	●
	TZCP	161	Сравнение данных часов в области	7.10.2	●	●	●	●	●
	TADD	162	Сложение данных часов	7.10.3	●	●	●	●	●
	TSUB	163	Вычитание данных часов	7.10.4	●	●	●	●	●
	HTOS	164	Преобразование времени в формате "часы, минуты, секунды" в секунды	7.10.5					●
	STON	165	Преобразование времени в секундах в формат "часы, минуты, секунды"	7.10.6					●
	TRD	166	Чтение данных часов	7.10.7	●	●	●	●	●
	TWR	167	Запись данных часов	7.10.8	●	●	●	●	●
	HOUR	169	Счетчик рабочего времени	7.10.9	●	●			●
Инструкции кода Грея	GRY	170	Преобразование целых чисел в коде Грея	7.11.1			●	●	●
	GBIN	171	Преобразование кода Грея в целое число	7.11.2			●	●	●
Обмен данными с FXON_3A	RD3A	176	Чтение аналогового значения из FXON-3A	7.12.1		●			●
	WR3A	177	Передача данных к FXON-3A	7.12.2		●			●
Функция внешнего ПЗУ	EXTR	180	Выполнение команды из внешнего ПЗУ	7.13.1			●	●	
Разные команды	COMRD	182	Считывание комментария к операнду	7.14.1					●
	RND	184	Тактовый генератор с регулируемой скважностью импульсов	7.14.2					●
	DUTY	186	Выдача импульса определенной длины	7.14.3					●
	CRC	188	Проверка данных (проверка CRC)	7.14.4					●
	DHCMOV	189	Передача фактического значения высокоскоростного счетчика	7.14.5					●
Команды для данных, хранящихся в следующих друг за другом операндах (блоках данных)	BK+	192	Сложение данных двух блоков	7.15.1					●
	BK-	193	Определение разности данных двух блоков	7.15.2					●
	BKCMP=	194	Сравнение на "равно"	7.15.3					●
	BKCMP>	195	Сравнение на "больше"						●
	BKCMP<	196	Сравнение на "меньше"						●
	BKCMP<>	197	Сравнение на "неравно"						●
	BKCMP<=	198	Сравнение на "меньше или равно"						●
BKMPA>=	199	Сравнение на "больше или равно"					●		
Команды для обработки строковых величин	STR	200	Преобразование двоичных данных в строковые	7.16.1					●
	VAL	201	Преобразование строковых данных в двоичные	7.16.2					●
	\$+	202	Объединение строковых величин	7.16.3					●
	LEN	203	Определение длины строковой величины	7.16.4					●
	RIGHT	204	Фрагмент строковой величины справа	7.16.5					●
	LEFT	205	Фрагмент строковой величины слева	7.16.6					●
	MIDR	206	Копирование знаков из строковой величины	7.16.7					●
	MIDW	207	Замена цепочки символов	7.16.7					●
	INSTR	208	Поиск цепочки символов	7.16.8					●
	\$MOV	209	Передача цепочки символов	7.16.10					●
Команды для обработки перечней данных	FDEL	210	Стирание данных из перечня данных	7.17.1					●
	FINS	211	Ввод данных в перечень данных	7.17.2					●
	POP	212	Считывание данных, введенных в перечень последними	7.17.3					●
	SFR	213	Перемещение 16-битного слова данных вправо	7.17.4					●
	SFL	214	Перемещение 16-битного слова данных влево	7.17.5					●

Табл. 7-3: Обзор специальных команд

Классификация	Инструкция	FNC	Назначение	Раздел	Серия типов ПК				
					FX1S	FX1N	FX2N	FX2NC	FX3U
ЗАГРУЗКА-логическая связь сравнений	LD=	224	Инструкция сравнения, равно	7.18.1	●	●	●	●	●
	LD>	225	Инструкция сравнения, больше						
	LD<	226	Инструкция сравнения, меньше						
	LD<>	228	Инструкция сравнения, не равно						
	LD≤	229	Инструкция сравнения, меньше равно						
	LD≥	230	Инструкция сравнения, больше равно						
Логическое "И"-логическая связь сравнений	AND=	232	Инструкция сравнения логического "И", равно	7.18.2	●	●	●	●	●
	AND>	233	Инструкция сравнения логического "И", больше						
	AND<	234	Инструкция сравнения логического "И", меньше						
	AND<>	236	Инструкция сравнения логического "И", не равно						
	AND≤	237	Инструкция сравнения логического "И", меньше равно						
	AND≥	238	Инструкция сравнения логического "И", больше равно						
Логическое "ИЛИ"-логическая связь сравнений	OR=	240	Инструкция сравнения логического "ИЛИ", равно	7.18.3	●	●	●	●	●
	OR>	241	Инструкция сравнения логического "ИЛИ", больше						
	OR<	242	Инструкция сравнения логического "ИЛИ", меньше						
	OR<>	244	Инструкция сравнения логического "ИЛИ", не равно						
	OR≤	245	Инструкция сравнения логического "ИЛИ", меньше равно						
	OR≥	246	Инструкция сравнения логического "ИЛИ", больше равно						
Команды контроля данных	LIMIT	256	Ограничение диапазона вывода значений	7.19.1					●
	BAND	257	Установка смещения входа	7.19.2					●
	ZONE	258	Установка смещения выхода	7.19.3					●
	SCL	259	Масштабирование значений	7.19.4					●
	DABIN	260	Преобразование числа в кодировке ASCII в двоичное значение	7.19.5					●
	BINDA	261	Преобразование двоичного значения в код ASCII	7.19.6					●
	SCL2	269	Масштабирование значений (таблица значений имеет иную структуру, чем у команды SCL)	7.19.7					●
Команды для коммуникации с преобразователями частоты	IVCK	270	Проверка состояния преобразователя частоты	7.20.1					●
	IVDR	271	Управление преобразователем частоты	7.20.2					●
	IVRD	272	Считывание параметров преобразователя частоты	7.20.3					●
	IVWR	273	Запись параметров в преобразователь частоты	7.20.4					●
	IVBWR	274	Поблочная запись параметров в преобразователь частоты	7.20.5					●
Обмен данными со специальными модулями	RBFM	278	Считывание из буферной памяти специальных модулей	7.21.1					●
	WBFM	279	Запись в буферную память специальных модулей	7.21.2					●
Команда для высокоскоростного счетчика	DHSCT	280	Сравнение фактического значения высокоскоростного счетчика с данными в перечнях данных	7.22					●
Команды для расширенных регистров файлов	LOADR	290	Считывание данных из расширенных регистров файлов	7.23.1					●
	SAVER	291	Запись данных в расширенные регистры файлов	7.23.2					●
	INITR	292	Инициализация расширенных регистров и расширенных регистров файлов	7.23.3					●
	LOGR	293	Сохранение значений операндов в расширенных регистрах или расширенных регистрах файлов	7.23.4					●
	RWER	294	Передача данных из расширенных регистров в расширенные регистры файлов	7.23.5					●
	INITER	295	Инициализация расширенных регистров файлов	7.23.6					●

Табл. 7-4: Обзор специальных команд

7.2 ИНСТРУКЦИИ ВВОДА/ВЫВОДА

Команды с FNC 70 по FNC 77 служат для обмена данными с внешними приборами, подключенными ко входам или выходам контроллера. С помощью команд FROM и TO можно вводить данные в буферную память специальных модулей и считывать из их буферной памяти.

ОБЗОР ИНСТРУКЦИЙ FNC 70...79

Символ	FNC	Назначение	Разд.
TKY	70	Десятичная клавиатура	7.2.1
HKY	71	Шестнадцатеричная клавиатура	7.2.2
DSW	72	Числовой переключатель	7.2.3
SEGD	73	7-ми сегментный индикатор	7.2.4
SEGL	74	7-ми сегментный индикатор с запоминанием	7.2.5
ARWS	75	7-ми сегментный индикатор с доп. кнопками	7.2.6
ASC	76	ASCII-конвертирование	7.2.7
PR	77	Выдача данных по выходам	7.2.8

Табл. 7-5: Обзор команд ввода-вывода

7.2.1 ДЕСЯТИЧНАЯ КЛАВИАТУРА (ТКУ)

				TKY		FNC 70			
				Десятичная клавиатура					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
					●	●			
Операнды	S1+	D1+	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	X, Y, M, S ①	KnY, KnM, KnS, T, C, D, V, Z	Y, M, S ②				TKY		
					16бита	32бита	DTKY	7 13	

- ① 10 следующих друг за другом бит
- ② 11 следующих друг за другом бит

ФУНКЦИЯ

Чтение десятичной клавиатуры по входам

ОПИСАНИЕ

- Клавиатура с 10 клавишами читается в ПК по битам (S+)...((S+)+9)
- Указанные значения один за другим заносятся в слово данных (D1+). При 16-ти битовом операнде могут записываться 4 места (max.999.9) и при 32-х битовом операнде 8 мест (макс. 99.999.999).
- Если указываются больше возможных 4 или 8 мест, записываются только последние 4 или 8 мест.
- Биты (D2+)...((D2+)+10) отображают состояния клавиш.

УАЗАНИЕ

TKY-инструкция может применяться в программе только один раз.

Если TKY-инструкция больше не активна, то биты (D2+) стираются. Содержание (D1+) сохраняется.

ПРИМЕР ▾

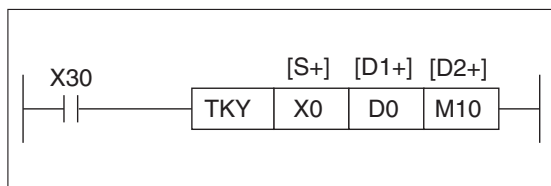


Рис. 7-1:
Программирование TKY-инструкции

C000242C

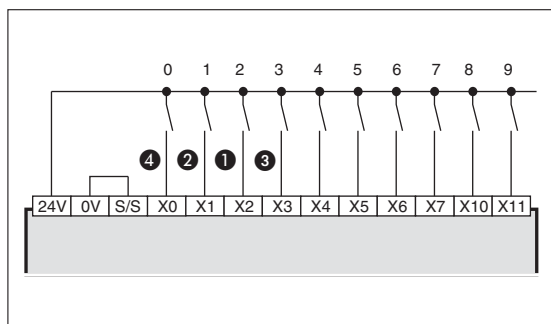


Рис. 7-2:
Распределение клавиш

C000241C

В примере цифровые клавиши от 0 до 9 распределены на входы X. В (S+) указан стартовый адрес X0.

ПРИМЕР ▾

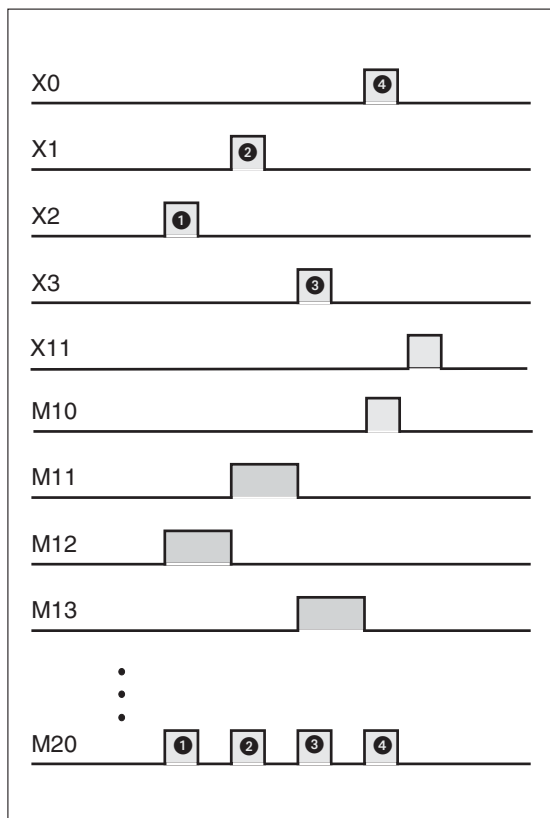


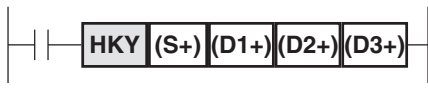
Рис. 7-3:

Пример программирования включения и отключения входов и меркеров

C000243C

Если клавиши X0...X3 задействуются в последовательности ①...④, то в регистре данных D0 будет записано значение 2 130. Если затем задействовать клавишу X11, то два из регистров выключаться и новое содержание D0 будет 1 309. △

7.2.2 ШЕСТНАДЦАТИРИЧНАЯ КЛАВИАТУРА (НКУ)

					НКУ		FNC 71			
					Шестнадцатеричная клавиатура					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●	●	●
Операнды	S+	D1+	D2+	D3+	Имп. инструкция(P)	Обработка		Шаги программы		
	X	Y	T, C, D, V, Z	Y, M, S [Ⓞ]		16бита	32бита	НКУ	9	
						●	●	DNKY	17	

① 8 следующих друг за другом бит

ФУНКЦИЯ

Считывание шестнадцатеричной клавиатуры (цифры 0...9 и буквы A...F) через входы

ОПИСАНИЕ

- Клавиши читаются в мультиплексным методом (методом умножения). Соответственно задействовано 4 входа и выхода.
- Клавиши от 0 до 9 воспринимаются как число и заносятся в (D2+).
- Клавиши от A до F включают биты от (D3+) до ((D3+)+5).
- (D3+) принадлежит первым 8 меркерам для запоминания функций воздействия на клавиши и контрольных сигналов. Клавиши от A до F включают меркеры от (D3+) до ((D3+)+5). Меркер ((D3+)+6) заносится при воздействии на одну из клавиш от A до F, а меркер ((D3+)+6) заносится при воздействии на одну из клавиш от 0 до 9. После каждой регистрации воздействия на клавишу включается меркер M8029.
- Число, указанное посредством клавиш от 0 до 9, записывается в (D2+). Максимум может быть задано 4 места (макс. 9.999).
Если выполняется 32-х битный операнд, может быть задано 8 мест (макс.99.999.999).
- Если оказано воздействие больше чем на одну клавишу, то обрабатывается первая нажатая клавиша.
- Если указано больше 4 или 8 мест, то учитываются только последние указанные 4 или 8 мест.

УАЗАНИЕ

| НКУ-инструкция может применяться в программе только один раз.

| При применении НКУ-инструкции ПК должно работать с постоянным временем цикла.

ПРИМЕР ▾

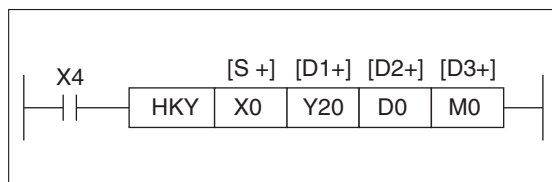


Рис. 7-4:
Пример программирования НКУ-инструкции

C000244C

ПРИМЕР ▾

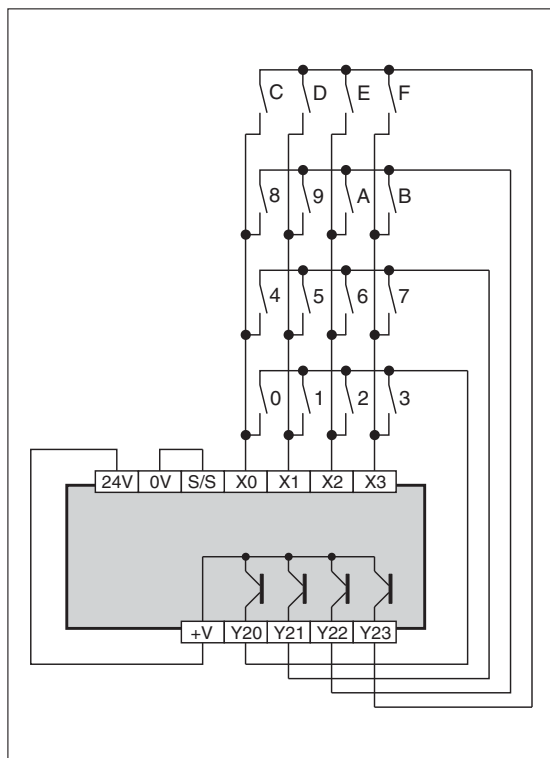


Рис. 7-5:

Пример программирования включения и отключения входов и меркеров

C000245C

ФУНКЦИОНАЛЬНЫЕ КЛАВИШИ

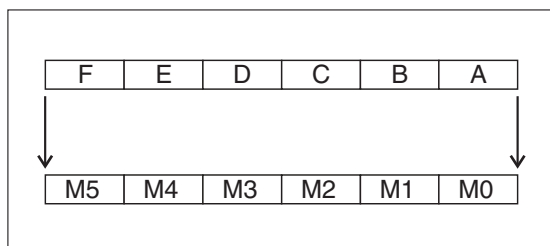


Рис. 7-6:

Пример распределения меркеров

C000246C

Если задействована клавиша A, включается меркер M0. M0 остается включенным до тех пор, пока не будут задействованы другие клавиши. Если в конце будет задействована клавиша D, то M0 отключится, а M3 включится.

Если задействуются две или больше клавиш, то принимается во внимание первая нажатая клавиша.

ВЫХОДЫ

До тех пор, пока удерживаются нажатыми клавиши A...F, M6 остается активным.

До тех пор, пока удерживаются нажатыми клавиши 1...9, M7 остается активным.

Если X4 отключается, данные в D0 не изменяются. Меркеры от M0 до M7 выключаются.

Восприятие воздействия на клавиши требует 8 циклов.

НАКОПИТЕЛЬ ДАННЫХ

Указанные значения записываются в регистр данных D0 четырехзначными.

△

ПРИНЦИП ДЕЙСТВИЯ СО СПЕЦИАЛЬНЫМ МЕРКЕРОМ M8167

Чтение шестнадцатеричной клавиатуры по входам

ОПИСАНИЕ

- Включение специального меркера M8167 действует так, что интерпретируется ввод по 16-ти клавишам (1–9, A–F), как шестнадцатеричный формат.
- Клавиши считываются по мультиплексному методу. Соответственно задействуются 4 входа и 4 выхода, причем (S+) определяет первый вход, а (D1+) первый выход.
- Клавиши воспринимаются как число и заносятся в (D2+).
- Посредством клавиш 0...9 и A...F указанное число записывается в (D2+). Максимум может указываться 4 места (макс. FFFF_н).
Если выполняется 32-х битный операнд, может быть задано 8 мест (макс. FFFFFFFF_н)
- Если оказано воздействие больше чем на одну клавишу, то обрабатывается первой нажатая клавиша.
- Если указано больше 4 или 8 мест, то учитываются только последние указанные 4 или 8 мест.

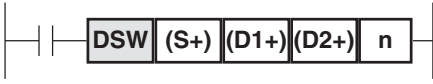
УКАЗАНИЕ

| НКУ-инструкция может применяться в программе только один раз.

| При применении НКУ-инструкции ПК должно работать с постоянным временем цикла.

Если время цикла очень мало, нужно работать с временным прерыванием.

7.2.3 ЦИФРОВОЙ ПЕРЕКЛЮЧАТЕЛЬ (DSW)

					DSW		FNC 72			
					Цифровой переключатель					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●	●	●	●
Операнды	S+	D1+	D2+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	X ^①	Y ^②	T, C, D, V, Z	Y, M, S ^③		16бита	32бита	DSW	9	
						●				

- ① 8 следующих друг за другом бит
 ② 4 следующих друг за другом бит
 ③ n = 1 или 2

ФУНКЦИЯ

Чтение BCD-переключателя по методу мультиплекса (умножения)

ОПИСАНИЕ

- В ПК могут считываться один или два (n) четырехзначные BCD-переключатели.
- DSW-инструкция контролирует 4 выхода и 4 входа. Если считываются два четырехзначных BCD-переключателя, необходимо 8 входов.
- (S+) определяет первый из четырех следующих один за другим входов.
- (D1+) определяет первый из четырех следующих один за другим выходов.
- (D2+) определяет словный операнд, который содержит считанное значение.

УКАЗАНИЕ

В контроллере серии FX2N эту команду разрешается вставлять в программу только два раза. Для контроллеров серий FX1S, FX1N и FX3U ограничений по количеству используемых команд DSW не имеется.

Для корректного выполнения DSW-инструкции ПК должно применяться с транзисторными выходами.

ПРИМЕР ▾

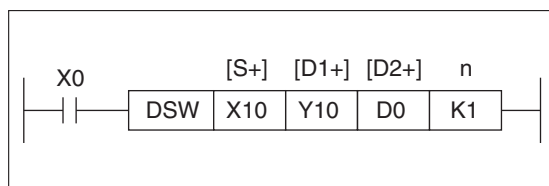


Рис. 7-7:
Пример программирования DSW-инструкции

C000247C

ПРИМЕР ▾

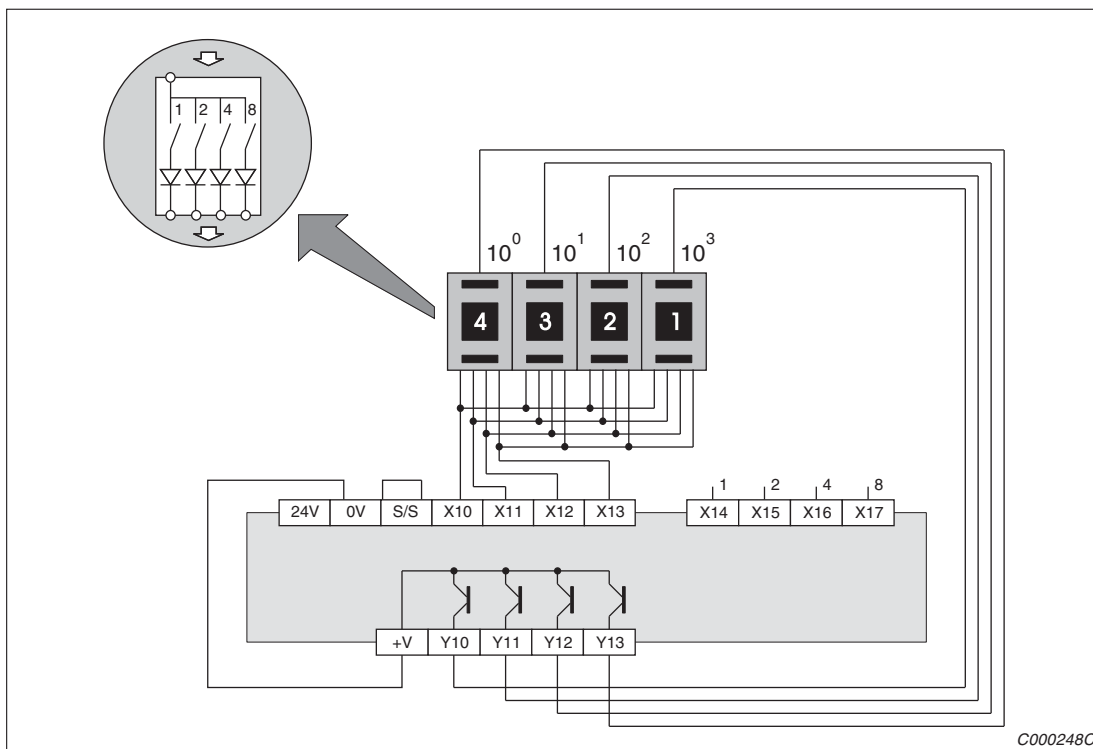


Рис. 7-8: Схема присоединения входов/выходов

ЧТЕНИЕ ПЕРВОГО ЧЕТЫРЕХЗНАЧНОГО БЛОКА

Настройка четырехместного цифрового переключателя (BCD), который связан с входами X10...X13, считывается одно место (секция) за другим по выходам Y10...Y13 и запоминается в двоичном коде в регистре данных D0.

В этом случае произведена настройка по n равному 1.

ЧТЕНИЕ ВТОРОГО ЧЕТЫРЕХЗНАЧНОГО БЛОКА

Настройка переключателя (BCD), который связан с входами X14...X17, считывается одно место (секция) за другим по выходам Y10...Y13 и запоминается в двоичном коде в регистре данных D0.

В этом случае произведена настройка по n равному 2.

Если включен X0, то выходы Y10...Y13 один за другим обрабатывают состояния соответствующих входов X.

Если процесс работы завершен, включается меркер M8029.

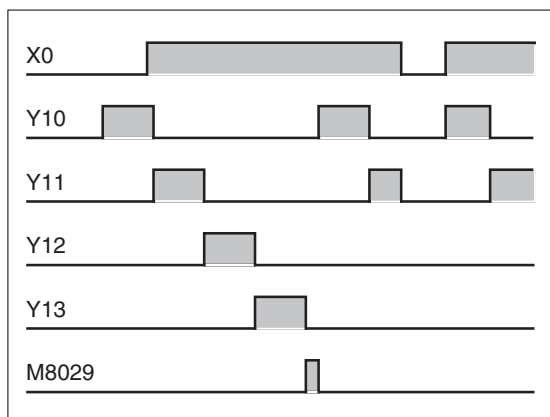


Рис. 7-9: Временная диаграмма включения выходов

C000249C

7.2.4 7-МИ СЕГМЕНТНЫЙ ЦИФРОВОЙ ИНДИКАТОР (SEGD)

		SEGD		FNC 73			
		7-ми сегментный цифровой индикатор					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16бита	32бита	SEGD/ SEGDP	5

ФУНКЦИЯ

Выдача одноместного шестнадцатеричного числа на 7-ми сегментный цифровой индикатор

ОПИСАНИЕ

- Шестнадцатеричное число в (S+) автоматически преобразовывается в формат, необходимый для 7-ми сегментного цифрового индикатора, и заносится в (D+).
- Биты от B0 до B6 (D+) соответствуют сегментам 7-ми сегментного цифрового индикатора:

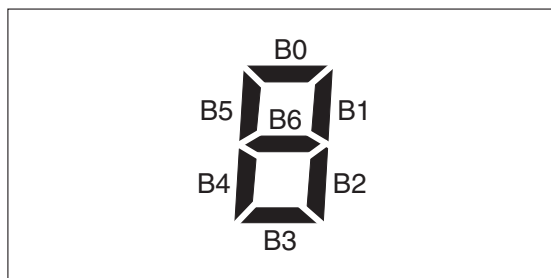


Рис. 7-10:
7-ми сегментный цифровой индикатор

C000251C

УКАЗАНИЕ

Биты с b15 по b8 в регистре (D+) не изменяются.

ПРИМЕР ▾

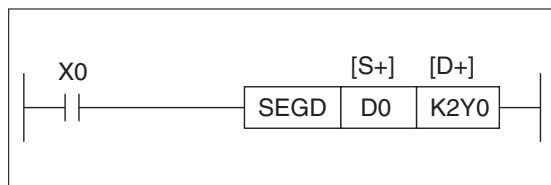


Рис. 7-11:
Пример программирования SEGDP-инструкции

C000250C

ВЫХОД	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
СЕГМЕНТ	B0	B1	B2	B3	B4	B5	B6	B7

△

7.2.5 7-МИ СЕГМЕНТНЫЙ ЦИФРОВОЙ ИНДИКАТОР С ЗАПОМИНАНИЕМ (SEGL)

			SEGL		FNC 74			
			7-ми сегментный цифровой индикатор с запоминанием					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●	
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y ①	K, H		16бита	32бита	SEGL	7

① n = 0...3: 8 друг за другом следующих выходов
 n = 4...7: 12 друг за другом следующих выходов

ФУНКЦИЯ

Управление одним или двумя четырехзначными 7-сегментными индикаторами с сохраняющейся индикацией

ОПИСАНИЕ

- С помощью этой инструкции можно управлять до двух четырехзначными 7-ми сегментными индикаторами. Управление выполняется по мультиплексному методу. Для работы записываются 4 тактовые выходы и для каждого четырехзначного индикатора еще по 4 выходных данных.
- Содержащееся в (S+) числовое значение (макс. 9999) преобразуется в двоично-десятичный код и выводится через выходы с (D+) по ((D+)+3). Если требуется управлять двумя четырехзначными индикаторами, то данные для второго индикатора выводятся через выходы с ((D+)+10) по ((D+)+13). Если используются менее 4 разрядов, не используемые для индикации операнды не могут использоваться для других целей.
- BCD-код, находящийся соответственно на выходах данных, автоматически распределяется по тактовым выходам ((D+)+4)...((D+)+7) согласно местам индикатора.
- Настройка (n) зависит от четырех факторов:
 - а) выходной логики выходов ПК (+/- включения)
 - б) логики на проводе данных 7-ми сегментного индикатора
 - в) логики тактовых входов 7-ми сегментного индикатора
 - г) количества примененных 7-ми сегментных индикаторов

ПОЛОЖИТЕЛЬНАЯ ЛОГИКА (Source-выход)

При Source-выходе (положительном) выход является старшим (HIGH), если внутренняя логика равна 1.

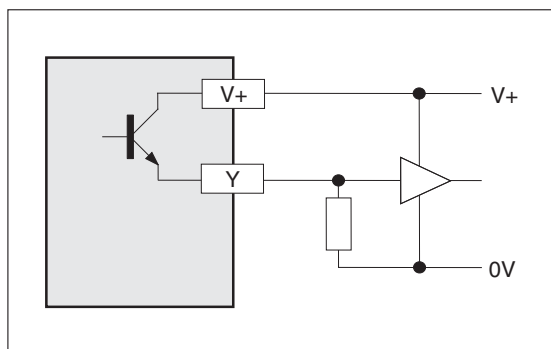


Рис. 7-12: Положительная логика

C000254C

Логический тактовый сигнал: данные запоминаются, если тактовый сигнал высокий (HIGH).
 Логический сигнал данных: активный провод данных является высоким

ОТРИЦАТЕЛЬНАЯ ЛОГИКА (Sink-выход)

При Sink-выходах (HIGH). (отрицательных) выход является низкий (LOW), если внутренняя логика равна 1.

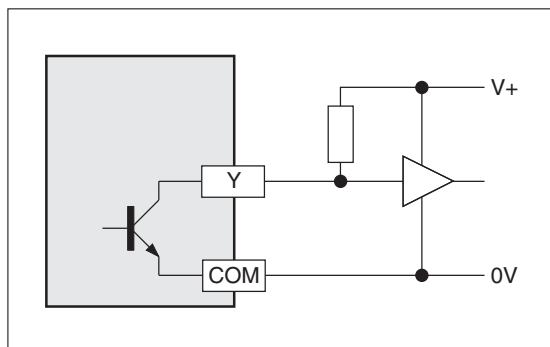


Рис. 7-13:
Отрицательная логика

C000255C

Логический тактовый сигнал: данные запоминаются, если тактовый сигнал младший (LOW).
Логический сигнал данных: активный провод данных является младшим (LOW).

ПК-ЛОГИКА	ТАКТОВЫЙ СИГНАЛ	ПРОВОД ДАННЫХ	n	
			1 ИНДИКАТОР	2 ИНДИКАТОРА
Positiv (+)	Positiv (HIGH)	Positiv (HIGH)	0	4
Negativ (-)	Negativ (LOW)	Negativ (LOW)		
Positiv (+)	Positiv (HIGH)	Negativ (LOW)	1	5
Negativ (-)	Negativ (LOW)	Positiv (HIGH)		
Negativ (-)	Positiv (HIGH)	Negativ (LOW)	2	6
Positiv (+)	Negativ (LOW)	Positiv (HIGH)		
Negativ (-)	Positiv (HIGH)	Positiv (HIGH)	3	7
Positiv (+)	Negativ (LOW)	Negativ (LOW)		

Табл. 7-6: 7-ми сегментная логика индикатора

УКАЗАНИЕ

Для корректного выполнения инструкции ПК должно применяться с транзисторными выходами.

Может применяться только 7-ми сегментный индикатор с сохранением данных.

В контроллерах серий FX2N и FX2NC эту команду разрешается использовать в программе только два раза. Для контроллеров серий FX1S, FX1N и FX3U ограничений по количеству используемых команд SEGL не имеется.

Команда SEGL выполняется синхронно со временем цикла программы. Это время должно быть больше 10 мс. При более коротких значениях времени цикла используйте контроллер в режиме постоянного времени цикла длительностью не меньше 10 мс.

Для обновления 4-разрядной индикации требуется 12-кратное время цикла программы.

ПРИМЕР ▾

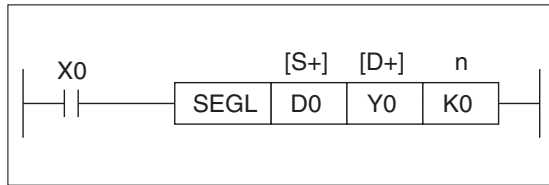
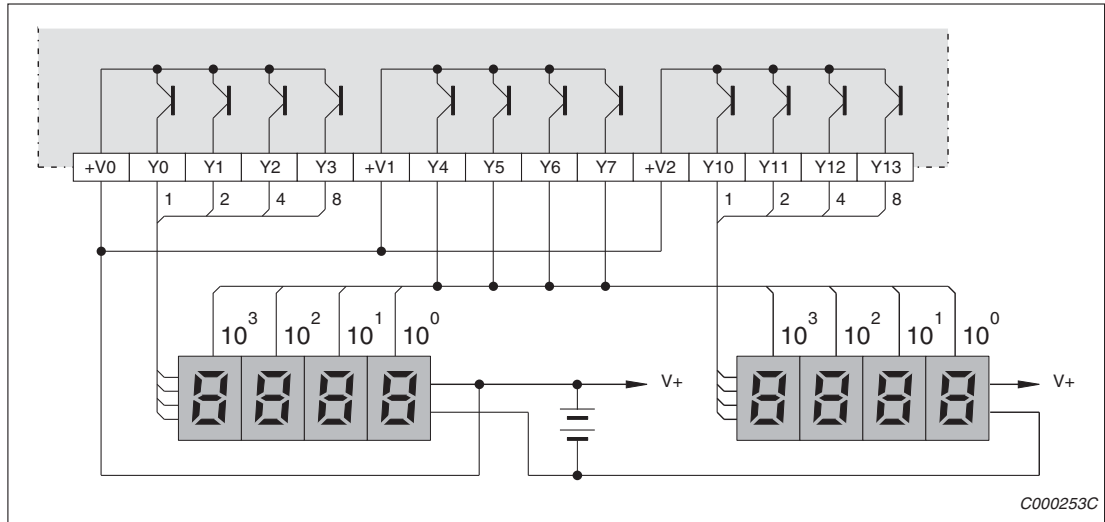


Рис. 7-14:
Пример программирования SEGL-инструкции

C000252C



C000253C

Рис. 7-15: Схема подключения выходов



7.2.6 7-МИ СЕГМЕНТНЫЙ ИНДИКАТОР С ДОПОЛНИТЕЛЬНЫМИ КЛАВИШАМИ (ARWS)

					ARWS		FNC 75			
					7-ми сегментный индикатор с дополнительными клавишами					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●	●	●
Операнды	S+	D1+	D2+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	X, Y, M, S ①	T, D, V, Z	Y ②	K, H n= 0 ... 3			16бита	32бита	ARWS	9
						●				

① 4 следующих друг за другом операнда

② 8 следующих друг за другом операнда

ФУНКЦИЯ

Выбор и изменение места четырехместного BCD-числа на 7-ми сегментном индикаторе

ОПИСАНИЕ

- Опрашиваются четыре клавиши от (S+) до ((S+)+3):
 (S+) = увеличение выбранного места
 ((S+)+1) = уменьшение выбранного места
 ((S+)+2) = курсор вправо
 ((S+)+3) = курсор влево
- Данные, заносимые в (D1+), отображаются на четырехместном 7-ми сегментном индикаторе и изменяются с помощью клавиш (S+)
- Данные, заносимые в (D1+), являются двоичными данными.
- С помощью (D2+) и (n) определяются выходы и вид схемы подключения, к которой подключается 7-ми сегментный индикатор (см. SEGL-инструкцию).

УКАЗАНИЕ

Для корректного выполнения инструкции ПК должно применяться с транзисторными выходами.

ARWS-инструкция может использоваться в программе только один раз.

ПРИМЕР ▾

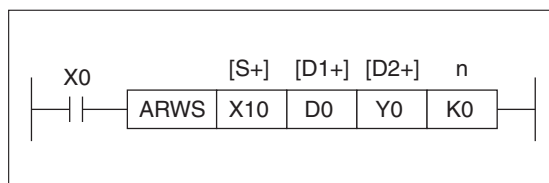


Рис. 7-16:

Пример программирования ARWS-инструкции

C000256C

ПРИМЕР ▾

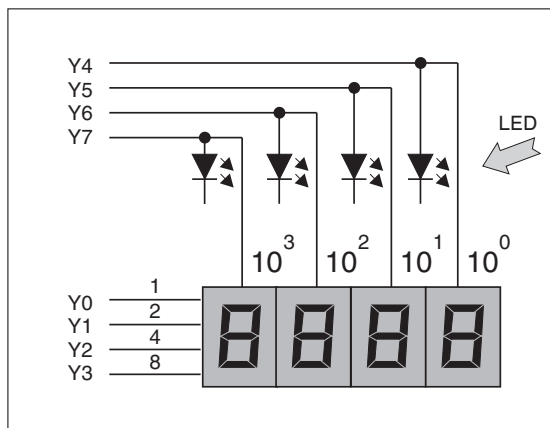


Рис. 7-17:

Пример присоединения 7-ми сегментного индикатора

C000257C

В 16-ти битовом регистре данных D0 записывается четыре BCD-числа. Каждое BCD-число имеет четыре бита. Максимум в D0 можно записать число 9 999.

Клавишами или входами X10...X13 могут изменяться позиции и числовые значения индикаторов (см.рисунок 7-18).

X11: счет вверх выбранных мест 0-1-2-3

X10: счет вниз выбранных мест 0-9-8-7

X13: сдвиг влево

X12: сдвиг вправо

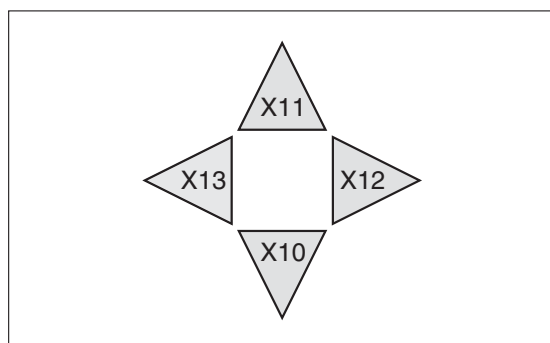


Рис. 7-18:

Пример выходов X10...X13

C000258C

По входам X12 и X13 определяются изменяемые позиции.

Если X0 включено, позиция 10^3 рассматривается как начальная позиция.

Каждый нажим на X12 и X13 действует так, что изменяются позиции индикаторов в предварительно задаваемой последовательности:

Воздействие на X12 (сдвиг вправо):

$$10^3 - 10^2 - 10^1 - 10^0 - 10^3$$

Воздействие на X13 (сдвиг влево):

$$10^3 - 10^2 - 10^1 - 10^0 - 10^3$$

Позиция, определенная по X12 и X13, может отображаться (Y4...Y7) дополнительной лампой LED в проводе строб-сигнала.

ПРИМЕР ▾

По клавишам или входам X10...X13 изменяются числовые значения на установленных позициях отображения.

С помощью X10 и X11 определяется последовательность ввода данных.

Данные в регистре данных D0 изменяются в следующей последовательности:

X11: счет вверх: 0 - 1 - 2-...8 - 9 - 0 - 1

X10: счет вниз: 0 - 9 - 8 - 7-...1 - 0 - 9

Текущее включенное значение отображается на 7-ми сегментном индикаторе.

С помощью ARWS-инструкции в регистр данных D0 может быть записано желаемое значение и одновременно отображено на 7-ми сегментном индикаторе.



7.2.7 ASCII-КОНВЕРТИРОВАНИЕ (ASC)

		ASC		FNC 76			
		ASCII-конвертирование					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●	●	
Operanden	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	ASCII-Daten z.B. 0-9; A-Z; a-z	T, C, D ①		16бита	32бита	ASC	7
				●			

① 4 следующих друг за другом выхода

ФУНКЦИЯ

Конвертирование (преобразование) буквенно-цифровых данных в ASCII-данные

ОПИСАНИЕ

- Буквенно-цифровые данные, указанные в (S+), конвертируются в ASCII-знаки и сохраняются в (D+).
- Могут задаваться максимум 8 буквенно-цифровых данных.

ПРИМЕР ▾

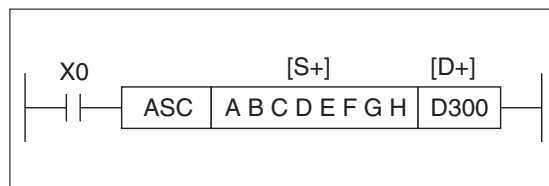


Рис. 7-19:
Пример программирования
ASC-инструкции

C000259C

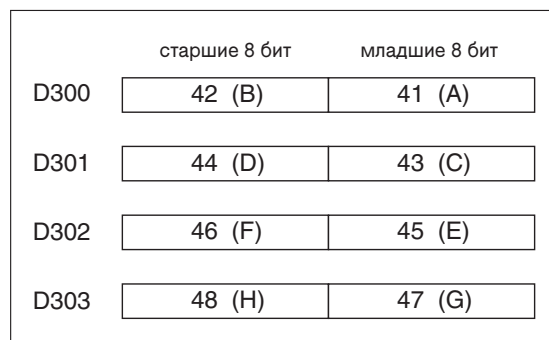


Рис. 7-20:
Запись и сохранение данных от A до H

C000260C

биты с 3 по 0		биты с 6 по 4							
		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	'	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	}
E	1110	SO	RS	.	>	N	x	n	~
F	1111	SI	VS	/	?	O	-	o	DEL

Табл. 7-7: Кодировка ASCII

Функция при установленном специальном маркере M8161


Для сохранения одного знака в кодировке ASCII достаточно 8 битов. Если специальный маркер M8161 не установлен, в одном словном операнде (D+) сохраняются два знака ASCII. Если специальный маркер M8161 установлен, в каждом словном операнде (D+) знак ASCII сохраняется в младших 8 битах. Биты с 8 по 15 заполняются значениями "00".

УКАЗАНИЕ

Если маркер M8161 равен "0", то количество необходимых для (D+) операндов равно половине количества преобразуемых знаков ASCII. При установленном маркере M8161 для (D+) нужно столько операндов, сколько имеется знаков в строковой величине (S+).

Специальный маркер M8161 влияет также на поведение команд RS, ASCII, HEX, CCD и CRC. Если одна из этих команд применяется в той же программе, что и команда ASC, то перед выполнением команды ASC следует установить маркер M8161, а сразу после выполнения команды ASC снова сбросить этот маркер.

7.2.8 ВЫДАЧА ДАННЫХ ПО ВЫХОДАМ (PR)

		PR		FNC 77			
		Выдача данных по выходам					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	S+	D+	Имп. инструкция(P)		Обработка		Шаги программы
	T, C, D	Y ①			16бита	32бита	PR
					●		

① 10 следующих друг за другом выходов

ФУНКЦИЯ

Выдача ASCII-знаков по выходам

ОПИСАНИЕ

- Вывод 8 знаков ASCII с (S+) по ((S+)+3) или 16 знаков ASCII с (S+) по ((S+)+7) через выходы (D+)
- Выходы (D+)...((D+)+7) представляют биты b0...b7 сформированные по (S+).
- ((D+)+8) можно использовать в качестве тактового сигнала, ((D+)+9) показывает завершение обработки команды.

УАЗАНИЕ

PR-инструкция может использоваться в программе только два раза.

Для корректного выполнения инструкции ПК должно применяться с транзисторными выходами.

На количество выводимых знаков можно влиять с помощью специального маркера M8027. Если M8027 не установлен ("0"), выводятся 8 знаков (8 байтов). Если M8027 установлен (на "1"), выводятся 16 знаков в 16 байтах. В этом случае во время выполнения не должно быть установлено входное условие команды PR. Команду PR можно запустить с помощью короткого импульса. После вывода 16 знаков на время одного программного цикла устанавливается специальный маркер M8029, чтобы показать конец выполнения. Учитывайте, что M8029 устанавливается только в 16-байтовом режиме (M8027 = "1").

Если специальный маркер M8027 установлен и адреса с (S+) по ((S+)+7) содержат знак "NUL" (00H), этот знак не выводится. Команда PR обрабатывается полностью, а затем устанавливается маркер 8029.

ПРИМЕР ▾

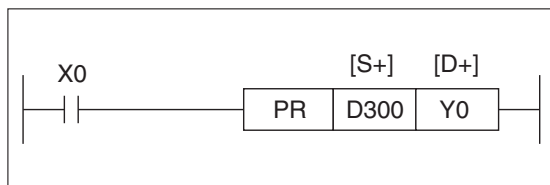


Рис. 7-21:
Пример программирования PR-инструкции

C000261C

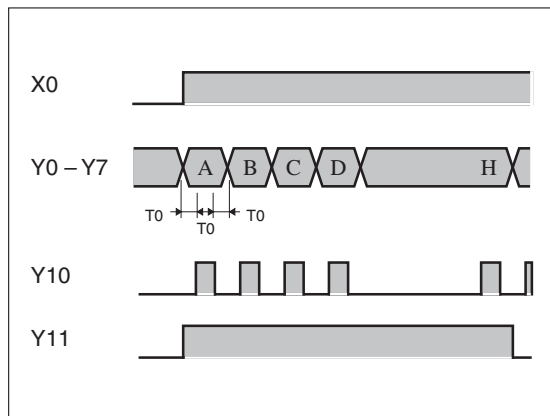


Рис. 7-22:
Диаграмма примера программирования для включения входов/выходов

TO = время цикла (скана)

C000262C

В регистрах данных D300...D303 находятся ASCII-данные примера программирования по рисунку 7.2.7. Согласно инструкции выдаются буквы "А"... "Н".

В качестве выходов используются Y0 (младший бит) и до Y7 (старший бит), а также Y10 (строб-сигнал) и Y11 (флаг выполнения).

ФОРМАТ ВЫВОДА

Если X0 во время работы инструкции выключится, то передача данных прекращается.

Процесс начинается вновь, как только X0 снова включится.



7.3 ОБМЕН ДАННЫМИ СО СПЕЦИАЛЬНЫМИ МОДУЛЯМИ

Команды FNC 78 и FNC 79

Символ	FNC	Значение	Разд.
FROM	78	Считывание данных из специального модуля	7.3.1
TO	79	Запись данных в специальный модуль	7.3.2

Табл. 7-8: Команды для обмена данными со специальными модулями

За исключением серии FX1S, количество цифровых входов и выходов у всех базовых блоков MELSEC семейства FX можно увеличить, подключив расширительные модули. Кроме того, можно еще более увеличить объем выполняемых функций контроллера, установив так называемые специальные модули. Например, специальные модули определяют аналоговые значения (токи, напряжения), регулируют температуру или коммуницируют с внешней аппаратурой.

В то время как для цифровых модулей не нужны особые команды (дополнительные входы и выходы обрабатываются точно так же, как входы и выходы базового блока), для обмена данными между базовым блоком и специальным модулем используются две прикладные команды: FROM и TO.

В специальном модуле предусмотрена область памяти, в которой, например, можно промежуточно хранить (с буферизацией) аналоговые измеренные значения или принятые данные. В связи с ее назначением эта область памяти называется "буферной памятью". Доступ к буферной памяти в специальном модуле может получить и базовый блок, например, чтобы считать измеренные значения или принятые данные. Базовый блок может и записывать данные в буферную память, после чего специальный модуль подвергает эти данные дальнейшей обработке. Например, это могут быть настройки для функционирования специального модуля, передаваемые данные и т. п.

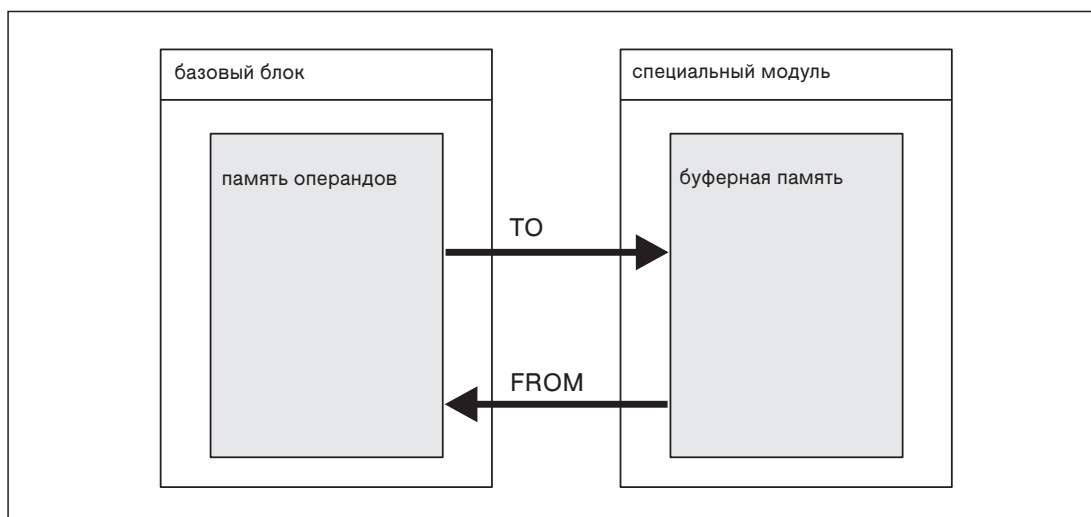


Рис. 7-23: Между базовым блоком программируемого контроллера и специальным модулем возможен обмен данными с помощью команд FROM и TO.

Буферная память может содержать до 32767 отдельных ячеек. Каждый из этих адресов буферной памяти может хранить 16 битов информации. Функции адресов буферной памяти зависят от типа специального модуля и разъяснены в руководствах по отдельным специальным модулям.

Для правильного функционирования команд FROM и TO они нуждаются в определенных данных:

- Из какого специального модуля требуется считать данные или в какой специальный модуль требуется передать данные?

- Каков первый адрес в буферной памяти, из которого требуется считать данные или в который требуется записать данные?
- Из скольких адресов буферной памяти требуется считать данные или в какое количество адресов данные требуется записать?
- Где в базовом блоке следует сохранить данные из буферной памяти или где находятся данные, которые требуется передать в специальный модуль?

Адрес специального модуля

Если имеется несколько специальных модулей, то для того, чтобы данные были переданы в правильный модуль или считаны из правильного модуля, необходимо особое обозначение модулей. Для этого каждый специальный модуль автоматически получает номер из диапазона от 0 до 7. (К одному контроллеру можно подключить максимум 8 специальных модулей.) Номера присваиваются без разрыва, и нумерация начинается с модуля, подключенного к контроллеру первым.

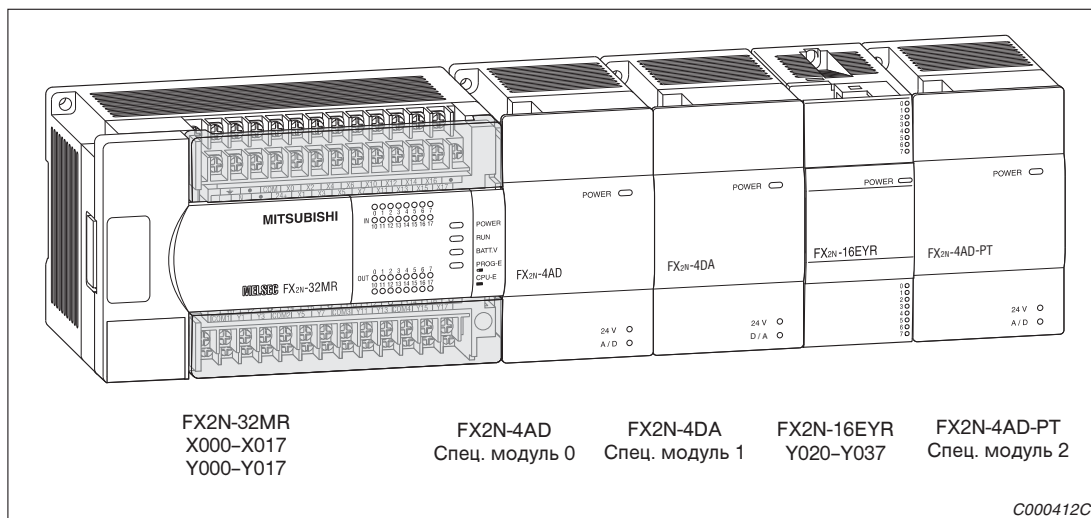


Рис. 7-25: Специальные модули, к которым можно обращаться с помощью команд FROM и TO, расположены справа от базового блока.

Начальный адрес в буферной памяти

Каждый из максимум 32767 адресов буферной памяти может быть обозначен с помощью десятичного числа в диапазоне от 0 до 32766 (FX1N: от 0 до 31). 32-битные данные записываются в буферную память так, чтобы ячейка памяти с более низким адресом содержала младшие 16 битов, а следующий адрес в буферной памяти - старшие 16 битов.

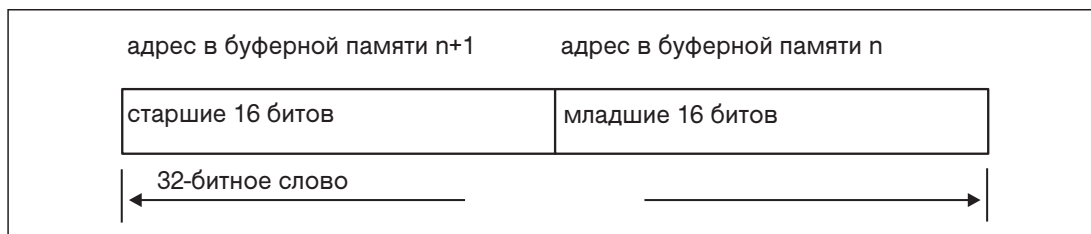


Рис. 7-24: В качестве начального адреса для 32-битных данных необходимо всегда указывать адрес, содержащий младшие 16 битов.

Количество передаваемых данных

Количество данных относится к передаваемым единицам данных. Если команда FROM или TO выполняется в качестве 16-битной команды, указанное количество данных соответствует количеству передаваемых слов. В случае 32-битной команды в виде DFROM или DTO указывается передаваемое количество двойных слов.

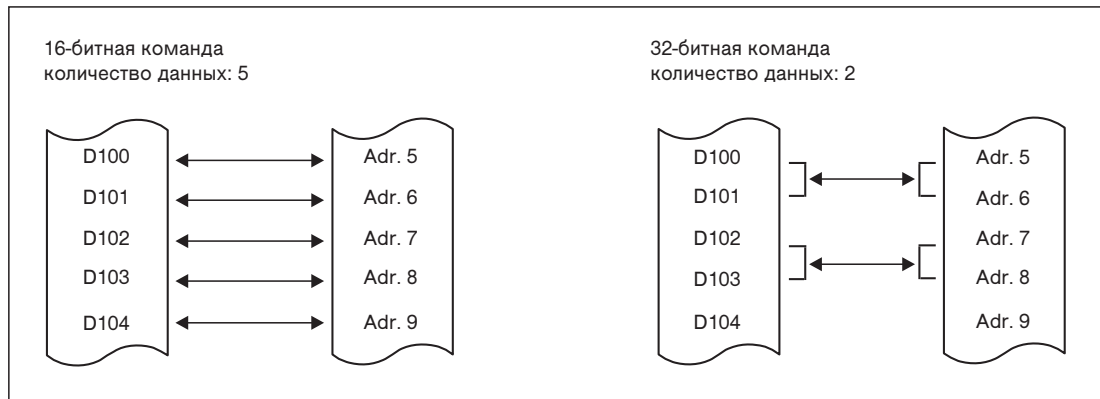


Рис. 7-26: Количество слов данных в случае 16-битной и 32-битной команды

Место назначения и источник в базовом блоке

В большинстве случаев данные считываются из регистров и передаются в специальный модуль, или считываются из его буферной памяти в регистры данных базового блока. Однако в качестве цели или источника передачи данных могут использоваться и выходы, маркеры и фактические значения таймеров и счетчиков.

7.3.1

ЧТЕНИЕ ДАННЫХ ИЗ СПЕЦИАЛЬНОГО МОДУЛЯ (FROM)

		FROM		FNC 78			
		Чтение данных из специального модуля					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●	●	●
Операнды	D+	n1, n2, n3	Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z	K, H	●	16бита	32бита	FROM/FROMP	9
				●	●	DFROM/DFROMP	17

ФУНКЦИЯ

Чтение данных из буферной памяти подключенного специального модуля

ОПИСАНИЕ

- Чтение слов данных n3 из специального модуля с адресом n1.
- Слова данных n3, получаемые по адресу n2 буферной памяти, записываются в (D+)... [(D+) + (n3 -1)].

УКАЗАНИЕ

Допустимые диапазоны: n1 = 0...7
n2 = 0...31, 0...32767 у FX2N/FX2NC/FX3U
n3= 1...32 при 16-битной команде (1...32767 у FX3U)
n3= 1...16 при 32-битной команде (1...16383 у FX3U)

Если с помощью команды FROM передается большой объем данных, может возникнуть ошибка контрольного таймера. В этом случае следует разделить данные на части или изменить настройку контрольного таймера. В серии FX3 можно использовать альтернативную команду RBFM.

В контроллерах FX2N, начиная с версии 3.0, и контроллерах FX3U можно с помощью специального маркера M8028 установить, должна ли передача данных при прерывании приостанавливаться или полностью завершаться:

Прерывание заблокировано (M8028 = 0)	Прерывание разрешено (M8028 = 1)
Переход на подпрограмму ожидает окончания выполнения инструкции FROM	Переход на подпрограмму выполняется немедленно.
При переходе по прерыванию возможна небольшая задержка, не более чем $(800m+200)$ мкр сек, где m=количество 32 битных слов Гарантирует выполнение инструкции FROM в программе прерывания, не оказывая влияние на другие инструкции	Переход по прерыванию происходит немедленно. После возврата из программы прерывания возобновляется работа инструкции FROM. Если в программе прерывания запрограммирована инструкция FROM, то данный режим будет выполняться не корректно. M8028 не должен использоваться если необходима временная синхронизация.

ПРИМЕР ▾

Чтение данных

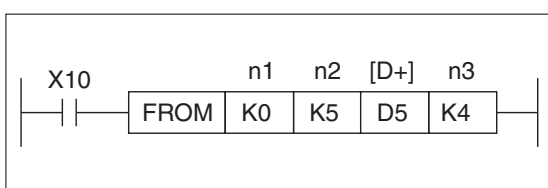


Рис. 7-27:
Пример программирования FROM-инструкции

C000264C

По примеру, показанному на рисунке, содержание адресов буферной памяти #5...#8 специального модуля с позиционного номера 0 передается в регистры D5...D8. Значение адресации представлено по отдельности на следующем рисунке.

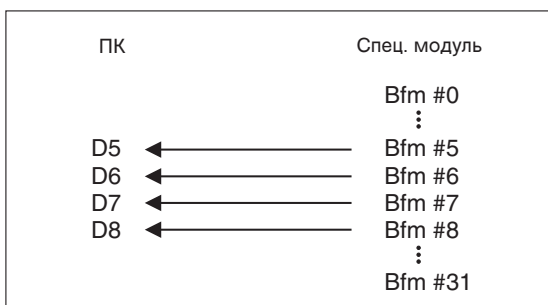


Рис. 7-28:
Адресация при FROM-инструкции

C000265C



УКАЗАНИЕ

Если должно передаваться только содержание адресов буферной памяти, то нужно занести в n3 значение 1.

7.3.2 ЗАПИСЬ ДАННЫХ В СПЕЦИАЛЬНЫЙ МОДУЛЬ (ТО)

		ТО		FNC 79			
		Запись данных в специальный модуль					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●	●	●
Операнды	S+	n1, n2, n3	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z	K, H		16бита	32бита	ТО/ТОР	9
				●	●	ДТО/ДТОР	17

ФУНКЦИЯ

Запись данных из ПК в буферную память подключенного специального модуля

ОПИСАНИЕ

- Запись слов данных n3 из ПК в специальный модуль с адресом n1.
- Слова данных n3, получаемые из буферной памяти, записываются в n2...[(n2) + (n3 - 1)].

УКАЗАНИЕ

Допустимые диапазоны: n1 = 0...7
n2 = 0...31, 0...32767 у FX2N/FX2NC/FX3U
n3= 1...32 при 16-битной команде (1...32767 у FX3U)
n3= 1...16 при 32-битной команде (1...16383 у FX3U)

Если с помощью команды ТО передаются большие объемы данных, может возникнуть ошибка контрольного таймера. В этом случае следует разделить данные на части или изменить настройку контрольного таймера. В серии FX3U можно использовать альтернативную команду WBFM.

В контроллерах FX2N, начиная с версии 3.0, и контроллерах FX3U можно с помощью специального маркера M8028 установить, должна ли передача данных при прерывании приостанавливаться или полностью завершаться:

Прерывание заблокировано (M8028 = 0)	Прерывание разрешено (M8028 = 1)
Переход на подпрограмму ожидает окончания выполнения инструкции FROM	Переход на подпрограмму выполняется немедленно.
При переходе по прерыванию возможна небольшая задержка, не более чем (800m+200)мкр сек, где m=количество 32 битных слов Гарантирует выполнение инструкции FROM в программе прерывания, не оказывая влияние на другие инструкции	Переход по прерыванию происходит немедленно. После возврата из программы прерывания возобновляется работа инструкции FROM. Если в программе прерывания запрограммирована инструкция FROM, то данный режим будет выполняться не корректно. M8028 не должен использоваться если необходима временная синхронизация.

ПРИМЕР ▾

Запись данных (ТО)

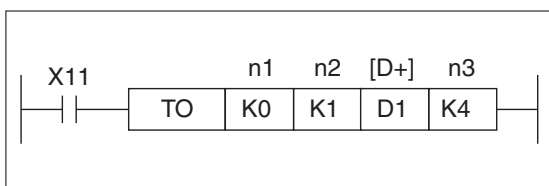


Рис. 7-29:
Пример программирования ТО-инструкции

C000266C

По примеру, показанному на рисунке, данные из регистров D1...D4 передаются к адресам буферной памяти #1...#4 специального модуля, начиная с позиционного номера 0. Значение адресации представлено по отдельности следующим образом:

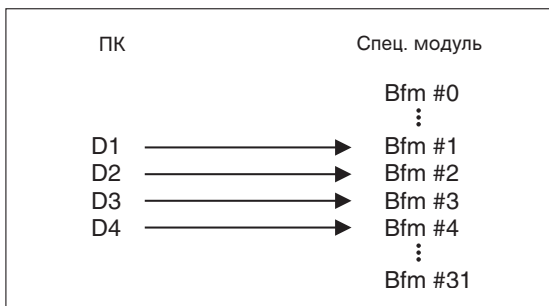


Рис. 7-30:
Адресация при ТО-инструкции

C000267C



УКАЗАНИЕ

Если должно передаваться только содержание адресов буферной памяти, то нужно занести в n3 значение 1.

7.4 КОММУНИКАЦИОННЫЕ КОМАНДЫ

Команды FNC80...89 можно применять для модулей, подключенных к последовательному порту программируемого контроллера.

ОБЗОР ИНСТРУКЦИЙ FNC 80...87

Символ	FNC	Назначение	Abschnitt
RS	80	Последовательная передача данных	7.4.1
PRUN	81	Перенос входов или меркеров	7.4.2
ASCI	82	Преобразование в ASCII-знаки	7.4.3
HEX	83	Преобразование в шестнадцатеричный код	7.4.4
CCD	84	Проверка на сумму и паритет	7.4.5
VRRD	85	Чтение задаваемых значений из FX1N-8AV-BD и FX2N-8AV-BD	7.4.6
VRSC	86	Чтение положений переключателя из FX1N-8AV-BD и FX2N-8AV-BD	7.4.7
RS2	87	Последовательная передача данных	7.4.8

Табл. 7-9: Обзор команд для последовательной коммуникации

7.4.1 ПРИЕМ И ПЕРЕДАЧА ДАННЫХ (RS)

				RS		FNC 80				
				Перенос входов или меркеров						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	●	●	
Операнды	S+	D+	n1, n2	Имп. инструкция(P)	Обработка		Шаги программы			
	D	D	K, D		16бита	32бита	RS	9		
					●					

ФУНКЦИЯ

Передача данных через последовательный интерфейс контроллера FX, например, FX2N-232BD, FX2N-32ADP, FX0N-485ADP или FX2N-485-BD.

ОПИСАНИЕ

С помощью RS-инструкции возможны прием и передача данных от большого числа приборов с последовательным интерфейсом. При этом соединение через последовательный интерфейсный адаптер осуществляется в четыре шага:

- ❶ Настройка коммутационных параметров
- ❷ Выдача RS-инструкции, состоящей из:
 - (S+) = адрес старта буфера передачи
 - n1 = количество передаваемых байтов
(FX1S/FX1N: макс. 256 байтов, FX2N/FX2NC/FX3U: макс. 4096 байтов)
 - (D+) = адрес старта буфера приема
 - n2 = количество принимаемых байтов
(FX1S/FX1N: макс. 256 байтов, FX2N/FX2NC/FX3U: макс. 4096 байтов)

Сумма n1 и n2 в контроллерах серий FX1S/FX1N не должна превышать 512 байтов, а в контроллерах серий FX2N/FX2NC и FX3U - 8000 байтов.

- ❸ Передача сообщения
- ❹ Прием сообщения



Рис. 7-31:
Пример программирования RS-инструкции

C000220C

В контроллерах серии FX3U можно также использовать команду RS2. В следующей таблице сравниваются команды RS и RS2.

Признак	Команда RS	Команда RS2
Размер метки начала	1 знак (байт)	1...4 знака (байта)
Размер метки конца	1 знак (байт)	1...4 знака (байта)
Присоединение контрольной суммы	контрольную сумму должна присоединять пользовательская программа	контрольная сумма может присоединяться автоматически *

Табл. 7-10:Различия между командами RS и RS2

* Если при команде RS2 контрольная сумма передается вместе с данными автоматически, то следует применять и метку конца

УКАЗАНИЕ

В контроллере FX3U команду RS можно использовать только для 1-го последовательного порта (кан.1).

Для одного порта разрешается активировать только одну команду RS или RS2.

В контроллерах FX3U команда RS или RS2 не должна выполняться одновременно с командами IVCK, IVDR, IVRD, IVWR или IVBWR для того же порта.

КОММУТАЦИОННЫЕ ПАРАМЕТРЫ

Каждый протокол последовательной обмена должен сначала конфигурироваться, чтобы гарантировать полную совместимость с внешним коммутационным устройством. Протокол обмена для модуля конфигурируется с помощью специального регистра D8120. Однако это возможно только для неактивной RS-инструкции. На приведенной ниже таблице показано соответствии специального регистра D8120 и его значение для порта RS232.

	Описание	0	1	
b0	Длина данных	7 бит	8 бит	
b1	Паритет	(00): нет паритета		
b2		(01): паритет по нечету (11): паритет по чету		
b3	Стоповый бит	1 бит	2 бит	
b4	Скорость (бит/с)	(0011): 300, (0100): 600,		
b5		(0101): 1200, (0110): 2400,		
b6		(0111): 4800, (1000): 9600		
b7		(1001): 19200		
b8	Стартовый байт	Нет	С D8124	
b9	Конечный байт	Нет	С D8125	
b10	Контроль линии тип 1	Нет	Н/W1	
b11	Контроль линии (DTR)	Чтение/Запись	Чтение	
b12	Контроль линии тип 2	Нет	Н/W2	
b13	FX-485 сеть	Проверка суммы	Нет проверки	Проверка
b14		Протокол	Свободный протокол	Dedicated-протокол
b15		Формат протокола	Формат 1	Формат 4

Табл. 7-11:

Биты и их соответствующее значение для RS232-коммуникации

Специальный регистр D8124 содержит значение стартового бита, в случае если он был выбран. Базовым значением является ASCII "STX" или 02_H. Однако оно может быть изменено пользователем перед началом сеанса связи.

Специальный регистр D8125 содержит значение конечного бита, в случае если он был выбран. Значением по умолчанию является ASCII "ETX" или 03_H. Однако оно может быть изменено пользователем перед началом сеанса связи.

Если периферийное коммуникационное устройство работает с техническими средствами Hardware-Handshare, то должен быть выбран этот режим работы. Если он выбирается, то берутся DST- и DTR-зажимы (зажимы 6 и 20) интерфейсного адаптера для управления коммуникацией. Диаграмма подключений находится в описании технических средств адаптера.

Детальная информация находится в fx-communication (rs232 rs485) (руководство пользователя).

СОСТАВЛЕНИЕ RS-ИНСТРУКЦИИ

В качестве формата сохранения данных имеется выбор 16-ти или 8-ми битового модуля. 16-ти битовый модуль использует старший и младший байты буферных областей передачи и приема, в то время как 8-ми битовый модуль использует только 8 младших бита. Управляется это посредством специального меркера M8161. При этом "ВКЛ" применяется для 8-ми битового модуля.

Буфер D200 K4 (16 бит)			Данные "ABCD" (8 бит)		
	Ст	Мл		Ст	Мл
D200	"B"	"A"	D200	–	"A"
D201	"D"	"C"	D201	–	"B"
			D202	–	"C"
			D203	–	"D"

Рис. 7-32:

Адресация для RS-инструкции

RS-инструкция создается, в частности, следующим образом:

- **Инструкция:**
Если RS-инструкция активна, то это означает, что коммуникация возможна и данные могут посылаться и приниматься. Внутри программы допускается больше, чем одна инструкция, при этом однако одновременно активными могут быть не больше, чем одна инструкция.
- **Стартовый адрес буфера передачи:**
Стартовый адрес буфера передачи является первым регистром данных или файловым регистром (D) области для передаваемых сообщений.
- **Длина передаваемого сообщения:**
Здесь определяется длина передаваемого сообщения. Значение может быть константой (K); но при варьируемой длине сообщений может применяться также регистр данных (D). При применении регистра данных допускаются изменения значения между отдельными требованиями передачи, однако во время самой передачи нельзя выполнять изменения.
- **Стартовый адрес буфера приема:**
Стартовый адрес буфера приема является первым регистром данных (D) области для принимаемых сообщений.
- **Длина принимаемых сообщений:**
Здесь определяется максимальная длина принимаемого сообщения. Значение может быть константой (K); но при варьируемой длине сообщений может применяться также регистр данных (D). При применении регистра данных допускаются изменение значения между отдельными процессами приема, однако во время самого приема нельзя выполнять изменения.

ПЕРЕДАЧА СООБЩЕНИЯ

Управление передачей или приемом сообщения выполняется посредством специального меркера M8122.

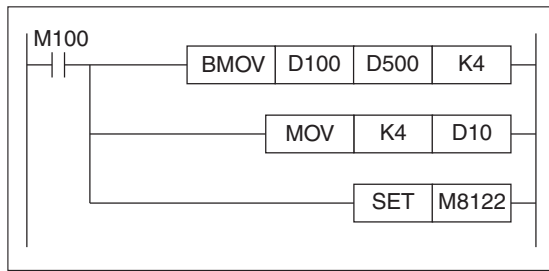


Рис. 7-33:

Пример программирования для передачи сообщения с помощью специального меркера M8122

C000175C

Сначала должна быть сохранены передаваемые данные в буферной области передачи. В них они могут передаваться двумя способами:

- ❶ Перед передачей скопируйте сообщение в буферную область сообщений или создайте их с помощью MOV- или BMOV-инструкций.
- ❷ Измените параметр RS-инструкции, чтобы получить доступ к соответствующей области регистров данных, в которой находится сообщение. Для каждого сообщения требуется собственная RS-инструкция.

В выше описанном примере данные, сохраняемые в регистрах данных D100...D103, копируются в буферную область передачи, начиная с D500. Затем определяется длина сообщений на 8 байт, в которой значение для длины передаваемого сообщения изменяется с помощью регистра данных D10.

Если данные определены и конкретно локализованы, можно включать меркер передачи M8122 по ВКЛ. Теперь начинается передача данных, и меркер M8122 автоматически отключится, как только закончится передача данных.

Рекомендуется, включить меркер одним импульсным сигналом, так как в противном случае меркер после успешной передачи снова сработает на ВКЛ. и повторится передача данных.

При применении пускового и/или конечного сигнала (Header/Terminator) они автоматически введут сообщение перед передачей.

Также возможен счетчик передач посредством специального регистра. Специальный регистр D8122 может проверяться во время процесса передачи, так что можно отслеживать процесс передачи. При полной длине сообщения в D8122 начинают фиксироваться значения и на каждый передаваемый байт данных соответственно на один уменьшается счет.

УКАЗАНИЕ

| Сигналы старта и конечный в D8122 не считаются.

ПРИЕМ СООБЩЕНИЯ

RS-инструкция автоматически управляет приемом сообщений. Как только сообщение полностью принято, данные запоминаются в буферной области приема, и включается специальный меркер M8123 в состояние ВКЛ.

В случае примененных стартового и конечного сигналов он автоматически удаляется, прежде чем сообщение будет занесено в буферную область.

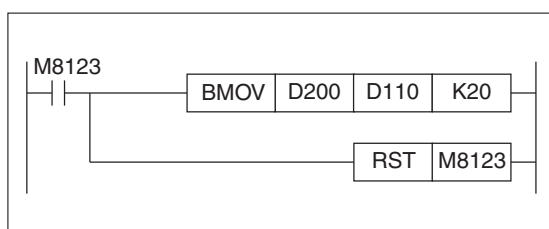


Рис. 7-34:

Пример программирования для приема сообщения с помощью специального меркера M8123

C000176C

Как только специальный меркер установится в состояние ВКЛ., данные, сохраняемые в буфере приема, обрабатываются дальше и меркер возвращается в состояние ВЫКЛ., чтобы подготовить область приема для других сообщений. Меркер отключается автоматически, если отключается RS-инструкция.

В описанном выше примере программы проверяется специальный меркер M8123 “Прием сообщения”. Если он активен, то все данные, содержащиеся в буфере приема, копируются по другому адресу и меркер приема отключаются. После освобождения области буфера приема, принимаемые данные при необходимости могут обрабатываться дальше.

УКАЗАНИЕ

Одновременно передача и прием данных невозможны. Меркер M8121 (опознание приема) находится во время процесса приема в состоянии ВКЛ. И хотя меркер M8122 (опознание передачи) может в это же время находиться в состоянии ВКЛ., но практически процесс передачи задерживается до тех пор, пока сообщение не будет полностью принято.

Также возможно создание счетчика приема. Во время приема в специальном регистре D8123 может проверяться, сколько байт актуально было передано. После полного приема сообщения показывается полная длина сообщения.

СТАРТОВЫЙ И КОНЕЧНЫЙ СИГНАЛЫ (Headers, Terminators)**● Описание**

При коммуникации данных часто требуется опознать начало и конец сообщения. Обычно это происходит с помощью определенного состояния относительно сообщения, так называемые стартовый и конечный сигналы. С помощью RS-инструкции имеется возможность автоматически добавить в сообщение стартовый байт и/или конечный байт.

Выбираются стартовый и конечные байты включением битов b8 и b9 в регистр данных D8120 параметров коммуникации.

● Во время передачи

Если был выбран стартовый сигнал, младший байт специального регистра D8124 посылается как первый байт каждого передаваемого сообщения.

Если был выбран конечный сигнал, младший байт специального регистра D8125 посылается как последний байт каждого передаваемого сообщения.

● Во время приема

Если был выбран стартовый сигнал, все принимаемые данные игнорируются до тех пор, пока принимается стартовый байт. Если стартовый байт не выбран, первый принимаемый байт считывается как содержание сообщения.

Если был выбран конечный сигнал, то относительно процесса чтения считываются все принимаемые данные как сообщения, пока не примется конечный сигнал или не будет получена полная длина сообщения, т.е. буфер приема заполнен.

Если не был выбран конечный сигнал, процесс чтения продолжается до тех пор, пока не заполнится буфер приема, т.е. сообщение должно прибыть в полной длине, прежде чем оно будет оценено как законченное.

После приема полного сообщения включается меркер M8123. Все принимаемые после этого данные игнорируются до тех пор, пока этот меркер приема снова не выключится.

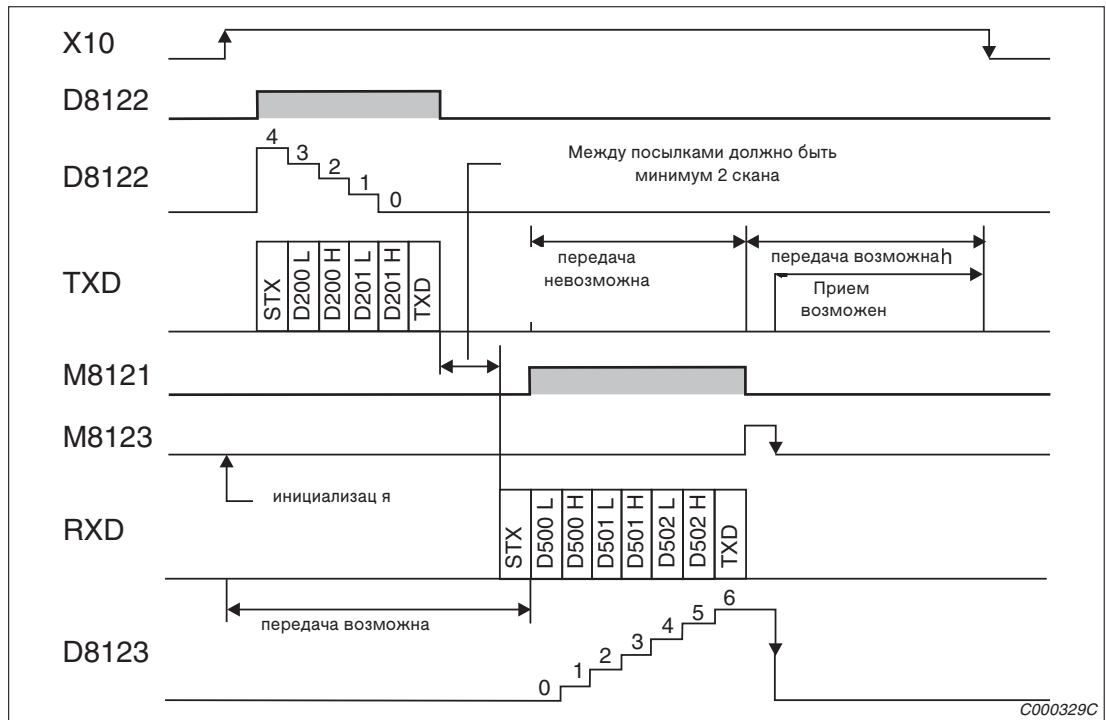


Abb. 7-35: Представление временного процесса

7.4.2 ПЕРЕНОС ВХОДОВ ИЛИ МЕРКЕРОВ (PRUN)

		PRUN		FNC 81			
		Перенос входов или меркеров					
		CPU	FX1S	FX1N	FX2N	FX3U	
			●	●	●	●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnM n = 1 – 8	KnM, KnY n = 1 – 8		●	16 Bit	32 Bit	PRUN/PRUNP
				●	●	DPRUN/ DPRUNP	9

ФУНКЦИЯ

Передача со входов в область маркеров или из маркеров в область выходов.

ОПИСАНИЕ

- PRUN-инструкция отличается от MOV-инструкции тем, что она работает в восьмеричном коде.
- Так как передача выполняется восьмерично, (S+) должно начинаться с X / M0, 10, 20, 30...

ПРИМЕР ▾

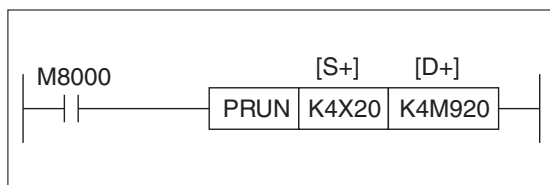
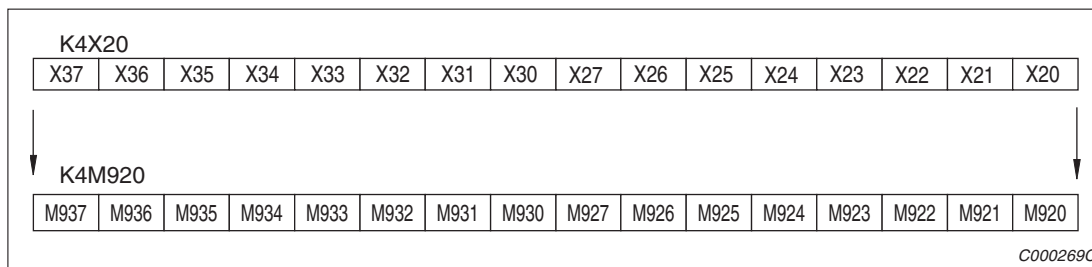


Рис. 7-34.
Пример программирования PRUN-инструкции

C000268C



C000269C

Рис. 7-35. Распределение маркеров

Маркеры M929 и M928 не записаны, так как PRUN-инструкция работает восьмерично.



7.4.3 ASCII-ПРЕОБРАЗОВАНИЕ (ASCII)

				ASCII		FNC 82		
				Преобразование в ASCII-знаки				
				CPU	FX1S	FX1N	FX2N	FX2NC
				●	●	●	●	●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, T, C, D, KnX, KnY, KnM, KnS	T, C, D, KnY, KnM, KnS	K, H		●	16бита	32бита	ASCII/ASCIP

ФУНКЦИЯ

Преобразование шестнадцатеричные значений в ASCII-знаки

ОПИСАНИЕ

ASCII-инструкция позволяет преобразование шестнадцатеричные значений из регистров данных в ASCII-знаки.

ASCII-инструкция составляется следующим образом:

- Стартовым адресом (S+), где сохраняются шестнадцатеричные данные.
- Адресом цели (D+), где должны храниться преобразованные ASCII-знаки.
- Вводом по (n) числа знаков, т.е. количество шестнадцатеричные цифр, преобразуемых в ASCII-знаки.

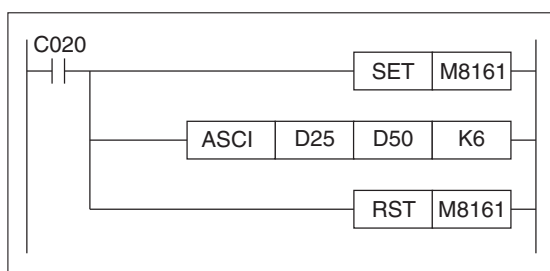


Рис. 7-38:

Пример программирования ASCII-инструкции

C000177C

Приведенная программа выполняет преобразование, если счетчик C20 достигнул свое установленное значение. Во время активного состояния ASCII-инструкции шесть (K6) шестнадцатеричные цифр из регистров данных D25 и D26 преобразуются в ASCII-знаки и запишутся в регистры данных D50...D55 (8-ми битового формата). При этом каждый знак является байтом.

В отношении формата сохранения данных имеется выбор между 16-битным и 8-битным режимом. 16-битный режим использует старший и младший байт целевых адресов (D+), а 8-битный режим - только младшие 8 битов. Для выбора режима используется специальный маркер M8161 (M8161 = "1": 8-битный режим).

УВАЖЕНИЕ

Инструкция "SET M8161" требуется только тогда, если должен применяться 8-ми битовый режим.

Специальный маркер M8161 влияет также на поведение команд RS, HEX, CCD и CRC. Если одна из этих команд применяется в той же программе, что и команда ASCII, то перед выполнением команды ASCII необходимо установить M8161, а сразу после выполнения команды ASCII снова сбросить этот маркер.

СОСТАВЛЕНИЕ ASCII-ИНСТРУКЦИИ

- **Стартовый адрес**
Определенное здесь значение обозначает первый словный операнд, который содержит преобразуемое шестнадцатеричное число. Если нужно преобразовывать больше двух чисел, то считываются следующие словные операнды, пока не будут преобразованы все желаемые цифры.
- **Адрес цели (конечный)**
Определенное здесь значение обозначает первый словный операнд, который должен содержать ASCII-знаки. Каждый словный операнд может содержать два знака (2 байта). Стартовые адреса следующих словных операндов применяются столько, пока не будут запомнены все знаки.
- **Количество знаков**
Заданное здесь значение может быть только десятичным (K) или шестнадцатеричным (H). Оно означает количество шестнадцатеричных цифр, которые должны быть преобразованы и сколько должно быть запомнено ASCII-знаков. Количество знаков может быть от 1 до 256 цифр.
- **Отключение формата хранения данных**
Функция "RST M8161" переключает формат хранения данных на исходное значение 16 бит.

УКАЗАНИЕ

Функция "RST M8161" нужна только тогда, если ASCII-инструкция использует 8-ми битовый формат хранения данных, в то время как другие инструкции применяют в своих программах 16-ти битовые форматы хранения данных.

ПРИМЕР ПРИМЕНЕНИЯ ИНСТРУКЦИИ

В соответствии с примером программы на Рис. 7-36 отображается следующая диаграмма результатов, как для 16-ти битового, так и 8-ми битового форматов.

Цель D50 (16 бит)			Знаков K6 (8 бит)	
	Ст	Мл	Ст	Мл
D50	„9“	„8“	–	„8“
D51	„A“	„0“	–	„9“
D52	„C“	„B“	–	„0“
			D53	„A“
			D54	„B“
			D55	„C“

Рис. 7-39:

Графическое представление 16-ти битового и 8-ми битового форматов.

Следующая таблица содержит ASCII-коды для всех преобразуемых шестнадцатеричных чисел, как в шестнадцатеричные так и десятичные цифры.

HEX	ASCII			HEX	ASCII			HEX	ASCII			HEX	ASCII		
	HEX	DEZI	Симв.		HEX	DEZI	Симв.		HEX	DEZI	Симв.		HEX	DEZI	Симв.
0	30	48	„0“	4	34	52	„4“	8	38	56	„8“	C	43	67	„C“
1	31	49	„1“	5	35	53	„5“	9	39	57	„9“	D	44	68	„D“
2	32	50	„2“	6	36	54	„6“	A	41	65	„A“	E	45	69	„E“
3	33	51	„3“	7	37	55	„7“	B	42	66	„B“	F	46	70	„F“

Табл. 7-12: Таблица ASCII-кодов

7.4.4 ШЕСТНАДЦАТЕРИЧНОЕ -ПРЕОБРАЗОВАНИЕ (HEX)

				HEX		FNC 83		
				Преобразование в шестнадцатеричное значение				
				CPU	FX1S	FX1N	FX2N	FX2NC
				●	●	●	●	●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, T, C, D, KnX, KnY, KnM, KnS	T, C, D, KnY, KnM, KnS	K, H		●	16бита	32бита	HEX/HEXP
					●			

ФУНКЦИЯ

Преобразование ASCII-знаков в шестнадцатеричные значения

ОПИСАНИЕ

HEX-инструкция позволяет преобразование ASCII-знаков из регистров данных в шестнадцатеричные значения.

HEX-инструкция составляется, в частности, следующим образом:

- Стартовым адресом (S+), где хранятся ASCII-знаки.
- Адресом цели (D+), где должны сохраняться преобразуемые шестнадцатеричные данные.
- Вводом по (n) числа знаков, т.е. количество ASCII-знаков, преобразуемых в шестнадцатеричные цифры.

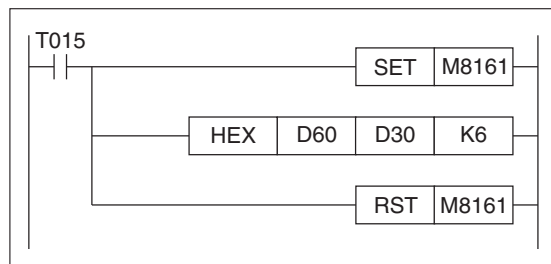


Рис. 7-40:

Пример программирования HEX-инструкции

C000178C

Приведенная на Рис. 7-38 программа выполняет преобразование, если счетчик C015 достигнул свое установленное значение. Во время активного состояния HEX-инструкции шесть (K6) ASCII-знаков из регистров данных D60 и D65 преобразуются в шестнадцатеричные цифры и запишутся в регистры данных D30...D31 (8-ми битового формата). При этом каждая цифра является байтом.

В качестве формата сохранения данных имеются по выбору 16-ти битовые или 8-ми битовые режимы. 16-ти битовый режим использует старший и младший байт примененного устройства, в то время как 8-ми битовый режим использует только младшие 8 бит. Управление этим осуществляет специальный меркер M8161. Здесь при ВКЛ. выбран 8-ми битовый режим.

УКАЗАНИЕ

Инструкция "SET M8161" требуется только тогда, если должен применяться 8-ми битовый режим.

Специальный меркер M8161 влияет также на поведение команд RS, ASCII, CCD и CRC. Если одна из этих команд применяется в той же программе, что и команда HEX, то перед выполнением команды HEX следует установить M8161, а сразу после выполнения команды HEX снова сбросить этот меркер.

СОСТАВЛЕНИЕ HEX-ИНСТРУКЦИИ

- **Стартовый адрес**
Определенное здесь значение обозначает первый словный операнд, который содержит преобразуемые ASCII-знаки. Если нужно преобразовывать больше двух знаков (2 байт), то считываются следующие словные операнды, пока не будут преобразованы все желаемые знаки.
- **Адрес цели (конечный)**
Определенное здесь значение обозначает первый словный операнд, который должен содержать шестнадцатеричные числа. Каждый словный операнд может содержать 4 цифры. Стартовые адреса следующих словных операндов применяются столько, пока не будут запомнены все цифры.
- **Количество знаков**
Заданное здесь значение может быть только десятичным (K) или шестнадцатеричным (H). Оно означает количество ASCII-знаков, которые должны быть преобразованы и сколько шестнадцатеричных цифр, должно быть запомнено. Количество знаков может быть от 1 до 256 цифр.
- **Отключение формата хранения данных**
Функция "RST M8161" переключает формат хранения данных на значение по умолчанию в 16 бит.

УКАЗАНИЕ

Функция "RST M8161" нужна только тогда, если в этой инструкции использует 8-ми битовый формат хранения данных, в то время как другие инструкции применяют в своих программах 16-ти битовые форматы хранения данных.

ПРИМЕР ПРИМЕНЕНИЯ ИНСТРУКЦИИ

В соответствии с примером программы на Рис. 7-38 отображается следующая диаграмма результатов, как для 16-ти битового, так и 8-ми битового форматов.

Цель D60 (16 бит)			Знаков K6 (8 бит)	
	Ст	Мл	Ст	Мл
			D60	– „8“
D60	„9“	„8“	D61	– „9“
D61	„A“	„0“	D62	– „0“
D62	„C“	„B“	D63	– „A“
			D64	– „B“
			D65	– „C“

Рис. 7-41:

Графическое представление 16-ти битового и 8-ми битового форматов

Следующая таблица содержит ASCII-коды для всех преобразуемых шестнадцатеричных чисел, как в шестнадцатеричные так и десятичные цифры.

HEX	ASCII		Симв.	HEX	ASCII		Симв.	HEX	ASCII		Симв.	HEX	ASCII		Симв.
	HEX	DEZI			HEX	DEZI			HEX	DEZI			HEX	DEZI	
0	30	48	„0“	4	34	52	„4“	8	38	56	„8“	C	43	67	„C“
1	31	49	„1“	5	35	53	„5“	9	39	57	„9“	D	44	68	„D“
2	32	50	„2“	6	36	54	„6“	A	41	65	„A“	E	45	69	„E“
3	33	51	„3“	7	37	55	„7“	B	42	66	„B“	F	46	70	„F“

Табл. 7-13: Таблица ASCII-кодов

7.4.5 ПРОВЕРКА НА СУММУ И ПАРИТЕТ (CCD)

				CCD		FNC 84		
				Проверка на сумму и паритет				
CPU				FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●	●	●	●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	T, C, D, KnX, KnY, KnM, KnS	T, C, D, KnY, KnM, KnS	K, H		16бита	32бита	CCD	CCDP

ФУНКЦИЯ

Расчет контрольной суммы и контроль по паритету

ОПИСАНИЕ

CCD-инструкция позволяет выполнять расчет для проверки контрольной суммы и контроль по паритету области данных.

CCD-инструкция составляется, в частности, следующим образом:

- Стартовым адресом (S+), где хранятся данные.
- Адресом цели (D+), где должно сохраняться значение проверяемой суммы.
- Вводом по (n) числа знаков, т.е. количество байт проверяемых данных.

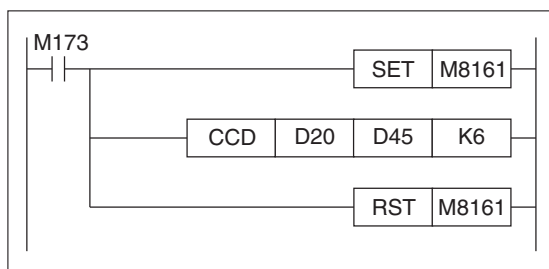


Рис. 7-42:

Пример программирования CCD-инструкции

C000180C

Приведенная на Рис. 7-40 программа выполняет проверку суммы и паритета, если включен меркер M173. Во время активного состояния CCD-инструкции суммируются шесть (K6) байт данных из регистров данных D20...D25 (8-ми битового формата) и суммируемое значение и проверка паритета сохранятся в регистре данных D45 или D46.

В качестве формата сохранения данных имеются по выбору 16-ти битовые или 8-ми битовые режимы. 16-ти битовый режим использует старший и младший байт примененного устройства, в то время как 8-ми битовый режим использует только младшие 8 бит. Управление этим осуществляет специальный меркер M8161. Здесь при его ВКЛ выбран 8-ми битовый режим.

УКАЗАНИЕ

Инструкция "SET M8161" требуется только тогда, если должен применяться 8-ми битовый режим.

Специальный маркер M8161 влияет также на поведение команд RS, ASCII, HEX и CRC. Если одна из этих команд применяется в той же программе, что и команда CCD, то перед выполнением команды CCD необходимо установить M8161, а сразу после выполнения команды CCD снова сбросить этот маркер.

СОСТАВЛЕНИЕ ССД-ИНСТРУКЦИИ

● Стартовый адрес

Определенное здесь значение обозначает первый словный операнд, который содержит суммируемые данные. Если нужно суммировать больше байт, то считываются следующие словные операнды, пока не будут обработаны все желаемые данные.

● Адрес цели (конечный)

Определенное здесь значение обозначает первые два словные операнда, которые должны содержать значения проверяемой суммы или паритета. Проверка суммы выполняется по байту (8 бит) и изображается в двухбайтовом слове (16 бит). Проверка паритета выполняется как четный контроль битового формата для каждого байта.

● Количество знаков

Заданное здесь значение может быть только десятичным (K), шестнадцатеричным (H) или регистром данных (D). Оно означает количество байт (8 бит), которые должны проверяться. Количество знаков может быть от 1 до 256 цифр.

● Отключение формата хранения данных

Функция "RST M8161" переключает формат хранения данных на значение по умолчанию в 16 бит.

УКАЗАНИЕ

Функция "RST M8161" нужна только тогда, если в этой инструкции использует 8-ми битовый формат хранения данных, в то время как другие инструкции применяют в своих программах 16-ти битовые форматы хранения данных.

ПРИМЕР ПРИМЕНЕНИЯ ИНСТРУКЦИИ

Источник D20 (16 бит)			Знаков K6 (8 бит)		
	Ст	Мл		Ст	Мл
D20	5 A	9 3	D20	–	9 3
	01011010	10010011			10010011
D21	7 4	0 F	D21	–	5 A
	01110100	00001111			01011010
D22	B 2	4 D	D22	–	0 F
	10110010	01001101			00001111
			D23	–	7 4
					01110100
			D24	–	4 D
					01001101
			D25	–	B 2
					10110010
Результат цели					
				Ст	Мл
	Сумма	D45	0 2	6 F	
			00000010	01101111	
	Паритет	D46	–	4 D	
			00000000	01001101	

Рис. 7-43:

Графическое представление 16-ти битового и 8-ми битового форматов

7.4.6 ЧТЕНИЕ ЗНАЧЕНИЯ ИЗ FX1N-8AV-BD И FX2N-8AV-BD (VRRD)

		VRRD		FNC 85				
		Чтение значения настройки у FX-8AV						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●		
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H S = 0 ... 7	KnY, KnM, KnS, T, C, D, V, Z	●	16бита ●	32бита	VRRD/ VRRDP	5	

ФУНКЦИЯ

Чтение значения настройки, установленного на FX1N-8AV-BD и FX2N-8AV-BD в системе управления FX-ПК

ОПИСАНИЕ

С помощью VRRD-инструкции преобразовывается настройка потенциометра (S+) в 8-ми битном значении и запоминается в (D+).

УКАЗАНИЕ

Потенциометры нумеруются от 0 до 7.

Модули FX1N-8AV-BD и FX2N-8AV-BD вставляются в платоместо расширения FX1S/FX1N- или FX2N-CPU.

ПРИМЕР ▾

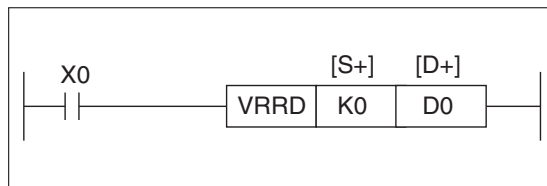


Рис. 7-44:

Пример программирования для VRRD-инструкции

C000270C

Считываются показания потенциометра "0" и установленное значение запоминается в регистре данных D0. △

7.4.7 ЧТЕНИЕ ПОЛОЖЕНИЯ ПЕРЕКЛЮЧАТЕЛЯ ИЗ FX1N-8AV-BD И FX2N-8AV-BD (VRSC)

		VRSC		FNC 86				
		Чтение положения переключателя у FX-8AV						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●		
Операнды	S+	D+	Имп. инструкция(P)		Обработка		Шаги программы	
	K, H S = 0 - 7	KnY, KnM, KnS, T, C, D, V, Z	●		16бита ●	32бита	VRSC/ VRSCP	5

ФУНКЦИЯ

Чтение значения включения, установленного на FX1N-8AV-BD и FX2N-8AV-BD в системе управления FX-ПК

ОПИСАНИЕ

- С помощью VRSC-инструкции считывается положение переключателя (от 0 до 10) потенциометра (S+) и запоминается в (D+).

УКАЗАНИЕ

Потенциометры нумеруются от 0 до 7.

Модули FX1N-8AV-BD и FX2N-8AV-BD вставляются в слот расширения FX1S/FX1N- или FX2N-CPU.

ПРИМЕР ▾

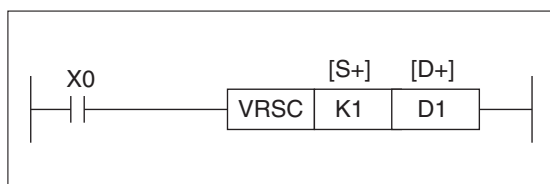


Рис. 7-45:

Пример программирования для VRSC-инструкции

C000271C

Положение переключения переключателя "1" запоминается в регистре данных D1.



7.4.8 ПОСЛЕДОВАТЕЛЬНАЯ ПЕРЕДАЧА ДАННЫХ (RS2)

					RS2		FNC 80			
					Serielle Datenübertragung					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
										●
Operanden	S+	D+	n1, n2	n3	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R	D, R	K, H, D, R	K, H		16бита	32бита	RS2	11	
						●				

Функция

Передача данных через последовательный интерфейс контроллера FX3U

Описание

(S+): начальный адрес области регистра данных, в которой сохранены передаваемые данные

n1: количество передаваемых байтов (диапазон: от 0 до 4096 байтов)

(D+): начальный адрес принимающего буфера

n2: количество принимаемых байтов (диапазон: от 0 до 4096 байтов)

n3: номер последовательного интерфейса (K1 = Ch. 1, K2 = Ch. 2)

Сумма n1 и n2 не должна превышать 8000 байтов.

Можно использовать только следующие коммуникационные адаптеры:

- FX3U-232-BD, FX3U-232ADP
- FX3U-485-BD, FX3U-485ADP

Последовательная передача управляется специальными маркерами и регистрами:

Специальные маркеры			Специальные регистры		
Ch. 1	Ch. 2	Значение	Ch. 1	Ch. 2	Значение
			D8400	D8420	Формат передачи
M8401	M8421	Ожидание передачи данных			
M8402	M8422	Запрос передачи данных	D8402	D8422	Количество байтов, которое еще осталось передать
M8403	M8423	Прием данных окончен	D8403	D8423	Количество принятых байтов
M8404	M824	Распознавание несущей частоты			
			D8405	D8425	Индикация параметров коммуникации
M8409	M8429	Превышение времени	D8409	D8429	Время контроля
			D8410	D8430	Метка конца 1, 2
			D8411	D8431	Метка конца 3, 4
			D8412	D8432	Метка конца 1, 2
			D8413	D8433	Метка конца 3, 4
			D8414	D8434	Принятая контрольная сумма
			D8415	D8435	Рассчитанная контрольная сумма
			D8416	D8436	Переданная контрольная сумма
			D8419	D8439	Индикация режима
M8063	M8438	Ошибка при последовательной коммуникации	D8063	D8438	Код ошибки

Табл. 7-15: Специальные маркеры и регистры для последовательного соединения

Вместо команды RS2 для интерфейса Ch. 1 можно использовать и команду RS (раздел 7.4.1). В следующей таблице сравниваются команды RS и RS2.

Свойство	Команда RS	Команда RS2
Размер метки начала	1 знак (байт)	от 1 до 4 знаков (байтов)
Размер метки конца	1 знак (байт)	от 1 до 4 знаков (байтов)
Присоединение контрольной суммы	Контрольную сумму должна добавлять пользовательская программа	Контрольная сумма может добавляться автоматически*.

Табл. 7-14: Различия между командами RS и RS2

* Если в случае команды RS2 контрольная сумма передается вместе с данными автоматически, то должна применяться и метка конца передачи.

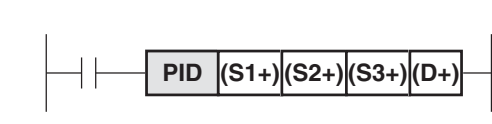
УКАЗАНИЕ

Для одного интерфейса разрешается активировать только одну команду RS или RS2.

Команду RS или RS2 нельзя выполнять одновременно с командами IVCK, IVDR, IVRD, IVWR или IVBWR для того же интерфейса.

Описание последовательной коммуникации имеется в разделе (команда RS) и руководстве по коммуникации для контроллеров MELSEC семейства FX, артикул 137315.

7.5 ПРОГРАММИРОВАНИЕ ЗАМКНУТОГО КОНТУРА РЕГУЛИРОВАНИЯ (PID)

				PID		FNC 88				
				Контроль контура регулирования						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					● ^①	●	●	●	●	●
Operanden	S1+, S2+	S3+	D+	Имп. инструкция(P)	Обработка		Шаги программы			
	D	D [®]	D		16бита	32бита	PID	9		
					●					

① ПК серии FX1S не имеют аналоговых входов/выходов, поэтому необходимо использовать RS232 порт.

② Допустимо использование регистров от D0 до D975 .

ФУНКЦИЯ

Программирование замкнутого контура регулирования при задании 25 параметров регулирования.

ОПИСАНИЕ

- С помощью PID-инструкции можно проследить регулирование текущего значения в замкнутом контуре регулирования.
- (S1+) указывает задаваемое значение контура регулирования.
- (S2+) считывает действительное значение как обратную связь для регулятора.
- (S3+) - это начальный адрес области регистра, в которой сохраняются параметры контура регулирования. Для этого имеются 25 следующих друг за другом регистров данных. Обзор параметров дан в таблице на следующей странице.
- В (D+) записываются расчетные выходные значения, которые выдаются по процессу регулирования.

PID-инструкция работает со следующей математической формуле для расчета выходных значений:

$$\text{Выходные значения} = K_p \left\{ \varepsilon + K_D T_D \frac{d\varepsilon}{dt} + \frac{1}{T_I} \int \varepsilon dt \right\}$$

Здесь обозначены:

- K_p = коэффициент пропорциональности
- ε = отклонения
- K_D = дифференциальный коэффициент
- T_D = дифференциальная временная константа
- T_I = интегральная константа

S3+№ пар	Обозначение	Описание	значения
+0	Время отсчета-выборки	Интервал проверки процесса изменения действительного значения	1 – 32767 мс
+1	Направление движения/ контроль аварии	Бит 0: 0 = вперед; 1 = назад	—
		Бит 1: 0/1 = вперед; 1 = назад	
		Бит 2: 0/1 = вперед; 1 = назад	
		Бит 3: Резерв	
		Бит 4: Функция автоматической отработки ^① 1 = пуск; 0 = отключение	
		Бит 5: Ограничение выходных значений (FX2N) 1 = активизировано; 0 = деактивизировано	
Бит 6: Резерв			
+2	Входной фильтр (α)	Установленное значение для входного фильтра	0 – 99 %
+3	Коэффициент пропорциональности (K_p)	Коэффициент умножения для пропорционального регулирования	1 – 32767 %
+4	Интегральная временная константа (T_I)	Коэффициент для умножения обратных значений при интегральном регулировании. Выбор значения нуля препятствует интегральному регулированию.	0 – 32767 x 100 мс
+5	Дифференциальный Коэффициент (K_D)	Коэффициент умножения дифференциального регулирования	0 – 100 %
+6	Дифференциальная временная константа (T_D)	Коэффициент умножения дифференциального регулирования. Выбор значения нуля препятствует дифференциальному регулированию.	0 – 32767 x 10 мс
+7 – +19	Зарезервировано	—	—
+20	Контрольное значение для текущего значения аварийного сообщения (возрастающего)	Выдача аварии, если превышено текущее значение этого контролируемого значения	0 – 32767
+21	Контрольное значение для текущего значения аварийного сообщения (падающего)	Выдача аварии, если уменьшено текущее значение этого контролируемого значения	0 – 32767
+22	Контрольное значение для выходного значения аварийного сообщения (возрастающего)	Выдача аварии, если превышено выходное значение этого контролируемого значения	0 – 32767
	Верхнее ограничение выходного значения (FX2N)	Пользователем установлено верхнее ограничение выходного значения (D+). (Активно, если введен бит 5 в (S3+)+1)	-32768 – 32767
+23	Контрольное значение для выходного значения аварийного сообщения (падающего)	Выдача аварии, если уменьшено выходное значение этого контролируемого значения	0 – 32767
	Нижнее ограничение выходного значения (FX2N)	Пользователем установлено нижнее ограничение выходного значения (D+). (Активно, если введен бит 5 в (S3+)+1)	-32768 – 32767
+24	Выдача аварии	Бит 0: авария текущего значения (превышено)	—
		Бит 1: авария текущего значения (занижено)	
		Бит 2: авария выходного значения (превышено)	
		Бит 3: авария выходного значения (занижено)	

Табл. 7-16: Обзор параметров регулирования

- ① Контроллеры MELSEC семейства FX имеют функцию автонастройки. Эта функция определяет исходные значения параметров регулирования K_p ((S3+)+3), T_I ((S3+)+4), T_D ((S3+)+6) и направление регулирования ((S3+)+1), бит (0). Все другие параметры должен указать пользователь. Функция автонастройки активируется установкой бита 4 в ((S3+)+1). Выходное значение (MV) удерживается в зоне указанного начального значения. Ответ регулируемой системы (PV) контролируется. Если это значение достигает $1/3$ от заданного значения (SV), функция автонастройки отключается и бит 4 в ((S3+)+1) сбрасывается.

НЕОБХОДИМО СЛЕДИТЬ ЗА СЛЕДУЮЩИМИ ТОЧКАМИ ПРИ ПРИМЕНЕНИИ ФУНКЦИИ AUTO-TURNING.

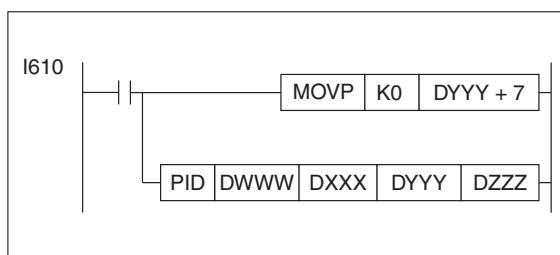
- Различие между текущим значением (CV) и заданным значением (SV) должно составлять 150%.
- Время (интервал) проверки должен быть кратным времени цикла (скана) и больше 1 с.
- Перед пуском функции Auto-Tuning система должна быть стабильной.

Значения параметров могут записываться непосредственно в регистр данных. Если данные должны сохраняться при отключении питающего напряжения, то должна применяться буферная батарея для регистров данных. Другой возможностью является занесение данных параметров в файловый регистр и запись по BMOV-инструкции (BMOV, FNC 15) в желаемый регистр данных. Это имеет то преимущество, что можно сохранять несколько наборов с данными параметрами, и по потребности их можно обменивать по командам программы.

УКАЗАНИЕ

Количество PID-инструкций в программе неограничено. Однако нужно следить за тем, чтобы не была пересечена область данных (S3+), чтобы избежать конфликтов данных внутри управления.

PID-инструкция включается в прерывание, в подпрограммы, в STL-программу или в программу перехода. При этом PID-инструкция должна предварительно установить MOVP-инструкцию. В этой MOVP-инструкции должно записываться K0 в ((S3+)+7), чтобы предупредить возможные ошибки программы.

**Рис. 7-46:**

Пример программирования для PID-инструкции с предварительно установленной MOVP-инструкцией

C000330C

Интервал времени проверки (Sampling-TIME) (Ts) должен быть выбран больше, чем время цикла программы, так как иначе это приведет к ошибке. Если это не выполнить, время проверки автоматически станет равным времени цикла. При использовании инструкции прерывания от I6XX до I8XX время проверки должно быть не меньше, чем время цикла прерывания.

Интервал времени проверки (Sampling-TIME) (Ts) может варьироваться на основе скана программы. Максимальная область отклонения лежит в пределах от (Ts - (время цикла программы)) до (Ts + (время цикла программы)). Это отклонение может минимизироваться благодаря введению PID-инструкции внутрь тактируемой программы прерывания.

PID-инструкция позволяет отображать аварийные сообщения при нарушениях в процессе работы. Эти аварийные сообщения могут активизироваться и деактивизироваться пользователем. Также может настраиваться по определенному пользователем параметру, при каком состоянии должно генерироваться аварийное сообщение.

PID-инструкция сохраняет определенные сообщения об ошибках, которые помогут решить возникающие проблемы. Эти сообщения запоминаются в регистре данных D8067. Если появилась ошибка, она отображается благодаря включению специального меркера M8067. Таблица 7-12 содержит обзор сообщений об ошибках и их значение.

Так как ПК серии FX1S не может читать никакие аналоговые значения, то ПК должен получать данные для регулирования по последовательным интерфейсам.

Сообщение об ошибках	Описание	Влияние на PID-инструкцию
K6705	PID-инструкция не указывает регистр данных.	Останов выполнения
K6706	Указанный регистр данных лежит в вне допустимой области.	
K6730	Время проверки (sampling) (T_s) лежит в вне допустимой области ($T_s < 0$).	
K6732	Значение настройки для входного фильтра (α) лежит в вне допустимой области ($\alpha < 0$ или $\alpha > 100$).	
K6733	Коэффициент пропорциональности (K_p) лежит в вне допустимой области ($K_p < 0$).	
K6734	Константа времени интегрирования (T_i) лежит в вне допустимой области ($T_i < 0$).	
K6735	Дифференциальный коэффициент (K_D) лежит в вне допустимой области ($K_D < 0$ или $K_D \geq 101$).	
K6736	Константа времени дифференцирования (T_D) лежит вне допустимой области ($T_D < 0$).	Время проверки (sampling) равно времени цикла программы, и выполнение продолжается
K6740	Время проверки (sampling) (T_s) \leq времени цикла программы	
K6742	Изменение текущего значения лежит в вне допустимой области (Δ текущего значения < -32768 или Δ текущего значения $> +32767$).	Используемые данные введены больше или меньше разрешенных граничных значений, и выполнение продолжается
K6743	Отклонение лежит в вне допустимой области ($\epsilon < -32768$ или $\epsilon > +32767$).	
K6744	Результат интегрирования лежит в вне допустимой области ($-32768 - +32767$).	
K6745	Дифференциальный коэффициент (K_D) лежит выше или дифференциальное значение лежит в вне допустимой области.	
K6746	Дифференциальный результат лежит в вне допустимой области ($-32768 - +32767$).	
K6747	Общий результат PID лежит в вне допустимой области ($-32768 - +32767$).	
K6750	Заданное значение - текущее значение < 150 или контур регулирования нестабилен (Разность между заданным и текущим значениями колеблется очень сильно и очень быстро)	Выполнение заканчивается
K6751	Заданное значение значительно завышено	
K6752	Большие колебания при функции автоматической настройки (Auto-Tuning)	

Табл. 7-17: Обзор сообщений об ошибках, запоминаемых в регистре D8067

ПРИМЕР ▾

Ввод PID-инструкции

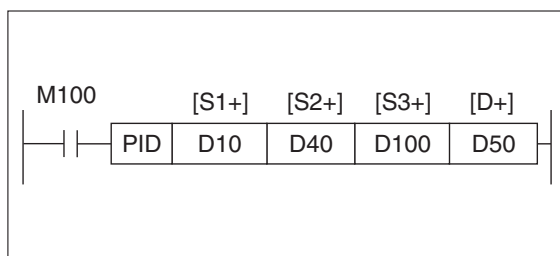


Рис. 7-47:
Пример программирования PID-инструкции

C000331C

PID-инструкция начнет выполняться, как только включится M100. Задаваемое значение запомнено в D10, текущее значение считывается в D40, выходные значения записываются в D50. Параметры регулирования хранятся в регистрах данных D100 и D124.

Если должно регулироваться аналоговое значение, то основным устройством - ПК считываются два дополнительных специальных модуля для обработки их в системе управления и для введения в систему управления выдаваемых сигналов.

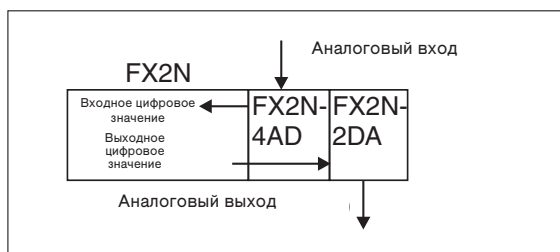


Рис. 7-48:
Пример аналоговой системы

C000332C

У первого специального модуля преобразователь A/D типа FX2N-4AD, преобразует аналоговые данные в цифровые, т.е. оцифровываются аналоговые данные процесса. Базовое устройство (ПК) применяет FROM-инструкцию, чтобы занести эти данные в D40. Примером аналоговых текущих значений может быть переменная температура и соответственно переменный сигнал напряжения.

У второго специального модуля преобразователь A/D типа FX2N-2DA, выдает в D50 аналоговые выходные значения, преобразованные из хранимых цифровых значений. Примером аналогового регулирования может быть исполнительный гидрораспределитель, значение места поршня которого зависит от уровня напряжения.

Параметры регулирования зависят от специфических данных системы регулирования.

Другой возможностью для получения выходного значения регулировки является PWM-инструкция (PWV, FNC 58), которая выдает последовательно поступающие импульсы. Ширина импульсов может определяться выходными значениями.

НАСТРОЙКА PID-ИНСТРУКЦИИ

Метод регулировки	Выбор по регистру данных			Описание
	(S3+)+3(K _p)	(S3+)+4(T _i)	(S3+)+6(T _d)	
P	Значение пользователя	Установка на ноль	Установка на ноль	Пропорциональное регулирование
PI	Значение пользователя	Установка на ноль	Установка на ноль	Пропорциональное и интегральное регулирование
PD	Значение пользователя	Установка на ноль	Значение пользователя	Пропорциональное и дифференциальное регулирование
PID	Значение пользователя	Значение пользователя	Значение пользователя	Полное PID-регулирование

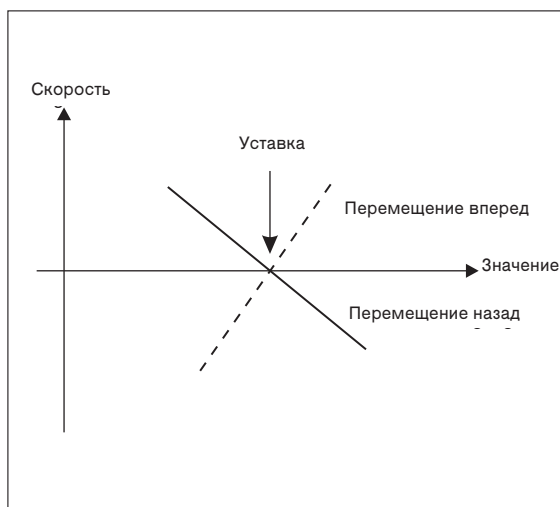
Табл. 7-18: Настройка PID-инструкции

ОПЕРАЦИИ ДВИЖЕНИЯ ВПЕРЕД И НАЗАД ((S3+)+1), b0

Обозначение операций вперед-назад появляется по имени. По представлению происходит движение точки в системе координат до следующей точки. Для обоих параметров может представиться следующая ситуация:

- Текущее значение (CV, регистр данных (S2+)) больше, чем заданное значение (SP или регистр данных (S1+)).
- Текущее значение (CV, регистр данных (S2+)) меньше, чем заданное значение (SP или регистр данных (S1+)).

На следующем рисунке изображена система координат, в которой координатные оси означают заданное (или выходное) значение контура регулирования, а также фактическое значение. Пунктирная линия на диаграмме отображает движение вперед, а сплошная линия - движение назад.

**Рис. 7-49:**

Представление движений вперед/назад в координатной системе

C000343C

Движение вперед при большом текущем значении состоит в регулировке вниз и назад текущего значения или при малом текущем значении в регулировке вверх и вперед текущего значения.

Движение назад имеется тогда, если при очень большом текущем значении состоит оно регулируется вниз и вперед или же при очень малом текущем значении оно регулируется вверх и назад.

На графике не принимается во внимание никаких коэффициентов корректировки P, I или D или комбинация из них.

7.6 ОБМЕН ДАННЫХ С ИНДЕКСНЫМ РЕГИСТРОМ

С помощью команд ZPUSH и ZPOP можно сохранить или восстановить содержимое индексных регистров V0...V7 и Z0...Z7.

Символ	FNC	Значение	Раздел
ZPUSH	102	Сохранить содержимое индексных регистров	7.6.1
ZPOP	103	Восстановить содержимое индексных регистров	7.6.2

Табл. 7-19: Обзор команд обмена данными для индексных регистров

7.6.1 СОХРАНЕНИЕ СОДЕРЖИМОГО ИНДЕКСНЫХ РЕГИСТРОВ (ZPUSH)

		ZPUSH		FNC 102			
		Сохранение содержимого индексных регистров					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Операнды	D+	Имп. инструкция(P)		Обработка		Шаги программы	
	D, R	●		16бита	32бита	ZPUSH ZPUSHP	3

Принцип работы

Перенос содержимого индексных регистров в область регистров данных

Описание

- Команда ZPUSH передает содержимые индексных регистров V0...V7 и Z0...Z7 в область регистров данных, первый адрес которой указывается начиная с (D+).
- Для передачи сохраненных значений из области регистров данных в индексные регистры применяется команда ZPOP (раздел 7.6.2).
- Операнд ((D+)+0) содержит значение счета, которое указывает, сколько раз были сохранены содержимые индексных регистров. При каждом выполнении команды ZPUSH это значение повышается на 1, а при каждом выполнении команды ZPOP - понижается на 1.
- В операнды с ((D+)+1) по ((D+)+16) записываются содержимые индексных регистров.
- При применении этих команд на различных уровнях (вложениях) для каждого выполнения команды ZPUSH необходимо место размером в 16 регистров за операндом, указанным в (D+). По этой причине начиная с (D+) необходимо предусмотреть область памяти, согласованную с выполнениями команды ZPUSH.



Рис. 7-51: Действие команд ZPUSH/ZPOP без вложения

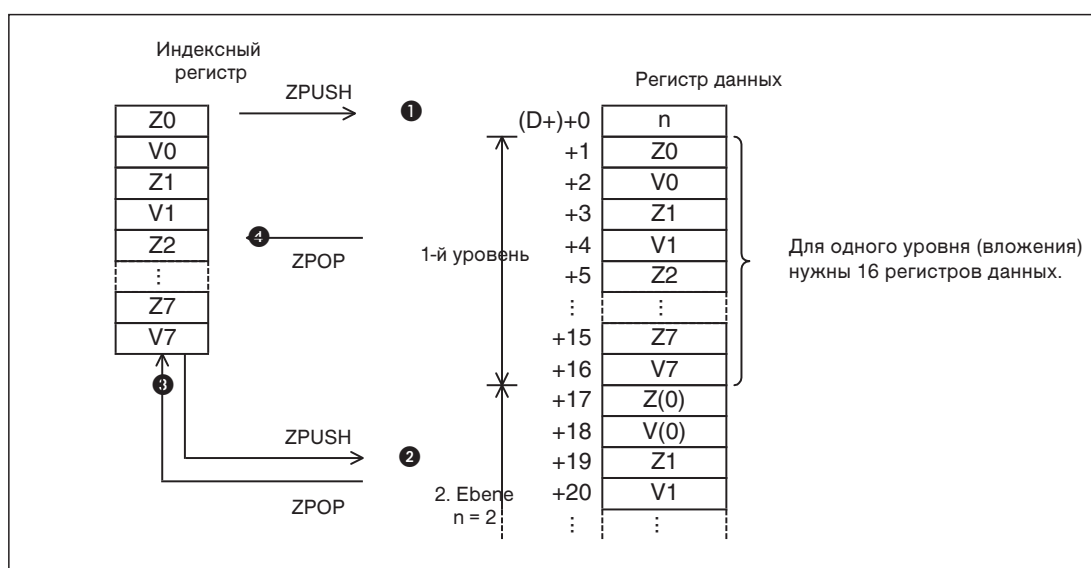


Рис. 7-50: Действие команд ZPUSH/ZPOP с вложениями

- 1 Выполняется команда ZPUSH для 1-го уровня вложения. В ((D+)+0) вводится значение "1".
- 2 Выполняется команда ZPUSH для 2-го уровня вложения. Содержимые индексных регистров вводятся за данными, сохраненными перед этим. Содержимое ((D+)+0) повышается на значение "2".
- 3 С помощью команды ZPOP данные 2-го уровня снова передаются в индексные регистры. В ((D+)+0) вводится значение "1".
- 4 Команда ZPOP передает данные 1-го уровня вложения из регистров данных в индексные регистры. Содержимое ((D+)+0) становится равным "0".

УКАЗАНИЕ

Если уровни вложения не используются, то перед вызовом команды ZPUSH необходимо стереть содержимое ((D+)+0).

Если используются уровни вложения, то содержимое ((D+)+0) необходимо стереть перед первым вызовом команды ZPUSH.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- Количество операндов, начиная с (D+), превышает допустимую область операндов (код ошибки 6706).
- В случае выполнения команды ZPUSH регистр ((D+)+0) (число сохранений) содержит отрицательное значение (код ошибки 6707).

ПРИМЕР ▾

В следующей программе перед вызовом подпрограммы содержимое индексных регистров V0...V7 и Z0...Z7 вводится в регистры данных, начиная с D0. Перед выполнением подпрограммы индексные регистры необходимо "спаcать", так как они применяются и в подпрограмме.

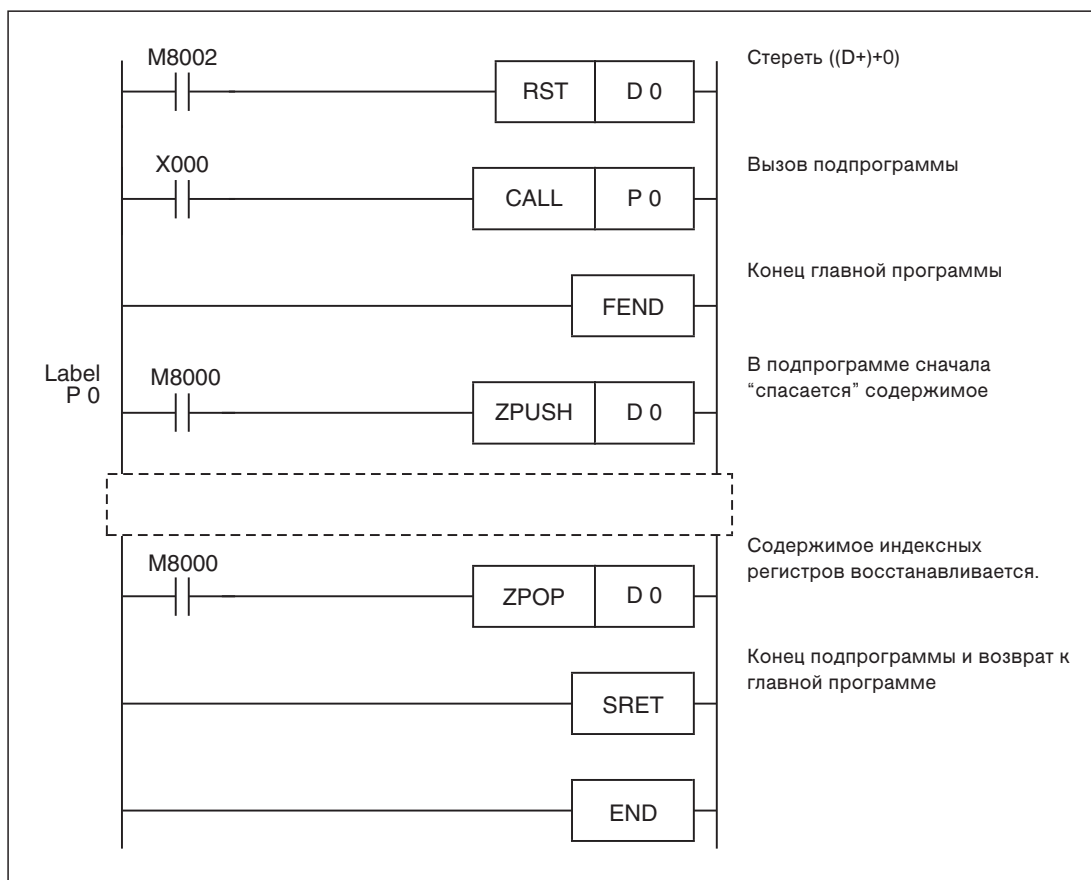


Рис. 7-52: Пример применения команд ZPUSH и ZPOP

7.6.2 ВОССТАНОВЛЕНИЕ СОДЕРЖИМОГО ИНДЕКСНЫХ РЕГИСТРОВ (ZPOP)

		ZPOP		FNC 103			
		Восстановление содержимого индексных регистров					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Операнды	D+	Имп. инструкция(P)		Обработка		Шаги программы	
	D, R	●		16бита	32бита	ZPOP ZPOPP	3
			●				

Принцип работы

Перенос содержимого индексных регистров, сохраненного с помощью команды ZPUSH, из области регистров данных обратно в индексные регистры.

Описание

- Команда ZPOP передает из области регистров данных, первый адрес которой указывается начиная с (D+), хранящиеся в ней содержимые индексных регистров V0...V7 и Z0...Z7 обратно в индексные регистры.
- Для сохранения содержимого индексных регистров применяйте команду ZPUSH (раздел 7.6.1).
- Операнд ((D+)+0) содержит значение счета, которое указывает, сколько раз были сохранены содержимые индексных регистров. При каждом выполнении команды ZPUSH это значение повышается на 1, а при каждом выполнении команды ZPOP - понижается на 1.

Команды ZPUSH и ZPOP используются парами. Принцип действия и пример программы описаны в предыдущем разделе (команда ZPUSH).

Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- При выполнении команды ZPOP регистр ((D+)+0) (число сохранений) содержит значение "0" или отрицательное значение (код ошибки 6706).

7.7 ИНСТРУКЦИИ НАД ЧИСЛАМИ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (FX2N)

инструкции с числами с плавающей запятой работают только с контроллерами серий FX2N, FX2NC и FX3U

ОБЗОР ИНСТРУКЦИЙ FNC 110...132

Символ	FNC	Назначение	Разд.
DECMP	110	Сравнение чисел с плавающей запятой	7.7.1
DEZCP	111	Сравнение чисел с плавающей запятой в области	7.7.2
DEMOV	112	Перенос чисел с плавающей запятой	7.7.3
DESTR	116	Преобразование числа с плавающей запятой в строковую величину	7.7.4
DEVAL	117	Преобразование строковой величины в числа с плавающей запятой	7.7.5
DEBCD	118	Преобразование чисел с плавающей запятой в научный формат чисел	7.7.6
DEBIN	119	Преобразование научного формата чисел в числа с плавающей запятой	7.7.7
DEADD	120	Сложение чисел с плавающей запятой	7.7.8
DESUB	121	Вычитание чисел с плавающей запятой	7.7.9
DEMUL	122	Умножение чисел с плавающей запятой	7.7.10
DEDIV	123	Деление чисел с плавающей запятой	7.7.11
DEXP	124	Число с плавающей запятой в виде показателя степени по основанию e	7.7.12
DLOGE	125	Вычисление натурального логарифма	7.7.13
DLOG10	126	Вычисление десятичного логарифма	7.7.14
DESQR	127	Корень квадратный из числа с плавающей запятой	7.7.15
DENEG	128	Изменение арифметического знака чисел с плавающей запятой	7.7.16
INT	129	Преобразование формата с плавающей запятой в десятичный формат	7.7.17
SIN	130	Расчет синуса числа с плавающей запятой	7.7.18
COS	131	Расчет косинуса числа с плавающей запятой	7.7.19
TAN	132	Расчет тангенса числа с плавающей запятой	7.4.20
DASIN	133	Вычисление арксинуса	7.7.21
DACOS	134	Вычисление арккосинуса	7.7.22
DATAN	135	Вычисление арктангенса	7.7.23
DRAD	136	Пересчет из градусов в радианы	7.7.24
DDEG	137	Пересчет из радиан в градусы	7.7.25

Табл. 7-20: Обзор инструкций FNC 110...132

7.7.1 СРАВНЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DECMP)

				DECMP		FNC 110			
				Сравнение чисел с плавающей запятой					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S1+	S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	К, Н Целые числа автоматически конвертируются в числа с плавающей запятой (число с плавающей запятой (32 бита))				Y, M, S, Применяется три один за другим следующих адреса операндов	16бита	32бита	DECMP	13
				●		●	DECMP	13	

ФУНКЦИЯ

Сравнение двух чисел с плавающей запятой с выдачей результата сравнения

ОПИСАНИЕ

- DECMP-инструкция сравнивает число с плавающей запятой в (S1+) с числом с плавающей запятой в (S2+).
- Результат сравнения сохраняется в соответственно 3-х один за другим следующих операндах.
- Если число в (S2+) меньше числа (S1+), то включается битовый операнд (D+).
- Если число в (S2+) равно числу (S1+), то включается битовый операнд ((D+)+1).
- Если число в (S2+) больше числа (S1+), то включается битовый операнд ((D+)+2).

УКАЗАНИЕ

Опрошенные операнды выходов остаются включенными после отключения условий выполнения DECMP-инструкции.

Сравнение выполняется алгебраически (например, опознается, что $-1,79 \times 10^{27}$ меньше чем $9,43 \times 10^{-15}$.)

ПРИМЕР ▾

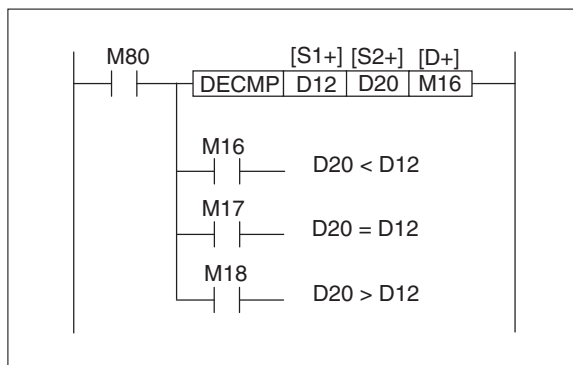


Рис. 7-53

Пример программирования DECMP-инструкции

C000350C

При установке маркера M80 число с плавающей запятой, указанное начиная с D20 (S2+), сравнивается с числом с плавающей запятой, указанным начиная с D12 (S1+).

Если число в D20 меньше числа D12, то включается меркер M16.

Если число в D20 равно числу D12, то включается меркер M17.

Если число в D20 больше числа D12, то включается меркер M18.

7.7.2 ЗОННЫЙ КОМПАРАТОР (DEZCP)

					DEZCP		FNC 111		
					Сравнение числа с плавающей запятой с областью				
CPU					FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S1+	S2+	S3+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	К, Н Целые числа автоматически конвертируются в числа с плавающей запятой. D (число с плавающей запятой (32 бита))			Y, M, S, Применяется три один за другим следующих адреса операндов		●	16бита	32бита	DEZCP
							●	DEZCPP	13

ФУНКЦИЯ

Сравнение чисел с плавающей запятой с выделенной (указанной) областью с выдачей результата сравнения

ОПИСАНИЕ

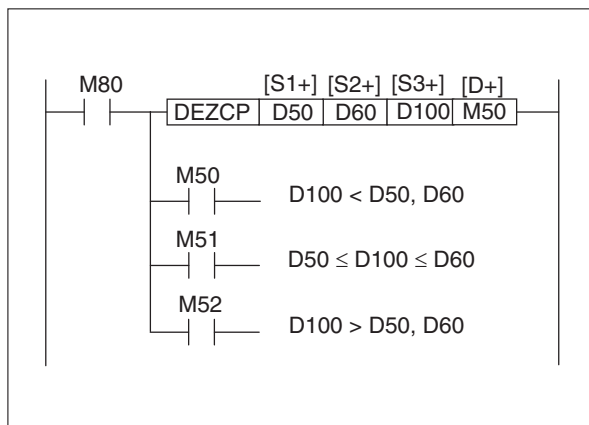
- DEZCP-инструкция сравнивает число с плавающей запятой в (S3+) с областью между (S1+) и (S2+).
- Результат сравнения сохраняется в соответственно 3-х один за другим следующих операндах.
- Если число в (S3+) меньше чисел между (S1+) и (S2+), то включается битовый операнд (D+).
- Если число в (S3+) равно числу между (S1+) и (S2+), то включается битовый операнд ((D+)+1).
- Если число в (S3+) больше числа между (S1+) и (S2+), то включается битовый операнд ((D+)+2).

УКАЗАНИЕ

Опрошенные операнды выходов остаются включенными после отключения условий выполнения DEZCP-инструкции.

Сравнение выполняется алгебраически (например, опознается, что $-1,79 \times 10^{27}$ меньше, чем $9,43 \times 10^{-15}$.)

ПРИМЕР ▾

**Рис. 7-54:**

Пример программирования
DEZCP-инструкции

C000351C

При установке маркера M80 число с плавающей запятой, указанное начиная с D100 (S3+), сравнивается с числами в области между D50 (S1+) и D60 (S2+).

Если число в D100 меньше числа между D50 и D60, то включается меркер M50.

Если число в D100 равно числу между D50 и D60, то включается меркер M51.

Если число в D100 больше числа между D50 и D60, то включается меркер M52.

△

7.7.3

ПЕРЕНОС ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEMOV)

		DEMOV		FNC 112				
		Перенос чисел с плавающей запятой						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	S1+	D+	Имп. инструкция(P)		Обработка		Шаги программы	
	D, R, специальные модули (U□/G□), константа с плавающей запятой Используются 2 следующих друг за другом адреса операндов.	D, R, специальные модули (U□/G□)	●	16бита	32бита	DMOV	9	
					DMOVP	9		

Функция

Передача числа с плавающей запятой из источника данных в место назначения

Описание

- При выполнении команды DEMOV содержимое (S+) и ((S+)+1) передается в (D+) и ((D+)+1).

УКАЗАНИЕ

Команда DMOV выполняется в каждом программном цикле. Команда DMOVP выполняется только при положительном фронте входного сигнала.

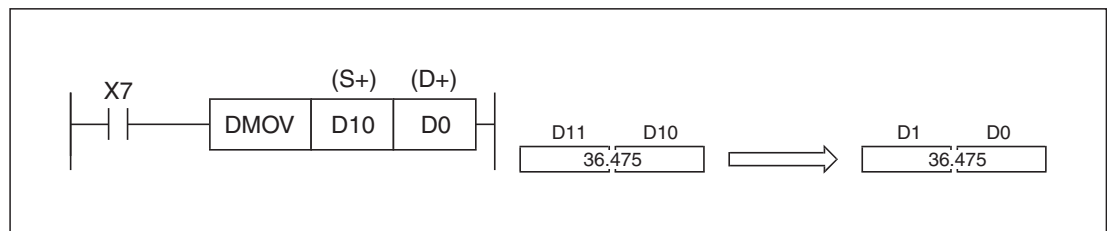
ПРИМЕР ▾

Рис. 7-55: В этом примере, если вход X7 включен, содержимое D10 и D11 передается в D0 и D1.

△

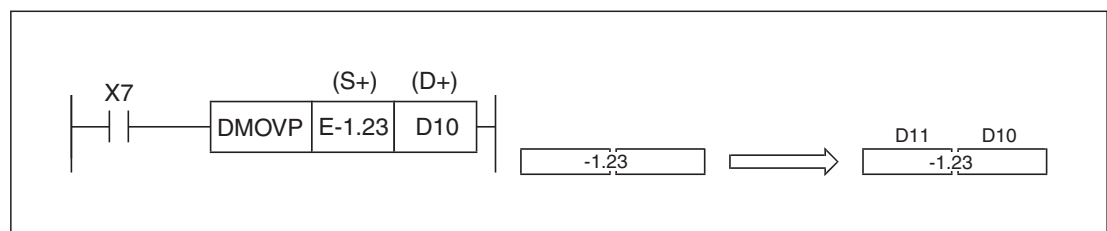

ПРИМЕР ▾

Рис. 7-56: В этом примере постоянная "-1.23" вводится в регистры данных D10 и D11 только при включении входа X7.

7.7.4 ПРЕОБРАЗОВАНИЕ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В СТРОКОВУЮ ВЕЛИЧИНУ (DESTR)

				DESTR		FNC 116		
				Преобразование числа с плавающей запятой в строковую величину				
CPU				FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D, R, специальные модули(U□/G□), константа с плавающей запятой	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули(U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули(U□/G□)		●	16бита	32бита	DESTR
						●	DESTRP	13

Функция

Преобразование числа с плавающей запятой в строковую величину (ASCII-код) с определенным количеством знаков

Описание

- Число с плавающей запятой или константа с плавающей запятой, хранящееся в (S1+) и ((S+)+1), в соответствии с данными в (S2+)+0)...(S2+)+2 преобразуется в строковую величину в кодировке ASCII и сохраняется начиная с операнда (D+).
- В (S2+)+0 указывается формат строковой величины:
(S2+)+0 = 0: десятичное представление
(S2+)+0 = 1: экспоненциальное представление
- (S2+)+1 содержит число знаков, которое должен иметь результат преобразования. Можно указать от 2 до 24 знаков.
- В (S2+)+2 вводится количество разрядов дробной части.

Десятичное представление строковой величины

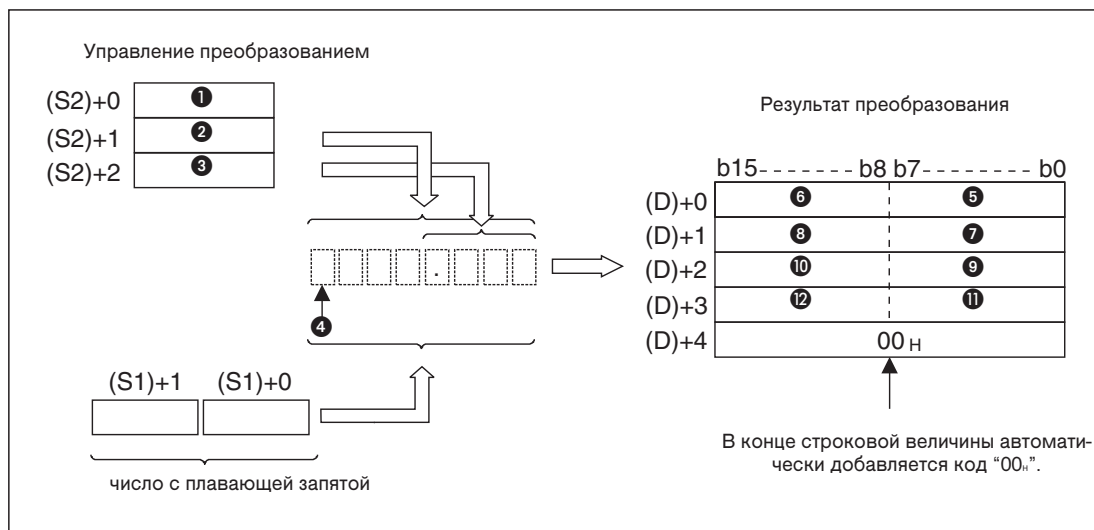


Рис. 7-57: Принцип преобразования числа с плавающей запятой в строковую величину при десятичном представлении результата

- ① Формат отображения; в случае десятичного представления регистр (S2+)+0 должен содержать значение "0".

- ② Общее количество разрядов
- ③ Количество разрядов дробной части
- ④ арифметический знак
- ⑤ ASCII-код арифметического знака
- ⑥ ASCII-код знака "общее количество разрядов - 1"
- ⑦ ASCII-код знака "общее количество разрядов - 2"
- ⑧ ASCII-код знака "общее количество разрядов - 3"
- ⑨ ASCII-код точки (2EH)
- ⑩ ASCII-код знака "общее количество разрядов - 5"
- ⑪ ASCII-код знака "общее количество разрядов - 6"
- ⑫ ASCII-код знака "общее количество разрядов - 7"

Общее количество разрядов указывается в (S2+)+1):

- Результат преобразования может содержать максимум 24 знака.
- Если количество разрядов дробной части равно нулю, количество всех разрядов = 2.
- Если количество разрядов дробной части не равно 0, количество всех разрядов равно "3 плюс количество разрядов дробной части".

(S2+)+2) содержит количество разрядов дробной части:

- Количество разрядов дробной части может находиться в диапазоне от 0 до 7.
- Общее правило: количество разрядов дробной части = (общему количеству разрядов минус 3).

Если действительное количество разрядов дробной части числа с плавающей запятой больше введенного количества разрядов дробной части конвертированного числа, лишние разряды отрезаются.

ПРИМЕР ▾

Число -1.23456 конвертируется в строковую величину, имеющее в общей сложности 8 разрядов (из них 3 в дробной части). Результат сохраняется по адресу начиная с (D+).

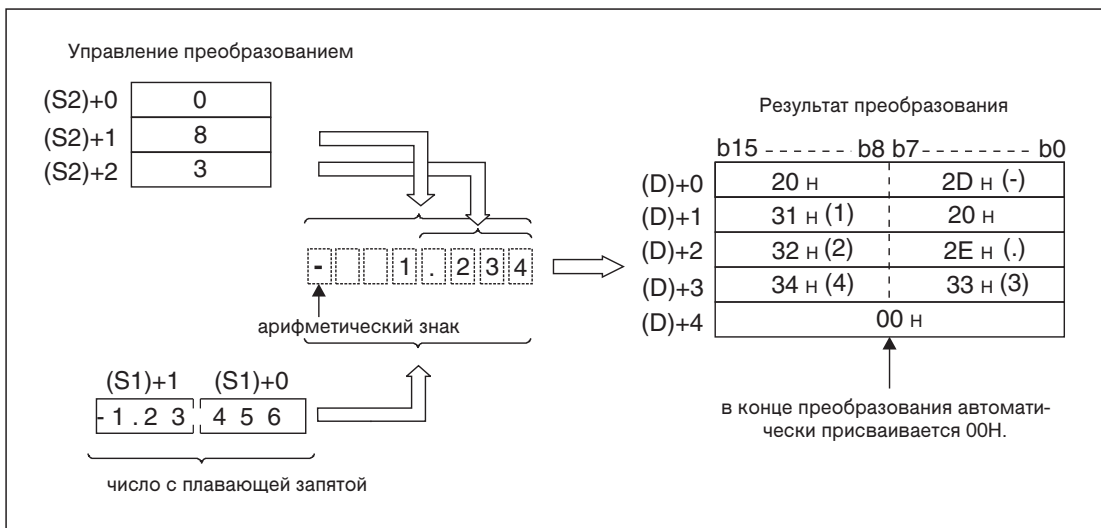


Рис. 7-58: Так как преобразуемое число с плавающей запятой имеет 5 разрядов дробной части, а строковая величина только 3 разряда дробной части, разряды утрачиваются.

В области начиная с (D+) записывается преобразованная строковая величина:

- В случае положительного числа в качестве арифметического знака в ((D+)+0) вводится код "20_h" (знак пробела). Отрицательное число помечается знаком минус (2D_h).
- Дробная часть числа с плавающей запятой, которая не может быть отображена в виде знаков ASCII, так как количество разрядов дробной части слишком мало, округляется.

ПРИМЕР ▾



Рис. 7-59: Остальная дробная часть в строковой величине не появляется, однако учитывается и округляется.

△

- Если количество разрядов дробной части в (S2+)+2 устанавливается на иное значение кроме нуля, в указанном месте автоматически вставляется ASCII-код "2E_h" (десятичная точка). Если количество разрядов дробной части устанавливается на ноль, десятичная точка не вставляется.
- Если общее количество отображаемых разрядов (включая десятичную точку и арифметический знак) меньше, чем указанное общее количество разрядов, разряды между арифметическим знаком и первым отображаемым разрядом заполняются ASCII-кодом "20_h" (знаком пробела).

ПРИМЕР ▾

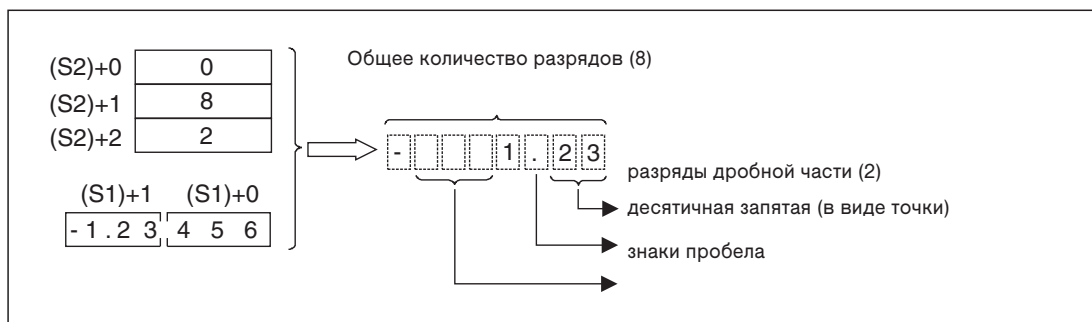


Рис. 7-60: В строковую величину автоматически вставляются десятичная запятая (в виде точки) и знаки пробела.

△

Экспоненциальное представление

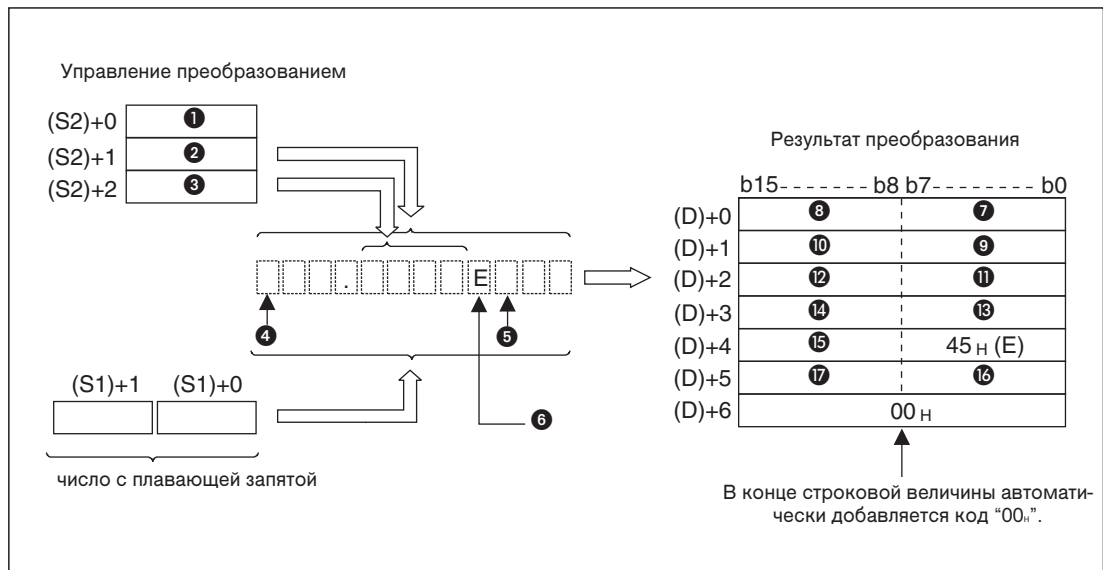


Рис. 7-61: Принцип преобразования числа с плавающей запятой в строковую величину при отображении результата в экспоненциальном виде

- 1 формат отображения; в случае экспоненциального представления регистр (S2)+0 должен содержать значение "1".
- 2 общее количество разрядов
- 3 количество разрядов десятичной дробной части
- 4 арифметический знак числа
- 5 арифметический знак показателя степени
- 6 "E" вставляется автоматически..
- 7 ASCII-код арифметического знака числа
- 8 ASCII-код знака "общее количество разрядов - 1"
- 9 ASCII-код знака "общее количество разрядов - 2"
- 10 ASCII-код точки (2EH)
- 11 ASCII-код знака "общее количество разрядов - 4"
- 12 ASCII-код знака "общее количество разрядов - 5"
- 13 ASCII-код знака "общее количество разрядов - 6"
- 14 ASCII-код знака "общее количество разрядов - 7"
- 15 ASCII-код арифметического знака показателя степени
- 16 ASCII-код знака "общее количество разрядов - 10"
- 17 ASCII-код знака "общее количество разрядов - 11"

Общее количество разрядов указывается в (S2+)+1):

- Результат преобразования может содержать максимум 24 знака.
- Если количество разрядов дробной части равно нулю, количество всех разрядов = 6.
- Если количество разрядов дробной части не равно 0, количество всех разрядов равно "7 плюс количество разрядов дробной части".

(S2+)+2) содержит количество разрядов дробной части:

- Количество разрядов дробной десятичной части может находиться в диапазоне от 0 до 7. Общее правило: количество разрядов дробной части = (общее количество разрядов минус 7).

ПРИМЕР ▾

Число -12.34567 требуется представить в экспоненциальном виде. Общее количество разрядов равно 12, из них 4 разряда дробной десятичной части. Результат сохраняется по адресу начиная с (D+).

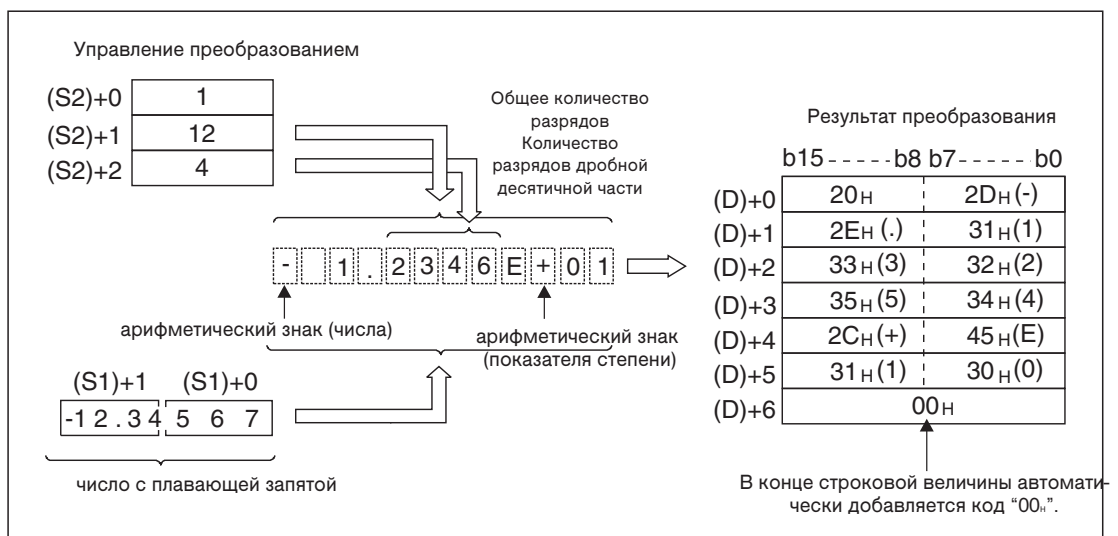


Рис. 7-62:Значение -12.34567 преобразуется в экспоненциальный вид -1.2345 x 10¹.



В области начиная с (D+) записывается преобразованная строковая величина:

- Если число положительное, в качестве арифметического знака в ((D+)+0) вводится код "20_h" (знак пробела). Отрицательное число помечается знаком минус (2D_h).
- Целая часть числа занимает только один разряд. Между арифметическим знаком и этим числом вставляется знак пробела (20_h).

ПРИМЕР ▾

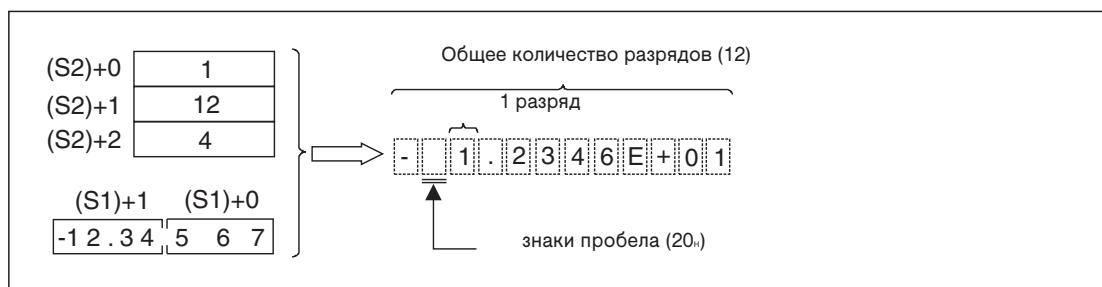


Рис. 7-63:Отображение целой части числа арифметическим знаком, знаком пробела и одним разрядом



- Если десятичная область числа с плавающей запятой длиннее, чем предусмотренная область памяти, то разряды, которые не могут сохранены, отрезаются.

ПРИМЕР ▾

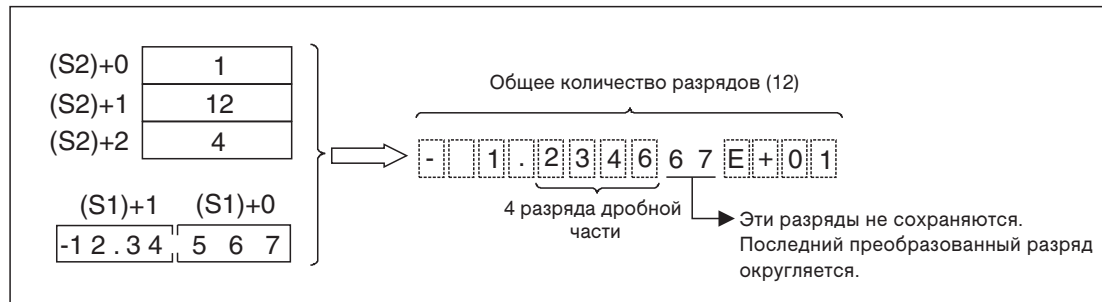


Рис. 7-64: Остальная дробная часть в строковой величине не появляется, однако учитывается и округляется

△

- Если количество разрядов дробной части в (S2+)+2) устанавливается на иное значение кроме нуля, в указанном месте автоматически вставляется ASCII-код “2E_n” (десятичная точка). Если количество разрядов дробной части устанавливается на ноль, десятичная точка не вставляется.

ПРИМЕР ▾

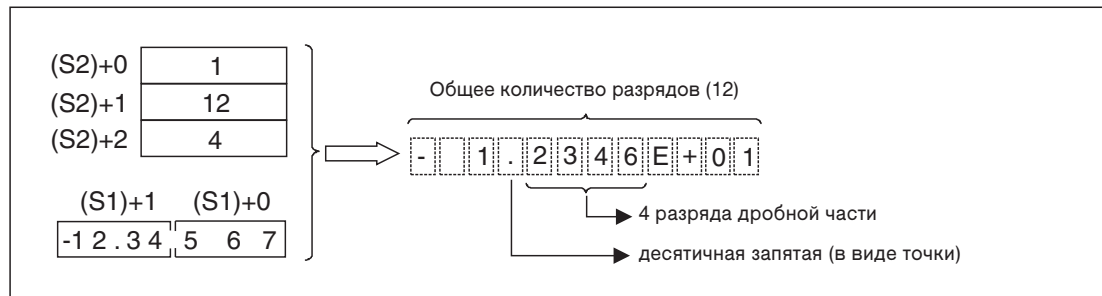


Рис. 7-65: В строковую величину автоматически вставляется десятичная запятая (в виде точки)

△

- Показатель степени отображается двумя разрядами. В случае одноразрядного показателя степени между арифметическим знаком показателя степени и показателем степени вставляется ASCII-код “30_n” (0).
- В качестве арифметического знака показателя степени вставляется ASCII-код “2B_n” (+) для положительного и ASCII-код “2D_n” (-) для отрицательного показателя степени.

ПРИМЕР ▾



Рис. 7-66: Отображение показателя степени

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Указанное в ((S1)+0) и ((S1)+1) значение не равно нулю или не находится в диапазоне $\pm 2^{-127} \leq s1 < \pm 2^{129}$.
- Содержимое ((S2)+0) не равно ни "0", ни "1".
- Общее количество разрядов в ((S2)+1) находится вне следующих диапазонов значений:

В десятичном формате

- ≥ 2 (Если в качестве количества разрядов дробной части был указан "0")
- количество всех разрядов > (количество разрядов дробной части + 3)
(Если в качестве количества разрядов дробной части было указано иное значение кроме "0")

В экспоненциальном формате

- ≥ 6 (Если в качестве количества разрядов дробной части был указан "0")
- количество всех разрядов > (количество разрядов дробной части + 7)
(Если в качестве количества разрядов дробной части было указано иное значение кроме "0")

- Количество разрядов дробной части в ((S2)+2) находится вне следующих диапазонов значений:

В десятичном формате

- количество разрядов дробной части \leq (общее количество разрядов минус 3)

В экспоненциальном формате

- количество разрядов дробной части \leq (общее количество разрядов минус 7)

- Область памяти начиная с (D+) превышает допустимую область для этих операндов.
- Результат преобразования превышает указанное общее количество знаков.

Примеры программ

При включении X0 число с плавающей запятой в регистрах файлов R1 и R0 преобразуется в строковую величину в соответствии с настройками в R10...R12 и сохраняется начиная с D0.

ПРИМЕР ▾

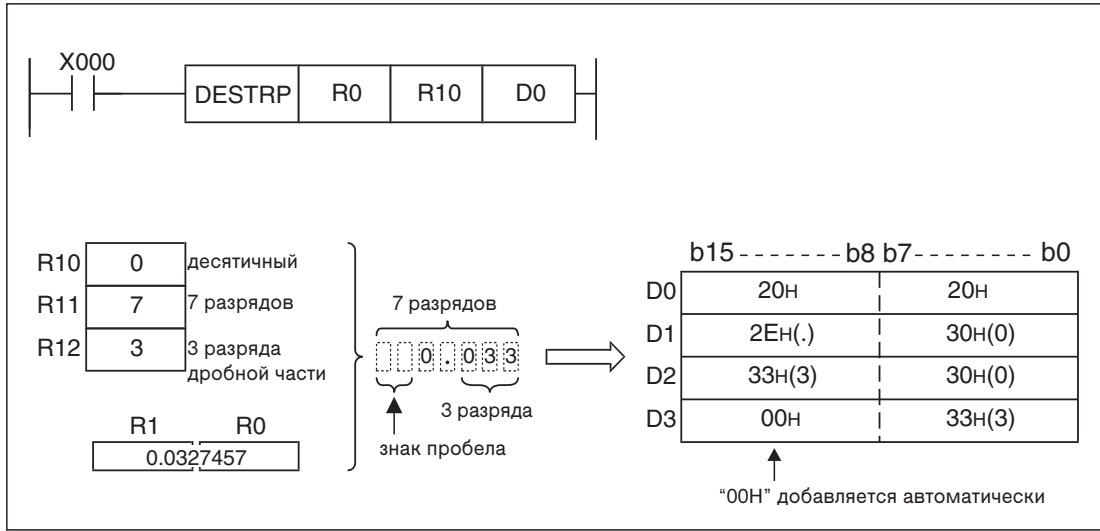


Рис. 7-67:Пример программы для преобразования в десятичное представление



ПРИМЕР ▾

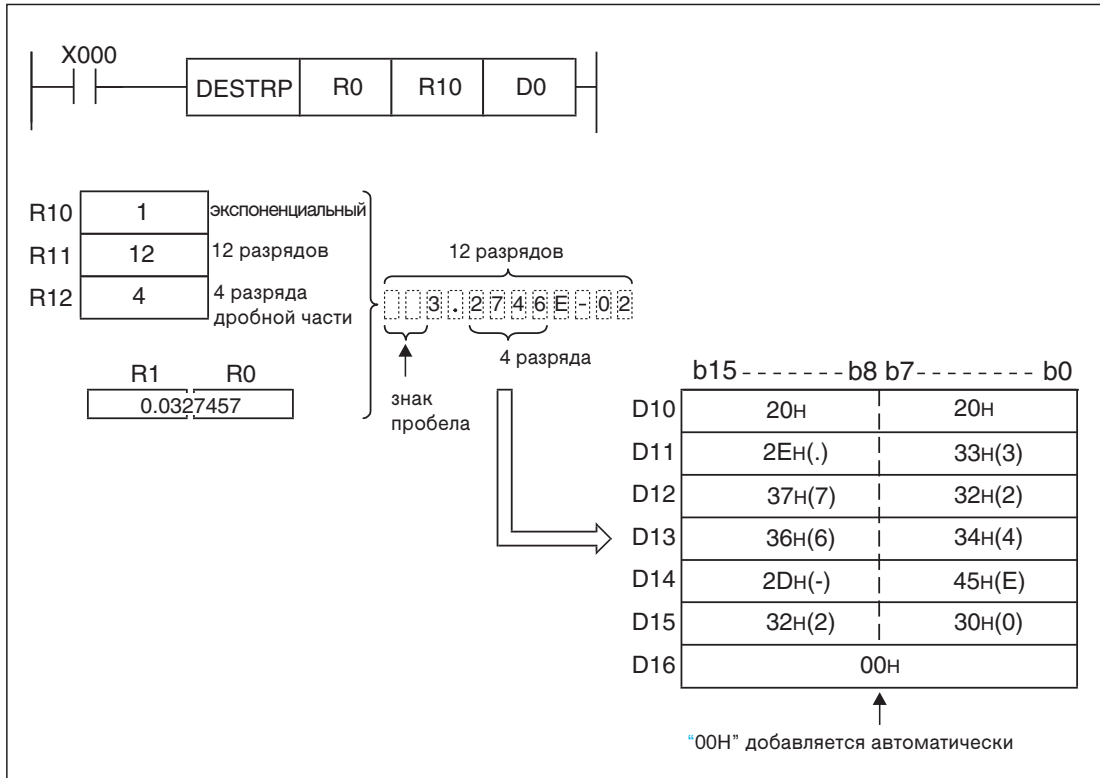



Рис. 7-68:Пример программы для преобразования в экспоненциальное представление



7.7.5 ПРЕОБРАЗОВАНИЕ СТРОКОВОЙ ВЕЛИЧИНЫ В ЧИСЛО С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEVAL)

		DEVAL		FNC 117				
		Преобразование строковой величины в число с плавающей запятой						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	S1+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	D, R специальные модули (U□/G□) Используются 2 следующих друг за другом адреса операндов.		●	16бита	32бита	DEVAL	13
						●	DEVAL	13

Функция

Преобразование строковой величины (ASCII-кода) в число с плавающей запятой

Описание

- Строковая величина в кодировке ASCII, хранящаяся начиная с (S+), преобразуется в число с плавающей запятой и сохраняется в (D+) и ((D+)+1).
- Конвертируемую строковую величину можно преобразовать в десятичный формат с плавающей запятой или в экспоненциальный формат.

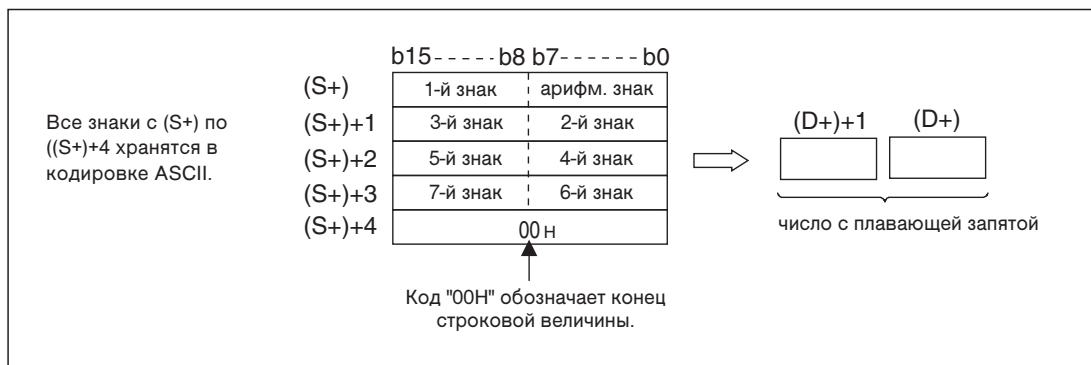


Рис. 7-69: Принцип преобразования строковой величины в число с плавающей запятой с помощью команды DVAL

Десятичное представление

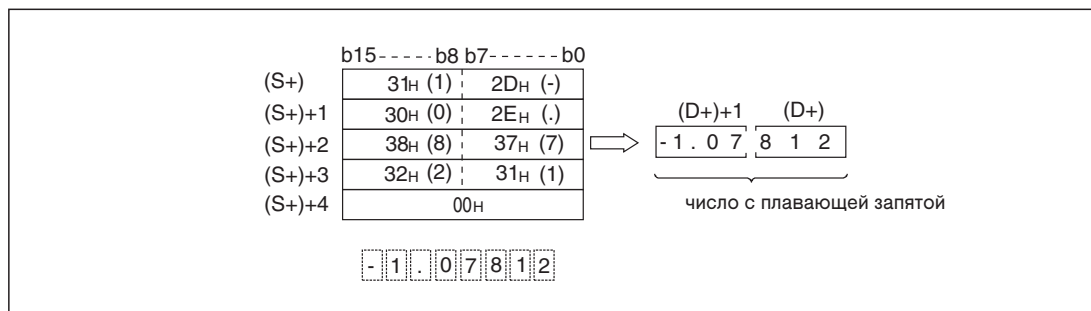


Рис. 7-70: Преобразование строковой величины, содержащей число с плавающей запятой в десятичном формате

Экспоненциальное представление

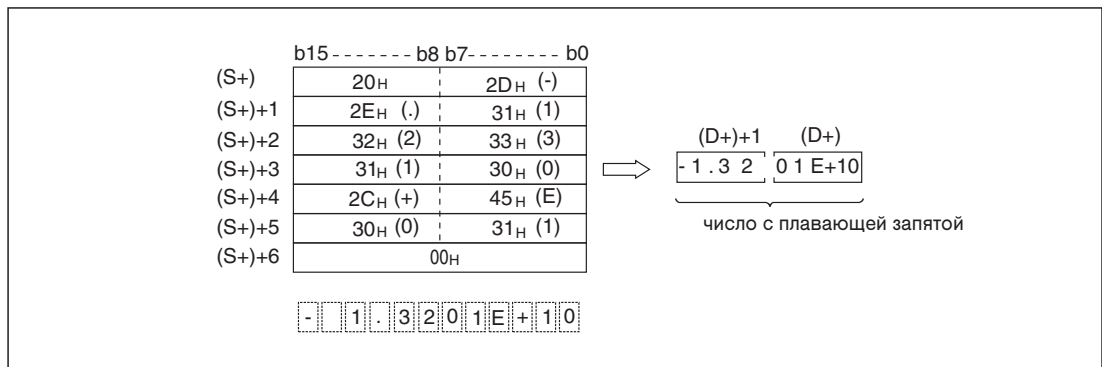


Рис. 7-71: Преобразование строковой величины, содержащей число с плавающей запятой в экспоненциальном виде

Конвертируются 6 разрядов (без арифметического знака, десятичной запятой и разрядов показателя степени результата) строковой величины, хранящейся начиная с (S+), в десятичное число с плавающей запятой. Начиная с 7-го разряда дальнейшие разряды результата отрезаются. В связи с этим могут возникнуть ошибки округления.

ПРИМЕР ▾

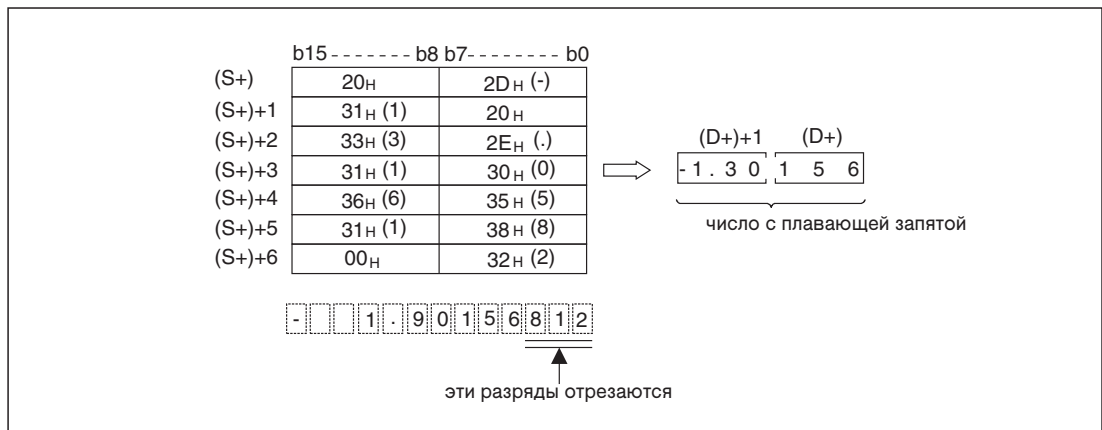


Рис. 7-72: После 6-го разряда все дальнейшие разряды более не преобразуются(десятичное представление)



ПРИМЕР ▾

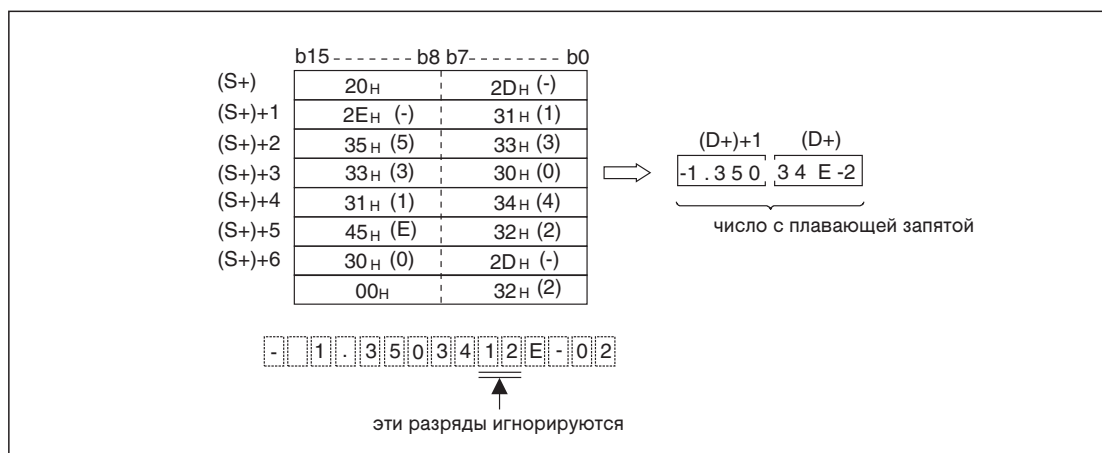


Рис. 7-73: При экспоненциальном представлении преобразуются 6 разрядов целой и десятичной части.



Если в строковой величине начиная с (S+) перед первой отображаемой цифрой установлен ASCII-код "20н" (знак пробела) или "30н" (ноль), то при конвертации эти знаки игнорируются.

ПРИМЕР ▾

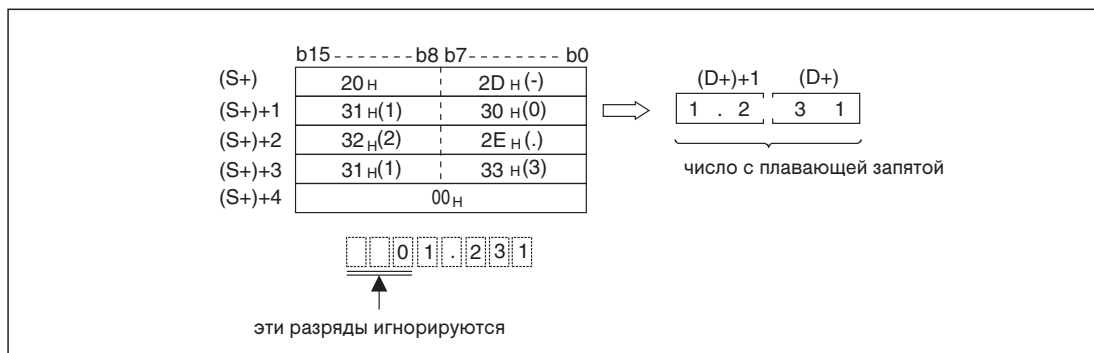


Рис. 7-74: Предшествующие нули и знаки пробела не преобразуются.



Если между знаком "E" и строковой величиной для экспоненциальной части установлен ASCII-код "30н" (ноль), то при конвертации этот знак игнорируется.

ПРИМЕР ▾

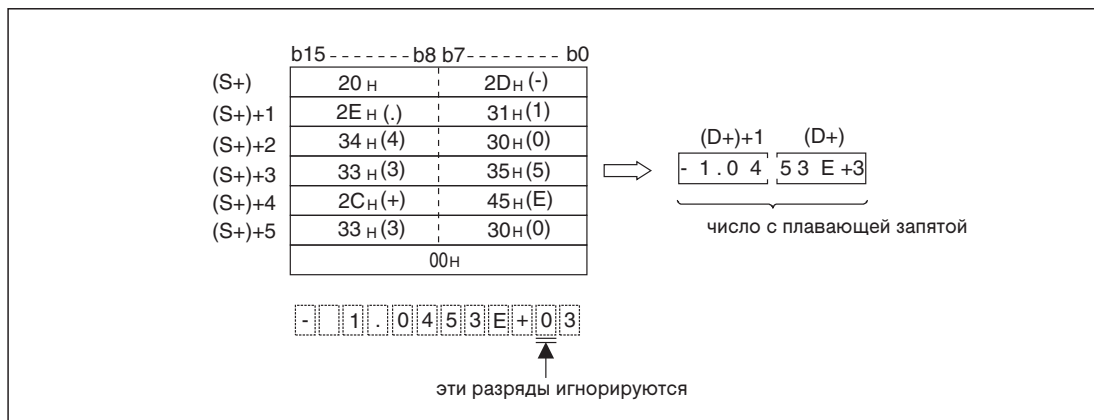


Рис. 7-75: Предшествующий ноль в показатель степени не преобразуется.



Конвертируемая строковая величина может состоять максимум из 24 знаков.

Команда DEVAL влияет на следующие специальные маркеры:

● M8020

Если результат преобразования равен "0", этот специальный маркер имеет состояние "1".

● M8021

Если результат преобразования меньше 2^{-126} , устанавливается маркер M8021 ("1"). В этом случае в качестве результата преобразования выводится минимальное значение (2^{-126}).

● M8022

– M8022 устанавливается, если результат преобразования больше 2^{128} . В этом случае результат преобразования соответствует максимальному значению (2^{128}).

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Разряды перед запятой или дробная часть содержат иные знаки кроме ASCII-кодов цифр 0...9. ("30н"... "39н").
- Знак "2Ен" (".") в строковой величине использован более одного раза.
- В экспоненциальной части использованы иные знаки кроме "45н" (Е), "2Сн" (+) или "2Dн" (-), или имеется более одного показателя степени.
- В области операндов начиная с (S+) не имеется метки конца "00н".
- Число знаков в строковой величине равно нулю или больше 24.

ПРИМЕР ▾

Примеры программ

При включении X0 строковая величина, хранящаяся начиная с регистра файлов R0, преобразуется в число с плавающей запятой и сохраняется в D1 и D0.

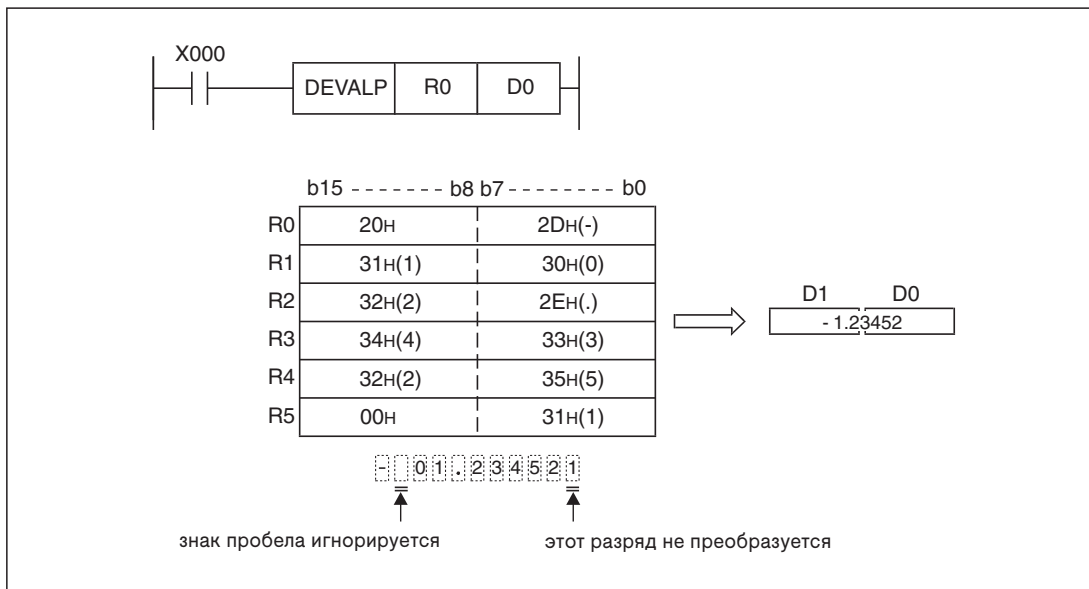


Рис. 7-76:Преобразование строковой величины в десятичное число с плавающей запятой



ПРИМЕР ▾

При включении X0 строковая величина, хранящаяся начиная с регистра данных D10, преобразуется в число с плавающей запятой и сохраняется в D101 и D100.

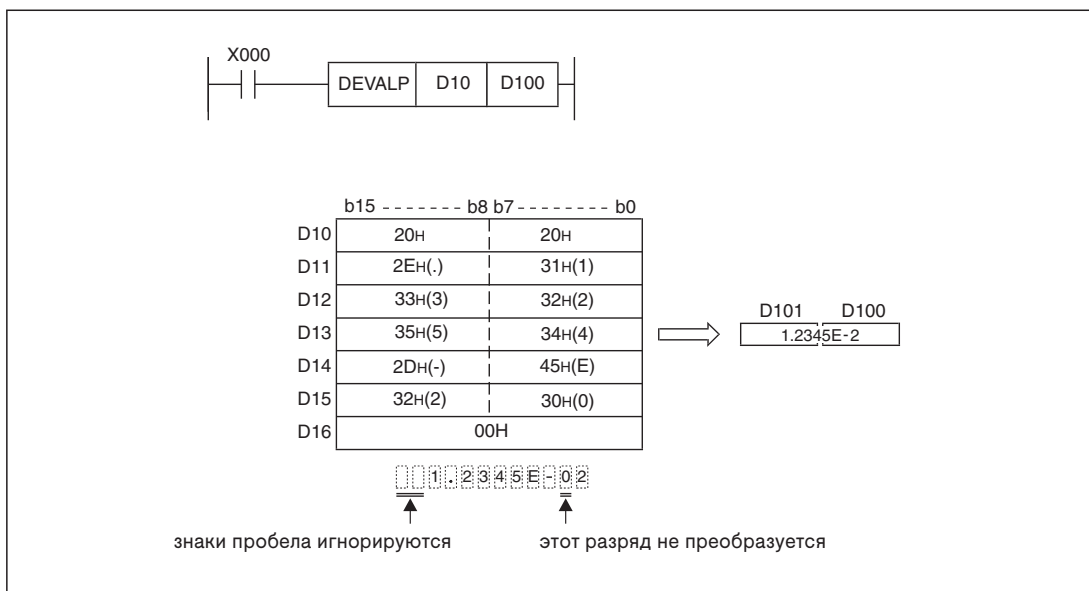


Рис. 7-77:Преобразование строковой величины в число с плавающей запятой в экспоненциальном виде

7.7.6 ПРЕОБРАЗОВАНИЕ ФОРМАТА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В НАУЧНЫЙ ФОРМАТ ЧИСЕЛ (DEBCD)

		DEBCD		FNC 118				
		Преобразование формата с плавающей запятой в научный формат чисел						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	
Операнды	S+	D+	Имп. инструкция(P)		Обработка		Шаги программы	
	D (числа с плавающей запятой (32 бита))	D Применяется два один за другим следующих адреса операндов	●	16бита	32бита	DEBCD	9	
				●	DEBCDP	9		

ФУНКЦИЯ

Преобразование числа в формате с плавающей запятой в научный формат чисел.

ОПИСАНИЕ

- Число, заданное в формате с плавающей запятой, в (S+) конвертируется и запоминается в (D+).
- Мантисса запоминается в (D+).
- Экспонента запоминается в ((D+)+1).

УКАЗАНИЕ

Чтобы представить результат конвертирования с максимальной точностью, мантисса (D+) указывается 0 или значением между 1000 и 9999. Указание экспоненты ((D+)+1) соответствующим образом корректируется (например, $3,4567 \times 10^{-5}$ (S+, (S+)+1) конвертируется и запоминается как 3456 (D+) и -8 ((D+)+1))

ПРИМЕР ▾

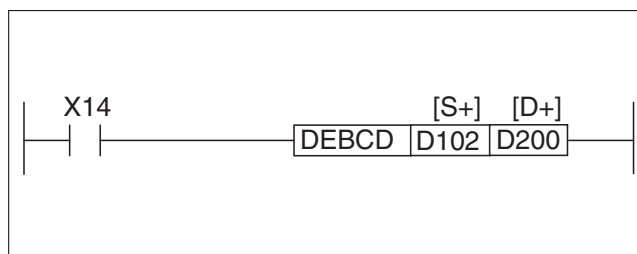


Рис. 7.78:

Пример программирования DEBCD-инструкция

C000352C

При включении входа X14 числа в формате с плавающей запятой, записанные в D102 и D103, конвертируются в научный формат и записываются в D200.

Мантисса запоминается в D200.

Экспонента запоминается в D201.

△

7.7.7 ПРЕОБРАЗОВАНИЕ ИЗ НАУЧНОГО ФОРМАТА ЧИСЕЛ В ФОРМАТ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEBIN)

		DEBIN		FNC 119				
		(S+)	(D+)	Преобразование научного формата чисел в формат с плавающей запятой				
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	D Применяется два один за другим следующих адреса операндов	D (числа с плавающей запятой (32 бита))	●	16бита	32бита	DEBIN	9	
					●	DEBINP	9	

ФУНКЦИЯ

Преобразование числа в научном формате в число в формате с плавающей запятой

ОПИСАНИЕ

- Число, заданное в научном формате (S+), конвертируется в формат с плавающей запятой и запоминается в (D+).
- Мантисса запоминается в (S+).
- Экспонента запоминается в ((S+)+1).

УКАЗАНИЕ

Чтобы представить результат конвертирования с максимальной точностью, мантисса (S+) указывается 0 или значением между 1000 и 9999. Указание экспоненты ((S+)+1) соответствующим образом корректируется (например, значения 5432 (S+) и 12 ((S+)+1), указанные для мантиссы и экспоненты, конвертируется в число $5,432 \times 10^9$ (D+), (D+)+1) в формате с плавающей запятой.

ПРИМЕР ▾

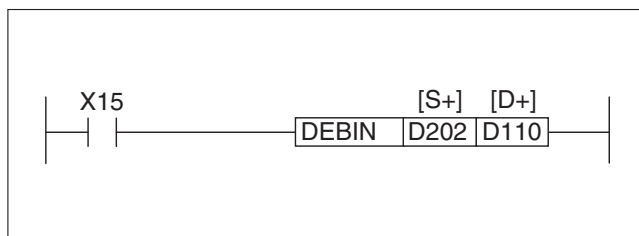


Рис. 7-79:

Пример программирования DEBIN-инструкции

C000353C

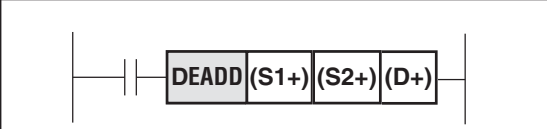
При включении входа X15 числа, записанные в D202 и D203, в научном формате чисел конвертируются в формат с плавающей запятой и записываются в D110.

Мантисса указывается в D202.

Экспонента указывается в D203.

△

7.7.8 СЛОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEADD)

				DEADD		FNC 120		
				Сложение чисел с плавающей запятой				
CPU				FX1S	FX1N	FX2N	FX2NC	FX3U
						●	●	●
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	К, Н целые значения автоматически конвертируются в числа с плавающей запятой. D (Числа с плавающей запятой (32 бита))		D (числа с плавающей запятой (32 бита))		●	16бита	32бита	DEADD
						●	DEADDP	13

ФУНКЦИЯ

Сложение двух чисел с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Число с плавающей запятой, заданное в (S1+), суммируется с числом с плавающей запятой в (S2+). Результат запоминается в (D+).
- Для каждого операнда используется по 2 следующих друг за другом регистра.
- Задаваемые константы (К, Н) перед сложением автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.
- Если результат сложения равен 0, то включается флаг zero (нуля) M8020.
- Если результат сложения больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M8022.
- Если результат сложения меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M8021.

УКАЗАНИЕ

Сложение выполняется по требуемым математическим закономерностям (например, сложение $2,3456 \times 10^2 + (-5,6 \times 10^{-1})$ даст результат $2,34 \times 10^2$.)

ПРИМЕР ▾

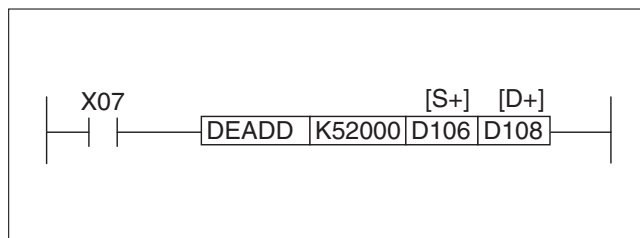


Рис. 7-80:

Пример программирования DEADD-инструкции

C000354C

При включении входа X7 к константе K52000 прибавится число с плавающей запятой, записанное в D106.

Результат сохранится в D108.



7.7.9 ВЫЧИТАНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DESUB)

				DESUB		FNC 121			
				Вычитание чисел с плавающей запятой					
Операнды				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
				S1+	S2+	D+	Имп. инструкция(P)	Обработка	Шаги программы
К, Н целые значения автоматически конвертируются в числа с плава- ющей запятой.		D (числа с плава- ющей запятой (32 бита))	●	16бита	32бита	DESUB	13		
D (числа с плавающей запятой (32 бита))					●	DESUBP	13		

ФУНКЦИЯ

Вычитание двух чисел с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Число с плавающей запятой, заданное в (S2+), вычитается из числа с плавающей запятой в (S1+). Результат запоминается в (D+).
- Для каждого операнда используется по 2 следующих друг за другом регистра.
- Задаваемые константы (К, Н) перед вычитанием автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.
- Если результат вычитания равен 0, то включается флаг zero (нуля) M8020.
- Если результат сложения больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M8022.
- Если результат сложения меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M8021.

УКАЗАНИЕ

Вычитание выполняется по требуемым математическим закономерностям (например, вычитание $2,3456 \times 10^2 - 5,6 \times 10^{-1}$ даст результат $2,34 \times 10^2$.)

ПРИМЕР ▾

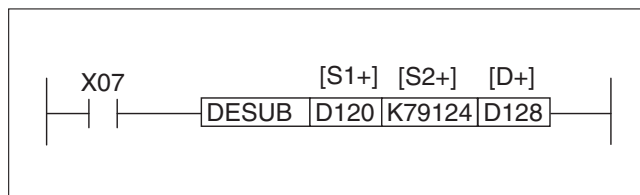


Рис. 7-81:

Пример программирования
DESUB-инструкции

C000355C

При включении входа X07 из числа с плавающей запятой D120 вычитается константа K79124.

Результат сохранится в D128.



7.7.10 УМНОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEMUL)

				DEMUL		FNC 122			
				Умножение чисел с плавающей запятой					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●	●	●
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	К, Н целые значения автоматически конвертируются в числа с плава- ющей запятой D (числа с плавающей запятой (32 бита))		D (числа с плава- ющей запятой (32 бита))		●	16бита	32бита	DEMUL	13
						●	DEMULP	13	

ФУНКЦИЯ

Умножение двух чисел с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Число с плавающей запятой, заданное в (S1+), умножается на число с плавающей запятой в (S2+). Результат запоминается в (D+).
- Для каждого операнда используется по 2 следующих друг за другом регистра.
- Задаваемые константы (К,Н) перед умножением автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.

УКАЗАНИЕ

Умножение выполняется по требуемым математическим закономерностям.

ПРИМЕР ▾

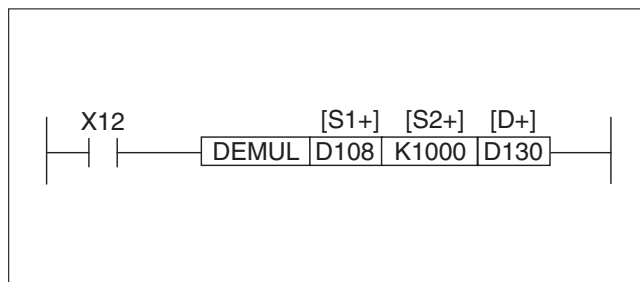


Рис. 7-82:

Пример программирования DEMUL-инструкции

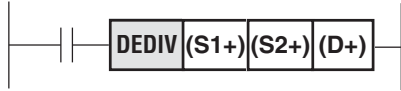
C000356C

При включении меркера M12 число с плавающей запятой в D108 перемножается с константой K1000.

Результат сохранится в D130.

△

7.7.11 ДЕЛЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEDIV)

				DEDIV		FNC 123		
				Деление чисел с плавающей запятой				
CPU				FX1S	FX1N	FX2N	FX2NC	FX3U
						●	●	●
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	К, Н целые значения автоматически конвертируются в числа с плава- ющей запятой		D (числа с плава- ющей запятой (32 бита))		●	16бита	32бита	DEDIV
	D (числа с плавающей запятой (32 бита))					●	DEDIVP	13

ФУНКЦИЯ

Деление двух чисел с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Число с плавающей запятой, заданное в (S1+), делится на число с плавающей запятой в (S2+). Результат запоминается в (D+).
- Для каждого операнда используется по 2 следующих друг за другом регистра.
- Задаваемые константы (К, Н) перед делением автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.

УКАЗАНИЕ

Деление выполняется по требуемым математическим закономерностям.

ИСТОЧНИК ОШИБКИ

Операнд (S2+) не может быть равен нулю т.к. Деление на нуль не допустимо.

ПРИМЕР ▾

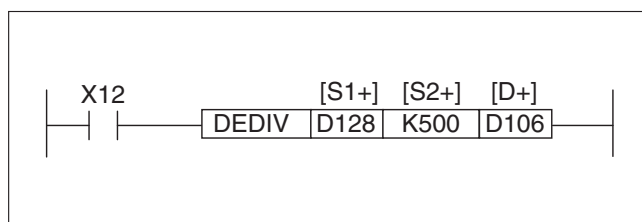


Рис. 7-83:

Пример программирования DE-DIV-инструкции

C000357C

При включении входа X10 число с плавающей запятой в D128 делится на константу K500.

Результат сохранится в D106.

△

7.7.12 ПРЕДСТАВЛЕНИЕ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В ВИДЕ ПОКАЗАТЕЛЯ СТЕПЕНИ ПО ОСНОВАНИЮ E (DEXP)

		DEXP		FNC 124			
		Корень квадратный из числа с плавающей запятой					
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U	
						●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	К, Н Целые значения автоматически конвертируются в числа с плавающей запятой D (числа с плавающей запятой (32 бита))	D (числа с плавающей запятой (32 бита))	●	16бита	32бита	DEXP	9
	Используются 2 следующих друг за другом адреса операндов..				●	DEXPP	9

Функция

Число с плавающей запятой интерпретируется в виде показателя степени по основанию e ($e = 2,71828$).

Описание

- Команда DEXP преобразует число с плавающей запятой, хранящееся по адресу $((S+)+1)$ и $(S+)$, в показатель степени по основанию e и сохраняет результат по адресу $((D+)+1)$ и $(D+)$.

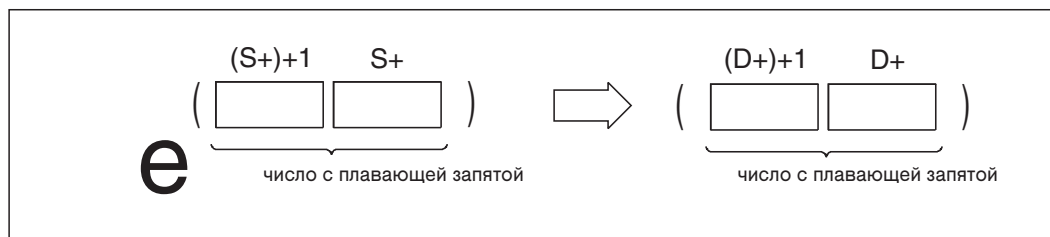


Рис. 7-84: Функция команды DEXP

- В $((S+)+1)$ и $(S+)$ можно также указать константу с плавающей запятой.

Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Результат вычисления не находится в диапазоне от 2-126 до 2128.

ПРИМЕР ▾

Следующая программа вычисляет экспоненциальную функцию по основанию e из двузначного двоично-десятичного значения по адресу с X20 по X27 и сохраняет результат в виде числа с плавающей запятой по адресу D0 и D1.

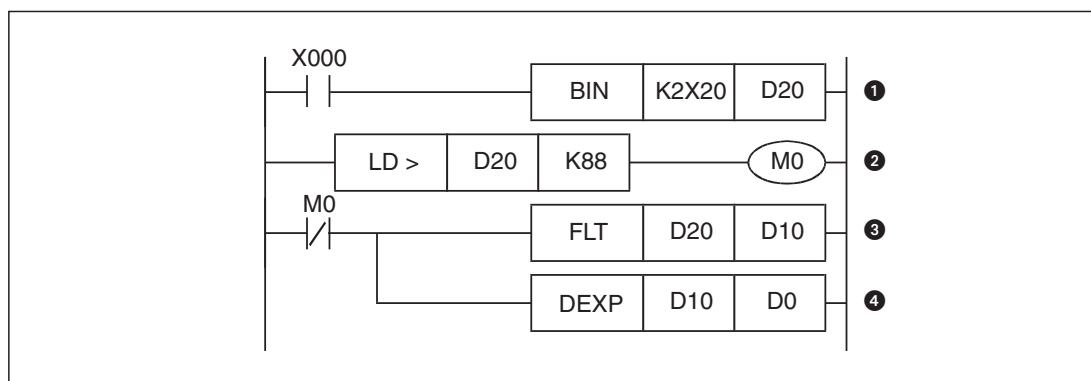


Рис. 7-85: Пример программы для применения команды DEXP

- ① Число в двоично-десятичном представлении на входах с X20 по X27 (например, введенное с помощью двузначного двоично-десятичного выключателя), преобразуется в двоичный формат.
- ② Введенное значение проверяется. Результат команды DEXP меньше 2128, если число в двоично-десятичном представлении меньше или равно "88". ($\ln 2128 = 88,7$). Если введено большее значение, чем 88, устанавливается M0 и команда DEXP не выполняется.
- ③ Введенное значение преобразуется в число с плавающей запятой.
- ④ Команда DEXP выполняется и введенное значение используется в виде показателя степени по основанию e .

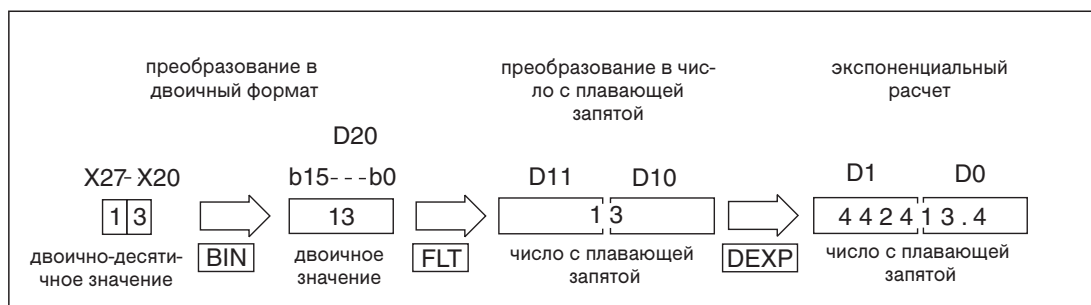



Рис. 7-86: в вышеприведенном примере с помощью двоично-десятичных выключателей введено значение "13".



7.7.13 ВЫЧИСЛЕНИЕ НАТУРАЛЬНОГО ЛОГАРИФМА (DLOGE)

		DLOGE		FNC 125				
		Вычисление натурального логарифма						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U□/G□) константа с плавающей запятой	D, R, специальные модули (U□/G□)		●	16бита	32бита	DLOGE	9
	Используются 2 следующих друг за другом адреса операндов.					●	DLOGEP	9

Функция

Вычисление натурального логарифма (логарифма с основанием e, сокращенно “ln”) числа с плавающей запятой.

Описание

- Команда DLOGE вычисляет натуральный логарифм числа с плавающей запятой, хранящегося по адресу ((S+)+1) и (S+), и сохраняет результат по адресу ((D+)+1) и (D+).

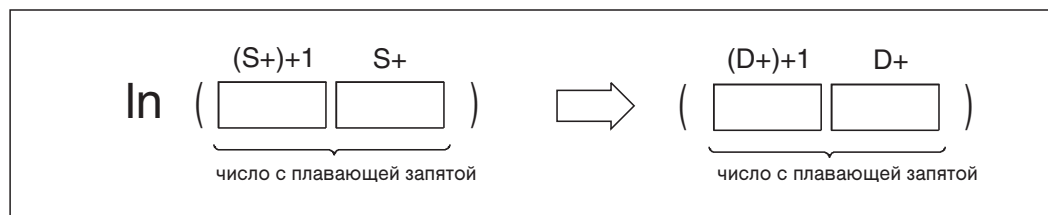


Рис. 7-87: Принцип действия команды DLOGE

- В ((S+)+1) и (S+) можно вводить только положительные значения. Вычисление натурального логарифма отрицательного значения не возможно.

УКАЗАНИЕ

Взаимосвязь между натуральным и десятичным логарифмом:

Для пересчета десятичного логарифма (логарифма с основанием 10) можно воспользоваться следующей формулой:

$$10^x = e^{\frac{x}{0,4342945}}$$

С помощью команды DLOG10 (см. раздел 7.7.14) десятичный логарифм числа с плавающей запятой можно вычислить непосредственно.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки “6706”:

- ((S+)+1) и (S+) содержит отрицательное значение.
- ((S+)+1) и (S+) содержит значение “0”.

ПРИМЕР ▾

Если включен вход X0, следующая программа вычисляет натуральный логарифм числа "10" и сохраняет результат по адресу D30 и D31.

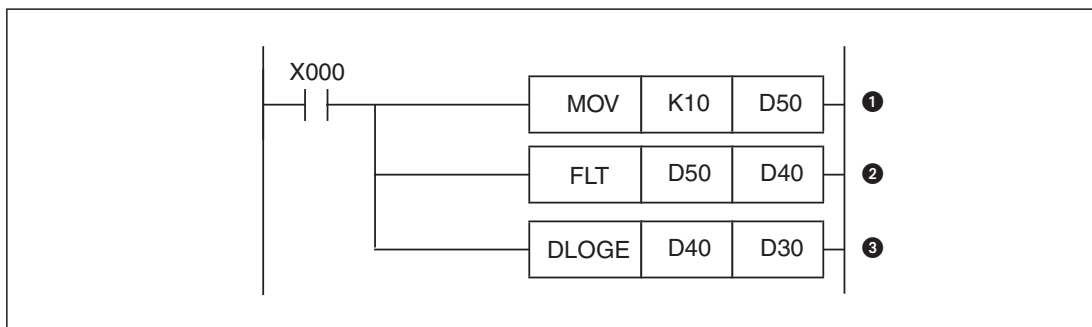


Рис. 7-88:Пример программы для применения команды DLOGE

- ❶ Константа "10" передается в регистр данных D50.
- ❷ Константа преобразуется в число с плавающей запятой.
- ❸ Вычисление натурального логарифма константы.

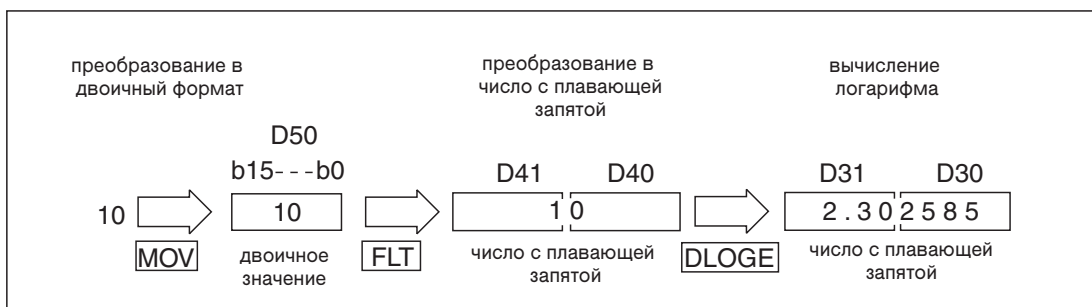


Рис. 7-89:Обработка данных в этом примере программы

△

7.7.14 ВЫЧИСЛЕНИЕ ДЕСЯТИЧНОГО ЛОГАРИФМА (DLOG10)

		DLOG10		FNC 126				
		Вычисление десятичного логарифма						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U□/G□) константа с плавающей запятой	D, R, специальные модули (U□/G□)		●	16бита	32бита	DLOG10	9
	Используются 2 следующих друг за другом адреса операндов.					●	DLOG10P	9

Функция

Вычисление десятичного логарифма (логарифма с основанием 10, сокращенно “lg”) числа с плавающей запятой.

Описание

- Команда DLOG10 вычисляет десятичный логарифм числа с плавающей запятой, хранящегося по адресу ((S+)+1) и (S+), и сохраняет результат по адресу ((D+)+1) и (D+).

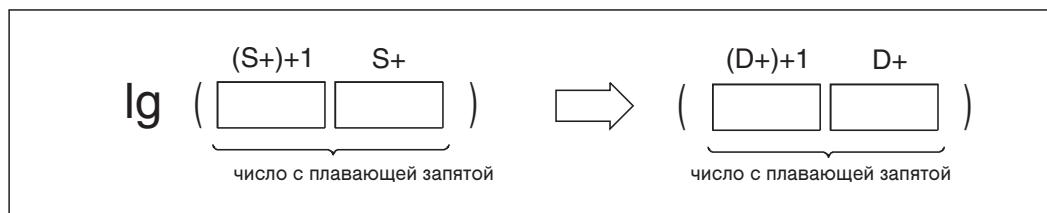


Рис. 7-90: Принцип действия команды DLOG10

- В ((S+)+1) и (S+) можно вводить только положительные значения. Вычисление десятичного логарифма отрицательного значения не возможно.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067, и в специальный регистр D8067 записывается код ошибки “6706”:

- ((S+)+1) и (S+) содержит отрицательное значение.
- ((S+)+1) и (S+) содержит значение “0”.

ПРИМЕР ▾

При включенном входе X0 следующая программа вычисляет десятичный логарифм числа "15" и сохраняет результат по адресу D30 и D31.

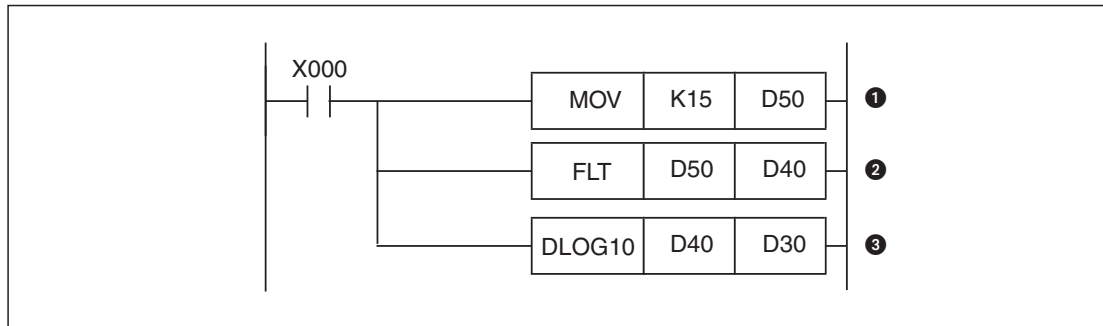


Рис. 7-91:Пример программы для применения команды DLOGE

- ① Константа "15" передается в регистр данных D50.
- ② Константа преобразуется в число с плавающей запятой.
- ③ Вычисление десятичного логарифма константы.

7.7.15 КОРЕНЬ КВАДРАТНЫЙ ИЗ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DESQR)

		DESQR		FNC 127			
		(S+)	(D+)	Корень квадратный из числа с плавающей запятой			
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	К, Н Целые значения автоматически конвертируются в числа с плавающей запятой D (числа с плавающей запятой (32 бита))	D (числа с плавающей запятой (32 бита))		16бита	32бита	DESQR	9
			●	●	DESQRP	9	

ФУНКЦИЯ

Вычисление корня квадратного из числа с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Из числа с плавающей запятой, заданного в (S+), вычисляется корень квадратный.
- Для каждого операнда используется по 2 следующих друг за другом регистра.
- Задаваемые константы (К,Н) перед делением автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.
- Если результат вычисления равен 0, то включается флаг zero (нуля) M8020.

УКАЗАНИЕ

Вычисление корня квадратного выполняется по требуемым математическим закономерностям

ИСТОЧНИК ОШИБКИ

Если операнд (S+) отрицательное число, то активизируется меркер ошибки M8067.

ПРИМЕР ▾

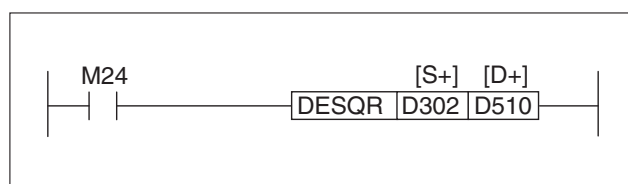


Рис. 7-92:

Пример программирования DESQR-инструкции

C000358C

При включении меркера M24 вычисляется корень квадратный из числа с плавающей запятой в D302.

Результат сохранится в D510.

△

7.7.16 ИЗМЕНЕНИЕ АРИФМЕТИЧЕСКОГО ЗНАКА ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DENEГ)

		DENEГ		FNC 128			
		Преобразование формата с плавающей запятой в десятичный формат					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Операнды	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U□/G□) Используются 2 следующих друг за другом адреса операндов	●	16бита	32бита	DENEГ	5	
				●	DENEГP	5	

Функция

Изменение арифметического знака числа с плавающей запятой на противоположный

Описание

- Изменяется арифметический знак числа с плавающей запятой, хранящегося по адресу ((D+)+1) и (D+). После этого число с инвертированным арифметическим знаком снова сохраняется по адресу ((D+)+1) и (D+).

УКАЗАНИЕ

Если команда DENEГ выполняется циклически, арифметический знак числа с плавающей запятой изменяется в каждом программном цикле. Чтобы изменить арифметический знак в определенный момент времени, используйте вариант команды, управляемый фронтом сигнала (DENEГP).

ПРИМЕР ▾

При включении входа X0 арифметический знак числа с плавающей запятой, хранящегося по адресу D101 и D100, изменяется на противоположный и результат снова вводится в регистры D101 и D100.

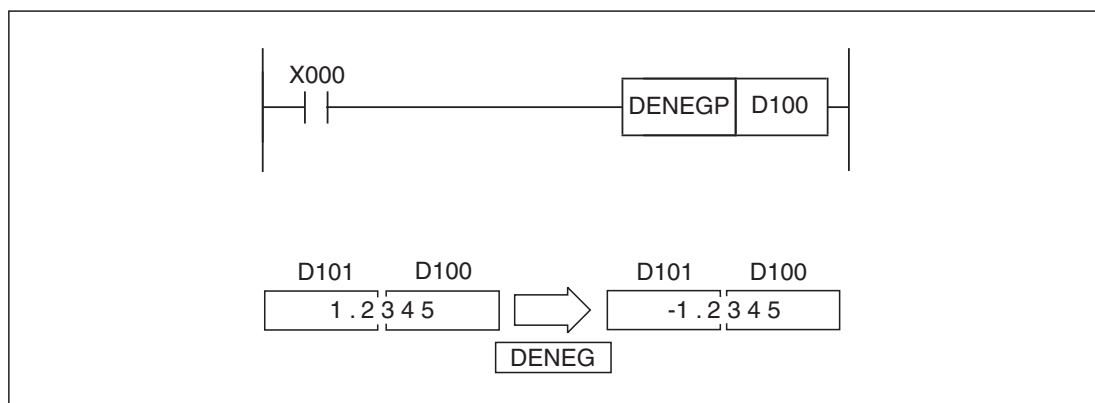
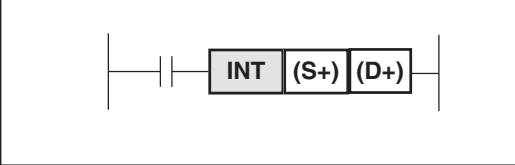


Рис. 7-93: Пример программы для инвертирования арифметического знака с помощью команды DENEГ

△

7.7.17 ПРЕОБРАЗОВАНИЕ ФОРМАТА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В ЦЕЛОЕ ЧИСЛО (INT)

			INT		FNC 129		
			Преобразование формата с плавающей запятой в десятичный формат				
CPU			FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D (числа с плавающей запятой (32 бита))	D (десятичный формат) INT, INTP (16 бит), DINT, DINTP (32 бита)		16бита	32бита	INT, INTP	5
			●	●	●	DINT, DINTP	9

ФУНКЦИЯ

Конвертирование числа с плавающей запятой в десятичный формат

ОПИСАНИЕ

- Число с плавающей запятой, заданное в (S+), округляется до ближайшего меньшего целого значения и запоминается в (D+).
- Операнд-источник всегда является операндом двойного слова.
- При применении INT-инструкции словный операнд является операндом цели.
- При применении DINT-инструкции операнд цели является операндом двойного слова.
- INT-инструкция является обратной функцией FLT-инструкции.
- Если результат конвертирования является 0, то включается флаг нуля (зеро) M8020.
- Если в (S+) указывается не целое число, то это число округляется до ближайшего меньшего целого значения и включается флаг заимствования (borrow) M8021.
- Если конвертируемое целое значение находится вне области памяти операнда цели, появляется превышение и включается флаг переноса (carry) M8022.

УКАЗАНИЕ

При появлении превышения возникает ошибочный результат в операнде цели.

ПРИМЕР ▾

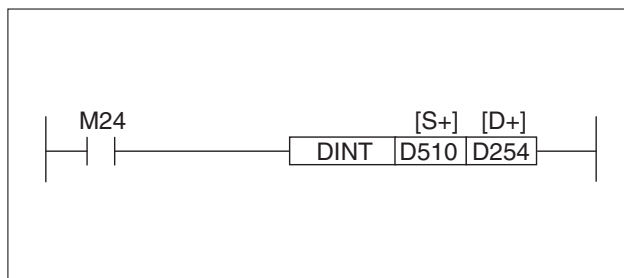


Рис. 7-94:

Пример программирования DINT-инструкции

C000359C

При включении меркера M25 число с плавающей запятой в D510 и D511 округляется до ближайшего меньшего целого значения и включается флаг заимствования (borrow) M8021.

Результат сохранится в D254 и D255.

△

7.7.18 РАСЧЕТ СИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DSIN)

		DSIN		FNC 130			
		Расчет синуса числа с плавающей запятой					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D (числа с плавающей запятой в радианах (32 бита))	D (число с плавающей запятой (32 бита))	●	16бита	32бита	DSIN	9
					●	DSINP	9

ФУНКЦИЯ

Расчет синуса числа с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Рассчитывается синус числа с плавающей запятой, заданного в (S+). Результат запоминается в (D+).
- Для каждого операнда применяется соответственно 2 следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.
- Значение угла, указанное начиная с адреса (S+), должно быть в диапазоне между 0 и 360° (0 и 2π рад). Значение угла указывается в радианах (градусов × π/180 [рад]). Для пересчета между градусами и радианами можно использовать команды DRAD и DDEG (см. разделы 7.7.24 и 7.7.25).

ПРИМЕР ▾

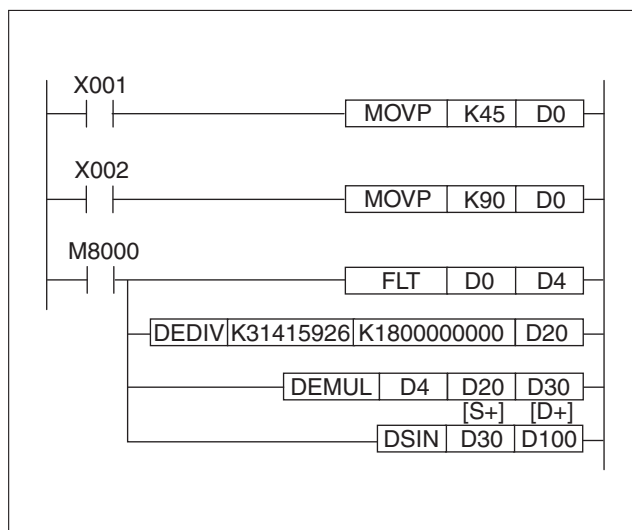


Рис. 7-95:

Пример программирования DSIN-инструкции с преобразованием в радианы

C000360C

По положительному фронту (P) входа X1 константа K45 (45 градусов) записывается в D0. По положительному фронту (P) входа X2 константа K90 (90 градусов) записывается в D0. При включении меркера M8000 значения в D0 конвертируются в число с плавающей запятой и сохраняются в D4 и D5. С помощью DEDIV- и DEMUL-инструкций происходит пересчет этих значений в радианы. Результат записывается в D30 и D31. С помощью DSIN-инструкции происходит расчет синуса. Результат записывается в D100 и D101.



7.7.19 РАСЧЕТ КОСИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DCOS)

		DCOS		FNC 131			
		Расчет косинуса числа с плавающей запятой					
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U	
				●	●	●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D (числа с плавающей запятой в радианх (32 бита))	D (число с плавающей запятой (32 бита))	●	16бита	32бита	DCOS	9
					●	DCOSP	9

ФУНКЦИЯ

Рассчитывается косинус числа с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Рассчитывается косинус числа с плавающей запятой, заданного в (S+). Результат запоминается в (D+).
- Для каждого операнда применяется соответственно 2 следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.
- Значение угла, указанное начиная с адреса (S+), должно быть в диапазоне между 0 и 360° (0 и 2π рад). Значение угла указывается в радианах (градусов × π/180 [рад]). Для пересчета между градусами и радианами можно использовать команды DRAD и DDEG (см. разделы 7.7.24 и 7.7.25).

ПРИМЕР ▾

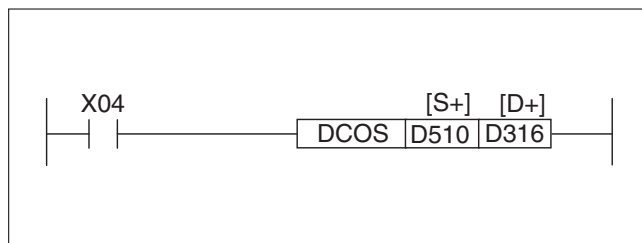


Рис. 7-96:

Пример программирования DCOS-инструкции

C000361C

При установке входа X4 вычисляется косинус угла, указанного в радианах по адресу D510, D511 (пересчет из градусов в радианы показан в примере раздела 7.7.18).

Результат записывается в D316 и D317.

△

7.7.20 РАСЧЕТ ТАНГЕНСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DTAN)

		DTAN		FNC 132			
		Расчет тангенса числа с плавающей запятой					
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U	
				●	●	●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D (числа с плавающей запятой в радианах (32 бита))	D (число с плавающей запятой (32 бита))	●	16бита	32бита	DTAN	9
					●	DTANP	9

ФУНКЦИЯ

Расчет тангенса числа с плавающей запятой и запоминание результата

ОПИСАНИЕ

- Рассчитывается тангенс числа с плавающей запятой, заданного в (S+). Результат запоминается в (D+).
- Для каждого операнда применяется соответственно 2 следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.
- Значение угла, указанное начиная с адреса (S+), должно быть в диапазоне между 0 и 360° (0 и 2π рад). Значение угла указывается в радианах (градусов × π / 180 [рад]). Для пересчета между градусами и радианами можно использовать команды DRAD и DDEG (см. разделы 7.7.24 и 7.7.25).

ПРИМЕР ▾

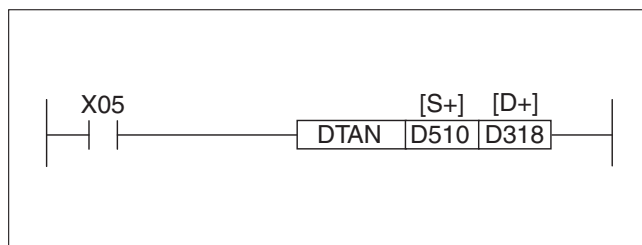


Рис. 7-97:

Пример программирования DTAN-инструкции

C000362C

С включением входа X5 рассчитывается значение тангенса угла, указанного в радианах (D510 и D511).

Результат записывается в D318 и D319.

7.7.21 ВЫЧИСЛЕНИЕ АРКСИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DASIN)

		DASIN		FNC 133				
		Вычисление арксинуса						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули(U□/G□) константа с плавающей запятой	D, R, специальные модули (U□/G□)		●	16бита	32бита	DASIN	9
	Используются 2 следующих друг за другом адреса операндов.					●	DASINP	9

Функция

Вычисление арксинуса (\arcsin) числа с плавающей запятой и сохранение результата

Вычисление арксинуса представляет собой функцию, обратную вычислению синуса. Команда DASIN определяет угол в радианах по синусу этого угла.

Описание

- Угол вычисляется из синуса, записанного по адресу начиная с (S+). Результат сохраняется по адресу начиная с (D+).
- В качестве (S+) и ((S+)+1) можно также указать константу с плавающей запятой.
- Значения в исходных и целевых операндах имеют формат с плавающей запятой.
- Синус, указанный начиная с (S+), должен находиться в диапазоне от -1,0 до 1,0.
- Результат в ((D+)+1 и (D+) - это угол в радианах (диапазон: от $-\pi/2$ до $\pi/2$). Для пересчета между градусами и радианами можно использовать команды DRAD и DDEG (см. разделы 7.7.24 и 7.7.25).

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Операнды ((S+)+1) и (S+) содержат значение, находящееся вне диапазона -1,0...1,0.

ПРИМЕР ▾

При включенном входе X0 следующая программа вычисляет угол из синуса, указанного в D0 и D1, и выводит результат в виде четырехзначной двоично-десятичной индикации. Дисплей для индикации подключен к 16 выходам с Y40 по Y57.

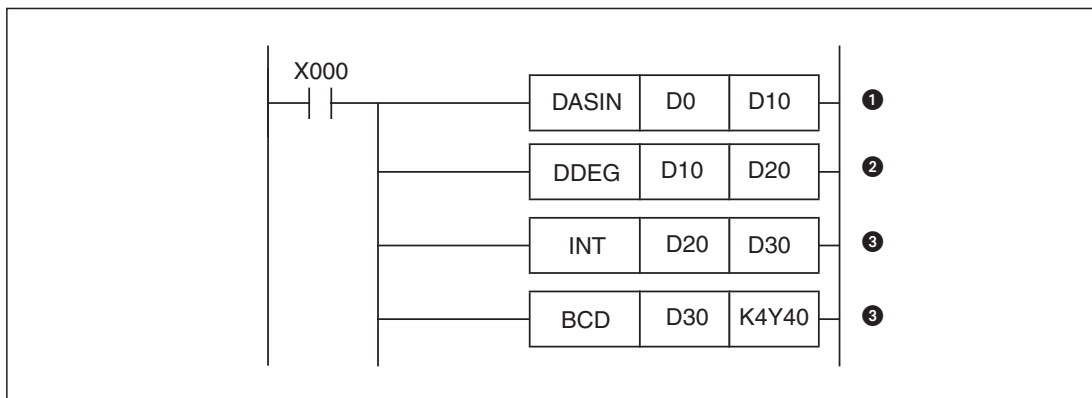


Рис. 7-98:Пример программы для вычисления арксинуса и вывода угла на двоично-десятичный дисплей.

- ① Команда DASIN вычисляет угол в радианах по его синусу
- ② С помощью команды DDEG угол пересчитывается в градусы
- ③ Число с плавающей запятой, обозначающее угол, преобразуется в целое десятичное число.
- ④ Угол отображается на двоично-десятичном дисплее.

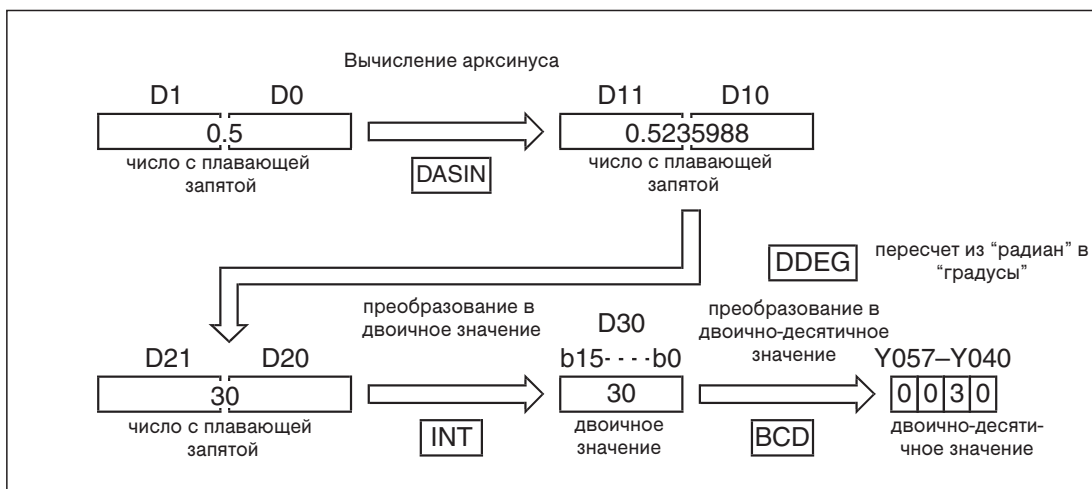


Рис. 7-99:При задании синуса "0,5" вышеприведенная программа выводит угол "30".

7.7.22 ВЫЧИСЛЕНИЕ АРККОСИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DACOS)

			DACOS		FNC 134		
			Вычисление арккосинуса				
CPU			FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D, R, специальные модули (U□/G□) (U□/G□) константа с плавающей запятой	D, R, специальные модули (U□/G□)	●	16бита	32бита	DACOS	9
	Используются 2 следующих друг за другом адреса операндов.				●	DACOSP	9

Функция

Вычисление арккосинуса (\arccos) числа с плавающей запятой и сохранение результата

Вычисление арккосинуса представляет собой функцию, обратную вычислению косинуса. Команда DACOS вычисляет угол в радианах из косинуса этого угла.

Описание

- Угол вычисляется из косинуса, хранящегося по адресу (S+). Результат сохраняется по адресу начиная с (D+).
- В качестве (S+) и ((S+)+1) можно также указать константу с плавающей запятой.
- Значения в исходных и целевых операндах имеют формат с плавающей запятой.
- Косинус, указанный в (S+), должен находиться в пределах от -1,0 до 1,0.
- Результат в ((D+)+1 и (D+) - это угол в радианах (диапазон: от 0 до π). Для пересчета между градусами и радианами можно использовать команды DRAD и DDEG (см. разделы 7.7.24 и 7.7.25).

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Операнды ((S+)+1) и (S+) содержат значение, находящееся вне диапазона -1,0...1,0.

ПРИМЕР ▾

При включенном входе X0 следующая программа вычисляет угол из косинуса, хранящегося в D0 и D1, и выводит результат в виде четырехзначной двоично-десятичной индикации, подключенной к 16 выходам с Y40 по Y57.

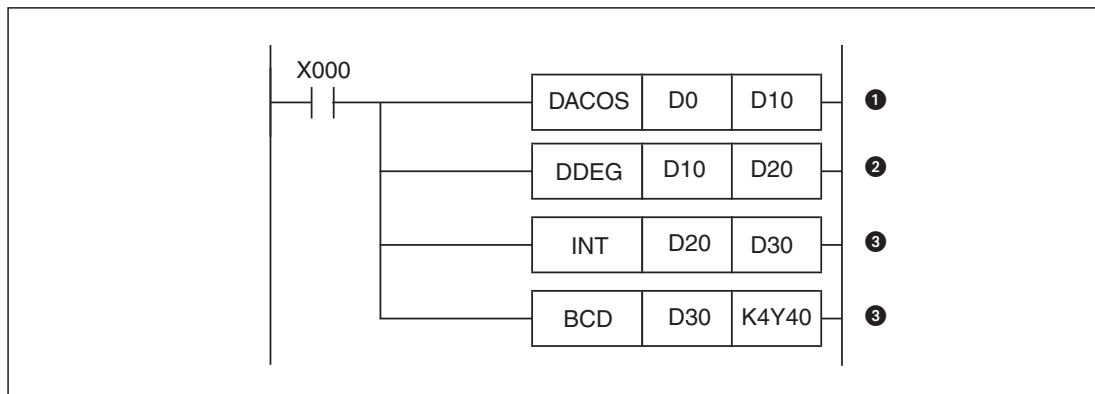


Рис. 7-100: Пример программы для вычисления арксинуса и вывода угла на двоично-десятичный дисплей.

- ❶ Команда DACOS вычисляет из косинуса угол в радианах.
- ❷ С помощью команды DDEG угол пересчитывается в градусы.
- ❸ Число с плавающей запятой, обозначающее угол, преобразуется в целое десятичное число.
- ❹ Угол отображается на двоично-десятичном дисплее.

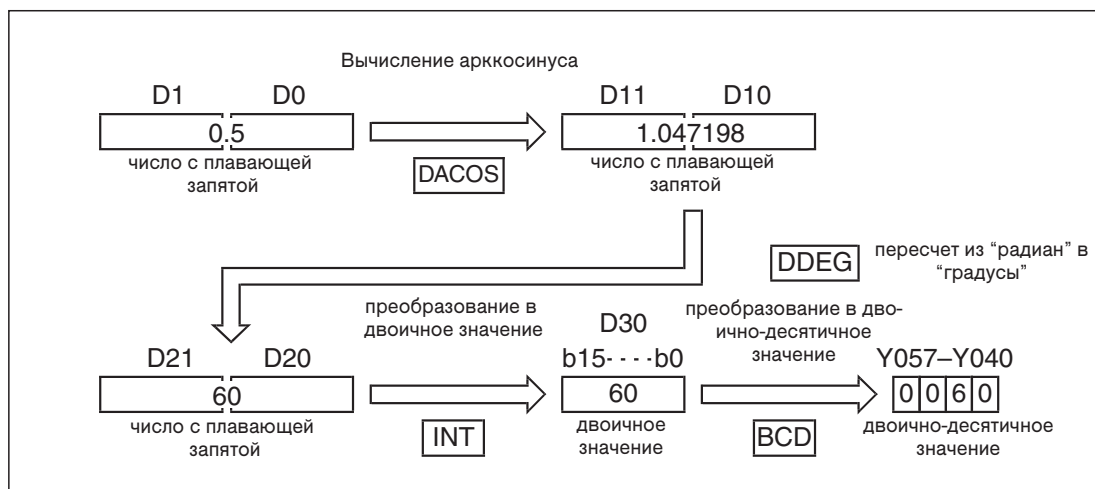


Рис. 7-101: При задании косинуса "0,5" вышеприведенная программа показывает угол "60".



7.7.23 ВЫЧИСЛЕНИЕ АРКТАНГЕНСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DATAN)

		DATAN		FNC 135			
		Вычисление арктангенса					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D, R, специальные модули (U□/G□) константа с плавающей запятой	D, R, специальные модули (U□/G□)	●	16бита	32бита	DATAN	9
	Используются 2 следующих друг за другом адреса операндов.				●	DATANP	9

Функция

Вычисление арктангенса (\arctg) числа с плавающей запятой и сохранение результата

Вычисление арктангенса представляет собой функцию, обратную вычислению тангенса. Команда DATAN вычисляет из тангенса соответствующий угол в радианах.

Описание

- Угол вычисляется из тангенса, указанного начиная с адреса (S+). Результат сохраняется по адресу начиная с (D+).
- В качестве (S+) и ((S+)+1) можно также указать константу с плавающей запятой.
- Значения в исходных и целевых операндах имеют формат с плавающей запятой.
- Результат в ((D+)+1 и (D+) - это угол в радианах (диапазон: от $-\pi/2$ до $\pi/2$). Для пересчета между градусами и радианами можно использовать команды DRAD и DDEG (см. разделы 7.7.24 и 7.7.25).

ПРИМЕР ▾

При включенном входе X0 следующая программа вычисляет угол из тангенса, записанного в D0 и D1, и выводит результат в виде четырехзначной двоично-десятичной индикации. Дисплей для индикации подключен к 16 выходам с Y40 по Y57.

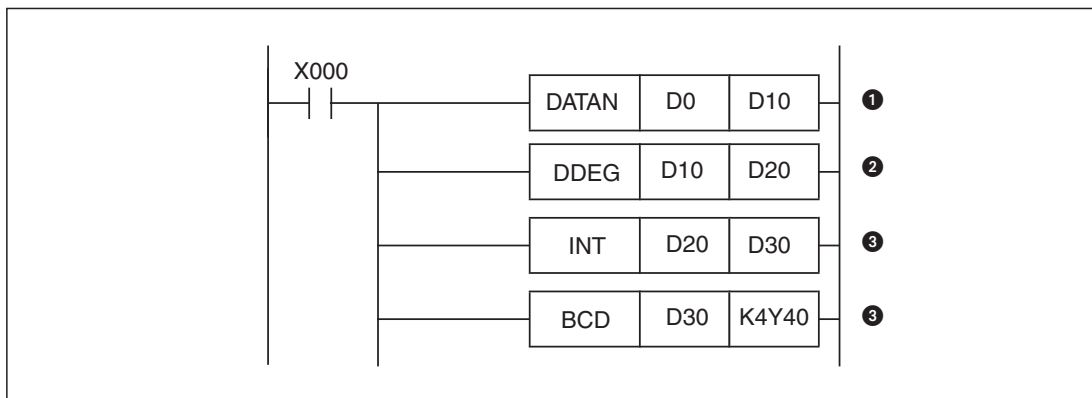


Рис. 7-102:Пример программы для вычисления арктангенса и вывода угла на двоично-десятичный дисплей.

- ❶ Команда DATAN вычисляет из тангенса угол в радианах.
- ❷ С помощью команды DDEG угол пересчитывается в градусы.
- ❸ Число с плавающей запятой, обозначающее угол, преобразуется в целое десятичное число.
- ❹ Угол отображается на двоично-десятичном дисплее.

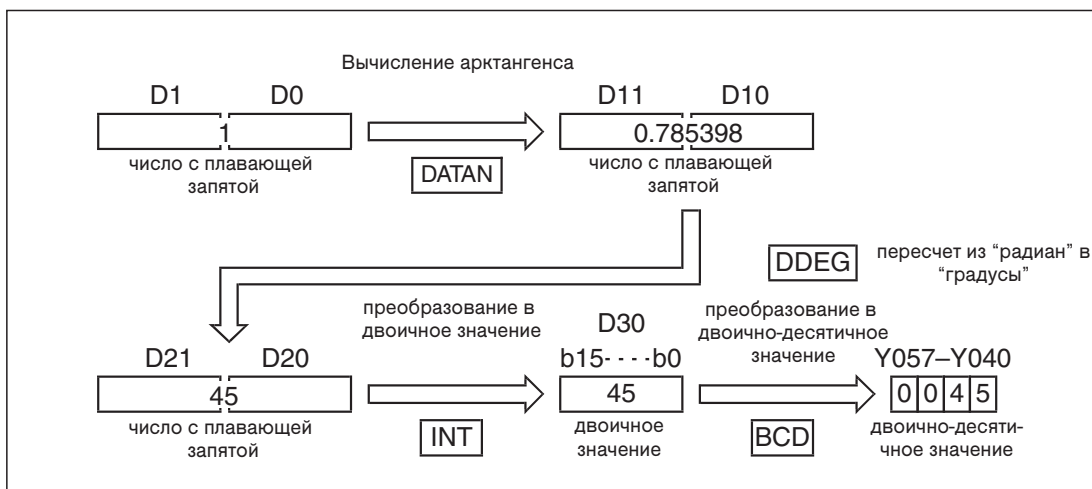



Рис. 7-103:При задании тангенса "1" вышеприведенная программа показывает угол "45".

7.7.24 ПЕРЕСЧЕТ ИЗ ГРАДУСОВ В РАДИАНЫ (DRAD)

		DRAD		FNC 136				
		Пересчет из градусов в радианы						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U□/G□) константа с плавающей запятой	D, R, специальные модули (U□/G□)		●	16бита	32бита	DRAD	9
	Используются 2 следующих друг за другом адреса операндов.					●	DRADP	9

Функция

Пересчет угла из градусов в радианы

Описание

- Команда DRAD пересчитывает значение в градусах ($^{\circ}$), сохраненное по адресу ((S+)+1) и (S+), в соответствующее значение в радианах (рад) и сохраняет результат по адресу ((D+)+1) и (D+).

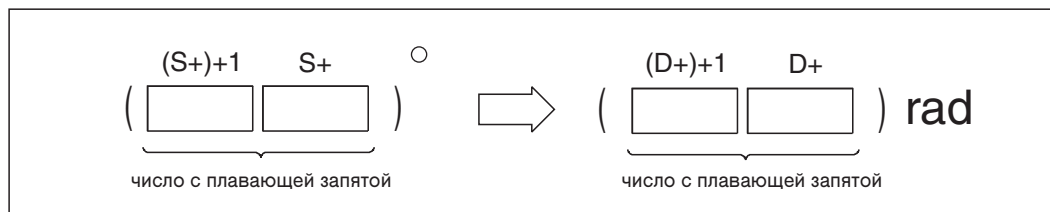


Рис. 7-104: Принцип действия команды DRAD

- Для пересчета градусов в радианы применяется следующая формула:

$$\text{Radiant} = \text{Grad} \times \frac{\pi}{180}$$

- В качестве (S+) и ((S+)+1) можно также указать константу с плавающей запятой.
- Значения в исходных и целевых операндах имеют формат с плавающей запятой.

ПРИМЕР ▾

Ко входам X020...X037 подключен четырехразрядный двоично-десятичный выключатель, с помощью которого можно вводить значение угла в градусах. При включенном входе X0 следующая программа пересчитывает значение в радианы и сохраняет результат по адресу D20 и D21.

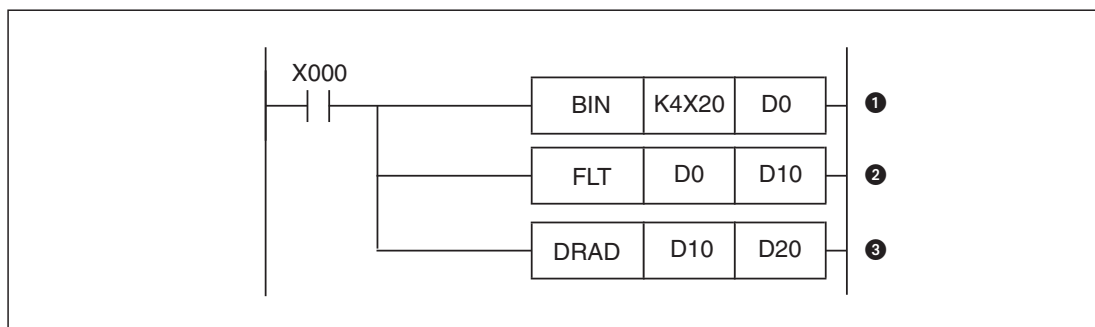


Рис. 7-105: Пример программы для применения команды DRAD

- ① Считывание настроенного значения в градусах
- ② Угол преобразуется в число с плавающей запятой.
- ③ Пересчет из градусов в радианы

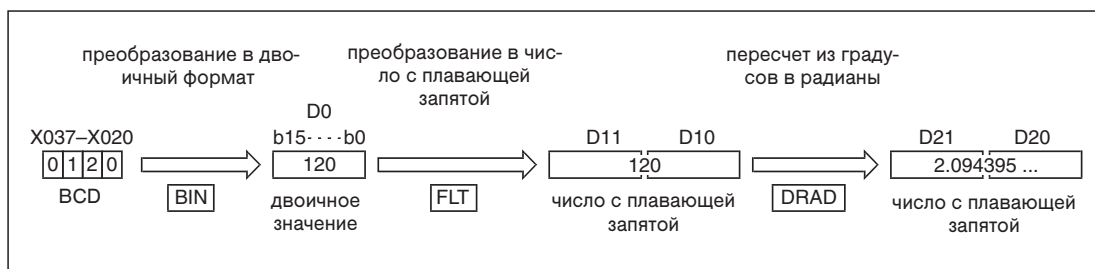



Рис. 7-106: Пример программы, которая при вводе 120° выводит результат преобразования 2,094395 рад.

7.7.25 ПЕРЕСЧЕТ ИЗ РАДИАН В ГРАДУСЫ (DDEG)

		DRAD		FNC 136			
		Пересчет из радиан в градусы					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	D, R, специальные модули (U□/G□) константа с плавающей запятой	D, R, специальные модули (U□/G□)		●	16бита	32бита	DDEG
	Используются 2 следующих друг за другом адреса операндов.				●	DDEGP	9

Функция

Пересчет угла из радиан в градусы

Описание

- Команда DDEG преобразует угол, указанный в радианах (рад) по адресу ((S+)+1) и (S+), в соответствующее значение в градусах (°) и сохраняет результат по адресу ((D+)+1) и (D+).

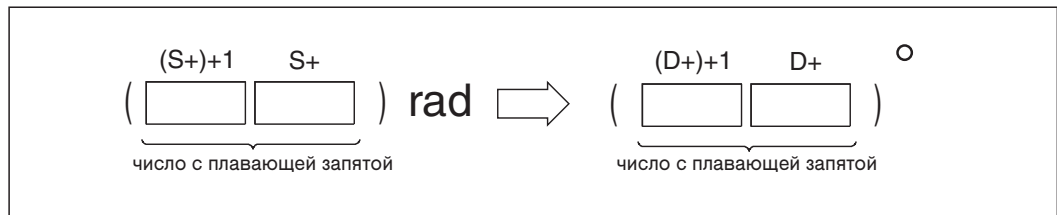


Рис. 7-107: Принцип действия команды DDEG

- Для пересчета градусов в радианы применяется следующая формула:

$$\text{Grad} = \text{Radiant} \times \frac{180}{\pi}$$

- В (S+) и ((S+)+1) можно также указать константу с плавающей запятой.
- Значения в исходных и целевых операндах имеют формат с плавающей запятой.

ПРИМЕР ▾

Следующий пример программы преобразует угол в радианах, указанный по адресу D20 и D21, в угол в градусах и выдает результат в виде четырехзначной двоично-десятичной индикации. Дисплей для индикации подключен к 16 выходам с Y40 по Y57.

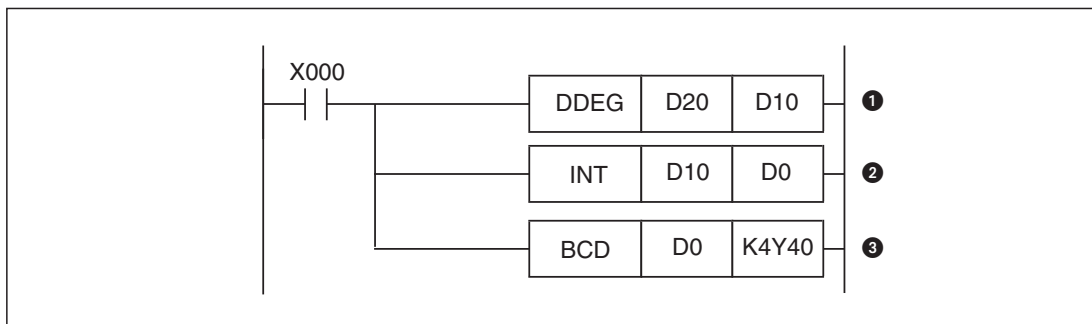


Рис. 7-108: Пример программы, применяющей команду DDEG

- ① Угол в радианах пересчитывается в градусы.
- ② Число с плавающей запятой, обозначающее угол, преобразуется в целое десятичное число.
- ③ Угол отображается на двоично-десятичном дисплее.

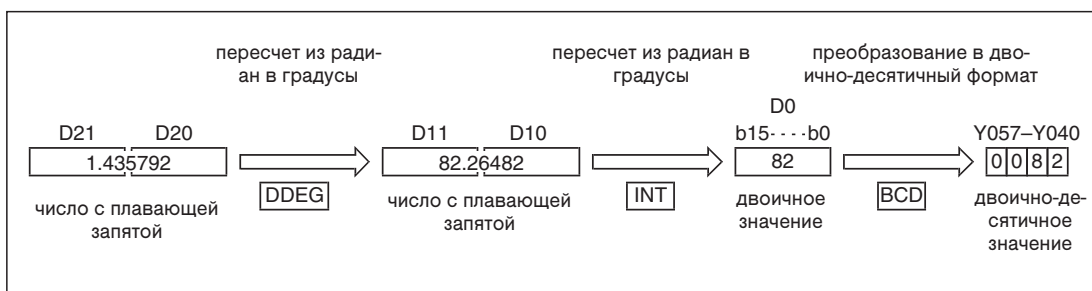


Рис. 7-109: Пример программы, которая при вводе 1,435792 рад преобразует это значение в 82 градуса и выводит результат.

7.8 ИНСТРУКЦИИ ПО ОБРАБОТКЕ ДАННЫХ

ОБЗОР ИНСТРУКЦИЙ FNC 140 ... 149

Символ	FNC	Назначение	Разд
WSUM	140	Суммирование содержимого словных операндов	7.8.1
WTOB	141	Разделение данных в словных операндах на байты	7.8.2
BTOW	142	Образование словного операнда из отдельных байтов	7.8.3
UNI	143	Объединение групп по 4 бита в словный операнд	7.8.4
DIS	144	Разделение словного операнда на группы по 4 бита	7.8.5
SWAP	147	Обмен старшими и младшими байтами	7.8.6
SORT2	149	Сортировка данных в таблице	7.8.7

Табл. 7-21: Обзор команд обработки данных

7.8.1 СУММИРОВАНИЕ СОДЕРЖИМОГО СЛОВНЫХ ОПЕРАНДОВ (WSUM)

				WSUM		FNC 140			
				Суммирование содержимого словных операндов					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R, специальные модули (U□/G□)		D, R, K, H	●	16бита	32бита	WSUM WSUMP	7	
					●	●	DWSUM DWSUMP	13	

Функция

Образование суммы из содержимых словных операндов

Описание

- Команда WSUM образует сумму указанного в (n) количества блоков 16-битных или 32-битных двоичных данных. В (S+) указывается первый адрес области операндов, в которой хранятся суммируемые операнды. Результат сохраняется в операнде начиная с адреса (D+).

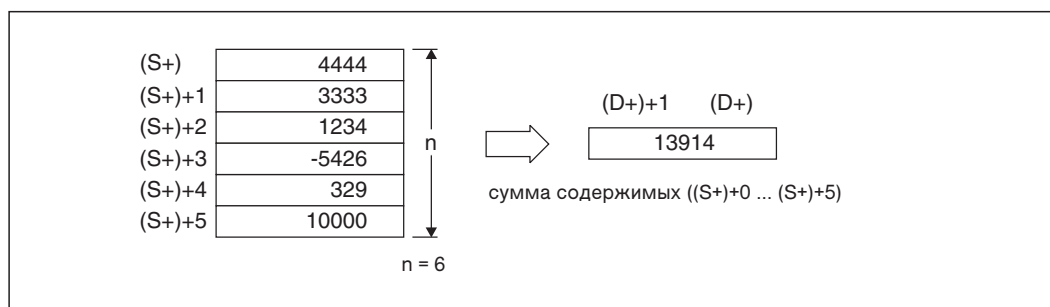


Рис. 7-110:Пример выполнения команды WSUM для суммирования 16-битных данных

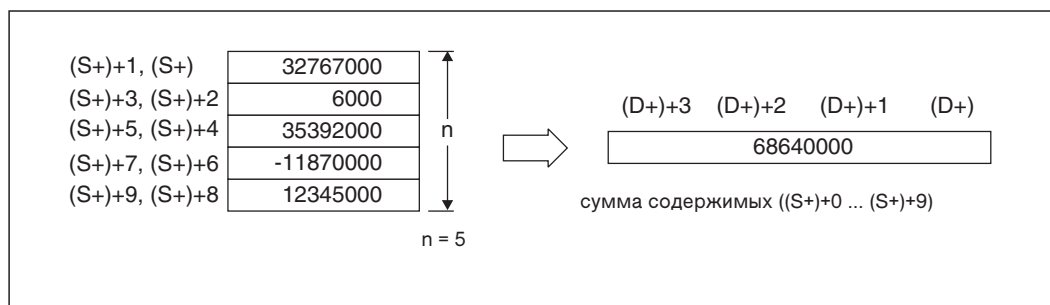


Рис. 7-111:Пример выполнения команды DWSUM для суммирования 32-битных данных

- В (n) необходимо указать значение больше "0".

УКАЗАНИЕ

При сложении 32-битных данных с помощью команды DWSUM результат сохраняется в виде 64-битного значения. Однако контроллеры серии FX3U не могут оперировать 64-битными данными. Если сумма находится в допустимом диапазоне значений для 32-битных данных (от -2.147.483.648 до 2.147.483.647), то можно использовать содержимое регистров (D+) и ((D+)+1), а содержимым ((D+)+3) и ((D+)+2) можно пренебречь.

Для суммирования содержимых байтов можно использовать команду CCD.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- В результате указания (n) превышает допустимая область для операндов, указанная с помощью (S+).
- В (n) указано отрицательное значение или "0".

ПРИМЕР ▾

При включении входа X10 (нарастающий фронт) следующий пример программы вычисляет сумму 16-битных данных, хранящихся в регистрах с D10 по D14, и записывает результат в D101 и D100.

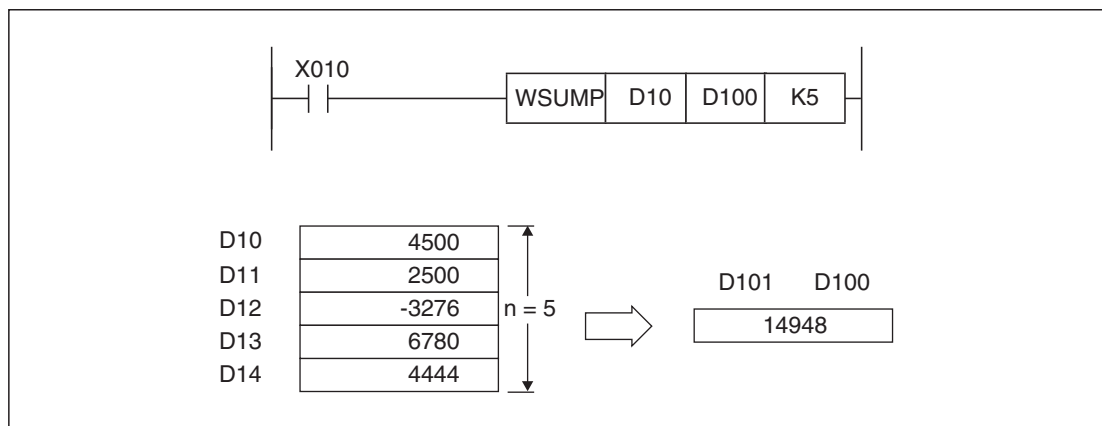


Рис. 7-112: Суммирование с помощью команды WSUMP

△

7.8.2 РАЗДЕЛЕНИЕ ДАННЫХ В СЛОВНЫХ ОПЕРАНДАХ НА БАЙТЫ (WTOB)

				WTOB		FNC 141			
				Разделение данных в словных операндах на байты					
Операнды				Имп. инструкция(P)	Обработка		Шаги программы		
					CPU	FX1S	FX1N	FX2N	FX2NC
S+			D+	n	●	16бита	32бита	WTOB	7
T, C, D, R			D, R, K, H	●			WTOBP		

Функция

Разделение данных

Описание

- Команда WTOB разделяет 16-битовое значение на байты и сохраняет отдельные байты один за другим в целевой области, первый адрес которой указан в (D+). Первый адрес области, в которой хранятся разделяемые данные, указывается в (S+). В (n) указывается количество байтов в целевой области. Для сохранения используются только младшие байты операндов, указанных в (D+).

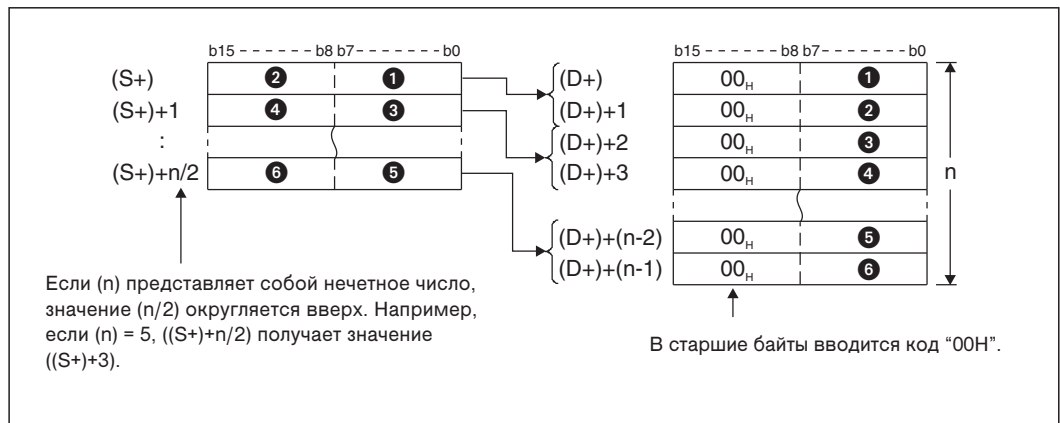


Рис. 7-113:Принцип действия команды WTOB

- Старшие байты операндов, указанных в (D+), заполняются значениями "00H".
- Если в (n) указано нечетное число, от последнего исходного операнда используется только младший байт.

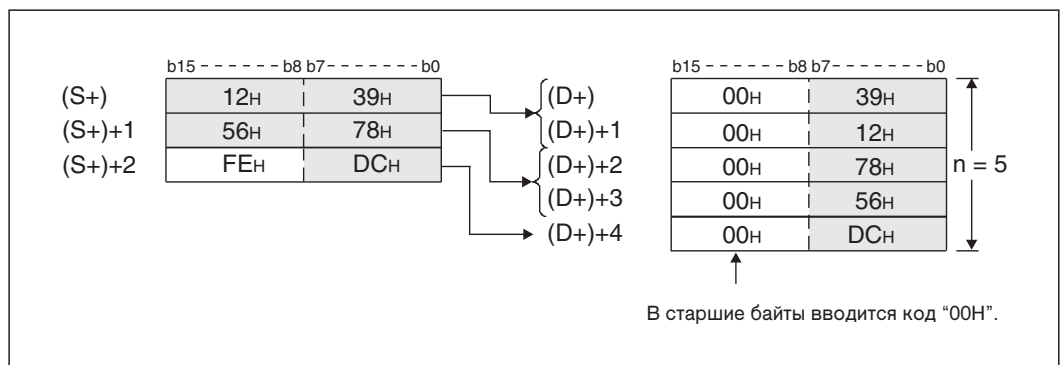


Рис. 7-114:Например, если в (n) указано "5", используются данные от (S+) до младшего байта ((S+)+2).

- Если в (n) указано значение "0", команда WTOB не выполняется.
- Области операндов в (S+) и (D+) могут перекрываться. Однако если в этом случае в (n) указано нечетное значение, старший байт последнего исходного операнда перезаписывается значением "00H".

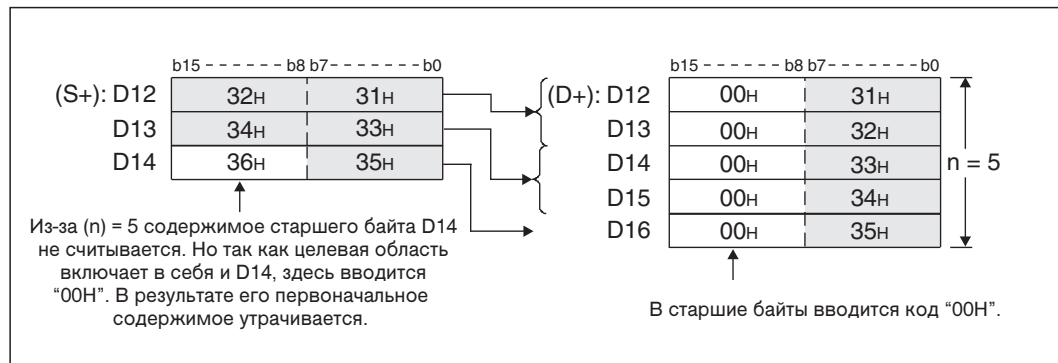


Рис. 7-115:Пример использования одних и тех же операндов в качестве исходной и целевой области.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Допустимая область операндов превышает операндами, указанными по адресу с (S+) по $((S+)+n/2)$. Если число (n) нечетное, для количества операндов принимается значение $(n/2)$ после его округления в большую сторону.
- Допустимая область операндов превышает операндами, указанными по адресу с (D+) по $((D+)+(n-1))$.

ПРИМЕР ▾

При включении входа X0 данные, хранящиеся в регистрах с D10 по D12, разделяются на байты и результат записывается по адресу с D20 по D25.

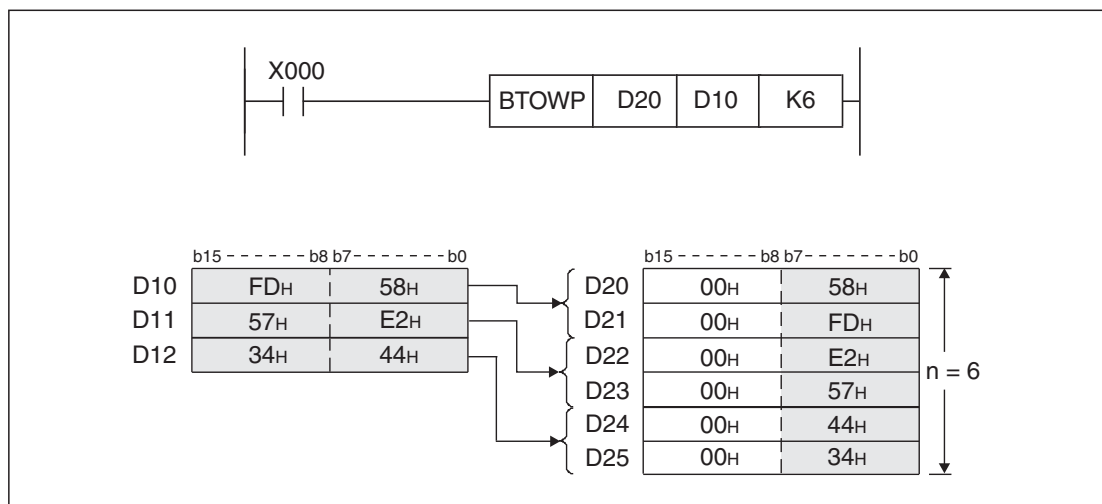


Рис. 7-116:Пример разделения данных с помощью команды WTOBP



7.8.3 ОБРАЗОВАНИЕ СЛОВНОГО ОПЕРАНДА ИЗ ОТДЕЛЬНЫХ БАЙТОВ (BTOW)

				BTOW		FNC 142			
				Образование словного операнда из отдельных байтов					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R				●	16бита	32бита	BTOW BTOWP	

Функция

Объединение данных

Описание

- Команда BTOW считывает содержимые младших байтов словных операндов, начиная с адреса (S+), и записывает данные попеременно в младший и старший байт целевой области, первый адрес которой указан в (D+). В (n) указывается количество байтов и, тем самым, число исходных операндов.

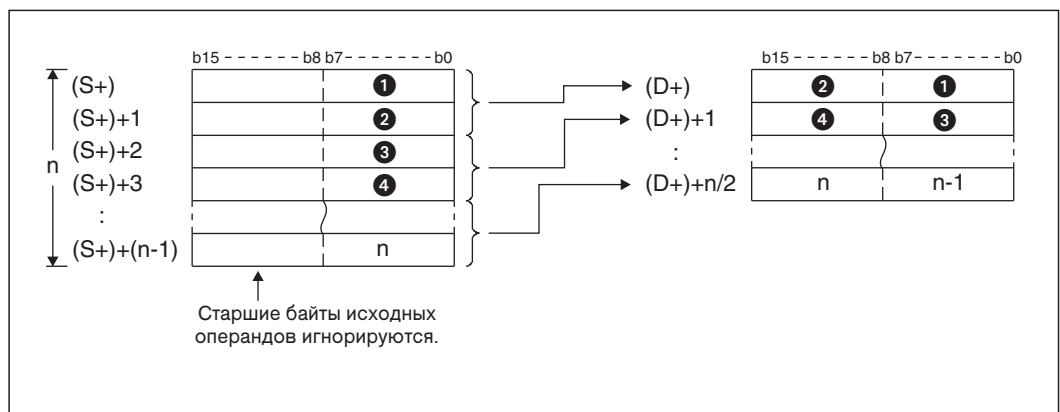


Рис. 7-117: Принцип действия команды BTOW

- Старшие байты словных операндов, указанных в (S+), игнорируются.
- Если в (n) указано нечетное число, в старший байт последнего целевого операнда записывается значение "00_n".

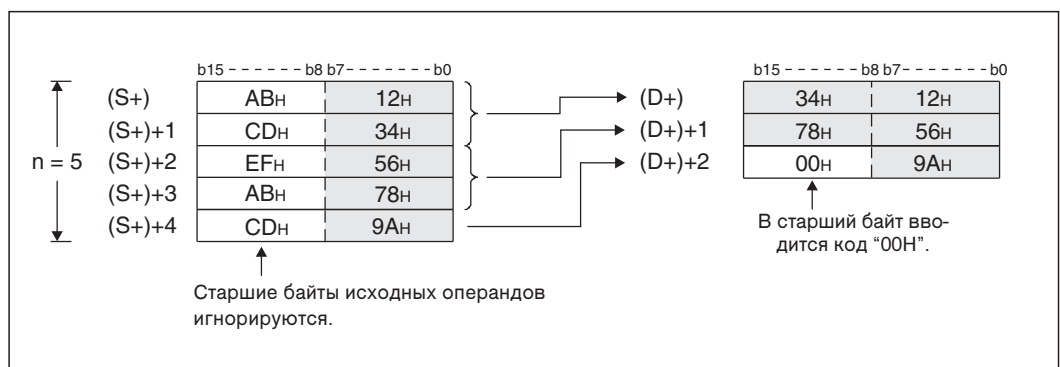


Рис. 7-118: Например, если в (n) указано "5", данные вводятся от (D+) до младшего байта ((D+)+2), а старший байт ((D+)+2) стирается.

- Если в (n) указано значение "0", команда BTOW не выполняется.

- Указанные в (S+) и (D+) области операндов могут перекрываться. Однако в этом случае перезаписываются содержимые старших байтов исходных операндов, которые одновременно используются в качестве целевых операндов.

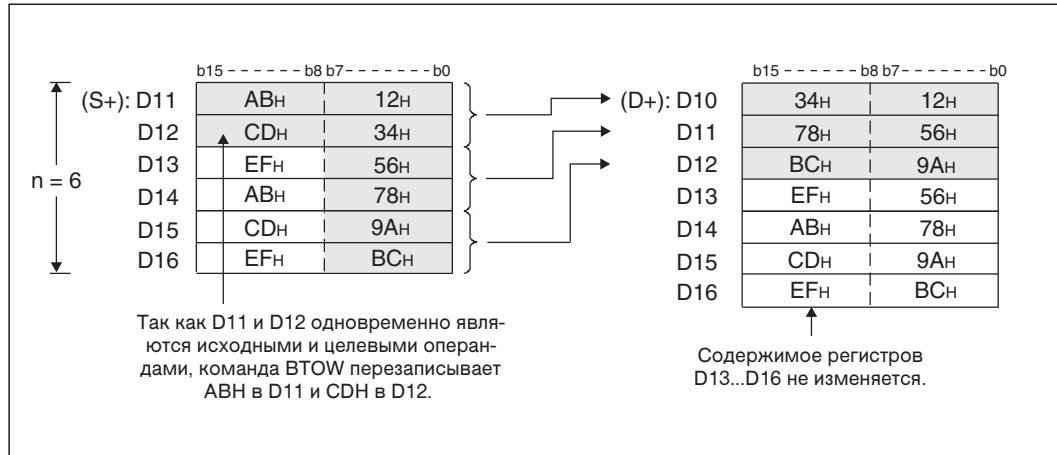


Рис. 7-119: Пример использования одних и тех же операндов в качестве исходной и целевой области.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Операнды, указанные в регистрах с (S+) по ((S+)+(n-1)), превышают допустимую область операндов.
- Допустимая область операндов превышает операндами, указанными по адресу с (D+) по ((D+)+n/2). Если число (n) нечетное, для количества операндов принимается значение (n/2) после его округления в большую сторону.

ПРИМЕР ▽

При включении входа X0 данные, хранящиеся в младших байтов регистров данных с D20 по D25, объединяются и записываются в регистры с D10 по D12.

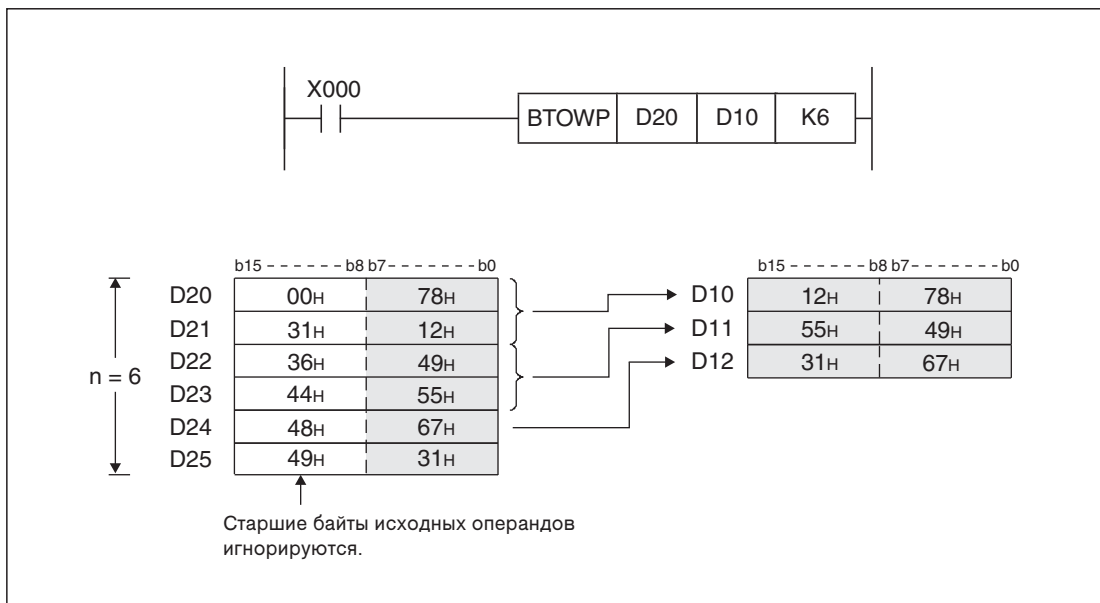


Рис. 7-120: Пример объединения данных с помощью команды BTOWP



7.8.4 ОБЪЕДИНЕНИЕ ГРУПП ПО 4 БИТА В СЛОВНЫЙ ОПЕРАНД (UNI)

				UNI		FNC 143				
				Объединение групп по 4 бита в словный операнд						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
										●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы			
	T, C, D, R		D, R, K, H	●	16бита	32бита	UNI	UNIP	7	

Функция

До четырех 4-битных групп объединяются в один словный операнд.

Описание

- Команда UNI отделяет 4 самых младших бита до четырех 16-битных значений и сохраняет их состояния вместе в одном словном операнде (16 битов). В (S+) устанавливается начальный адрес объединяемых значений, в (n) - количество операндов, следующих друг за другом, а в (D+) - целевой адрес.

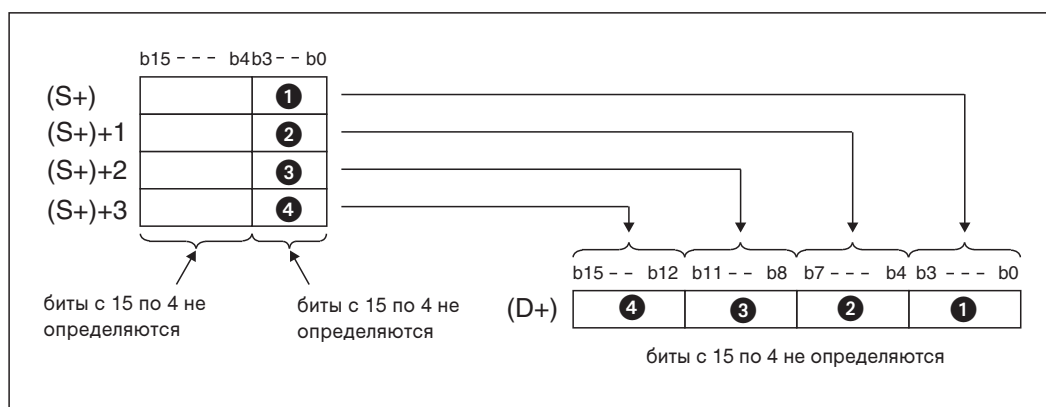


Рис. 7-121: Принцип действия команды UNI

- Для (n) можно задать значение между 1 и 4. Если в (n) указано значение "0", команда UNI не выполняется.
- Если для (n) указано значение между 1 и 3, то в (D+) вводится только соответствующее количество 4-битных групп. Остальные биты в (D+) сбрасываются на "0".

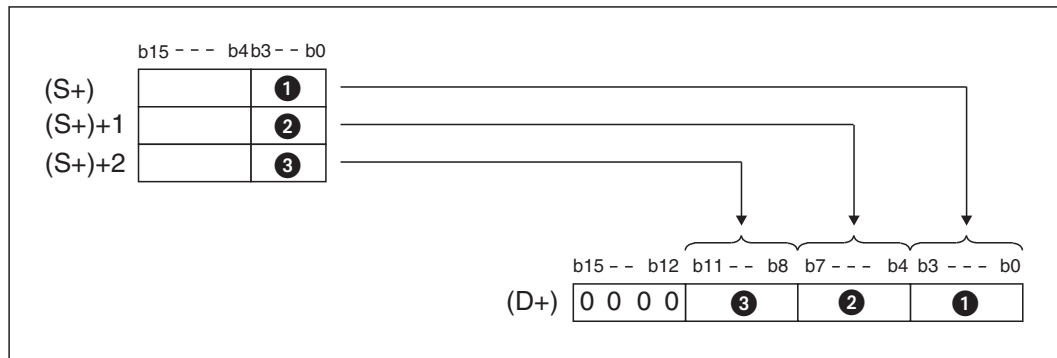


Рис. 7-122: Например, если в (n) указано “3”, биты с 15 по 12 в регистре (D+) сбрасываются.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки “6706”:

- В результате указания (n) превышает допустимая область для операндов, указанная с помощью (S+).
- Для (n) не было указано значение между 1 и 4.

ПРИМЕР ▽

При включении входа X0 биты с 3 по 0 регистров данных с D0 по D2 считываются, объединяются и записываются в D10. Так как для (n) указано “3”, биты с 15 по 12 в регистре D10 сбрасываются.

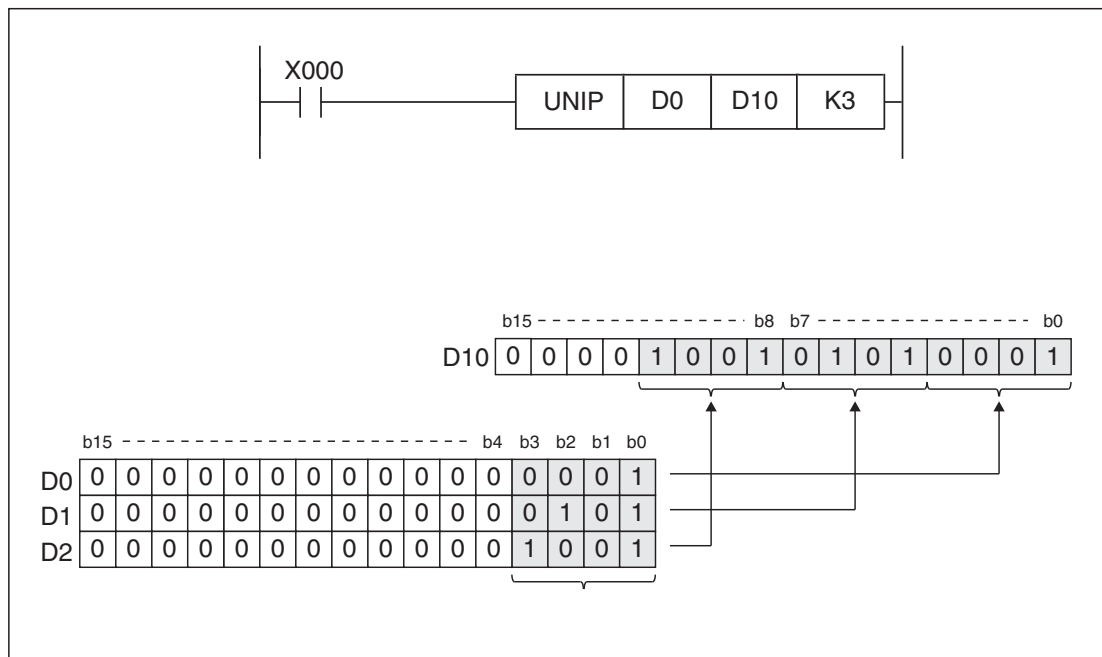


Рис. 7-123: Пример группирования 4-битных данных с помощью команды UNIP



7.8.5 РАЗДЕЛЕНИЕ СЛОВНОГО ОПЕРАНДА НА ГРУППЫ ПО 4 БИТА (DIS)

				DIS		FNC 144		
				Разделение словного операнда на группы по 4 бита				
				CPU	FX1S	FX1N	FX2N	FX2NC
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	T, C, D, R		D, R, K, H		●	16бита	32бита	DIS DISP

Функция

Словный операнд разделяется на 4-битные группы (максимум четыре).

Описание

- Команда DIS разделяет словный операнд с 16-битным значением на группы по 4 бита и записывает их состояния в максимум 4 следующих друг за другом целевых операнда. В (S+) указывается разделяемый 16-битный операнд, в (n) - количество 4-битных групп, а в (D+) - первый целевой адрес. Прочие 4-битные группы записываются до операнда ((D+)+n).

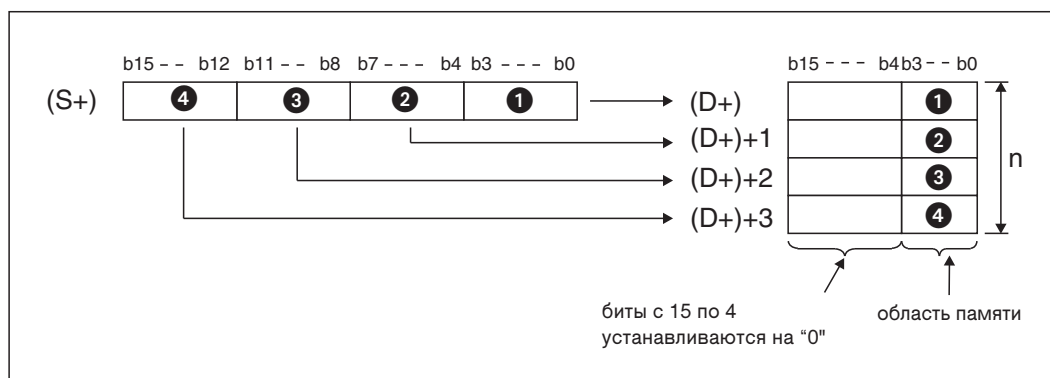


Рис. 7-124: Принцип действия команды DIS

- Для (n) можно задать значение между 1 и 4. Если в (n) указано значение "0", команда DIS не выполняется.
- Самые старшие 12 битов n операндов, начиная с адреса, указанного в (D+), устанавливаются на "0".

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- При указании (n) превышает допустимая область для операнда, указанного в (D+).
- Для (n) не было указано значение между 1 и 4.

ПРИМЕР ▾

Следующая программа при включении X0 разделяет 16-битовое значение из D0 и сохраняет этот набор битов в виде групп по 4 бита, следующих друг за другом в регистрах с D10 по D13.

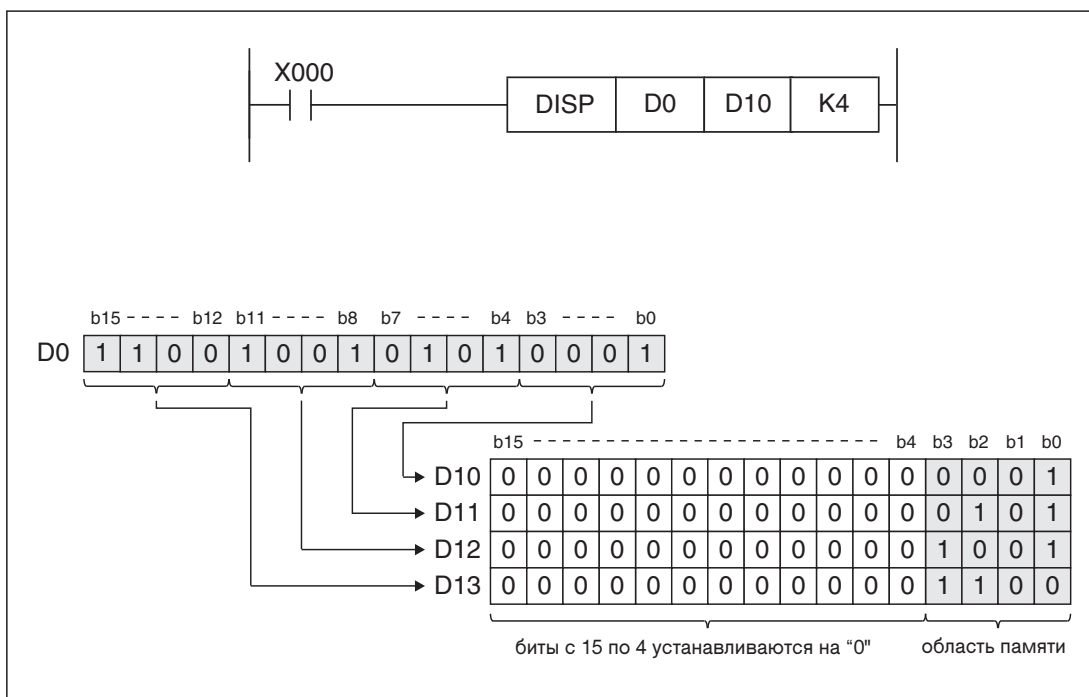


Рис. 7-125:Пример разбивки данных с помощью команды `DISP`



7.8.6 ОБМЕН СТАРШИМИ И МЛАДШИМИ БАЙТАМИ (SWAP)

		SWAP		FNC 147			
		ОБМЕН СТАРШИМИ И МЛАДШИМИ БАЙТАМИ					
Операнды		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	●
S+		Имп. инструкция(P)		Обработка		Шаги программы	
KnY, KnM, KnS, T, C, D, V, Z		●		16бита	32бита	SWAP, SWAPP	5*
				●	●	DSWAP, DSWAPP	9*

* В контроллерах серии FX3U для команд SWAP и SWAPP нужны 3 шага программы, а для DSWAP и DSWAPP - 5 шагов программы.

ФУНКЦИЯ

Обмен старшими и младшими байтами операнда

ОПИСАНИЕ

- При применении SWAP-инструкции старший и младший байты операнда (D+) обмениваются местами.
- При применении DSWAP-инструкции соответственно обмениваются местами старшие и младшие байты операндов (D+) и ((D+)+1).
- Эта операция выполняется вновь в каждом цикле программы. Чтобы гарантировать одноразовое выполнение, нужно применить импульсную инструкцию или блокировку.

УКАЗАНИЕ

Функция команды SWAP соответствует функции команды XCH при установленном специальном маркере M8160 (см. раздел 6.3.8).

ПРИМЕР ▾

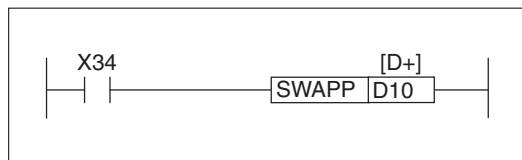


Рис. 7-126:
Пример программирования SWAPP-инструкции

C000363C

При положительном фронте входа X34 старший и младший байты D10 обмениваются местами.

Операнд	Байт	Перед выполнением инструкции	После выполнения инструкции
D10	1	1FH	8BH
	2	8BH	1FH

Табл. 7-21:
Обмен байтами при SWAPP-инструкции

Если вместо SWAPP-инструкции применяется DSWAPP-инструкция, то при положительном фронте X34 соответственно обмениваются местами старшие и младшие байты операндов D10 и D11.

Операнд	Байт	Перед выполнением инструкции	После выполнения инструкции
D10	1	1FH	8BH
	2	8BH	1FH
D11	1	C4H	35H
	2	35H	C4H

Табл. 7-22:
Обмен байтами при DSWAPP-инструкции



7.8.7 СОРТИРОВКА ДАННЫХ В ТАБЛИЦЕ (SORT2)

						SORT2		FNC149			
						Сортировка данных в таблице					
						CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
											●
Операнды	S+	n1	n2	D+	m	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R ①	D, R K, H ②	K, H ③	D, R	D, R, K, H ④			16бита	32бита	SORT2	11
							●	●	DSORT2	21	

- ① ((n1) x (n2)) следующих друг за другом регистров данных
 ② (n1) указывает количество строк таблицы (от 1 до 32)
 ③ (n2) указывает количество столбцов таблицы (от 1 до 6)
 ④ (m) является критерием сортировки (номер столбца); для (m) можно указывать значения от 1 до (n2).

Принцип действия

Сортировка таблицы по возрастанию или по убыванию значений столбца.

Порядок сортировки указывается специальным маркером M8156:

- M8156 = 0: возрастающая последовательность
- M8156 = 1: убывающая последовательность

Описание (для 16-битных данных)

- При выполнении команды SORT2 внутренняя матрица данных (таблица), заданная путем указания начального регистра данных (S+) и размера ((n1) строк и (n2) столбцов), сортируется по значениям столбца (m) и результат записывается, начиная с регистра данных (D+).

ПРИМЕР ▾

Следующая таблица из трех строк и четырех столбцов содержит данные персонала:

№ столбца	1	2	3	4
№ строки	Идент. номер	Рост	Вес	Возраст
1	(S+)	((S+)+1)	((S+)+2)	((S+)+3)
	1	150	45	20
2	((S+)+4)	((S+)+5)	((S+)+6)	((S+)+7)
	2	180	50	40
3	((S+)+8)	((S+)+9)	((S+)+10)	((S+)+11)
	3	160	70	30

Табл. 7-22: Несортированная таблица

После выполнения команды SORT2 (n1 = K3, n2 = K4, m = K2) таблица сортируется по возрастанию значений столбца 2 (M8156 = 1):

№ столбца	1	2	3	4
№ строки	Идент. номер	Рост	Вес	Возраст
1	(D+)	((D+)+1)	((D+)+2)	((D+)+3)
	1	150	45	20
2	((D+)+4)	((D+)+5)	((D+)+6)	((D+)+7)
	3	160	70	30
3	((D+)+8)	((D+)+9)	((D+)+10)	((D+)+11)
	2	180	50	40

Табл. 7-23: Сортированная таблица после выполнения команды SORT2

Описание (для 32-битных данных)

- При выполнении команды SORT2 внутренняя матрица данных (таблица), заданная путем указания начального регистра данных ((S+)+1) и (S+) и размера ((n1) строк и (n2) столбцов), сортируется по значениям столбца (m) и записывается заново, начиная с адреса ((D+)+1) и (D+).

ПРИМЕР ▾

Следующая таблица из трех строк и четырех столбцов содержит рецепты:

№ столбца	1	2	3	4
№ строки	Блок данных	Вода	Мука	Сахар
1	((S+)+1), (S+)	((S+)+3), ((S+)+2)	((S+)+5), ((S+)+4)	((S+)+7), ((S+)+6)
	1	56879	27478	16890
2	((S+)+9), ((S+)+8)	((S+)+11), ((S+)+10)	((S+)+13), ((S+)+12)	((S+)+15), ((S+)+14)
	2	44878	21388	15722
3	((S+)+17), ((S+)+16)	((S+)+19), ((S+)+18)	((S+)+21), ((S+)+20)	((S+)+23), ((S+)+22)
	3	23898	11999	18743

Табл. 7-24: Несортированная таблица

После выполнения команды DSORT2 (n1 = K3, n2 = K4, m = K2) таблица сортируется по возрастанию значений столбца 2 (M8156 = 1):

№ столбца	1	2	3	4
№ строки	Блок данных	Вода	Мука	Сахар
1	((D+)+1), (D+)	((D+)+3), ((D+)+2)	((D+)+5), ((D+)+4)	((D+)+7), ((D+)+6)
	3	23898	11999	18743
2	((D+)+9), ((D+)+8)	((D+)+11), ((D+)+10)	((D+)+13), ((D+)+12)	((D+)+15), ((D+)+14)
	2	44878	21388	15722
3	((D+)+17), ((D+)+16)	((D+)+19), ((D+)+18)	((D+)+21), ((D+)+20)	((D+)+23), ((D+)+22)
	1	56879	27478	16890

Табл. 7-25: Сортированная таблица после выполнения команды DSORT2

- Если в качестве (n1) указывается регистр данных (D) или регистр файлов (R), то длина данных составляет 32 бита. Например, если в (n1) указан адрес "D0", количество строк в D1 и D2 сохраняется в виде 32-битного значения.

УКАЗАНИЕ

Команду SORT2 в программе разрешается использовать максимум два раза.

Сортировка начинается, если выполнено входное условие команды SORT2, и завершается по истечении (n1) программных циклов. По окончании команды SORT2 устанавливается специальный маркер M8029.

Чтобы выполнить команду SORT2 еще раз, перед этим необходимо сбросить связь команды со входом.

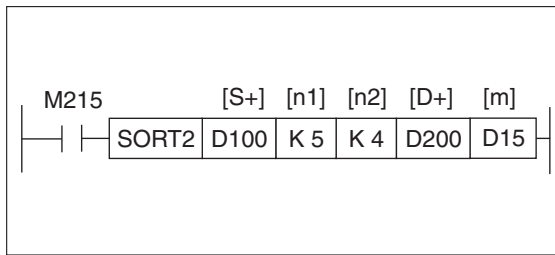
Программу, содержащую команду SORT2, невозможно передать в контроллер или изменить при режиме RUN контроллера.

Если в (S+) и (D+) указаны одни и те же операнды, исходные данные перезаписываются отсортированными данными.

Во время процесса сортировки данные в таблице нельзя изменять, так как в противном случае будут записаны неправильные данные.

ПРИМЕР ▾

Сортировка таблицы из 5 строк и 4 столбцов.

**Рис. 7-127:**

Пример программы для применения команды SORT

В другом месте программы в D15 записывается номер столбца, по значениям которого требуется отсортировать таблицу.

Столбец	1	2	3	4
Строка	№	Знач. 1	Знач. 2	Знач. 3
1	D100	D101	D102	D103
	1	150	45	20
2	D104	D105	D106	D107
	2	180	50	40
3	D108	D109	D110	D111
	3	160	70	30
4	D112	D113	D114	D115
	4	100	20	8
5	D116	D117	D118	D119
	5	150	35	45

Табл. 7-26:

Несортированная таблица

Для лучшей обзорности в первом столбце желательно предусмотреть контрольный номер, чтобы после сортировки можно было легко идентифицировать строки.

Столбец	1	2	3	4
Строка	№	Знач. 1	Знач. 2	Знач. 3
1	D200	D201	D202	D203
	4	100	20	8
2	D204	D205	D206	D207
	1	150	45	20
3	D208	D209	D210	D211
	5	150	35	45
4	D212	D213	D214	D215
	3	160	70	30
5	D216	D217	D218	D219
	2	180	50	40

Табл. 7-27:

Сортированная таблица

Сортировка выполнена по возрастанию содержимого столбца 2 (m = содержимое D15 = K2).

Столбец	1	2	3	4
Строка	№ .	Знач. 1	Знач. 2	Знач. 3
1	D200	D201	D202	D203
	3	160	70	30
2	D204	D205	D206	D207
	2	180	50	40
3	D208	D209	D210	D211
	1	150	45	20
4	D212	D213	D214	D215
	5	150	35	45
5	D216	D217	D218	D219
	4	100	20	8

Табл. 7-28:

Сортированная таблица

Сортировка выполнена по убыванию содержимого столбца 3 (m = содержимое D15 = K3).

△

7.9 КОМАНДЫ ПОЗИЦИОНИРОВАНИЯ

Благодаря командам позиционирования, контроллеры серий FX1S, FX1N и FX3U в сочетании с сервоусилителями МИЦУБИСИ могут применяться для позиционирования.

УКАЗАНИЕ

Позиционирование с помощью контроллера FX3U и применение следующих команд подробно описано в руководстве по позиционированию для серии FX3U. Поэтому примеры в этом разделе относятся к контроллерам серий FX1S и FX1N.

ОБЗОР ИНСТРУКЦИЙ FNC 150...159

Символ	FNC	Назначение	Разд
DSZR	150	Движение в базовую точку с использованием выключателя приближения	7.9.6
DVIT	151	Позиционирование с использованием прерывания	7.9.7
TBL	152	Позиционирование по таблице данных	7.9.8
DABS	155	Чтение абсолютной текущей позиции	7.9.9
ZRN	156	Перемещение к нулевой (исходной) точке	7.9.10
PLSV	157	Выдача импульсов переменной частоты	7.9.11
DRVI	158	Позиционирование по инкрементальным (приращиваемым) значениям	7.9.12
DRVA	159	Позиционирование по абсолютным значениям	7.9.13

Табл. 7-29: Обзор команд позиционирования

7.9.1 УКАЗАНИЯ ПО ПРИМЕНЕНИЮ КОМАНД ПОЗИЦИОНИРОВАНИЯ ДЛЯ КОНТРОЛЛЕРОВ FX1S И FX1N

Инструкции позиционирования ZRN (FNC 156), PLSV (FNC 157), DRVI (FNC 158) и DRVA (FNC 159) работают только с выходами Y0 и Y1.

Инструкции позиционирования могут использоваться в программе многократно.

- В одинаковых инструкциях нельзя использовать одинаковые выходы (Y0 и Y1). При использовании одинаковых выходов ПК воспринимает их, как двойной выход, и данная инструкция будет работать не корректно..
- Перед повторной активизацией инструкции убедитесь, что специальные меркеры M8147 для Y0 и M8148 для Y1 выключены.

Предупреждение при использовании инструкций импульсных выходов FNC 57 (PLSY) и FNC 59 (PLSR)

- Инструкции FNC 57 (PLSY) и FNC 59 (PLSR) работают только с выходами Y0 и Y1.
- Если инструкции позиционирования и инструкции импульсных выходов используют одновременно одинаковый выход, то ПК воспринимает это, как двойной выход, и программа работает не корректно.
- Если необходимо использовать функциональные возможности инструкции FNC 57 (PLSY) и FNC 59 (PLSR) и функциональные возможности инструкции FNC 158 (DRVI), то целесообразно использовать только инструкцию FNC 158 (DRVI).

Выхода.

Для работы инструкций позиционирования необходимо использовать ПК, только с транзисторными выходами.

Для более устойчивой работы на больших частотах рекомендуется нагрузка на транзисторный выход от 10 до 100 мА.

Технические данные транзисторных выходов Y0 и Y1 у контроллеров FX1S и FX1N		
Номинальная коммутируемая мощность	напряжение	5...24 В пост.
	ток	10...100 мА
Максимальная частота переключений		100 кГц

7.9.2 ФОРМАТ ВЫХОДНЫХ ИМПУЛЬСОВ

При реализации позиционирования с помощью выходов ПК используется формат сигналов “Последовательность импульсов + Знак”, как показано на рисунке.

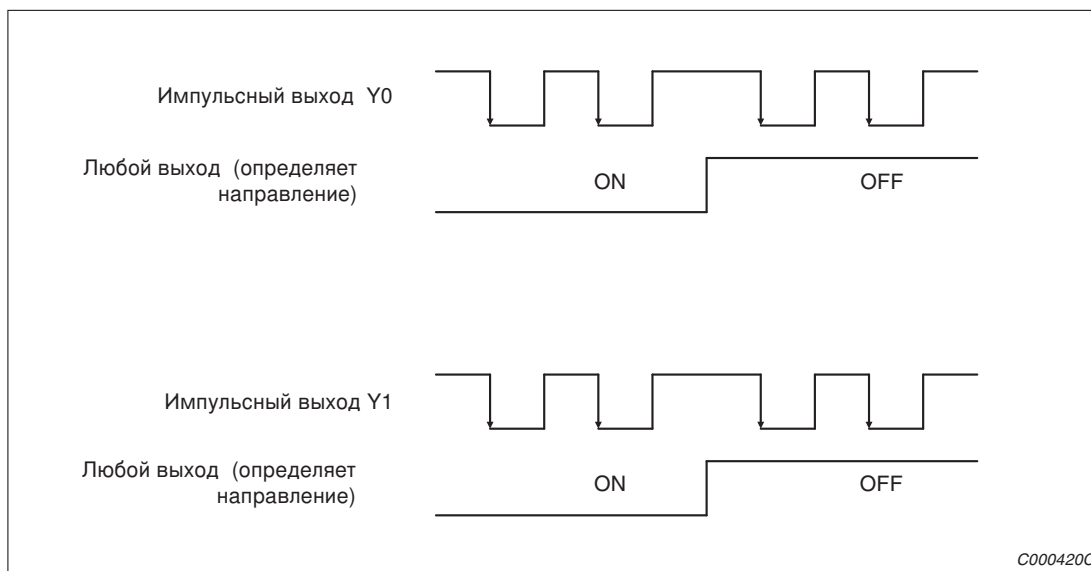


Рис. 7-128:Формат выходных сигналов

Обязательно установите в сервоусилителе входные данные:

Формат входа: Последовательность импульсов + знак

Логика: Негативная

7.9.3 ОПЕРАНДЫ ИСПОЛЬЗУЕМЫЕ ПРИ ПОЗИЦИОНИРОВАНИИ

Операнд	Доступ	Описание
M8145	Могут устанавливаться в программе	Impulsausgabe an Y0 sofort stoppen
M8146		Impulsausgabe an Y1 sofort stoppen
M8147	Только чтение	Заданные импульсы выданы Y0
M8148		Заданные импульсы выданы Y1

Табл. 7-30:Специальные меркеры для инструкций позиционирования

Операнд		Значение по умолчанию	Описание
D8140	Двойное слово	0	Регистр текущего значения Y0 (32бита) ①
D8141	Двойное слово		
D8142	Двойное слово	0	Регистр текущего значения Y1(32бита) ①
D8143	Двойное слово		
D8145		0	Приращение скорости при разгоне/торможении для инструкций FNC 158 и FNC 159 Устанавливается 1/10 или менее от максимальной частоты (D8146, D8147)
D8146	Двойное слово	100000	Максимальная скорость для инструкций FNC 158 и FNC 159 (32бита) Диапазон: от 100 Hz до 100 kHz
D8147	Двойное слово		
D8148		100	Время разгона/торможения до максимальной скорости для инструкций FNC156, FNC158 и FNC159 ② Диапазон: от 50 до 5000 мсек

Табл. 7-31: Специальные регистры для команд позиционирования в контроллерах FX1S и FX1N

- ① Для инструкций FNC 157 (PLSV), FNC 158 (DRVI) и FNC 159 (DRVA) текущее значение уменьшается или увеличивается в соответствии с направлением вращения.
Для инструкций FNC 57 (PLSY) и FNC 59 (PLSR) используются те же регистры, но в них представлено суммарное выданное число импульсов.
- ② Задается время разгона до максимальной частоты (D8146, D8147), с установленной величиной приращения скорости (D8145).
Задается время торможения с максимальной частоты (D8146, D8147) до нулевой скорости, с установленной величиной приращения скорости (D8145)

7.9.4 ПРИМЕР ПОДКЛЮЧЕНИЯ СЕРВОПРИВОДА

На рисунке показана схема подключения к ПК FX1S-30MT сервопривода MELSERVO MR-J2-mA:

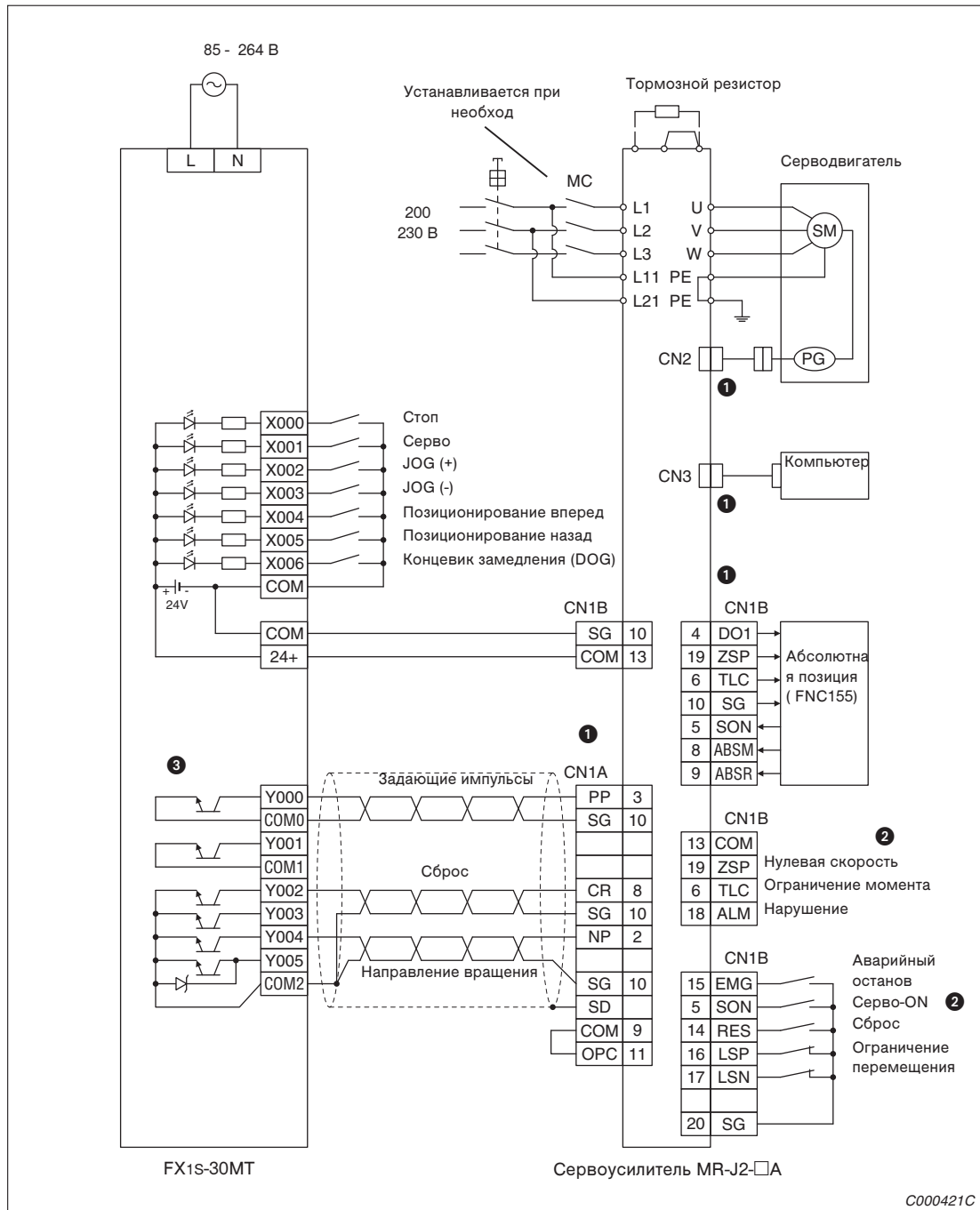


Рис. 7-129: Принципиальная схема.

- ① Разъемы CN1A, CN1B, CN2 und и CN3 имеют одинаковую форму, не перепутайте.
- ② Данные сигналы используются при абсолютном позиционировании.
- ③ Убедитесь в наличие соединения SPS с транзисторными выходами.

7.9.5 ПРИМЕР ПРОГРАММЫ

При работе, позиционирование выполняется абсолютным способом.

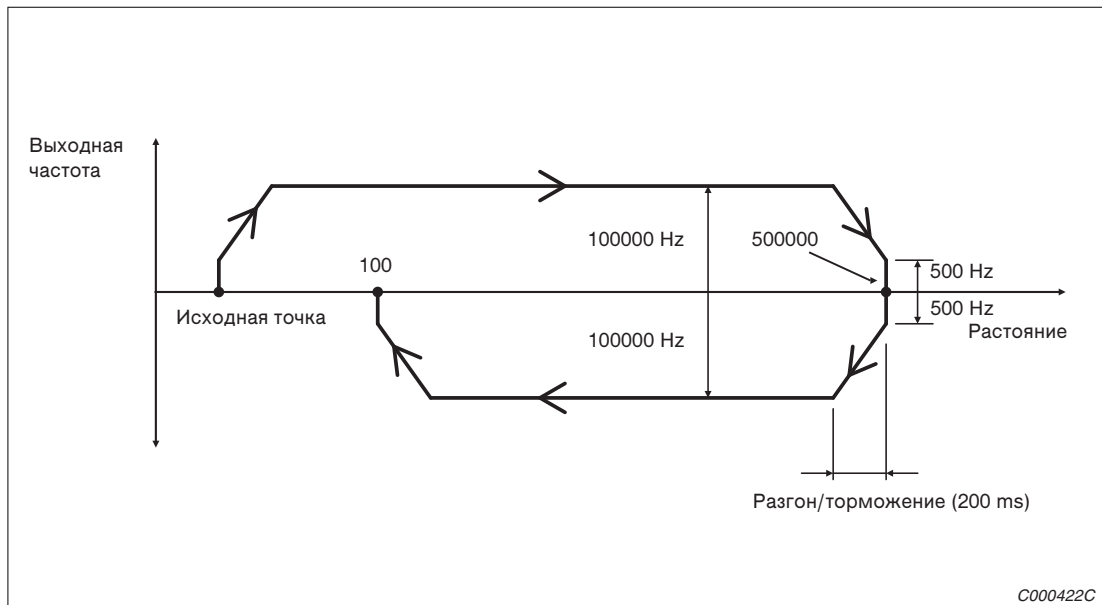


Рис. 7-130: Пример программы

Расчет минимальной частоты:

$$f_{\min} = \sqrt{\frac{f_{\max}}{2 \times t_B}}$$

f_{\max} - задается в регистрах D8146 и D8147 в герцах..

t_B - время разгона /торможения в секундах.

Пример: $f_{\max} = 100$ кГц и $t_B = 0,2$ сек, тогда минимальная частота -

$$f_{\min} = \sqrt{\frac{100000}{2 \times 0,2}} = 500 \text{ Hz}$$

SPS подключение показано в гл 7.6.4

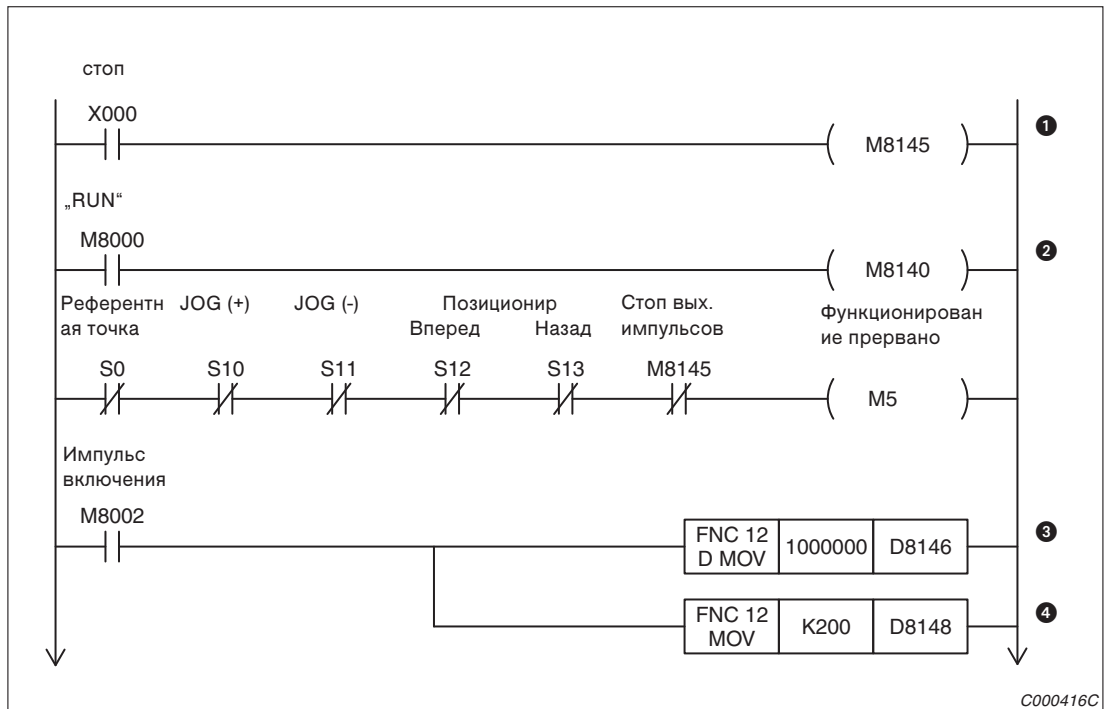


Рис. 7-131: Пример программы движения

- ① Останов выдачи импульсов оси X (Y0)
- ② Бит M8140 разрешение возврата в референтную точку
- ③ Установка максимальной скорости 100.000 Гц в регистры D8147 и D8146 .
- ④ Установка времени разгона/торможения 200 мсек в регистр D8148 .

УКАЗАНИЕ

Если максимальная скорость или время разгона/торможения не отличаются от заводских настроек, то регистры D8146 и D8148 не программируются.

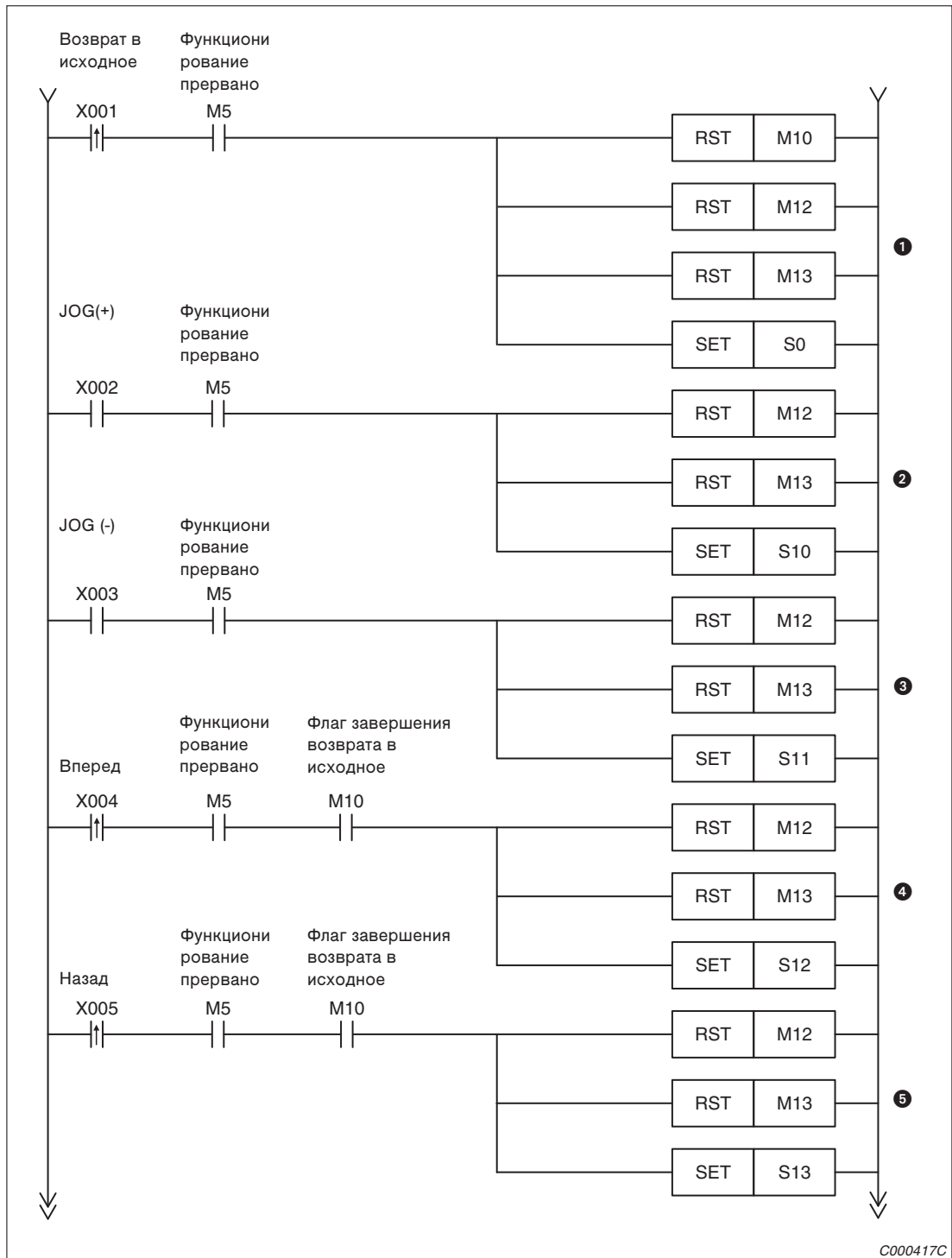


Рис. 7-132: Пример программы

- ① Сброс флага завершения возврата в исходное, Сброс флага завершения позиционирования, Сброс флага завершения позиционирования при реверсе, Установка флага состояния 'в исходном' (S0).
- ② Сброс флага завершения возврата в исходное, Сброс флага завершения позиционирования, Активизация режима jog (+), S10.
- ③ Сброс флага завершения возврата в исходное, Сброс флага завершения позиционирования, Активизация режима jog (-), S11.
- ④ Сброс флага завершения возврата в исходное, Сброс флага завершения позиционирования, Активизация режима позиционирования вперед.
- ⑤ Сброс флага завершения возврата в исходное, Сброс флага завершения позиционирования, Активизация режима позиционирования вперед.

УКАЗАНИЕ

Макс. величина перемещения в режиме JOG - 999,999 импульсов, т.к. данное значение является максимальным значением для функции FNC 158 (DRVI). Большие перемещения достигаются последовательным выполнением инструкций JOG.

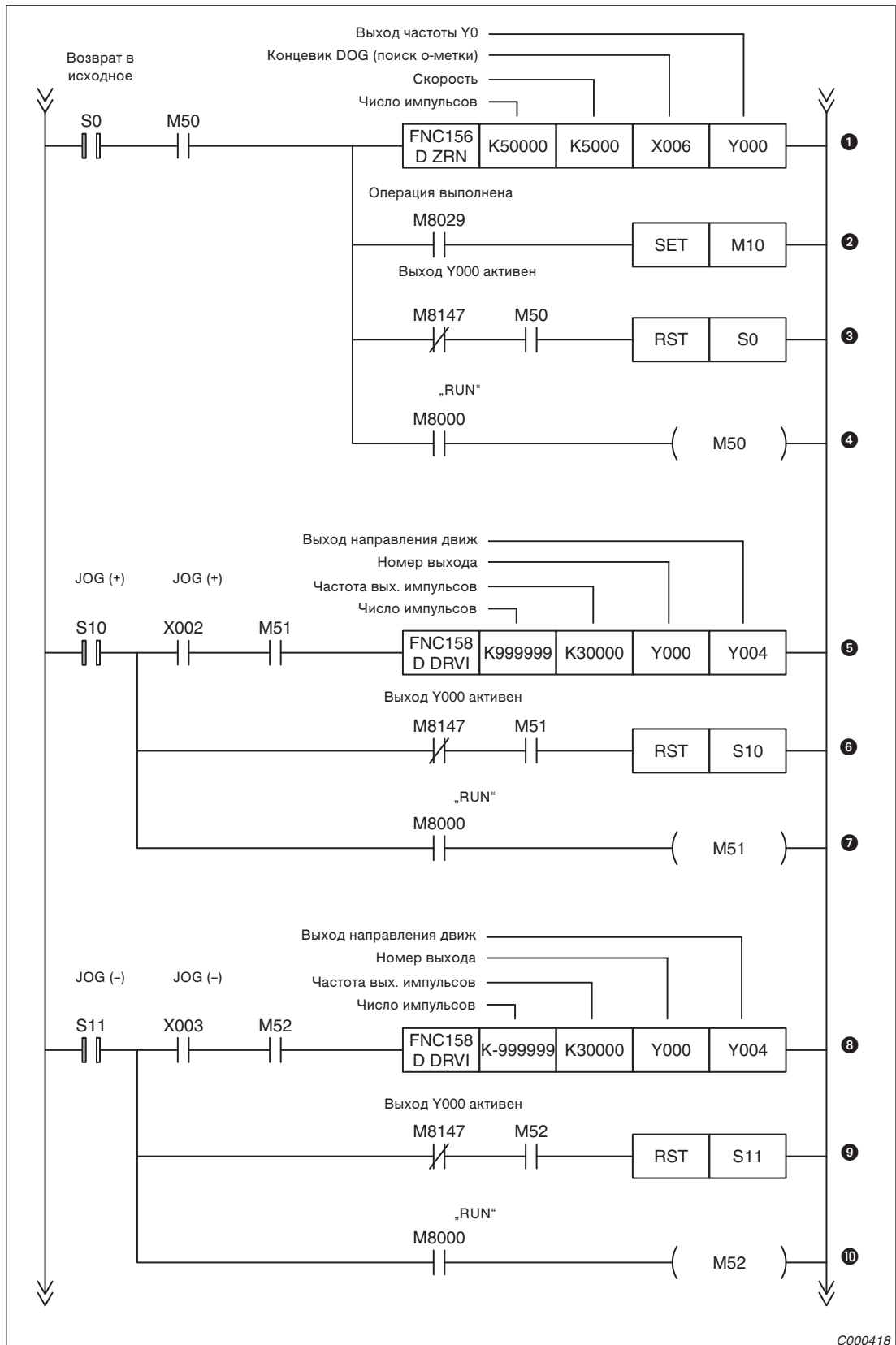


Рис. 7-133: Пример программы

- ① Инструкция возврата в исходное для реверсивного движения
- ② Флаг завершения возврата в исходное
- ③ Возврат в исходное завершен (самосброс).
- ④ Задержка выполнения инструкции на время одного цикла предотвращает одновременное выполнение инструкций позиционирования.
- ⑤ Выполнение режима JOG в направлении +, используя инструкцию инкрементального позиционирования (Y004 включен).
- ⑥ Режим JOG (+) завершен (самосброс).
- ⑦ Задержка выполнения инструкции на время одного цикла предотвращает одновременное выполнение инструкций позиционирования.
- ⑧ Выполнение режима JOG в направлении -, используя инструкцию инкрементального позиционирования (Y004 выключен).
- ⑨ Режим JOG (-) завершен (самосброс).
- ⑩ Задержка выполнения инструкции на время одного цикла предотвращает одновременное выполнение инструкций позиционирования.

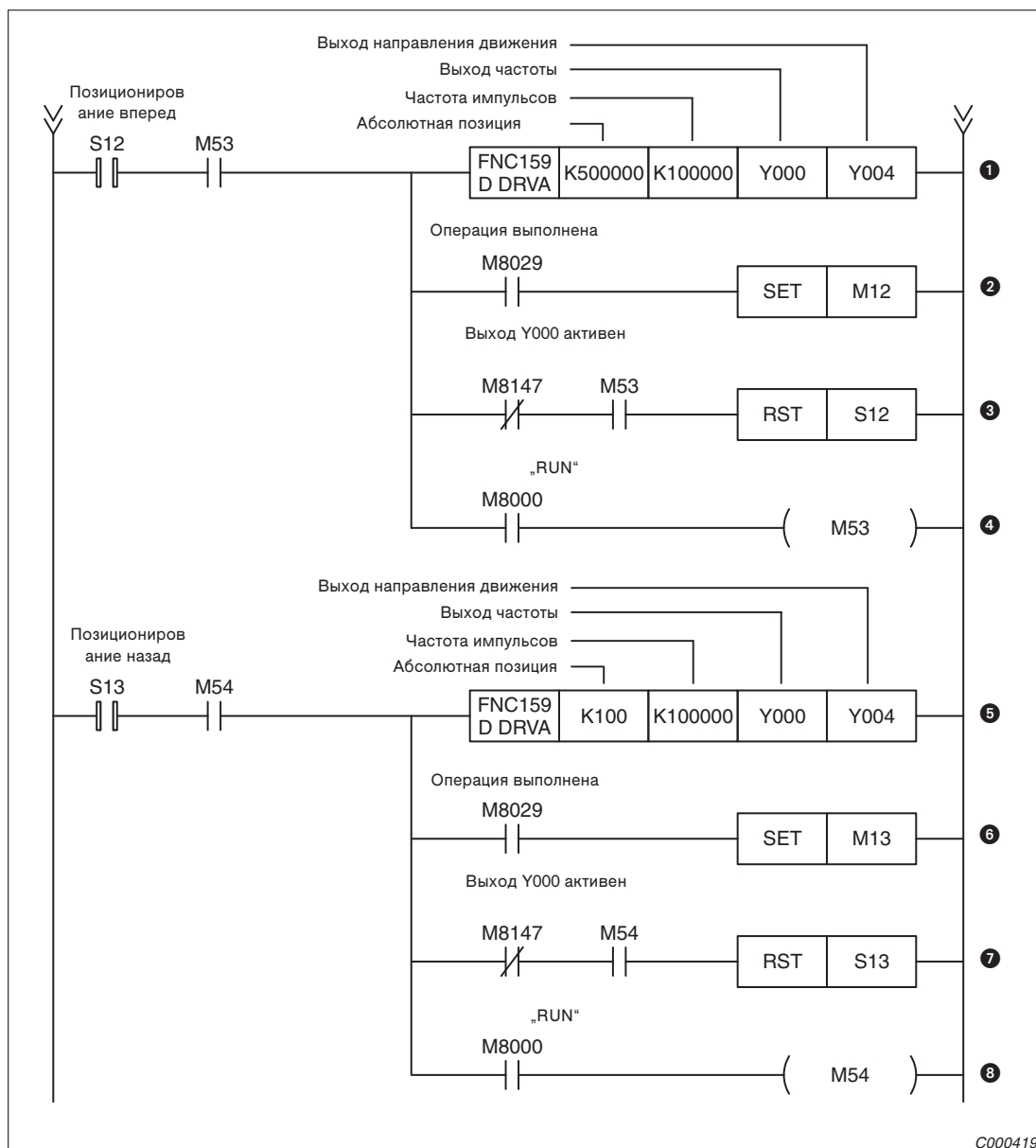


Рис. 7-134:Пример программы

- ① Перемещение в абсолютную позицию 500,000 используя инструкцию позиционирования в абс. координатах (Y004 включен).
- ② Флаг завершения позиционирования в направлении +.
- ③ Позиционирование в направлении + завершено (самосброс).
- ④ Задержка выполнения инструкции на время одного цикла предотвращает одновременное выполнение инструкций позиционирования.
- ⑤ Перемещение в абсолютную позицию 100 используя инструкцию позиционирования в абс. координатах (Y004 выключен).
- ⑥ Флаг завершения позиционирования в направлении -.
- ⑦ Позиционирование в направлении - завершено (самосброс).
- ⑧ Задержка выполнения инструкции на время одного цикла предотвращает одновременное выполнение инструкций позиционирования.

7.9.6 ДВИЖЕНИЕ В БАЗОВУЮ ТОЧКУ С ИСПОЛЬЗОВАНИЕМ ВЫКЛЮЧАТЕЛЯ ПОЗИЦИОНИРОВАНИЯ (DSZR)

					DSZR		FNC 150			
					Движение в базовую точку с использованием выключателя приближения					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
										●
Операнды	S1+	S2+	D1+	D2+	Имп. инструкция (P)	Обработка		Шаги программы		
	X, Y, M, T,	X [⊙]	Y [⊙]	Y [⊙] , M, T			16бита	32бита	DSZR	9
						●				

- ① Вход из области X0...X7
- ② Только транзисторные выходы Y0, Y1 или Y2 в базовом блоке или выходы Y0, Y1, Y2 или Y3 высокоскоростного модуля вывода FX3U-2HSY-ADP. Для указания Y2 или Y3 нужен второй модуль высокоскоростного вывода.
- ③ Если используются иные выходы кроме выходов высокоскоростного модуля вывода FX3U-2HSY-ADP, необходимо указать транзисторный выход.

Функция

С помощью этой команды запускается движение в базовую (нулевую) точку (называемое также “движение референцирования”). В дополнение к команде ZRN (раздел 7.9.10), можно с помощью команды DSZR искать выключатель приближения и определить базовую точку на основе этого выключателя и сигнала нулевой фазы. Однако сигнал нулевой фазы не поддается счету и поэтому нулевая точка не может быть определена.

Описание

- В (S1+) указывается операнд, содержащий состояние выключателя для переключения на ползучее движение вблизи базовой точки.
- (S2+) содержит вход, к которому подключен сигнал нулевой фазы.
- (D1+) указывает выход для вывода импульсов.

Для вывода импульсов можно использовать только определенные выходы (см. выше). Так как импульсы выдаются с высокой частотой, необходимо применять контроллеры с транзисторными выходами. Релейные контакты в этом случае преждевременно изнашиваются и поэтому не пригодны. У серии FX3U можно для первого модуля вывода FX3U-2HSY-ADP указать выходы Y0 и Y1, а для второго модуля вывода - выходы Y2 и Y3. Состояния Y4 и Y5 или Y6 и Y7 определяют направление вращения.

- В D2+ указывается операнд, задающий направление вращения для сервоусилителя.

УКАЗАНИЕ

Программа, содержащая команду DSZR, не должна изменяться в режиме RUN контроллера во время выполнения этой команды. При нарушении этого требования сервопривод замедляется и останавливается.

Позиционирование с помощью контроллера FX3U и применение команды DSZR подробно описаны в руководстве по позиционированию для контроллеров серии FX3U.

7.9.7 ПОЗИЦИОНИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ПРЕРЫВАНИЯ (DVIT)

					DVIT		FNC 151			
					Позиционирование с использованием прерывания					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
										●
Операнды	S1+	S2+	D1+	D2+	Имп. инструкция (P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U/G), V, Z		Y ^⓪	Y ^⓪ , M, S			16бита	32бита	DVIT	9
							●	●	DDVIT	17

- ① Только транзисторные выходы Y0, Y1 или Y2 в базовом блоке или выходы Y0, Y1, Y2 или Y3 высокоскоростного модуля вывода FX3U-2HSY-ADP. Для указания Y2 или Y3 нужен второй модуль высокоскоростного вывода.
- ② Если используются иные выходы кроме выходов высокоскоростного модуля вывода FX3U-2HSY-ADP, необходимо указать транзисторный выход.

Функция

С помощью этой команды после прерывания выполняется движение в позицию с постоянной скоростью.

Описание

- В (S1+) задается количество выдаваемых импульсов. В случае словного операнда диапазон составляет от -32767 до 32767 приращений, а в случае 32-битного операнда от -99999 до 999999 приращений. Значение "0" задавать нельзя.
- В (S2+) указывается частота выходных импульсов. В случае словных операндов диапазон составляет от 10 до 32767 Гц, а в случае 32-битных операндов, в сочетании с модулем высокоскоростного вывода, возможно выводить частоты от 10 Гц до 200 кГц.
- (D1+) указывает выход для вывода импульсов.

Для вывода импульсов можно использовать только определенные выходы (см. выше). Так как импульсы выдаются с высокой частотой, необходимо применять контроллеры с транзисторными выходами. Релейные контакты в этом случае преждевременно изнашиваются и поэтому не пригодны. У серии FX3U можно для первого модуля вывода FX3U-2HSY-ADP указать выходы Y0 и Y1, а для второго модуля вывода - выходы Y2 и Y3. Состояния Y4 и Y5 или Y6 и Y7 определяют направление вращения.

- В (D2+) указывается операнд, задающий направление вращения для сервоусилителя.

УКАЗАНИЕ

Программу, содержащую команду DVIT, не следует изменять в режиме RUN контроллера во время выполнения этой команды. При нарушении этого требования сервопривод замедляется и останавливается.

Позиционирование с использованием контроллера FX3U и применение команды DVIT подробно описаны в руководстве по позиционированию для контроллеров серии FX3U.

7.9.8 ПОЗИЦИОНИРОВАНИЕ ПО ТАБЛИЦЕ ДАННЫХ (TBL)

			TBL		FNC 152			
			Позиционирование по таблице данных					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	Y *	K, H		16бита	32бита	DVIT	9	
				●	●	DDVIT	17	

* Только транзисторные выходы Y0, Y1 или Y2 в базовом блоке или выходы Y0, Y1, Y2 или Y3 высокоскоростного модуля вывода FX3U-2HSY-ADP. Для указания Y2 или Y3 нужен второй модуль высокоскоростного вывода.

Функция

Команда TBL позиционирует по параметрам, сохраненным в контроллере в виде таблицы.

Описание

- (D+) указывает выход для вывода импульсов.

Для вывода импульсов можно использовать только определенные выходы (см. выше). Так как импульсы выдаются с высокой частотой, необходимо применять контроллеры с транзисторными выходами. Релейные контакты в этом случае преждевременно изнашиваются и поэтому не пригодны. У серии FX3U можно для первого модуля вывода FX3U-2HSY-ADP указать выходы Y0 и Y1, а для второго модуля вывода - выходы Y2 и Y3. Состояния Y4 и Y5 или Y6 и Y7 определяют направление вращения.

- (n) указывает номер таблицы (от 1 до 100).

УКАЗАНИЕ

Программа, содержащая команду TBL, не может быть изменена в режиме RUN контроллера.

Позиционирование с использованием контроллера FX3U и применение команды TBL подробно описаны в руководстве по позиционированию для контроллеров серии FX3U.

7.9.9 ЧТЕНИЕ АБСОЛЮТНОЙ ТЕКУЩЕЙ ПОЗИЦИИ (DABS)

				DABS		FNC 155		
				Чтение абсолютной текущей позиции				
CPU				FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●			●
Operanden	S+	D1+	D2+	Имп. инструкция (P)	Обработка		Шаги программы	
	X, Y, M, S	Y [○] , M, S	KnY, KnM, KnS, T, C, D, R, специальные модули (U/G), V, Z		16бита	32бита	DABS	13
						●		

- ① Только транзисторные выходы
- ② Только у FX3U

ФУНКЦИЯ

С помощью этой команды можно определить абсолютное положение подключенного сервоусилителя МИЦУБИСИ типа MR-H, MR-J2(S) или MR-J3. Сервопривод должен быть способен определять абсолютное положение.

ОПИСАНИЕ

- ABS-инструкция является 32-х битной инструкцией. Поэтому всегда указывайте ее как “DABS”.
- Чтение позиции начинается, если выполнено стартовое условие ABS-инструкции. При окончании чтения включается меркер M8029. Если во время чтения стартовые условия отключаются, процесс чтения прекращается.
- При проектировании нужно следить за тем, чтобы сервоусилителей включался раньше или одновременно с ПК.
- Абсолютная позиция заносится в операнд, который указан в D2+. Он может быть словным операндом. Однако позднее это значение должно передаваться в программе в специальные регистры D8141 и D8140.
- Стартовые условия ABS-инструкции должны быть включены всегда, также и после восприятия позиции. Если стартовые условия после прочтения позиции станут недействительными, отключится сигнал “Servo-ON” (SON) и привод остановится.
- Также, если сервопривод сможет воспринять абсолютную позицию, после включения привода должен выполняться возврат в исходное положение.

ПРИМЕР ▾

ABS-инструкции

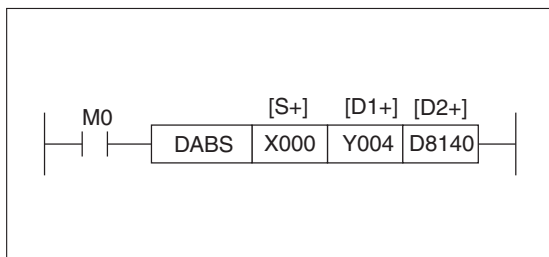


Рис. 7-135:
Пример программирования ABS-инструкции

C000400C



7.9.10 ПЕРЕМЕЩЕНИЕ К НУЛЕВОЙ (ИСХОДНОЙ) ТОЧКЕ (ZRN)

					ZRN		FNC 156			
					Перемещение к нулевой (исходной) точке					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
						●	●			●
Операнды	S1+	S2+	S3+	D	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U/G), V, Z,	X, Y, M, S	Y ^②			16бита	32бита	ZRN	9	
						●	●	DZRN	17	

① Только у FX3U

② У FX1S и FX1N: только Y0 или Y1;

У FX3U: только транзисторные выходы Y0, Y1 или Y2 в базовом блоке или выходы Y0, Y1, Y2 или Y3 высокоскоростного модуля вывода FX3U-2HSY-ADP. При указании Y2 или Y3 в этом случае нужен второй модуль высокоскоростного вывода.

ФУНКЦИЯ

С помощью этой инструкции можно управлять перемещением к нулевой (исходной, референтной) точке. Во время позиционирования в ПК запоминается текущая позиция и в зависимости от направления движения повышается или снижается по мере перемещения. Благодаря этому позиция машины в ПК всегда актуальна. При отключении напряжения ПК теряет это значение позиции. Поэтому после включения управления или для отладки машины выполняется возврат в исходное к определенной позиции.

ОПИСАНИЕ

- В (S1+) указывается скорость, с которой должно выполняться движение референцирования. В случае словных операндов возможна частота от 10 до 32767 Гц, а в случае 32-битных операндов диапазон составляет от 10 Гц до 100 кГц (или от 10 Гц до 200 кГц для контроллера FX3U и модуля высокоскоростного вывода).
- Ползучая (медленная) скорость от 10 до 32,767 Гц указывается в S2+.
- S3+ содержит переключатель для управления ползучей скоростью вблизи нулевой точки. Если в S3+ указан другой операнд, чем вход (X), то на временную точку торможения окажет влияние время цикла ПК (программы) и отклонение от нулевой точки может оказаться очень большим.
- Для вывода импульсов можно использовать только определенные выходы (см. выше). Так как импульсы выдаются с высокой частотой, необходимо применять контроллеры с транзисторными выходами. Релейные контакты в этом случае преждевременно изнашиваются и поэтому не пригодны. Для выработки выходных сигналов с крутым фронтом ток нагрузки транзисторных выходов должен быть не меньше 200 мА. Может понадобиться применение нагрузочных резисторов.
- Если включен специальный меркер M8014, то сервоусилитель сигнализирует, если достигается нулевая точка.

- Следующие операнды представляются в распоряжение для задач позиционирования:

Операнд	Описание
D8141(старшее слово) D8140 (младшее слово)	Текущее значение Y0 (32 бита)
D8143 (старшее слово) D8142 (младшее слово)	Текущее значение Y1 (32 бита)
M8145	Выдача импульса на Y0 - сразу останов
M8146	Выдача импульса на Y1 - сразу останов
M8147	Контроль выдачи импульса на Y0 (ШИНА/ЧТЕНИЕ)
M8148	Контроль выдачи импульса на Y1 (ШИНА/ЧТЕНИЕ)

Табл. 7-32: Операнд для позиционирования с использованием контроллеров серий FX1S и FX1N

Операнды для FX3U описаны в руководстве по позиционированию для контроллеров серии FX3U.

- Если используется сервоусилитель МИЦУБИСИ типа MR-H, MR-J2(S) или MR-J3, способный определять абсолютное положение, фактическое положение можно считать с помощью команды ABS (FNC155).

УКАЗАНИЕ

К нулевой точке нельзя подойти с любого направления. При перемещении в исходное всегда перемещаются назад в направлении нулевой точки.

При программировании стартовых условий обращайтесь внимание на временные процессы.

Избегайте изменять программу в режиме RUN контроллера, в то время как выполняется команда ZRN. Во время обновления программы, содержащей команду ZRN, вывод импульсов прерывается и в результате этого сервопривод останавливается.

ПРИМЕР ▾

ZRN-инструкции

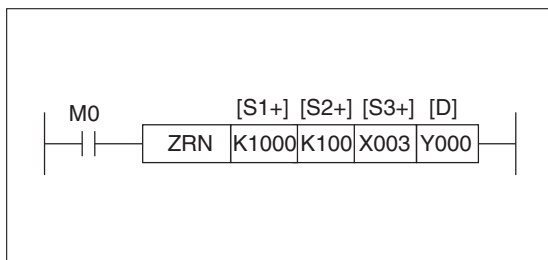


Рис. 7-136:

Пример программирования ZRN-инструкции

C000401C



7.9.11 ВЫДАЧА ИМПУЛЬСОВ С ПЕРЕМЕННОЙ ЧАСТОТОЙ (PLSV)

				PLSV		FNC 157				
				Выдача импульсов						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●			●	
Операнды	S+		D1	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U/G), V, Z Y		Y [®]	Y, M, S		16бита	32бита	PLSV	9	
						●	●	DPLSV	17	

① Только у FX3U

② У FX1S и FX1N: только Y0 или Y1;

У FX3U: только транзисторные выходы Y0, Y1 или Y2 в базовом блоке или выходы Y0, Y1, Y2 или Y3 высокоскоростного модуля вывода FX3U-2HSY-ADP. При указании Y2 или Y3 нужен второй модуль высокоскоростного вывода.

ФУНКЦИЯ

Вывод импульсов переменной частоты и сигнала направления вращения через выходы контроллера

ОПИСАНИЕ

- В (S+) указывается частота выходных импульсов. В случае словных операндов FX1S и FX1N диапазон составляет от 10 до 32767 Гц, а в случае 32-битных операндов возможно выводить частоты от 10 до 100 кГц.

В случае словных операндов FX3U возможна частота от -32768 до -1 и от 1 до 32767 Гц (кроме 0 Гц!), а в случае 32-битных операндов диапазон составляет от -200 кГц до -1 Гц и от 1 Гц до 200 кГц при использовании модуля высокоскоростного вывода.

- У контроллеров FX1S или FX1N для вывода импульсов в (D1) можно использовать только выходы Y0 и Y1. Так как импульсы выдаются с высокой частотой, необходимо применять контроллеры с транзисторными выходами. Релейные контакты в этом случае преждевременно изнашиваются и поэтому не пригодны. Для выработки выходных сигналов с крутым фронтом ток нагрузки транзисторных выходов должен быть не меньше 200 мА. Может понадобиться применение нагрузочных резисторов.

У серии FX3U можно для первого модуля вывода FX3U-2HSY-ADP указать выходы Y0 и Y1, а для второго модуля вывода - выходы Y2 и Y3. Состояния Y4 и Y5 или Y6 и Y7 определяют направление вращения.

- В D2+ указываются операнды, которые предварительно определяют для сервоусилителя направление вращения. Если операнд не включен, выбрано левое вращение, при включенном операнде - правое вращение.
- Направление вращения определяется знаком числа задаваемой в S+ частоты. Положительное значение означает правое вращение, при отрицательном значении привод работает с левым вращением.
- Во время выдачи импульсов может изменяться частота импульсов (S+).
- В начале и окончании выдачи импульсов нельзя осуществить, ни наклон ускорения, ни замедления. Если это требуется, то с помощью RAMP-инструкции (FNC 67) изменяется частота выхода.
- Если стартовые условия PLSV-инструкции во время выдачи импульсов больше не выполняются, то привод останавливается, даже без отработки замедления.

- После отключения стартовых условий PLSV-инструкция не запускается вновь до тех пор, пока включены специальные меркеры M8147 (контроль Y0) и M8148 (контроль Y1).
- Для задач позиционирования имеются следующие операнды:

Операнд	Описание
D8141 (старшее слово) D8140 (младшее слово)	Текущее значение Y0 (32 бита)
D8143 (старшее слово) D8142 (младшее слово)	Текущее значение Y1 (32 бита)
M8145	Выдача импульса на Y0 - сразу останов
M8146	Выдача импульса на Y1 - сразу останов
M8147	Контроль выдачи импульса на Y0 (ШИНА/ЧТЕНИЕ)
M8148	Контроль выдачи импульса на Y1 (ШИНА/ЧТЕНИЕ)

Табл. 7-33: Операнды для позиционирования с использованием контроллеров серий FX1S и FX1N

Операнды для FX3U описаны в руководстве по позиционированию для контроллеров серии FX3U.

УКАЗАНИЕ

При программировании стартовых условий обратите внимание на временные процессы.

Избегайте изменения программы в режиме RUN контроллера, в то время как выполняется команда PLSV. Программа, содержащая команду PLSV, во время обновления ведет себя следующим образом: при разгоне и замедлении происходит замедление и вывод импульсов прекращается. Если ни разгон, ни замедление не происходит, вывод импульсов прекращается сразу.

ПРИМЕР ▾

PLSV-инструкция

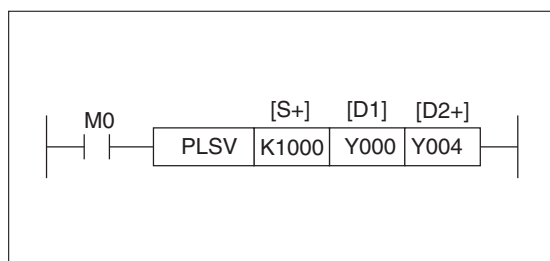


Рис. 7-137:

Пример программирования PLSV-инструкции

C000402C



7.9.12 ПОЗИЦИОНИРОВАНИЕ ПО ИНКРЕМЕНТАЛЬНЫМ ЗНАЧЕНИЯМ (DRVI)

					DRVI		FNC 158			
					Позиционирование по инкрементальным значениям					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
						●	●			●
Operanden	S1+	S2+	D1	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R [Ⓢ] , специальные модули (U□/G□) [Ⓢ] , V, Z		Y [Ⓢ]	Y, M, S		16бита	32бита	DRVI	9	
						●	●	DDRVI	17	

① Только у FX3U

② У FX1S и FX1N: только Y0 или Y1;

У FX3U: только транзисторные выходы Y0, Y1 или Y2 в базовом блоке или выходы Y0, Y1, Y2 или Y3 высокоскоростного модуля вывода FX3U-2HSY-ADP. При указании Y2 или Y3 нужен второй модуль высокоскоростного вывода.

ФУНКЦИЯ

Позиция, которая задается по инкрементальным значениям, достигается на постоянной скорости

ОПИСАНИЕ

- В S1+ указывается число приращения (инкременты). Область значений при словном операнде охватывает -32767...32767 приращений, а при 32-х битном операнде -999999...999999 приращений.
- В (S2+) указывается частота выходных импульсов. У словных операндов диапазон составляет от 10 до 32767 Гц, у 32-битных операндов возможен вывод частоты от 10 Гц до 100 кГц (у FX3U в сочетании с модулем высокоскоростного вывода - от 10 Гц до 200 кГц).
- У контроллеров FX1S или FX1N для вывода импульсов в (D1) можно использовать только выходы Y0 и Y1. Так как импульсы выдаются с высокой частотой, необходимо применять контроллеры с транзисторными выходами. Релейные контакты в этом случае преждевременно изнашиваются и поэтому не пригодны. Для выработки выходных сигналов с крутым фронтом ток нагрузки транзисторных выходов должен быть не меньше 200 мА. Может понадобиться применение нагрузочных резисторов.

У серии FX3U можно для первого модуля вывода FX3U-2HSY-ADP указать выходы Y0 и Y1, а для второго модуля вывода - выходы Y2 и Y3. Состояния Y4 и Y5 или Y6 и Y7 определяют направление вращения.

- В D2+ указываются операнды, которые предварительно определяют для сервоусилителя направление вращения. Если операнд не включен, выбрано левое вращение, при включенном операнде - правое вращение.
- При позиционировании по инкрементальным значениям направление определяется знаком числа приращения. Положительное значение означает правое вращение, при отрицательном значении привод работает с левым вращением.
- Если во время выполнения инструкции изменяется содержание операнда, то это не влияет на позиционирование. Измененный операнд будет принят во внимание при следующем выполнении инструкции.
- Если стартовые условия DRVI-инструкции во время выполнения инструкции больше не выполняются, то замедления обрабатывается и привод останавливается. В этом случае специальный меркер M8029, который показывает конец обработки, не включается.

- В контроллерах FX1S или FX1N после сброса начального условия команду DRVI можно снова запустить лишь в том случае, если специальные маркеры M8147 (контроль Y0) и M8148 (контроль Y1) более не установлены.

Условия для контроллера FX3U описаны в руководстве по позиционированию для контроллеров серии FX3U.

- Минимальную выходную частоту можно рассчитать по следующей формуле:

$$f_{\min} = \sqrt{\frac{f_{\max}}{2 \times t_b}}$$

f_{\max} : максимальная частота выходных импульсов (заданная в D8146/D8147 у FX1S и FX1N)

t_b : время ускорения и замедления в секундах

- Для задач позиционирования имеются следующие операнды:

Операнд	Описание
D8145	Смещение (оффсет) частоты вращения при применении FNC 158 или FNC 159
D8147 (старшее слово) D8146 (младшее слово)	Максимальная частота выходных импульсов при применении FNC 158 или FNC 159. Область: от 100 Гц до 100000 Гц
D8148	Время ускорения и замедления (мс) при применении FNC 158 или FNC 159
M8145	Выдача импульса на Y0 - сразу останов
M8146	Выдача импульса на Y1 - сразу останов
M8147	Контроль выдачи импульса на Y0 (ШИНА/ЧТЕНИЕ)
M8148	Контроль выдачи импульса на Y1 (ШИНА/ЧТЕНИЕ)

Табл. 7-34: Операнды для позиционирования с использованием контроллеров серий FX1S и FX1N

Операнды для FX3U описаны в руководстве по позиционированию для контроллеров серии FX3U.

УКАЗАНИЕ

При программировании стартовых условий следите за временными процессами.

Программу, содержащую команду DRVI, не следует изменять в режиме RUN контроллера, в то время как выполняется эта команда. При нарушении этого требования сервопривод замедляется и останавливается.

ПРИМЕР ▾

DRVI-инструкции

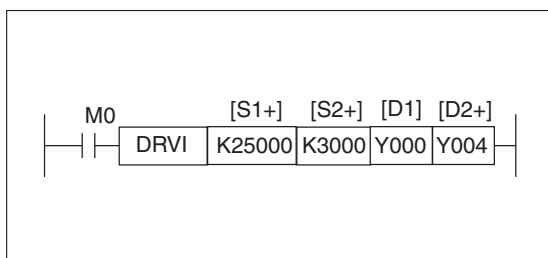


Рис. 7-138:

Пример программирования DRVI-инструкции

C000403C



7.9.13 ПОЗИЦИОНИРОВАНИЕ ПО АБСОЛЮТНЫМ ЗНАЧЕНИЯМ (DRVA)

					DRVA		FNC 159			
					Позиционирование по абсолютным значениям					CPU
Операнды	S1+	S2+	D1	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R [Ⓞ] , специальные модули (U□/G□) [Ⓞ] , V, Z		Y [Ⓞ]	Y, M, S		16бита	32бита	DRVA	9	
						●	●	DDRVA	17	

① Только у FX3U

② У FX1S и FX1N: только Y0 или Y1;

У FX3U: только транзисторные выходы Y0, Y1 или Y2 в базовом блоке или выходы Y0, Y1, Y2 или Y3 высокоскоростного модуля вывода FX3U-2HSY-ADP. При указании Y2 или Y3 нужен второй модуль высокоскоростного вывода.

ФУНКЦИЯ

Позиция, которая задается по абсолютным значениям, достигается на постоянной скорости. При таком виде позиционирования определяется нулевая (исходная) точка. Текущая действительная позиция измеряется как абсолютное значение по отношению к этой нулевой точке.

ОПИСАНИЕ

- В S1+ указывается абсолютная позиция. Область значений при словном операнде охватывает -32767...32767, а при 32-х битном операнде -999999...999999.
- В (S2+) указывается частота выходных импульсов. У словных операндов диапазон составляет от 10 до 32767 Гц, у 32-битных операндов возможен вывод частоты от 10 Гц до 100 кГц (у FX3U в сочетании с модулем высокоскоростного вывода - от 10 Гц до 200 кГц).
- У контроллеров FX1S или FX1N для вывода импульсов в (D1) можно использовать только выходы Y0 и Y1. Так как импульсы выдаются с высокой частотой, необходимо применять контроллеры с транзисторными выходами. Релейные контакты в этом случае преждевременно изнашиваются и поэтому не пригодны. Для выработки выходных сигналов с крутым фронтом ток нагрузки транзисторных выходов должен быть не меньше 200 мА. Может понадобиться применение нагрузочных резисторов.

У серии FX3U можно для первого модуля вывода FX3U-2HSY-ADP указать выходы Y0 и Y1, а для второго модуля вывода - выходы Y2 и Y3. Состояния Y4 и Y5 или Y6 и Y7 определяют направление вращения.

- В D2+ указываются операнды, которые предварительно определяют для сервоусилителя направление вращения. Если операнд не включен, выбрано левое вращение, при включенном операнде - правое вращение.
- При позиционировании по абсолютным значениям в качестве цели указывается расстояние от нулевой точки.
- Если во время выполнения инструкции изменяется содержание операнда, то это не влияет на позиционирование. Измененные операнды будут приняты во внимание при следующем выполнении инструкции.
- Если стартовые условия DRVA-инструкции во время выполнения инструкции больше не выполняются, то замедление обрабатывается и привод останавливается. В этом случае специальный меркер M8029, который показывает конец обработки, не включается.

- В контроллерах FX1S или FX1N после сброса начального условия команду DRVI можно снова запустить лишь в том случае, если специальные маркеры M8147 (контроль Y0) и M8148 (контроль Y1) более не установлены.

Условия для контроллера FX3U описаны в руководстве по позиционированию для контроллеров серии FX3U.

- Минимальную выходную частоту можно рассчитать по следующей формуле:

$$f_{\min} = \sqrt{\frac{f_{\max}}{2 \times t_b}}$$

f_{\max} : максимальная частота выходных импульсов (заданная в D8146/D8147 у FX1S и FX1N).
 t_b время ускорения и замедления в секундах

- Следующие операнды представляются в распоряжение для задач позиционирования:

Операнд	Описание
D8140 D8141	Текущее значение Y0 (32 Бит)
D8142 D8143	Текущее значение Y1 (32 Бит)
D8145	Смещение (оффсет) частоты вращения при применении FNC 158 или FNC 159
D8147 (старшее слово) D8146 (младшее слово)	Максимальная частота выходных импульсов при применении FNC 158 или FNC 159. Область: от 100 Гц до 100000 Гц
D8148	Время ускорения и замедления (мс) при применении FNC 158 или FNC 159
M8145	Выдача импульса на Y0 - сразу останов
M8146	Выдача импульса на Y1 - сразу останов
M8147	Контроль выдачи импульса на Y0 (ШИНА/ЧТЕНИЕ)
M8148	Контроль выдачи импульса на Y1 (ШИНА/ЧТЕНИЕ)

Табл. 7-35: Операнды для позиционирования с использованием контроллеров серий FX1S и FX1N

Операнды для FX3U описаны в руководстве по позиционированию для контроллеров серии FX3U.

УКАЗАНИЕ

При программировании стартовых условий следите за временными процессами.

Программу, содержащую команду DRVA, не следует изменять в режиме RUN контроллера, в то время как выполняется эта команда. При нарушении этого требования сервопривод замедляется и останавливается.

ПРИМЕР ▾

DRVA-инструкция

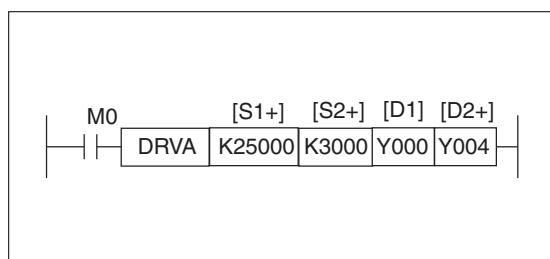


Рис. 7-139

Пример программирования DRVA-инструкции

C000404C



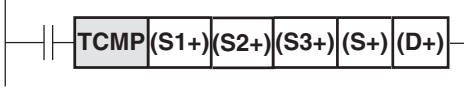
7.10 ИНСТРУКЦИИ РЕАЛЬНОГО ВРЕМЕНИ

ОБЗОР ИНСТРУКЦИЙ FNC 160...167

Символ	FNC	Назначение	Abschnitt
TCMP	160	Сравнение данных времени	7.10.1
TZCP	161	Сравнение данных времени в области	7.10.2
TADD	162	Сложение данных времени	7.10.3
TSUB	163	Вычитание данных времени	7.10.4
HTOS	164	Преобразование времени в формате "часы, минуты, секунды" в секунды	7.10.5
STON	165	Преобразование времени в секундах в формат "часы, минуты, секунды"	7.10.6
TRD	166	Чтение данных времени	7.10.7
TRW	167	Запись данных времени	7.10.8
HOUR	169	Счет рабочих часов	7.10.9

Табл. 7-23: Обзор инструкции FNC 160...167

7.10.1 СРАВНЕНИЕ ДАННЫХ ВРЕМЕНИ (TCMP)

						TCMP		FNC 160								
						Сравнение данных времени					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
Операнды	S1+	S2+	S3+	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы							
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, специальные модули (U□/G□)*, V, Z			K, H, T, C, D, R*, специальные модули (U□/G□)*	Y, M, S		●	16бита	32бита	TCMP	11					
				Соответственно применяется 3 следующих друг за другом адреса операндов			●		TCMP	11						

* Только у FX3U

ФУНКЦИЯ

Сравнение данных времени с выдачей результата сравнения

ОПИСАНИЕ

- Данные времени часы (S1+), минуты (S2+) и секунды (S3+) сравниваются с временем данных, записанном в (S+) до ((S+)+2).
- Результат сравнения записывается в 3 следующих один за другим битовых операнда.
- Если данные времени в (S+) до ((S+)+2) меньше данных времени в (S1+) до (S3+), включается битовый операнд (D+).
- Если данные времени в (S+) до ((S+)+2) равны данным времени в (S1+) до (S3+), включается битовый операнд ((D+)+1).
- Если данные времени в (S+) до ((S+)+2) больше данных времени в (S1+) до (S3+), включается битовый операнд ((D+)+2).

УКАЗАНИЕ

Сработавший выходной операнд остается после отключения условий выполнения TCMP-инструкции включенным.

В операндах (S1+) и (S+) могут указываться значения от 0 до 23 (часов).

В операндах (S2+) и ((S+)+1) могут указываться значения от 0 до 59 (минут).

В операндах (S3+) и ((S+)+2) могут указываться значения от 0 до 59 (секунд).

Для сравнения текущих данных встроенных часов можно выполнить команду TRD, а затем оценить содержимое указанных там целевых регистров. Возможен и альтернативный вариант - использование содержимых специальных регистров D8015 (часов), D8014 (минут) и D8013 (секунд).

ПРИМЕР ▾

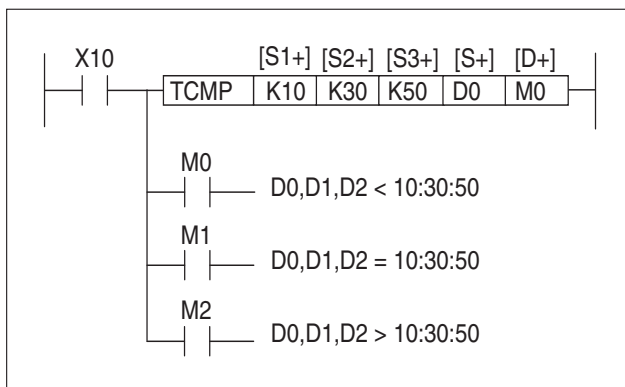


Рис. 7-140:

Пример программирования
TCMR-инструкции

C000364C

При включении входа X10 сравниваются заданные по K10, K30 и K50 соответственно 10 часов, 30 минут и 50 секунд с данными времени в D0 до D2.

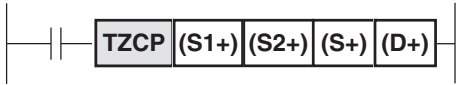
Если значение в D0 до D2 меньше чем значение 10:30:50, включается меркер M0.

Если значение в D0 до D2 равно значению 10:30:50, включается меркер M1.

Если значение в D0 до D2 больше чем значение 10:30:50, включается меркер M2.

△

7.10.2 СРАВНЕНИЕ ДАННЫХ ВРЕМЕНИ В ОБЛАСТИ (TZCP)

					TZCP		FNC 161			
					Сравнение данных времени в области					
CPU					FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	●	●	
Операнды	S1+	S2+	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R*, специальные модули (U□/G□)*, V, Z ((S1+) ≤ (S2+))			Y, M, S		●	16бита	32бита	TZCP	9
	Соответственно применяется 3 следующих друг за другом адреса операндов						●	●		TZCPP

* Только у FX3U

ФУНКЦИЯ

Сравнение данных времени в области с выдачей результата сравнения

ОПИСАНИЕ

- Данные времени в (S+) до ((S+)+2) и сравниваются с временем данных в области между ((S1+)+2) и ((S2+)+2).
- Результат сравнения записывается в 3 следующих один за другим битовых операнда.
- Если данные времени в (S+) до ((S+)+2) меньше данных времени в (S1+) до ((S1+)+2), включается битовый операнд (D+).
- Если данные времени в (S+) до ((S+)+2) лежат в области между (S1+) до ((S1+)+2) и (S2+) до ((S2+)+2), включается битовый операнд ((D+)+1).
- Если данные времени в (S+) до ((S+)+2) больше данных времени в (S2+) до ((S2+)+2), включается битовый операнд ((D+)+2).

УКАЗАНИЕ

Сработавший выходной операнд остается после отключения условий выполнения TZCP-инструкции включенным.

В операндах (S1+), (S2+) и (S+) могут указываться значения от 0 до 23 (часов).

В операндах ((S1+)+1), ((S2+)+1) и ((S+)+1) могут указываться значения от 0 до 59 (минут).

В операндах ((S1+)+2), ((S2+)+2) и ((S+)+2) могут указываться значения от 0 до 59 (секунд).

Для сравнения текущих данных встроенных часов можно выполнить команду TRD, а затем оценить содержимое указанных там целевых регистров. Возможен и альтернативный вариант - использование содержимых специальных регистров D8015 (часов), D8014 (минут) и D8013 (секунд).

ПРИМЕР ▾

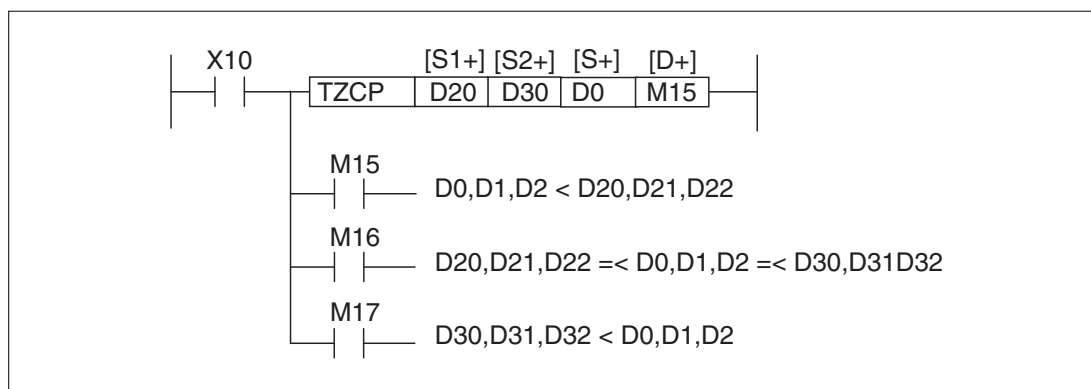


Рис. 7-141:Пример программы с командой TZCP

При включении входа X10 сравниваются данные времени D0 до D2 с областью данных времени между D20 до D22 до D30 и D32.

Если данные времени в D0 до D2 меньше данных времени в D20 до D22, включается меркер M15.

Если данные времени в D0 до D2 лежат в области между данными времени D20 до D22 и D30 до D32, включается меркер M16.

Если данные времени в D0 до D2 больше данных времени в D30 до D32, включается меркер M17.

△

7.10.3 СЛОЖЕНИЕ ДАННЫХ ВРЕМЕНИ (TADD)

				TADD		FNC 162				
				Сложение данных времени						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	●	●	
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы			
	T, C, D, R* Используются по 3 следующих друг за другом адреса операндов.				●	16бита	32бита	TADD	7	
					●		TADDP	7		

* Только у FX3U

ФУНКЦИЯ

Сложение данных времени и сохранение результата

ОПИСАНИЕ

- Данные времени в (S1+) до ((S1+)+2) суммируются с данными времени в (S2+) до ((S2+)+2).
- Расчет выполняется безошибочно относительно перехода времени (секунды-минуты и минуты-часы).

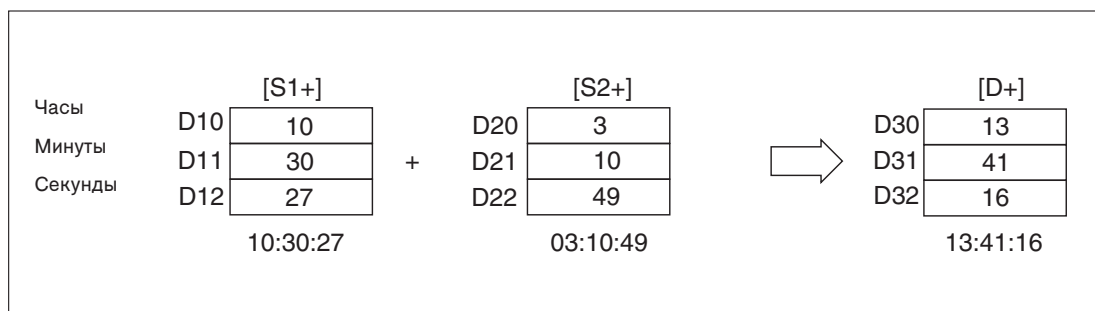


Рис. 7-142: Сложение данных времени

- Если результат сложения больше, чем 24 часа, выполняется переключение на 0 часов (“следующий день”), и включается флаг передачи (carry) M8022.

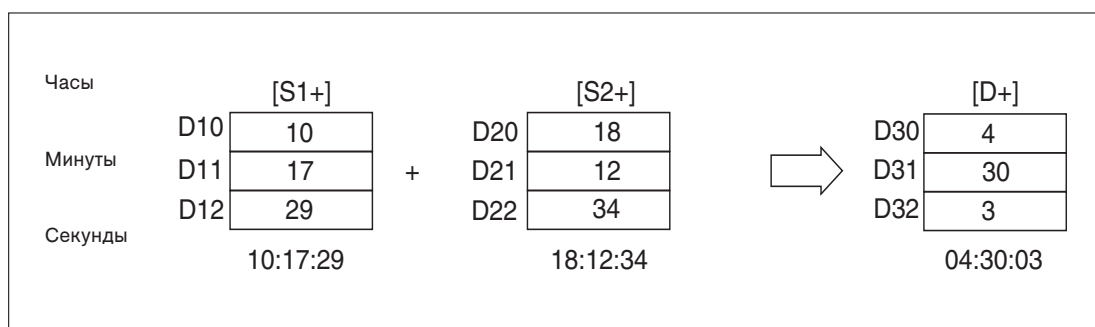


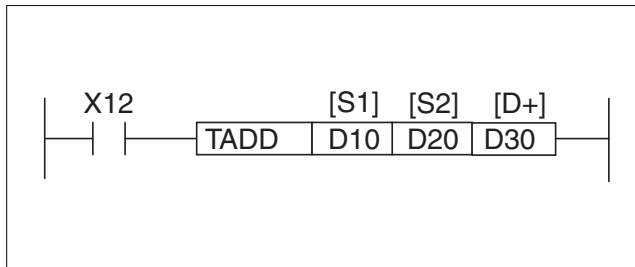
Рис. 7-143: Сложение данных времени с переходом часов

- Если результат сложения 0 (00:00:00, 0 часов, 0 минут, 0 секунд), то включается флаг нуля (zero) 8020.

- В качестве источника и цели может применяться один и тот же операнд ((S1+) до ((S1+)+2), (S2+) до ((S2+)+2). В этом случае расчетный результат снова запишется в операнд источника и может использоваться для следующих расчетов. Этот процесс повторяется в каждом цикле. Чтобы гарантировать одноразовое выполнение, применяется импульсная инструкция или блокировки.

УКАЗАНИЕ

Для вычислительной операции с текущими данными встроенных часов можно выполнить команду TRD, а затем оценить содержимое указанных там целевых регистров. Возможен и альтернативный вариант - использование содержимых специальных регистров D8015 (часов), D8014 (минут) и D8013 (секунд).

ПРИМЕР ▾**Рис. 7-144:**

Пример программирования TADD-инструкции

C000367C

При включении входа X12 к данным времени в D10 до D12 прибавляются данные времени из D20 до D22. Результат сохраняется в D30 до D32.

△

7.10.4 ВЫЧИТАНИЕ ДАННЫХ ВРЕМЕНИ (TSUB)

				TSUB		FNC 163				
				Вычитание данных времени						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●	●	●	●	
Operanden	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы			
	T, C, D, R* Используются по 3 следующих друг за другом адреса операндов.				●	16бита	32бита	TSUB	7	
					●		TSUBP	7		

* Только у FX3U

ФУНКЦИЯ

Вычитание данных времени и сохранение результата

ОПИСАНИЕ

- Данные времени в (S2+) до ((S2+)+2) вычитаются из данных времени в (S1+) до ((S1+)+2).
- Расчет выполняется безошибочно относительно перехода времени (секунды-минуты и минуты-часы)

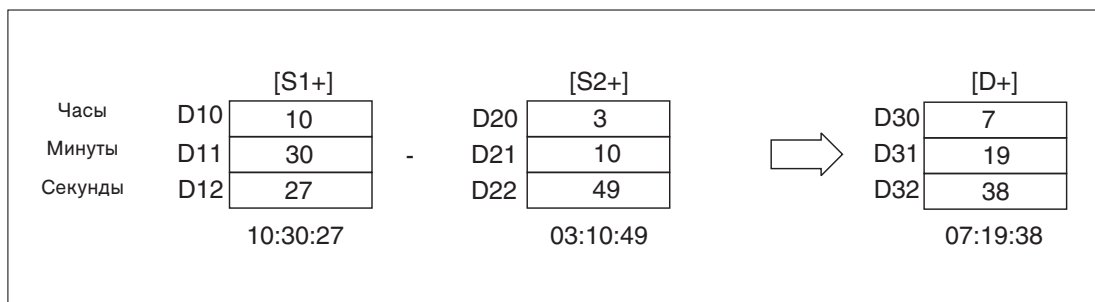


Рис. 7-145: Вычитание данных времени

- Если результат вычитания меньше, чем 0 часов (00:00:00), то остаток от 24 часов заимствуется (“предыдущий день”), и включается флаг заимствования (borrow) M8021.

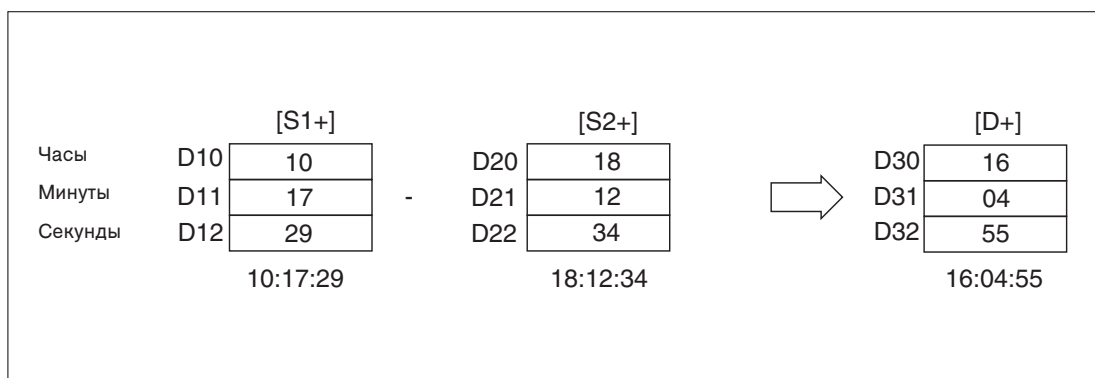


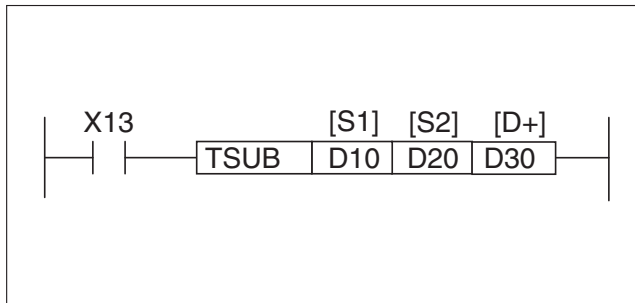
Рис. 7-146: Вычитание данных времени с заимствованием часов

- Если результат вычитания 0 (00:00:00, 0 часов, 0 минут, 0 секунд), то включается флаг нуля (zero) 8020.

- В качестве источника и цели может применяться один и тот же операнд ((S1+) до ((S1+)+2), (S2+) до ((S2+)+2). В этом случае расчетный результат снова запишется в операнд источника и может использоваться для следующих расчетов. Этот процесс повторяется в каждом цикле. Чтобы гарантировать одноразовое выполнение, применяется импульсная инструкция или блокировки.

УКАЗАНИЕ

Для вычислительной операции с текущими данными встроенных часов можно выполнить команду TRD, а затем оценить содержимое указанных там целевых регистров. Возможен и альтернативный вариант - использование содержимых специальных регистров D8015 (часов), D8014 (минут) и D8013 (секунд).

ПРИМЕР ▾**Рис. 7-147:**

Пример программирования TSUB-инструкции

C000370C

При включении входа X13 данные времени в D20 до D22 вычитаются из данных времени в D10 до D13. Результат сохраняется в D30 до D32.

△

7.10.5 ПРЕОБРАЗОВАНИЕ ЕДИНИЦЫ “ЧАСЫ” В “СЕКУНДЫ” (HTOS)

		HTOS		FNC 164			
		Преобразование единицы “часы” в “секунды”					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Operanden	S1+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□) Используются 3 следующих друг за другом адреса операндов.	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)		●	16бита	32бита	TSHTOS HTOSPUB
				●		DHTOS DHTOSP	9

Функция

Пересчет времени в другие единицы

Описание (16-битная обработка)

- С помощью команды HTOS значение времени в регистрах с (S+) по ((S+)+2) (час, минута и секунда) преобразуется в значение времени в секундах. Результат сохраняется в (D+).

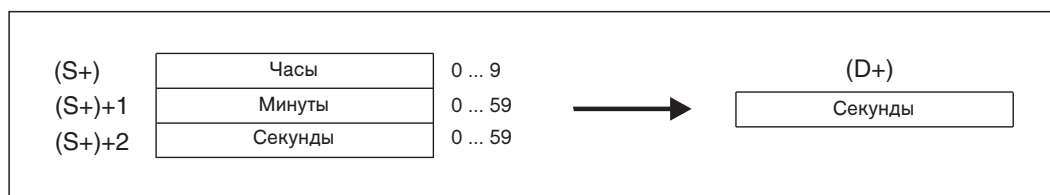


Рис. 7-148: Принцип действия команды HTOS

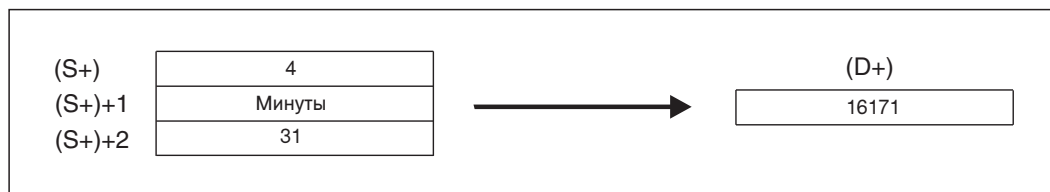


Рис. 7-149: Пример пересчета времени: 4 часа, 29 минут и 31 секунда соответствуют 16171 секундам.

Описание (32-битная обработка)

- С помощью команды DHTOS значение времени в регистрах с (S+) по ((S+)+2) (час, минута и секунда) преобразуется в значение времени в секундах. Результат сохраняется в ((D+)+1) и (D+).

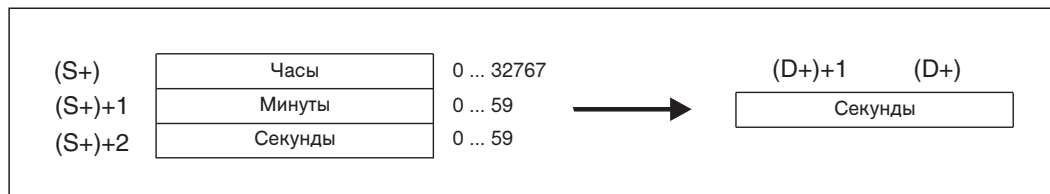


Рис. 7-150: Принцип действия команды DHTOS

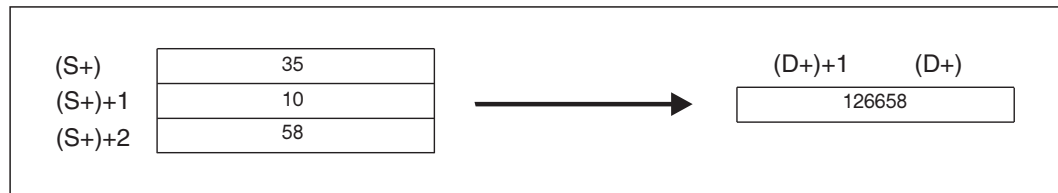


Рис. 7-152:Пример пересчета времени: 35 часов,10 минут и 58 секунд соответствуют 126658 секундам.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Значение в (S+), ((S+)+1) или ((S+)+1) находится вне допустимого диапазона.

ПРИМЕР ▾

Следующий пример программы считывает время суток и дату из встроенных часов контроллера и записывает эти данные в регистры данных с D10 по D16. Затем эти данные с помощью команды DHTOS преобразуются в секунды.

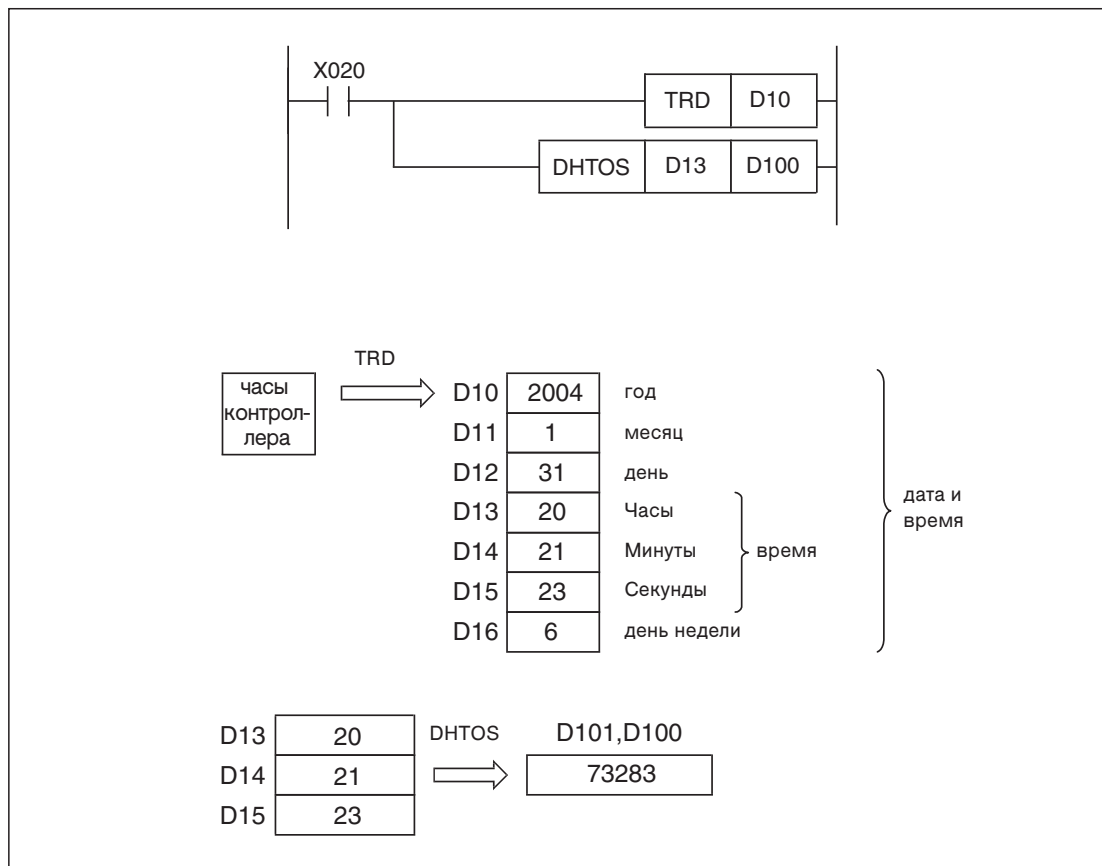



Рис. 7-151::После считывания даты и времени суток с помощью команды TRD команда DHTOS конвертирует время в другую единицу.

△

7.10.6 ПРЕОБРАЗОВАНИЕ ЕДИНИЦЫ ИЗ “СЕКУНД” В “ЧАСЫ” (STON)

		STON		FNC 165			
		Преобразование единицы из “секунд” в “часы”					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Operanden	S1+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□) Используются 3 следующих друг за другом адреса операндов.		●	16бита	32бита	STON STONP
				●		DSTON DSTONP	9

Функция

Пересчет времени в другие единицы

Описание (16-битная обработка)

- Команда STON преобразует значение времени, хранящееся в (S+) в формате “секунды”, в значение, выраженное в часах, минутах и секундах. Результат сохраняется в регистрах с (D+) по ((D+)+2).

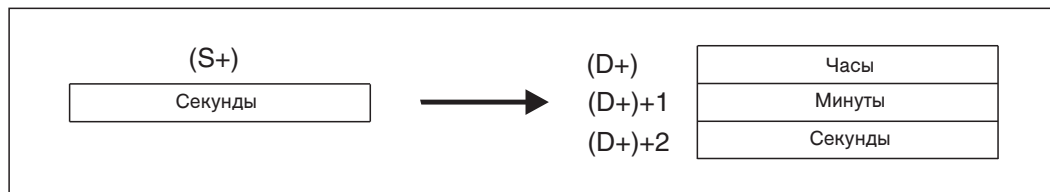


Рис. 7-153:Принцип действия команды STON

- Значение в (S+) может быть в пределах от 0 до 32767 [с].
- Значение в (D+) для часов может быть в пределах от 0 до 9.
- Значения минут и секунд в ((D+)+1) и ((D+)+2) могут принимать значения от 0 до 59.

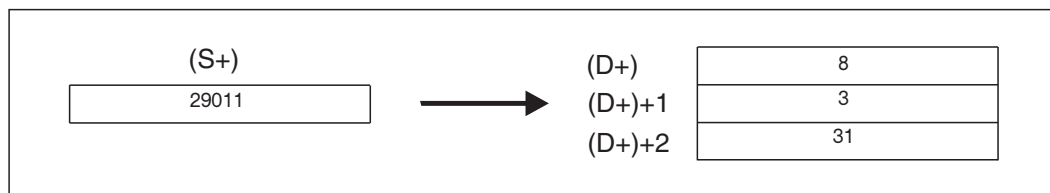


Рис. 7-154:Пример применения команды STON: 29011 секунд соответствуют 8 часам, 3 минутам и 31 секунде.

Описание (32-битная обработка)

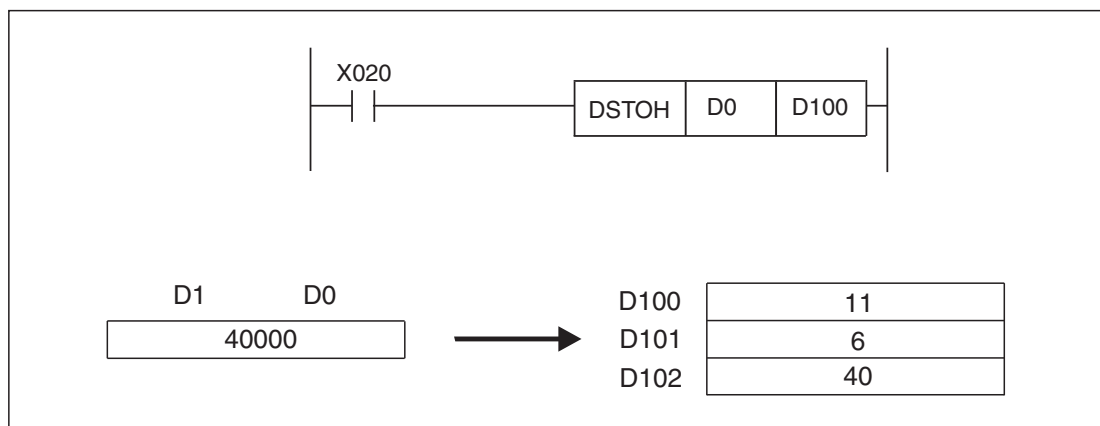
- Команда DSTOH преобразует значение времени, хранящееся в ((S+)+1) и (S+) в формате “секунды”, в формат “часы, минуты и секунды”. Результат сохраняется в регистрах с (D+) по ((D+)+2).

**Рис. 7-155:**Принцип действия команды DSTOH

- Значение в (S+) может находиться в пределах от 0 до 117964799 [с].
- Значение часов в (D+) может находиться в пределах от 0 до 32767.
- Значения минут и секунд ((D+)+1) и ((D+)+2) могут принимать значения в диапазоне от 0 до 59.


**Рис. 7-156:**Пример применения команды DSTOH: 45325 секунд соответствуют 12 часам, 35 минутам и 25 секундам.**ПРИМЕР ▾**

В следующем примере программы значение в секундах, хранящееся в D1 и D0, преобразуется в формат “часы, минуты, секунды” и записывается в регистры с D100 по D102.

**Рис. 7-157:**В этом примере регистры D1и D0 содержат значение “40000” [секунд], которое соответствует 11 часам, 6 минутам и 40 секундам.

△

7.10.7 ЧТЕНИЕ ДАННЫХ ВРЕМЕНИ (TRD)

		TRD		FNC 166			
		Чтение данных времени					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●	●	●	●	●
Операнды	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R*, специальные модули (U□/G□)* Используются 7 следующих друг за другом адресов операндов	●	16бита	32бита	TRD	5	
			●		TRDP	5	

* Только у FX3U

ФУНКЦИЯ

Чтение данных времени

ОПИСАНИЕ

- С помощью TRD-инструкции выполняется чтение данных времени год, месяц, дата, часы, минуты, секунды и день недели по реальному времени.
- Эти данные хранятся в 7 следующих друг за другом операнда в (D+).

Операнды	Назначение	Область значений	⇒	Операнды	Назначение
D8018	Год	00-99	⇒	D+	Год
D8017	Месяц	01-12	⇒	(D+)+1	Месяц
D8016	Дата	01-31	⇒	(D+)+2	Дата
D8015	Часы	00-23	⇒	(D+)+3	Часы
D8014	Минуты	00-59	⇒	(D+)+4	Минуты
D8013	Секунды	00-59	⇒	(D+)+5	Секунды
D8019	День недели	0-6 (воскресенье-суббота)	⇒	(D+)+6	День недели

Табл. 7-24: Считывание времени суток и даты из специальных регистров с D8013 по D8019

УКАЗАНИЕ

Число года обычно хранится в контроллере в виде двузначного числа. Четырехзначное отображение можно активировать, записав в первом программном цикле после запуска контроллера значение "2000" в специальный регистр D8018:

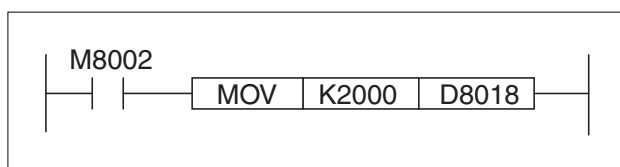


Рис. 7-158:

В сочетании с маркером M8002, значение "2000" вводится в D8018 только в первом программном цикле.

C000372C

Активированное таким образом четырехзначное отображение числа года действует после обработки. Перенос числа "2000" в D8018 должен повторяться при каждом запуске контроллера. На текущее время суток и дату это не влияет.

Если к контроллеру подключается панель управления FX-10DU-E или FX-20DU-E, то должно быть активировано двузначное отображение числа года, так как эти панели не способны отображать четырехзначное число года.

ПРИМЕР ▽

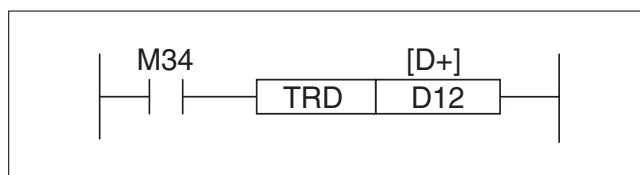


Рис. 7-159:

Пример программирования TRD-инструкции

C000373C



7.10.8 ЗАПИСЬ ДАННЫХ ВРЕМЕНИ (TWR)

		TWR		FNC 167				
		Запись данных времени						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
			●	●	●	●	●	
Операнды	S+	Имп. инструкция(P)	Обработка		Шаги программы			
	T, C, D, R*, специальные модули (U□/G□)* Используются 7 следующих друг за другом адресов операндов	●	16бита	32бита	TWR	5		
			●		TWRP	5		

* Только у FX3U

ФУНКЦИЯ

Запись данных времени

ОПИСАНИЕ

- С помощью TWR-инструкции выполняется запись данных времени год, месяц, дата, часы, минуты, секунды и день недели по реальному времени.
- Эти данные хранятся в 7 следующих друг за другом операнда в (S+).

Операнды	Назначение	Область значений
S+	Год	00-99
(S+)+1	Месяц	01-12
(S+)+2	Дата	01-31
(S+)+3	Часы	00-23
(S+)+4	Минуты	00-59
(S+)+5	Секунды	00-59
(S+)+6	День недели	0-6 (воскресенье-суббота)

Операнды	Назначение
D8018	Год
D8017	Месяц
D8016	Дата
D8015	Часы
D8014	Минуты
D8013	Секунды
D8019	День недели

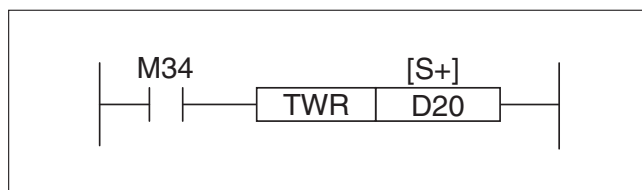
Табл. 7-25: Запись данных времени

* При четырехзначном отображении числа года (см. раздел 7.10.7) значения от "80" до "99" лет обозначают годы с 1980 по 1999, а значения от "00" до "79" - годы с 2000 по 2079. Например, "06" означает "2006".

УКАЗАНИЕ

При применении TWR-инструкции не требуется включение меркера M8015 (сохранение реального времени).

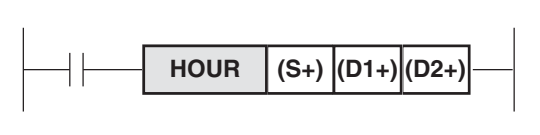
Если в регистрах с (S+) по ((S+)+6) задано значение, находящееся вне допустимого диапазона значений, часы не устанавливаются.

ПРИМЕР ▾**Рис. 7-160:**

Пример программирования TWR-инструкции

C000374C

7.10.9 СЧЕТ РАБОЧИХ ЧАСОВ (HOUR)

				HOUR		FNC 169				
				Счет рабочих часов						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
					●	●				●
Операнды	S+		D1+	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R* специальные модули (U□/G□)*, V, Z,		D, R*	Y, M, S			16бита	32бита	HOUR	7
							●	●	DHOUR	13

* Только у FX3U

ФУНКЦИЯ

По окончании отсчета установленного времени выдается сигнал

ОПИСАНИЕ

- В (S+) указывается время в часах, после отсчета которого включается операнд, указанный в (D2+).
- 16-ти битовая инструкция: (D1+) содержит текущее значение полных часов. В ((D1+)+1) заносится оставшееся время в секундах.
- 32-битная команда: (D1+) и (D1+1) содержат полное значение часов. В ((D1+) + 2) сохраняется остающееся время в секундах.
- При 16-битной команде указанный в (D2+) операнд включается, если фактическое значение в (D1+) больше заданного значения в (S+). При 32-битной команде указанный в (D2+) операнд включается, если фактическое значение в ((D1+)+1) и (D1+) больше заданного значения в (S+).

УКАЗАНИЕ

Чтобы действительное прошедшее время сохранить и при отключении напряжения, для (D1+) должен быть применен регистр с памятью.

Счет продолжается и после того, как был включен указанный в (D2+) операнд. Счет приостанавливается, если превышает максимальный диапазон 16-битного или 32-битного операнда в (D1+). Если сигнал в (D2+) требуется устанавливать с неизменными интервалами, то при достижении заданного значения необходимо стирать фактическое значение в регистрах (D1+) и ((D1+) + 1) (16 битов) или в регистрах с (D1+) по ((D1+) + 2) (32 бита).

ПРИМЕР ▾

HOUR-инструкции

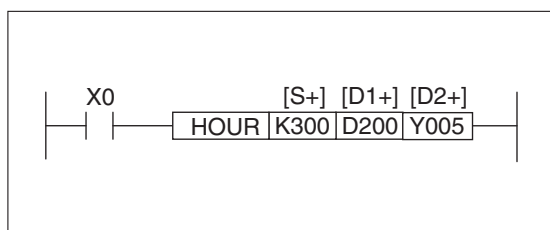


Рис. 7-161:

Пример программирования HOUR-инструкции

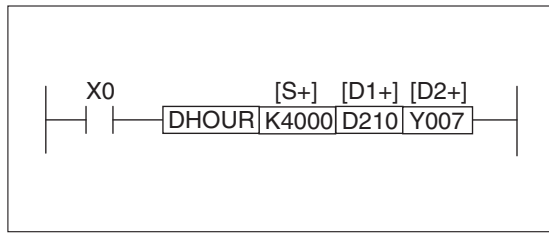
C000405C

После отсчета 300 часов и секунд, после того как будет включен вход X), включится Y5.

△

ПРИМЕР ▾

DHOURL-инструкции

**Рис. 7-162:**Пример программирования
DHOURL-инструкции

C000406C

Между включением X0 и Y7 проходит 4000 часов и секунда.

△

7.11 ИНСТРУКЦИИ КОДА ГРЕЯ

Энкодеры, способные определять абсолютное положение элементов машины, обычно передают эту информацию на контроллер в виде особого кода - кода Грея. Преимущество кода Грея заключается в том, что при каждом цифровом шаге изменяется только один бит. Благодаря этому достигается большая надежность передачи.

ОБЗОР ИНСТРУКЦИЙ FNC 170...171

Символ	FNC	Назначение	Разд.
GRY	170	Преобразование целого числа в код Грея	7.8.1
GBIN	171	Преобразование кода Грея в целое число	7.8.2

Табл. 7-26: Обзор инструкции FNC 170...171

7.11.1 ПРЕОБРАЗОВАНИЕ ЦЕЛОГО ЧИСЛА В КОД ГРЕЯ (GRY)

		GRY		FNC 170								
		(S+)	(D+)	Преобразование целого числа в код Грея								
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U					
					●	●	●					
Операнды	S+	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, специальные модули (U□/G□)*, V, Z	D+	KnY, KnM, KnS, T, C, D, R*, V, Z	Имп. инструкция(P)	●	Обработка	16бита	32бита	Шаги программы	GRY, GRYP	5
						●	●	DGRY, DGRYP	9			

* Только у FX3U

ФУНКЦИЯ

Конвертирование значения целого числа в код Грея

ОПИСАНИЕ

- С помощью GRY-инструкции выполняется конвертирование значения целого числа в (S+) в код Грея.
- Результат заносится в (D+).

УКАЗАНИЕ

Используя характеристику кода Грея можно числовое значения посредством инкрементирования (приращения) исходных данных выдать в каждом цикле программы, не применяя стробоскопные сигналы.

ПРИМЕР ▾

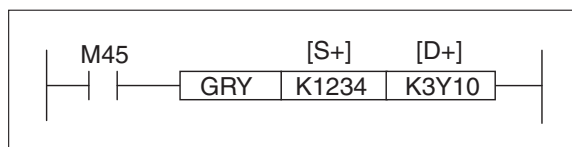


Рис. 7-163:

Пример программирования GRY-инструкции

C000375C

При включении меркера M45 целое значение K1234 конвертируется в код Грея. Результат выдается на выходы Y10...Y23.

△

7.11.2 ПРЕОБРАЗОВАНИЕ КОДА ГРЕЯ В ЦЕЛОЕ ЧИСЛО (GBIN)

		GBIN		FNC 171			
		(S+)	(D+)	Преобразование кода Грея в целое число			
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U	
				●	●	●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, специальные модули (U□/G□)*, V, Z	KnY, KnM, KnS, T, C, D, R*, V, Z	●	16бита	32бита	GBIN GBINP	5
				●	●	DGBIN DBBINP	9

* Только у FX3U

ФУНКЦИЯ

Конвертирование значения в коде Грея в целое число

ОПИСАНИЕ

- С помощью GBIN-инструкции выполняется конвертирование значения, закодированного в коде Грея в (S+), в целое число.
- Результат заносится в (D+).

УКАЗАНИЕ

Эта GBIN-инструкция может применяться для чтения данных закодированных в коде Грея.

Если для операндов источника применяются входы X0...X17, то время чтения может быть сокращено посредством настройки фильтров активизации (FNC51, REFF)

ПРИМЕР ▾

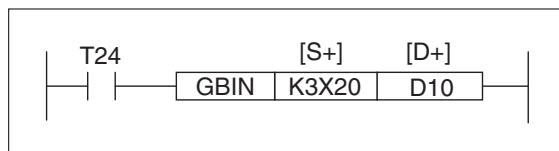


Рис. 7-164:

Пример программирования GBIN-инструкции

C000376C

При включении контакта таймера T24 значение в код Грея на входах X20...X33 конвертируется в значение целого числа. Результат заносится в D10.

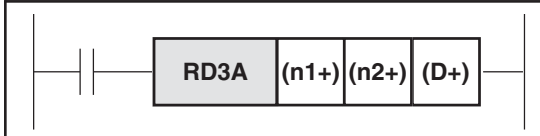
△

7.12 ОБМЕН ДАННЫМИ С АНАЛОГОВЫМИ МОДУЛЯМИ

Символ	FNC	Назначение	Abschnitt
RD3A	176	Чтение аналоговых входных значений	7.12.1
WR3A	177	Передача аналоговых выходных значений	7.12.2

Табл. 7-36: Обзор инструкций по обмену данными с аналоговыми модулями

7.12.1 ЧТЕНИЕ АНАЛОГОВЫХ ВХОДНЫХ ЗНАЧЕНИЙ (RD3A)

				RD3A		FNC 176					
				Чтение аналоговых входных значений							
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
						●	●	●	●		
Операнды	n1+	n2+	D+	Имп. Инструкция (P)	Обработка		Шаги программы				
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, V, Z		KnY, KnM, KnS, T, C, D, R*, V, Z		●	16бита	32бита	RD3A RD3AP	7		

* Только у FX3U

ФУНКЦИЯ

Чтение аналоговых входных значений из аналогового модуля FX0N-3A и FX2N-2AD

ОПИСАНИЕ

- В (n1+) указывается адрес специального модуля (K0 до K7).
- Аналоговое входное значение в (n2+) указанного входного канала (K1 или K2) записывается в (D+).

Модуль FX0N-3A: K1 = Канал 1, K2 = Канал 2; значение D = 0 ... 255 (8 бит)

Модуль FX2N-2AD: K21 = Канал 1, K22 = Канал 2; значение D = 0 ... 4095 (12 бит)

УКАЗАНИЕ

Перед чтением входного значения должна быть установлена входная характеристика аналогового модуля. Подробные указания этой процедуры находятся в руководстве по обслуживанию аналогового модуля.

ПК серии FX1N могут обмениваться только с модулями FX0N-3A.

ПРИМЕР ▾

RD3A-инструкция

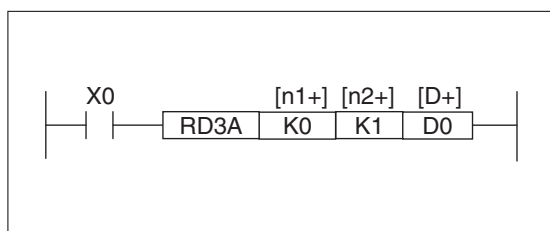


Рис. 7-165:

Пример программирования RD3A-инструкции

C000407C

Аналоговое значение аналогового модуля, установленное в канале 1, передается по адресу 0 в регистр D0.

7.12.2 ЗАПИСЬ АНАЛОГОВЫХ ВЫХОДНЫХ ЗНАЧЕНИЙ (WR3A)

				WR3A		FNC 177				
				Запись аналоговых выходных значений						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
Операнды	n1+	n2+	S+	Имп. инструкция(P)	Обработка		Шаги программы			
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, V, Z		KnY, KnM, KnS, T, C, D, R*, V, Z		●	16бита	32бита	WR3A WR3AP	7	

* Только у FX3U

ФУНКЦИЯ

Запись аналоговых выходных значений в аналоговый модуль FX0N-3A и FX2N-2DA

ОПИСАНИЕ

- В (n1+) указывается адрес специального модуля (K0 до K7).
- Аналоговое выходное значение в (S+) передается выходному каналу (только K1), указанному в (n2+).

Модуль FX0N-3A: K1 = Канал 1; значение в (S+) = 0 ... 255 (8 бит)

Модуль FX2N-2AD: K21 = Канал 1, K22 = Канал 2; значение в (S+) = 0 ... 4095 (12 бит)

УКАЗАНИЕ

Перед передачей выходного значения должна быть установлена выходная характеристика аналогового модуля. Подробные указания этой процедуры находятся в руководстве по обслуживанию аналогового модуля.

ПК серии FX1N могут обмениваться только с модулями FX0N-3A.

ПРИМЕР ▾

WR3A-инструкция

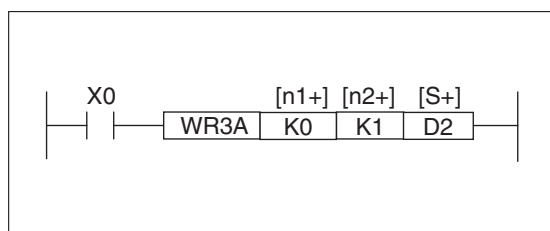


Рис. 7-166:

Пример программирования WR3A-инструкции

C000408C

Содержание регистра D2 передается для выдачи его как аналогового значения к аналоговому модулю с адресом 0.

△

7.13 ВНЕШНЯЯ КАССЕТА ПАМЯТИ

Символ	FNC	Назначение	Разд.
EXTR	180	команды внешней кассеты памяти ROM	7.13.1

Табл. 7-37:команды внешней кассеты памяти ROM

УКАЗАНИЕ

Для контроллера FX3U, который не может выполнять команду EXTR, вместо команд с EXTR K10 по EXTR K13 можно использовать команды с FNC270 по FNC274 (IVCK, IVDR, IVRD, IVWR и IVBWR).

7.13.1 ФУНКЦИИ ВНЕШНЕЙ ROM КАССЕТЫ (EXTR)

		EXTR		FNC 180			
		команды внешней кассеты памяти					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
					●	●	
Операнды	S	SD1+, SD2+, SD3+		Имп. инструкция(P)	Обработка		Шаги программы
	K, H	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z, X, Y, M, S		●	16бита	32бита	EXTR EXTRP
					●	●	DEXTR DEXTRP
							9
							17

ФУНКЦИЯ

Функция EXTR используется только со специальной кассетой FX2N-ROM-E1 или FX2NC-ROM-CE1.

ОПИСАНИЕ

- Значение S, сохраненное во внешней ROM (от K0 до K32767) определяет номер функции и инструкцию.
- SD1, SD2 и SD3 - параметры инструкции. S или D изменяются в зависимости от номера функции. Номер инструкции определяет режим (16 или 32-битный).
- В некоторых номерах функций параметры SD1 ..SD3 не требуются. В таких случаях K0 должен быть описан в программе. K0 игнорируется при внутренней обработке контроллера.

Внешние кассеты памяти ROM (FX2N-ROM-E1 и FX2NC-ROM-CE1) устанавливаются в порт памяти контроллеров FX2N(C) и имеют объем памяти EEPROM в 16кШагов. Кроме того, FX2NC-ROM-CE1 также содержит часы реального времени.

Модули FX2N-ROM-E1 и FX2NC-ROM-CE1 являются совместимыми представляют новое поколение модулей памяти FX-EEPROM-16 и FX2NC-EEPROM16C соответственно.

УКАЗАНИЕ

Модули FX2N-ROM-E1 и FX2NC-ROM-CE1 функционируют только с контроллерами FX2N(C) версии V3.00 или позже.

Сообщения об ошибках при выполнении функции EXTR

- При неправильно установленной ROM кассеты бит M8061 активизируется и в D8061 записывается код ошибки 6110.
- Если активизируется функция EXTR, в то время как никакой кассеты не установлено, вводится бит M8065 и вносится в D8065 записывается код ошибки 6512.

EXTR K10 и EXTR K13 (Связь с преобразователями частоты)**ФУНКЦИЯ**

Внешние кассеты ROM содержат инструкции 10-13, обеспечивающие считывание и запись данных в преобразователи частоты Mitsubishi Electric A500/E500/S500. Данные функции требуют использования в контроллерах коммуникационных опций FX2N-485-BD или FX0N-485ADP.

№ функции	Описание	Направление данных	Ссылка на документацию инверторов
EXTR K10	мониторинг работы	INV в PLC	Раздел связь с компьютером и коммуникационные функции .
EXTR K11	контроль работы	PLC в INV	
EXTR K12	считывание параметров	INV в PLC	См. список параметров преобразователей.
EXTR K13	запись параметров	PLC в INV	

Табл. 7-38: Команды для обмена данных с преобразователями частоты MITSUBISHI.

УКАЗАНИЕ

Шестиразрядные команды, поддерживаемые преобразователями частоты серий E500 и S500, не поддерживаются инструкцией EXTR.

ОПИСАНИЕ

- Если условие работы функции EXTR выполнено, то идет постоянный обмен с преобразователем частоты. Обмен данными начинается, если порт активирован. Если условие работы функции EXTR сбросилось во время обмена данных, связь все же продолжается, до тех пор пока обмен данных незакончится.
- Если в программе задано выполнение нескольких функций EXTR одновременно, то последующая функция EXTR автоматически активируется после исполнения предыдущей. В регистре D8155 находится номер шага программы с функцией, которая в данный момент выполняется.
- После выполнения функции EXTR M8029 ставится активным. M8029 используется также другими функциями. Поэтому этот меркер M8029 остается активным только до тех пор пока не начнет выполняться другая функция.
- Функция EXTR не может использоваться в пределах цикла FOR NEXT.
- Функция EXTR не может использоваться вместе с программой прерывания.
- Функция EXTR выполняется за несколько сканов программы. Обращайте внимание при программировании подпрограмм на то, чтобы функция EXTR повторно не активизировалась в скане.
- Если функцию EXTR перескакивают с использованием функций SJ-или CJP , порт не открывается и вследствие этого обмен данных останавливается.
- В сочетании с функцией Master-Control (MC, MCR), функция EXTR может использоваться без ограничений.
- Если шаг с функцией EXTR, в пошаговом режиме, STL деактивируется порт закрывается и вследствие этого обмен данных останавливается.
- При редактировании программы в режиме RUN, нельзя изменять операнд S(K10, K11, K12 и K13) и удалять функцию EXTR.

Параметры настройки порта

Функции от EXTR K10 до EXTR K13 работают через адаптер FX2N-485-BD или FX0N-485-BD тем же способом, как и функция RS (FNC 80).

В программном обеспечении GX Developer или GX IEC Developer в пункте меню "PLC-Parameter". Выберите следующие настройки :

- Длина данных 7 Бит
- Контроль паритета по четности
- 1 стоповый бит
- Скорость передачи данных 4 800, 9 600 или 19 200 Бод

Настройки для связи с преобразователем частоты

При подключении преобразователя частоты требуются следующие настройки порта:

Параметр		Установка
Стандарт связи		RS 485
Число преобразователей частоты в сети		1:n (8 максимум)
Скорость связи		выбирается 4800, 9600 или 19200 Бод
Управление связью		асинхронная передача данных
Способ передачи		полудуплекс
Параметры связи	Тип символов	Фиксирована (7 бит)
	Число стоповых битов	1
	Терминатор	CR
	Проверка четности	Фиксирована (чет)
	Контр. сумма	Фиксирована
	Время ожидания	A500 E500 S500 A500 + A5NR

Табл. 7-39:Параметры интерфейса связи преобразователя частоты для связи с контроллером

Параметры инверторов

Параметр	Описание	Содержимое	Установки для контроллера
117	Номер станции	Соответствует номеру станции, установленному с устройства программирования, подключенного в разъем PU. Если преобразователей частоты в сети более 2х, необходимо установить адрес в сети каждого преобразователя.	Установите соответствующие номера станций в программе контроллера.
118	Скорость передачи	192 (19200 Бод) 96 (9600 Бод) 48 (4800 Бод)	Обычно выбирается 192. Для скоростной обработки в контроллере используйте 96 или 48.
119	Число стоповых битов/число битов данных	10	Выберите 10
120	Проверка четности	2	Выберите 2
121	Число попыток восстановления связи	0 ... 10	При пробном пуске выберите 9999 и выполните настройку. При работе выберите значение в соответствии с характеристиками системы.
122	Интервал времени проверки связи	0,1 ... 999,8	
123	Время ожидания	9999	Выберите 9999
124	Выбор CR, LF	1	Выберите значение в соответствии с параметрами системы

Табл. 7-40: Параметры для преобразователей частоты А-500 и Е-500

Параметр	Описание	Содержимое	Установки для контроллера
331	Номер станции	Настроенный номер станции должен совпадать с номером станции, используемым в программе контроллера.	Если к одному контроллеру подключены несколько преобразователей частоты, с помощью номера станции выбирается определенный преобразователь.
332	Скорость передачи	192 (19200 Бод) 96 (9600 Бод) 48 (4800 Бод)	Обычно выбирается 192. Для скоростной обработки в контроллере используйте 96 или 48
333	Число стоповых битов/число битов данных	10	10 = 7 битов данных, 1 стоп-бит
334	Проверка четности	2	проверка на четный результат
335	Число попыток восстановления связи	0 ... 10	При пробном пуске выберите 9999 и выполните настройку. При работе выберите значение в соответствии с
336	Интервал времени проверки связи	0 ... 999,8	
337	Время ожидания	9999*	
338	Запись команд управления	0 или 1	Выберите значение в соответствии с параметрами системы
339	Запись команды скорости	0 или 1	
340	Выбор режима связи	0, 1 или 2	
341	Выбор использования CR/LF	1	Выберите 1 (только CR)
342	Выбор записи в EEPROM	0 или 1	0 = параметры вводятся в EEPROM 1 = параметры вводятся в RAM

Табл. 7-41: Параметры для преобразователей частоты А500 с опциональным модулем А5NR

- * Во избежание опасностей, а также для того, чтобы при сбоях коммуникации преобразователь оперативно отключался, время ожидания следует настроить на минимально возможное значение.

Параметр	Описание	Содержимое	Установки для контроллера
n1	Номер станции	Настроенный номер станции должен совпадать с номером станции, используемым в программе контроллера.	Если к одному контроллеру подключены несколько преобразователей частоты, с помощью номера станции выбирается определенный преобразователь.
n2	Скорость передачи	192 (19200 Бод) 96 (9600 Бод) 48 (4800 Бод)	Обычно выбирается 192. Для скоростной обработки в контроллере используйте 96 или 48
n3	Число стоповых битов/число битов данных	10	10 = 7 битов данных, 1 стоп-бит
n4	Проверка четности	2	проверка на четный результат
n5	Число попыток восстановления связи	0 ... 10	При пробном пуске выберите 9999 и выполните настройку. При работе выберите значение в соответствии с характеристиками системы.
n6	Интервал времени проверки связи	0,1 ... 999,8	
n7	Время ожидания	9999	время ожидания задается данными коммуникации
n11	Выбор использования CR/LF	1	Выберите 1 (только CR)

Табл. 7-42:Параметры для преобразователей частоты серии S500

Формат данных

Пример формата передачи для записи данных из контроллера в преобразователь частоты

ENQ	станция №6		Команда 80H		Время ожидания=0	Данные = 1234				Контрольная сумма		CR
05H	30H	36H	38H	30H	30H	31H	32H	33H	34H	43H	38H	0DH

Рис. 7-167:Формат передачи для записи данных из контроллера в преобразователь частоты

Подсчет контрольной суммы:

$$30H + 36H + 38H + 30H + 30H + 31H + 32H + 33H + 34H = 1C8H$$

(„C“ РАВНО 43H В ASCII-КОДЕ, „8“ РАВНО 38H В ASCII-КОДЕ)

ПРИМЕР ▾

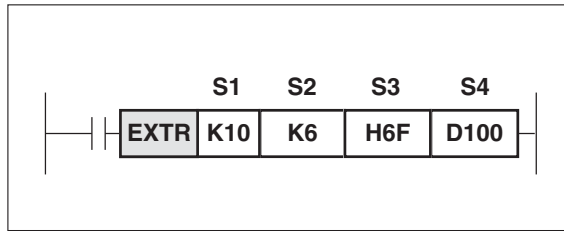


Рис. 7-168:

Пример команды EXTR K10

В этом примере выходная частота преобразователя частоты с номером станции 6 читается и сохраняется в регистре D100.



Функция	Описание	Обработка		
		16бита	32бита	Импульсная (P)
EXTR K10	Мониторинг состояния	●		

Операнды		Диапазон параметров
S1	K, H	K10: номер отображаемой функции
S2	K, H, D	Номер станции преобразователя частоты (0..31)
S3	K, H, D	Код инструкции преобразователя частоты (зависит от модели)
S4	D, KnY, KnM, KnS	Определение области хранения считанного значения

код инструкции	Описание кода инструкции	Возможность использования инструкции в данной серии преобразователя частоты			Число разрядов данных	Формат
		A500	E500	S500		
H7B	Режим работы	●	●	●	4	B ⇒ E F
H6F	Вых. частота	●	●	●	4	B ⇒ E F
H70	Вых. ток	●	●	●	4	B ⇒ E F
H71	Вых. напряжение	●	●		4	B ⇒ E F
H72	Спец. отображение	●			4	B ⇒ E F
H73	номер спец. отображ.	●			2	B ⇒ E' F
H74	Тип сбоя	●	●	●	4	B ⇒ E F
H75	Тип сбоя	●	●	●	4	B ⇒ E F
H76	Тип сбоя	●	●		4	B ⇒ E F
H77	Тип сбоя	●	●		4	B ⇒ E F
H7A	Отображение состояния преобразователя частоты	●	●	●	2	B ⇒ E' F
H6E	Считывание заданной частоты (EEPROM)			●	4	B ⇒ E F
H6D	Считывание заданной частоты (RAM)			●	4	B ⇒ E F

Табл. 7-43: Подробности по S3

УКАЗАНИЕ

Вы найдете подробные сведения о кодах инструкции и форматах данных в руководстве по эксплуатации преобразователя частоты.

ПРИМЕР ▾

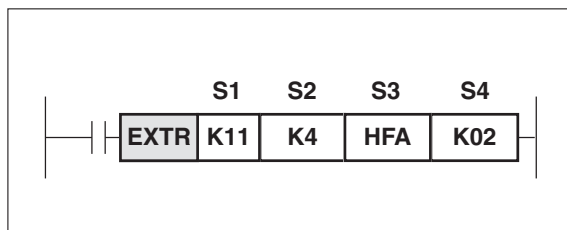


Рис. 7-169:

Пример команды EXTR K11

В этом примере в преобразователь частоты с номером станции 4 посылается команда вращения вперед.



Функция	Описание	Обработка		
		16бита	32бита	Импульсная (P)
EXTR K11	Управление	●		

Операнды		Диапазон параметров
S1	K, H	K11: номер функции для управления преобразователем частоты
S2	K, H, D	Номер станции преобразователя частоты (0..31)
S3	K, H, D	Код инструкции преобразователя частоты(зависит от модели)
S4	K, H, D, KnX, KnY, KnM, KnS	Значение, записываемое в преобразователь частоты

код инструкции	Описание кода инструкции	Возможность использования инструкции в данной серии преобразователя частоты			Число разрядов данных	Формат
		A500	E500	S500		
HFB	Режим работы	●	●	●	4	A ⇒ C D
HF3	Выбор номера спец. отображ.	●			2	A' ⇒ C D
HFA	Команды управления	●	●	●	2	A' ⇒ C D
HEE	Запись заданной частоты (EEPROM)	●	●	●	4	A ⇒ C D
HED	Запись заданной частоты (RAM)	●	●	●	4	A ⇒ C D
HFD	Сброс преобразователя частоты	●	●	●	4	A (нет ответа)
HF4	Сброс предыстории аварий	●		●	4	A ⇒ C D
HFC	Сброс всех параметров	●	●	●	4	A ⇒ C D
HFC	Сброс информации о пользователе	●			4	A ⇒ C D

Табл. 7-44:Подробности по S3

УКАЗАНИЕ

Вы найдете подробные сведения о кодах инструкции и форматах данных в руководстве по эксплуатации преобразователя частоты.

ПРИМЕР ▾

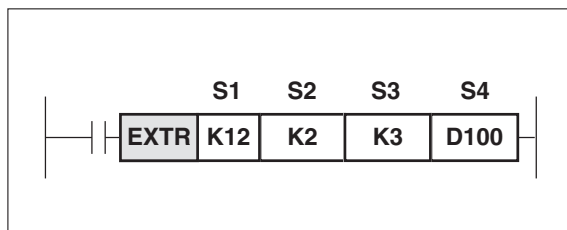


Рис. 7-170:

Пример команды EXTR K12

Считывание базовой частоты (параметр 3) с преобразователя частоты станционный номер 2 и сохранение в регистре D100.

Функция	Описание	Обработка		
		16бита	32бита	Импульсная (P)
EXTR K12	Считывание параметров	●		

Операнды		Диапазон параметров
S1	K, H	K12: номер функции для считывания параметров преобразователя частоты
S2	K, H, D	Номер станции преобразователя частоты(0..31)
S3	K, H, D	Код инструкции преобразователя частоты(зависит от модели)
S4	D, KnY, KnM, KnS	Значение, записываемое в преобразователь частоты

ПРИМЕР ▾

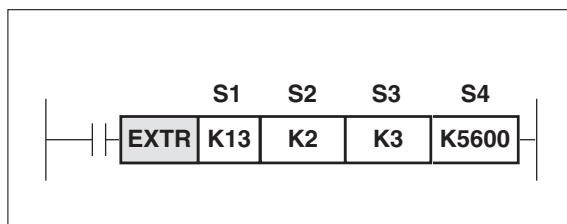


Рис. 7-171:

Пример команды EXTR K13

При этом примере значение 5 600 записывается, как базовая частота (параметр 3) в преобразователь с номером станции 2.



Функция	Описание	Обработка		
		16бита	32бита	Импульсная (P)
EXTR K13	Запись параметров	●		

Операнды		Диапазон параметров
S1	K, H	K13: номер функции для записи параметров в преобразователя частоты
S2	K, H, D	Номер станции преобразователя частоты(0..31)
S3	K, H, D	Код инструкции преобразователя частоты(зависит от модели)
S4	K, H, D, KnX, KnY, KnM, KnS	Значение, записываемое в преобразователь частоты

УКАЗАНИЕ

Вы найдете подробные сведения о кодах инструкции и форматах данных в руководстве по эксплуатации преобразователя частоты.

Взаимосвязь между EXTR K12/K13 и A500/E500/S500.

Для параметров, требующих связанных параметров. Если значение '+0', '+1000' или '+2000' устанавливается в номере параметра, второй параметр будет автоматически перезаписан перед считыванием/записью общих параметров.

Установки для 3-го набора параметров EXTR K12/K13 при программном режиме A500.

Параметр	Назначение	Запись/считывания		
		Частота	Время	Направление
201	Программные установки 1	201	1201	2201
202		202	1202	2202
203		203	1203	2203
204		204	1204	2204
205		205	1205	2205
206		206	1206	2206
207		207	1207	2207
208		208	1208	2208
209		209	1209	2209
210		210	1210	2210
211	Программные установки 2	211	1211	2211
212		212	1212	2212
213		213	1213	2213
214		214	1214	2214
215		215	1215	2215
216		216	1216	2216
217		217	1217	2217
218		218	1218	2218
219		219	1219	2219
220		220	1220	2220
221	Программные установки 3	221	1221	2221
222		222	1222	2222
223		223	1223	2223
224		224	1224	2224
225		225	1225	2225
226		226	1226	2226
227		227	1227	2227
228		228	1228	2228
229		229	1229	2229
230		230	1230	2230

Табл. 7-45: Установки для 3-го набора параметров

Считывание и запись смещения/наклона для A500/E500/S500

Параметр	Назначение	Запись/считывания		
		Смещение	Наклон	Значение на входе
902	Смещение входа по напряжению	201	1201	2201
903	Наклон входа по напряжению	202	1202	2202
904	Смещение входа по току	203	1203	2203
905	Наклон входа по току	204	1204	2204

Таб. 7-46:Считывание и запись смещения/наклона**Определение спец. D-регистров и спец. M-выходов**

Операнд	Описание
M8154	M8145 = 0: время ожидания 15 мс, M8145 = 1: время ожидания 1000 мс Обычно после обмена данными выдерживается время ожидания 15 мс, прежде чем начинается очередной обмен данными. Установкой M8154 время ожидания удлиняется до 1 с, благодаря чему становится возможной диагностика ошибок, например, путем анализа регистров D8155 или D8156.
M8155	M8155 = 0: коммуникация завершена, M8155 = 1: происходит коммуникация
M8156	M8156 = 1: ошибка при коммуникации Анализируйте M8156 сразу после выполнения команды EXTR. При выполнении следующей команды EXTR M8156 снова стирается.
M8157	M8157 = 1: ошибка при коммуникации (сохраненная) При переходе контроллера из режима STOP в режим RUN маркер M8157 сбрасывается.
D8154	Время ожидания ответа от преобразователя частоты (FU) Содержимое D8154 = 0: время ожидания = 2 с Содержимое D8154 > 0: время ожидания x 0,1 с Если в течение настроенного времени ожидания после передачи контроллером запроса на коммуникацию не поступает ответ от преобразователя частоты, выводится сообщение об ошибке.
D8155	Номер шага команды для коммуникации с преобразователем. Если коммуникация не происходит, D8155 содержит значение "-1".
D8156	Код ошибки (см. следующую таблицу) Если при выполнении команды EXTR возникла ошибка, D8156 обновляется. При переходе контроллера из режима STOP в режим RUN в регистр D8156 записывается "-1".
D8157	Номер шага, при котором после запуска контроллера возникла первая ошибка коммуникации. Номер шага сохраняется до тех пор, пока при переходе контроллера из режима STOP в режим RUN в регистр D8157 не записывается "-1".

Табл. 7-47:Определение спец. D-регистров и спец. M-выходов

Коды ошибок связи

D8156	Содержание ошибки	Реакция преобразователя частоты
0000H	Связь завершена нормально (ошибок нет)	
0001H	Преобразователь частоты не ответил	
0002H	Ошибка timeout взаимосвязана с M8129. Ошибка происходит если передача из преобразователя частоты прервана.	
0003H	Неопределенная станция ответила.	
0004H	Контрольная сумма не соответствует.	
0005H	В параметрах считывания/записи параметры 400...899 определены, но не поддерживаются. Устанавливается код ошибки 6702 в D8067.	
0006H	Порт связи используется другой функцией и поэтому не может использоваться инструкцией EXTR. Устанавливается код ошибки 6702 в D8067.	
0100H	Преобразователь частоты не передал код ошибки H0 - computer NAK error. Данные в запросе содержат ошибку за пределом допустимого числа попыток передачи.	Преобразователь частоты не принимает данные, не генерирует сбой и останавливается.
0101H	Преобразователь частоты не передал ошибку H1 -- контроль четности.	
0102H	Преобразователь частоты не передал ошибку H2 - ошибка контрольной суммы.	
0103H	Преобразователь частоты передал ошибку H3 - ошибка протокола. Синтаксическая ошибка в полученных инвертором данных, передача данных не завершена вовремя или ошибка в CR/LF.	
0104H	Преобразователь частоты передал ошибку H4 - структура посылки нарушена. Ошибка в числе стоповых битов.	
0105H	Преобразователь частоты передал ошибку H5 - переполнение.	Преобразователь частоты не принимает данные, не генерирует сбой и не останавливается.
0106H	Преобразователь частоты передал ошибку H6.	
0107H	Преобразователь частоты передал ошибку H7 - ошибка символов. Получен символ, отличный от 0...9, A...F.	
0108H	Преобразователь частоты передал ошибку H8.	
0109H	Преобразователь частоты передал ошибку H9.	

Табл. 7-48: коды ошибок (1)

D8155	Содержание ошибки	Реакция преобразователя частоты
010AH	Преобразователь частоты передал ошибку HA. Ошибка режима. Не выбран режим связи по интерфейсу.	Преобразователь частоты не принимает данные, не генерирует сбой и не останавливается.
010BH	Преобразователь частоты передал ошибку HB- ошибка кода инструкции. Несуществующая инструкция.	
010CH	Преобразователь частоты передал ошибку HC- ошибка данных. При записи использованы данные вне допустимого диапазона.	
010DH	Преобразователь частоты передал ошибку HD.	
010EH	Преобразователь частоты передал ошибку HE.	
010FH	Преобразователь частоты передал ошибку HF.	

Табл. 7-35: Коды ошибок связи

Ошибка связи

Связь выполняется в общем три раза, включая 2 повтора. Если происходит ненормальное завершение связи даже после третьей попытки, возникает ошибка. Тип ошибки классифицируется следующим образом:

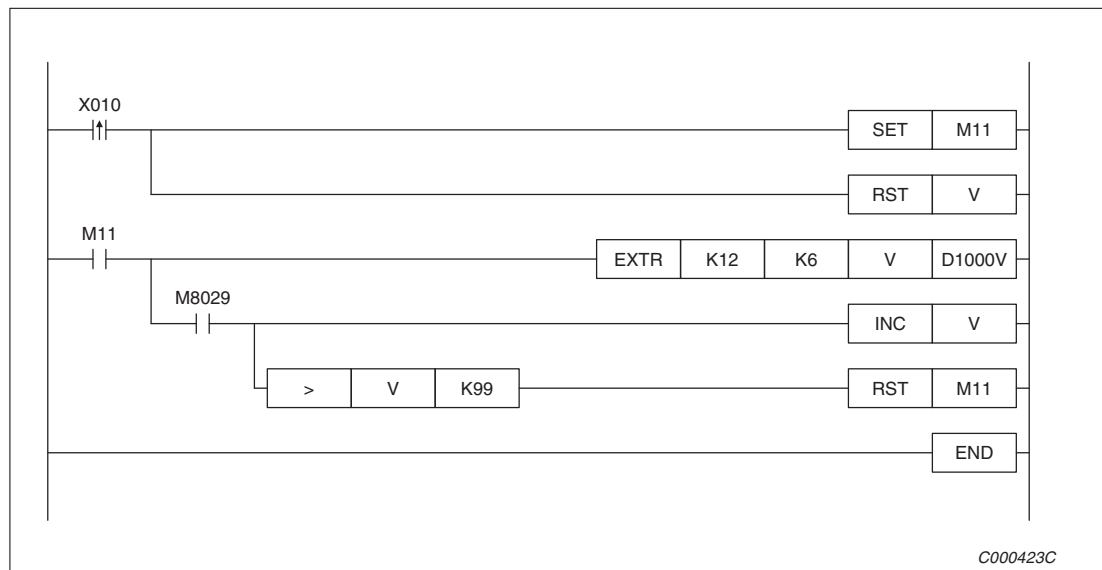
Вид ошибки	Бит ошибки		Код ошибки в D8156
	M8156	M8157	
Если код ошибки возвращается от преобразователя частоты	●	●	●
Если преобразователь частоты не отвечает	●	●	●
Если ответ дан неопределенной станцией	●	●	●
Если получена ошибка передачи		●	
Если бит M8063 включен и код ошибки 6301 установлен в D8067.		●	
Если контрольная сумма данных возвращенных преобразователем частоты не соответствует норме.		●	

Табл. 7-49: Таблица коммуникационных ошибок

ПРИМЕР ▾

Пример программы 1.

Программа считывает параметры 0..99 станции №6 в D1000..D1099 контроллера.

**Рис. 7-172:**Примерная программа чтения параметров

ПРИМЕР ▾

Пример программы 2

Программа считывает параметры 0..99 станций №6, 7, 8 и 9 в D1000...D1099, D1100...D1199, D1200... D1299 и D1300... D1399 соответственно.

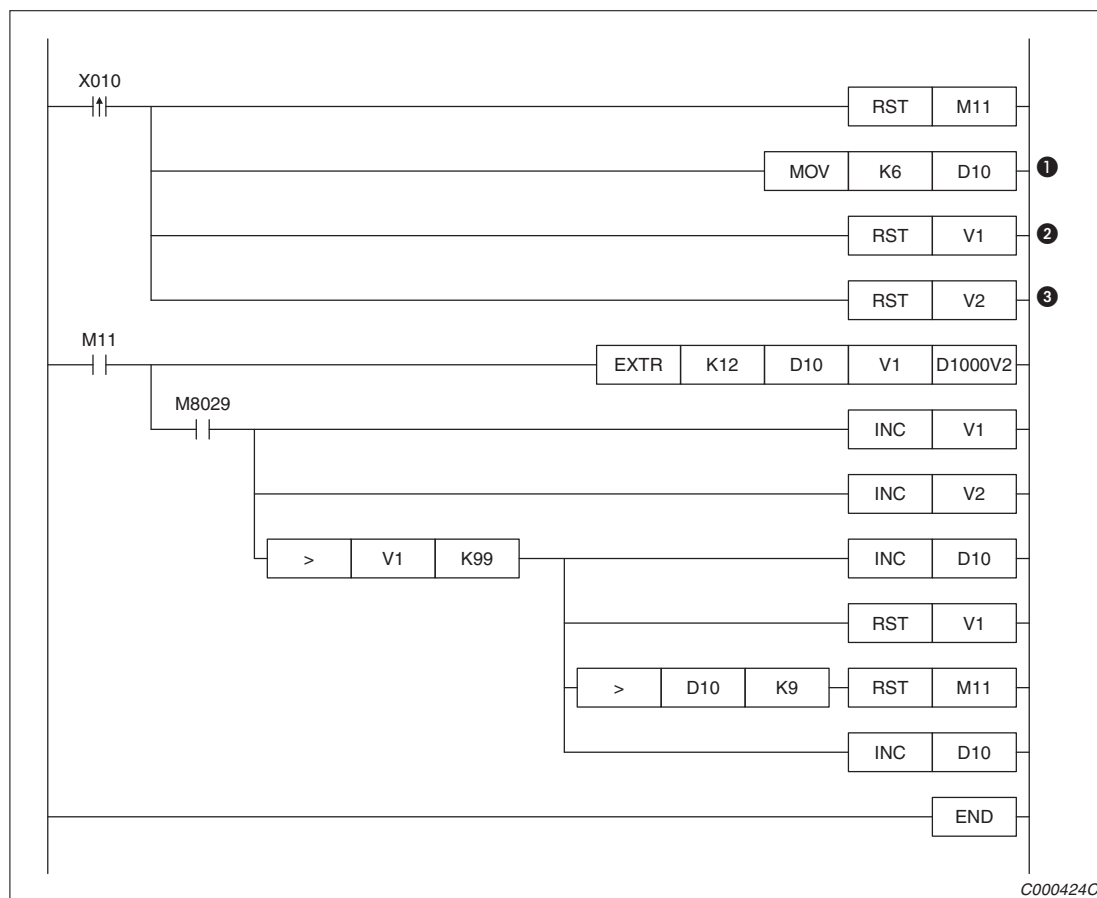


Рис. 7-173:Пример программы чтения параметров нескольких преобразователей

- ❶ Регистр D10 - номер станции
- ❷ Номер параметра
- ❸ Значение параметра

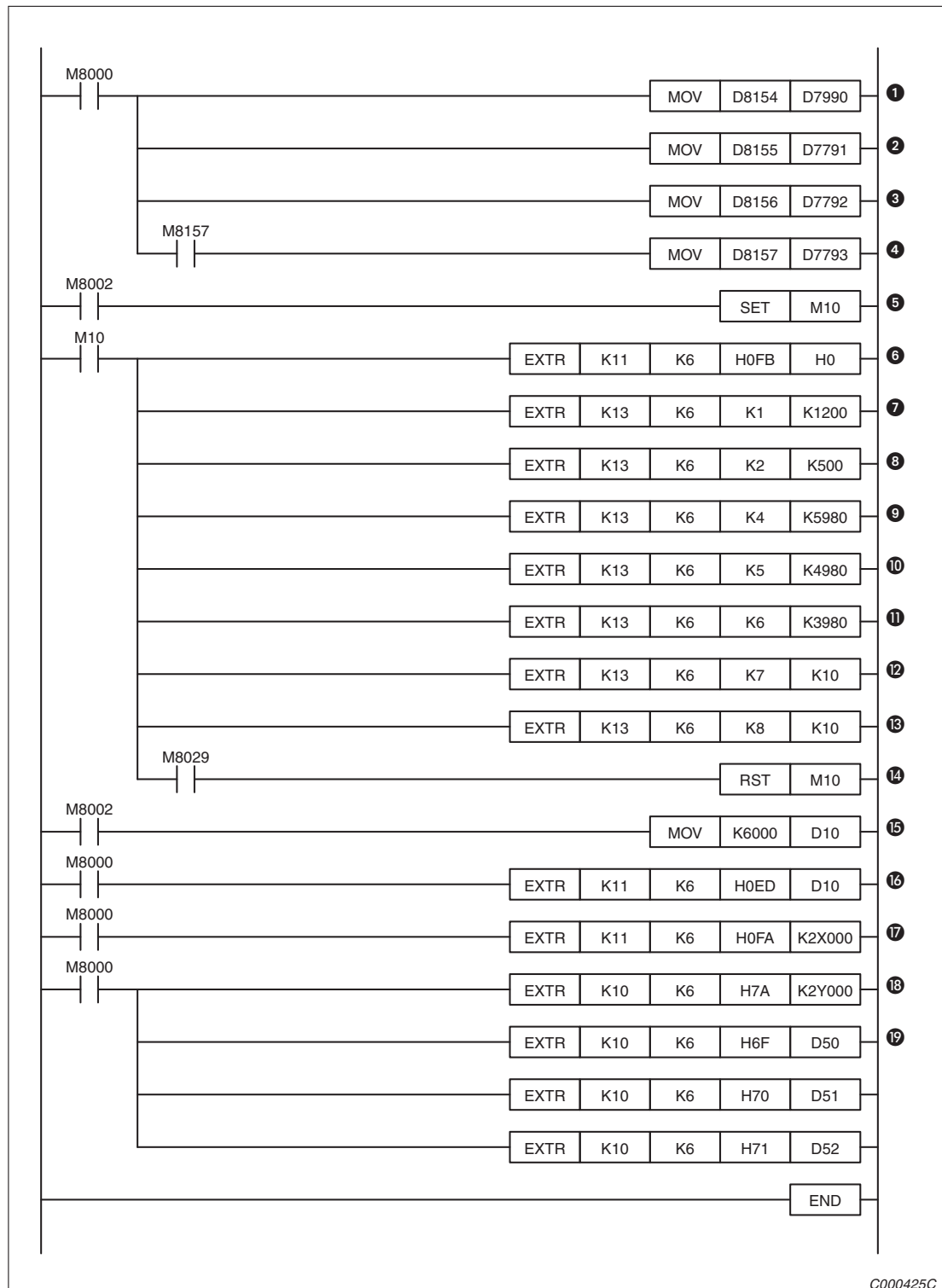
ПРИМЕР ▾

Пример программы 3

Программа записывает параметр скорости из контроллера в преобразователь частоты, выполняет вращение вперед по входу X1 и реверс по входу X2.

Частота преобразователя частоты может быть изменена перезаписью D10.

Программа также отображает частоту и выходной ток преобразователя частоты.



C000425C

Рис. 7-174:Пример программы по управлению и диагностике преобразователя частоты

В цепочках программы с 1 по 4, сохраняются коды ошибок для более поздней оценки в списке данных.

- ① Ожидание ответа от преобразователя частоты
- ② Номер шага текущей выполняемой инструкции
- ③ Код ошибки
- ④ Шаг в котором ошибка произошла впервые
- ⑤ Запись параметров в преобразователь частоты при пуске
- ⑥ Определяется режим связи по порту
- ⑦ Макс. частота
- ⑧ Мин. частота
- ⑨ Скорость 3
- ⑩ Скорость 2
- ⑪ Скорость 1
- ⑫ время разгона
- ⑬ время торможения
- ⑭ Сбрасывает M10 при завершении выполнения
- ⑮ Установка значения частоты по умолчанию
- ⑯ Перезапись значения частоты
- ⑰ Подача команды на преобразователь частоты
- ⑱ Отображение состояния
- ⑲ Отображение частоты

△

ПРИМЕР ▾

Пример программы 4

Если функция мониторинга (EXTR K10) запускается как в примере программы 3, изменение частоты или команда управления ожидает окончания выполнения мониторинга, что уменьшает скорость реакции преобразователя на команду управления.

В следующей программе функция мониторинга прерывается, если данные должны передаваться в преобразователь частоты. После окончания передачи данных в преобразователь частоты функция мониторинга продолжается.

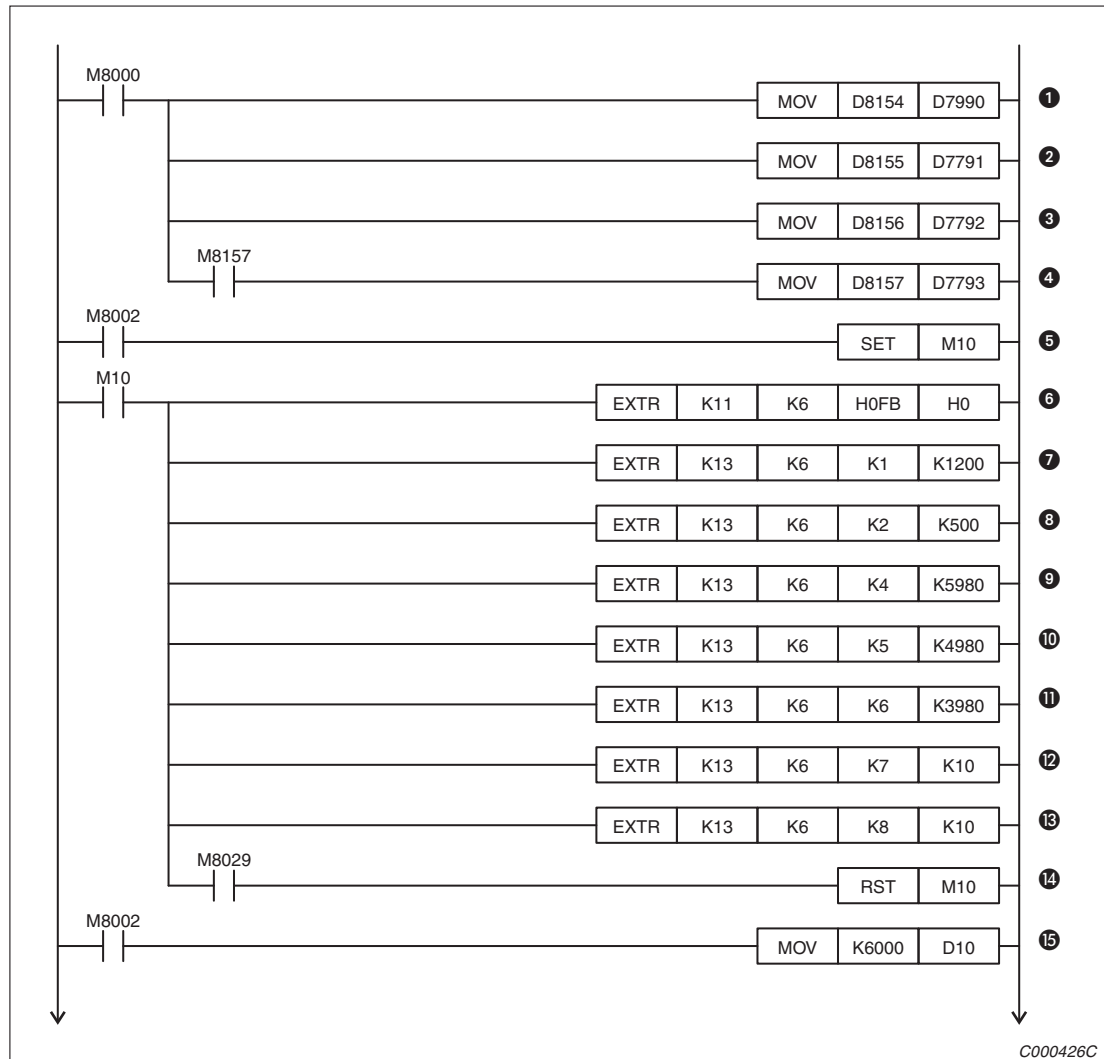
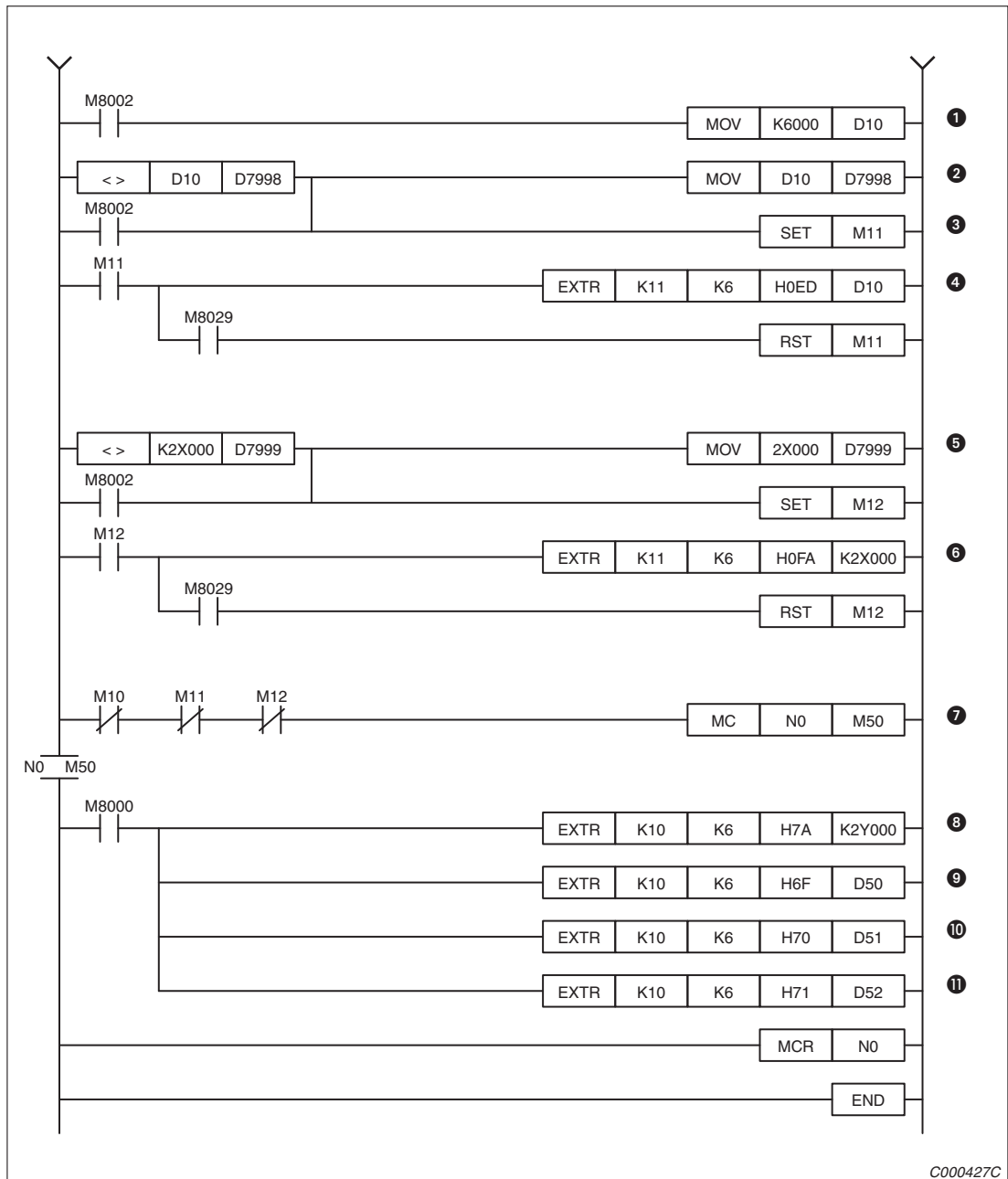


Рис. 7-175:Пример программы с приостановкой функции мониторинга (часть1)

В цепочках программы с 1 по 4, сохраняются коды ошибок для более поздней оценки в списке данных.

- ① Ожидание ответа от преобразователя частоты
- ② Номер шага текущей выполняемой инструкции
- ③ Код ошибки
- ④ Шаг в котором ошибка произошла впервые
- ⑤ Запись параметров в преобразователь частоты при пуске
- ⑥ Определяется режим связи по порту
- ⑦ Макс. частота
- ⑧ Мин. частота
- ⑨ Скорость 3
- ⑩ Скорость 2
- ⑪ Скорость 1
- ⑫ время разгона
- ⑬ время торможения
- ⑭ Сбрасывает M10 при завершении выполнения



C000427C

Рис. 7-176: Пример программы с приостановкой функции мониторинга (часть 2)

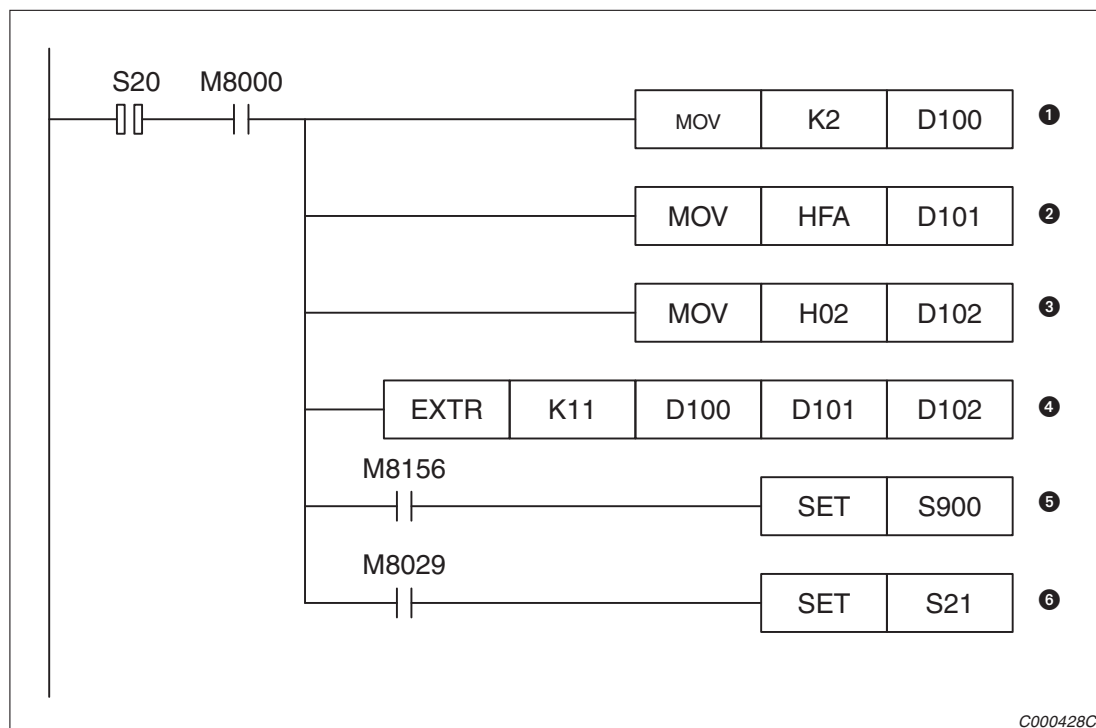
- ① Выполняет запись только если D10 изменен
- ② Извлечение для определения изменений
- ③ Запись частоты
- ④ Выполняется запись в преобразователь частоты, только если состояние одного из X0..X7 изменено.
- ⑤ Извлечение для определения изменений
- ⑥ Отображение состояния, если в преобразователь частоты ничего не записывается.
- ⑦ Макс. частота
- ⑧ Мин. частота
- ⑨ Скорость 3
- ⑩ Скорость 2
- ⑪ Скорость 1



ПРИМЕР ▾

Пример программы 5

При использовании инструкции STL

**Рис. 7-177:**Пример с вызовом функции EXTR в пошаговой программе

- ① Определение станции №2
- ② Код инструкции для рабочей команды
- ③ Команда вперед
- ④ Передача/прием команды в/из преобразователя частоты.
- ⑤ Переход в состояние обработки ошибок при возникновении ошибки
- ⑥ Переход в следующее состояние при нормальном получении данных.


△

7.14 РАЗНЫЕ КОМАНДЫ

Символ	FNC	Назначение	Разд
COMRD	182	Считывание комментария к операнду	7.14.1
RND	184	Генерирование случайного числа	7.14.2
DUTY	186	Тактовый генератор с регулируемой скважностью импульсов	7.14.3
CRC	188	Проверка данных (проверка CRC)	7.14.4
DHSMOV	189	Передача фактического значения высокоскоростного счетчика	7.14.5

Табл. 7-50: Обзор команд с различными функциями

7.14.1 СЧИТЫВАНИЕ КОММЕНТАРИЯ К ОПЕРАНДУ (COMRD)

		COMRD		FNC 182			
		Считывание комментария к операнду					
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U	
						●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	X, Y, M, S, T, C, D, R	T, C, D, R	●	16бита	32бита	COMRD COMRDP	5

Функция

Команда COMRD считывает комментарий к операнду и сохраняет его в виде кода ASCII. Комментарий к операнду вводится при программировании и с помощью программы может быть передан в контроллер.

Описание

- В (S+) указывается операнд, комментарий к которому требуется считать.
- В (D+) указывается первый операнд области, в которой требуется записать текст в кодировке ASCII.
- Можно записать максимум 16 знаков. Один знак занимает 8 битов.

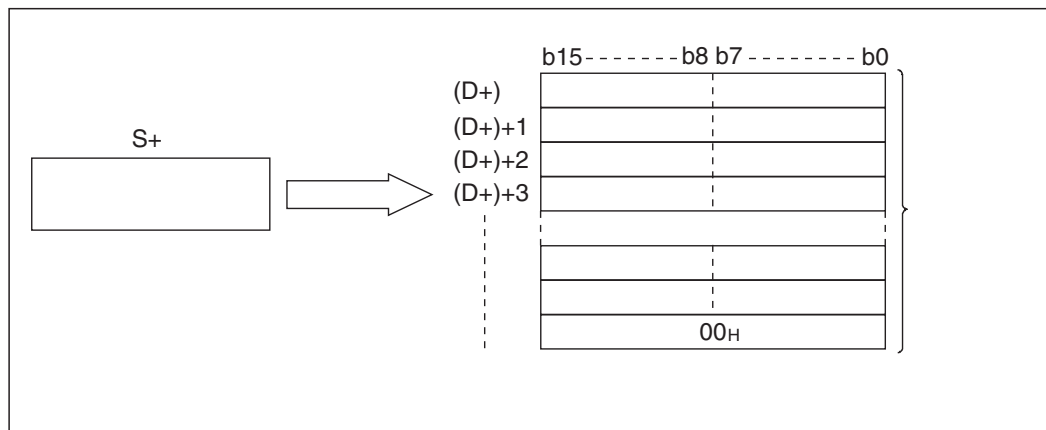


Рис. 7-178: Состояние меркера M8091 определяет, должен ли после текста в кодировке ASCII добавляться код "00H".

- Содержимое слова или байта после последнего знака ASCII зависит от состояния специального маркера M8091 и количества знаков (четное или нечетное):
 - M8091 = „0“
В случае нечетного количества знаков в старшем байте операнда, в котором был записан последний знак, вводится “00H”. В случае четного количества знаков “00H” вводится в следующем операнде после последнего знака.
 - M8091 = „1“
В случае нечетного количества знаков содержимое старшего байта операнда, в котором был записан последний знак, не изменяется. В случае четного количества знаков после последнего знака ASCII значение “00H” не добавляется.

На рисунке ниже изображен пример, при котором M8091 установлен на “1”, и комментарий состоит из нечетного количества знаков.

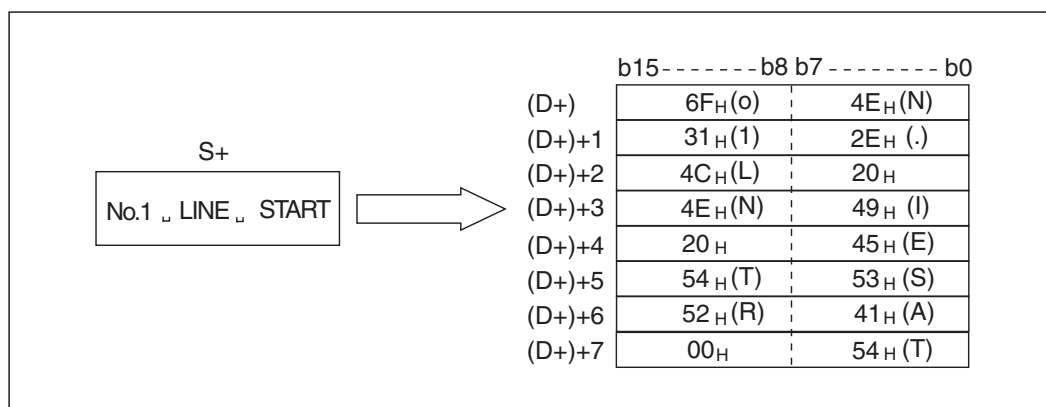


Рис. 7-179: В этом примере в байте после последнего знака вводится “00H”.

- Если указанный в (S+) операнд не имеет комментария, регистры с (D+) по ((D+)+8) заполняются 16 знаками пробела (20H).

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки “6706”:

- Указанный в (S+) операнд не имеет комментария.
- В результате сохранения знаков начиная с адреса (D+) превышает допустимая область для указанного операнда.

ПРИМЕР ▾

Для регистра данных D100 был введен комментарий “Target Line A”. Следующий фрагмент программы считывает этот комментарий при включении X10 и сохраняет его начиная с адреса D0. Перед считыванием комментария сбрасывается маркер M8091. В результате этого после комментария добавляется знак “NUL” (00H).

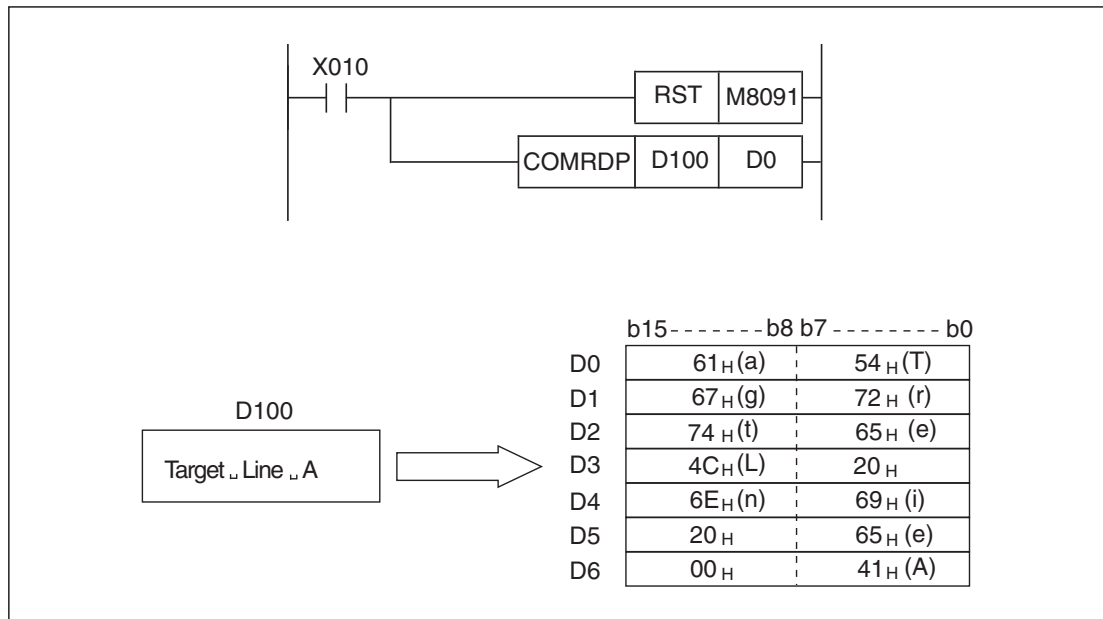


Рис. 7-180: Комментарий к операнду D100 записывается в виде знаков ASCII в регистры с D0 по D6.

7.14.2 ГЕНЕРИРОВАНИЕ СЛУЧАЙНОГО ЧИСЛА (RND)

		RND		FNC 183				
		Генерирование случайного числа						
Операнды	D+	Имп. инструкция(P)	CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●
			Обработка		Шаги программы			
			16бита	32бита	RND	RNDP	3	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	●	●					

Функция

Выработка случайного числа в диапазоне от 0 до 32767.

Описание

- Команда RND генерирует случайное число и сохраняет его в операнде, указанном в (D+).
- В качестве источника для расчета случайного числа используются специальные регистры D8311 и D8310. После включения контроллера в эти регистры вводится значение "1". После перехода в режим "RUN" в регистры D8311 и D8310 можно ввести любое значение между 0 и 2.147.483.647. Этот перенос должен выполняться только один раз после запуска контроллера.

ПРИМЕР ▾

Следующий пример программы при каждом включении X10 вырабатывает случайное число и сохраняет его в D100.

После перехода из режима "STOP" в режим "RUN" из внутренних часов контроллера считывается время суток и дата. Затем эти данные преобразуются по формуле "{((год + месяц) x день) + время суток}", которая служит основанием для расчета случайного числа.

Маркер M8002 установлен только в первом цикле после включения контроллера.

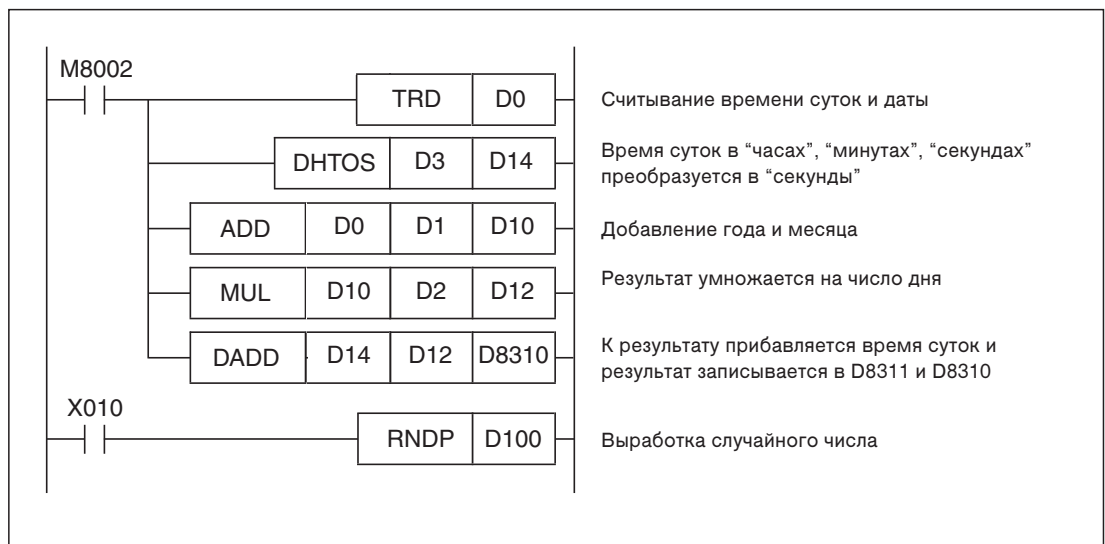


Рис. 7-181: В этом примере из времени суток и даты образуется основа для генерирования случайного числа.

7.14.3 ТАКТОВЫЙ ГЕНЕРАТОР С РЕГУЛИРУЕМОЙ СКВАЖНОСТЬЮ ИМПУЛЬСОВ (DUTY)

				DUTY		FNC 186		
				Тактовый генератор с регулируемой скважностью импульсов				
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U		
							●	
Операнды	n1	n2	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	T, C, D, R, K, H				M*	16бита	32бита	DUTY
					●			

* Можно указать только один маркер из области M8330...M8334.

Функция

Специальные маркеры циклически устанавливаются на определенное количество программных циклов и сбрасываются.

Описание

- Команда DUTY включает указанный в (D+) операнд на количество программных циклов, указанное в (n1), и выключает его на количество программных циклов, указанное в (n2).
- В (D+) можно указать только один маркер из области M8330...M8334.
- Количество программных циклов для M8330...M8334 подсчитывается в специальных регистрах D8330...D8334. Счетчик сбрасывается, если значение достигло "(n1) + (n2)" или если включается входное условие для команды DUTY.
- Выполнение команды DUTY начинается при включении входного условия. Операнд в (D+) устанавливается или сбрасывается в конце программного цикла при обработке команды END. Даже если входное условие становится ложным, выполнение команды DUTY не заканчивается. Лишь после того, как контроллер перешел в состояние "STOP" или было выключено напряжение питания, эта команда перестает выполняться.
- Если значение в (n1) = 0, операнд в (D+) постоянно остается выключенным.
- Если значение в (n2) = 0, операнд в (D+) постоянно включен.

УКАЗАНИЕ

Так как для использования в качестве операнда команды DUTY имеются только 5 специальных маркеров, в программе можно использовать максимум 5 команд DUTY.

Специальный маркер, используемый в команде DUTY, нельзя использовать еще раз в качестве операнда другой команды DUTY.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- Для (n1) или (n2) указано отрицательное значение (код ошибки "6706").
- Для (D+) указан иной операнд кроме маркера из области M8330...M8334 (код ошибки "6705").

ПРИМЕР ▾

В следующем примере программы маркер M8330 включается на время одного программного цикла, а затем выключается на 3 программных цикла.

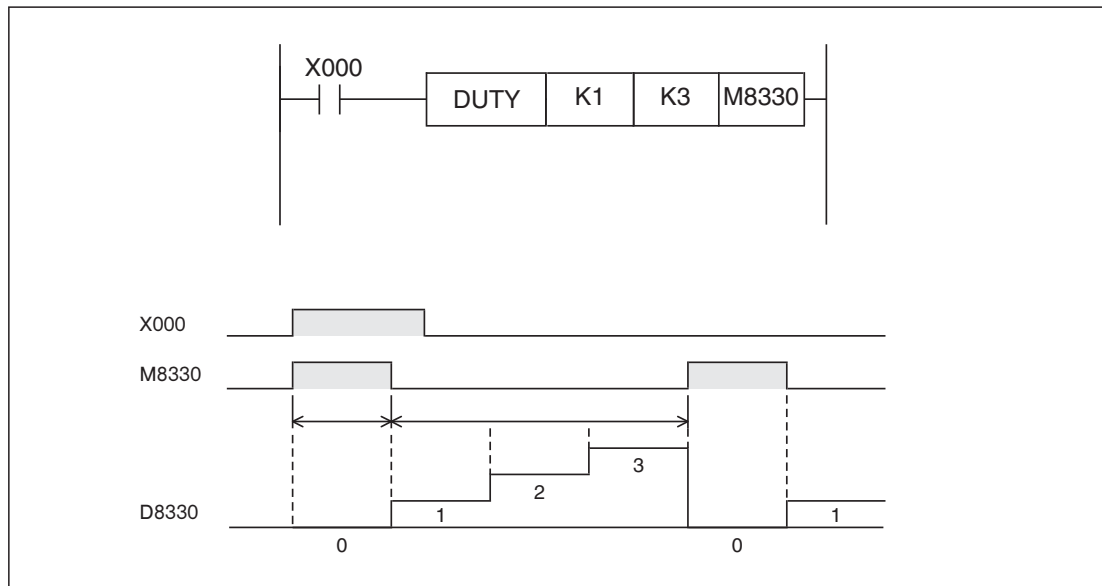


Рис. 7-182:Пример применения команды DUTY

7.14.4 ПРОВЕРКА ДАННЫХ (CRC)

				CRC		FNC 188				
				Проверка данных						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы			
	KnX*, KnY*, KnM*, KnS*, T, C, D, R, специальные модули (U/G)	KnY*, KnM*, KnS*, T, C, D, R, специальные модули (U/G)	D, R, K, H	●	16бита	32бита	CRC	CRCP	7	

При указании битовых операндов следует применять коэффициент К “4” (например, K4M0).

Функция

Расчет суммы CRC для проверки данных

Описание

- Команда CRC рассчитывает контрольное значение для проверки по методу “Cyclic Redundancy Check”, с помощью которого можно распознавать ошибки при передаче данных. Проверяемые 8-битные данные хранятся в области операндов, первый адрес которой указывается в (S+). Результат проверки записывается в операнд, указанный в (D+). (n) задает количество проверяемых 8-битных данных (от 1 до 256).
- Для проверки применяется формула CRC-16:
 Значение CRC = $X^{16} + X^{15} + X^2 + 1$
- Для проверки команда CRC может извлекать из одного из операндов в (S+) одну или две 8-битных группы. Это управляется с помощью специального маркера M8161.

Функция при сброшенном специальном маркере M8161 (16-битный режим)

При 16-битном режиме проверяемые 8-битные группы попеременно извлекаются из младших и старших байтов операндов в (S+). Например, если в (S+) указан D100, а маркер M8161 имеет значение "0", извлекаются данные младшего байта D100, затем старшего байта D100, младшего байта D101, старшего байта D101 и так далее.

Результат записывается только в 16-битном операнде (D+).

Функция при установленном специальном маркере M8161 (8-битный режим)

Проверяемые 8-битные группы в 8-битном режиме извлекаются только из младших байтов операндов, указанных в (S+).

Результат записывается в два 16-битных операнда. Младший байт (D+) содержит младший байт, а младший байт ((D+)+1) - старший байт результата.

УКАЗАНИЕ

Специальный маркер M8161 влияет также на поведение команд RS, ASCII, HEX и ASC. Если одна из этих команд применяется в той же программе, что и команда CRC, то перед выполнением команды CRC следует установить M8161, а сразу после выполнения команды CRC снова сбросить этот маркер.

Команда CRC рассчитывает контрольное значение CRC по полиному CRC-16. Кроме того, имеются и другие стандартные методы проверки, например:

CRC-12: $X^{12} + X^{11} + X^3 + X^2 + X + 1$

CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$

При этих методах проверки образуются совершенно иные значения, чем при проверке CRC-16. Поэтому убедитесь в том, что для вашего применения можно использовать полином CRC-16.

Помимо метода CRC, для распознавания ошибок коммуникации применяются проверка по четности и проверка суммы. Для этого в контроллерах MELSEC FX можно использовать команду CCD (раздел 7.4.5).

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- При указании битовых операндов в (S+) или (D+) не был использован коэффициент K 4.
- Для (n) было указано значение вне допустимого диапазона от 1 до 256.
- Операнд ((S+)+(n-1)) или операнд ((D+)+1) превышает допустимую область для операндов указанного типа.

ПРИМЕР ▾

В следующих примерах программ из знаков "0123456", записанных в кодировке ASCII начиная с регистра данных D100, вычисляется контрольное значение CRC. Результат сохраняется начиная с D0.

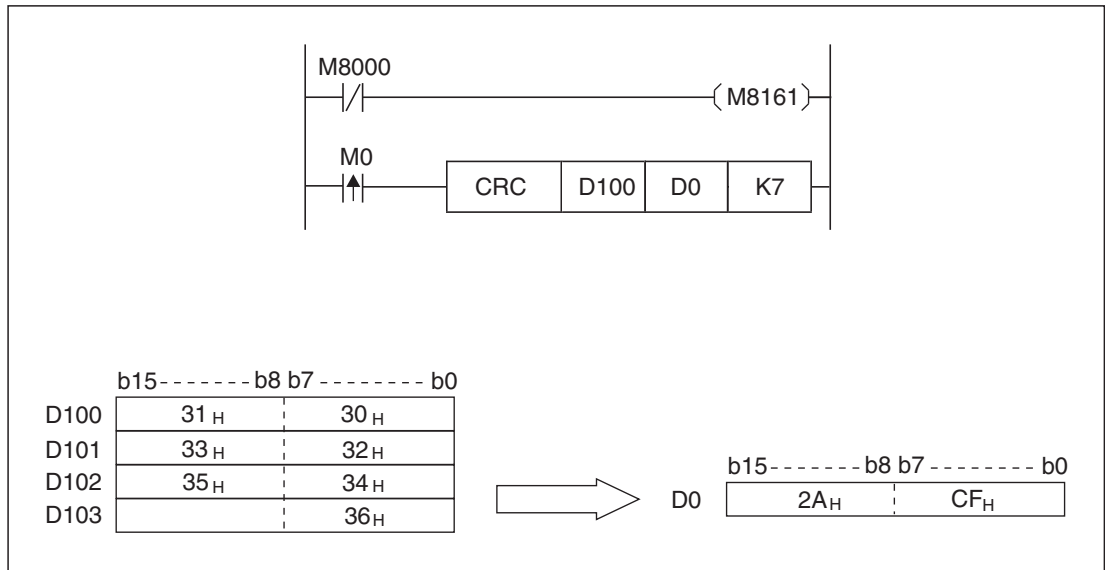


Рис. 7-183:Обработка данных в 16-битном режиме (M8161 не установлен)

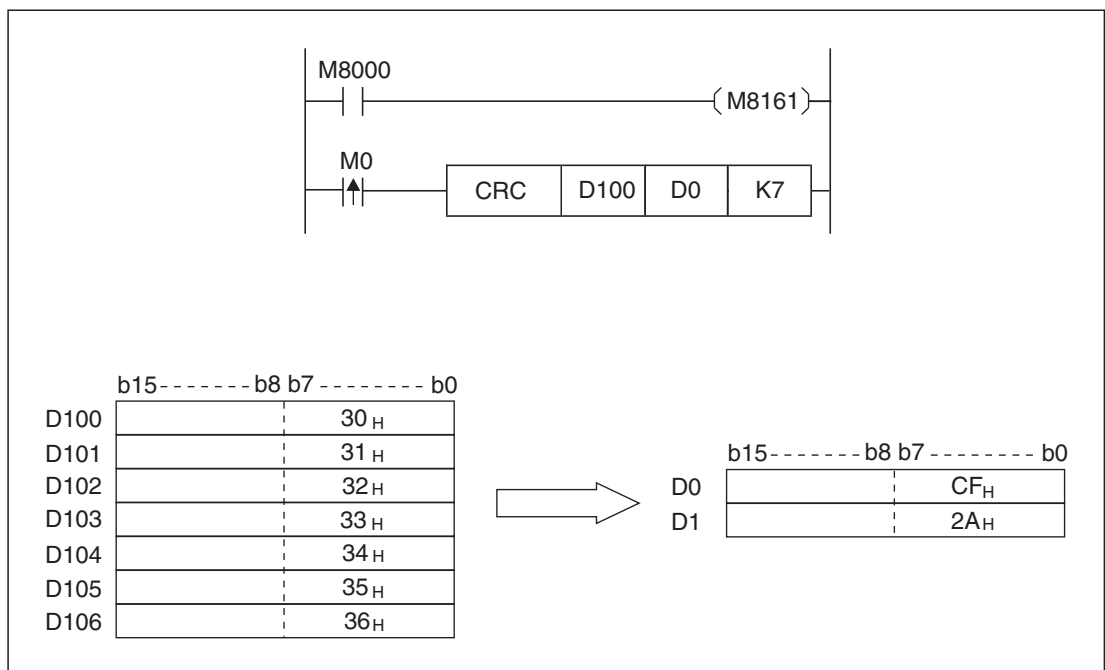
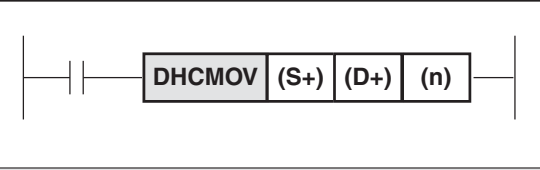


Рис. 7-184:Обработка данных в 8-битном режиме (M8161 установлен)

7.14.5 ПЕРЕДАЧА ФАКТИЧЕСКОГО ЗНАЧЕНИЯ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА (DHCMOV)

				DHCMOV		FNC 189			
				Передача фактического значения высокоскоростного счетчика					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	C*, D*	D, R	K, H		16бита	32бита	DHCMOV		13
						●			

Могут быть указаны только высокоскоростные счетчики C235...C255 или кольцевые счетчики D8099 и D8398.

Функция

Перенос фактического значения высокоскоростного счетчика или кольцевого счетчика

Описание

- Фактическое значение указанного в (S+) высокоскоростного счетчика или кольцевого счетчика передается в операнд, указанный в (D+).
- Для (S+) можно указывать только высокоскоростные счетчики C235...C255, а также кольцевые счетчики D8099 и D8398.
- Фактические значения счетчиков C235...C255 записываются в ((D+)+1) и (D+).
- Фактическое значение 16-битного кольцевого счетчика D8099 записывается в (D+).
- Если в (D+) указан 32-битный кольцевой счетчик D8398, в ((D+)+1) записывается содержимое D8399, а в (D+) - содержимое D8398.
- С помощью (n) можно указать, требуется ли стирать фактическое значение счетчика после переноса:
 - (n) = 0: не стирать фактическое значение
 - (n) = 1: после считывания стереть фактическое значение счетчика

Применение команды DHCMOV

При сигнале на входе высокоскоростной счетчик (C235...C255) считает вверх или вниз. В случае аппаратного счетчика фактическое значение обновляется, если в отношении счетчика выполняется команда OUT, т. е. обновление фактического значения зависит от программного цикла. Фактическое значение программного счетчика обновляется при каждом входном импульсе.

Если фактическое значение считано с помощью команды передачи (например, MOV), то может оказаться, что считано не самое последнее фактическое значение. Применяя команду DHCMOV в сочетании с прерыванием, активируемым входом счета, можно считывать текущее фактическое значение счетчика по возрастающему или падающему фронту входного сигнала.

Если команда DHCMOV запрограммирована непосредственно перед командой сравнения (CMP, ZCP или операторами сравнения с логическими связями И/ИЛИ), то при сравнении применяется текущее фактическое значение. Сравнение с помощью команд CMP или ZCP или операторов сравнения с логическими связями И/ИЛИ имеет преимущество относительно операторов сравнения для высокоскоростного счетчика, которое заключается в том, что аппаратный счетчик не рассматривается как программный счетчик. Аппаратные счетчики могут регистрировать более высокие частоты, чем программные счетчики. Если требуется сравнивать фактическое значение высокоскоростного счетчика и активировать какой-либо выход, как только фактическое значение изменилось, следует применять операторы сравнения для высокоскоростного счетчика (HSCS, HSCR, HSZ).

Команду DHCMOV можно использовать в программе сколько угодно часто.

При выполнении команды DHCMOV в программе прерывания учитывайте следующие указания:

- В главной программе требуется деблокировать прерывания с помощью команды EI (раздел 6.2.4). Главную программу необходимо завершить командой FEND (раздел 6.2.5). При работе в редакторе I программного обеспечения GX IEC Developer этого делать не требуется.
- Если команда DHCMOV программируется в первой строке программы прерывания, следует применять специальный маркер M8394:

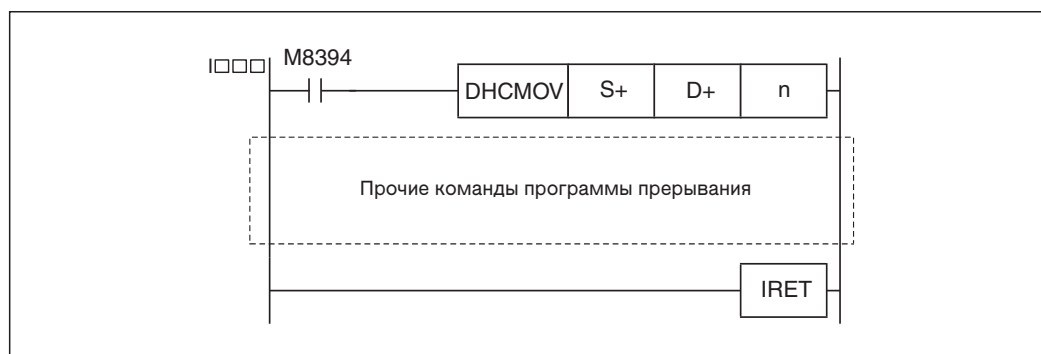


Рис. 7-185: В программе прерывания команда DHCMOV управляется с помощью специального маркера M8394.

- Если в программе прерывания используются несколько команд DHCMOV, то специальный маркер M8394 управляет только первой командой DHCMOV, следующей непосредственно за указателем прерывания.

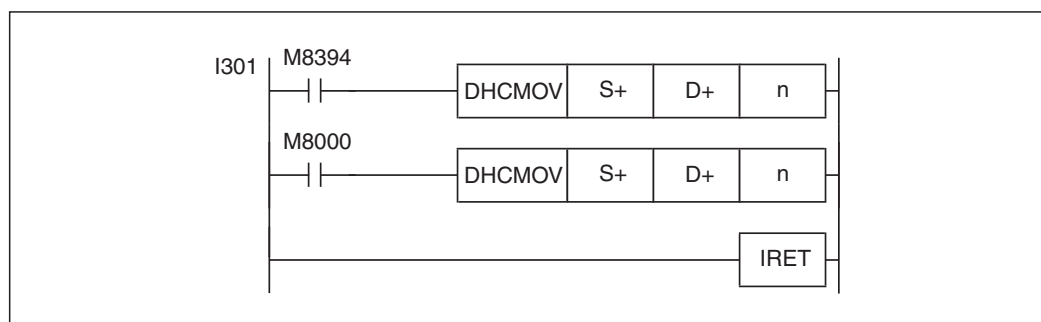


Рис. 7-186: В этом примере из-за указателя прерывания I301 при включении входа X3 сначала выполняется первая команда DHCMOV. Затем обрабатывается вторая команда DHCMOV.

- Один и тот же счетчик нельзя указывать для команд DHCMOV в нескольких программах прерывания.

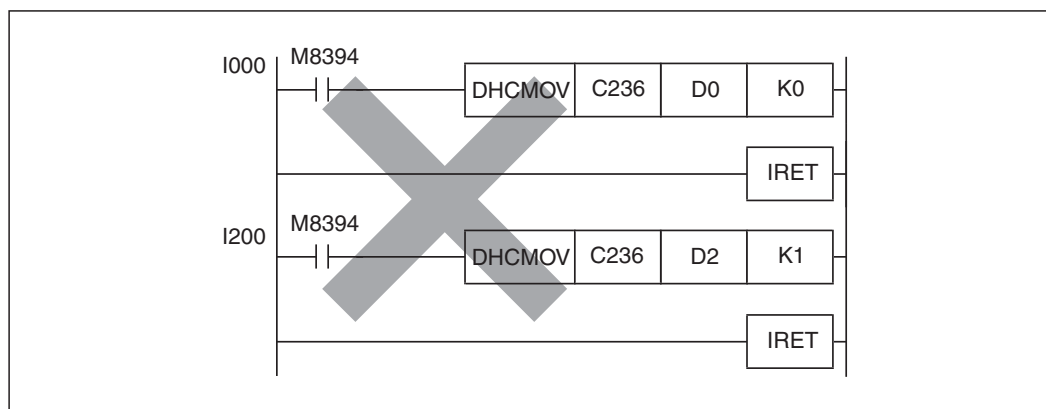


Рис. 7-187: В этом примере команды DHCMOV два раза обращаются к одному и тому же счетчику. Это недопустимо.

- Прерывания, активируемые через входы, можно заблокировать с помощью специальных маркеров. В результате этого не выполняются и соответствующие программы прерывания и запрограммированные в них команды.

Специальный маркер	Значение	Вход
M8050	Заблокировать программы прерывания I000 и I001	X000
M8051	Заблокировать программы прерывания I100 и I101	X001
M8052	Заблокировать программы прерывания I200 и I201	X002
M8053	Заблокировать программы прерывания I300 и I301	X003
M8054	Заблокировать программы прерывания I400 и I401	X004
M8055	Заблокировать программы прерывания I500 и I501	X005

Табл. 7-51: С помощью специальных маркеров M8050...M8055 можно заблокировать программы прерывания.

Для блокировки программы прерывания необходимо установить соответствующий специальный маркер. Специальные маркеры с M8050 по M8055 сбрасываются при переводе контроллера в режим "STOP".

- Если активируется прерывание, но исполнение соответствующей программы прерывания заблокировано по иной причине, кроме установленного специального маркера M8050...M8055, то выполняется только команда DHCMOV в начале программы прерывания, а другие команды не выполняются. Например, это происходит, если в программе была выполнена команда DI. Лишь после выполнения команды EI (раздел 6.2.4) прерывания снова деблокируются и программы прерывания обрабатываются полностью.

Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6705":

- Операнд в (S+) или (D+) превышает допустимую область для операндов указанного типа.

ПРИМЕР ▾

В следующем примере фактическое значение высокоскоростного счетчика C235 в каждом программном цикле сравнивается с заданным значением. Если фактическое значение больше или равно "500", устанавливается выход Y000. После считывания фактическое значение C235 не стирается ((n) = "0").

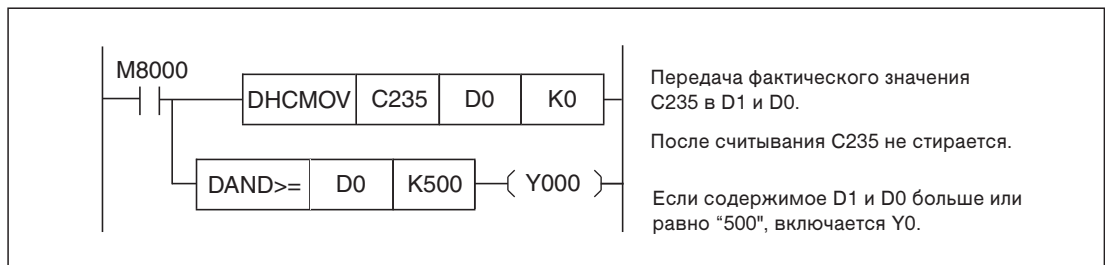


Рис. 7-188:Пример выполнения команды DHCMOV в циклической программе



ПРИМЕР ▾

В этом примере программы прерывания фактическое значение высокоскоростного счетчика C235 передается в регистры данных D201 и D200, а затем фактическое значение C235 стирается ((n) = "1").

Программа прерывания выполняется при включении входа X001 (указатель прерывания I101).

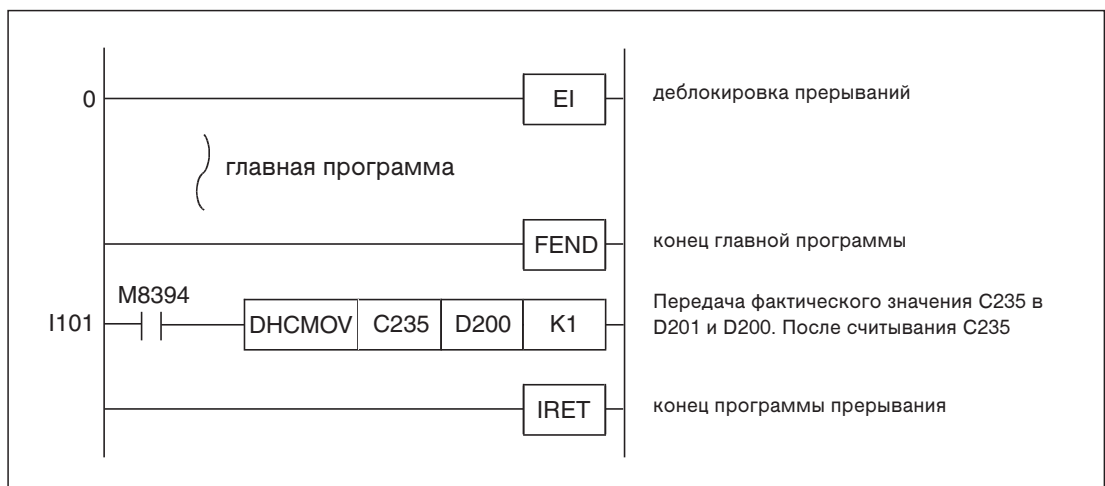


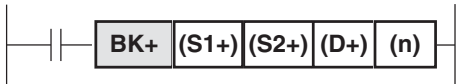
Рис. 7-189:Пример выполнения команды DHCMOV в программе прерывания

7.15 КОМАНДЫ ДЛЯ БЛОКОВ ДАННЫХ

Символ	FNC	Назначение	Разд
ВК+	192	Сложение данных двух блоков	7.15.1
ВК-	193	Определение разности данных двух блоков	7.15.2
ВКСМР=	194	Сравнение данных в блоках на "равно"	7.15.3
ВКСМР>	195	Сравнение данных в блоках на "больше"	
ВКСМР<	196	Сравнение данных в блоках на "меньше"	
ВКСМР<>	197	Сравнение данных в блоках на "неравно"	
ВКСМР<=	198	Сравнение данных в блоках на "меньше или равно"	
ВКСМРА>=	199	Сравнение данных в блоках на "больше или равно"	

Табл. 7-52: Обзор команд для блоков данных

7.15.1 СЛОЖЕНИЕ ДАННЫХ ДВУХ БЛОКОВ (BK+)

					BK+		FNC 192			
					Сложение данных двух блоков					FX1S
					CPU					●
Операнды	S1+	S2+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R	T, C, D, R, K, H	T, C, D, R	D, R, K, H		●	16бита	32бита	BK+ BK+P	9
						●	●	DBK+ DBK+P	17	

Функция

Данные из двух взаимосвязанных областей (блоков данных) суммируются и результат записывается в другой блок данных.

Описание

- В (S1+) указывается первый адрес 1-й исходной области операндов.
- В (S2+) указывается первый адрес 2-й исходной области операндов или константа.
- Результат сложения вводится в области, 1-й адрес которой указан в (D+).
- Размер областей (S1+), (S2+) и (D+) указывается с помощью (n).
- Эта команда суммирует не содержимое одного блока данных, а два различных операнда либо содержимое операнда и константу. При 16-битной обработке константа может находиться в диапазоне от -32768 до 32767, а при 32-битной обработке - в диапазоне от -2.147.483.648 до 2.147.483.647.

16-битная обработка

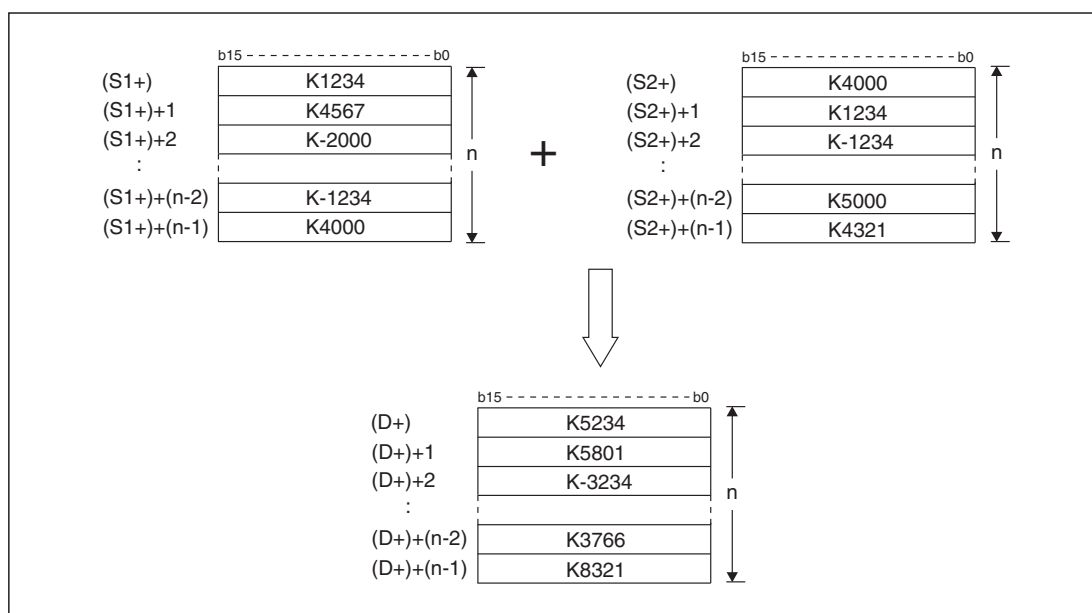


Рис. 7-190: Суммирование содержимых двух блоков данных

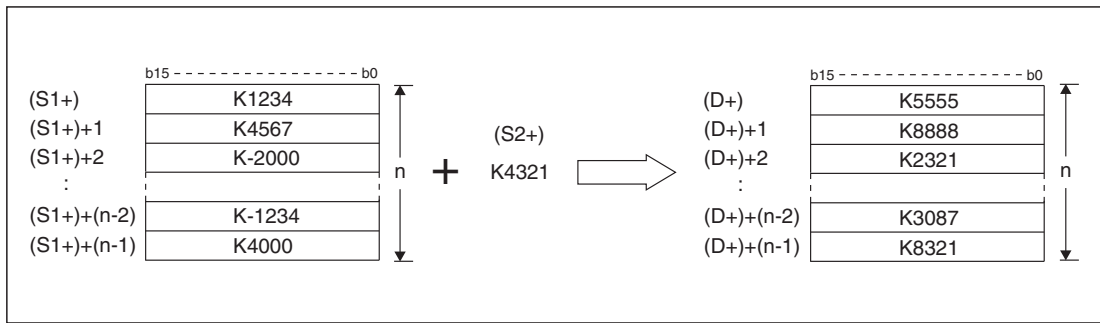


Рис. 7-193: Прибавление константы к содержимому блока данных

32-битная обработка

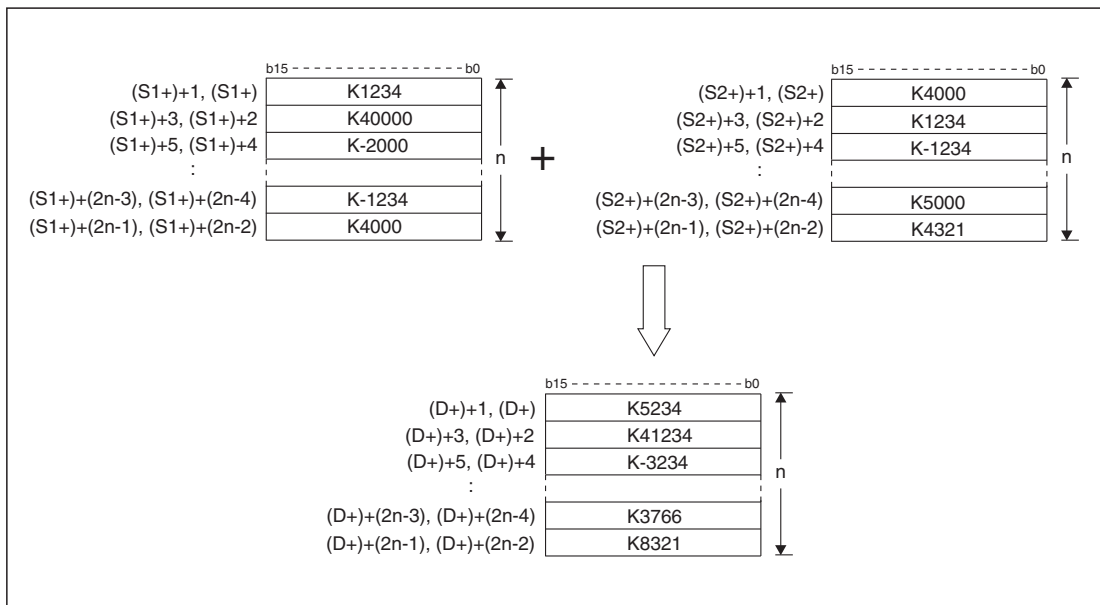


Рис. 7-191: Суммирование содержимых двух блоков данных, состоящих из 32-битных операндов

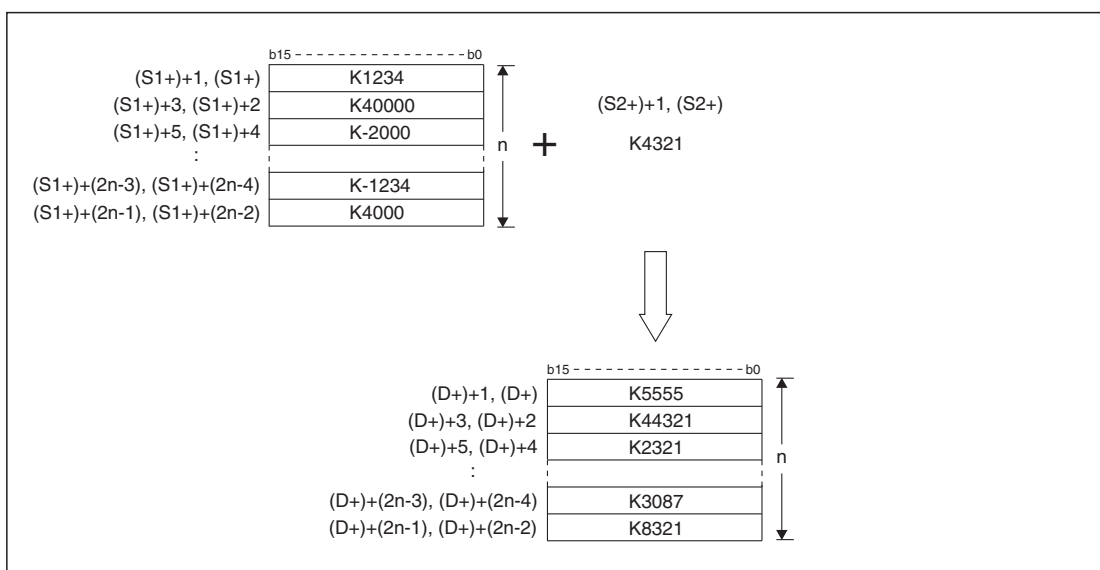


Рис. 7-192: Прибавление константы к содержимому блока данных, состоящему из 32-битных операндов

УКАЗАНИЕ

Если результат сложения занижает или превышает допустимый диапазон значений, записываются следующие значения. Флаг переноса в этих случаях не устанавливается.

16-битная обработка:

32767 (7FFF_H) + 2 (0002_H) → -32767 (8001_H)

-32767 (8000_H) + (-2) (FFFF_H) → 32766 (7FFE_H)

32-битная обработка:

2.147.483.647 (7FFFFFFF_H) + 2 (00000002_H) → -2.147.483.647 (80000001_H)

-2.147.483.648 (80000000_H) + (-2) (FFFFFFF_H) → 2.147.483.646 (7FFFFFFE_H)

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- В результате указания (n) (2n при 32-битной обработке) превышает допустимый диапазон для типа операнда, указанного в (S1+), (S2+) или (D+).
- Диапазоны, указанные с помощью (S1+) или (S2+) и (n) (2n при 32-битной обработке), перекрываются с диапазоном, установленным с помощью (D+).

ПРИМЕР

Следующая программа прибавляет содержимые регистров данных с D100 по (D100+n) к содержимым регистров данных с D150 по (D150+n) и сохраняет результаты, начиная с регистра D200. Количество операндов диапазона (n) указывается в D0.

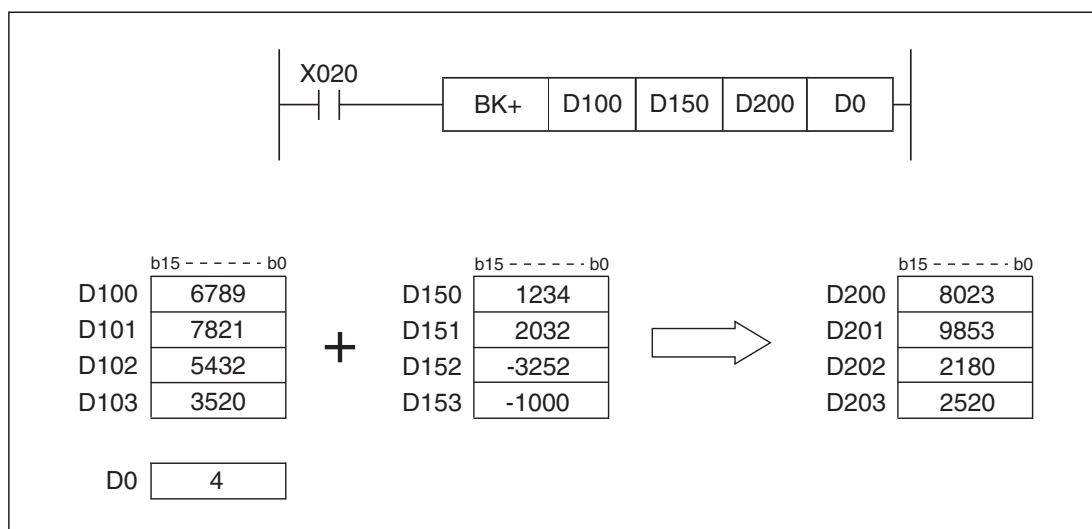


Рис. 7-194: Прибавление константы к содержимому блока данных



7.15.2 ВЫЧИСЛЕНИЕ РАЗНОСТИ ДАННЫХ ДВУХ БЛОКОВ (BK-)

					BK+		FNC 192			
					Вычисление разности данных двух блоков					FX1S
					CPU					●
Операнды	S1+	S2+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R	T, C, D, R, K, H	T, C, D, R	D, R, K, H		●	16бита	32бита	BK- BK-P	9
						●	●	DBK- DBK-P	17	

Функция

Вычисляется разность данных из двух взаимосвязанных областей (блоков данных) и результат записывается в другой блок данных.

Описание

- В (S1+) указывается первый адрес области, содержащей уменьшаемое (т. е. значение, из которого вычитается другое значение).
- В (S2+) указывается константа или первый адрес области, содержащей вычитаемое.
- Разность записывается в области, 1-й адрес которой указан в (D+).
- Размер областей (S1+), (S2+) и (D+) указывается с помощью (n).
- Если в качестве вычитаемого указывается константа, то при 16-битной обработке она может находиться в диапазоне от -32768 до 32767, а при 32-битной обработке - в диапазоне от -2.147.483.648 до 2.147.483.647.

16-битная обработка

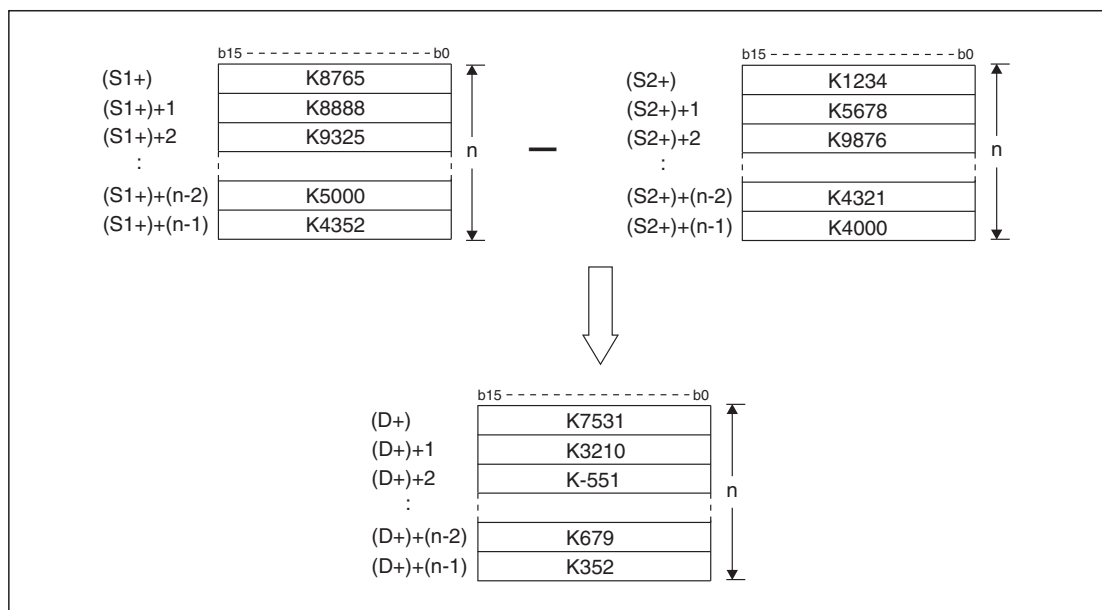


Рис. 7-195: Определение разности содержимых двух блоков данных

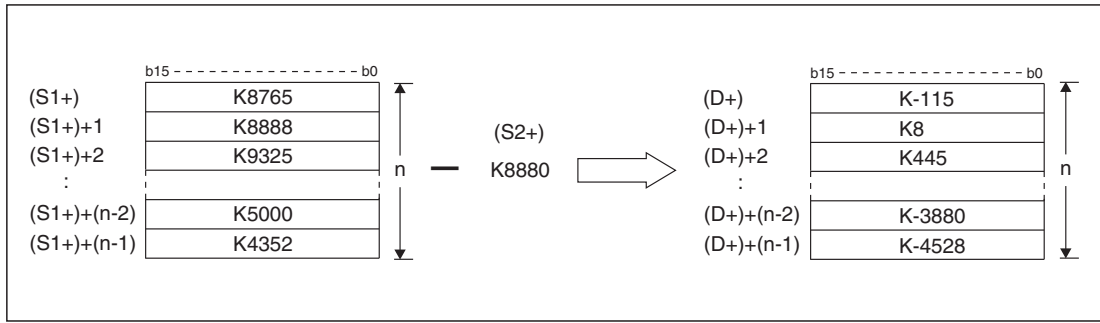


Рис. 7-196: Из содержимого блока данных вычитается константа

32-битная обработка

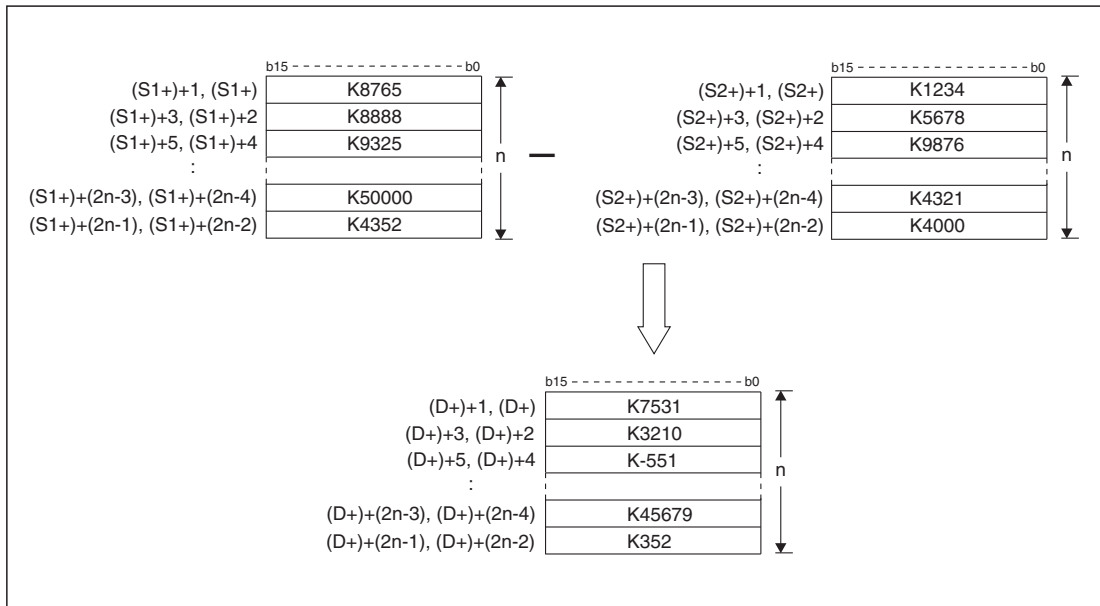


Рис. 7-197: Определение разности содержимых двух блоков данных, состоящих из 32-битных операндов

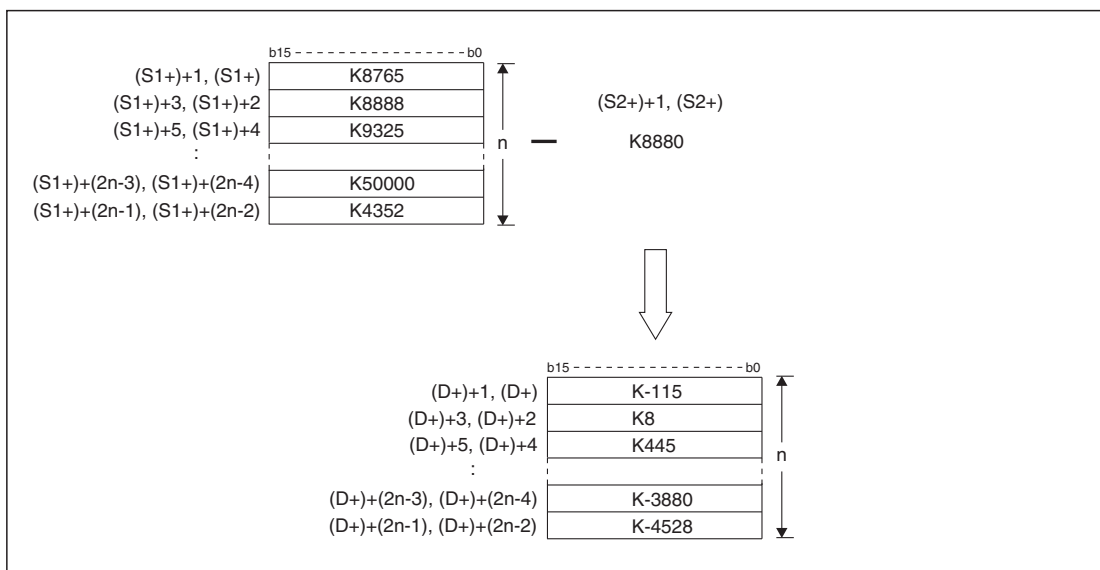


Рис. 7-198: Из содержимого блока данных, состоящего из 32-битных операндов, вычитается константа

УКАЗАНИЕ

Если разность занижает или превышает допустимый диапазон значений, записываются следующие значения. Флаг переноса в этих случаях не устанавливается.

16-битная обработка:

-32767 (8000H) - 2 (0002H) → 32766 (7FFEH)
 32767 (7FFFH) - (-2) (FFFEH) → -32767 (8001H)

32-битная обработка:

-2.147.483.648 (80000000H) - 2 (00000002H) → 2.147.483.646 (7FFFFFFEH)
 2.147.483.647 (7FFFFFFFH) - (-2) (FFFFFFFEH) → -2.147.483.647 (80000001H)

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- В результате указания (n) (2n при 32-битной обработке) превышает допустимый диапазон для типа операнда, указанного в (S1+), (S2+) или (D+).
- Диапазоны, указанные с помощью (S1+) или (S2+) и (n) (2n при 32-битной обработке), перекрываются с диапазоном, установленным с помощью (D+).

ПРИМЕР

В следующем примере программы при включении входа X10 из содержимых регистров данных с D100 по D102 вычитается значение "8765". Результаты сохраняются начиная с регистра D200.

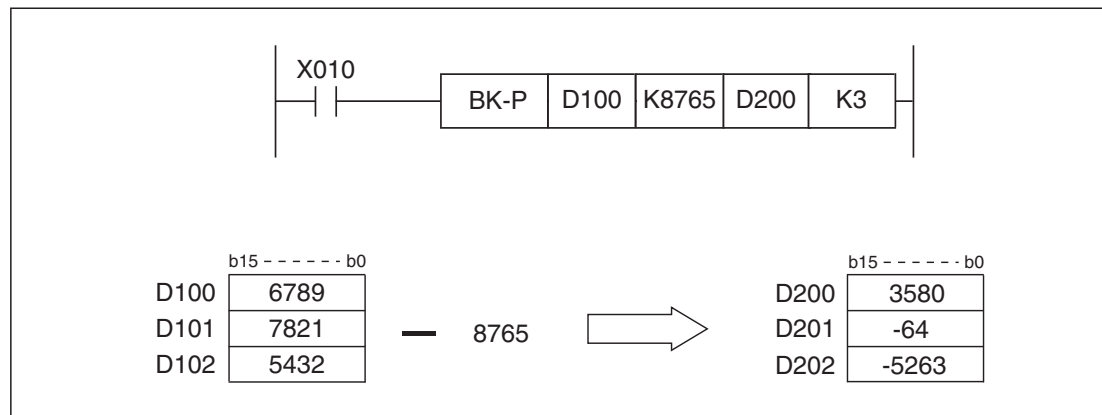


Рис. 7-199: Пример определения разности содержимых блоков данных

△

7.15.3 СРАВНЕНИЕ ДАННЫХ В БЛОКАХ ДАННЫХ (ВКСМР \square)

					ВКСМР \square		FNC 194 – 199				
					Сравнение данных в блоках данных					CPU	FX1S
Операнды	S1+	S2+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы			
	T, C, D, R, K, H	T, C, D, R	Y, M, S	D, R, K, H	●	16бита	32бита	ВКСМР \square ВКСМР \square P	9		
						●	●	ДВКСМР \square ДВКСМР \square P	17		

Функция

Данные двух взаимосвязанных областей (блоков данных) сравниваются друг с другом и результат сравнения записывается в другой блок данных.

Описание

- В (S1+) указывается первый адрес первой области, содержащей сравниваемые значения, или константа.
- В (S2+) указывается первый адрес области, которая сравнивается с областью, указанной в (S1+).
- Результат сравнения записывается в области, первый адрес которой указан в (D+). Если сравнение имеет результат "истинно", то в качестве результата записывается "1". Если сравнение имеет результат "ложно", записывается значение "0".
- Размер областей (S1+), (S2+) и (D+) указывается с помощью (n).
- Пустой квадрат в команде ВКСМР означает одну из операций сравнения =, >, <, <>, ≤ и ≥. В следующей таблице операции сравнения сопоставлены соответствующим номерам функций.

FNC	Команда	Истинно, если	Ложно, если
194	ВКСМР =	(S1+) = (S2+)	(S1+) <> (S2+)
195	ВКСМР >	(S1+) > (S2+)	(S1+) ≤ (S2+)
196	ВКСМР <	(S1+) < (S2+)	(S1+) ≥ (S2+)
197	ВКСМР <>	(S1+) <> (S2+)	(S1+) = (S2+)
198	ВКСМР ≤	(S1+) ≤ (S2+)	(S1+) > (S2+)
199	ВКСМР ≥	(S1+) ≥ (S2+)	(S1+) < (S2+)

Табл. 7-53: Операторы сравнения для 16-битной обработки

FNC	Команда	Истинно, если	Ложно, если
194	ДВКСМР =	((S1+)+1), (S1+) = ((S2+)+1), (S2+)	((S1+)+1), (S1+) <> ((S2+)+1), (S2+)
195	ДВКСМР >	((S1+)+1), (S1+) > ((S2+)+1), (S2+)	((S1+)+1), (S1+) ≤ ((S2+)+1), (S2+)
196	ДВКСМР <	((S1+)+1), (S1+) < ((S2+)+1), (S2+)	((S1+)+1), (S1+) ≥ ((S2+)+1), (S2+)
197	ДВКСМР <>	((S1+)+1), (S1+) <> ((S2+)+1), (S2+)	((S1+)+1), (S1+) = ((S2+)+1), (S2+)
198	ДВКСМР ≤	((S1+)+1), (S1+) ≤ ((S2+)+1), (S2+)	((S1+)+1), (S1+) > ((S2+)+1), (S2+)
199	ДВКСМР ≥	((S1+)+1), (S1+) ≥ ((S2+)+1), (S2+)	((S1+)+1), (S1+) < ((S2+)+1), (S2+)

Табл. 7-54: Операторы сравнения для 32-битной обработки

- Если все n сравнений дают результат “истинно”, дополнительно устанавливается специальный маркер M8090.
- При 16-битной обработке константа может находиться в диапазоне от -32768 до 32767, а при 32-битной обработке - в диапазоне от -2.147.483.648 до 2.147.483.647.

УКАЗАНИЕ

Для сравнения фактических значений 32-битных счетчиков (C200...C255) необходимо применять команды для 32-битной обработки (DBKCMР=, DBKCMР>, DBKCMР< и т. п.). Если для этого сравнения выбрана 16-битная обработка (BKCMР=, BKCMР> и т. п.), возникает ошибка с кодом 6705.

16-битная обработка

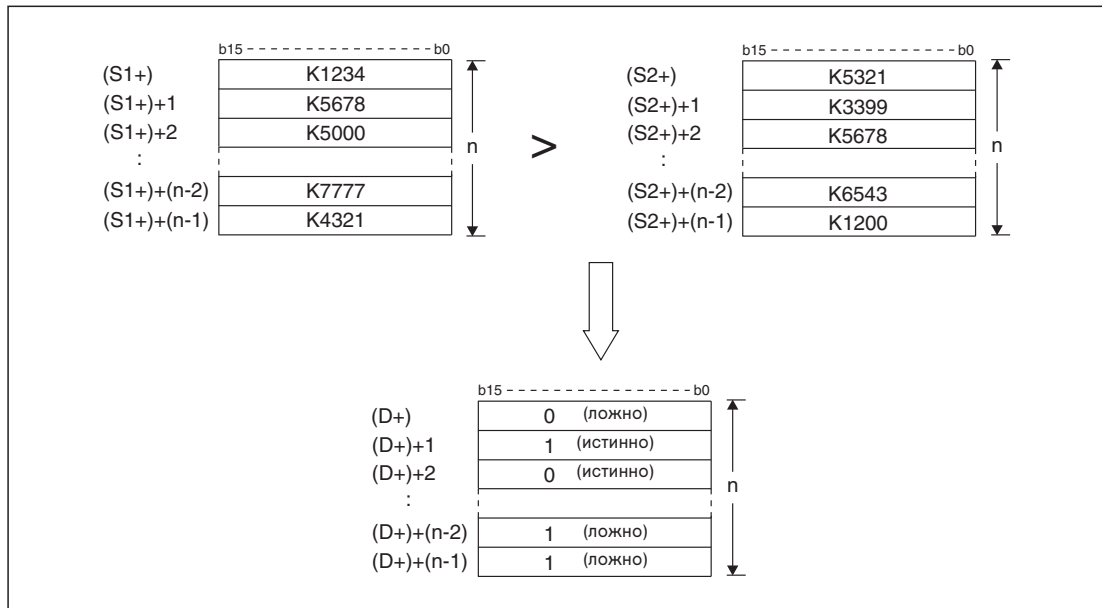


Рис. 7-200: В этом примере содержимые двух блоков данных с помощью команды BKCMР> сравниваются на “больше”.

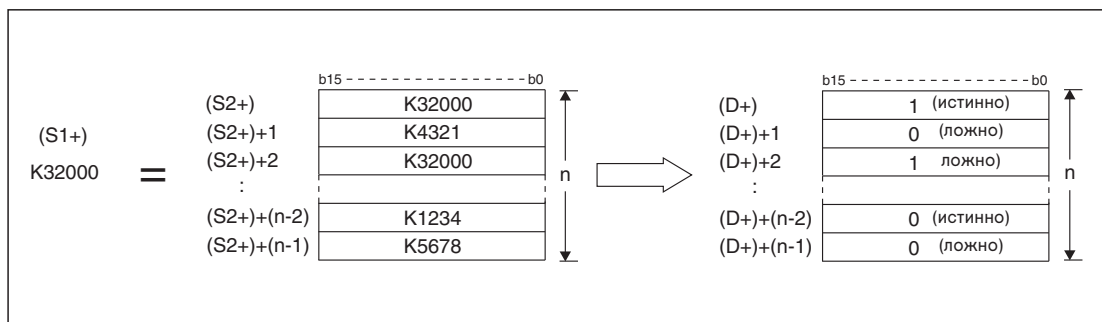


Рис. 7-201: Константа сравнивается с содержимым блока данных на “равно” (BKCMР=)

32-битная обработка

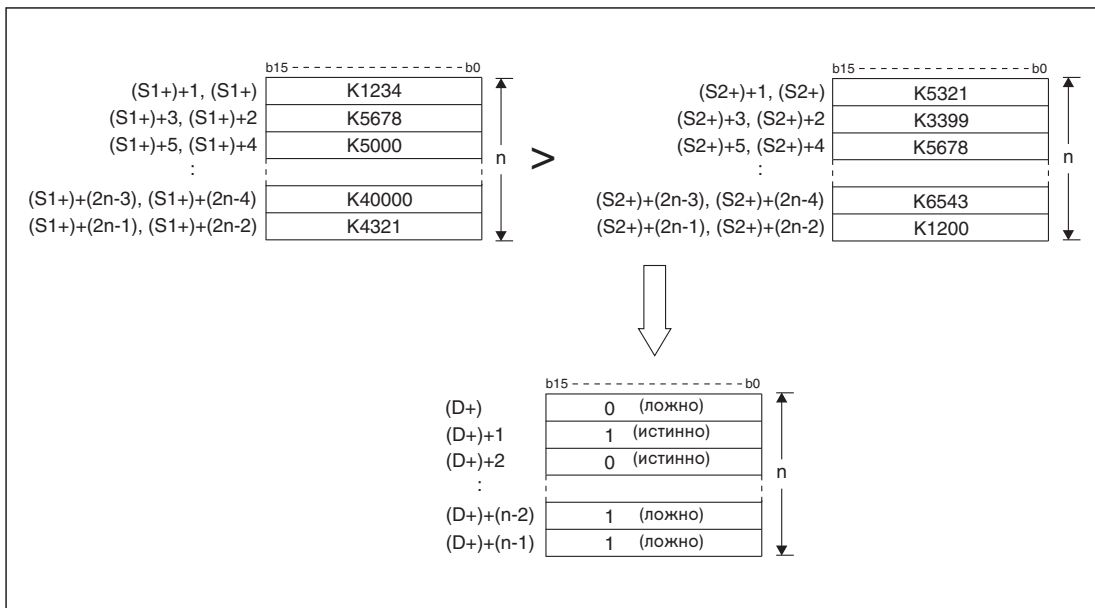


Рис. 7-202:Сравнение содержимых двух блоков данных на “больше” (DBKCMP>)

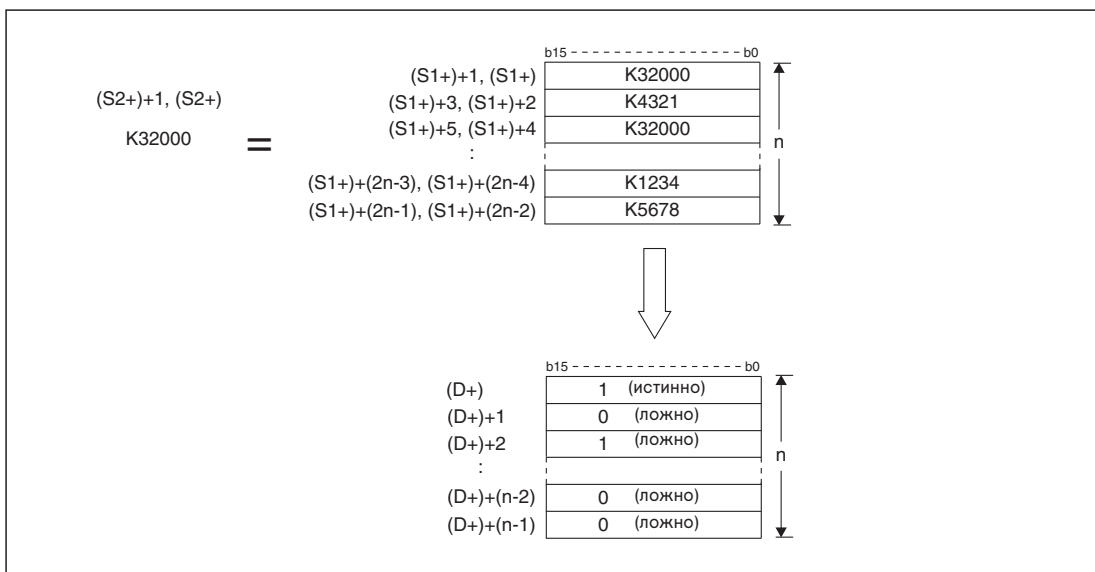


Рис. 7-203:Константа сравнивается с содержимым блока данных на “равно” (DBKCMP=)

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- В результате указания (n) (при 32-битной обработке: 2n) превышает допустимая область для типа операнда, указанного в (S1+) или (S2+). (Код ошибки "6706")
- В результате указания (n) превышает допустимая область для типа операнда, указанного в (D+). (Код ошибки "6706")
- В (D+) указаны биты регистра (D.n) и эти операнды перекрываются с областями, установленными в (S1+) или (S2+). (Код ошибки "6706")
- Для (S1+) и/или (S2+) команды 16-битной версии указан 32-битный счетчик (C200...C255). (Код ошибки "6705")

ПРИМЕР ▾

Следующая программа при включенном входе X20 сравнивает на равенство четыре двоичных числа (16-битных), хранящиеся начиная с D100, с четырьмя значениями, хранящимися начиная с D200. Результат сравнения записывается, начиная с M10. Если все значения начиная с D100 и D200 совпадают, специальный маркер M8090 включает выход Y000.

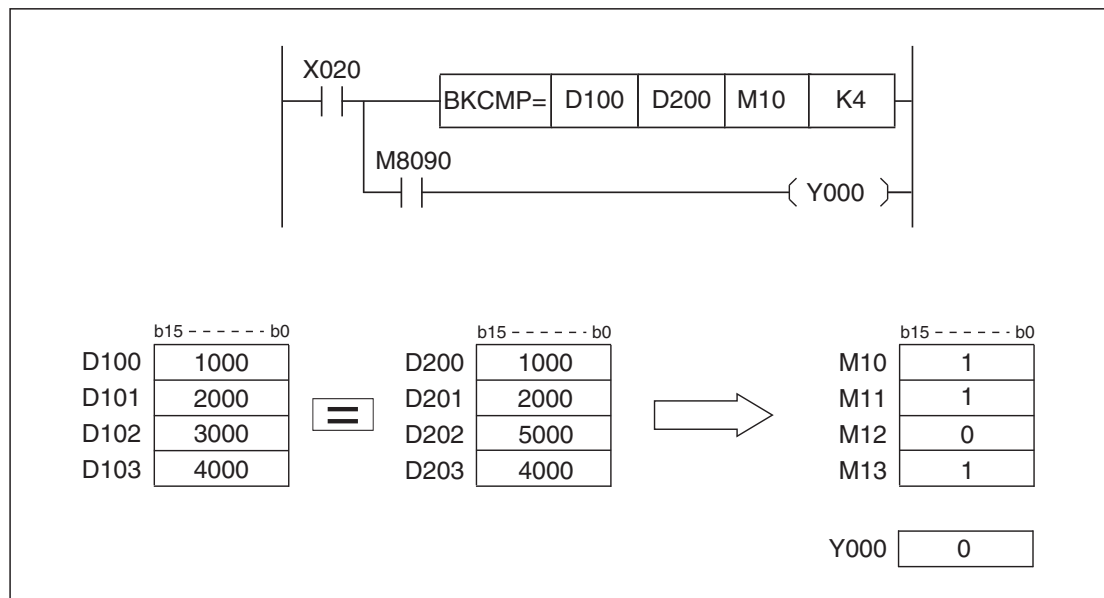


Рис. 7-204: В этом примере совпадают не все значения. Поэтому M8090 не установлен и выход остается выключенным.

△

ПРИМЕР ▾

При включенном входе X010 содержимые 4 регистров данных с D10 по D13 сравниваются со значением "1000" на "неравно". Результаты сравнения показываются битами 4, 5, 6 и 7 регистра D0.

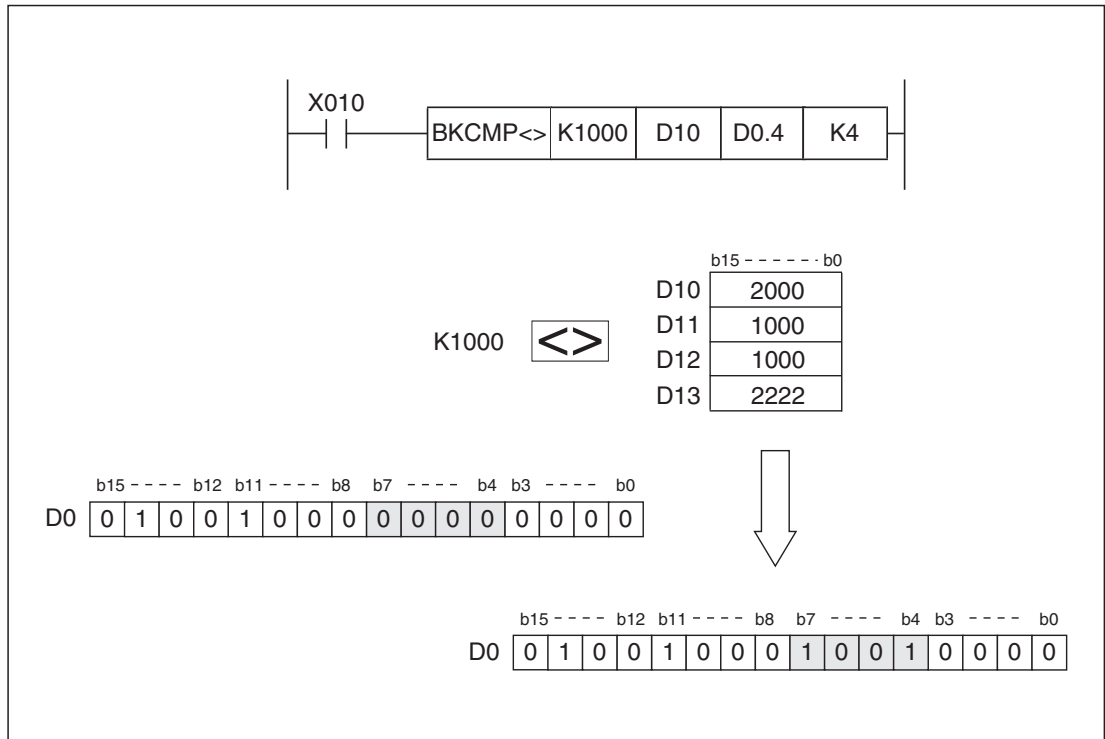


Рис. 7-205: В этом примере результаты сравнения сохраняются в регистре данных. На остальные биты D0 сравнение не влияет.



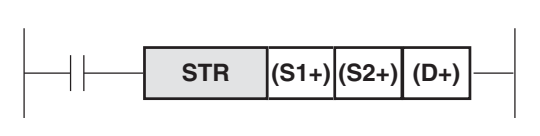
7.16 КОМАНДЫ ДЛЯ РАБОТЫ СО СТРОКОВЫМИ ВЕЛИЧИНАМИ

Строковые величины - это некоторое количество взаимосвязанных знаков, которыми контроллер оперирует в кодированном виде (часто в кодировке ASCII). Например, слово "MITSUBISHI" можно хранить в контроллере в кодировке ASCII и передавать его на подключенный прибор. Обзорная таблица кодов ASCII имеется в разделе В.4.

Символ	FNC	Назначение	Разд.
STR	200	Преобразование двоичных данных в строковые	7.16.1
VAL	201	Преобразование строковых данных в двоичные	7.16.2
\$+	202	Объединение строковых величин	7.16.3
LEN	203	Определение длины строковой величины	7.16.4
RIGHT	204	Фрагмент строковой величины справа	7.16.5
LEFT	205	Фрагмент строковой величины слева	7.16.6
MIDR	206	Выборка цепочки символов	7.16.7
MIDW	207	Замена цепочки символов	7.16.8
INSTR	208	Поиск цепочки символов	7.16.9
\$MOV	209	Передача цепочки символов	7.16.10

Табл. 7-55: Обзор команд для работы со строковыми величинами

7.16.1 ПРЕОБРАЗОВАНИЕ ДВОИЧНЫХ ДАННЫХ В СТРОКОВЫЕ (STR)

				STR		FNC 200			
				Преобразование двоичных данных в строковые					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R	T, C, D, R KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z, K, H	T, C, D, R		●	16бита	32бита	STR STRP	7
					●	●	DSTR DSTRP	13	

Функция

Преобразование 16/32-битных двоичных значений в число, отображаемое в виде строковой величины в кодировке ASCII с определенным количеством знаков.

Описание

- Хранящееся в (S2+) двоичное значение или константа преобразуется в соответствии с данными в (S1+)+0 и (S1+)+1 в строковую величину в кодировке ASCII и сохраняется начиная с операнда (D+).
- (S1+)+0 содержит число знаков, которое должен иметь результат преобразования (включая арифметический знак и десятичную запятую в виде точки). При 16-битной обработке можно указать от 2 до 8 знаков, а при 32-битной обработке - от 2 до 13 знаков.
- В (S1+)+1 вводится количество разрядов дробной части строкового числа. (16-битная обработка: от 0 до 5, 32-битная обработка: от 0 до 10 разрядов после запятой)

Правило: количество разрядов дробной части \leq (общее количество разрядов минус 3).

- Двоичное значение в (S2+) при 16-битной обработке может находиться в диапазоне от -32768 до 32767. При 32-битной обработке передается значение в (S2+)+1 и (S2+)+0, которое может находиться в диапазоне от -2.147.483.648 до 2.147.483.647.
- В области начиная с (D+) записывается результат преобразования в строковую величину:
 - В случае положительного числа в качестве арифметического знака записывается "20H" (знак пробела) в младшем байте ((D+)+0). Отрицательное число в этом месте обозначается знаком минус (2DH).
 - За концом преобразованной строковой величины автоматически вставляется знак "NUL" (00H). При четном количестве знак "00H" вставляется в операнд, следующий за операндом с последним преобразованным знаком. При нечетном количестве знаков код "00H" вставляется в старший байт операнда, в младшем байте которого содержится последний знак.
 - Если количество разрядов дробной части в (S1+)+1 установлено на иное значение кроме нуля, в указанном месте автоматически записывается ASCII-код "2EH" (точка), означающий десятичную запятую. Если количество разрядов дробной части установлено на ноль, десятичная точка не записывается.

Если количество всех отображаемых разрядов (включая десятичную точку и арифметический знак) меньше указанного общего количества разрядов, разряды между арифметическим знаком и первым отображаемым разрядом заполняются ASCII-кодами "20H" (знаками пробела).

- Если в $((S1+)+0)$ указано меньше разрядов, чем имеет двоичное значение в $(S2+)$ или $(S2+)+1$ и $(S2+)+0$, возникает ошибка.

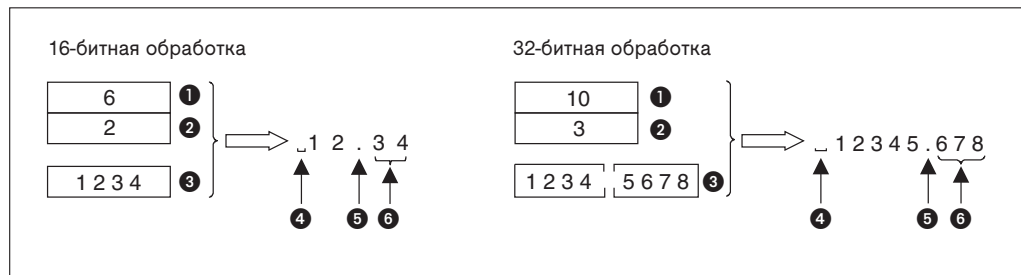


Рис. 7-206: Автоматическое вставление десятичной запятой (в виде точки) и знака пробела

- ❶: общее количество знаков в $((S1+)+0)$
- ❷: количество разрядов дробной части в $((S1+)+1)$
- ❸: двоичное значение в $(S2+)$ или $(S2+)+1$ и $(S2+)+0$
- ❹: автоматически вставленный знак пробела (20H).
- ❺: автоматически вставленная десятичная точка
- ❻: дробная часть

- Если в $((S1+)+1)$ указана более длинная дробная часть, чем дробная часть двоичного числа в $(S2+)$ или $(S2+)+1$ и $(S2+)+0$, значение смещается вправо и автоматически вставляются нули (30H).

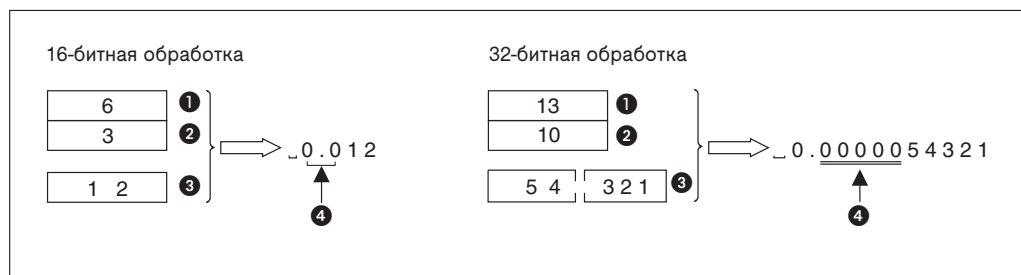


Рис. 7-207: Автоматическое вставление нулей

- ❶: общее количество знаков в $((S1+)+0)$
- ❷: количество разрядов дробной части в $((S1+)+1)$
- ❸: двоичное значение в $(S2+)$ или $(S2+)+1$ и $(S2+)+0$
- ❹: автоматически вставленные нули (31H).

16-битная обработка

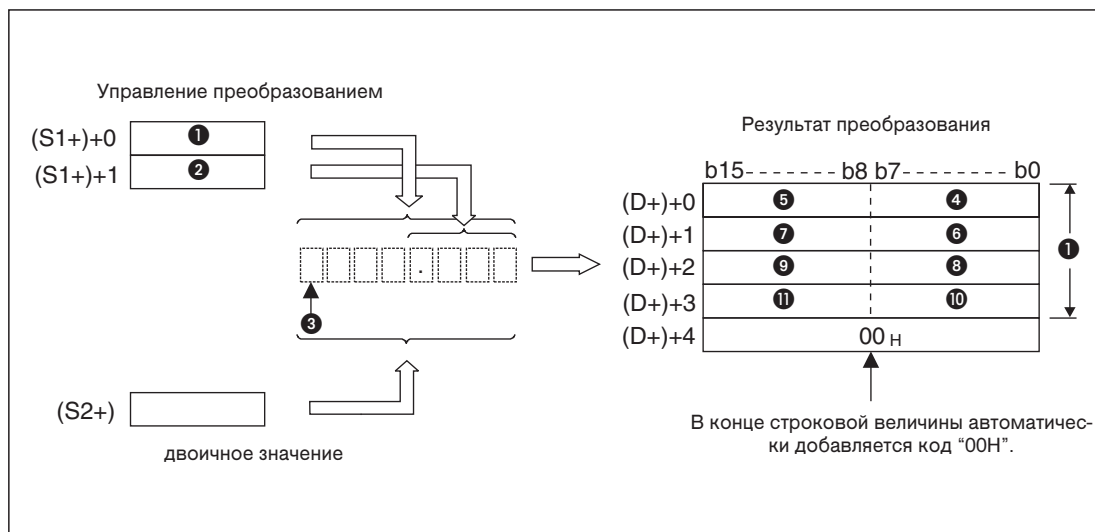


Рис. 7-208: Принцип преобразования 16-битного двоичного значения в строковую величину

- ① общее количество разрядов
- ② количество разрядов дробной части
- ③ арифметический знак
- ④ ASCII-код арифметического знака
- ⑤ ASCII-код знака "общее количество разрядов - 1"
- ⑥ ASCII-код знака "общее количество разрядов - 2"
- ⑦ ASCII-код знака "общее количество разрядов - 3"
- ⑧ ASCII-код точки (2EH)
- ⑨ ASCII-код знака "общее количество разрядов - 5"
- ⑩ ASCII-код знака "общее количество разрядов - 6"
- ⑪ ASCII-код знака "общее количество разрядов - 7"

ПРИМЕР ▾

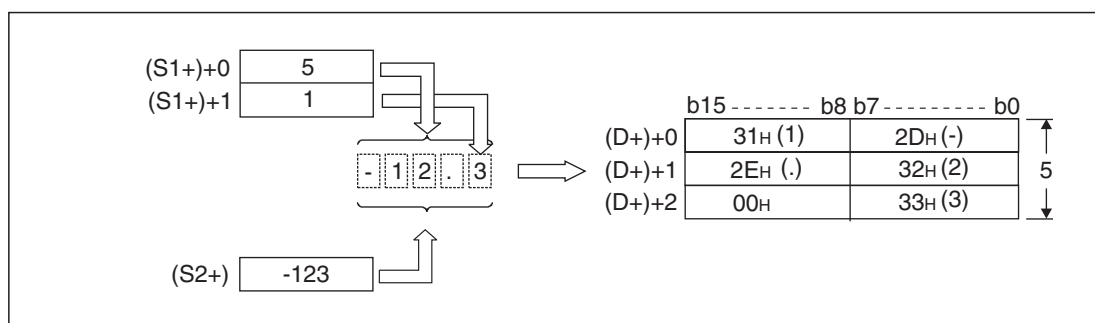


Рис. 7-209: Двоичное значение -123 преобразуется в строковую величину с общим числом разрядов 5 (из них один десятичный разряд после запятой). Результат сохраняется по адресу начиная с (D+).



32-битная обработка

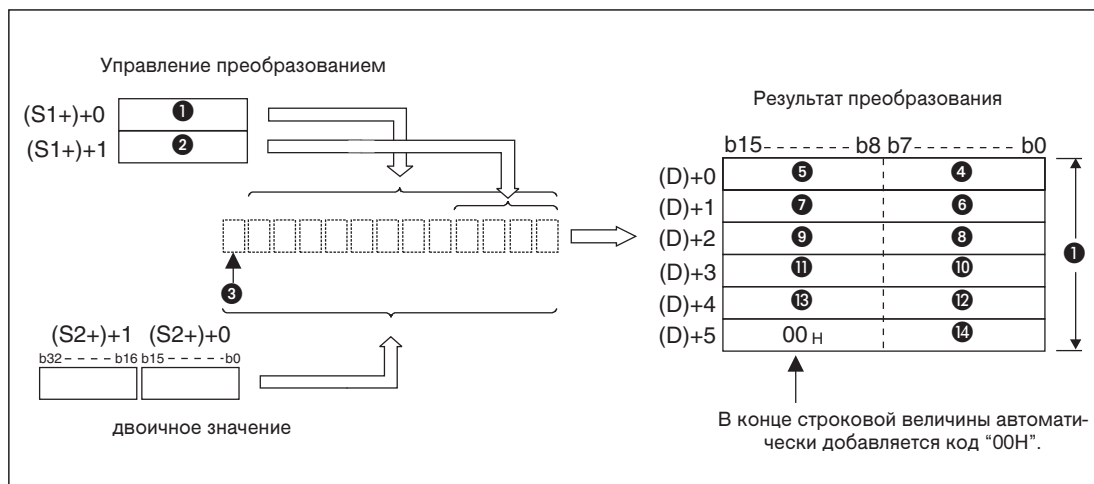


Рис. 7-210: Принцип преобразования 32-битного двоичного значения в строковую величину

- ① общее количество разрядов
- ② количество разрядов дробной части
- ③ арифметический знак
- ④ ASCII-код арифметического знака
- ⑤ ASCII-код знака "общее количество разрядов - 1"
- ⑥ ASCII-код знака "общее количество разрядов - 2"
- ⑦ ASCII-код знака "общее количество разрядов - 3"
- ⑧ ASCII-код точки (2EH)
- ⑨ ASCII-код знака "общее количество разрядов - 5"
- ⑩ ASCII-код знака "общее количество разрядов - 6"
- ⑪ ASCII-код знака "общее количество разрядов - 7"
- ⑫ ASCII-код знака "общее количество разрядов - 8"
- ⑬ ASCII-код знака "общее количество разрядов - 9"
- ⑭ ASCII-код знака "общее количество разрядов - 10"

ПРИМЕР ▾

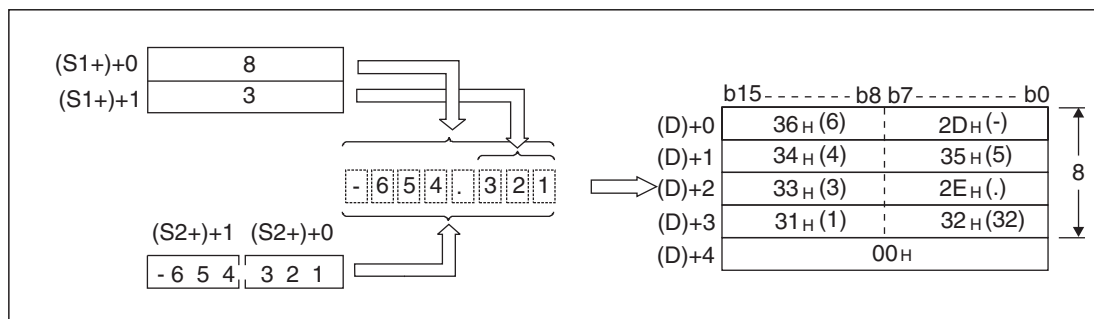


Рис. 7-211: Двоичное значение -654321 преобразуется в строковую величину с общим количеством разрядов 8 (из них 3 в дробной части). Результат сохраняется по адресу начиная с (D+).



Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Общее количество разрядов в ((S1)+0) при 16-битной обработке не находится в пределах от 2 до 8, а при 32-битной обработке - в пределах от 2 до 13.
- Количество разрядов дробной части в ((S1)+1) при 16-битной обработке не находится в пределах от 0 до 5, а при 32-битной обработке - в пределах от 0 до 10.
- Соотношение между общим количеством разрядов в ((S1)+0) и количеством разрядов дробной части в ((S1)+1) не удовлетворяет следующему условию:
(общее количество разрядов минус 3) \geq количество разрядов дробной части
- В ((S1)+0) указано меньшее количество разрядов (включая арифметический знак и десятичную точку), чем имеется разрядов в двоичном значении в (S2+) или (S2+)+1) и (S2+)+0).
- Область памяти начиная с (D+) превышает допустимую область для этих операндов.

ПРИМЕР ▾

При включении входа X0 записанное перед этим в D10 двоичное значение преобразуется в строковую величину из 6 знаков, которая затем записывается по адресу начиная с D20. Перед выполнением команды STRP общее количество разрядов (6) вводится в D0, а количество разрядов дробной части (0) - в D1.

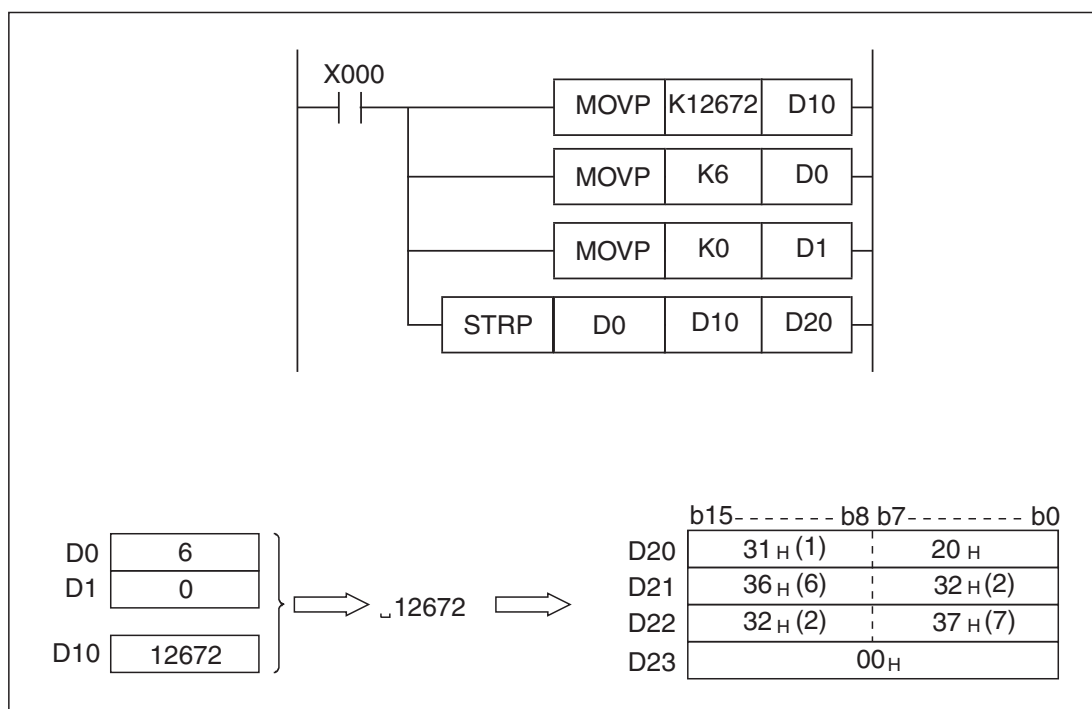



Рис. 7-212: Так как значение в D10 имеет только 5 разрядов, а строковая величина должна иметь 6 знаков, автоматически вставляется знак пробела.

△

7.16.2 ПРЕОБРАЗОВАНИЕ СТРОКОВЫХ ДАННЫХ В ДВОИЧНЫЕ (VAL)

				VAL		FNC 201			
				Преобразование строковых данных в двоичные					
CPU				FX1S	FX1N	FX2N	FX2NC	FX3U	
								●	
Операнды	S+	D1+	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R	T, C, D, R	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)		●	16бита	32бита	VAL VALP	7
					●	●	DVAL DVALP	13	

Функция

Преобразование строковой величины (ASCII-кода) в двоичное значение

Описание

- Строковое число, хранящееся по адресу начиная с (S+) в виде кодов ASCII, в соответствии с настройками в (D1+) и ((D1+)+1) преобразуется в двоичные данные и записывается по адресу начиная с (D2+). При 16-битной обработке двоичное число занимает только регистр (D2+), а при 32-битной обработке - регистры ((D2+)+1) и (D2+).
- ASCII-данные в (S+) конвертируются до тех пор, пока не распознается метка конца "00H". Если ((S+)+0) содержит знак пробела "20H", строковое число интерпретируется как положительное. Знак минус (2DH) в ((S+)+0) указывает на отрицательное число. Данные начиная с (S+) должны содержать только следующие знаки:
 - знак пробела (20H)
 - знак минус (2DH)
 - десятичная точка (2EH)
 - цифры от 0 до 9 (от 30H до 39H)
- (D1+)+0 содержит число знаков, которое требуется преобразовать (включая арифметический знак и десятичную точку). При 16-битной обработке можно указать от 2 до 8 знаков, а при 32-битной обработке - от 2 до 13 знаков.
- В (D1+)+1 вводится количество разрядов дробной части строкового числа, которое требуется преобразовать (при 16-битной обработке: от 0 до 5, при 32-битной обработке: от 0 до 10 разрядов после запятой).

Правило: количество разрядов дробной части \leq (общее количество разрядов минус 3).

В качестве десятичной запятой в строковом числе используется точка с кодом "2EH". Преобразованное двоичное значение всегда является целым числом (без десятичной запятой или точки).

- В области начиная с (D2+) записывается результат преобразования - двоичное значение:
 - Двоичное значение в (D2+) при 16-битной обработке может находиться в диапазоне от -32768 до 32767. При 32-битной обработке значение передается в (D2+)+1) и (D2+)+0) и может находиться в диапазоне от -2.147.483.648 до 2.147.483.647.
 - Если в строковой величине начиная с (S+) между арифметическим знаком и первой цифрой имеется ASCII-код "20H" (знак пробела) или "30H" (ноль), то эти знаки при конвертации игнорируются.

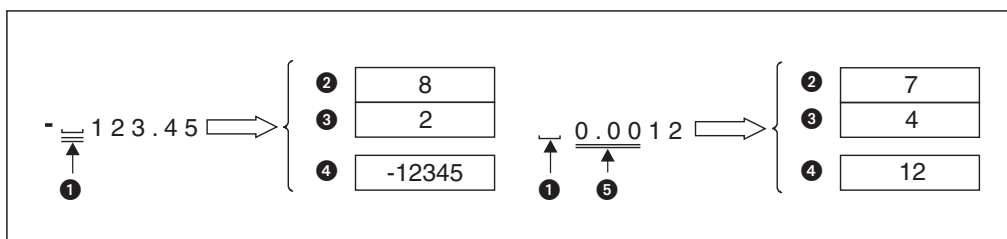


Рис. 7-213:Примеры игнорирования знака пробела и нулей при 16-битной обработке

- ①: знак пробела (не конвертируется)
- ②: общее количество знаков в $((D1+)+0)$
- ③: количество разрядов дробной части в $((D1+)+1)$
- ④: двоичное значение в $(D2+)$
- ⑤: нули (не конвертируются)

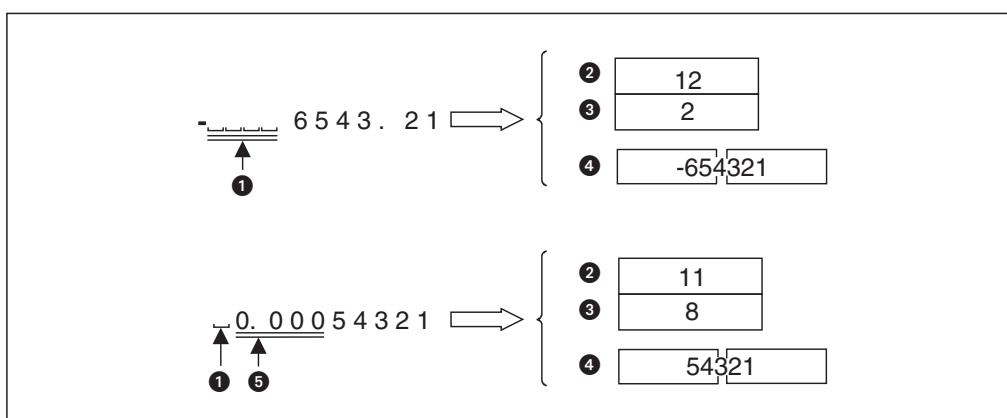


Рис. 7-214:Примеры игнорирования знака пробела и нулей при 32-битной обработке

- ①: знак пробела (не конвертируется)
- ②: общее количество знаков в $((D1+)+0)$
- ③: количество разрядов дробной части в $((D1+)+1)$
- ④: двоичное значение в $((D2+)+1)$ и $(D2+)+0)$
- ⑤: нули (не конвертируются)

16-битная обработка

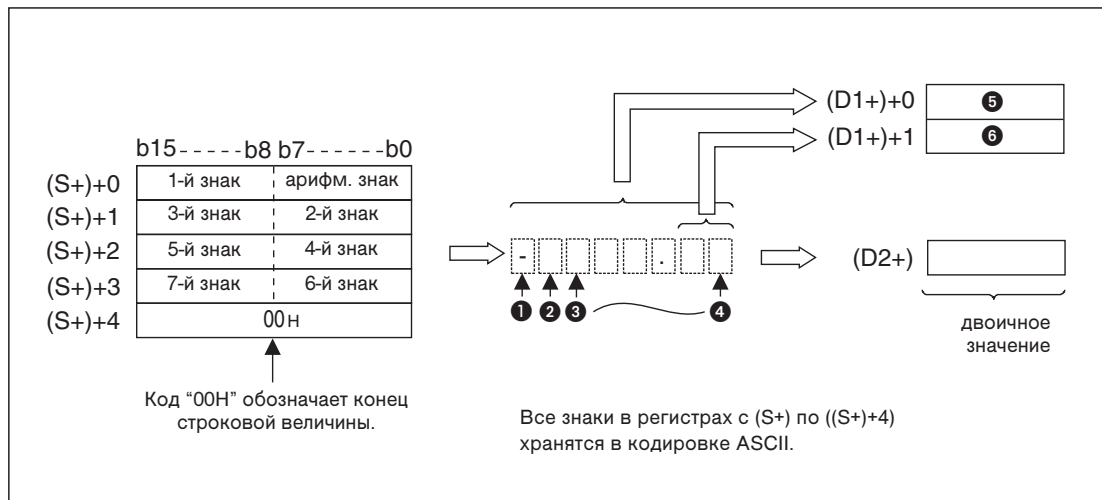


Рис. 7-215: Принцип преобразования строкового числа в 16-битовое двоичное значение с помощью команды VAL или VALP

- ① арифметический знак
- ② 1-й знак
- ③ 2-й знак
- ④ 7-й знак
- ⑤ общее количество знаков
- ⑥ количество разрядов дробной части

ПРИМЕР ▾

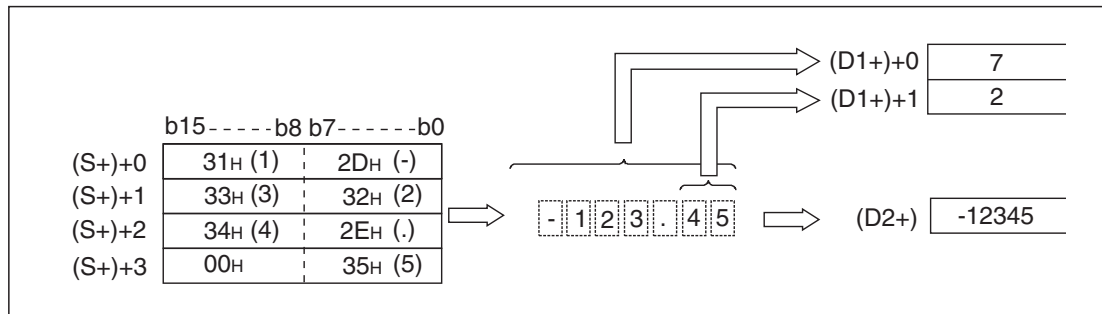


Рис. 7-216: Строковое число "-123.45" преобразуется в двоичный вид. Результат записывается в (D2+).



32-битная обработка

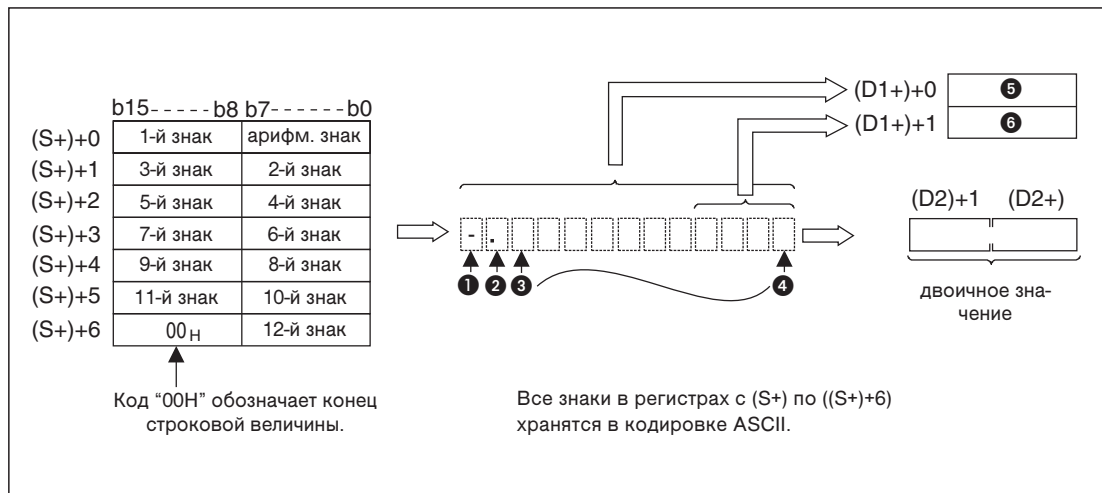


Рис. 7-217: Принцип преобразования строкового числа в 32-битовое двоичное значение с помощью команды DVAL или DVALP

- ❶ арифметический знак
- ❷ 1-й знак
- ❸ 2-й знак
- ❹ 12-й знак
- ❺ общее количество знаков
- ❻ количество разрядов дробной части

ПРИМЕР ▾

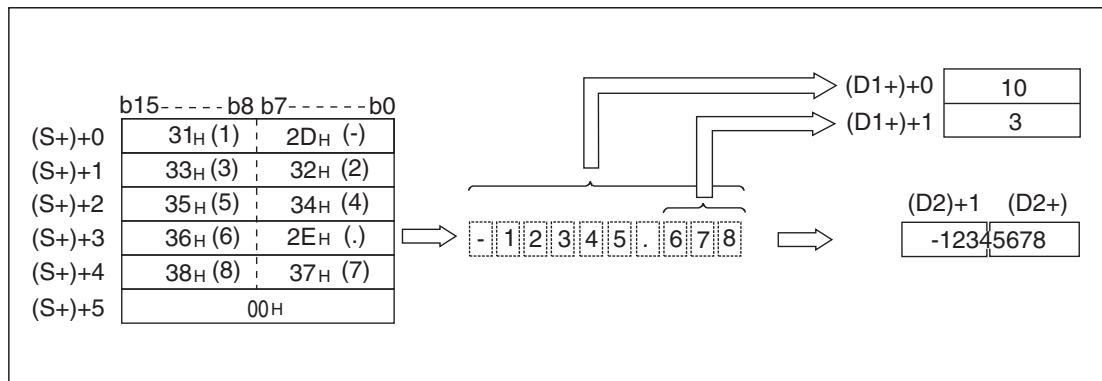


Рис. 7-218: Строковое число "-12345.78" преобразуется в двоичный вид. Результат записывается в ((D2+)+1) и (D2+).



Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Общее количество разрядов в ((D1)+0) при 16-битной обработке не находится в пределах от 2 до 8, или при 32-битной обработке - в пределах от 2 до 13.
- Количество разрядов дробной части в ((D1)+1) при 16-битной обработке не находится в пределах от 0 до 5, или при 32-битной обработке - в пределах от 0 до 10.
- Соотношение между общим количеством разрядов в ((D1)+0) и количеством разрядов дробной части в ((D1)+1) не удовлетворяет следующему условию:
(общее количество разрядов минус 3) \geq количество разрядов дробной части
- Арифметический знак строкового числа не состоит из знака ASCII 20H (пробела) или 2DH (знака минус).
- Строковая величина содержит иные знаки кроме цифр от 0 до 9 (от 30H до 39H), знаков пробела (20H), десятичной точки (2EH) или знака минус (2DH).
- Конвертируемая строковая величина содержит более одной десятичной точки (2EH).
- Преобразованное из строкового числа двоичное значение при 16-битной обработке находится вне диапазона от -32768 до 32767, а при 32-битной обработке - вне диапазона от -2.147.483.648 до 2.147.483.647.
- Строковая величина не заканчивается кодом "00H".

ПРИМЕР ▽

При включении X20 строковая величина, хранящаяся начиная с регистра D20, преобразуется в 16-битовое двоичное значение и записывается в D0. Регистр данных D19 содержит общее количество разрядов (6), а регистр D11 - количество разрядов дробной части строкового числа (2).

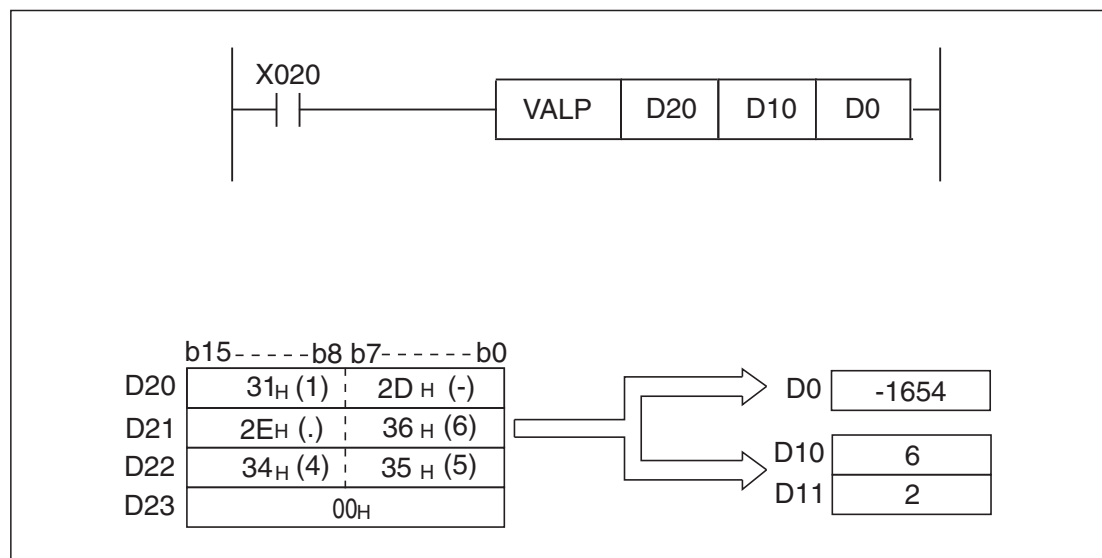
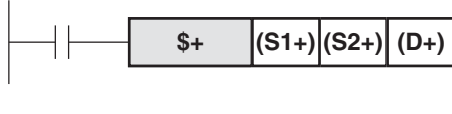


Рис. 7-219:Пример программы для преобразование строкового числа в 16-битовое двоичное число



7.16.3 ОБЪЕДИНЕНИЕ СТРОКОВЫХ ВЕЛИЧИН (\$+)

				\$+		FNC 202				
						Объединение строковых величин				
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
									●	
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы			
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U/G), черед знаков		KnY, KnM, KnS, T, C, D, R, специальные модули (U/G)	●	●	16бита	32бита	\$+ \$+P	7	

Функция

Объединяются две строковые величины

Описание

- Строковая величина, хранящаяся начиная с (S2+), присоединяется к строковой величине, указанной в (S1+).
- Результат сохраняется по адресу начиная с (D+).
- Конец строковых величин в (S1+) и (S2+) помечается знаком ASCII NUL (00H).
- Объединенная строковая величина начинается со знака в младшем байте строковой величины, указанной в (S1+), и заканчивается кодом "00H" строковой величины, указанной в (S2+). Метка конца "00H" строковой величины, указанной в (S1+), в регистр (D+) не записывается.
- За концом объединенной строковой величины автоматически вставляется знак "NUL" (00H). При четном количестве знаков "00H" вставляется в операнд, следующий за операндом с последним преобразованным знаком. При нечетном количестве знаков код "00H" вставляется в старший байт операнда, в младшем байте которого содержится последний знак.

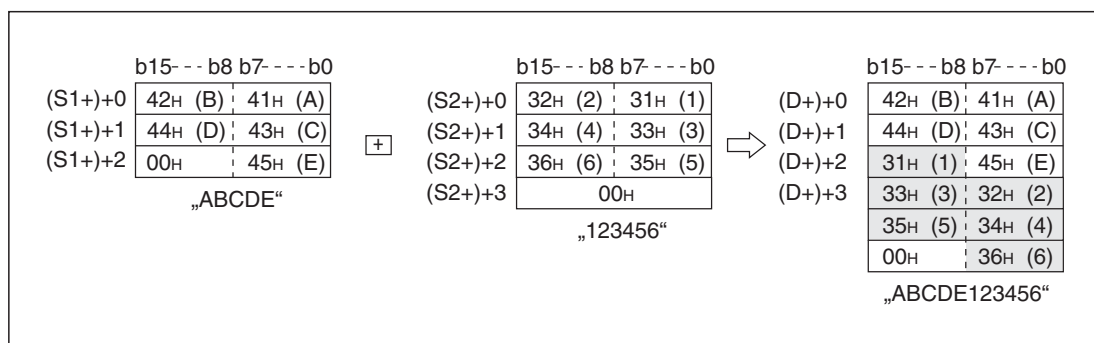


Рис. 7-220: Строковые величины в (S1+) и (S2+) объединяются и записываются в (D+).
(Строковая величина из (S2+) здесь изображена на сером фоне.)

- В (S1+) и (S2+) можно также непосредственно указывать строковые величины длиной до 32 знаков. Если в (S1+) и (S2+) указаны словные операнды, ограничений по числу знаков не имеется.
- Если обе строковые величины в (S1+) и (S2+) начинаются с кода "00H" (в этом случае число знаков равно 0), то по адресу начиная с (D+) записывается "0000H".

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Полная, объединенная строковая величина не может быть записана по адресу начиная с (D+).
- Области данных (S1+), (S2+) или (D+) перекрываются.
- В областях операндов, указанных в (S1+) или (S2+), не встречается код "00H".

ПРИМЕР ▾

В следующем примере при включении входа X000 объединяются строковая величина "abcde", хранящаяся по адресу начиная с D10, и строковая величина "ABCD". Объединенная строковая величина сохраняется по адресу, начиная с D100.

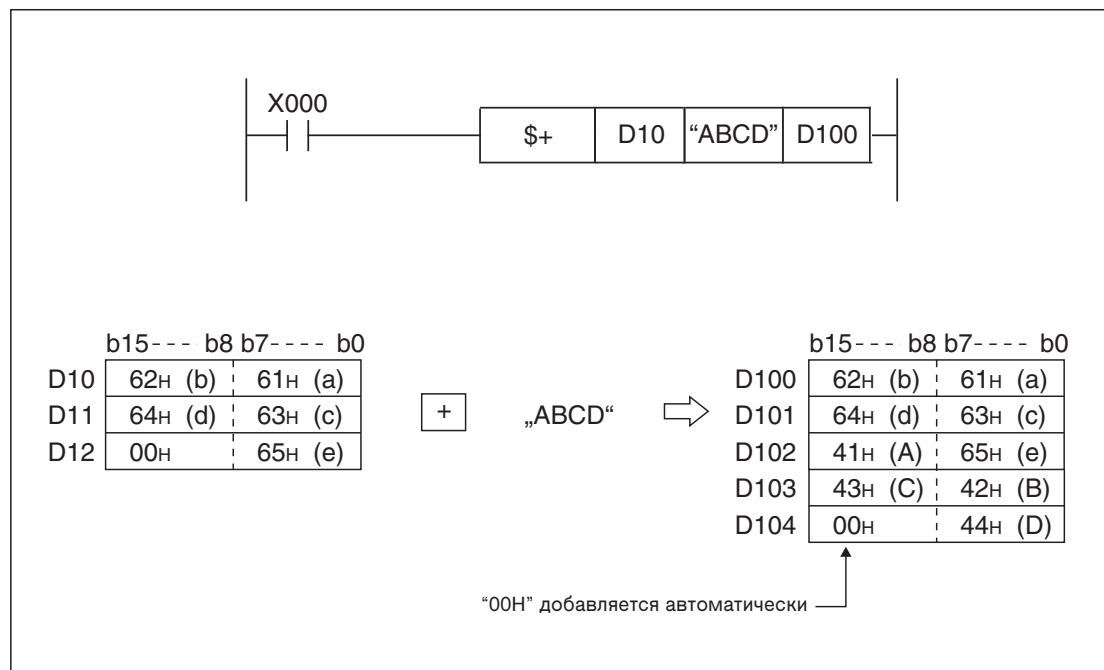


Рис. 7-221:Пример объединения двух строковых величин

△

7.16.4 ОПРЕДЕЛЕНИЕ ДЛИНЫ СТРОКОВОЙ ВЕЛИЧИНЫ (LEN)

			LEN		FNC 203			
			Определение длины строковой величины					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	S+	D+	Имп. инструкция(P)		Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□),	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	●	16бита	32бита	LEN LENP	5	

Функция

Определение длины строковой величины

Описание

- Команда LEN определяет число знаков указанной в (S+) строковой величины и сохраняет результат по адресу (D+).
- Счет начинается со знака в младшем байте строковой величины, указанной в (S1+), и заканчивается при распознании кода "00H".
- Так как один знак ASCII занимает один байт, единицей измерения длины строковой величины являются байты.
- Команда LEN распознает также знаки, записанные не в кодировке ASCII. Однако единицей знака всегда является "байт". Например, если подсчитывается количество знаков в кодировке JIS (1 знак = 16 битов), для одного знака выводится длина "2".

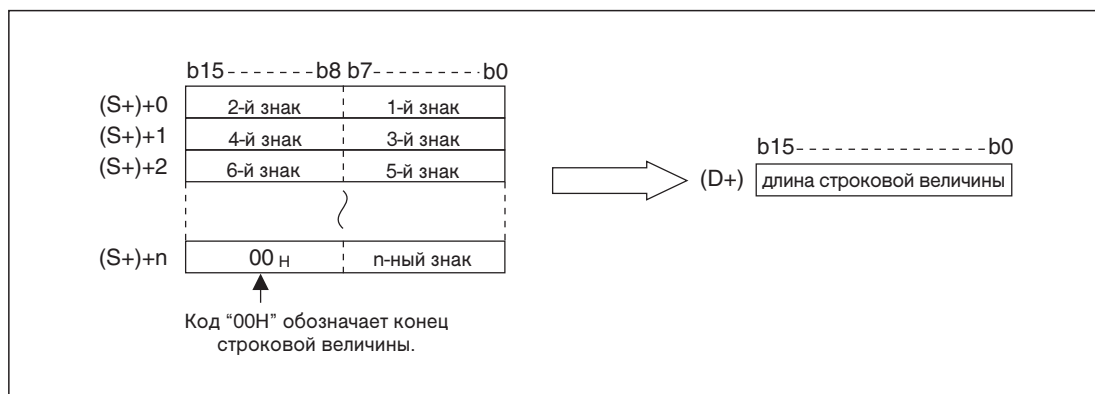


Рис. 7-222: Команда LEN подсчитывает число знаков строковой величины и записывает значение в (D+).

ПРИМЕР ▾

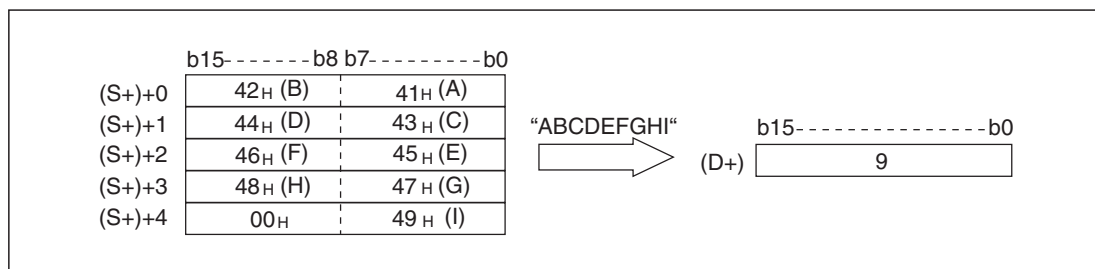


Рис. 7-223: В этом примере подсчитываются знаки в строке "ABCDEFGHI" и в качестве результата в (D+) вводится "9".



Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Длина строковой величины превышает 32768 байтов.
- В области операндов, указанной в (S+), не встречается код "00H".

ПРИМЕР ▾

В следующем примере при включенном входе X10 подсчитывается длина строковой величины, хранящейся начиная с регистра D0. Результат записывается в D10. Затем результат преобразуется в двоично-десятичный формат и выводится через выходы с Y057 по Y40.

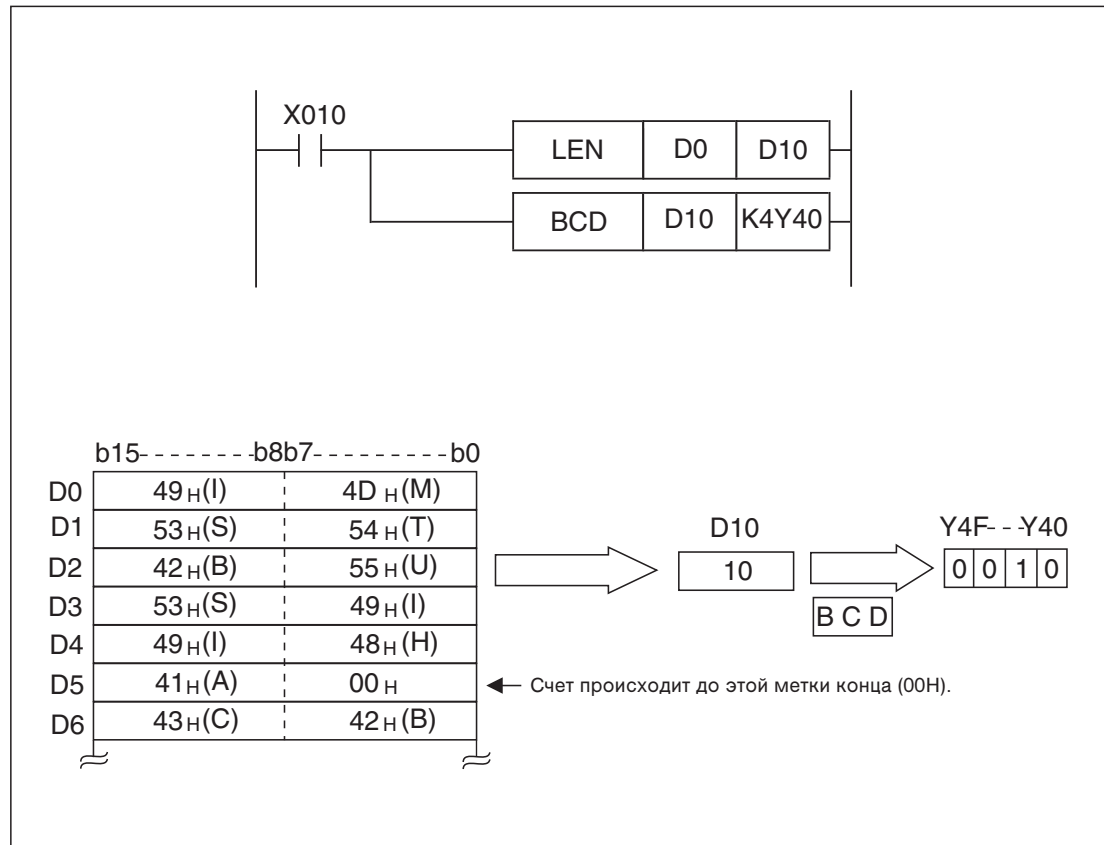



Рис. 7-224: В этом примере подсчитывается число знаков в слове "MITSUBISHI".

△

7.16.5 ФРАГМЕНТ СТРОКОВОЙ ВЕЛИЧИНЫ СПРАВА (RIGHT)

				RIGHT		FNC 204		
				Фрагмент строковой величины справа				
CPU				FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G)	D, R, K, H		●	16бита	32бита	RIGHT RIGHTP

Функция

Считывание знаков из строковой величины в кодировке ASCII. Считывание начинается с конца строковой величины (справа).

Описание

- Команда RIGHT считывает n знаков, начиная с конца строковой величины (с правой стороны строковой величины).
- В (S+) указывается, где хранится строковая величина. Строковая величина начинается со знака в младшем байте цепочки символов, указанной в (S+), и заканчивается кодом "00H".
- Извлеченные знаки сохраняются по адресу, начиная с (D+).
- После считанных знаков автоматически вставляется знак "NUL" (00H). При четном количестве знак "00H" вставляется в операнд, следующий за операндом с последним преобразованным знаком. При нечетном количестве знаков код "00H" вставляется в старший байт операнда, в младшем байте которого содержится последний знак.
- Количество считываемых знаков указывается с помощью (n) в байтах. При (n) = 0 в ((D+)+0) вводится код NUL (00H).

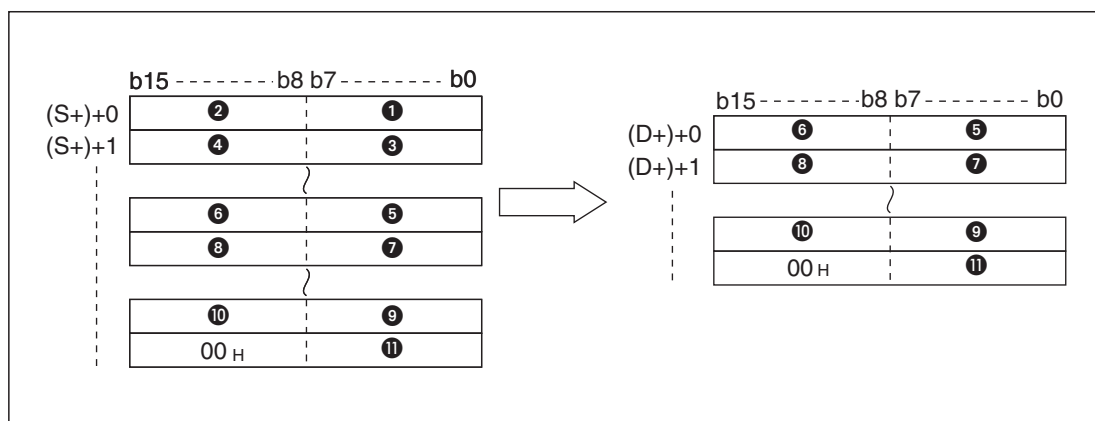


Рис. 7-225: С помощью команды RIGHT считываются знаки, начиная с конца строковой величины.

- 1 ASCII-код 1-го знака
- 2 ASCII-код 2-го знака
- 3 ASCII-код 3-го знака
- 4 ASCII-код 4-го знака

- ⑤ ASCII-код знака “последний знак - (n+1)”
- ⑥ ASCII-код знака “последний знак - (n+2)”
- ⑦ ASCII-код знака “последний знак - (n+3)”
- ⑧ ASCII-код знака “последний знак - (n+4)”
- ⑨ ASCII-код знака “последний знак - 2”
- ⑩ ASCII-код знака “последний знак - 1”
- ⑪ ASCII-код последнего знака

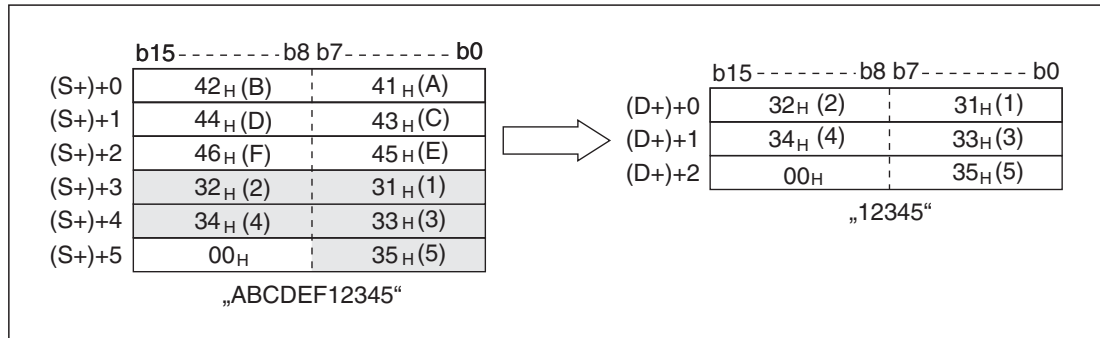
ПРИМЕР ▾

Рис. 7-226:Пример применения команды RIGHT: последние 5 знаков строковой величины “ABCDEF12345” считываются и записываются по адресу начиная с

**УКАЗАНИЕ**

Число знаков в (n) указывается в байтах. Это необходимо учитывать, если требуется считать знаки, сохраненные не в кодировке ASCII. Например, если требуется считать знаки в кодировке JIS (Японский промышленный стандарт) (1 знак = 16 битов), количество требуемых знаков следует умножить на 2 и ввести в (n) результат умножения.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки “6706”:

- Указанное в (n) количество считываемых знаков больше количества знаков, хранящихся начиная с адреса (S+).
- В (n) указано отрицательное число.
- В области операндов, указанной в (S+), не встречается код “00H”.
- В результате указания (n) превышает необходимая для сохранения знаков область операндов, начиная с (D+).

ПРИМЕР ▾

При включении входа X000 из строковой величины, хранящейся в регистрах начиная с R0, считываются последние 4 знака и записываются по адресу начиная с D0.

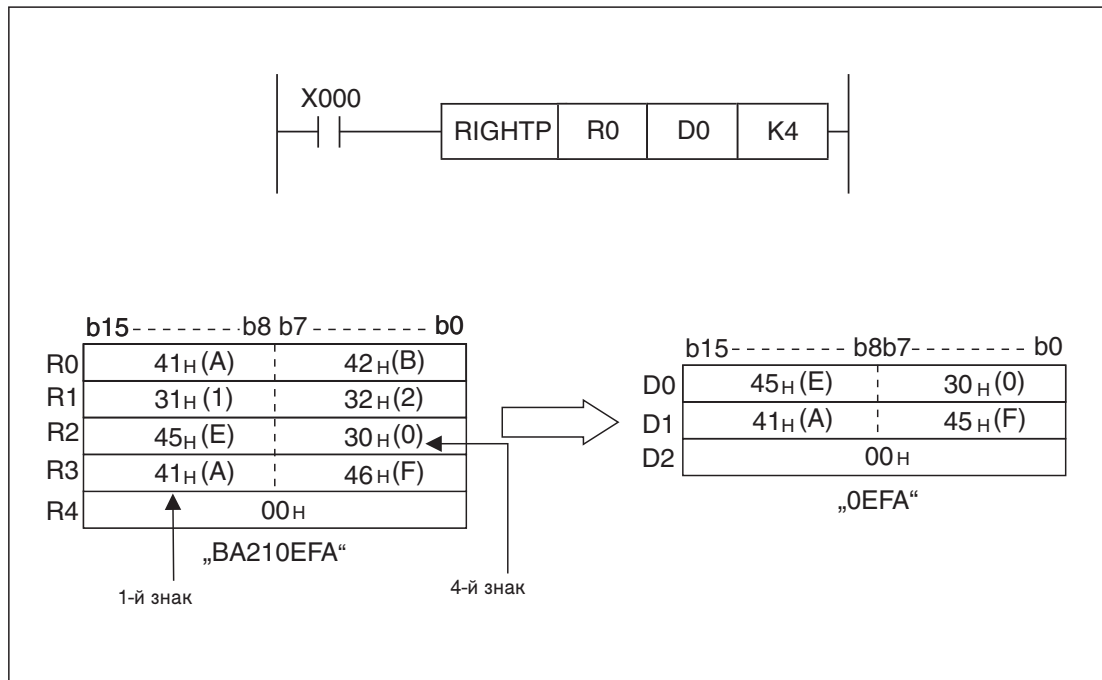


Рис. 7-227:Пример программы применения команды RIGHT(P)



7.16.6 ФРАГМЕНТ СТРОКОВОЙ ВЕЛИЧИНЫ СЛЕВА (LEFT)

				LEFT	FNC 205				
				Фрагмент строковой величины слева					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	D, R, K, H		●	16бита	32бита	LEFT LEFTP	7

Функция

Считывание знаков из строковой величины в кодировке ASCII. Считывание начинается с начала строковой величины (слева).

Описание

- Команда LEFT считывает n знаков, начиная с начала (левой стороны) строковой величины.
- В (S+) указывается, где хранится строковая величина. Строковая величина начинается со знака в младшем байте цепочки символов, указанной в (S1+), и заканчивается кодом "00H".
- Извлеченные знаки сохраняются по адресу, начиная с (D+).
- После считанных знаков автоматически вставляется знак "NUL" (00H). При четном количестве знак "00H" вставляется в операнд, следующий за операндом с последним преобразованным знаком. При нечетном количестве знаков код "00H" вставляется в старший байт операнда, в младшем байте которого содержится последний знак.
- Количество считываемых знаков указывается с помощью (n) в байтах. При (n) = 0 в ((D+)+0) записывается код NUL (00H).

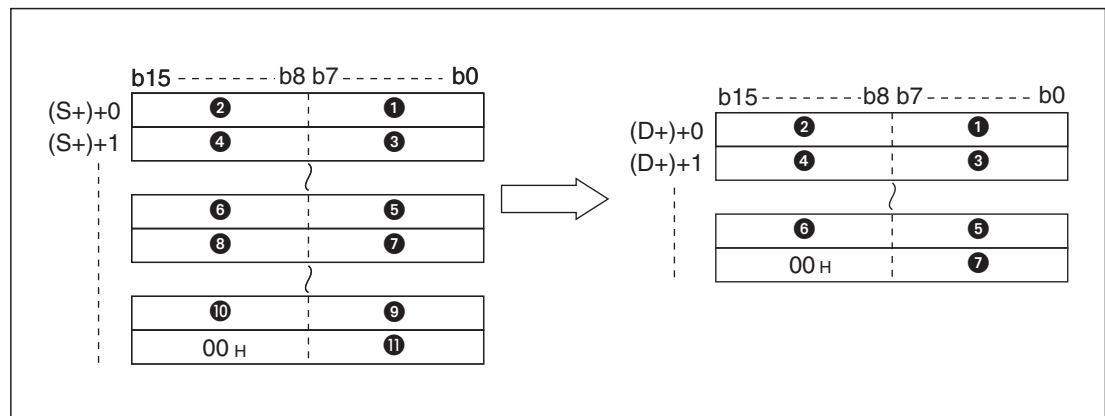


Рис. 7-228: С помощью команды LEFT определяются знаки с начала строковой величины.

- 1 ASCII-код 1-го знака
- 2 ASCII-код 2-го знака
- 3 ASCII-код 3-го знака
- 4 ASCII-код 4-го знака

- ⑤ ASCII-код знака “последний знак - (n+1)”
- ⑥ ASCII-код знака “последний знак - (n+2)”
- ⑦ ASCII-код знака “последний знак - (n+3)”
- ⑧ ASCII-код знака “последний знак - (n+4)”
- ⑨ ASCII-код знака “последний знак - 2”
- ⑩ ASCII-код знака “последний знак - 1”
- ⑪ ASCII-код последнего знака

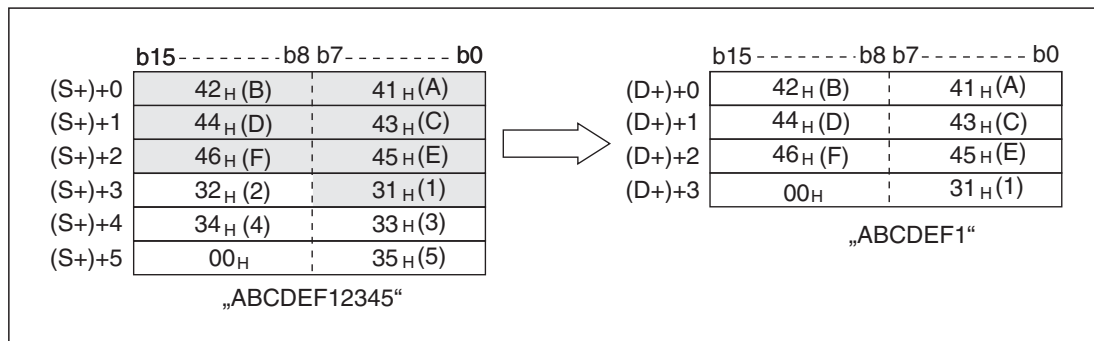
ПРИМЕР ▾

Рис. 7-229: Пример применения команды LEFT: первые 7 знаков строковой величины “ABCDEF12345” считываются и сохраняются по адресу начиная с (D+).

**УКАЗАНИЕ**

Число знаков в (n) указывается в байтах. Это необходимо учитывать, если требуется считать знаки, сохраненные не в кодировке ASCII. Например, если требуется считать знаки в кодировке JIS (Японский промышленный стандарт) (1 знак = 16 битов), количество требуемых знаков следует умножить на 2 и ввести в (n) результат умножения.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки “6706”:

- Указанное в (n) количество считываемых знаков больше количества знаков, хранящихся начиная с адреса (S+).
- В (n) указано отрицательное число.
- В области операндов, указанной в (S+), не встречается код “00H”.
- В результате указания (n) превышает необходимая для сохранения знаков область операндов, начиная с (D+).

ПРИМЕР ▾

В следующем примере программы при включении входа X010 считываются первые 6 знаков строковой величины, хранящейся по адресу, начиная с D100. Эти знаки записываются по адресу, начиная с R10. Количество знаков указано в регистре данных D0.

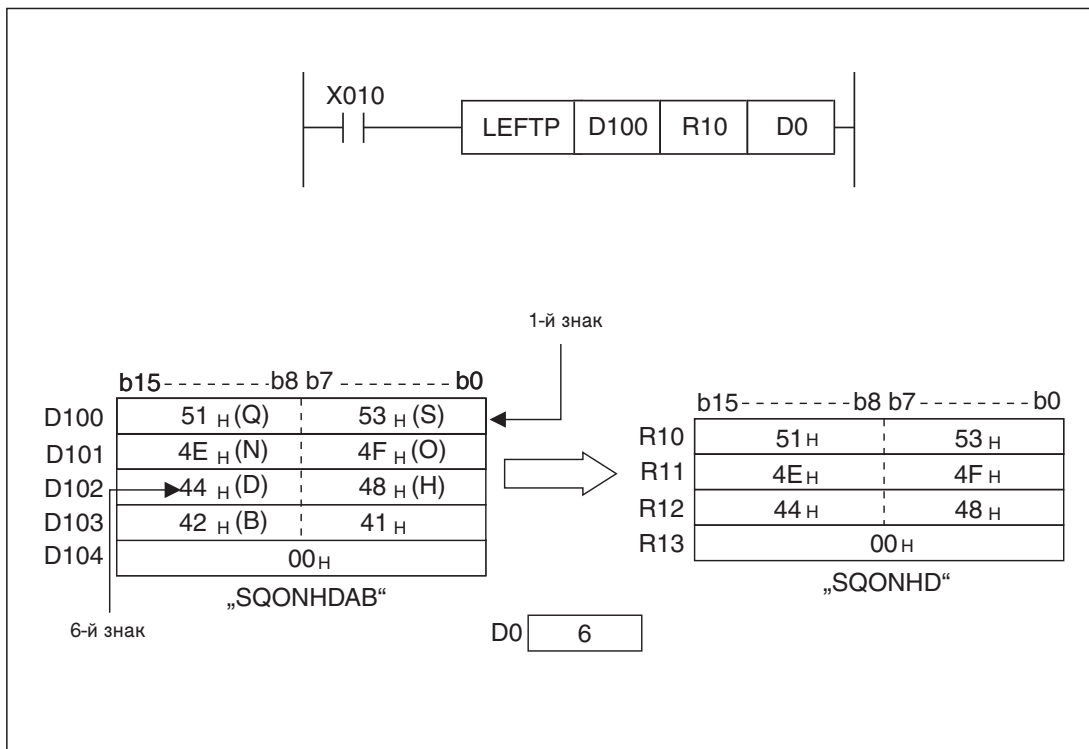



Рис. 7-230:Пример программы применения команды `LEFT(P)`



7.16.7 КОПИРОВАНИЕ ЗНАКОВ ИЗ СТРОКОВОЙ ВЕЛИЧИНЫ (MIDR)

				MIDR		FNC 206		
				Копирование знаков из строковой величины				
				CPU	FX1S	FX1N	FX2N	FX2NC
								●
Операнды	S1+	D+	S2+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)		●	16бита	32бита	MIDR MIDRP

Функция

Копирование знаков из строковой величины с любой позиции

Описание

- Команда MIDR считывает определенную часть строковой величины, для чего должно быть указано положение первого знака и считываемое количество знаков.
- В (S1+) указывается, где хранится строковая величина. Строковая величина начинается со знака в младшем байте цепочки символов, указанной в (S1+), и заканчивается кодом "00H".
- В ((S2+)+0) указывается, начиная с какого знака строковой величины требуется считывать знаки.
- ((S2+)+1) содержит количество байтов (знаков), которое требуется скопировать. Если здесь введено значение "0", команда MIDR не выполняется. При значении "-1" начиная с первого указанного знака копируются все остальные знаки строковой величины, хранящейся по адресу начиная с (S1+) (см. рис. 7-233).
- Извлеченные знаки сохраняются по адресу, начиная с (D+).
- После скопированных знаков автоматически вставляется знак "NUL" (00H). При четном количестве знаков "00H" вставляется в операнд, следующий за операндом с последним преобразованным знаком. При нечетном количестве знаков код "00H" вставляется в старший байт операнда, в младшем байте которого содержится последний знак.

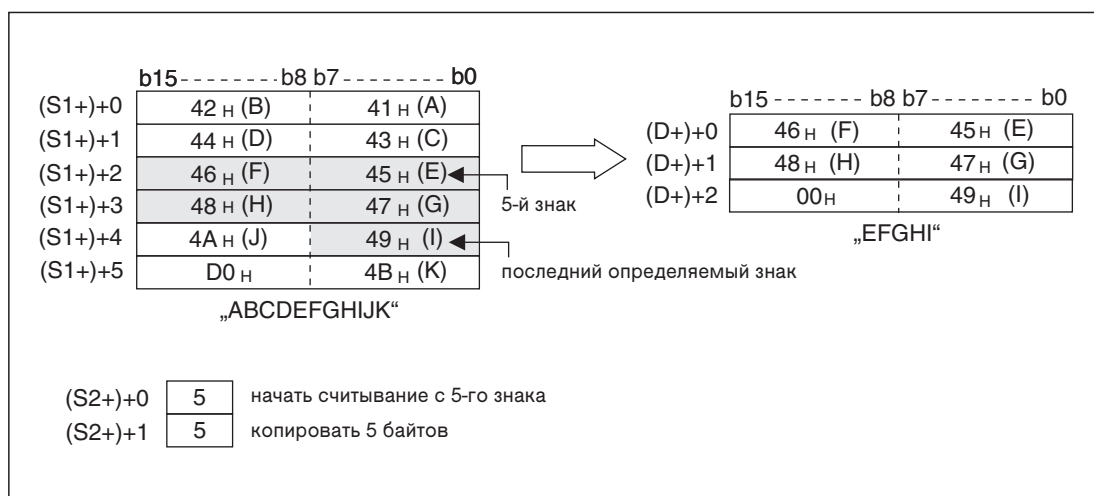


Рис. 7-231: В ((S2+)+0) и ((S2+)+1) для команды MIDR указывается, начиная с какого знака требуется копировать и сколько знаков должны быть скопированы

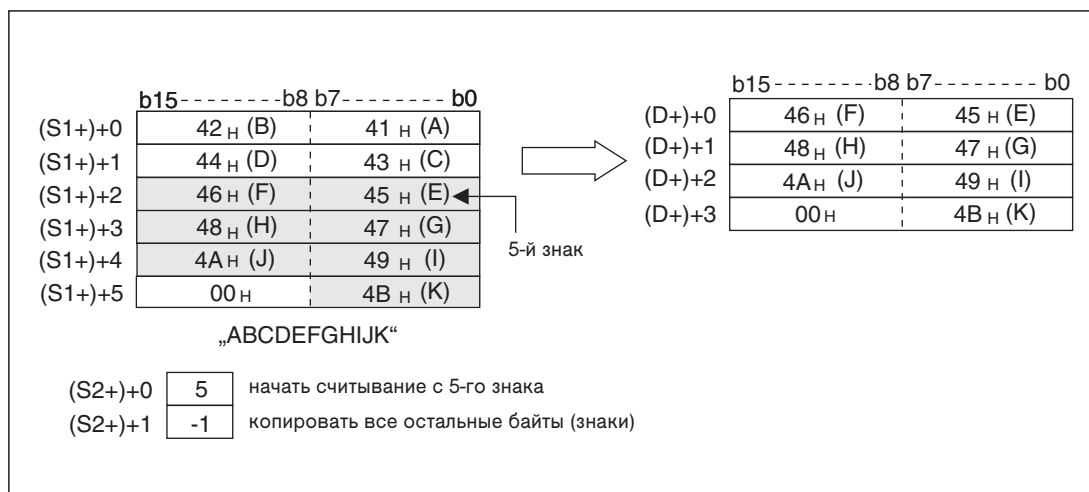


Рис. 7-232: Если в ((S2+)+1) введено значение “-1”, копируются все остальные знаки строковой величины.

УКАЗАНИЕ

Число знаков в ((S2+)+1) указывается в байтах. Это необходимо учитывать, если требуется считать знаки, сохраненные не в кодировке ASCII. Например, если требуется считать знаки в кодировке JIS (Японский промышленный стандарт) (1 знак = 16 битов), необходимо умножить количество требуемых знаков на 2 и ввести в ((S2+)+1) результат умножения.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки “6706”:

- Указанное в ((S2+)+1) количество копируемых знаков больше количества знаков, хранящихся начиная с (S1+).
- В области операндов, указанной в (S1+), не встречается код “00н”.
- В результате указания ((S2+)+1) превышает область операндов начиная с (D+), необходимая для сохранения знаков.
- В ((S2+)+0) указано отрицательное число.
- В ((S2+)+1) указано значение меньше -1.

ПРИМЕР ▾

В следующем примере программы при включении входа X000 из строковой величины, хранящейся начиная с адреса D10, считываются шесть знаков начиная с третьего знака. Считанные знаки записываются по адресу начиная с R0. Положение первого копируемого знака указано в R0, а число знаков - в R1.

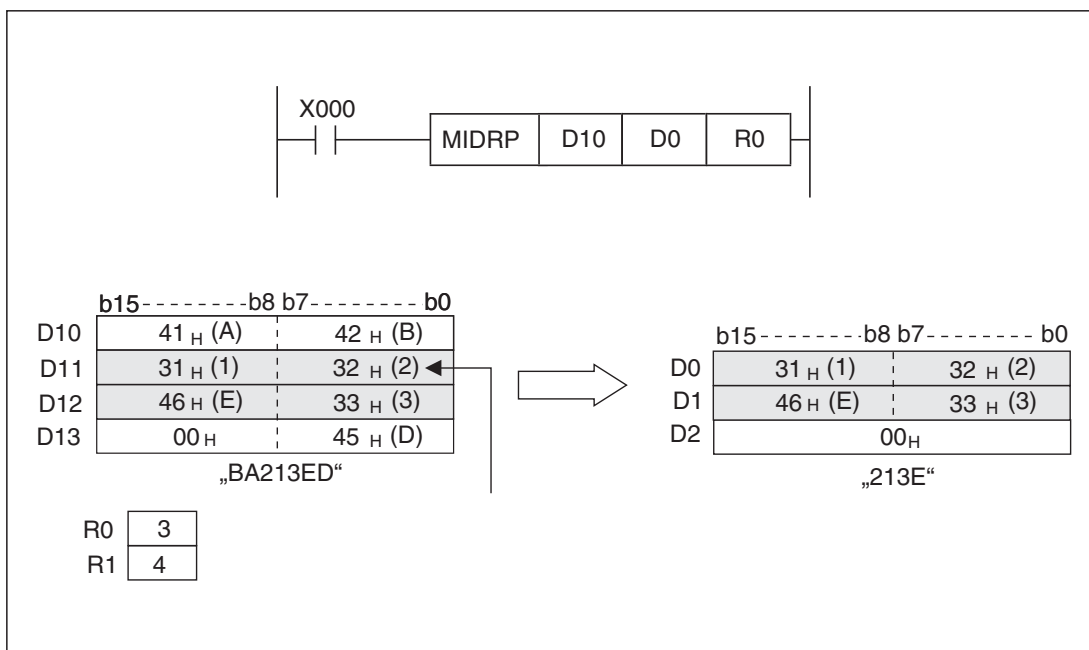


Рис. 7-233:Из строковой величины “BA213ED” копируются знаки “213E”.

7.16.8 ЗАМЕНА ЦЕПОЧКИ СИМВОЛОВ (MIDW)

				MIDW		FNC 207			
				Замена цепочки символов					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S1+	D+	S2+	Имп. инструкция(P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)		●	16бита	32бита	MIDW MIDWP	7

ФУНКЦИЯ

Замена знаков в строковой величине другими знаками

ОПИСАНИЕ

- Команда MIDW записывает знаки из строковой величины, указанной в (S1+), в строковую величину, указанную в (D+).
- Знак извлекается с начала (левого конца) строковой величины, указанной в (S1+).
- Строковые величины в (S1+) и (D+) начинаются со знака в младшем байте и заканчиваются кодом "00H".
- В ((S2+)+0) указывается, начиная с какого знака строковой величины в (D+) должны быть вставлены знаки.
- ((S2+)+1) содержит количество байтов (знаков), которое требуется перезаписать.

Если здесь ввести значение "0", никакие знаки не заменяются. Если значение в ((S2+)+1) превышает количество знаков, которое можно записать в (D+), передается только такое количество знаков, которое может быть записано (рис. 7-236). При значении "-1" передается вся строковая величина из (S1+) в строковую величину в (D+) (рис. 7-237).

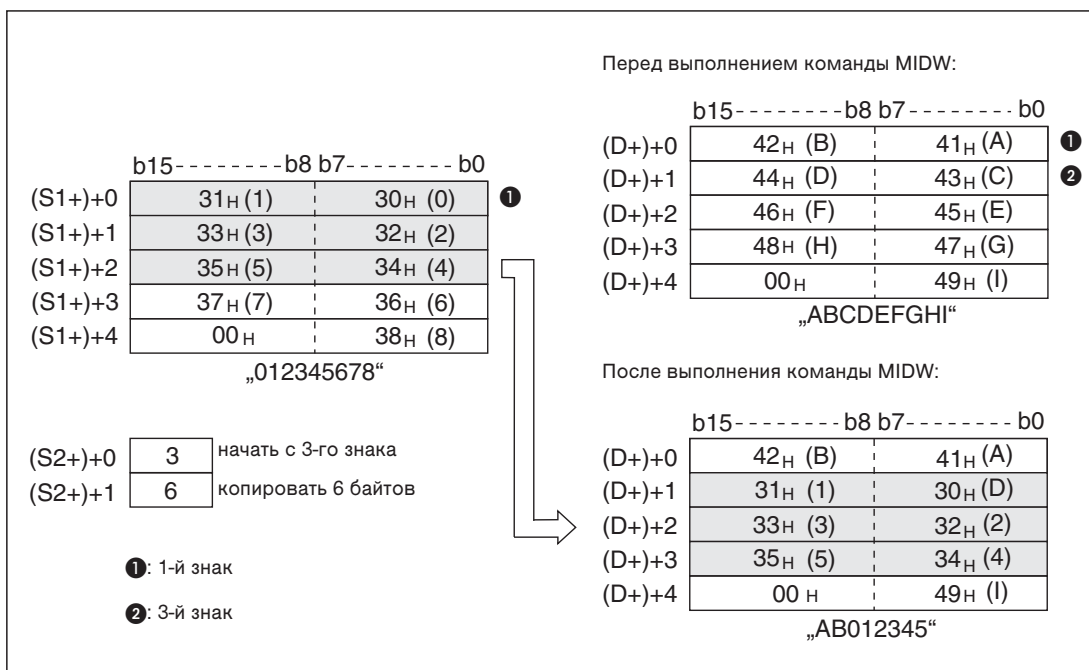


Рис. 7-235:С помощью команды MIDW можно заменять знаки в строковых величинах.

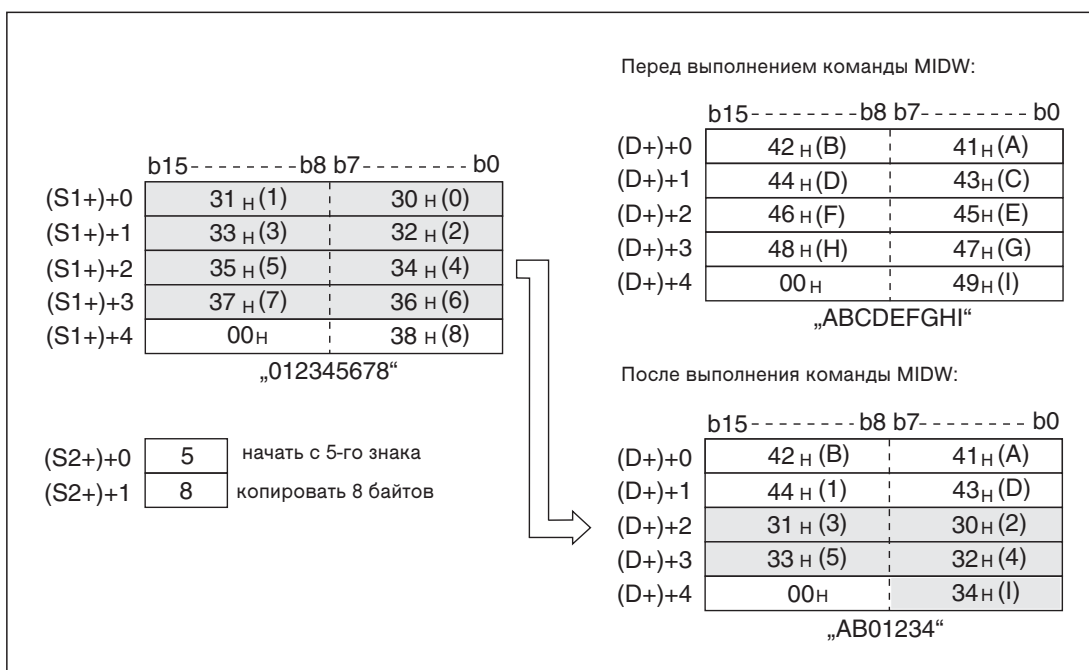


Рис. 7-234:В строковую величину в (D+) вставляются только знаки до кода “00H”. В этом примере требуется начиная с 5-го знака строковой величины в (D+) записать 8 знаков. Однако начиная с 5-го знака в ((D+)+2) можно заменить только 5 знаков. Остальные знаки строковой величины в (S1+) отрезаются.

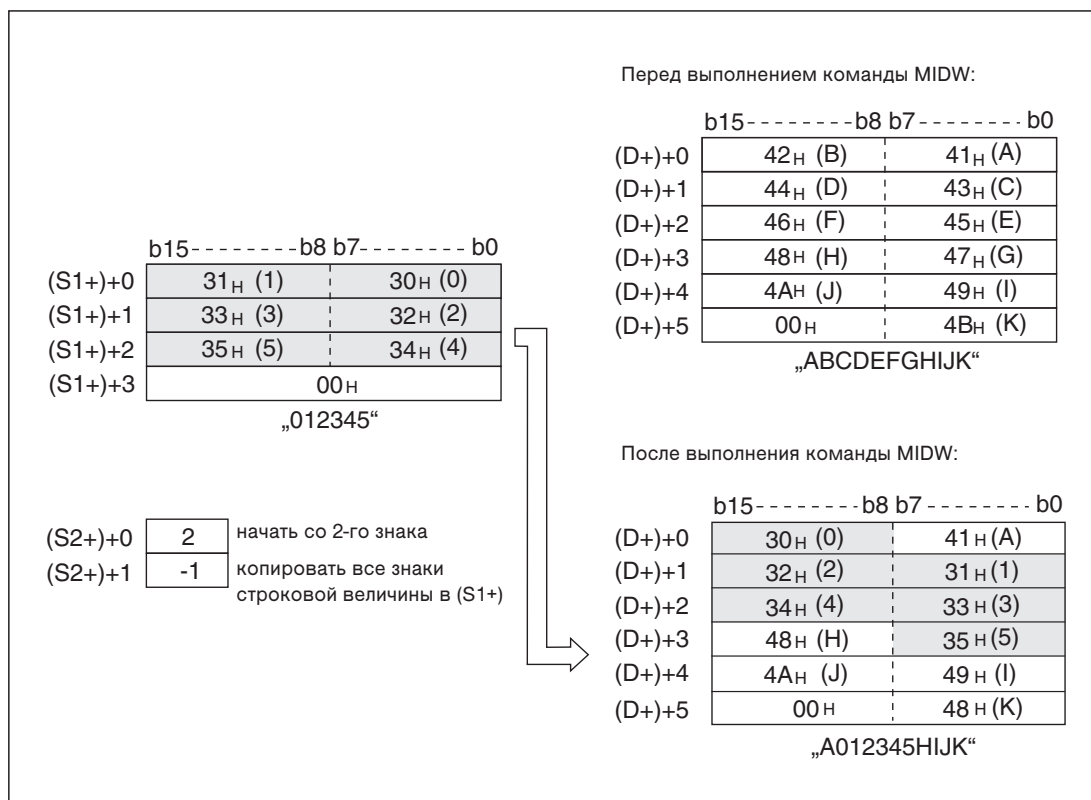


Рис. 7-236: Если ((S2+)+1) имеет значение "-1", копируются все знаки строковой величины, указанной в (S1+), в строковую величину, указанную в (D+), начиная со знака, указанного в ((S2+)+0).

УКАЗАНИЕ

Число знаков в ((S2+)+1) указывается в байтах. Это необходимо учитывать, если требуется считать знаки, сохраненные не в кодировке ASCII и занимающие более одного байта. Например, если требуется считать знаки в кодировке JIS (Японский промышленный стандарт) (1 знак = 16 битов), необходимо умножить количество требуемых знаков на 2 и ввести в ((S2+)+1) результат умножения.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Указанное в ((S2+)+1) количество копируемых знаков больше количества знаков, хранящихся начиная с (S1+).
- В ((S2+)+0) указана позиция знака, которого не имеется в строковой величине, находящейся по адресу начиная с (D+).
- Число знаков в ((S2+)+1) больше длины строковой величины в (S1+).
- В ((S2+)+0) указано отрицательное число.
- В ((S2+)+1) указано значение меньше -1.
- В областях операндов, указанных в (S1+) и (D+), не встречается код "00H".

ПРИМЕР ▾

При включении входа X010 из строковой величины, хранящейся в регистрах начиная с D0, считываются первые четыре знака. Эти знаки вставляются начиная с 3-го знака строковой величины, хранящейся в регистрах начиная с D100. Положение первого перезаписываемого знака указывается в R0, а число знаков - в R1.

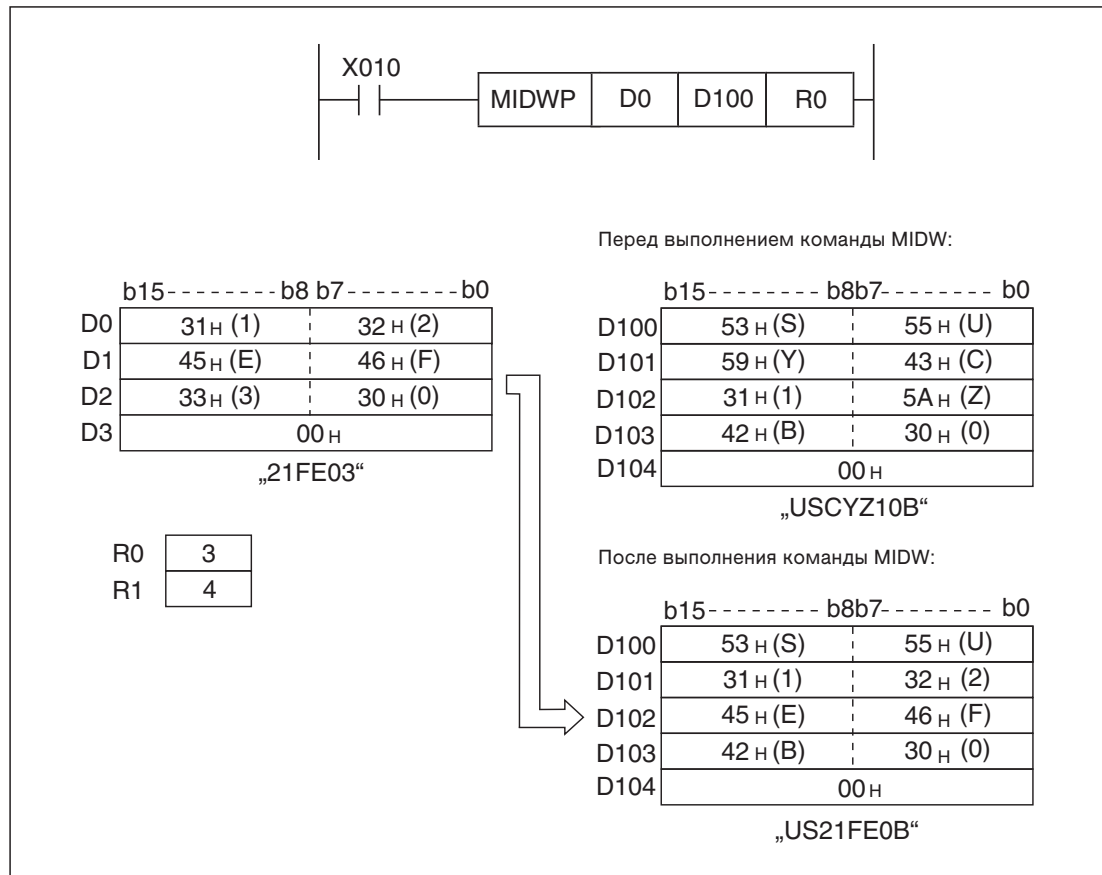
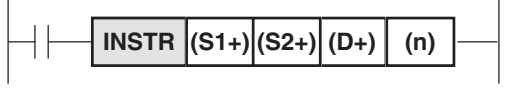


Рис. 7-237: В этом примере в строковой величине, хранящейся начиная с регистра D100, заменяются знаки с 3-го по 6-й.

7.16.9 ПОИСК ЦЕПОЧКИ СИМВОЛОВ (INSTR)

					INSTR		FNC 208			
					Поиск цепочки символов					CPU
Операнды	S1+	S2+	D+	S2+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R, цепочка символов	T, C, D, R	D, R, K, H	●		16бита	32бита	INSTR INSTRP	9	
						●				

Функция

Поиск знаков в строковой величине

Описание

- Команда INSTR ищет в строковой величине, хранящейся по адресу начиная с (S2+), цепочку символов, указанную в (S1+).
- Поиск начинается с позиции, указанной в (n).
- В (D+) вводится позиция, в которой обнаружена искомая цепочка символов. Указывается номер знака. Счет начинается в начале (левой части) строковой величины. (Первый знак проверяемой строковой величины сохранен в младшем байте ((S2+)+0).)

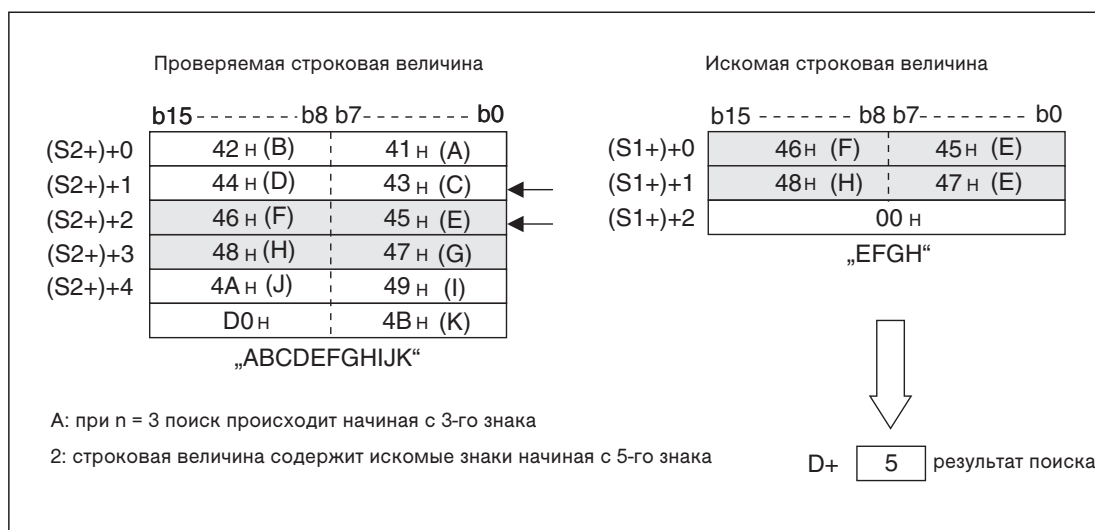


Рис. 7-238: Результат поиска показывает позицию первого знака искомой цепочки символов.

- Если искомая цепочка символов не обнаружена, в (D+) записывается ноль.
- Если в (n) указано отрицательное значение или ноль, команда INSTR не выполняется.

- Цепочку символов можно также непосредственно указать в (S1+):

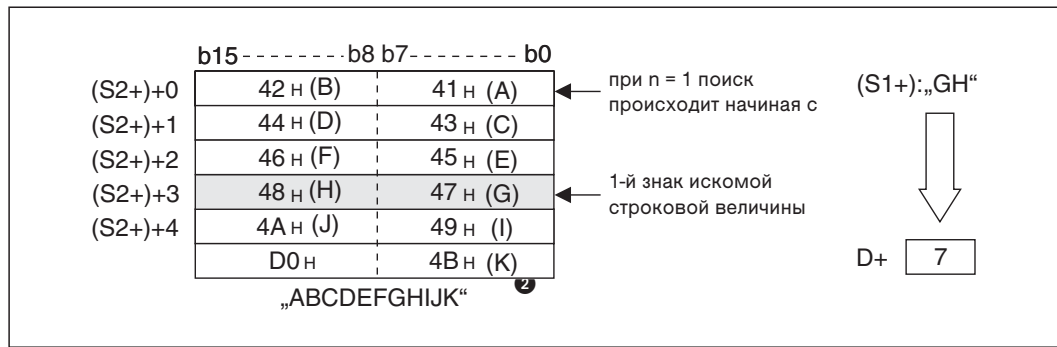


Рис. 7-239: В этом примере непосредственно указанная цепочка символов “GH” найдена начиная с 7-го знака.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки “6706”:

- Указанная в (n) позиция начала поиска превышает количество знаков, хранящихся по адресу начиная с ((S1+)+0).
- В областях операндов, указанных в (S1+) и (S2+), не встречается код “00н”.

ПРИМЕР ▾

Следующая программа при включенном входе X0 ищет в строковой величине, хранящейся начиная с R0, цепочку символов, указанную в D0...D2. Поиск начинается с 5-го знака строковой величины, хранящейся по адресу начиная с R0. Результат поиска записывается в D100.

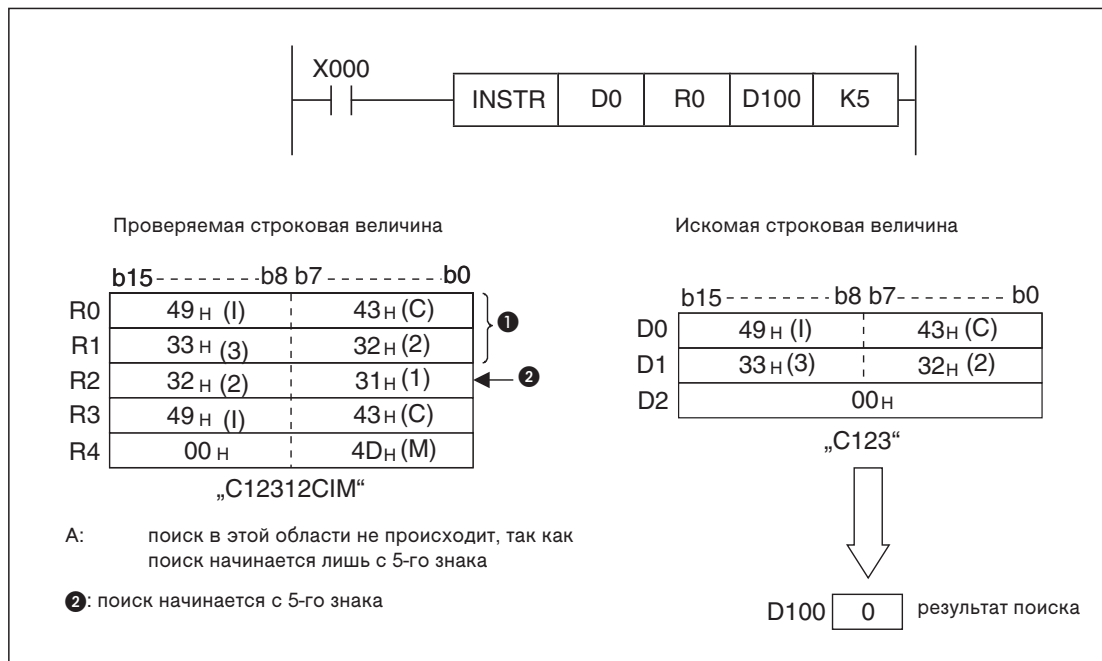
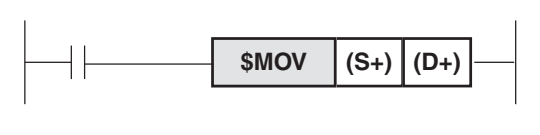


Рис. 7-240: В этом примере искомая цепочка символов не найдена, поэтому результатом поиска является “0”.



7.16.10 ПЕРЕДАЧА ЦЕПОЧКИ СИМВОЛОВ (\$MOV)

		\$MOV		FNC 209			
		Передача цепочки символов					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
			●				●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	●	16бита	32бита	\$MOV \$MOV P	5

Функция

Передача цепочки символов

Описание

- Команда \$MOV передает сохраненные в (S+) байты строковой величины в адрес (D+). При этом переносе за одну операцию передается вся строковая величина, начиная с первого знака (байта) и до конца, помеченного кодом "00H" (конец строковой величины).

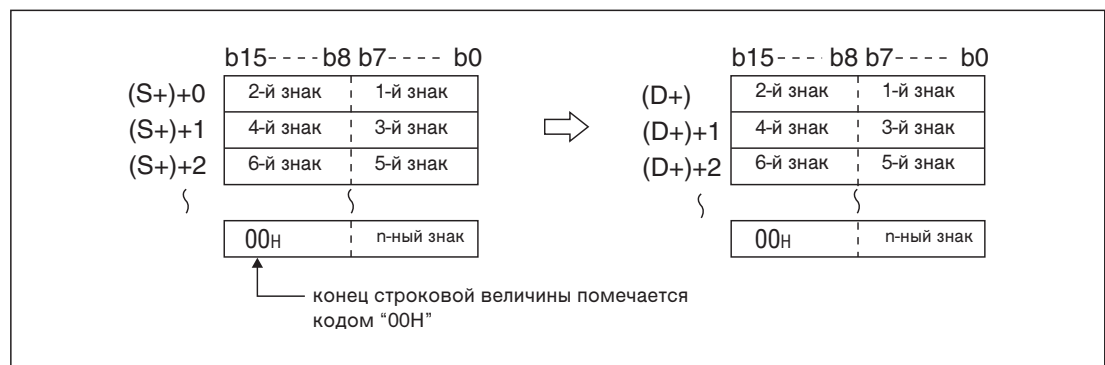


Рис. 7-241: Принцип передачи данных с помощью команды \$MOV

- Команда \$MOV выполняется даже в том случае, если предусмотренные для сохранения области данных от ((S+)+0) до ((S+)+n) перекрываются с областями данных от ((D+)+0) до ((D+)+n). Если хранящаяся в регистрах D10...D13 строковая величина передается в регистры D11...D14, образуется следующий результат.

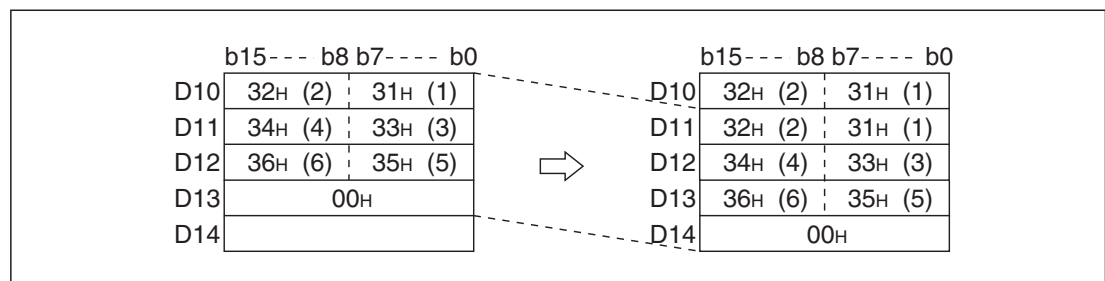


Рис. 7-242: В этом примере содержимое D10 не изменяется.

- Если в строковой величине в младшем байте ((S+)+n) находится код "00н", то при передаче следующий за ним знак не учитывается и "00н" записывается в обоих байтах ((D+)+n).

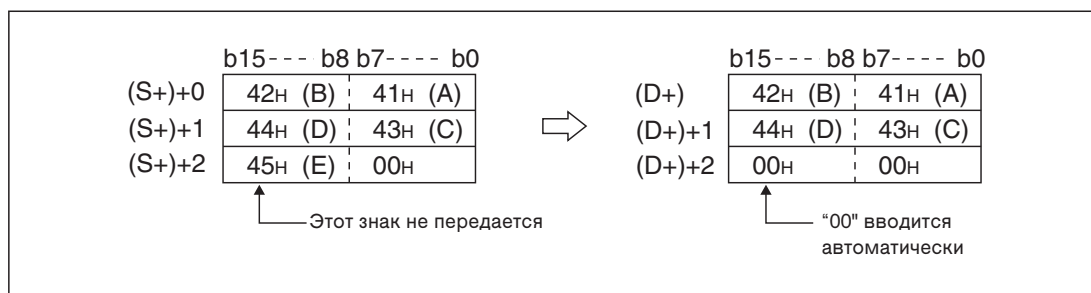


Рис. 7-243: Все знаки после кода "00" игнорируются.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- По адресу (D+) не может быть передана вся строковая величина.
- В области операндов, указанной в (S+), не встречается код "00н".

ПРИМЕР ▾

Следующая программа при включенном входе X0 передает строковую величину, хранящуюся в регистрах с D10 по D12, в регистры данных, начиная с D20.

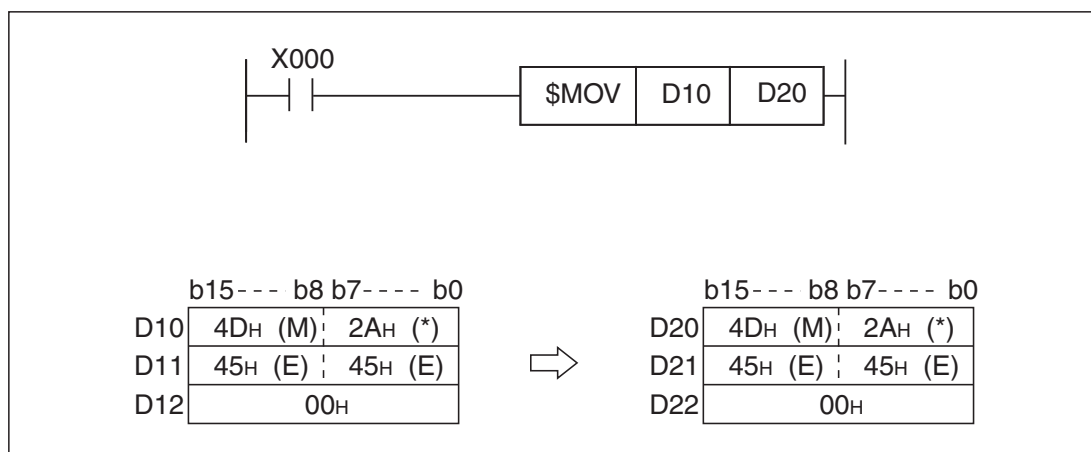


Рис. 7-244: Пример программы для применения команды \$MOV

7.17 КОМАНДЫ ДЛЯ РАБОТЫ С ПЕРЕЧНЯМИ ДАННЫХ

Символ	FNC	Назначение	Разд.
FDEL	210	Стирание данных из перечня данных	
FINS	211	Ввод данных в перечень данных	
POP	212	Считывание данных, введенных в перечень последними	
SFR	213	Перемещение 16-битного слова данных вправо	
SFL	214	Перемещение 16-битного слова данных влево	

Табл. 7-56: Обзор команд для обработки данных

7.17.1 СТИРАНИЕ ДАННЫХ ИЗ ПЕРЕЧНЯ ДАННЫХ (FDEL)

				FDEL		FNC 210							
				Стирание данных из перечня данных									
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U				
											●		
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы						
	T, C, D, R	T, C, D, R	D, R, K, H	●	16бита	32бита	FDEL FDELP		7				

Функция

Стирание блоков данных из перечня данных

Описание

- Команда FDEL стирает n-ые данные в перечне данных, указанном в (D+), и записывает содержимое стертого блока в операнд, указанный в (S+).
- После стирания данных из перечня оставшиеся данные смыкаются. ((D+)+0) содержит количество элементов перечня данных. После стирания это значение уменьшается на 1. Начиная с адреса ((D+)+1) записывается соответствующее количество данных.

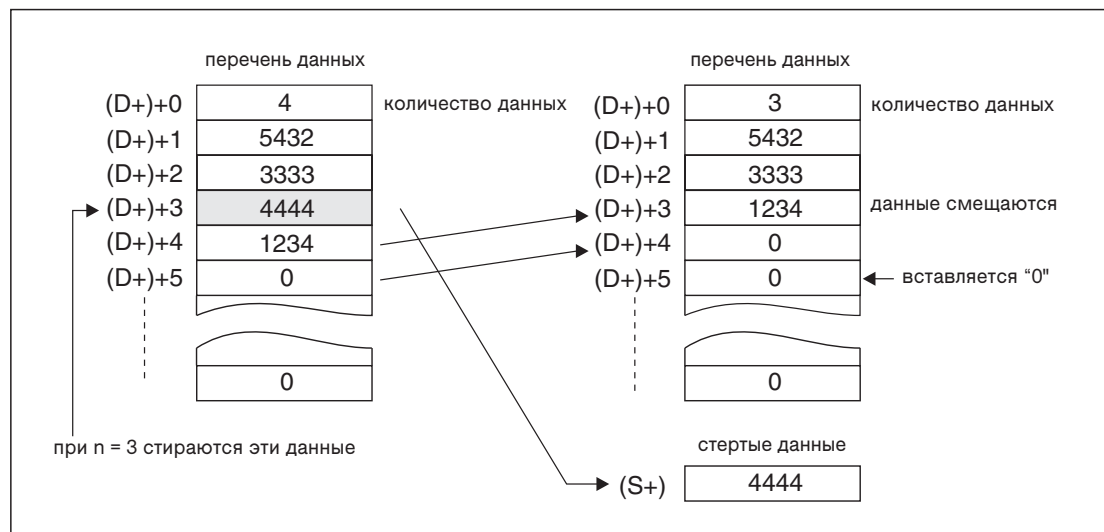


Рис. 7-245: Команда FDEL стирает данные из перечня данных и перемещает их по другому адресу.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Указанная в (n) позиция больше указанного в ((D+)+0) количества элементов в перечне данных.
- Указанный в (n) операнд превышает область операндов, указанную в (D+).
- В (n) указано отрицательное значение или "0".
- Указанное в ((D+)+0) количество элементов перечня равно "0".
- Длина перечня данных превышает область операндов.

ПРИМЕР ▾

Следующая программа при положительном фронте X10 стирает второй элемент из перечня данных, хранящегося в регистрах с D100 по D107. Стертые данные записываются в D0. Команда FDEL выполняется только в том случае, если в перечне данных имеется как минимум 1 элемент или максимум 7 элементов.

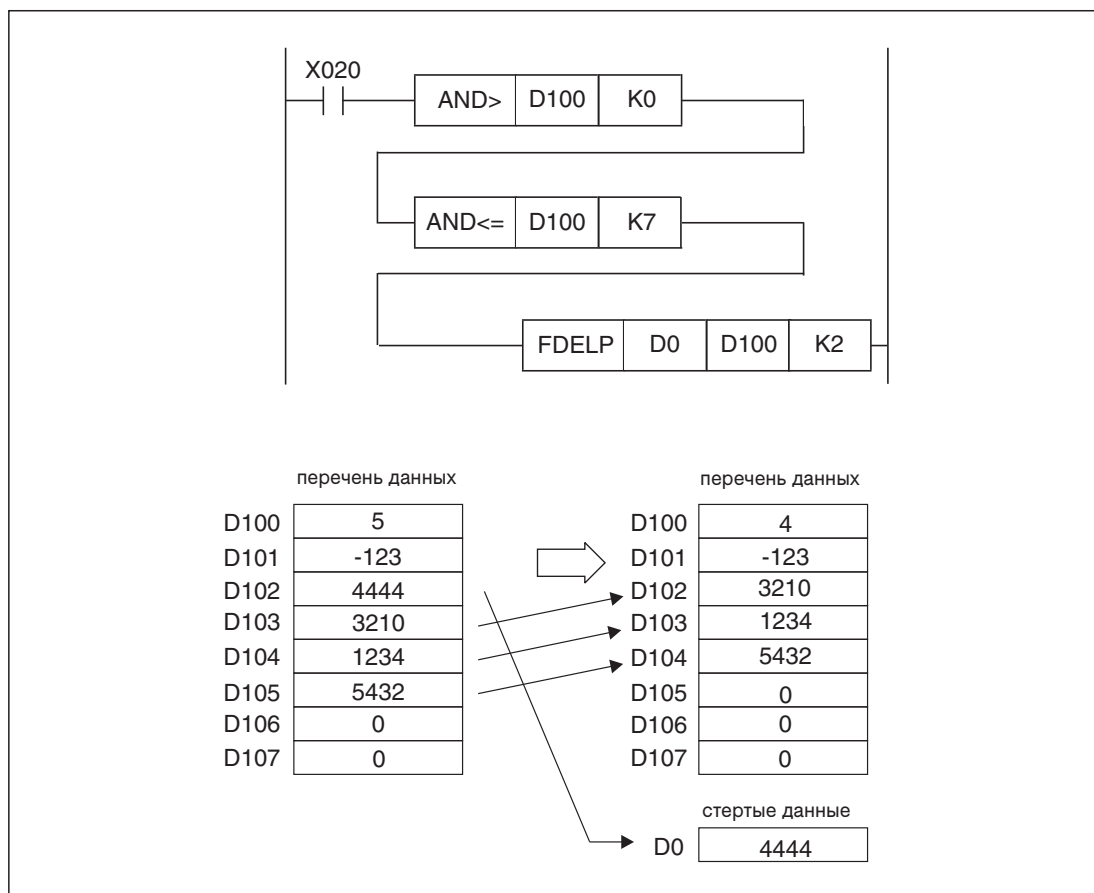



Рис. 7-246:Пример программы для команды FDEL

△

7.17.2 ВВОД ДАННЫХ В ПЕРЕЧЕНЬ ДАННЫХ (FINS)

				FINS		FNC 211			
				Ввод данных в перечень данных					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R, K, H	T, C, D, R	D, R, K, H		●	16бита	32бита	FINS FINSP	7

Функция

Вставление блоков данных в перечень данных

Описание

- Команда FINS вставляет 16-битные данные, указанные в (S+), в n-ую позицию перечня данных, указанного в (D+).
- Данные, следующие за местом вставления, сдвигаются дальше на один адрес. ((D+)+0) содержит количество элементов перечня данных. После вставления это значение повышается на 1. Начиная с адреса ((D+)+1) хранится соответствующее количество данных.

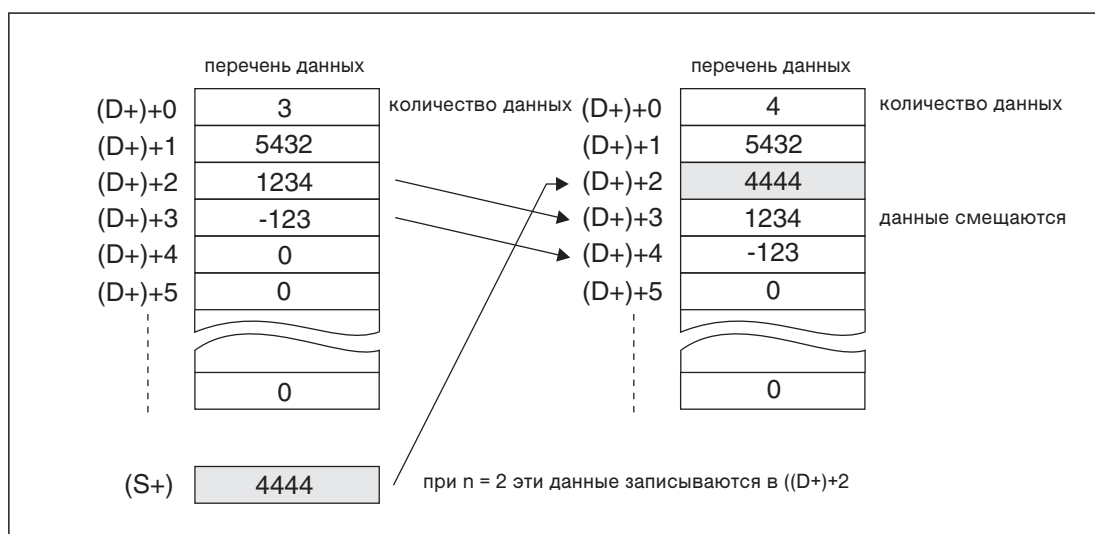


Рис. 7-247: Команда FINS вставляет данные в перечень данных. Имеющиеся данные сдвигаются.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Указанная в (n) позиция больше указанного в ((D+)+0) количества элементов в перечне данных.
- Указанный в (n) операнд превышает область операндов, указанную в (D+).
- В (n) указано отрицательное значение или "0".
- Указанное в ((D+)+0) количество элементов перечня равно "0".
- Длина перечня данных превышает область операндов.

ПРИМЕР ▾

Следующая программа при положительном фронте X10 вставляет содержащиеся в D100 данные в 3-ю позицию перечня данных, хранящегося в регистрах с D0 по D7. Если в перечне данных уже имеется 7 элементов, команда FINS не выполняется.

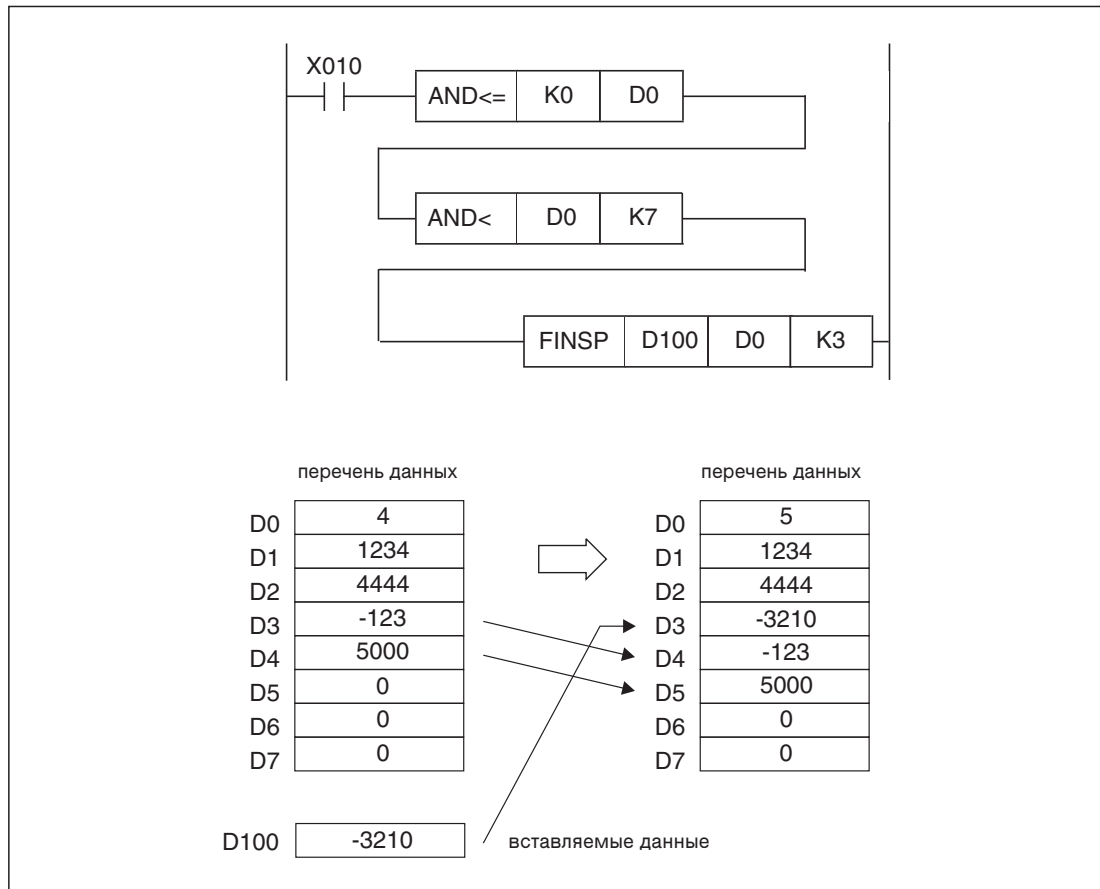


Рис. 7-248:Пример программы для команды FINS

7.17.3 СЧИТЫВАНИЕ ДАННЫХ, ВВЕДЕННЫХ В ПЕРЕЧЕНЬ ПОСЛЕДНИМИ (POP)

				POP		FNC 212				
				Считывание данных, введенных в перечень последними						
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
										●
Операнды	S+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы			
	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z	K, H			16бита	32бита	POP POPP	7	
				●	●					

Функция

Считывание данных, которые были последними введены с помощью команды SFWR (раздел 6.5.8) в память обратного магазинного типа.

Описание

- В (S+) указывается первый адрес области, в которой хранится перечень данных. ((S+)+0) содержит указатель, обозначающий количество элементов в перечне данных. Начиная с ((S+)+1) следует соответствующее количество данных.

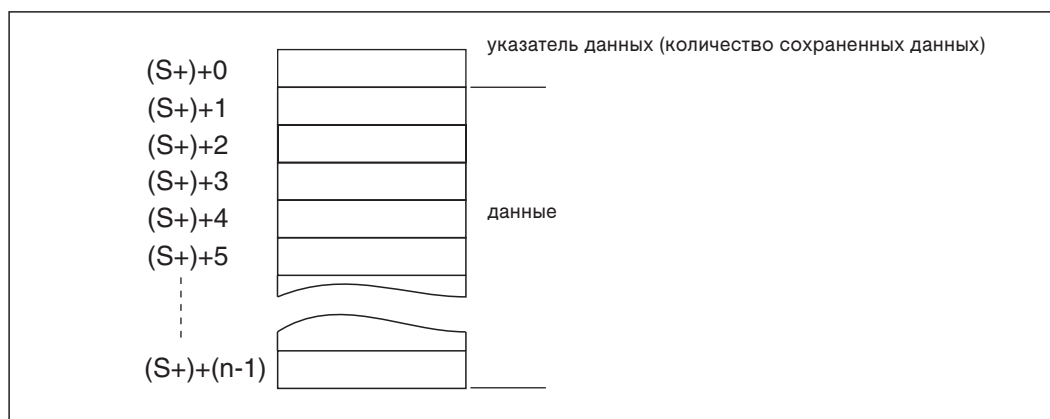


Рис. 7-249: Структура перечня данных

- Считанные из перечня 16-битные данные сохраняются в (D+).
- В (n) указывается длина перечня данных. Значение в (n) может находиться между 2 и 512. (K количеству данных, хранящихся в ((S+)+0), следует прибавить 1, так как засчитывается и ((S+)+0).)

- После считывания данных содержимое указателя данных уменьшается на 1. Содержимое перечня данных команда POP не изменяет.



Рис. 7-250: При считывании данных содержимое перечня данных не изменяется.

УКАЗАНИЕ

Если команда POP выполняется циклически, то после нескольких программных циклов указатель данных достигает значения "0". Поэтому используйте импульсный вариант команды POP (POPP).

Если указатель данных в ((S+)+0) достиг значения "0", устанавливается специальный маркер M8020. В этом случае команда POP не выполняется. Перед выполнением команды POP проверьте с помощью команды сравнения, имеет ли указатель данных в ((S+)+0) по меньшей мере значение "1" и максимум значение "n-1".

Если указатель данных в ((S+)+0) имеет значение "1", то после выполнения команды POP он принимает значение "0" и устанавливается специальный маркер M8020.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Значение в ((S+)+0) больше "n-1".
- Значение в ((S+)+0) отрицательное.

ПРИМЕР ▾

В следующей программе при включении X20 содержимое регистра D20 вводится в перечень данных, хранящийся в регистрах с D100 по D106. (При каждом выполнении команды SFWR данные вводятся в следующий свободный регистр данных и содержимое D100 повышается на 1).

При включении входа X21 считывается значение, введенное в перечень данных последним, и записывается в D10. При каждом выполнении команды POP содержимое D100 понижается на 1.

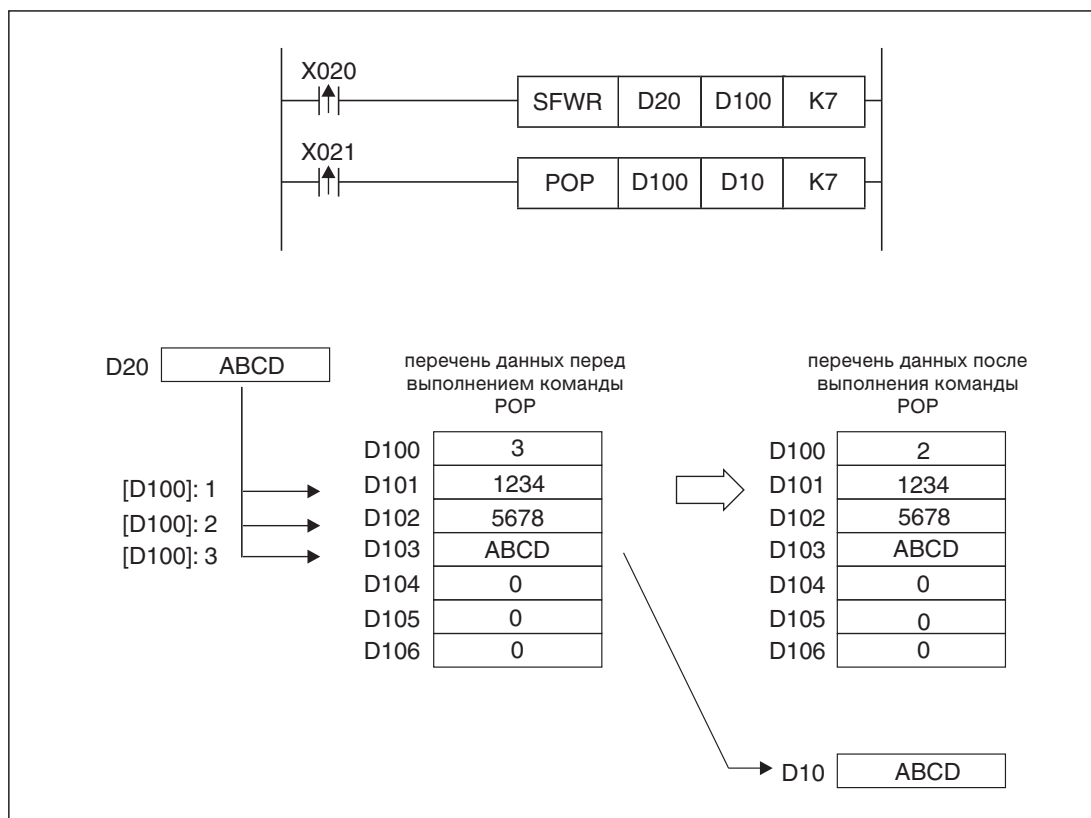


Рис. 7-251: Данные, введенные в перечень данных с помощью команды SFWR, можно извлечь с помощью команды POP.

7.17.4 ПЕРЕМЕЩЕНИЕ 16-БИТНОГО СЛОВА ДАННЫХ ВПРАВО (SFR)

		SFR		FNC 213				
		Перемещение 16-битного слова данных вправо						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	D+	n		Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z		●	16бита	32бита	SFR SFRP	5
					●			

Функция

Перемещение 16-битного слова данных на n битов вправо

Описание

- Команда SFR побитно перемещает заданное в (D+) 16-битовое слово данных на (n) битов вправо.
- Для (n) можно указать значение между 0 и 15. Если для (n) указано значение больше 15, перемещение равно остатку от деления n/16. Например, если n = 18, данные перемещаются на 2 бита ($18/16 = 1$, остаток 2).
- Самые старшие n битов, начиная с бита 15, устанавливаются на 0.
- n-ый перемещаемый бит (бит (n-1)) помещается во флаг переноса M8022.

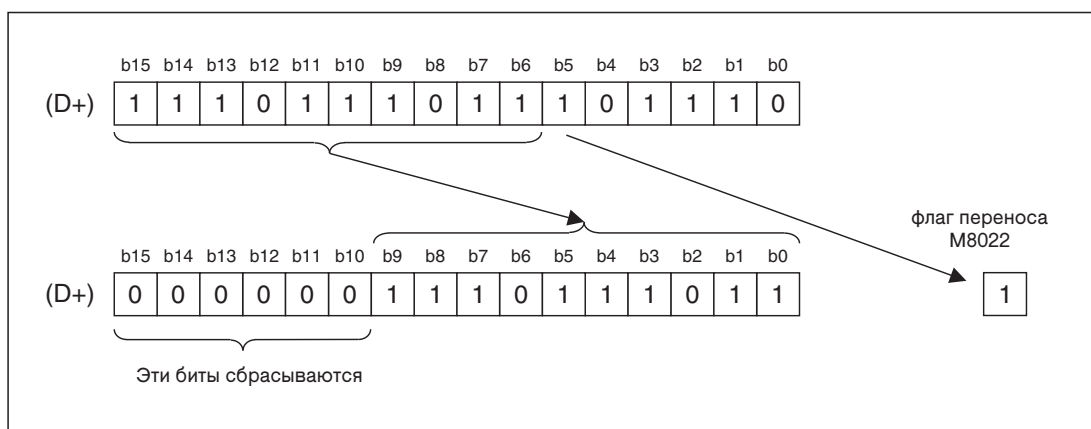


Рис. 7-252: При n = 6 данные смещаются вправо на 6 позиций

- Можно переместить и битовые операнды, обобщенные с помощью коэффициента К.

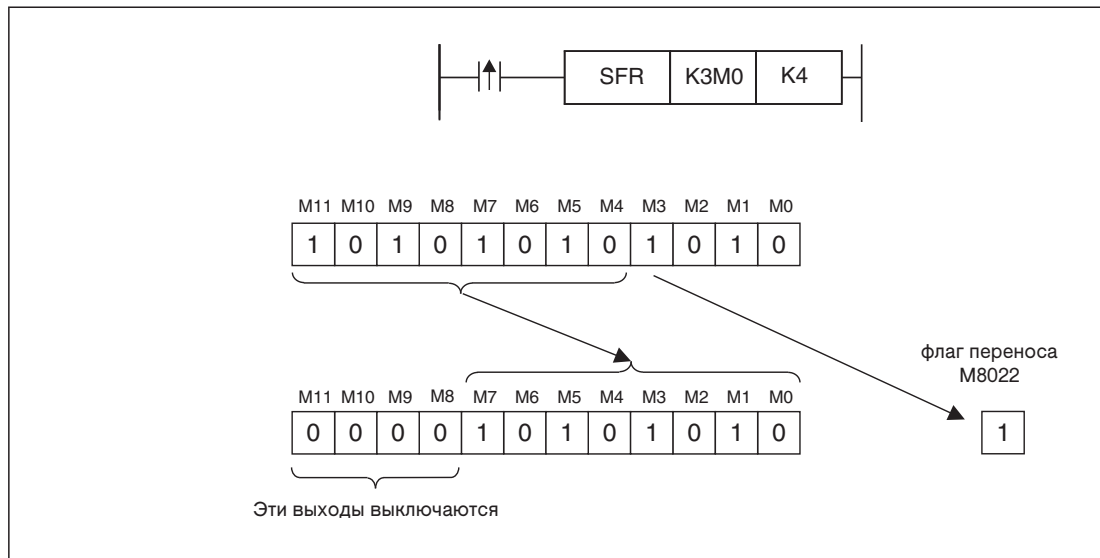


Рис. 7-253: В этом примере смещается содержимое маркеров с M0 по M11.

Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- В (n) указано отрицательное значение.

ПРИМЕР ▾

В следующей программе при возрастающем фронте X20 состояния операндов с Y10 по Y1B смещаются вправо на количество битов, указанное в D0. Состояние Y13 сохраняется во флаге переноса.

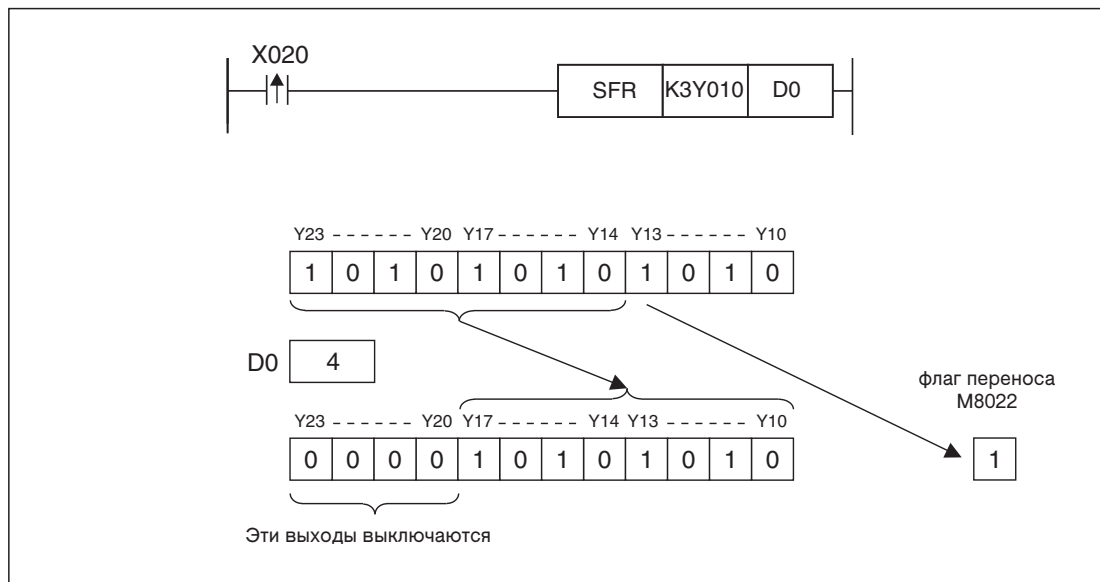


Рис. 7-254: Пример программы для команды SFR; данные сдвигаются вправо на 4 бита.

7.17.5 ПЕРЕМЕЩЕНИЕ 16-БИТНОГО СЛОВА ДАННЫХ ВЛЕВО (SFL)

		SFL		FNC 214				
		Перемещение 16-битного слова данных влево						
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U		
						●		
Операнды	D+	n		Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z		●	16бита	32бита	SFL SFLP	5

Функция

Перемещение 16-битного слова данных на n битов влево

Описание

- Команда SFL побитно перемещает заданное в (D+) 16-битовое слово данных на (n) битов влево.
- Для (n) можно указать значение между 0 и 15. Если для (n) указано значение больше 15, перемещение равно остатку от деления n/16. Например, если n = 18, данные перемещаются на 2 бита ($18/16 = 1$, остаток 2).
- Самый младшие n битов, начиная с бита 0, устанавливаются на 0.
- (n+1)-ый перемещаемый бит (бит n) вносится во флаг переноса M8022.

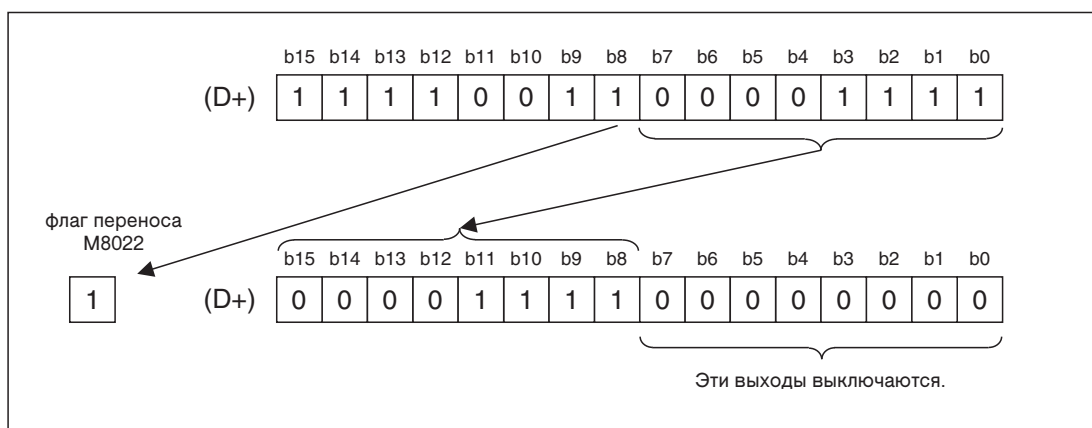


Рис. 7-255: При n = 8 данные смещаются влево на 8 позиций

- Можно переместить и битовые операнды, обобщенные с помощью коэффициента К.

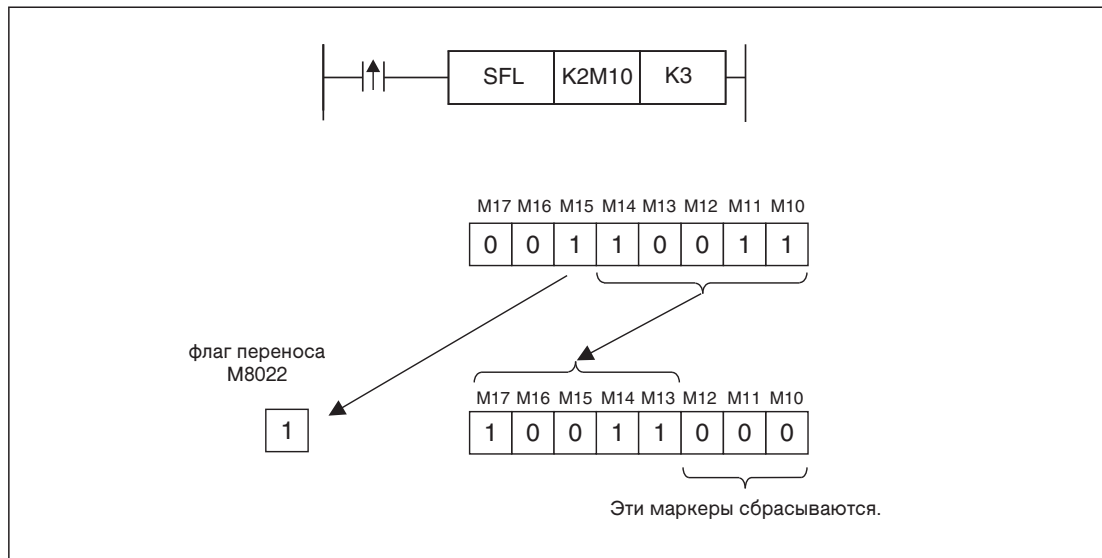


Рис. 7-256: В этом примере смещается содержимое маркеров с M10 по M17.

Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- В (n) указано отрицательное значение.

ПРИМЕР ▾

Следующая программа при возрастающем фронте X20 перемещает состояния выходов с Y10 по Y17 влево на 3 бита. Состояние Y15 вносится во флаг переноса.

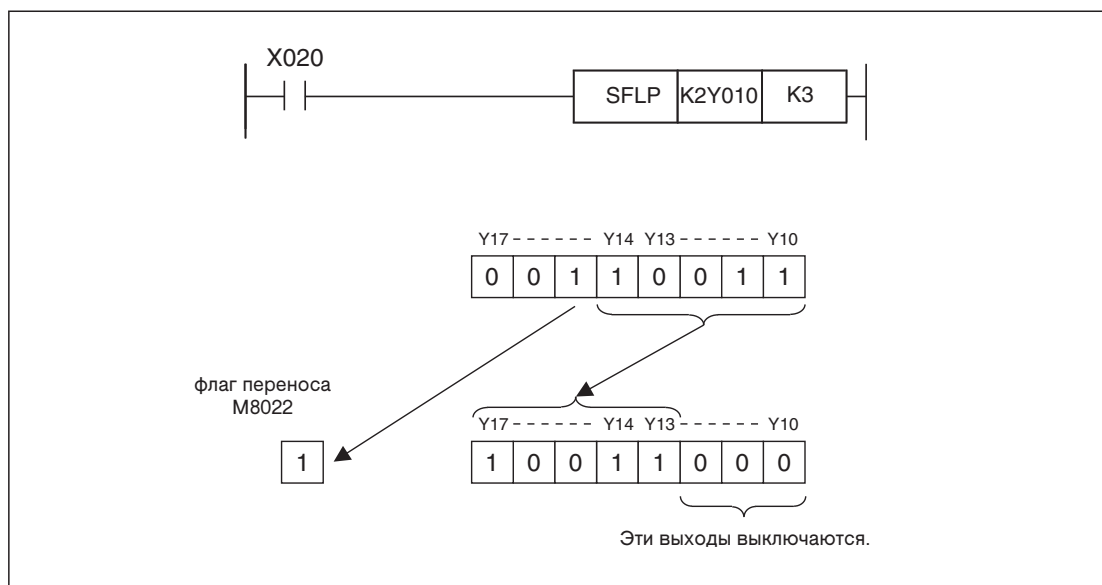


Рис. 7-257: Пример программы для команды SFL; данные смещаются на 3 бита влево.

7.18 ИНСТРУКЦИИ СРАВНЕНИЯ II

ОБЗОР ИНСТРУКЦИЙ FNC 224...246

Символ	FNC	Назначение	Разд.
LD =	224	Инструкция сравнения, равно	7.18.1
LD >	225	Инструкция сравнения, больше	
LD <	226	Инструкция сравнения, меньше	
LD <>	228	Инструкция сравнения, не равно	
LD ≤	229	Инструкция сравнения, меньше или равно	
LD ≥	230	Инструкция сравнения, больше или равно	
AND =	232	И-функция инструкции сравнения, равно	7.18.2
AND >	233	И-функция инструкции сравнения, больше	
AND <	234	И-функция инструкции сравнения, меньше	
AND <>	236	И-функция инструкции сравнения, не равно	
AND ≤	237	И-функция инструкции сравнения, меньше или равно	
AND ≥	238	И-функция инструкции сравнения, больше или равно	
OR =	240	ИЛИ-функция инструкции сравнения, равно	7.18.3
OR >	241	ИЛИ-функция инструкции сравнения, больше	
OR <	242	ИЛИ-функция инструкции сравнения, меньше	
OR <>	244	ИЛИ-функция инструкции сравнения, не равно	
OR ≤	245	ИЛИ- функция инструкции сравнения, меньше или равно	
OR ≥	246	ИЛИ- функция инструкции сравнения, больше или равно	

Табл. 7-27: Обзор инструкций FNC 224...246

7.18.1 ЗАГРУЗКА СРАВНЕНИЯ (LD□)

		LD□		FNC 224 – 230				
		Загрузка сравнения						
Операнды		Имп. инструкция(P)		Обработка		Шаги программы		
		S1+	S2+	Имп. инструкция(P)	Обработка	Шаги программы		
K, H, KnX, KnY, KnM, KnS, T, C, D, R*, специальные модули (U□/G□)*, V, Z				16бита	32бита	LD□	5	
				●	●	DLD□	9	

* Только у FX3U

ФУНКЦИЯ

Загрузка сравниваемых значений и получение результата анализа сравнения LD□

ОПИСАНИЕ

- С помощью LD□-инструкции сравниваются загружаемые в (S1+) и (S2+) значения и выдается результат сравнения.
- Если результат сравнения истинен, включается LD-контакт.
- Если результат сравнения ложен, LD-контакт не включается.
- Таблица отражает соответствие операций сравнения с соответствующим номером функции.

Номер функции	Инструкция		Истинно если	Ложно если
	16бита	32бита		
224	LD =	DLD =	(S1+) = (S2+)	(S1+) <> (S2+)
225	LD >	DLD >	(S1+) > (S2+)	(S1+) ≤ (S2+)
226	LD <	DLD <	(S1+) < (S2+)	(S1+) ≥ (S2+)
228	LD <>	DLD <>	(S1+) <> (S2+)	(S1+) = (S2+)
229	LD ≤	DLD ≤	(S1+) ≤ (S2+)	(S1+) > (S2+)
230	LD ≥	DLD ≥	(S1+) ≥ (S2+)	(S1+) < (S2+)

Табл. 7-28: Обзор LD□-инструкций

УКАЗАНИЕ

LD□-инструкция может применяться как одна LD-инструкция (см.пример).

ПРИМЕР ▾

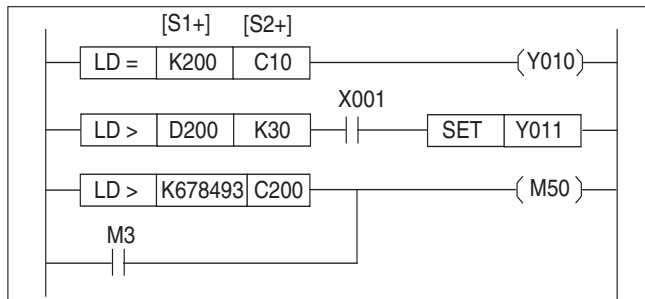


Рис. 7-258:

Пример программирования LD□-инструкции

C000377C

Если значение K200 равно значению счетчика C10, выход Y10 включается.

Если значение в D200 больше, чем значение K30 и включен вход X1, то включается выход Y11 по SET-инструкции. Если значение K678493 больше чем значение счетчика C200 или включен меркер M3, то включается меркер M50.

7.18.2 ЗАГРУЗКА СРАВНЕНИЯ (ANDI)

		AND□		FNC 230 – 238				
		Загрузка сравнения функции "И"						
		CPU		FX1S	FX1N	FX2N	FX2NC	FX3U
				●	●	●	●	●
Операнды	S1+	S2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, R*, специальные модули (U□/G□)*, V, Z			16бита	32бита	AND□	5	
				●	●	DAND□	9	

* Только у FX3U

ФУНКЦИЯ

Загрузка сравниваемых значений по функции "И" и получение результата анализа сравнения AND□

ОПИСАНИЕ

- С помощью AND□-инструкции сравниваются загружаемые в (S1+) и (S2+) значения и при выполнении функции "И" выдается результат сравнения.
- Если результат сравнения истинен, то по результату функции "И" включается выход логической цепи.
- Если результат сравнения ложен, то выход логической цепи "И" не включается.
- Таблица отражает соответствие операций сравнения с соответствующим номером функции.

Номер функции	Инструкция		Истинно если	Ложно если
	16бита	32бита		
232	AND =	DAND =	(S1+) = (S2+)	(S1+) <> (S2+)
233	AND >	DAND >	(S1+) > (S2+)	(S1+) ≤ (S2+)
234	AND <	DAND <	(S1+) < (S2+)	(S1+) ≥ (S2+)
236	AND <>	DAND <>	(S1+) <> (S2+)	(S1+) = (S2+)
237	AND ≤	DAND ≤	(S1+) ≤ (S2+)	(S1+) > (S2+)
238	AND ≥	DAND ≥	(S1+) ≥ (S2+)	(S1+) < (S2+)

Табл. 7-29: Обзор AND□-инструкций

УКАЗАНИЕ

| AND□-инструкция может применяться как одна AND-инструкция

ПРИМЕР ▾

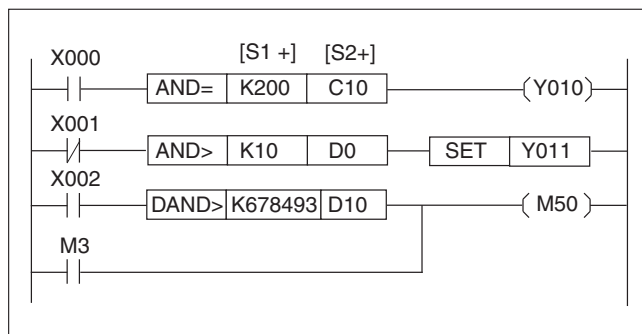


Рис. 7-259:
Пример программирования
AND□-инструкции

C000378C

Если значение K200 равно значению счетчика C10 и включен вход X0, выход Y10 включается.

Если значение в K10 больше, чем значение в D0 и не включен вход X1, то включается выход Y11 по SET-инструкции.

Если значение K678493 больше, чем значение в D10 и D11, и включен вход X2, то включается меркер M50. Меркер M50 включается также, если включен меркер M3.

△

7.18.3 ЗАГРУЗКА СРАВНЕНИЯ (ORI)

		OR□		FNC 240 – 246				
		Загрузка сравнения функции "ИЛИ"						
Операнды		S1+	S2+	Имп. инструкция(P)	Обработка		Шаги программы	
					16бита	32бита	OR□	5
		K, H, KnX, KnY, KnM, KnS, T, C, D, R*, специальные модули (U□/G□)*, V, Z			●	●	DOR□	9

ФУНКЦИЯ

Загрузка сравниваемых значений по функции "ИЛИ" и получение результата анализа сравнения OR□

ОПИСАНИЕ

- С помощью OR□-инструкции сравниваются загружаемые в (S1+) и (S2+) значения и при выполнении функции "ИЛИ" выдается результат сравнения.
- Если результат сравнения истинен, то по результату функции "ИЛИ" включается выход логической цепи.
- Если результат сравнения ложен, то выход логической цепи "ИЛИ" зависит от состояния цепи параллельной цепи сравнения.
- Таблица отражает соответствие операций сравнения с соответствующим номером функции.

Номер функции	Инструкция		Истинно если	Ложно если
	16бита	32бита		
240	OR =	DOR =	(S1+) = (S2+)	(S1) <> (S2+)
241	OR >	DOR >	(S1+) > (S2+)	(S1) ≤ (S2+)
242	OR <	DOR <	(S1+) < (S2+)	(S1) ≥ (S2+)
244	OR <>	DOR <>	(S1+) <> (S2+)	(S1) = (S2+)
245	OR ≤	DOR ≤	(S1+) ≤ (S2+)	(S1) > (S2+)
246	OR ≥	DOR ≥	(S1+) ≥ (S2+)	(S1) < (S2+)

Табл. 7-57: Обзор OR□-инструкций

УКАЗАНИЕ

| OR□-инструкция может применяться как одна OR-инструкция (см.примеры).

ПРИМЕР ▾

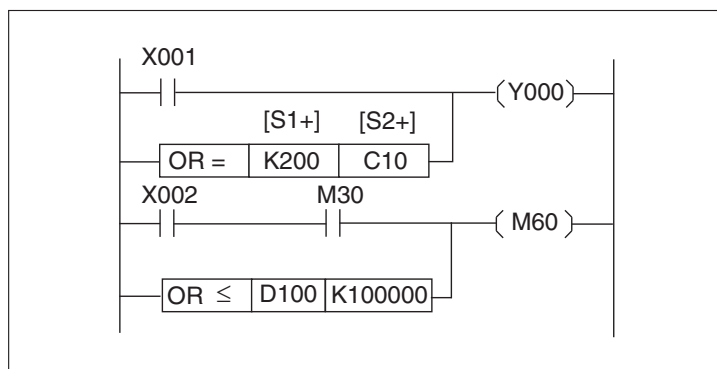


Рис. 7-260:

Пример программирования OR□-инструкции

C000379C

Если значение K200 равно значению счетчика C10 или включен вход X1, то выход Y0 включается. Если значение в D100 меньше или равно значению K100000 или включены вход X2 и меркер M30, то включается меркер M60.

7.19 ОБЗОР КОМАНД КОНТРОЛЯ ДАННЫХ

Символ	FNC	Назначение	Разд.
LIMIT	256	Ограничение диапазона вывода значений	7.19.1
BAND	257	Установка смещения входа	7.19.2
ZONE	258	Установка смещения выхода	7.19.3
SCL	259	Масштабирование значений	7.19.4
DABIN	260	Преобразование числа в кодировке ASCII в двоичное значение	7.19.5
BINDA	261	Преобразование двоичного значения в код ASCII	7.19.6
SCL2	269	Масштабирование значений	7.19.7

Табл. 7-58: Обзор команд контроля данных

7.19.1 ОГРАНИЧЕНИЕ ДИАПАЗОНА ВЫВОДА ЗНАЧЕНИЙ (LIMIT)

					LIMIT		FNC 256			
					Ограничение диапазона вывода значений					CPU
Операнды	S1+	S2+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), K, H		KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)		●	16бита	32бита	LIMIT LIMITP	9
					●	●	●	DLIMIT DLIMITP	17	

Функция

Ограничение диапазона выходных значений 16-битных и 32-битных двоичных данных

Описание для 16-битной обработки (LIMIT, LIMITP)

- Команда LIMIT проверяет указанные в операнде (S+) данные - находятся ли они выше указанного в (S1+) нижнего предела и ниже указанного в (S2+) верхнего предела. В зависимости от результата проверки, в указанном в (D+) операнде сохраняются следующие значения:
 - Если указанное в (S3+) значение данных меньше нижнего предела, указанного в (S1+), в указанном в (D+) операнде сохраняется нижний предел.
 - Если указанное в (S3+) значение данных больше верхнего предела, указанного в (S2+), в указанном в (D+) операнде сохраняется верхний предел.
 - Если указанное в (S3+) значение данных находится между нижним и верхним пределом, в указанном в (D+) операнде сохраняется значение (S3+).
- Указанные в (S1+), (S2+) и (S3+) значения могут находиться между -32768 и 32767.
- Если требуется контролировать только верхний предел, то для нижнего предела в (S1+) необходимо ввести значение -32768.
- Если требуется контролировать только нижний предел, то для верхнего предела в (S2+) необходимо ввести значение 32767.

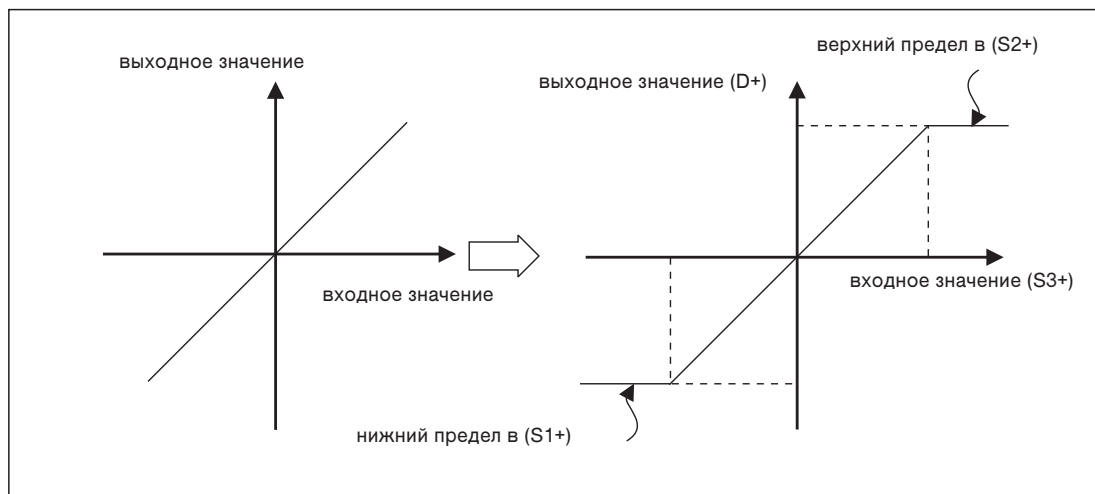


Рис. 7-261:Взаимосвязь между входным и выходным значением без ограничения (слева) и с ограничением (справа) с помощью команды LIMIT

Описание для 32-битной обработки (DLIMIT, DLIMITP)

- Команда DLIMIT проверяет указанные в операндах $((S3+)+1)$ и $((S3)+0)$ данные - находятся ли они выше нижнего предела, указанного в $((S1+)+1)$ и $((S1)+0)$, и ниже верхнего предела, указанного в $((S2+)+1)$ и $((S2)+0)$. В зависимости от результата проверки, в операндах, указанных в $((D+)+1)$ и $((D+)+0)$, сохраняются следующие значения:
 - Если значение данных, указанное в $((S3+)+1)$ и $((S3)+0)$, меньше нижнего предела, указанного в $((S1+)+1)$ и $((S1)+0)$, то в операнде, указанном в $((D+)+1)$ и $((D+)+0)$, сохраняется нижний предел.
 - Если значение данных, указанное в $((S3+)+1)$ и $((S3)+0)$, больше верхнего предела, указанного в $((S2+)+1)$ и $((S2)+0)$, то в операнде, указанном в $((D+)+1)$ и $((D+)+0)$, сохраняется верхний предел.
 - Если входное значение, указанное в $((S3+)+1)$ и $((S3)+0)$, находится между нижним и верхним пределом, то в операнде, указанном в $((D+)+1)$ и $((D+)+0)$, сохраняется входное значение.
- В $((S1+)+1)$ и $((S1)+0)$, $((S2+)+1)$ и $((S2)+0)$, а также в $((S3+)+1)$ и $((S3)+0)$ можно указывать значения между -2147483648 и 2147483647.
- Если требуется контролировать только верхний предел, то для нижнего предела в $((S1+)+1)$ и $((S1)+0)$ следует ввести значение -2147483648.
- Если требуется контролировать только нижний предел, то для верхнего предела в $((S2+)+1)$ и $((S2)+0)$ следует ввести значение 2147483647.

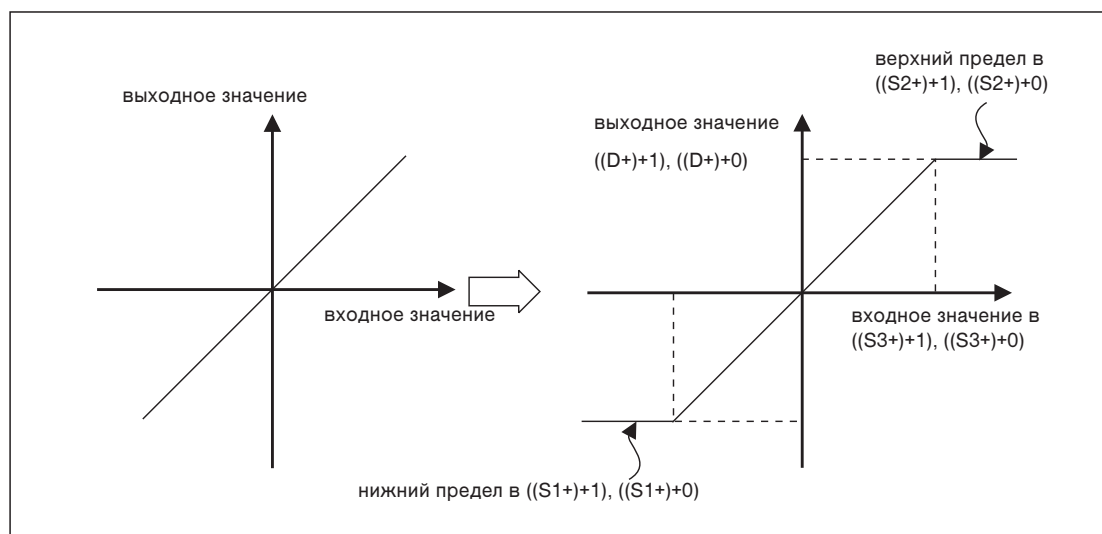


Рис. 7-262: Взаимосвязь между входным и выходным значением без ограничения (слева) и с ограничением (справа) с помощью команды LIMIT

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- При 16-битной обработке указанный в $(S1+)$ нижний предел больше верхнего предела, указанного в $(S2+)$.
- При 32-битной обработке указанный в $((S1+)+1)$ и $((S1)+0)$ нижний предел больше верхнего предела, указанного в $((S2+)+1)$ и $((S2)+0)$.

ПРИМЕР ▾

В следующей программе значение двоично-десятичных данных на входах с X20 по X37 ограничивается нижним пределом "500" и верхним пределом "5000". Выходное значение сохраняется в D1.

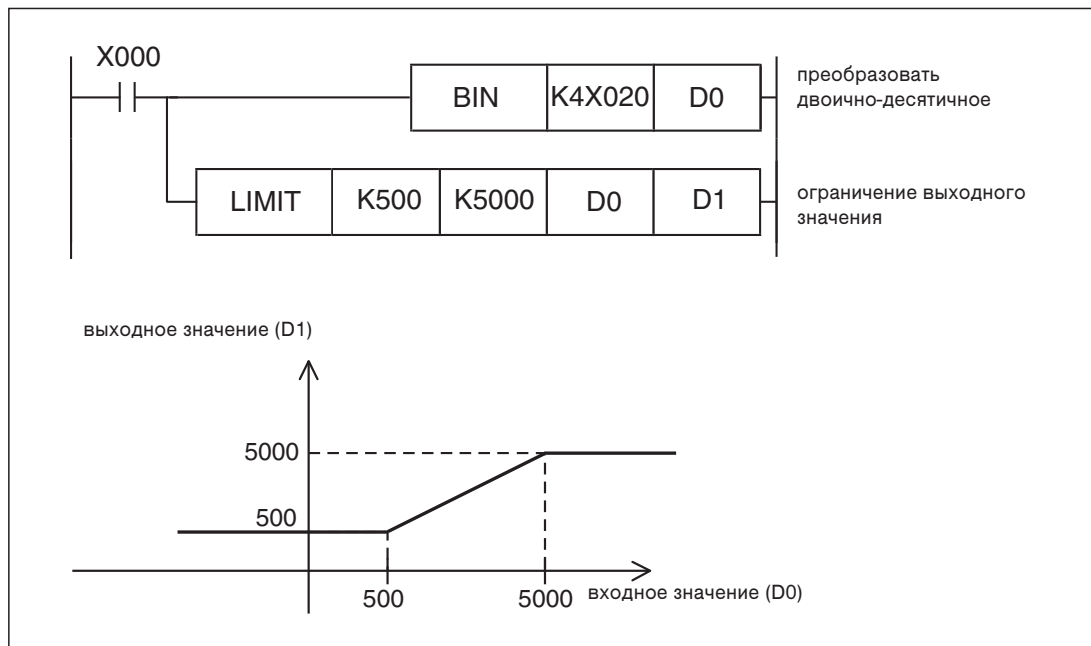


Рис. 7-263:Эта программа ограничивает значение в D1 пределами 500 и 5000.



ПРИМЕР ▾

В следующей программе значение двоично-десятичных данных на входах с X20 по X57 ограничивается нижним пределом "10000" и верхним пределом "1000000". Выходное значение сохраняется в D11 и D10.

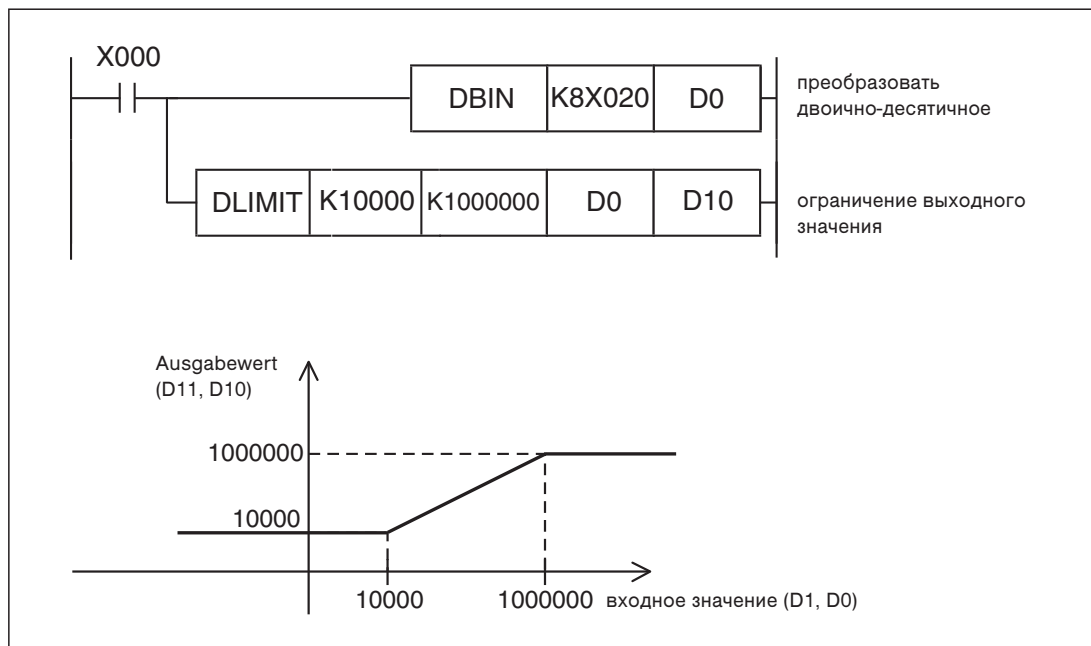
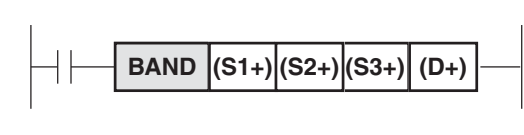


Рис. 7-264:Эта программа ограничивает значение в D11 и D10 пределами 10000 и 100000.



7.19.2 УСТАНОВКА СМЕЩЕНИЯ ВХОДА (BAND)

					BAND		FNC 257			
					Установка смещения входа					CPU
Операнды	S1+	S2+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), K, H	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	●		16бита	32бита	BAND BANDP	9	
						●	●	DBAND DBANDP	17	

Функция

Смещение входных 16-битных или 32-битных двоичных данных

Описание для 16-битной обработки (BAND, BANDP)

- Команда BAND вычитает из входного значения (16-битного двоичного числа), указанного в (S3+), нижнюю (отрицательную) и верхнюю (положительную) величину смещения. Нижняя величина смещения указана в (S1+), а верхняя величина смещения - в (S2+). В зависимости от входного значения, в указанном в (D+) операнде сохраняется следующий результат:
 - Если значение данных в (S3+) меньше нижней величины смещения в (S1+), в указанный в (D+) операнд записывается разность (S3+) - (S1+).
 - Если значение данных в (S3+) больше верхней величины смещения в (S2+), в указанный в (D+) операнд записывается разность (S3+) - (S2+).
 - Если значение данных в (S3+) находится в пределах обеих зон смещения, в указанный в (D+) операнд записывается значение 0.

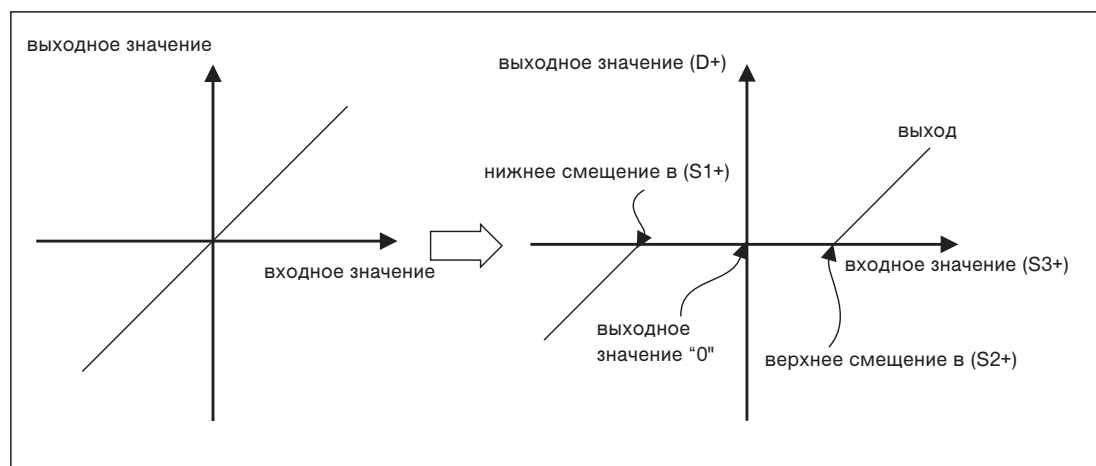


Рис. 7-265: Взаимосвязь между входным и выходным значением без ограничения (слева) и с ограничением (справа) командой BAND

- Указанное в (S1+), (S2+) и (S3+) значение может находиться между -32768 и 32767.

- Если разность находится вне диапазона от -32768 до 32767, происходит следующий процесс:
 - При занижении значения -32768 вычитание продолжается со значения 32767, т. е. остаток вычитается уже из значения 32767. Например, если в (S3+) указано значение -32768, и из него вычитается (S1+) со значением 10, получаем:
 $-32768 - 10 = 8000H - AH = 7FF6H = 32758$.
 - При превышении значения 32767 вычитание продолжается со значения -32768, т. е. остаток вычитается уже из значения -32768.

Описание для 32-битной обработки (DBAND, DBANDP)

- Команда DBAND вычитает из входного значения (32-битного двоичного числа), указанного в ((S3+)+1) и ((S3)+0), нижнюю (отрицательную) и верхнюю (положительную) величину смещения. Нижняя величина смещения указана в ((S1+)+1) и ((S1)+0), а верхняя величина смещения - в ((S2+)+1) и ((S2)+0). В зависимости от входного значения, в операнд, указанный в ((D+)+1) и ((D)+0), записывается следующий результат:
 - Если значение данных в ((S3+)+1) и ((S3)+0) меньше нижней величины смещения в ((S1+)+1) и ((S1)+0), то в операнд, указанный в ((D+)+1) и ((D)+0), записывается разность $[(S3+)+1], ((S3)+0) - ((S1+)+1), ((S1)+0)$.
 - Если значение данных в ((S3+)+1) и ((S3)+0) больше верхней величины смещения в ((S2+)+1) и ((S2)+0), то в операнд, указанный в ((D+)+1) и ((D)+0), записывается разность $[(S3+)+1], ((S3)+0) - ((S2+)+1), ((S2)+0)$.
 - Если значение данных в ((S3+)+1) и ((S3)+0) находится в пределах обеих зон смещения, то в операнд, указанный в ((D+)+1) и ((D)+0), записывается значение 0.

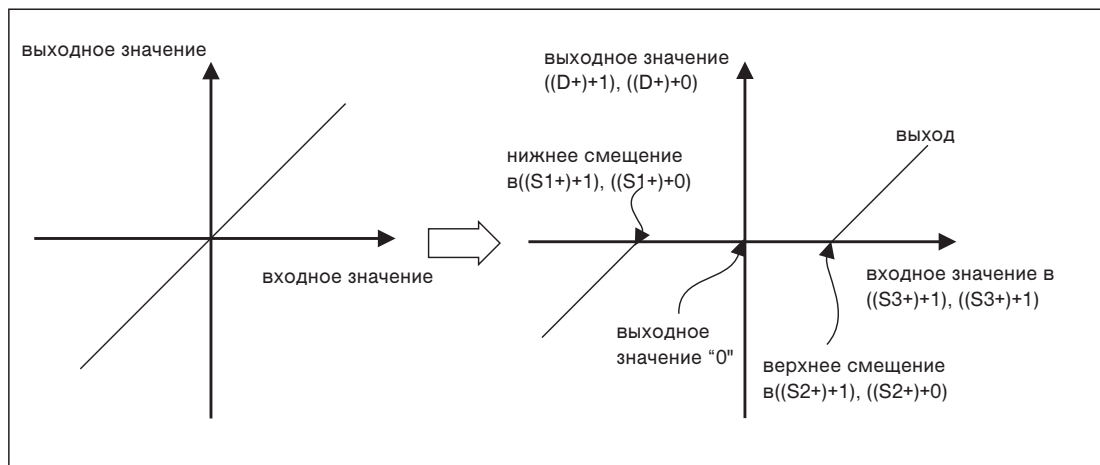


Рис. 7-266:Взаимосвязь между входным и выходным значением без ограничения (слева) и с ограничением (справа) командой DBAND

- В ((S1+)+1) и ((S1)+0), ((S2+)+1) и ((S2)+0), а также в ((S3+)+1) и ((S3)+0) можно указывать значения между -2147483648 и 2147483647.

- Если разность находится вне диапазона от -2147483648 до 2147483647, происходит следующий процесс:
 - При занижении значения -2147483648 вычитание продолжается с числа 2147483647, т. е. остаток вычитается уже из 2147483647. Например, если в ((S3+)+1) и ((S3)+0), указано значение -2147483648 и из него вычитается ((S1+)+1) и ((S1)+0) со значением 1000, получаем:
 $-2147483648 - 1000 = 80000000H - 3E8H = 7FFFFC18H = 2147482648$.
 - При превышении значения 2147483647 вычитание продолжается со значения -2147483648, т. е. остаток вычитается уже из -2147483648.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- При 16-битной обработке указанное в (S1+) нижнее смещение больше верхнего смещения, указанного в (S2+).
- При 32-битной обработке указанное в ((S1+)+1) и ((S1)+0) нижнее смещение больше верхнего смещения, указанного в ((S2+)+1) и ((S2)+0).

ПРИМЕР ▾

В следующей программе при включенном входе X0 из двоично-десятичных данных на входах с X20 по X37 вычитается нижняя (отрицательная) величина смещения -1000 и верхняя (положительная) величина смещения 1000. Результат сохраняется в D1.

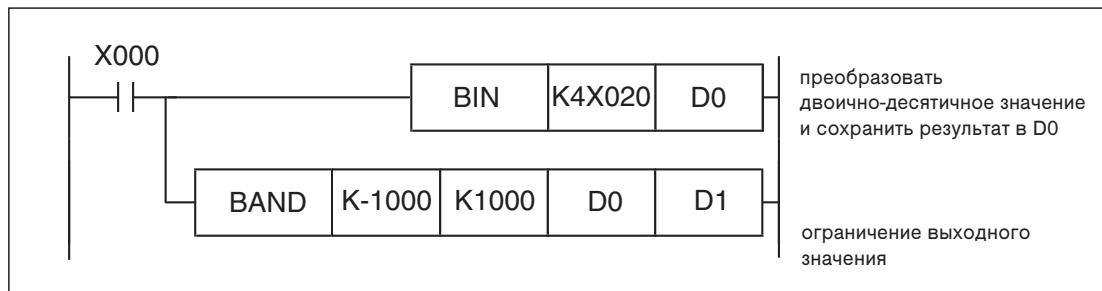


Рис. 7-267: Эта программа не сохраняет в D1 все входные значения между -1000 и 1000.

ПРИМЕР ▾

В следующей программе при включенном входе X0 из двоично-десятичных данных на входах с X20 по X57 вычитается нижняя (отрицательная) величина смещения -10000 и верхняя (положительная) величина смещения 10000. Результат сохраняется в D10 и D11.

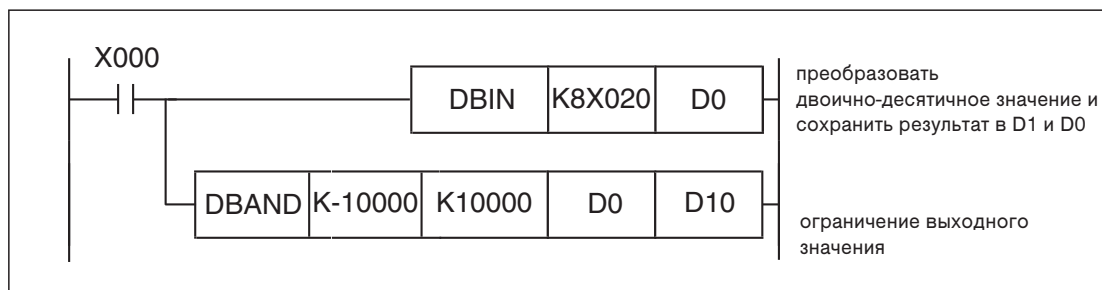
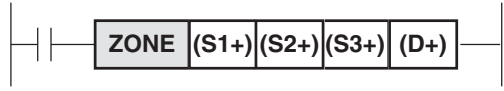


Рис. 7-268: С помощью этой программы все входные значения между -10000 и 10000 не сохраняются в D11 и D10.

7.19.3 УСТАНОВКА СМЕЩЕНИЯ ВЫХОДА (ZONE)

					ZONE		FNC 258		
Установка смещения выхода									
CPU					FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S1+	S2+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), K, H		KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	●	16бита	32бита	ZONE ZONEP	9
					●	●	DZONED DZONEP	17	

Функция

Смещение выходных 16-битных или 32-битных двоичных данных

Описание для 16-битной обработки (ZONE, ZONEP)

- Команда ZONE прибавляет к 16-битному двоичному содержимому операнда, указанного в (S3+), отрицательную и положительную величину смещения. Отрицательная величина смещения указана в (S1+), а положительная величина смещения - в (S2+). В зависимости от входного значения, в указанном в (D+) операнде сохраняется следующий результат:
 - Если значение данных в (S3+) меньше 0, в указанный в (D+) операнд записывается результат сложения (S3+) + (S1+).
 - Если значение данных в (S3+) больше 0, в указанный в (D+) операнд записывается результат сложения (S3+) + (S2+).
 - Если значение данных в (S3+) равно 0, в указанный в (D+) операнд также записывается значение 0.

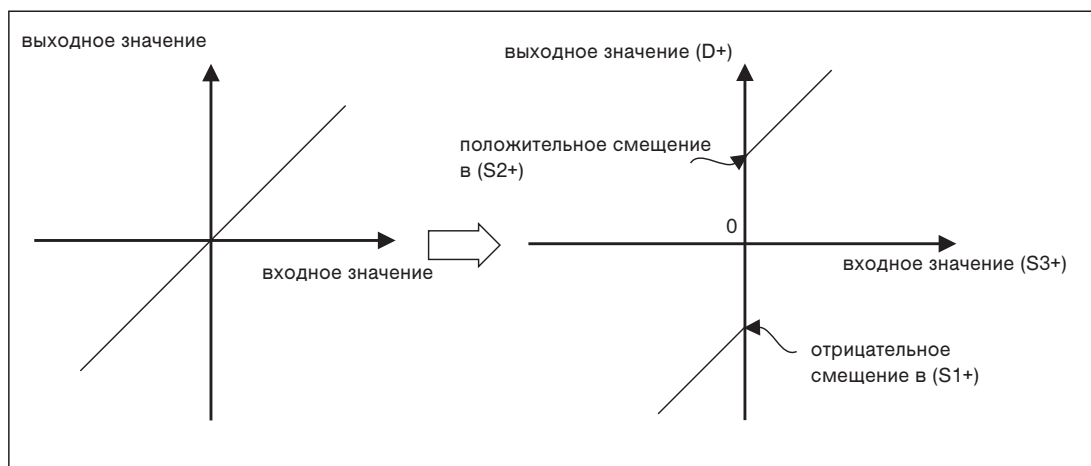


Рис. 7-269:Взаимосвязь между входным и выходным значением без ограничения (слева) и с ограничением (справа) командой ZONE

- Если сумма выходит за пределы диапазона от -32768 до 32767, происходит следующий процесс:
 - При занижении значения -32768 сложение продолжается с числа 32767. Например, если в (S3+) указано значение -32768, и к нему прибавляется (S1+) со значением -100, получаем:
 $-32768 + (-100) = 8000H + FF9CH = 7F9CH = 32668$
 - При превышении значения 32767 сложение продолжается с числа -32768.

Описание для 32-битной обработки (DZONE, DZONEP)

- Команда ZONE прибавляет к 16-битному двоичному числу в ((S3+)+1) и ((S3)+0) отрицательную и положительную величину смещения. Отрицательная величина смещения указана в ((S1+)+1) и ((S1)+0), а положительная величина смещения - в ((S2+)+1) и ((S2)+0). В зависимости от входного значения, в ((D+)+1) и ((D)+0) записывается следующий результат:
 - Если значение данных в ((S3+)+1) и ((S3)+0) меньше 0, в ((D+)+1) и ((D)+0) записывается сумма $[(S3+)+1], ((S3)+0) + ((S1+)+1), ((S1)+0)]$.
 - Если значение данных в ((S3+)+1) и ((S3)+0) больше 0, в ((D+)+1) и ((D)+0) записывается сумма $[(S3+)+1], ((S3)+0) + ((S2+)+1), ((S2)+0)]$.
 - Если значение данных в ((S3+)+1) и ((S3)+0) равно 0, в ((D+)+1) и ((D)+0) также записывается 0.

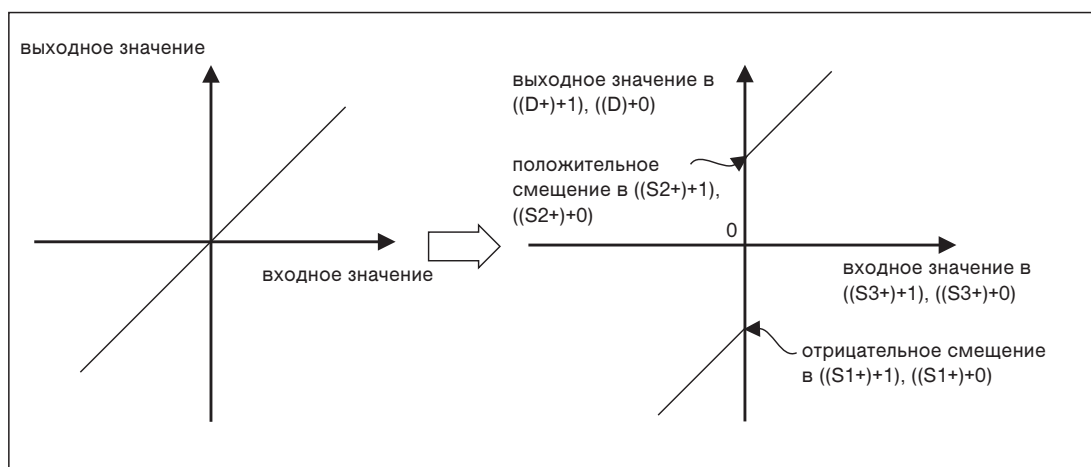


Рис. 7-270:Взаимосвязь между входным и выходным значением без ограничения (слева) и с ограничением (справа) командой DZONE

- Если результат сложения выходит за пределы диапазона от -2147483648 до 2147483647, происходит следующий процесс:
 - При занижении значения -2147483648 сложение продолжается со значения 2147483647. Например, если в ((S3+)+1) и ((S3)+0) указано значение -2147483648, и к нему прибавляется ((S1+)+1) и ((S1)+0) со значением -1000, получаем:
 $-2147483648 + (-1000) = 80000000H + FFFFFFFC18H = 7FFFFFFC18H = 2147482648$.
 - превышении значения 2147483647 сложение продолжается со значения -2147483648.

ПРИМЕР ▾

В следующей программе при включении входа X0 к двоично-десятичным данным на X20...X37 прибавляется отрицательная величина смещения -100 и положительная величина смещения 100. Результат сохраняется в D1.

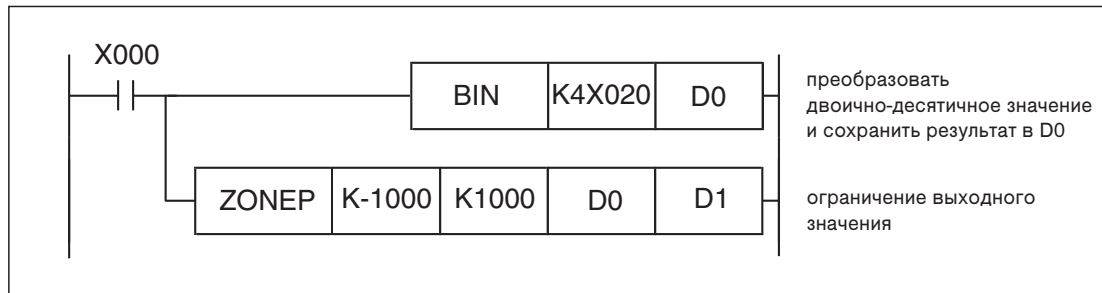


Рис. 7-271:С помощью этой программы не выводятся выходные значения между -1000 и 1000.

Если значение в D0 больше 0, в D1 записывается значение "D0 + 100".

Если значение в D0 меньше 0, в D1 записывается значение "D0 + (-100)".

Если значение в D0 равно 0, в D1 записывается "0".

△

ПРИМЕР ▾

Следующая программа при включении входа X0 прибавляет к двоично-десятичным данным на X20...X57 отрицательную величину смещения -10000 и положительную величину смещения 10000. Результат сохраняется в D10 и D11.

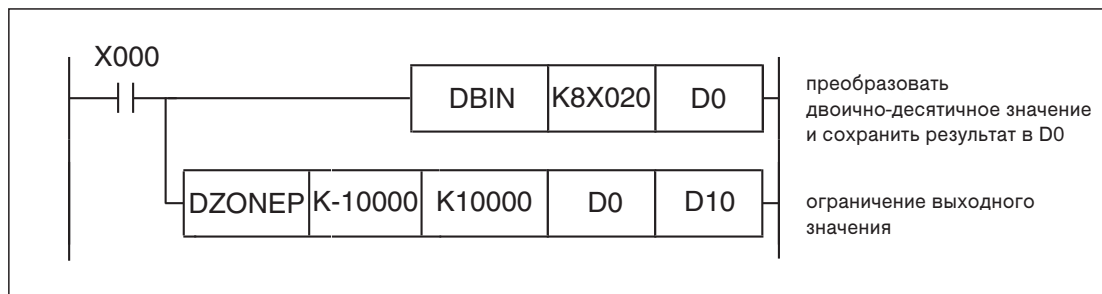


Рис. 7-272:С помощью этой программы не выводятся выходные значения между -10000 и 10000.

Если значение в D1 и D0 больше 0, в D11 и D10 записывается значение "D1, D0 + 10000".

Если значение в D1 и D0 меньше 0, в D11 и D10 записывается значение "D0, D1 + (-10000)".

Если значение в D1 и D0 равно 0, в D11 и D10 записывается 0.

△

7.19.4 МАСШТАБИРОВАНИЕ ЗНАЧЕНИЙ (SCL)

				SCL		FNC 259				
Масштабирование значений										
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	●
Операнды	S1+	S2+	D+	Имп. инструкция(P)		Обработка		Шаги программы		
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), K, H	D, R	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)	●		16бита	32бита	SCL SCLP	7	
						●	●	DSCL DSCLP	13	

Функция

Значения масштабируются по таблице, в которой сохранены пары координат.

Описание для 16-битной обработки (SCL, SCLP)

- Входное значение в (S1+) масштабируется на основе заданной характеристики преобразования и сохраняется в (D+).

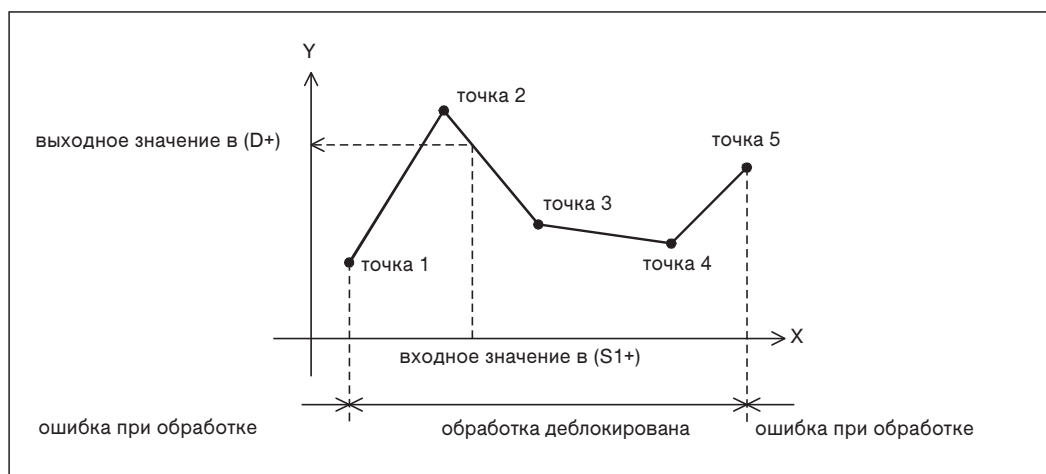


Рис. 7-273:Если входное значение находится вне кривой, заданной таблицей, возникает ошибка.

- Характеристика преобразования заложена в таблице, первый адрес которой указывается в (S2+).

Значение		Операнд
Количество точек		(S2+)
точка 1	Координаты X	(S2+)+1
	Координаты Y	(S2+)+2
точка 2	Координаты X	(S2+)+3
	Координаты Y	(S2+)+4
точка 3	Координаты X	(S2+)+5
	Координаты Y	(S2+)+6
⋮	⋮	⋮
точка n	Координаты X	(S2+)+(2n-1)
	Координаты Y	(S2+)+2n

Табл. 7-274: Заполнение таблицы точками кривой при 16-битной обработке

- Если рассчитанное исходное значение имеет дробную часть, значение округляется. Принцип округления зависит от значения первого десятичного разряда после запятой.

Описание для 32-битной обработки (DSCL, DSCLP)

- Входное значение в ((S1+)+1) и ((S1+)+0) масштабируется на основе заданной характеристики преобразования и сохраняется в ((D+)+1) и ((D+)+0).

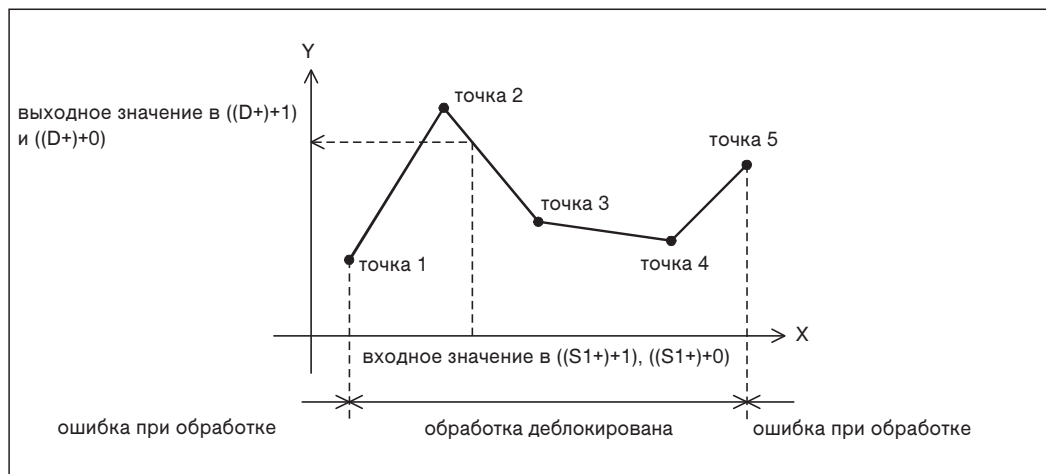


Рис. 7-275: Если входное значение находится вне формы кривой, заданной таблицей, возникает ошибка.

- Характеристика преобразования заложена в таблице, первый адрес которой указывается в (S2+).

значение	операнд	
Количество точек	(S2+)+1, (S2+)	
точка 1	Координаты X	(S2+)+3, (S2+)+2
	Координаты Y	(S2+)+5, (S2+)+4
точка 2	Координаты X	(S2+)+7, (S2+)+6
	Координаты Y	(S2+)+9, (S2+)+8
точка 3	Координаты X	(S2+)+11, (S2+)+10
	Координаты Y	(S2+)+13, (S2+)+12
:	:	:
точка n	Координаты X	(S2+)+(4n-1), (S2+)+(4n-2)
	Координаты Y	(S2+)+(4n+1), (S2+)+(4n)

Табл. 7-59:

Заполнение таблицы точками кривой при 32-битной обработке

- Если рассчитанное исходное значение имеет дробную часть, значение округляется. Принцип округления зависит от значения первого десятичного разряда после запятой.

УКАЗАНИЕ

Значения можно масштабировать и с помощью команды SCL2 (7.19.7). Эта команда отличается от команды SCL иной структурой таблицы с точками кривой.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- В таблице с точками кривой значения не упорядочены по возрастанию координаты X.
- Значение в (S1+) или ((S1+)+1) и ((S1+)+0) находится вне табличных значений.
- Расстояние между двумя следующими друг за другом точками больше 65535.

Пример задания точек кривой

На рисунке ниже в качестве примера 16-битной обработки изображена кривая с координатами X и Y ее точек.

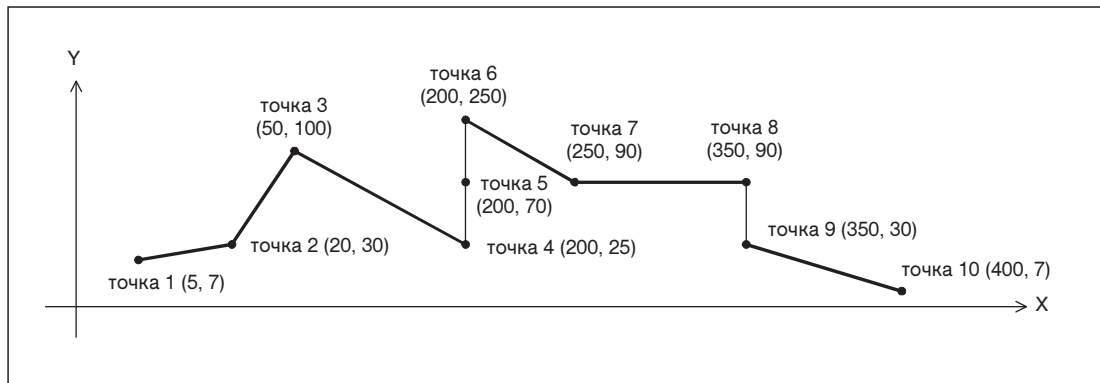


Рис. 7-276:Пример конфигурации кривой

Так как для точек 4, 5 и 6 указана одна и та же координата X (200), выводится промежуточное значение. Если координата X одинакова у трех или более точек, выводится координата Y второй точки. В этом примере исходное (промежуточное) значение определяется координатой Y точки 5.

Если одну и ту же координату X имеют две точки (как в этом примере точки 8 и 9), определяется выходное значение второй точки (в этом примере точки 9).

Значение	Операнд	Адрес операнда	Содержимое
Количество точек	(S2+)	D0	10
точка 1	Координаты X (S2+)+1	D1	5
	Координаты Y (S2+)+2	D2	7
точка 2	Координаты X (S2+)+3	D3	20
	Координаты Y (S2+)+4	D4	30
точка 3	Координаты X (S2+)+5	D5	50
	Координаты Y (S2+)+6	D6	100
точка 4	Координаты X (S2+)+7	D7	200
	Координаты Y (S2+)+8	D8	25
точка 5	Координаты X (S2+)+9	D9	200
	Координаты Y (S2+)+10	D10	70
точка 6	Координаты X (S2+)+11	D11	200
	Координаты Y (S2+)+12	D12	250
точка 7	Координаты X (S2+)+13	D13	250
	Координаты Y (S2+)+14	D14	90
точка 8	Координаты X (S2+)+15	D15	350
	Координаты Y (S2+)+16	D16	90
точка 9	Координаты X (S2+)+17	D17	350
	Координаты Y (S2+)+18	D18	30
точка 10	Координаты X (S2+)+19	D19	400
	Координаты Y (S2+)+20	D20	7

Табл.7-60:Содержание таблицы координат для этого примера (столбец “Адрес операнда” указывает операнды для случая, если в (S2+) был указан адрес D0.)

△

ПРИМЕР ▾

В следующем примере программы содержимое D0 масштабируется на основе таблицы, хранящейся начиная с R0. Выходное значение сохраняется в D10.

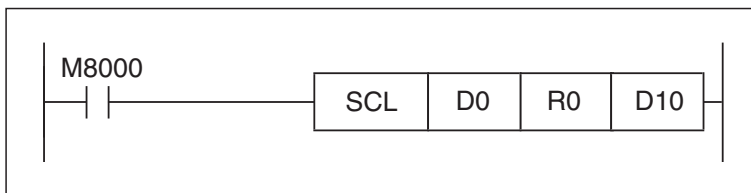


Рис. 7-277:
Пример программы для команды SCL

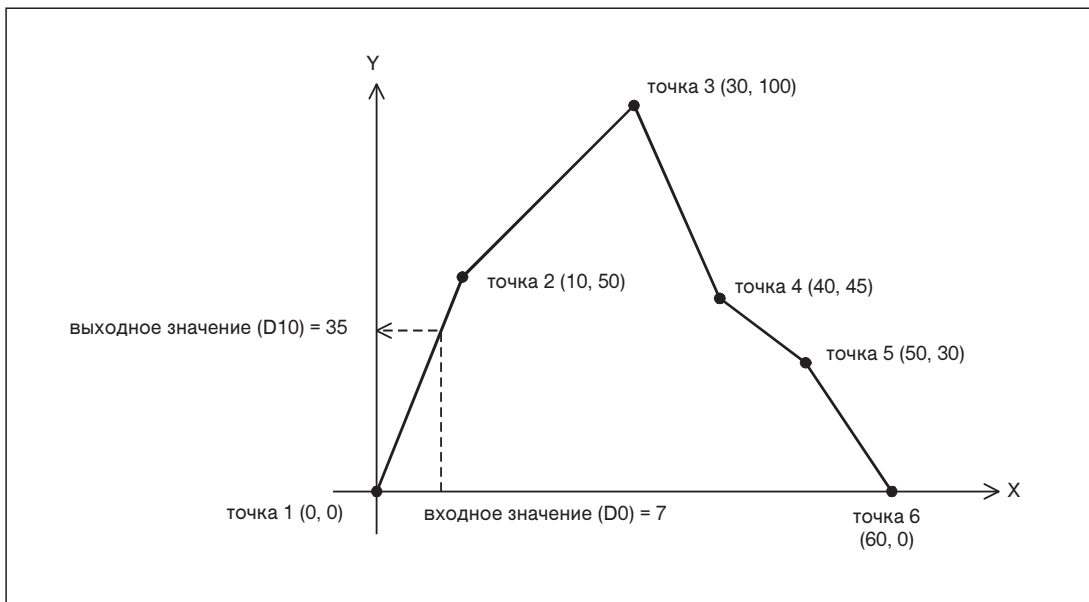
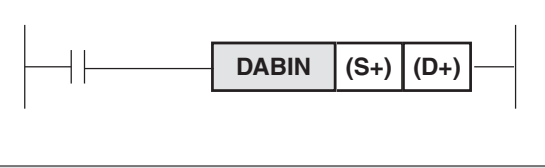


Рис. 7-278:Кривая для масштабирования входного значения в этом примере

Значение	Операнд	Адрес операнда	Содержимое	
Количество точек	(S2+)	R0	6	
точка 1	Координаты X	(S2+)+1	R1	0
	Координаты Y	(S2+)+2	R2	0
точка 2	Координаты X	(S2+)+3	R3	10
	Координаты Y	(S2+)+4	R4	50
точка 3	Координаты X	(S2+)+5	R5	30
	Координаты Y	(S2+)+6	R6	100
точка 4	Координаты X	(S2+)+7	R7	40
	Координаты Y	(S2+)+8	R8	45
точка 5	Координаты X	(S2+)+9	R9	50
	Координаты Y	(S2+)+10	R10	30
точка 6	Координаты X	(S2+)+11	R11	60
	Координаты Y	(S2+)+12	R12	0

Табл. 7-61:Таблица координат для этого примера

7.19.5 ПРЕОБРАЗОВАНИЕ ЧИСЛА В КОДИРОВКЕ ASCII В ДВОИЧНОЕ ЗНАЧЕНИЕ (DABIN)

			DABIN		FNC 260			
			Преобразование числа в кодировке ASCII в двоичное значение					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D, R	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z		●	16бита	32бита	DABIN DABINP	5
				●	●	DDABIN DDABINP	9	

Функция

Преобразование десятичных данных в кодировке ASCII в 16-битные и 32-битные двоичные данные

Описание для 16-битной обработки (DABIN, DABINP)

- Команда DABIN конвертирует десятичное число, указанное в регистрах с ((S+)+0) по ((S+)+2) в кодировке ASCII, в двоичный 16-битный формат и сохраняет его в (D+).
- Каждый сохраненный разряд числа в кодировке ASCII может принимать значение между “30H” и “39H”.
- Значение в кодировке ASCII, указанное в регистрах с ((S+)+0) по ((S+)+2), может находиться в диапазоне между -32768 и 32767.
- Если в качестве арифметического знака в младшем байте от ((S+)+0) записан ASCII-код “20H” (знак пробела), двоичное значение является положительным. Если в качестве знака пробела используется код ASCII “2DH” (знак минус), сохраняется отрицательное двоичное значение.
- Если один из разрядов преобразуемого числа содержит значение “20H” (знак пробела) или “00H” (NUL), это значение автоматически интерпретируется как “30H” (“0”).

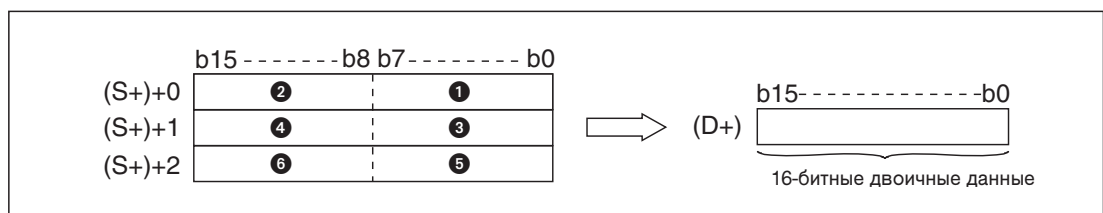


Рис. 7-279: Команда DABIN преобразует число, имеющее до 5 разрядов, в двоичные данные.

- 1 ASCII-код арифметического знака
- 2 ASCII-код разряда десятков тысяч
- 3 ASCII-код разряда тысяч
- 4 ASCII-код разряда сотен
- 5 ASCII-код разряда десятков
- 6 ASCII-код разряда единиц

ПРИМЕР ▾

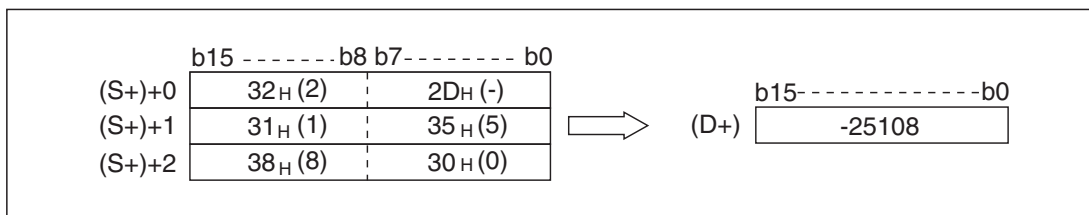


Рис. 7-280: Число “-25108” в кодировке ASCII преобразуется с помощью команды DABIN.



Описание для 32-битной обработки (DDABIN, DDABINP)

- Команда DDABIN конвертирует десятичное число в кодировке ASCII, указанное в регистрах с ((S+)+0) по ((S+)+5), в двоичный 32-битный формат и сохраняет его по адресу ((D+)+1) и ((D+)+0).
- Каждый сохраненный разряд числа в кодировке ASCII может принимать значение между “30H” и “39H”.
- Значение в кодировке ASCII, указанное в регистрах с ((S+)+0) по ((S+)+5), может находиться в диапазоне между -2147483648 и 2147483647. Содержимое старшего байта ((S+)+5) игнорируется.
- Если в качестве арифметического знака в младшем байте от ((S+)+0) записан ASCII-код “20H” (знак пробела), двоичное значение является положительным. Если в качестве знака пробела используется код ASCII “2DH” (знак минус), сохраняется отрицательное двоичное значение.
- Если один из разрядов преобразуемого числа содержит значение “20H” (знак пробела) или “00H” (NUL), это значение автоматически интерпретируется как “30H” (“0”).

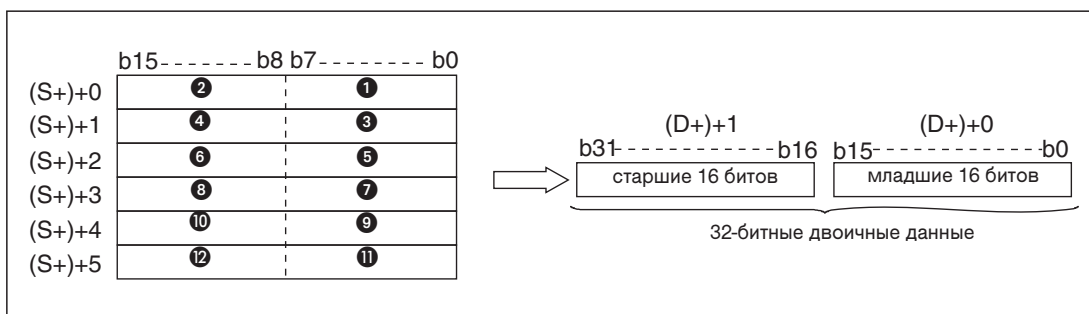


Рис. 7-281: Команда DDABIN преобразует число, содержащее до 10 разрядов, в двоичные данные.

- ① ASCII-код арифметического знака
- ② ASCII-код разряда миллиардов
- ③ ASCII-код разряда сотен миллионов
- ④ ASCII-код разряда десятков миллионов
- ⑤ ASCII-код разряда миллионов
- ⑥ ASCII-код разряда сотен тысяч
- ⑦ ASCII-код разряда десятков тысяч
- ⑧ ASCII-код разряда тысяч
- ⑨ ASCII-код разряда сотен
- ⑩ ASCII-код разряда десятков
- ⑪ ASCII-код разряда единиц
- ⑫ Содержимое этого байта игнорируется.

ПРИМЕР ▾

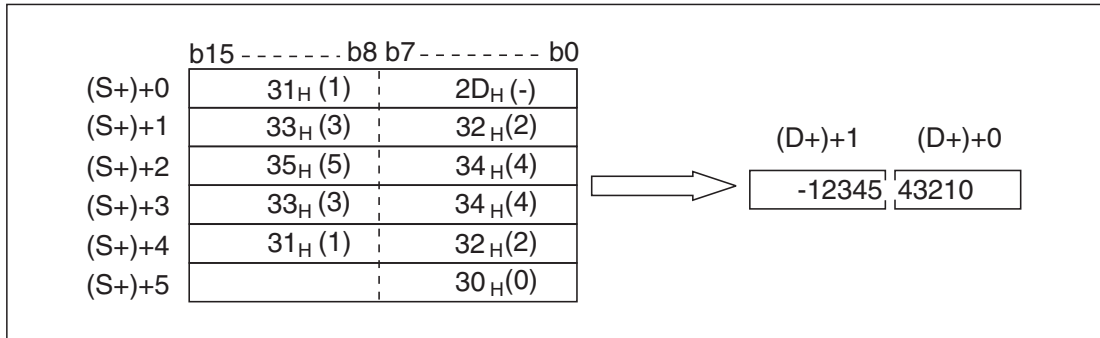


Рис. 7-282: Число "-1234543210" в кодировке ASCII преобразуется с помощью команды DDABIN.



Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- Арифметический знак в нижних 8 битах операнда ((S+)+0) содержит иное значение кроме "20_H" или "2D_H".
- ASCII-код знаков, содержащихся в регистрах с ((S+)+0) по ((S+)+2) или с ((S+)+0) по ((S+)+5), не является одним из кодов с "30_H" по "39_H,"20_H" или "00_H".
- Число начиная с (S+) находится вне следующих диапазонов:
 - команда DABIN, DABINP: от -32768 до 32767
 - команда DDABIN, DDABINP: от -2147483648 до 2147483647
- При указании (S+) превышает допустимая область для операнда этого типа. (В случае 16-битной версии после (S+) занимают еще два операнда, а в случае 32-битной версии - еще 5 операндов.)

ПРИМЕР ▾

Следующая программа при включении X0 преобразует пятизначное десятичное число, сохраненное в кодировке ASCII в регистрах, начиная с D20, в двоичное значение и сохраняет его в D0.

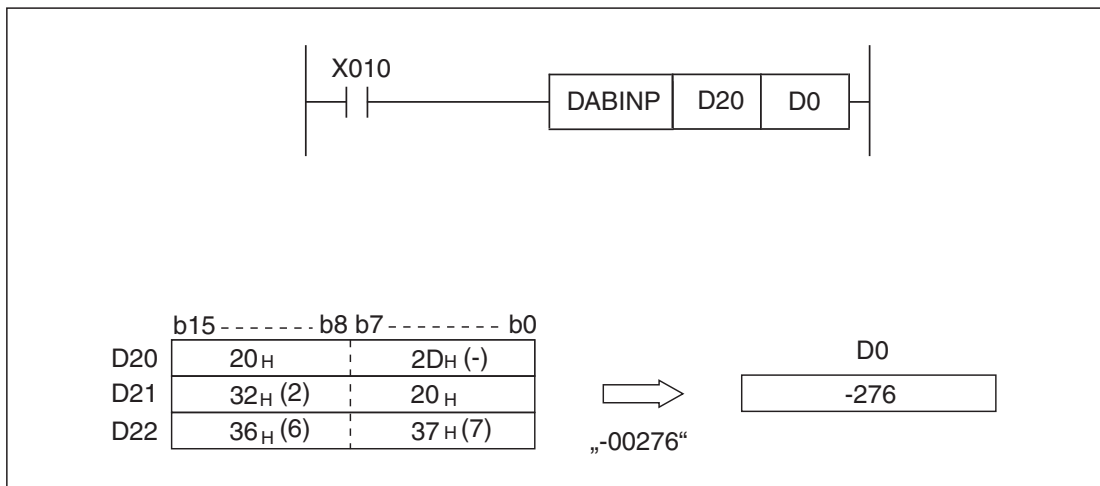


Рис. 7-283: Значение "-276" интерпретируется как "-00276" и сохраняется в виде "-276".



7.19.6 ПРЕОБРАЗОВАНИЕ ДВОИЧНОГО ЗНАЧЕНИЯ В КОД ASCII (BINDA)

		BINDA		FNC 261			
		Преобразование двоичного значения в код ASCII					
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
							●
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), V, Z	T, C, D, R		●	16бита	32бита	BINDA BINDAP
				●	●	DBINDA DBINDAP	9

Функция

Преобразование 16-битных и 32-битных двоичных данных в десятичные числа в кодировке ASCII

Описание для 16-битной обработки (BINDA, BINDAP)

- Команда BINDA конвертирует указанное в (S+) 16-битовое двоичное число в десятичное число в кодировке ASCII и сохраняет результат в (D+).
- Указанное в (S+) 16-битовое двоичное число может находиться в диапазоне между -32768 и 32767.
- Если 16-битовое двоичное число положительное, то в младшем байте ((D+)+0) в качестве арифметического знака сохраняется ASCII-код "20H" (знак пробела). В случае отрицательного двоичного числа в качестве арифметического знака здесь сохраняется ASCII-код "2DH" (знак минус).
- Если двоичные числа начинаются с нулей, то в преобразованном числе эти нули заменены знаком пробела ("20H"). Например, в числе "00325" нули в разрядах десятков тысяч и тысяч заменяются кодами "20H".
- Сохранение данных в ((D+)+3) зависит от состояния специального маркера M8091.
 - Если M8091 не установлен, в ((D+)+3) вводится ASCII-код "00H" (NUL).
 - Если M8091 установлен, содержимое ((D+)+3) не изменяется.

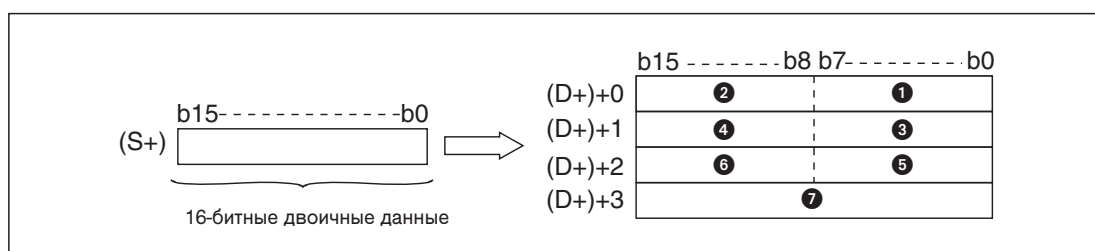


Рис. 7-284: Команда BINDA преобразует двоичные данные в число с количеством разрядов до 5.

- 1 ASCII-код арифметического знака
- 2 ASCII-код разряда десятков тысяч
- 3 ASCII-код разряда тысяч
- 4 ASCII-код разряда сотен
- 5 ASCII-код разряда десятков
- 6 ASCII-код разряда единиц
- 7 "00H" (M8091 = 0) или не измененное содержимое (M8091 = 1)

ПРИМЕР ▾

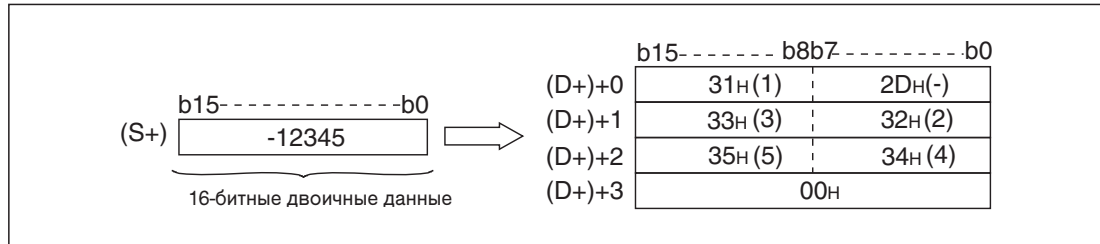


Рис. 7-285: Число “-12345” преобразуется с помощью команды BINDA. Специальный маркер M8091 в этом примере не установлен.

△

Описание для 32-битной обработки (DBINDA, DBINDAP)

- Команда BINDA конвертирует указанное в ((S+)+1) и ((S+)+0) 32-битное двоичное число в десятичное число в кодировке ASCII и сохраняет результат в (D+).
- Указанное в ((S+)+1) и ((S+)+0) 32-битное двоичное число может находиться в диапазоне между -2147483648 и 2147483647.
- Если 16-битное двоичное число положительное, то в младшем байте ((D+)+0) в качестве арифметического знака сохраняется ASCII-код “20H” (знак пробела). В случае отрицательного двоичного числа в качестве арифметического знака здесь сохраняется ASCII-код “2DH” (знак минус).
- Если двоичные числа начинаются с нулей, то в преобразованном числе эти нули заменены знаком пробела (“20H”). Например, в числе “0012034560” нули в разрядах миллиардов и сотен миллионов заменяются кодами “20H”.
- Сохранение данных в ((D+)+5) зависит от состояния специального маркера M8091.
 - Если M8091 не установлен, в старший байт ((D+)+5) вводится ASCII-код “00H” (NUL).
 - Если M8091 установлен, первоначальное содержимое старшего байта ((D+)+5) не изменяется.

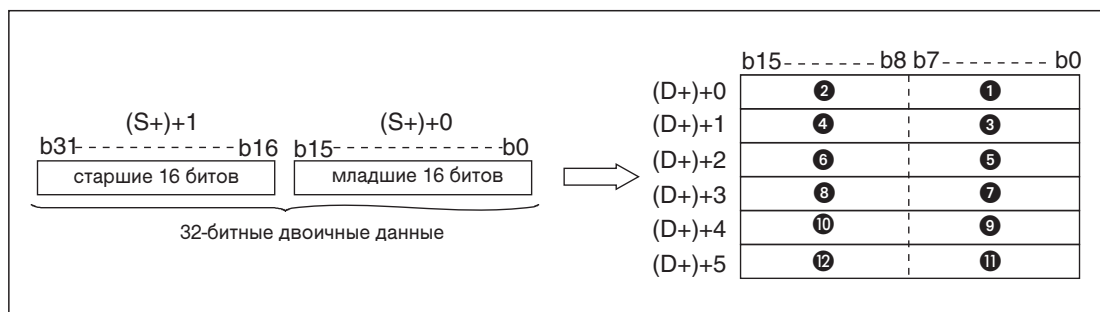


Рис. 7-286: Команда DBINDA преобразует двоичные данные в число с количеством разрядов до 10.

- 1 ASCII-код арифметического знака
- 2 ASCII-код разряда миллиардов
- 3 ASCII-код разряда сотен миллионов
- 4 ASCII-код разряда десятков миллионов
- 5 ASCII-код разряда миллионов
- 6 ASCII-код разряда сотен тысяч
- 7 ASCII-код разряда десятков тысяч
- 8 ASCII-код разряда тысяч

- ⑨ ASCII-код разряда сотен
- ⑩ ASCII-код разряда десятков
- ⑪ ASCII-код разряда единиц
- ⑫ "00H" (M8091 = 0) или не измененное содержимое (M8091 = 1)

ПРИМЕР ▾

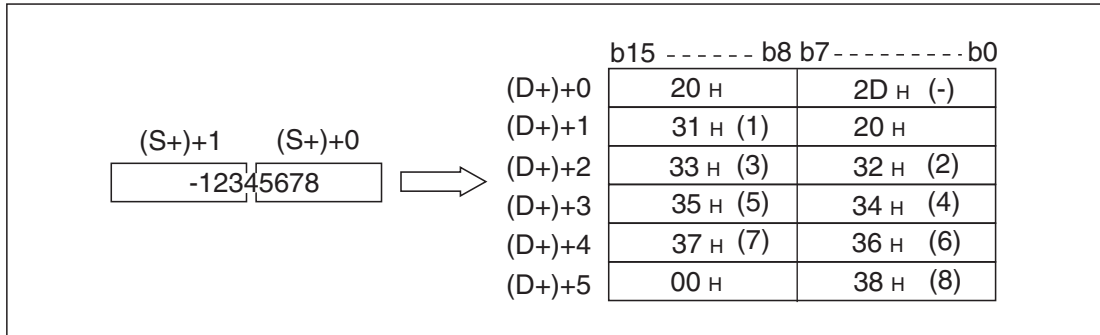


Рис. 7-287: Число "-12345678" преобразуется с помощью команды DBINDA в ASCII-код. Специальный маркер M8091 в этом примере не установлен.



Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- При указании (D+) превышает допустимая область для операнда этого типа. (В случае 16-битной версии после (D+) занимают еще два или три операнда, а в случае 32-битной версии - еще 5 операндов.)

ПРИМЕР ▾

Следующая программа при включении X0 преобразует в хранящееся D1000 двоичное значение в формат ASCII и сохраняет результат в D0.

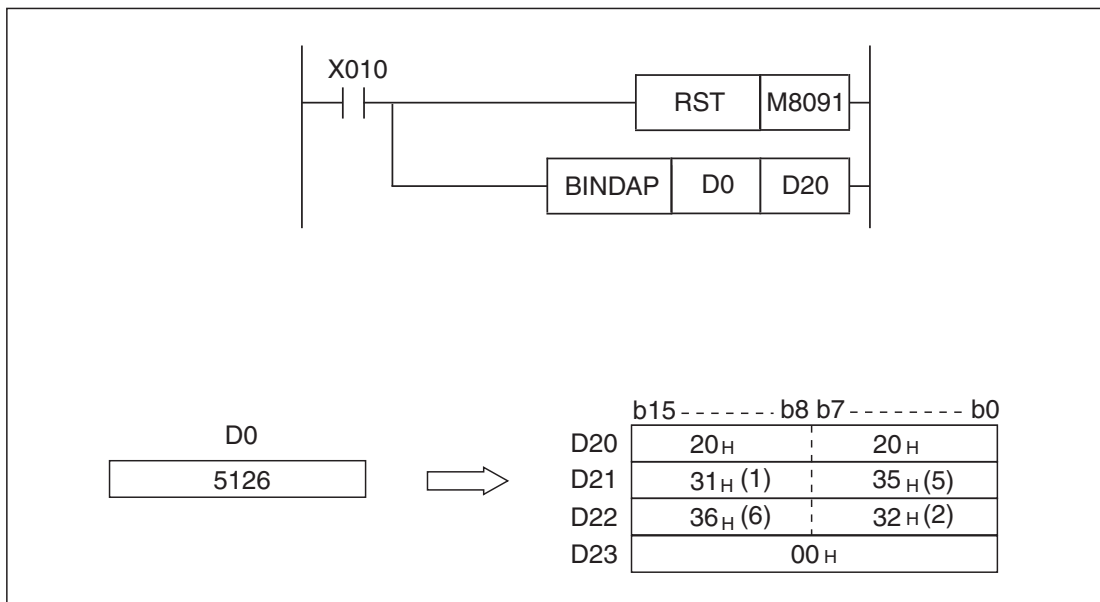


Рис. 7-288: Перед преобразованием сбрасывается маркер M8091, чтобы в D23 было записано значение "00H".

7.19.7 МАСШТАБИРОВАНИЕ ЗНАЧЕНИЙ (SCL2)

				SCL2		FNC 269					
				Масштабирование значений							
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U		
										●	
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы				
	KnX, KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□), K, H	D, R	KnY, KnM, KnS, T, C, D, R, специальные модули (U□/G□)		●	16бита	32бита	SCL2	SCL2P	7	
						●	●	DSCL2	DSCL2P	13	

Функция

Масштабирование значений по таблице, отсортированной по точкам кривой.

Описание для 16-битной обработки (SCL2, SCL2P)

- Входное значение в (S1+) масштабируется на основе заданной характеристики преобразования и сохраняется в (D+).

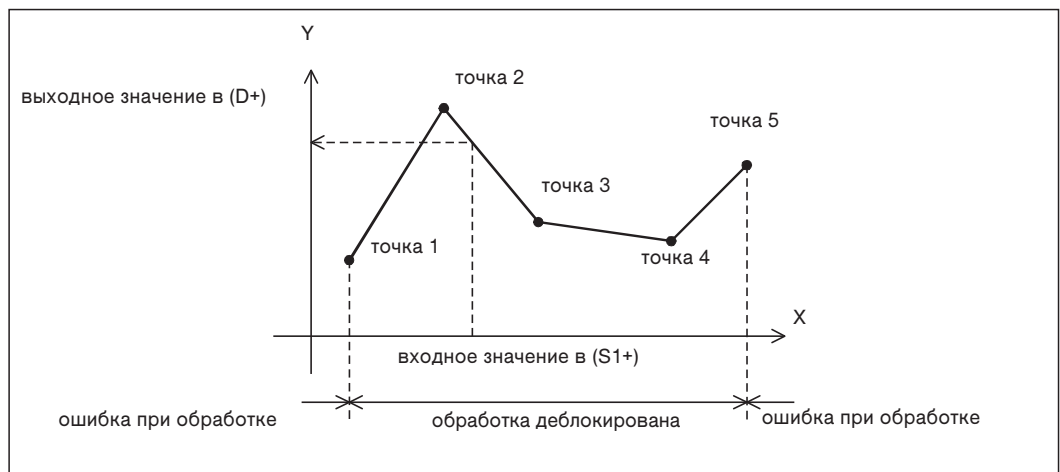


Рис. 7-289:Если входное значение находится вне формы кривой, заданной таблицей, возникает ошибка.

- Характеристика преобразования заложена в таблице, первый адрес которой указывается в (S2+).

Значение	Операнд
Количество точек	(S2+)
Координаты X	точка 1 (S2+)+1
	точка 2 (S2+)+2
	:
	точка n (S2+)+n
Координаты Y	точка 1 (S2+)+(n+1)
	точка 2 (S2+)+(n+2)
	:
	точка n (S2+)+(2n)

Табл. 7-62:

Заполнение таблицы точками кривой при 16-битной обработке

- Если рассчитанное исходное значение имеет дробную часть, значение округляется. Принцип округления зависит от значения первого десятичного разряда после запятой.

Описание для 32-битной обработки (DSCL2, DSCL2P)

- Входное значение в ((S1+)+1) и ((S1+)+0) масштабируется на основе заданной характеристики преобразования и сохраняется в ((D+)+1) и ((D+)+0).

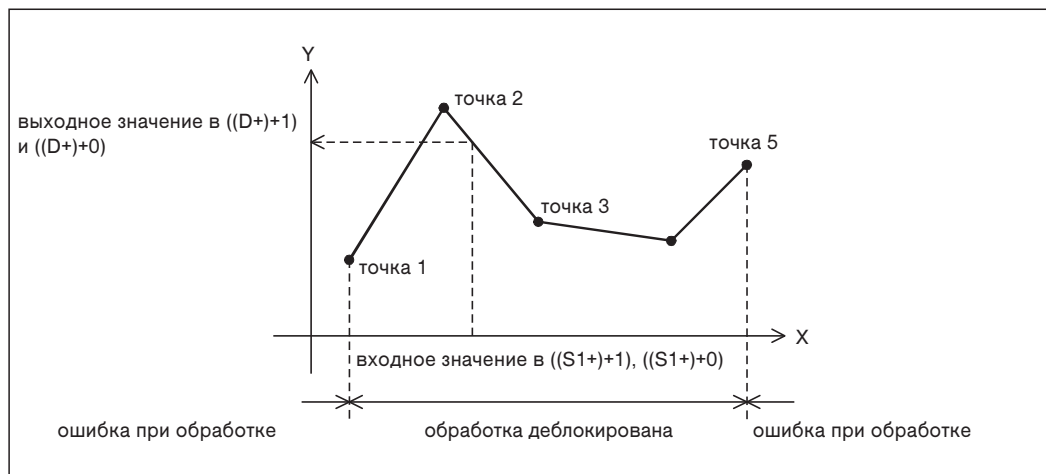


Рис. 7-290: Если входное значение находится вне формы кривой, заданной таблицей, возникает ошибка.

- Характеристика преобразования заложена в таблице, первый адрес которой указывается в (S2+).

Значение		Операнд
Количество точек		(S2+)+1, (S2+)
Координаты X	точка 1	(S2+)+3, (S2+)+2
	точка 2	(S2+)+5, (S2+)+4
	:	:
	точка n	(S2+)+(2n+1), (S2+)+(2n)
Координаты Y	точка 1	(S2+)+(2n+3), (S2+)+(2n+2)
	точка 2	(S2+)+(2n+5), (S2+)+(2n+4)
	:	:
	точка n	(S2+)+(4n+1), (S2+)+(4n)

Табл. 7-63:

Заполнение таблицы точками кривой при 32-битной обработке

- Если рассчитанное исходное значение имеет дробную часть, значение округляется. Принцип округления зависит от значения первого десятичного разряда после запятой.

УКАЗАНИЕ

Значения можно также масштабировать с помощью команды SCL (7.19.4). Она отличается от команды SCL2 иной структурой таблицы с точками кривой.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки "6706":

- В таблице с точками кривой значения не упорядочены по возрастанию координаты X.
- Значение в (S1+) или ((S1+)+1) и ((S1+)+0) находится вне табличных значений.
- Расстояние между двумя следующими друг за другом точками больше 65535.

Пример задания точек кривой

На рисунке ниже в качестве примера 16-битной обработки изображена кривая с координатами X и Y ее точек.

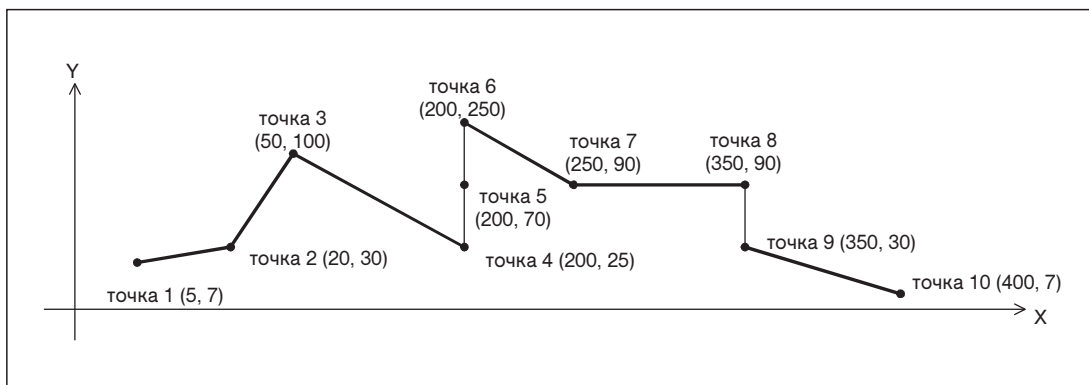


Рис. 7-291:Пример конфигурации кривой

Так как для точек 4, 5 и 6 указана одна и та же координата X (200), выводится промежуточное значение. Если координата X одинакова у трех или более точек, выводится координата Y второй точки. В этом примере исходное (промежуточное) значение определяется координатой Y точки 5.

Если одну и ту же координату X имеют две точки (как в этом примере точки 8 и 9), определяется выходное значение второй точки (в этом примере точки 9).

Значение	Операнд	Адрес операнда	Содержимое	
Количество точек	(S2+)	D0	10	
Координаты X	точка 1	(S2+)+1	D1	5
	точка 2	(S2+)+2	D2	20
	точка 3	(S2+)+3	D3	50
	точка 4	(S2+)+4	D4	200
	точка 5	(S2+)+5	D5	200
	точка 6	(S2+)+6	D6	200
	точка 7	(S2+)+7	D7	250
	точка 8	(S2+)+8	D8	350
	точка 9	(S2+)+9	D9	350
	точка 10	(S2+)+10	D10	400
Координаты Y	точка 1	(S2+)+11	D11	7
	точка 2	(S2+)+12	D12	30
	точка 3	(S2+)+13	D13	100
	точка 4	(S2+)+14	D14	25
	точка 5	(S2+)+15	D15	70
	точка 6	(S2+)+16	D16	250
	точка 7	(S2+)+17	D17	90
	точка 8	(S2+)+18	D18	90
	точка 9	(S2+)+19	D19	30
	точка 10	(S2+)+20	D20	7

Табл. 7-64:Содержание таблицы координат для этого примера (столбец “Адрес операнда” указывает операнды для случая, если в (S2+) указан адрес D0)

△

ПРИМЕР ▾

В следующем примере программы содержимое D0 масштабируется на основе таблицы, хранящейся начиная с R0. Выходное значение сохраняется в D10.

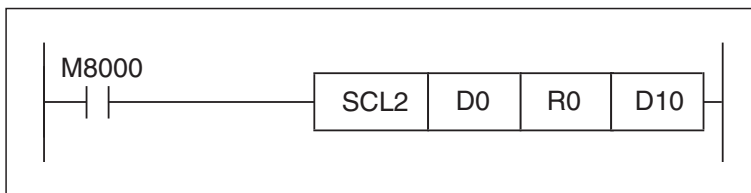


Рис. 7-292:
Пример программы для команды SCL

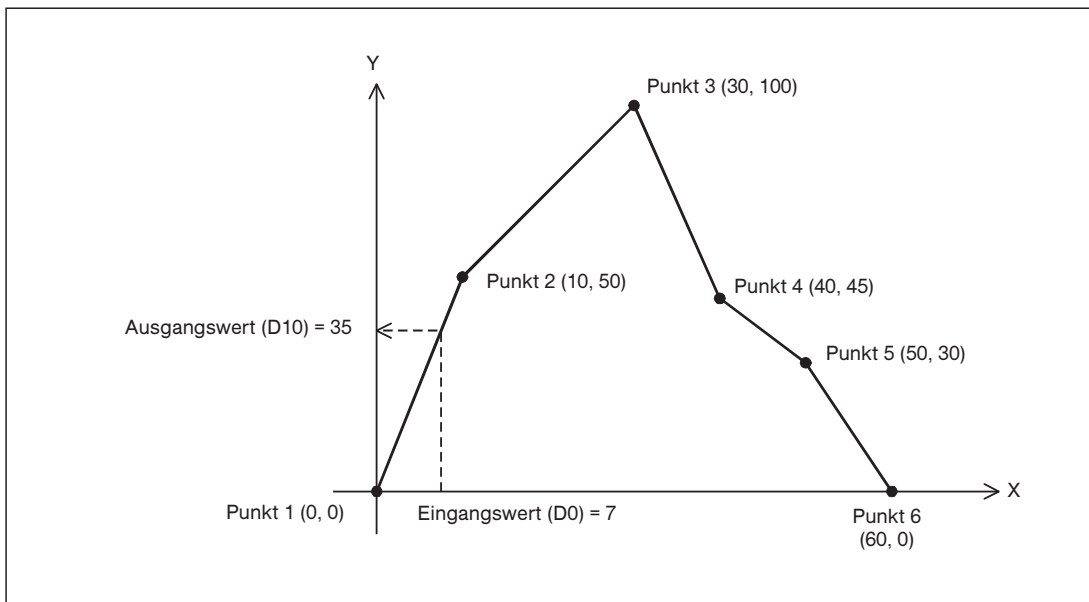


Рис. 7-293:Кривая для масштабирования входного значения в этом примере

ЗНАЧЕНИЯ	ОПЕРАНД	АДРЕС ОПЕРАНДА	ЗНАЧЕНИЕ	
ЧИСЛО ТОЧЕК	(S2+)	D0	6	
X-координата	точка 1	(S2+)+1	D1	0
	точка 2	(S2+)+2	D2	10
	точка 3	(S2+)+3	D3	30
	точка 4	(S2+)+4	D4	40
	точка 5	(S2+)+5	D5	50
	точка 6	(S2+)+6	D6	60
Y-координата	точка 1	(S2+)+11	D11	0
	точка 2	(S2+)+12	D12	50
	точка 3	(S2+)+13	D13	100
	точка 4	(S2+)+14	D14	45
	точка 5	(S2+)+15	D15	30
	точка 6	(S2+)+16	D16	0

Табл. 7-65:Таблица координат для этого примера

7.20 ОБМЕН ДАННЫМИ С ПРЕОБРАЗОВАТЕЛЯМИ ЧАСТОТЫ

Описываемые в этом разделе команды позволяют обмениваться данными с преобразователями частоты серий F700, A700, V500, F500, A500, E500 и S500.

Символ	FNC	Назначение	Разд.
IVCK	270	Проверка состояния преобразователя частоты	7.20.1
IVDR	271	Управление преобразователем частоты	7.20.2
IVRD	272	Считывание параметров преобразователя частоты	7.20.3
IVWR	273	Запись параметров в преобразователь частоты	7.20.4
IVBWR	274	Поблочная запись параметров в преобразователь частоты	7.20.5

Табл. 7-66: Обзор команд для обмена данными с преобразователями частоты

УКАЗАНИЕ

Команды IVCK, IVDR, IVRD, IVWR и IVBWR нельзя выполнять одновременно с командой RS или RS2 для того же интерфейса.

Две команды для коммуникации с преобразователями частоты (FNC270...FNC274) можно выполнять одновременно для одного и того же интерфейса.

При коммуникации с преобразователями частоты могут опрашиваться следующие специальные маркеры и регистры:

Специальный маркер		Значение
Интерфейс, кан. 1	Интерфейс, кан. 2	
M8029		Выполнение команды окончено
M8063	M8438	Ошибка при последовательной коммуникации
M8151	M8156	Происходит обмен данными с преобразователем частоты ^①
M8152	M8157	Ошибка при обмене данными с преобразователем частоты ^①
M8153	M8158	Сохраненная ошибка при обмене данными с преобразователем частоты ^①
M8154	M8159	Ошибка при выполнении команды IVBWR ^①

Табл. 7-67: Специальные маркеры, относящиеся к соединению с преобразователями частоты

^① При изменении режима контроллера со „STOP“ на „RUN“ эти маркеры сбрасываются.

Специальный маркер		Значение
Интерфейс, кан. 1	Интерфейс, кан. 2	
D8063	D8438	Код ошибки при сбое последовательной коммуникации
D8150	D8155	Время ожидания для реакции преобразователя частоты
D8151	D8156	Номер шага при обмене данными с преобразователем частоты ^②
D8152	D8157	Код ошибки при сбое обмена данными с преобразователем частоты ^①
D8153	D8158	Сохраненный код ошибки при сбое обмена данными с преобразователем частоты ^②
D8154	D8159	Номер параметра при сбое выполнения команды IVBWR ^②

Табл. 7-68: Специальные регистры, связанные с обменом данными с преобразователями частоты

^① При изменении режима контроллера со „STOP“ на „RUN“ эти регистры стираются.

^② После запуска контроллера эти регистры заполняются значением „-1“.

7.20.1 ПРОВЕРКА СОСТОЯНИЯ ПРЕОБРАЗОВАТЕЛЯ ЧАСТОТЫ (IVCK)

					IVCK		FNC 270			
Проверка состояния преобразователя частоты										
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
										●
Операнды	S1+	S2+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U/G), K, H		KnY, KnM, KnS, D, R, специальные модули (U/Gm), K, H	K, H		16бита	32бита	IVCK		9
						●				

Функция

Считывание состояния преобразователя частоты (функция мониторинга)

Описание

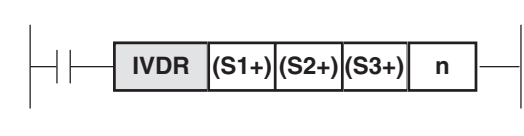
- С помощью команды IVCK можно проверить состояние преобразователя частоты.
- Функция команды IVCK соответствует функции команды EXTR K10 в контроллерах серии FX2N или FX2NC (раздел 7.13.1).
- В (S1+) указывается номер станции преобразователя частоты (от 0 до 31).
- (S2+) содержит код команды для преобразователя частоты (см. следующую таблицу)

Код команды	Значение данных, переданных преобразователем	Серии преобразователей, поддерживающих этот код команды						
		F700	A700	V500	F500	A500	E500	S500
H7B	режим	●	●	●	●	●	●	●
H6F	выходная частота	●	●	●	●	●	●	●
H70	выходной ток	●	●	●	●	●	●	●
H71	выходное напряжение	●	●	●	●	●	●	
H72	специальный контроль	●	●	●	●	●		
H73	выбранный номер для специального контроля	●	●	●	●	●		
H74	определение сигнализации	●	●	●	●	●	●	●
H75	определение сигнализации	●	●	●	●	●	●	●
H76	определение сигнализации	●	●	●	●	●	●	
H77	определение сигнализации	●	●	●	●	●	●	
H79	состояние преобразователя (расширенное)	●	●					
H7A	состояние преобразователя	●	●	●	●	●	●	●
H6E	настроенная выходная частота (EEPROM)	●	●	●	●	●	●	●
H6D	настроенная выходная частота (RAM)	●	●	●	●	●	●	●

Табл. 7-69: Код команды в (S2+) для команды IVCK

- В (D+) указывается операнд, в котором должны быть сохранены данные, переданные преобразователем частоты.
- В (n) указывается номер используемого интерфейса (1: кан. 1, 2: кан. 2).

7.20.2 УПРАВЛЕНИЕ ПРЕОБРАЗОВАТЕЛЕМ ЧАСТОТЫ (IVDR)

					IVDR		FNC 271			
					Управление преобразователем частоты					CPU
Операнды	S1+	S2+	S3+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U/Gm), K, H		KnX, KnY, KnM, KnS, D, R, специальные модули (U/Gm)			K, H	16бита	32бита	IVDR	

Функция

Управление преобразователем частоты

Описание

- С помощью команды IVDR можно управлять преобразователем частоты.
- Функция команды IVDR соответствует функции команды EXTR K11 в контроллерах серии FX2N или FX2NC (раздел 7.13.1).
- В (S1+) указывается номер станции преобразователя частоты (от 0 до 31).
- В (S2+) указывается код команды для преобразователя частоты (см. следующую таблицу)

Код команды	Значение данных, переданных на преобразователь	Серии преобразователей, поддерживающих этот код команды						
		F700	A700	V500	F500	A500	E500	S500
HFB	режим	●	●	●	●	●	●	●
HF3	выбранный номер для специального контроля	●	●	●	●	●		
HF9	сигнал работы (расширенный)	●	●					
HFA	сигнал работы	●	●	●	●	●	●	●
HEE	выходная частота (запись в EEPROM)	●	●	●	●	●	●	●
HED	выходная частота (запись в RAM)	●	●	●	●	●	●	●
HFD	выполнить сброс преобразователя	●	●	●	●	●	●	●
HF4	стереть перечень сигнализации	●	●		●	●	●	●
HFC	стереть все параметры	●	●	●	●	●	●	●
HFC	стирание, определенное пользователем	●	●		●	●		

Табл. 7-70: Код команды в (S2+) для команды IVDR

- В (S3+) содержатся данные, передаваемые на преобразователь частоты.
- В (n) указывается номер используемого интерфейса (1: кан. 1, 2: кан. 2)

7.20.3 СЧИТЫВАНИЕ ПАРАМЕТРОВ ПРЕОБРАЗОВАТЕЛЯ ЧАСТОТЫ (IVRD)

					IVRD		FNC 272			
					Считывание параметров из преобразователя частоты					CPU
Операнды	S1+	S2+	D+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U□/G□), K, H		D, R, специальные модули (U□/G□)	K, H		16бита	32бита	IVRD	9	
					●					

Функция

Считывание параметров преобразователя частоты

Описание

- Функция команды IVRD соответствует функции команды EXTR K12 в контроллерах серии FX2N или FX2NC (раздел 7.13.1).
- В (S1+) указывается номер станции преобразователя частоты (от 0 до 31).
- В (S2+) указывается номер параметра, настройку которого требуется считать.
- В (D+) указывается операнд, в котором должен быть сохранен считанный параметр.
- В (n) указывается номер используемого интерфейса (1: кан. 1, 2: кан. 2).

7.20.4 ЗАПИСЬ ПАРАМЕТРОВ В ПРЕОБРАЗОВАТЕЛЬ ЧАСТОТЫ (IVWR)

					IVWR		FNC 273			
					Запись параметров в преобразователь частоты					CPU
Операнды	S1+	S2+	S3+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U□/G□), K, H			K, H		16бита	32бита	IVWR	9	
					●					

Функция

Изменение значения параметра в преобразователе частоты

Описание

- С помощью команды IVWR можно изменить параметры в преобразователе частоты.
- Функция команды IVRD соответствует функции команды EXTR K13 в контроллерах серии FX2N и FX2NC (раздел 7.13.1).
- В (S1+) указывается номер станции преобразователя частоты (от 0 до 31).
- В (S2+) указывается номер параметра, который требуется изменить.
- (S3+) содержит данные, передаваемые на преобразователь частоты.
- В (n) указывается номер используемого интерфейса (1: кан. 1, 2: кан. 2)

7.20.5 ПОБЛОЧНАЯ ЗАПИСЬ ПАРАМЕТРОВ В ПРЕОБРАЗОВАТЕЛЬ ЧАСТОТЫ (IVBWR)

					IVBWR		FNC 274			
Запись параметров в преобразователь частоты										
CPU		FX1S	FX1N	FX2N	FX2NC	FX3U				
						●				
Операнды	S1+	S2+	S3+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	D, R, специальные модули (U□/G□), K, H				K, H		16бита	32бита	IVBWR	9
							●			

Функция

Изменение значений нескольких параметров в преобразователе частоты

Описание

- С помощью команды IVBWR можно одновременно изменить несколько параметров в преобразователе частоты.
- В (S1+) указывается номер станции преобразователя частоты (от 0 до 31).
- (S2+) содержит количество параметров, которые требуется изменить.
- (S3+) содержит первый адрес области операндов, содержащей номера изменяемых параметров и значения этих параметров.

значение	операнд	
(S3+)+0	1-й параметр	номер параметра
(S3+)+1		заданное значение
(S3+)+2	2-й параметр	номер параметра
(S3+)+3		заданное значение
:	:	:
(S3+)+(2m-4)*	(m-1)-ый параметр	номер параметра
(S3+)+(2m-3)*		заданное значение
(S3+)+(2m-2)*	m-тый параметр	номер параметра
(S3+)+(2m-1)*		заданное значение

Табл. 7-71:Использование операндов, начиная с (S3+)

* „m“ означает количество параметров, указанное в (S2+).

- В (n) указывается номер используемого интерфейса (1: кан. 1, 2: кан. 2)

7.21 ОБМЕН ДАННЫМИ СО СПЕЦИАЛЬНЫМИ МОДУЛЯМИ

Символ	FNC	Значение	Раздел
RBFM	278	Считывание из буферной памяти специальных модулей	7.21.1
WBFM	279	Запись в буферную память специальных модулей	7.21.2

Табл. 7-72: Обзор команд для коммуникации со специальными модулями

Помимо команд FROM и TO, обмениваться данными между базовым блоком программируемого контроллера и подключенными специальными модулями можно с помощью описываемых здесь команд RBFM и WBFM. В отличие от команд FROM/TO, при команде RBFM или WBFM данные передаются за несколько программных циклов.

УКАЗАНИЕ

Адресация специальных модулей, структура буферной памяти и команды FROM и TO описаны в разделе 7.3.

7.21.1 СЧИТЫВАНИЕ ИЗ БУФЕРНОЙ ПАМЯТИ СПЕЦИАЛЬНЫХ МОДУЛЕЙ (RBFM)

					RBFM	FNC 278				
					Считывание из буферной памяти специальных модулей					
					CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
Операнды	n1	n2	D+	n3	n4	Имп. инструкция(P)	Обработка		Шаги программы	
	D, R, K, H		D*, R	D, R, K, H			16бита	32бита	RBFM	11
							●			

* В (D+) нельзя указывать специальные регистры.

Функция

Считывание данных из буферной памяти подключенных специальных модулей. Передачу данных можно распределить на несколько программных циклов.

Описание

- В (n1) указывается адрес специального модуля.
- В (n2) указывается первый адрес буферной памяти, из которого требуется считать данные.
- В (n3) указывается количество слов данных, которое требуется считать.
- В (n4) указывается, сколько слов данных требуется передать в одном программном цикле.
- Первый адрес области операндов в базовом блоке, в которой должны быть сохранены считанные данные, указывается в (D+).

- Допустимые диапазоны:
 - n1 = 0...7
 - n2 = 0...32766
 - n3 = 1...32767
 - n4 = 1...32767
- Количество программных циклов, необходимых для выполнения команды RBFM, можно рассчитать, разделив указанное в (n3) общее количество слов данных на указанное в (n4) количество данных на каждый цикл:

$$\text{количество программных циклов} = [n3] / [n4]$$
- Если при делении образуется остаток, остающиеся данные передаются в дополнительном программном цикле.

При выполнении команды RBFM в программе должны контролироваться следующие специальные маркеры. Эти маркеры должны использоваться для блокировки и вывода сообщения об ошибке:

 - M8029: этот маркер устанавливается, если выполнение команды RBFM завершено без ошибок.
 - M8328: этот маркер показывает, что команда RBFM или WBFM выполняется в другом шаге программы и при этом она обращается к этому же специальному модулю. Выполнение других команд с тем же самым адресом в буферной памяти блокируется до тех пор, пока не будет завершена обработка этой команды.
 - M8329: если устанавливается этот маркер, то это означает, что при выполнении команды RBFM возникла ошибка.

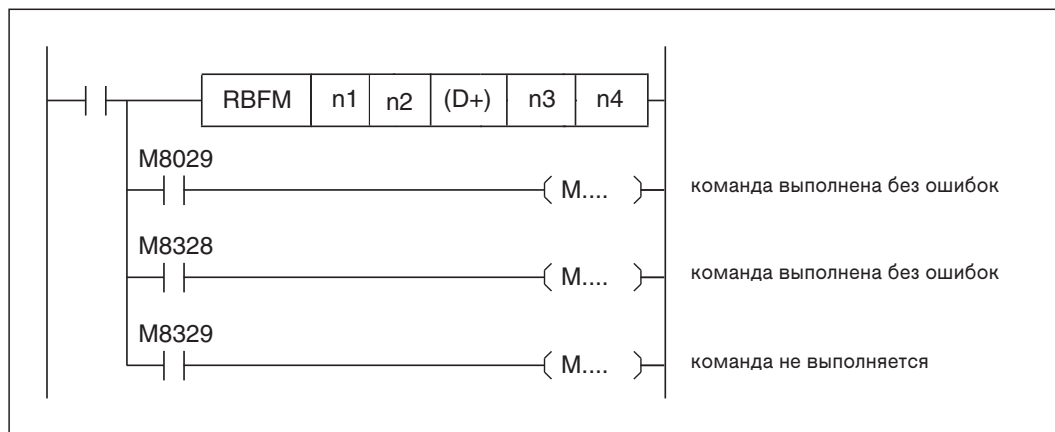


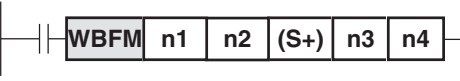
Рис. 7-294: Состояние маркеров M8029, M8328 и M8329 необходимо проверять сразу после команды RBFM.

Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки „6708“:

- Специальный модуль с адресом, указанным в (n1), не существует.

7.21.2 ЗАПИСЬ В БУФЕРНУЮ ПАМЯТЬ СПЕЦИАЛЬНЫХ МОДУЛЕЙ (WBFM)

						WBFM		FNC 279		
						Запись в буферную память специальных модулей				
CPU						FX1S	FX1N	FX2N	FX2NC	FX3U
										●
Операнды	n1	n2	S+	n3	n4	Имп. инструкция(P)	Обработка		Шаги программы	
	D, R, K, H		D*, R	D, R, K, H			16бита	32бита	WBFM	
						●				

* В (S+) нельзя указывать специальные регистры.

Функция

Передача данных в буферную память подключенных специальных модулей. Передачу данных можно распределить на несколько программных циклов.

Описание

- В (n1) указывается адрес специального модуля.
- В (n2) указывается первый адрес буферной памяти, в который требуется записать данные.
- В (n3) указывается количество слов данных, которые требуется передать.
- В (n4) указывается, сколько слов данных требуется передать в одном программном цикле.
- Первый адрес области операндов базового блока, в которой хранятся данные, указывается в (S+).
- Допустимые диапазоны:
n1 = 0...7
n2 = 0...32766
n3 = 1...32767
n4 = 1...32767
- Количество программных циклов, необходимых для выполнения команды WBFM, можно рассчитать, разделив указанное в (n3) общее количество слов данных на указанное в (n4) количество данных на каждый цикл:

$$\text{количество программных циклов} = [n3] / [n4]$$

Если при делении образуется остаток, остающиеся данные передаются в дополнительном программном цикле.
- При выполнении команды WBFM в программе должны контролироваться следующие специальные маркеры. Эти маркеры следует использовать для блокировки и вывода сообщения о ошибке:
 - M8029: этот маркер устанавливается, если выполнение команды WBFM завершено без ошибок.
 - M8328: этот маркер показывает, что команда RBFM или WBFM выполняется в другом шаге программы и при этом она обращается к этому же специальному модулю. Выполнение других команд с тем же самым адресом в буферной памяти блокируется до тех пор, пока не будет завершена обработка этой команды.
 - M8329: Если устанавливается этот маркер, то это означает, что при выполнении команды WBFM возникла ошибка.

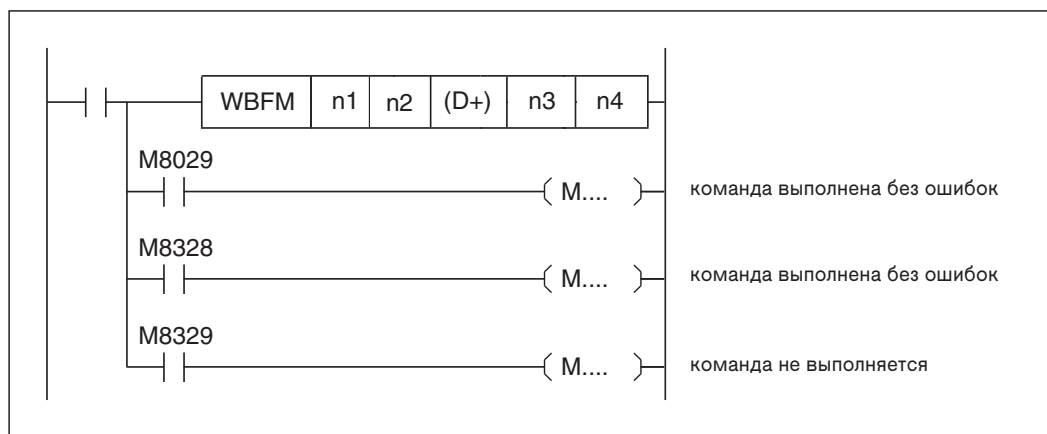


Рис. 7-295:Состояние маркеров M8029, M8328 и M8329 необходимо проверять сразу после команды WBFM.

Источники ошибок

В следующем случае возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки „6708“:

- Специальный модуль с адресом, указанным в (n1), не существует.

7.21.3

УКАЗАНИЯ ПО ПРИМЕНЕНИЮ КОМАНД RBFM И WBFM

Срабатывание контрольного таймера

Ошибка контрольного таймера может возникнуть и в том случае, если данные, передаваемые между базовым блоком программируемого контроллера и специальными модулями, распределены на несколько программных циклов. В этом случае можно изменить настройку контрольного таймера или параметры команды RBFM или WBFM.

- Изменение настройки контрольного таймера

Время контроля изменяется в результате записи нового значения в специальный регистр D8000. Это время предварительно настроено на 200 мс. Чтобы активировать новое время сразу, следует дополнительно выполнить команду WDT. Без этой команды новая настройка контрольного таймера начинает действовать лишь со следующего программного цикла.

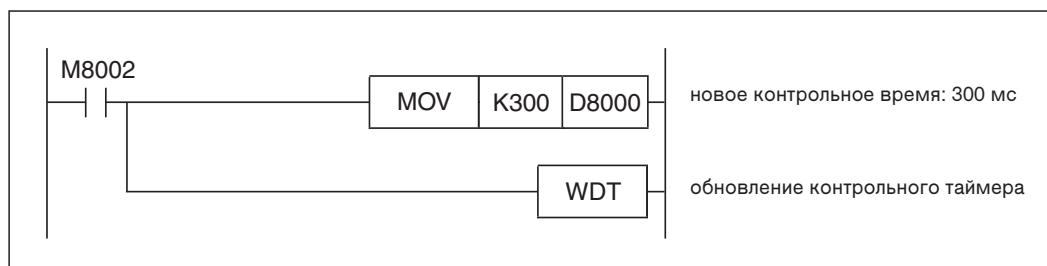


Рис. 7-296:Пример изменения настройки контрольного таймера: в первом программном цикле после запуска контроллера на один цикл устанавливается маркер M8002 и изменяется настройка контрольного

- Изменение количества данных, передаваемых за каждый программный цикл

В (n4) указывается, сколько слов данных требуется передать в одном программном цикле. Чтобы не срабатывал контрольный таймер, уменьшите это значение.

Прерывание команды RBFM или WBFM

Выполнение команды RBFM или WBFM нельзя прерывать. Если выполнение прервано, передаются не все данные. Однако переданные до прерывания данные записываются в базовый блок или буферную память специального модуля.

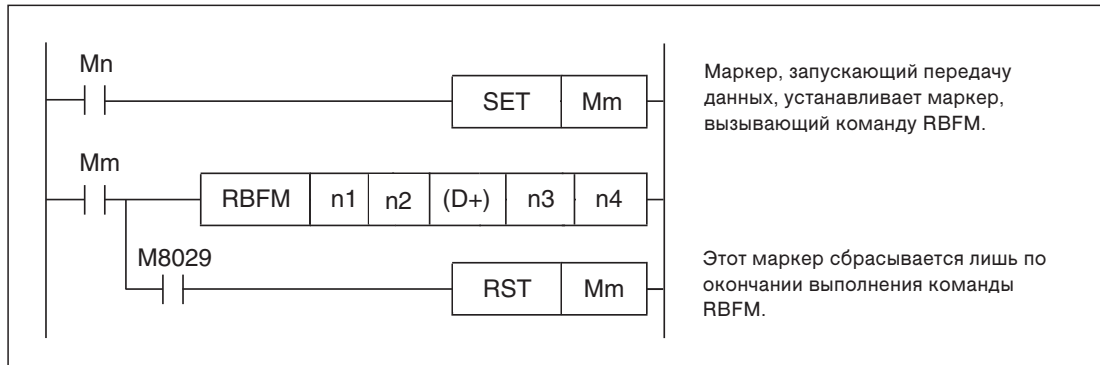


Рис. 7-297: В этом примере программы второй маркер (Mm) предотвращает неполное выполнение команды RBFM после сброса пусковой команды (Mn).

Применение индексных регистров

Если используются индексные регистры, то при вызове команды RBFM или WBFM учитывается содержимое этих регистров. Если содержимое индексных регистров изменилось после запуска команды RBFM или WBFM, оно не влияет на выполнение этих команд.

Области памяти в базовом блоке для данных, принимаемых от специального модуля и передаваемых на специальный модуль

Во время выполнения команды RBFM данные из буферной памяти специального модуля записываются в область операндов от $((D+)+0)$ до $((D+)+(n3))$. Содержимое этих операндов можно оценить лишь после того, как выполнение команды RBFM завершилось. На это указывает маркер M8029.

Команда WBFM считывает данные из области операндов от $((S+)+0)$ до $((S+)+(n3))$ и передает их в буферную память специального модуля. Во время выполнения команды WBFM содержание этой области операндов изменять нельзя, так как иначе могут передаться неправильные данные.

Одновременный доступ к одной и той же области буферной памяти

Во время выполнения команды RBFM не следует с помощью команды WBFM передавать данные в ту же самую область буферной памяти специального модуля. Из-за этого команда RBFM может считать неправильные данные.

7.21.4 ПРИМЕР ПРОГРАММЫ ДЛЯ КОМАНД WBFM И RBFM

В этом примере программы данные передаются в буферную память специального модуля с адресом 2 и считываются из буферной памяти этого модуля.

- При включении входа X0 содержимые регистров данных с D100 по D179 (80 адресов) передаются в адреса буферной памяти с 1001 по 1080. В каждом программном цикле происходит запись в 16 адресов буферной памяти. .
- При включении X1 содержимые адресов в буферной памяти с 2001 по 2080 (80 адресов) передаются в регистры данных с D200 по D279. В каждом программном цикле считываются 16 адресов буферной памяти.

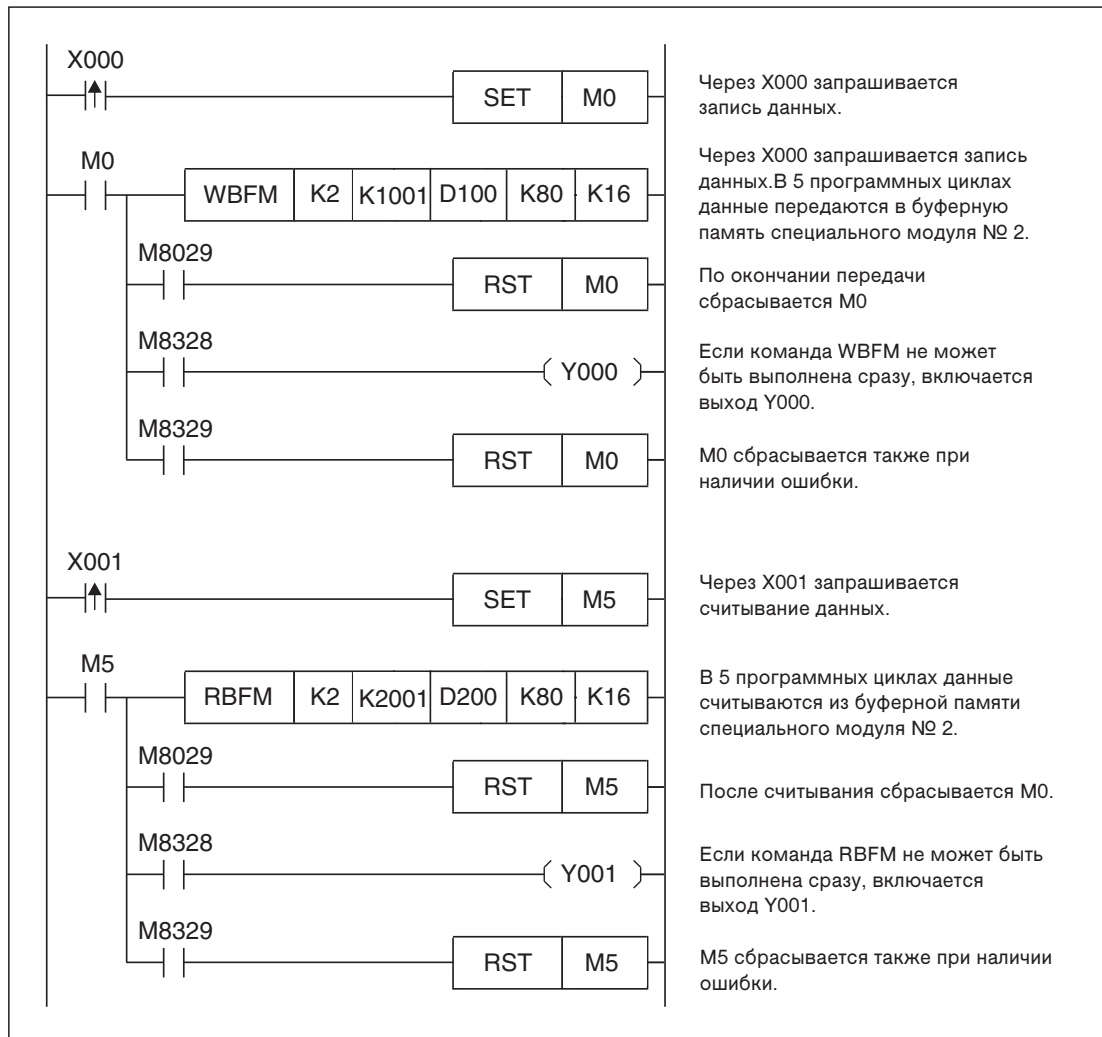


Рис. 7-298:Пример программы для записи 80 слов данных в буферную память специального модуля с адресом 2 и их считывания из этой буферной памяти. .

7.22 КОМАНДА ДЛЯ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА

						DHSCT		FNC 280		
						Сравнение фактического значения высокоскоростного счетчика с данными в таблицах				
Операнды	S1+	n1	S2+	D+	n2	Имп. инструкция(P)	Обработка		Шаги программы	
	D, R	K, H	C*	Y, M, S	K, H		16бита	32бита	DHSCT	21

* Для (S2+) можно указать только один высокоскоростной счетчик (C235...C255).

Функция

В зависимости от результата сравнения фактического значения высокоскоростного счетчика с заданными значениями (количество которых может достигать 128) устанавливаются или сбрасываются до 16 битовых операндов.

Описание

- В (S1+) указывается первый адрес области операндов, в которой хранятся сравниваемые значения и соответствующие наборы битов для вывода. Для каждого сравниваемого значения нужны три 16-битных операнда.
- В (n1) указывается количество сравниваемых значений. Размер указанной в (S1+) области операндов можно рассчитать по формуле (n1) x 3. Для (n1) можно указывать значения от 1 до 128.
- Высокоскоростной счетчик, фактическое значение которого требуется сравнивать, указывается в (S2+).
- В (D+) указывается первый операнд области, на которую требуется влиять по результатам сравнения.
- В (n2) указывается количество битовых операндов начиная с (D+), которые устанавливаются или сбрасываются по результатам сравнения. (n2) может принимать значения между 1 и 16.
- При выполнении команды DHSCT фактическое значение указанного в (S2+) счетчика поочередно сравнивается с заданными значениями, хранящимися в регистрах начиная с (S1+). При соответствии между заданным и фактическим значением набор битов, который также хранится в области операндов, указанной в (S1+), записывается в операнды, заданные с помощью (D+) и (n2).

Запись в таблице	Операнд	Содержимое	Операнд
0	(S1+)+1, (S1+)	(S1+)+2	(D+) ... [(D+)+(n2-1)]
1	(S1+)+4, (S1+)+3	(S1+)+5	
2	(S1+)+7, (S1+)+6	(S1+)+8	
:		:	
m-2	(S1+)+(3m-5), (S1+)+(3m-6)	(S1+)+(3m-4)	
m-1	(S1+)+(3m-2), (S1+)+(3m-3)	(S1+)+(3m-1)	

Табл. 7-73: Взаимосвязь между операндами команды DHSCT

ПРИМЕР

Для (D+) указан выход Y0, для (n2) - значение 10, а в качестве набора битов - „A716H“.

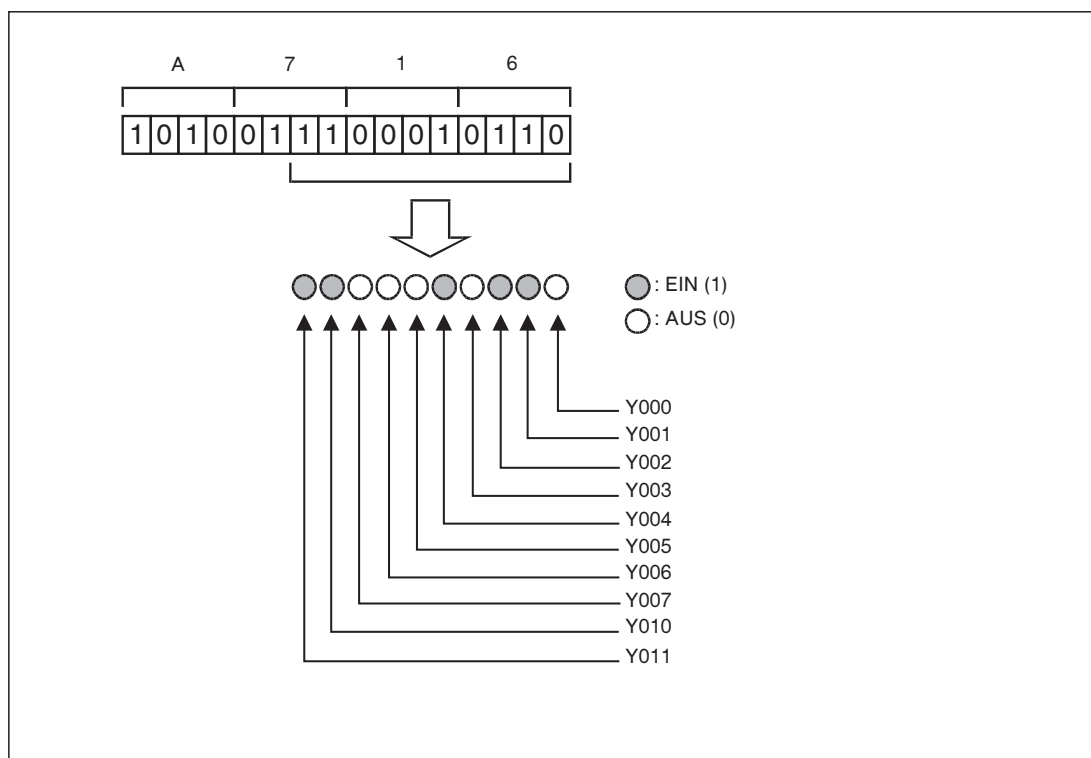


Рис. 7-299: Если фактическое значение соответствует указанному заданному значению, выходы с Y0 по Y11 переключаются так, как это задано набором битов.

- Если для (D+) указывается выход (Y), то состояние выхода обновляется непосредственно при выполнении команды DHSCT. (Обычно выходы обновляются лишь в конце программного цикла.) Самый младший адрес выхода в этом случае должен быть равен „0“ (например, Y000, Y010, Y020 и т. п.).
- С фактическим значением счетчика сравнивается только по одному значению. Сравнение со следующей записью в таблице происходит лишь в том случае, если фактическое значение счетчика совпадает с текущим сравниваемым значением.

Например, если требуется сравнивать фактическое значение счетчика возрастающего счета, необходимо запускать команду DHSCT до тех пор, пока фактическое значение меньше сравниваемого значения в первой записи в таблице. Это пояснено в примере на странице 7-292.

- После распознавания соответствия между сравниваемым и фактическим значением специальный регистр D8138, служащий в качестве указателя таблицы, повышается на „1“. После этого фактическое значение сравнивается со следующим значением. Это повторяется до тех пор, пока содержимое D8138 не достигнет значения, заданного с помощью (n2). В этом случае устанавливается маркер M8138 - чтобы показать, что все сравнения выполнены. Затем содержимое D8138 устанавливается на „0“ и сравнение заданного и фактического значения начинается заново, с первого сравниваемого значения.
- Если входное условие команды DHSCT выключено, сравнения более не выполняются и указатель таблицы D8138 устанавливается на „0“.

УКАЗАНИЕ

В программе можно выполнить только одну команду DHSCT. Если запускается еще одна команда DHSCT, возникает ошибка с кодом 6765 и команда не выполняется.

В первом программном цикле, в котором выполняется команда DHSCT, происходят внутренние логические переключения. Поэтому состояния выходов изменяются лишь начиная со второго программного цикла.

В одной программе могут одновременно действовать до 32 команд для высокоскоростного счетчика (DHSCT, DHSCS, DHSCR и DHSZ). При запуске 33-й команды возникает ошибка с кодом 6705 и команда не выполняется.

Если счетчик в (S2+) задается с помощью индексного регистра, то все высокоскоростные счетчики рассматриваются как программные счетчики.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- В (S2+) не указан высокоскоростной счетчик (C235...C255). (код ошибки „6706“)
- При указании (n1) превышает допустимая область для типа операнда, указанного в (S1+). (код ошибки „6706“)
- При указании (n2) превышает допустимая область для типа операнда, указанного в (D+). (код ошибки „6706“)
- Программа содержит несколько команд DHSCT. (код ошибки „6765“)
- Программа содержит более 32 команд для высокоскоростного счетчика, например, DHSCT, DHSCS, DHSCR или DHSZ. (код ошибки „6706“)

ПРИМЕР ▾

Следующая программа переключает выходы с Y010 по Y012 в зависимости от фактического значения счетчика C235. Фактическое значение сравнивается с 5 значениями, которые вместе с соответствующими наборами битов для выходов хранятся в контроллере, начиная с регистра данных D200.

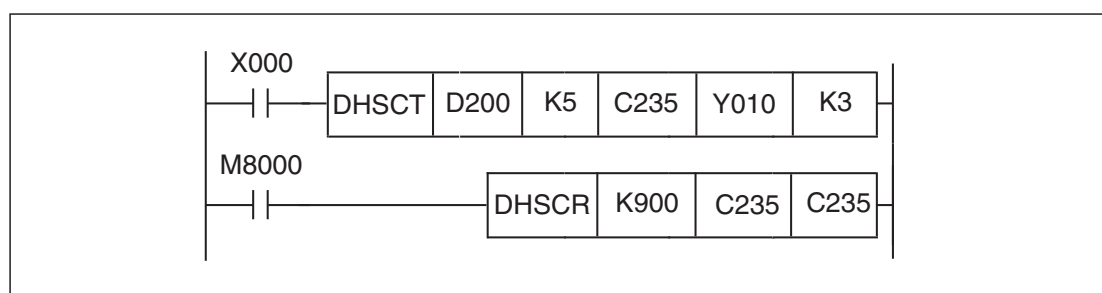


Рис. 7-300: Если счетчик C235 достиг значения „900“, этот счетчик сбрасывается.

Запись в таблице	Сравниваемое значение		Набор битов		Указатель таблицы D8138
	Операнд	Содержимое	Операнд	Содержимое	
0	D201, D200	321	D202	0001H	0
1	D204, D203	432	D205	0007H	1
2	D207, D206	543	D208	0002H	2
3	D210, D209	764	D211	0000H	3
4	D213, D212	800	D214	0003H	4

Табл. 7-74: Сравниваемые значения и наборы битов для этого примера

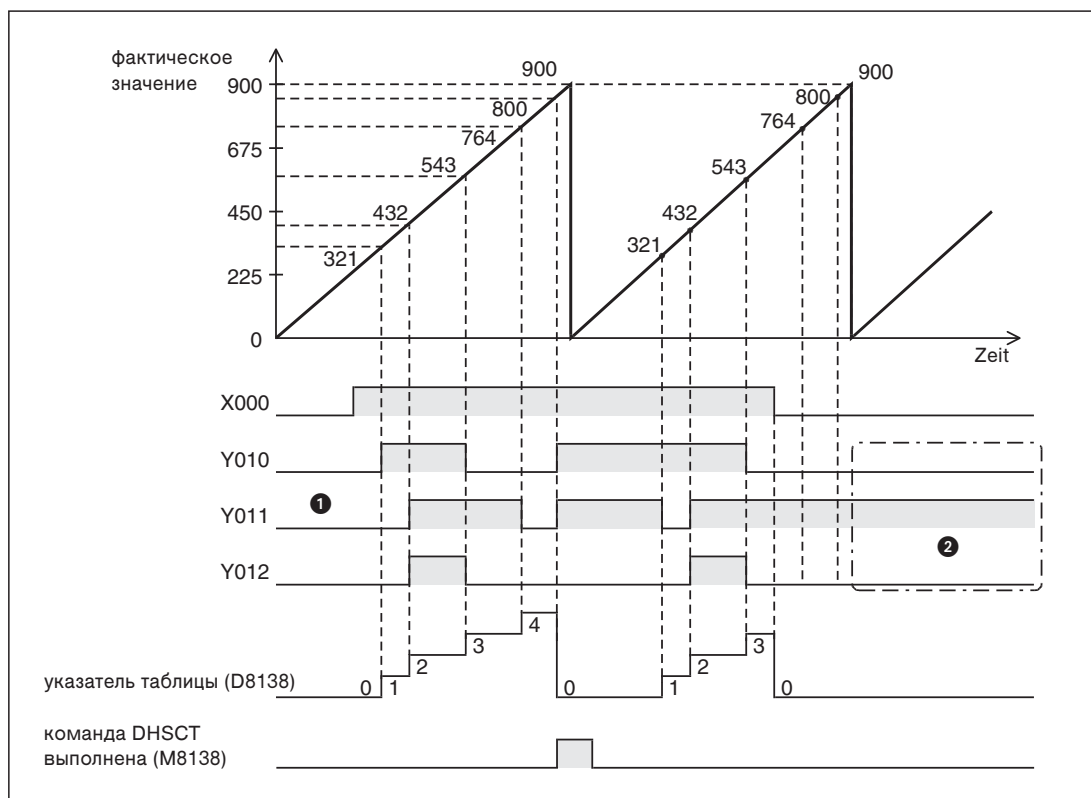


Рис. 7-301: Изменения сигналов для этого примера программы

- ① До тех пор, пока не выполняется команда DHSCT, выходы не управляются.
- ② Так как эта команда DHSCT более не выполняется, она более не изменяет состояния выходов Y10, Y11 и Y12.

7.23 КОМАНДЫ ДЛЯ РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ

Символ	FNC	Значение	Раздел
LOADR	290	Считывание данных из расширенных файловых регистров	7.23.1
SAVER	291	Запись данных в расширенные файловые регистры	7.23.2
INITR	292	Инициализация расширенных регистров и расширенных файловых регистров	7.23.3
LOGR	293	Сохранение значений операндов в расширенных регистрах или расширенных файловых регистрах	7.23.4
RWER	294	Передача данных из расширенных регистров в расширенные файловые регистры	7.23.5
INITER	295	Инициализация расширенных файловых регистров	7.23.6

Табл. 7-75: Обзор команд для расширенных файловых регистров

7.23.1 СЧИТЫВАНИЕ ДАННЫХ ИЗ РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ (LOADR)

		LOADR		FNC 290				
		Считывание данных из расширенных файловых регистров						
		CPU	FX1S	FX1N	FX2N	FX2NC	FX3U	
							●	
Операнды	S+	n		Имп. инструкция(P)	Обработка		Шаги программы	
	R	D, K, H		●	16бита	32бита	LOADR LOADRP	5

Функция

Передача данных из расширенных файловых регистров (ER) кассеты памяти (флэш-EPROM) в расширенные регистры (R) RAM контроллера.

Описание

- В (S+) указывается первый адрес расширенных регистров, начиная с которого должны сохраняться данные. Из расширенных файловых регистров данные считываются начиная с такого же адреса.
- В (n) указывается число адресов, содержимое которых требуется передать. Для (n) можно указывать значения от 0 до 32767. Если задается значение „0“, передаются 32767 адресов.

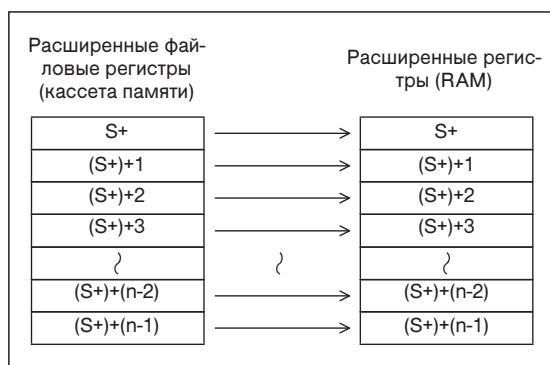


Рис. 7-302:

Команда LOADR передает данные из расширенных файловых регистров в расширенные регистры

- В отличие от команд SAVER, INTR и LOGR, для команды LOADR не требуется указывать сектора.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- Адрес передаваемых регистров превышает значение „32767“. (код ошибки „6706“) В этом случае передаются содержимые регистров до R32767.
- Кассета памяти не установлена. (код ошибки „6771“)

ПРИМЕР ▾

Следующая программа передает содержимые расширенных файловых регистров с ER1 по ER4000 кассеты памяти в расширенные регистры с R1 по R4001 в RAM контроллера.

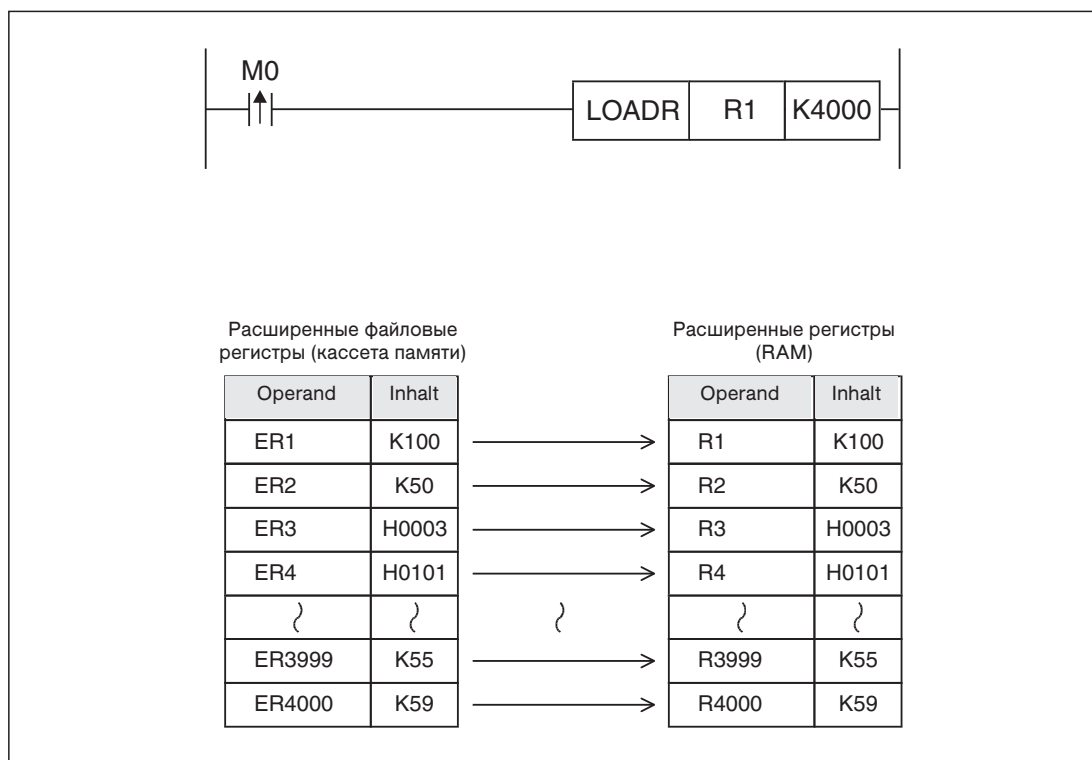


Рис. 7-303: При установке маркера M0 содержимые расширенных файловых регистров с ER1 по ER4001 передаются в расширенные регистры.

△

7.23.2 ЗАПИСЬ ДАННЫХ В РАСШИРЕННЫЕ ФАЙЛОВЫЕ РЕГИСТРЫ (SAVER)

				SAVER		FNC 291			
				Передача данных в расширенные файловые регистры					
				CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
									●
Операнды	S+	n	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	R	K, H	D		16бита	32бита	SAVER	7	
				●					

Функция

Передача данных из расширенных регистров (R) RAM контроллера в расширенные файловые регистры (ER) в кассете памяти (флэш-EPROM). Передаются содержимые 2048 регистров (1 сектор).

Описание

- В (S+) указывается первый адрес области регистра (сектора), который требуется передать.

Команда для (S+)	Сектор	Инициализированная область
R0	0	R0 ... R2047
R2048	1	R2048 ... R4095
R4096	2	R4096 ... R6143
R6144	3	R6144 ... R8191
R8192	4	R8192 ... R10239
R10240	5	R10240 ... R12287
R12288	6	R12288 ... R14335
R14336	7	R14336 ... R16383
R16384	8	R16384 ... R18431
R18432	9	R18432 ... R20479
R20480	10	R20480 ... R22527
R22528	11	R22528 ... R24575
R24576	12	R24576 ... R26623
R26624	13	R26624 ... R28671
R28672	14	R28672 ... R30719
R30720	15	R30720 ... R32767

Табл. 7-76: Сектор задается путем указания начального адреса.

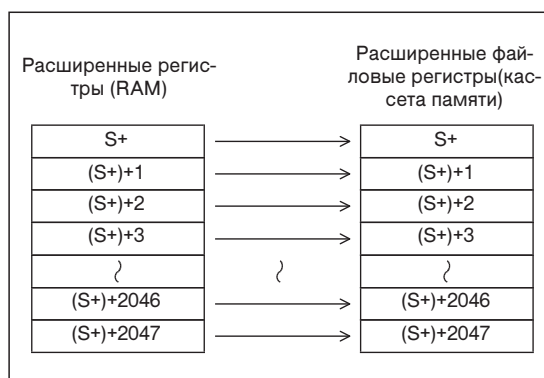


Рис. 7-304:

Команда SAVER передает содержимые 2048 расширенных регистров в расширенные файловые регистры

- В (n) указывается, сколько содержимых регистров должны передаваться в одном программном цикле. Для (n) можно указывать значения от 0 до 2048. Если задается значение „0“, в одном цикле передаются 2048 адресов.
Количество программных циклов, необходимых для выполнения команды SAVER, можно рассчитать, разделив 2048 передаваемых регистров на указанное в (n) количество данных на каждый цикл:
количество программных циклов = $2048 / [n]$
Если при делении образуется остаток, остающиеся данные передаются в дополнительном программном цикле.
- В (D+) записывается количество уже переданных регистров.

Указания по применению команды SAVER

- В качестве альтернативы для передачи данных можно использовать и команду RWER (см. раздел 7.23.5). В случае команды RWER не имеется ограничения по секторам - можно передавать любое количество регистров.
- Запись 2048 регистров в кассету памяти типа флэш-EPROM длится около 50...60 мс. Если в (n) предусмотрена передача большого количества данных в одном программном цикле (например, там указаны значения „0“ или „2048“), может сработать контрольный таймер. В этом случае для (n) выберите значение между „1“ и „1024“, т. е. передавайте данные на протяжении нескольких циклов.
- Если выполнение команды SAVER завершено без ошибок, устанавливается специальный маркер M8029. Но этот маркер управляется и другими командами. Поэтому состояние маркера M8029 необходимо проверять непосредственно после команды SAVER.
- Команду SAVER можно выполнить только в том случае, если перед этим указанные в команде SAVER регистры были инициализированы с помощью команды INITER или INITR. Без этой инициализации при запуске команды SAVER возникает ошибка с кодом 6770.
- Выполнение команды SAVER не разрешается прерывать. При прерывании ее выполнения в расширенные файловые регистры могут быть записаны неправильные данные.

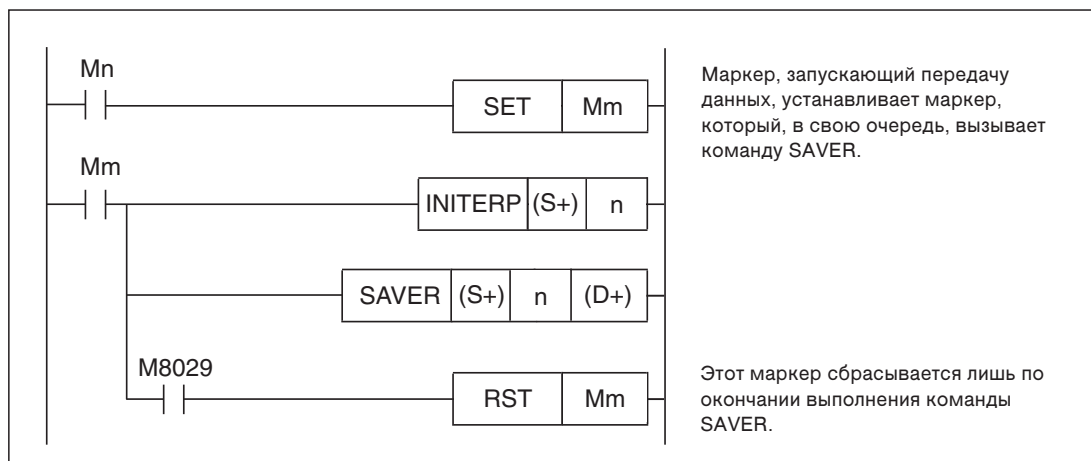


Рис. 7-305: В этом примере программы второй маркер (Mm) предотвращает неполное выполнение команды SAVER после сброса пусковой команды (Mn).

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- IB (S+) указаны иные операнды, а не начальные адреса секторов. (код ошибки 6706)
- Кассета памяти не установлена. (код ошибки „6771“)
- Активирована защита от записи на кассету памяти. (код ошибки „6770“)
- Данные не удалось передать, так как, например, расширенные файловые регистры не были инициализированы. (код ошибки „6770“)

Если возникает эта ошибка, данные в расширенных регистрах (R) утрачиваются. Во избежание этого перед выполнением команды SAVER следует зафиксировать содержимые расширенных регистров с помощью среды программирования GX Developer или GX IEC Developer.

ПРИМЕР ▽

Следующая программа применяется для передачи измененных содержимых расширенных регистров с R10 по R19 (сектор 0) в расширенные файловые регистры с целью сохранения данных. При включении входа X0 в каждом программном цикле передаются 128 регистров.

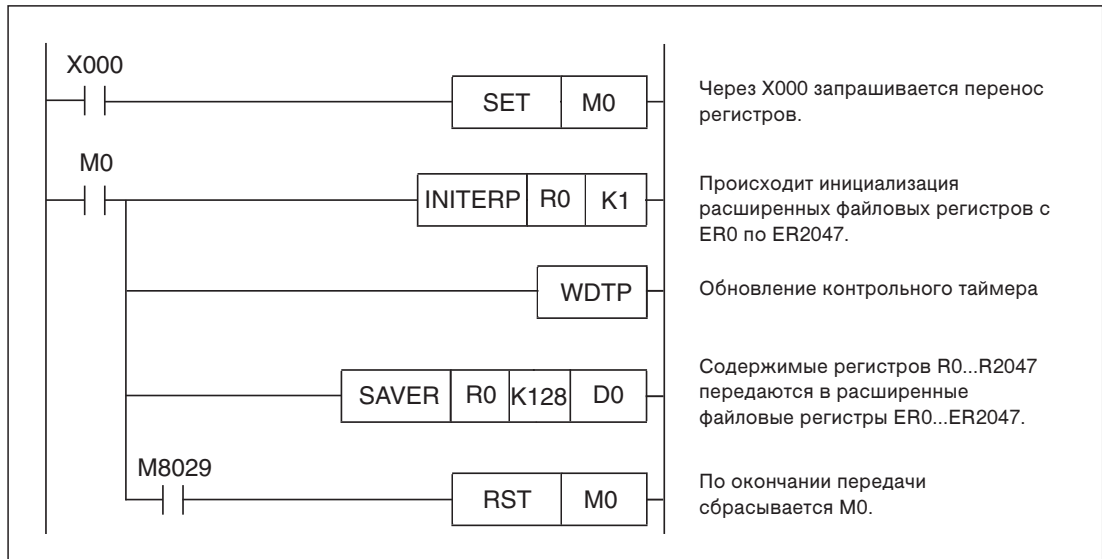


Рис. 7-306: Перед выполнением команды SAVER необходимо с помощью команды INITER инициализировать расширенные файловые регистры.

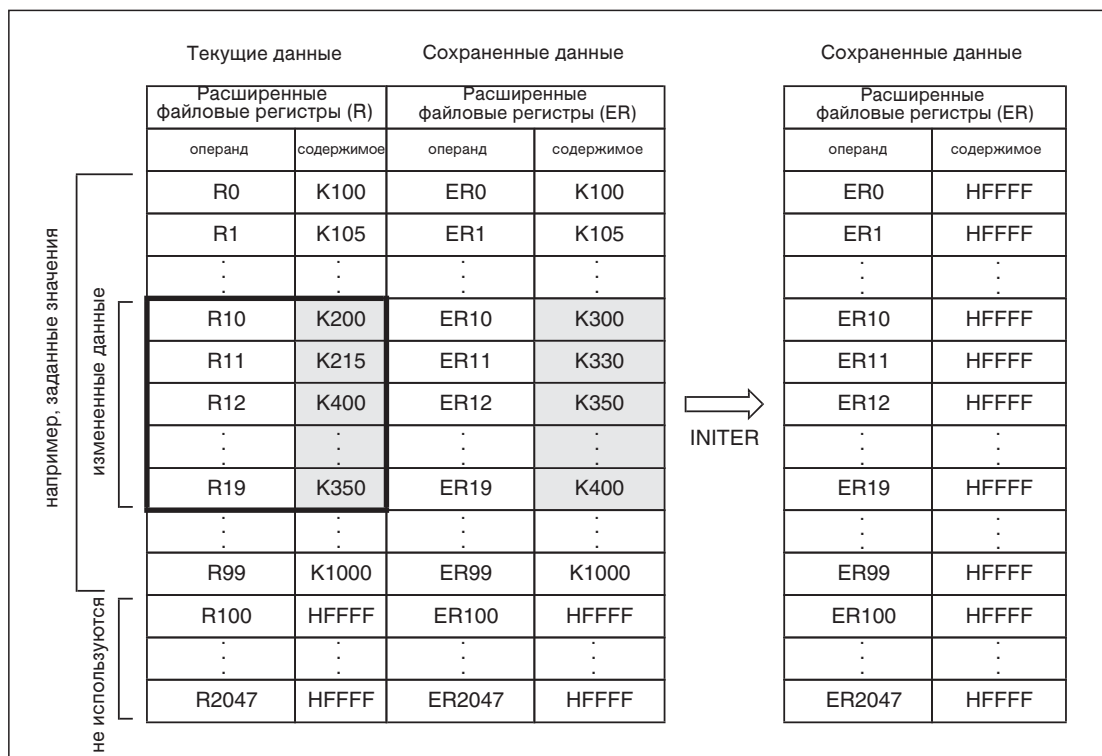


Рис. 7-307: Команда INITER записывает во все расширенные файловые регистры сектора 0 значение HFFFF.



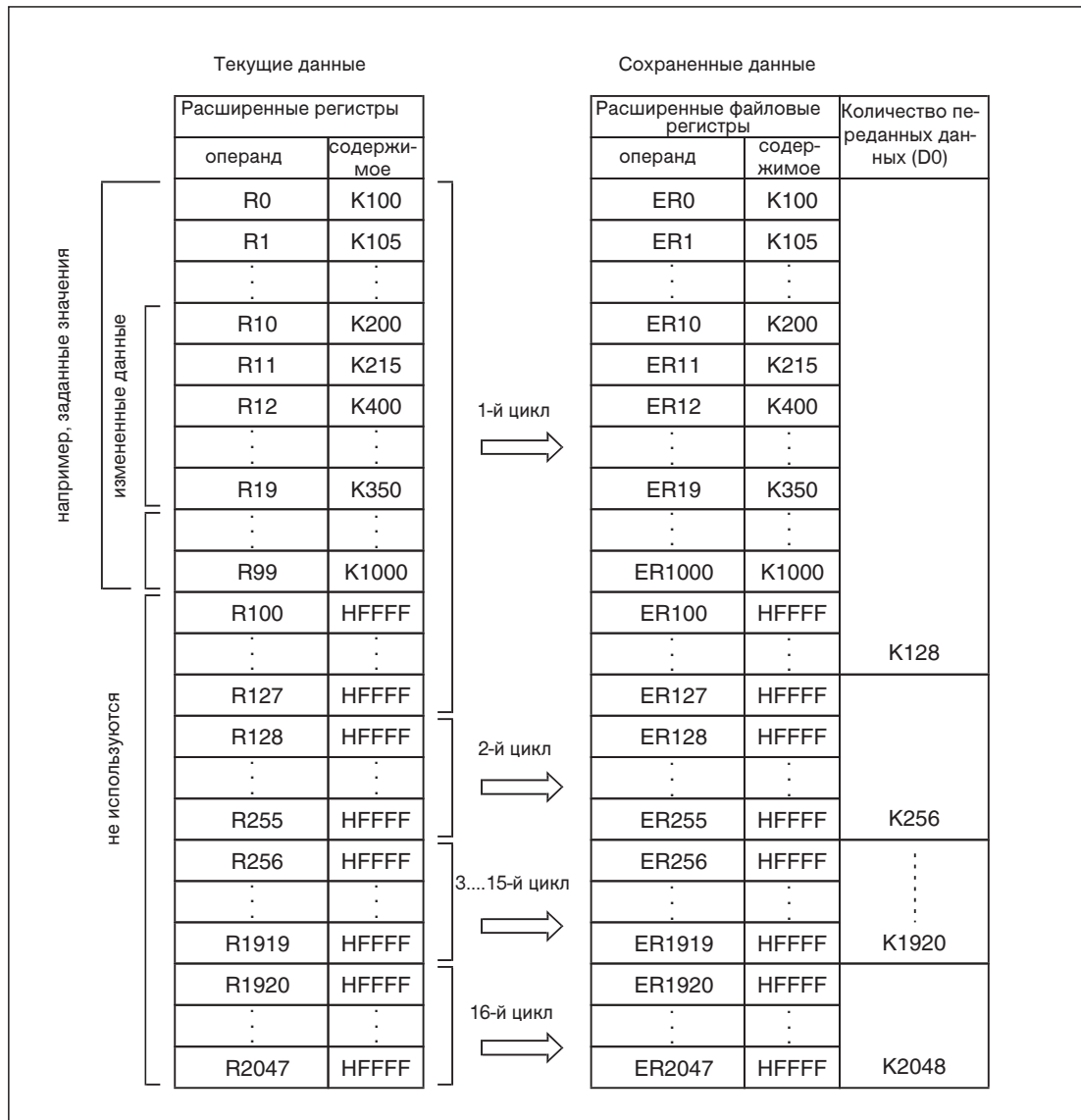


Рис. 7-308:Затем содержимые расширенных регистров с помощью команды SAVER передаются в кассету памяти.

7.23.3 ИНИЦИАЛИЗАЦИЯ РАСШИРЕННЫХ РЕГИСТРОВ И РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ (INITR)

			INITR		FNC 292			
			Инициализация расширенных регистров / файловых регистров					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	S+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	R	D, K, H	●	16бита	32бита	INITR	INITRP	5
				●				

Функция

Инициализация расширенных регистров (R) и расширенных файловых регистров (ER). При этом в регистры вводится значение „FFFFH“.

Описание

- Для инициализации расширенных регистров в RAM базового блока программируемого контроллера и расширенных файловых регистров в кассете памяти (флэш-EPROM) команда INITR заполняет их значениями „FFFFH“ (что соответствует десятичному значению „-1“). Расширенные регистры должны быть инициализированы до того, как в эти регистры происходит запись, например, с помощью команды LOGR.
- Если в контроллере не установлена кассета памяти, то с помощью команды INITR инициализируются только расширенные регистры в RAM центрального процессора.
- Инициализация происходит по секторам. На каждый сектор инициализируются 2048 регистров. С помощью команды INITR можно инициализировать несколько секторов.
- В (S+) указывается первый адрес первой области регистра (сектора), который требуется инициализировать.

Команда для (S+)	Сектор	Инициализированная область	
		Расширенные регистры	Расширенные файловые регистры
R0	0	R0 ... R2047	ER0 bis ER2047
R2048	1	R2048 ... R4095	ER2048 ... ER4095
R4096	2	R4096 ... R6143	ER4096 ... ER6143
R6144	3	R6144 ... R8191	ER6144 ... ER8191
R8192	4	R8192 ... R10239	ER8192 ... ER10239
R10240	5	R10240 ... R12287	ER10240 ... ER12287
R12288	6	R12288 ... R14335	ER12288 ... ER14335
R14336	7	R14336 ... R16383	ER14336 ... ER16383
R16384	8	R16384 ... R18431	ER16384 ... ER18431
R18432	9	R18432 ... R20479	ER18432 ... ER20479
R20480	10	R20480 ... R22527	ER20480 ... ER22527
R22528	11	R22528 ... R24575	ER22528 ... ER24575
R24576	12	R24576 ... R26623	ER24576 ... ER26623
R26624	13	R26624 ... R28671	ER26624 ... ER28671
R28672	14	R28672 ... R30719	ER28672 ... ER30719
R30720	15	R30720 ... R32767	ER30720 ... ER32767

Табл. 7-78:Первый сектор устанавливается путем указания начального адреса.

- В (n) указывается количество инициализируемых секторов.

Операнд	Расширенные регистры		Расширенные файловые регистры	
	Содержимое перед выполнением команды INITR	Содержимое после выполнения команды INITR	Содержимое перед выполнением команды INITR	Содержимое после выполнения команды INITR
(S+)	0010H	FFFFH	1234H	FFFFH
(S+)+1	0020H	FFFFH	5678H	FFFFH
(S+)+2	0011H	FFFFH	90ABH	FFFFH
:	:	:	:	:
(S+)+(2048 x n)-1	ABCDH	FFFFH	CDEFH	FFFFH

Табл. 7-77:Команда INITR заполняет расширенные регистры и расширенные файловые регистры значением FFFFH.

Указания по применению команды INITR

- С помощью команды INITR одновременно инициализируются расширенные регистры (R) и расширенные файловые регистры (ER). Если требуется инициализировать только расширенные файловые регистры в кассете памяти, используйте команду INITER (раздел 7.23.6).
- Инициализация одного сектора (2048 регистров) кассеты памяти длится около 25 мс. В связи с этим может возникнуть ошибка контрольного таймера. (Если кассета памяти не установлена и инициализируются только расширенные регистры в RAM, то инициализация одного сектора происходит не более чем за 1 мс.) В этом случае можно изменить настройку контрольного таймера или фрагмент программы, содержащий команду INITR.
- Изменение настройки контрольного таймера

Время контроля изменяется в результате записи нового значения в специальный регистр D8000. Это время предварительно установлено на 200 мс. Чтобы активировать новое время сразу, следует дополнительно выполнить команду WDT. Без этой команды новая

настройка контрольного таймера начинает действовать лишь со следующего программного цикла.

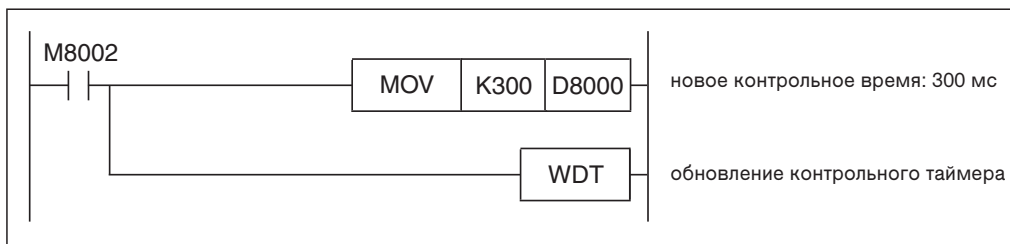


Рис. 7-309:Пример изменения настройки контрольного таймера: в первом цикле программы после запуска контроллера на один цикл устанавливается маркер M8002 и изменяется настройка контрольного таймера.

- Обновление контрольного таймера перед и после выполнения команды INITR

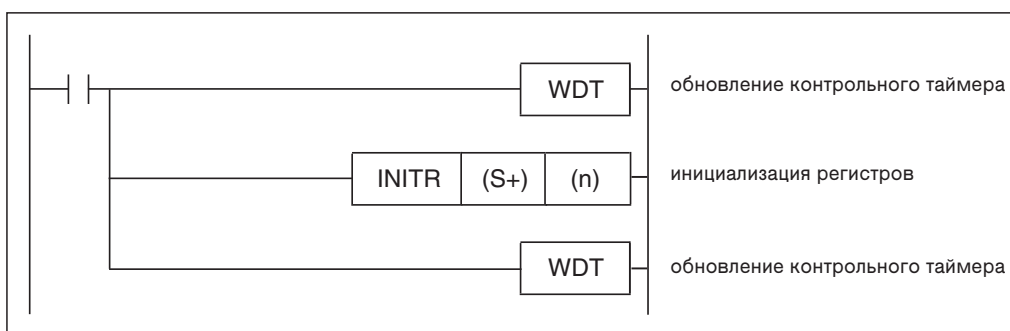


Рис. 7-310:Путем обновления контрольного таймера предотвращается возникновение ошибки при инициализации.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- В (S+) указаны иные операнды, а не начальные адреса секторов. (код ошибки „6706“)
- Адрес инициализируемых регистров превышает значение „32767“. (код ошибки „6706“). В этом случае передаются регистры до R32767 или до ER32767.
- Активирована защита от записи на кассету памяти. (код ошибки „6770“)

ПРИМЕР ▾

В следующем примере программы инициализируются расширенные регистры с R0 по R2048 (сектор 0). Если кассета памяти установлена, инициализируются также расширенные файловые регистры с ER0 по ER2048.

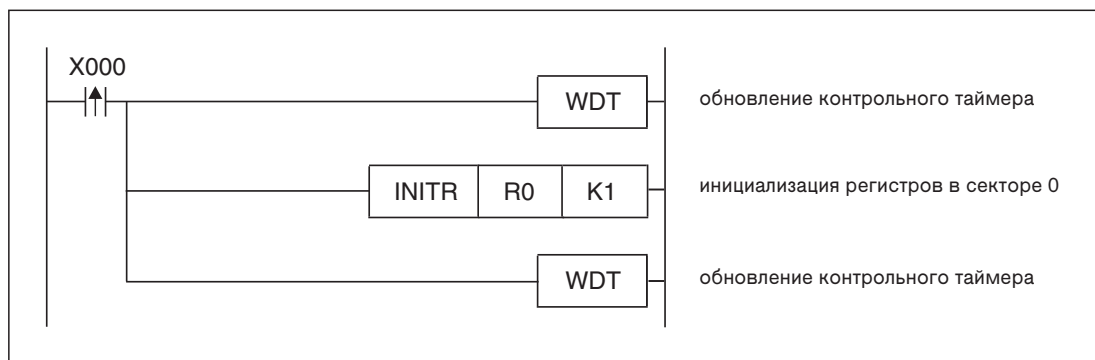
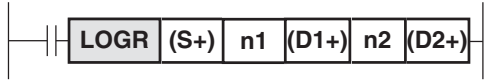


Рис. 7-311:При включении входа X0 инициализируются расширенные регистры с R0 по R2048 и расширенные файловые регистры с ER0 по ER2048.

7.23.4 СОХРАНЕНИЕ ЗНАЧЕНИЙ ОПЕРАНДОВ В РАСШИРЕННЫХ РЕГИСТРАХ И РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРАХ (LOGR)

						LOGR		FNC 293			
						Сохранение значений операндов в расширенных регистрах или расширенных файловых регистрах					CPU
Операнды	S+	n1	D1+	n2	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	T*, C, D	D	R	K, H	D		16бита	32бита	LOGR	LOGRP	11
						●	●				

* Счетчики с C200 по C255 указывать нельзя.

Функция

Сохранение содержимых словных операндов в расширенных регистрах (R) и расширенных файловых регистрах (ER).

Описание

- С помощью команды LOGR можно реализовать регистрацию данных. При этом сохраняются содержимые до 8000 словных операндов из взаимосвязанной области.
- Если в контроллере не установлена кассета памяти, данные передаются только в расширенные регистры RAM центрального процессора.
- В (S+) указывается первый адрес регистрируемой области операндов.
- В (n1) устанавливается количество операндов, содержимое которых требуется зарегистрировать. Можно регистрировать от 1 до 8000 операндов.
- В (D1+) указывается первый адрес первой области регистра (сектора), в котором должны сохраняться данные.

Указание в (D1+)	Сектор	Область памяти	
		Расширенные регистры	Расширенные файловые регистры
R0	0	R0 ... R2047	ER0 ... ER2047
R2048	1	R2048 ... R4095	ER2048 ... ER4095
R4096	2	R4096 ... R6143	ER4096 ... ER6143
R6144	3	R6144 ... R8191	ER6144 ... ER8191
R8192	4	R8192 ... R10239	ER8192 ... ER10239
R10240	5	R10240 ... R12287	ER10240 ... ER12287
R12288	6	R12288 ... R14335	ER12288 ... ER14335
R14336	7	R14336 ... R16383	ER14336 ... ER16383
R16384	8	R16384 ... R18431	ER16384 ... ER18431
R18432	9	R18432 ... R20479	ER18432 ... ER20479
R20480	10	R20480 ... R22527	ER20480 ... ER22527
R22528	11	R22528 ... R24575	ER22528 ... ER24575
R24576	12	R24576 ... R26623	ER24576 ... ER26623
R26624	13	R26624 ... R28671	ER26624 ... ER28671
R28672	14	R28672 ... R30719	ER28672 ... ER30719
R30720	15	R30720 ... R32767	ER30720 ... ER32767

Табл. 7-79:Первый сектор устанавливается путем указания начального адреса.

- В (n2) указывается количество секторов, в которые требуется записывать данные. (n2) может принимать значения между 1 и 16. Данные сохраняются до тех пор, пока не будут заполнены все указанные регистры.
- (D2) содержит количество уже зарегистрированных данных.

На рисунке ниже изображена структура данных при регистрации.

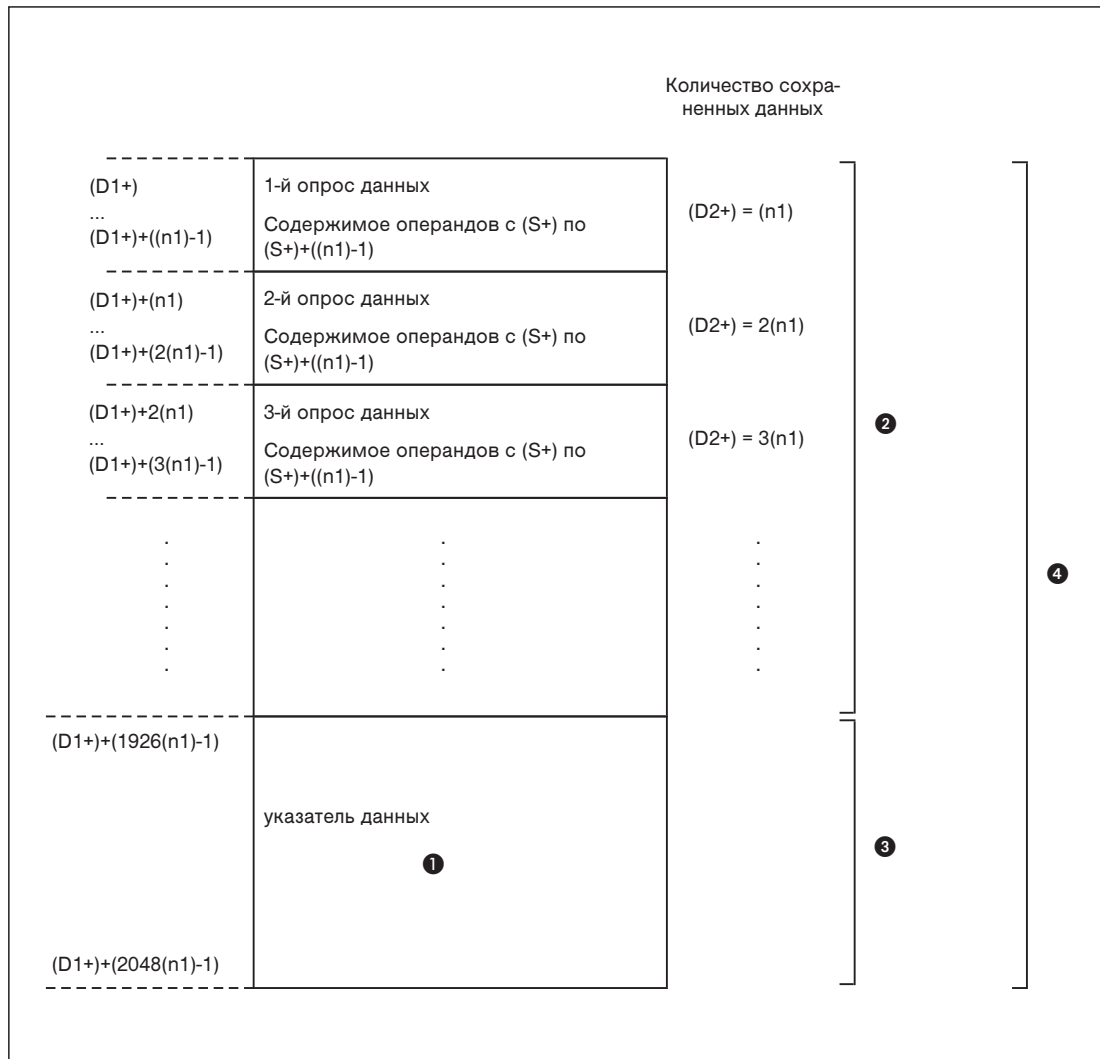


Рис. 7-312: Использование расширенных регистров и расширенных файловых регистров командой LOGR.

- ① В этой области для каждого слова, в которое были введены зарегистрированные данные, сбрасывается один бит. (При инициализации были установлены все биты регистра.) Этот процесс начинается с бита 0 операнда $(D1+) + (1926(n1)-1)$. Если сброшены все 16 битов этого операнда, этот процесс продолжается с бита 0 следующего операнда.
- ② В этой области вводятся зарегистрированные данные. Размер этой области составляет $1926 \times (n2)$ регистров.
- ③ Эта область служит для управления регистрацией и содержит указатель данных. Эта область занимает $122 \times (n2)$ регистров.
- ④ Вся область для регистрации данных имеет размер $2048 \times (n2)$ регистров.

Указания по применению команды LOGR

- С помощью команды LOGR регистрируются циклические данные. Если данные требуется зарегистрировать только один раз в определенный момент времени, используйте для этого команда LOGRP.
- Перед выполнением команды LOGR необходимо инициализировать расширенные регистры и расширенный регистр файлов с помощью команды INITR (раздел 7.23.3). Без этой инициализация при выполнении команды LOGR возникает ошибка с кодом 6770.

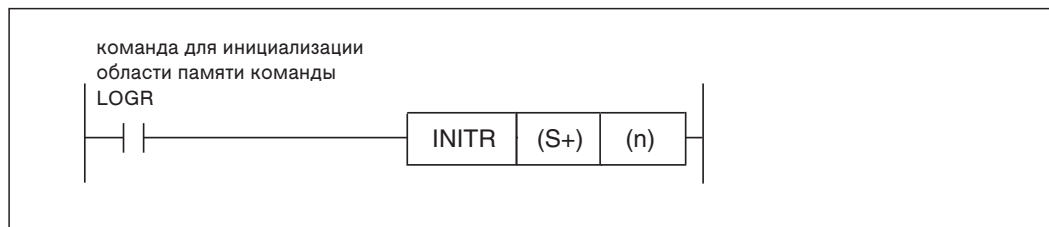


Рис. 7-313:Перед регистрацией данных необходимо инициализировать регистры.

В (S+) команды INITR необходимо указать тот же операнд, что и в (D1+) команды LOGR.

Параметр (n) команды INITR и параметр (n2) команды LOGR должны быть идентичны.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- В (S+) указаны иные операнды, а не начальные адреса секторов. (код ошибки 6706)
- Не удалось сохранить все зарегистрированные данные, так как имеющейся памяти не достаточно. (код ошибки „6706“)
- Активирована защита от записи на кассету памяти. (код ошибки „6770“)
- Данные не удалось сохранить, так как, например, не были инициализированы расширенные файловые регистры. (код ошибки „6770“)

Если возникает эта ошибка, данные в расширенных регистрах (R) утрачиваются. Во избежание этого содержимые расширенных регистров следует зафиксировать с помощью среды программирования GX Developer или GX IEC Developer.

ПРИМЕР ▾

При включенном входе X1 следующая программа определяет содержимые регистров данных D0 и D1 и сохраняет их в расширенных регистрах с R2048 по R6143.

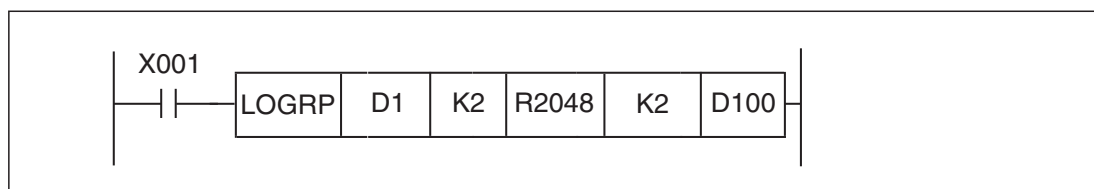


Рис. 7-314:В этом примере программы D100 содержит количество сохраненных данных.

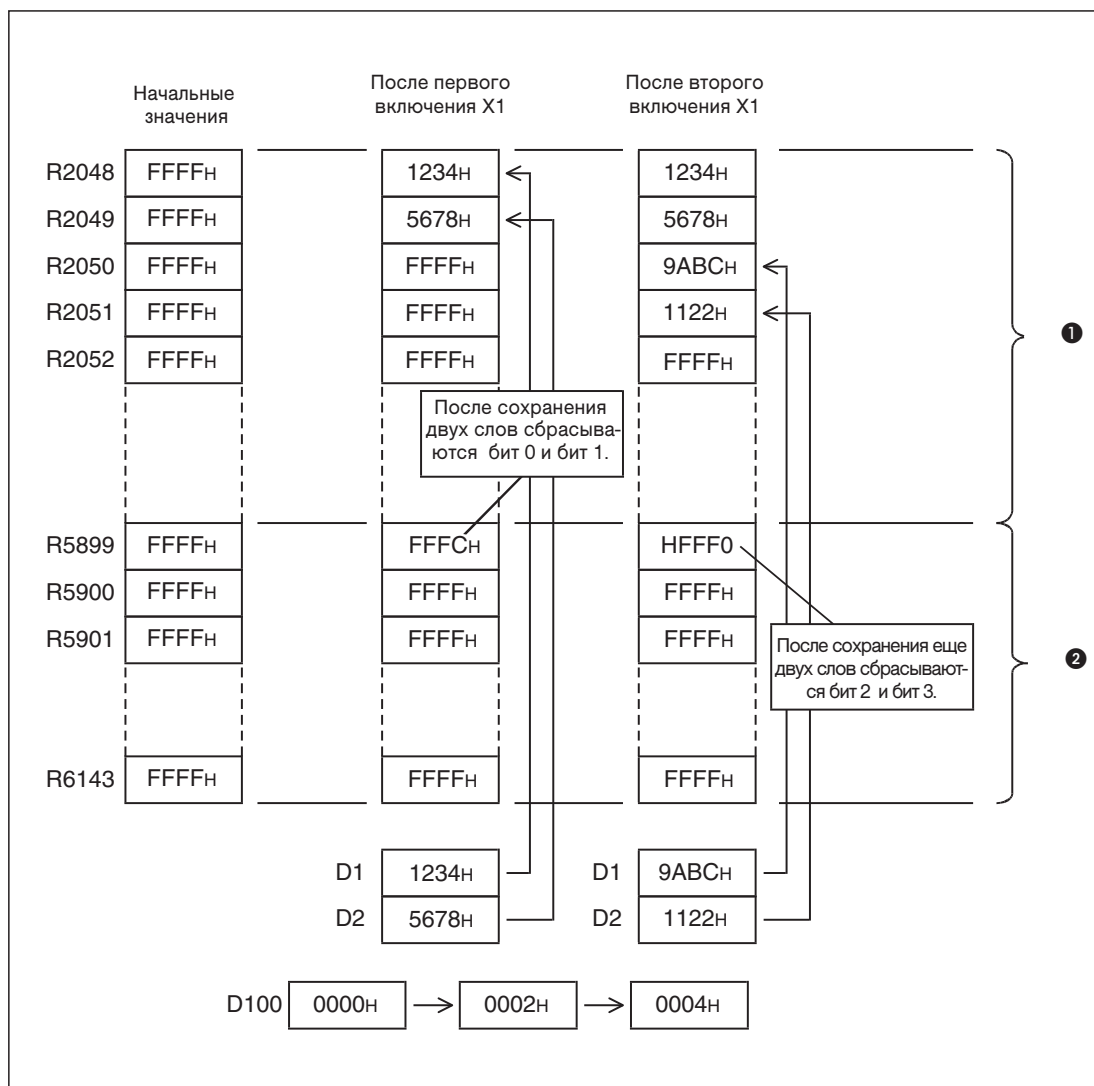
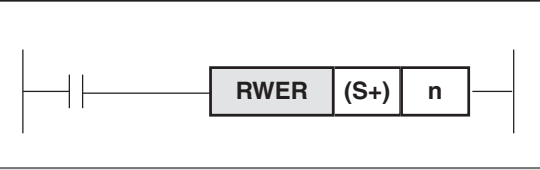


Рис. 7-315:Пример содержимых регистров, если вход X1 переключается два раза.

- 1 Эта область, охватывающая 3852 регистров (1926 x 2), может быть использована для сохранения содержимых D0 и D1.
- 2 Эта область с 244 регистрами служит для управления регистрацией и содержит указатель данных.

7.23.5 ПЕРЕДАЧА ДАННЫХ ИЗ РАСШИРЕННЫХ РЕГИСТРОВ В РАСШИРЕННЫЕ ФАЙЛОВЫЕ РЕГИСТРЫ (RWER)

			RWER		FNC 294			
			Передача расширенных регистров в расширенные файловые регистры					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	S+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	R	D, K, H	●	16бита	32бита	RWER	RWERP	5
				●				

Функция

Передача данных из расширенных регистров (R) RAM контроллера в расширенные файловые регистры (ER) в кассете памяти (флэш-EPROM). С помощью команды RWER можно передать от 1 до 32767 регистров. В отличие от команды SAVER (раздел 7.23.2), здесь нет привязки к секторам.

Описание

- В (S+) указывается первый адрес области регистра, содержимое которой требуется передать.
- (n) указывает, содержимое скольких регистров требуется передать. Для (n) можно указывать значения от 0 до 32767. Если задается значение "0", передается содержимое 32768 адресов.

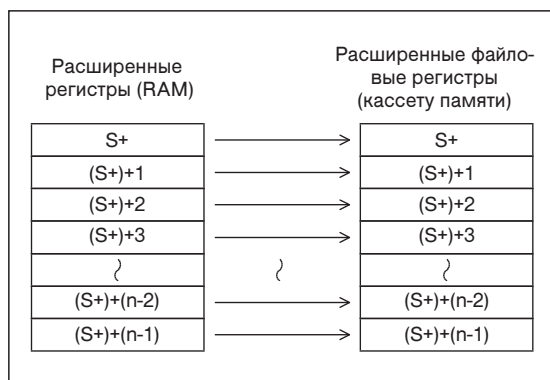


Abb. 7-316:

Команда RWER передает содержимое (n) расширенных регистров в расширенные файловые регистры

Указания по применению команды RWER

- Перед выполнением команды RWER нет необходимости инициализировать регистры с помощью команды INITR или INITER.
- Выполнение команды RWER нельзя прерывать. Например, если ее выполнение прервано исчезновением напряжения питания, в расширенные файловые регистры могут быть записаны неправильные данные. Регулярно фиксируйте содержимое расширенных регистров с помощью среды программирования GX Developer или GX IEC Developer.
- Запись данных одного сектора в кассету памяти типа флэш-EPROM длится около 66 мс. В результате этого может сработать контрольный таймер. На рисунке ниже показан пример программы, в которой перед выполнением команды RWER изменяется настройка контрольного таймера, а по окончании передачи данных снова восстанавливается первоначальная настройка контрольного таймера.

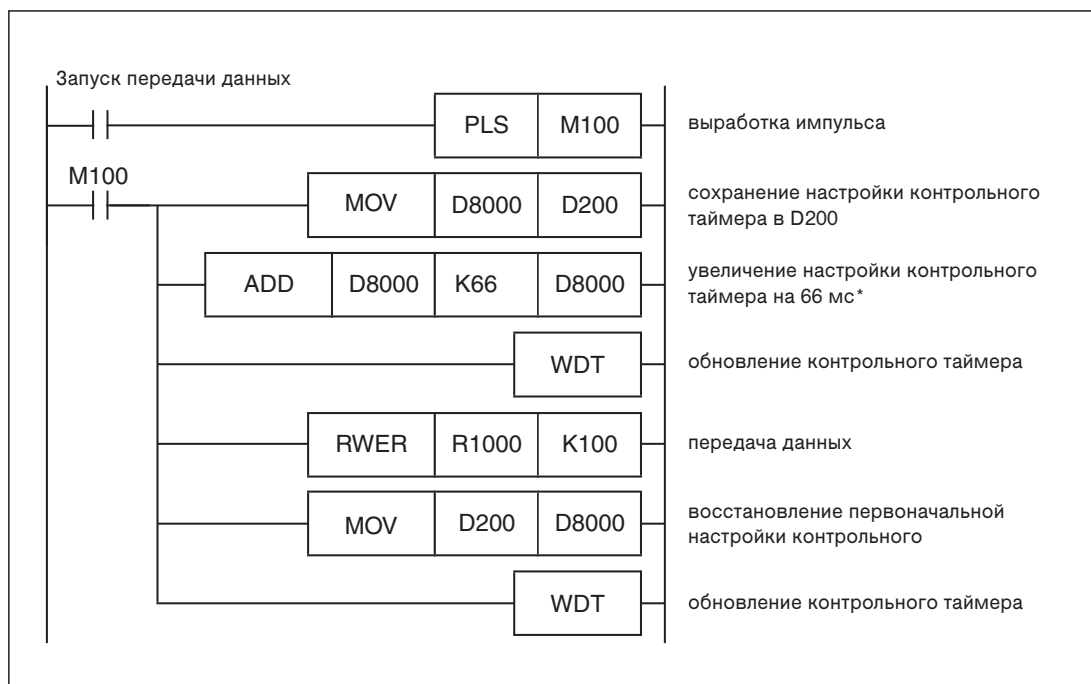


Рис. 7-317:Фрагмент программы для согласования контрольного таймера

* Для каждого сектора, в который записываются данные, настройку контрольного таймера требуется повысить на 66 мс. Границы секторов изображены в разделе 7.23.3.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- Последний адрес передаваемых регистров превышает значение „32767”. (код ошибки “6706”)
- Кассета памяти не установлена. (код ошибки „6771”)
- Активирована защита от записи на кассету памяти. (код ошибки „6770”)

ПРИМЕР ▽

При включенном входе X0 следующая программа передает содержимые расширенных регистров с R10 по R19 в расширенные файловые регистры с целью сохранения данных.

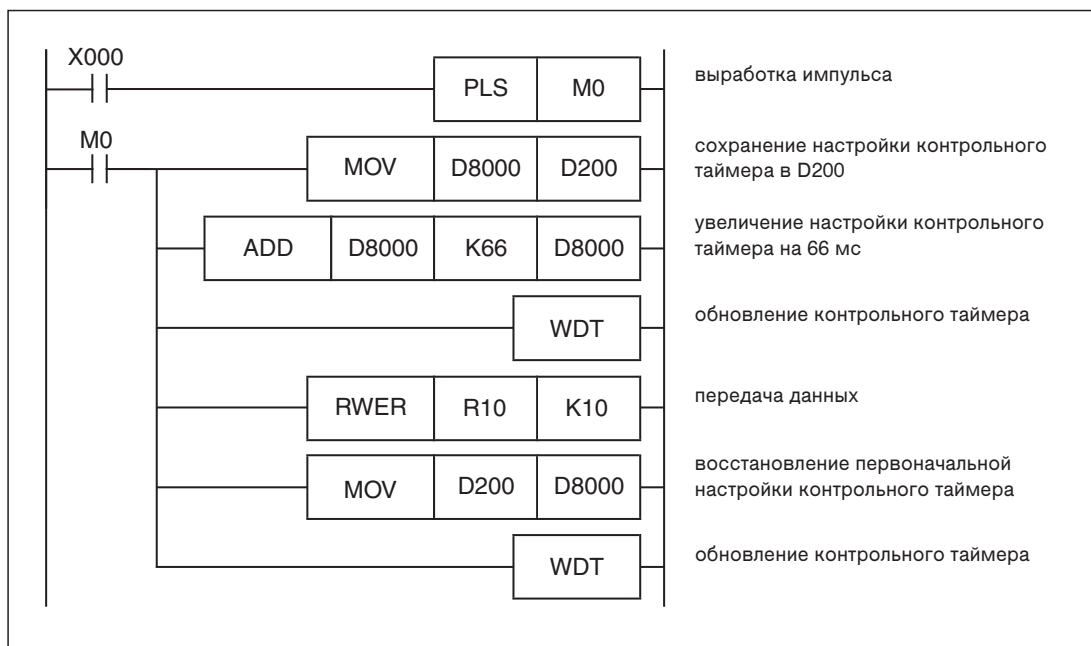


Рис. 7-318: В этом примере программы содержимые 10 расширенных регистров с R10 по R19 сохраняются в кассете памяти. На время передачи данных повышается настройка контрольного таймера.

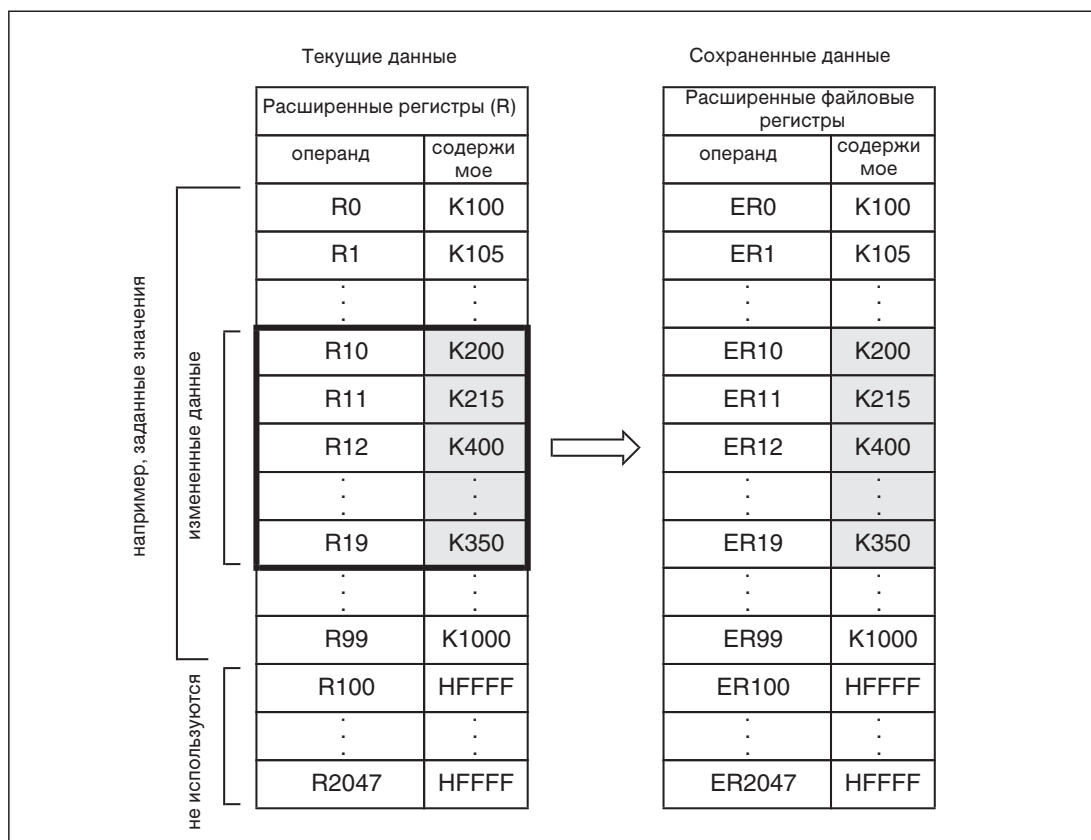


Рис. 7-319: Данные передаются при включении X001.

7.23.6 ИНИЦИАЛИЗАЦИЯ РАСШИРЕННЫХ ФАЙЛОВЫХ РЕГИСТРОВ (INITER)

			INITER		FNC 295			
			Инициализация расширенных файловых регистров					
			CPU	FX1S	FX1N	FX2N	FX2NC	FX3U
								●
Операнды	S+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	R	K, H	●	16бита	32бита	INITER INITERP		5

Функция

Инициализация расширенных файловых регистров (ER). При этом в регистры вводится значение „FFFFH“.

Описание

- В целях инициализации команда INITER заполняет расширенные файловые регистры, находящиеся в кассете памяти (флэш-EPROM), значением "FFFFH" (что соответствует десятичному значению "-1") . Инициализация расширенных файловых регистров должна быть выполнена прежде, чем эти регистры используются для записи (например, командой SAVER).
- Инициализация происходит по секторам. На каждый сектор инициализируются 2048 регистров. С помощью команды INITER можно инициализировать несколько секторов.
- В (S+) указывается первый адрес первой области регистра (сектора), который требуется инициализировать. Учитывайте, что в качестве адреса должен быть указан "R", а не "ER".

Указание в (S+)	Сектор	Область памяти
R0	0	ER0 ... ER2047
R2048	1	ER2048 ... ER4095
R4096	2	ER4096 ... ER6143
R6144	3	ER6144 ... ER8191
R8192	4	ER8192 ... ER10239
R10240	5	ER10240 ... ER12287
R12288	6	ER12288 ... ER14335
R14336	7	ER14336 ... ER16383
R16384	8	ER16384 ... ER18431
R18432	9	ER18432 ... ER20479
R20480	10	ER20480 ... ER22527
R22528	11	ER22528 ... ER24575
R24576	12	ER24576 ... ER26623
R26624	13	ER26624 ... ER28671
R28672	14	ER28672 ... ER30719
R30720	15	ER30720 ... ER32767

Табл. 7-80:Первый сектор устанавливается путем указания начального адреса. В (n) указывается количество инициализируемых секторов

- В (n) указывается количество инициализируемых секторов.

Операнд	Содержимое перед выполнением команды INITER	Содержание после выполнения команды INITER
(S+)	1234H	FFFFH
(S+)+1	5678H	FFFFH
(S+)+2	90ABH	FFFFH
:	:	:
(S+)+(2048 x n)-1	CDEFH	FFFFH

Табл. 7-81: Команда INITER заполняет расширенные файловые регистры кодами FFFFH.

Указания по применению команды INITER

- С помощью команды INITER инициализируются только расширенные файловые регистры в кассете памяти. Если одновременно требуется инициализировать расширенные регистры (R) и расширенные файловые регистры (ER), используйте команду INITR (раздел 7.23.3).
 - Инициализация одного сектора (2048 регистров) кассеты памяти длится около 25 мс. В связи с этим может возникнуть ошибка контрольного таймера. В этом случае можно изменить настройку контрольного таймера или фрагмент программы, содержащий команду INITR.
- Изменение настройки контрольного таймера

Время контроля изменяется в результате записи нового значения в специальный регистр D8000. Это время предварительно установлено на 200 мс. Чтобы активировать новое время сразу, следует дополнительно выполнить команду WDT. Без этой команды новая настройка контрольного таймера начинает действовать лишь со следующего программного цикла.

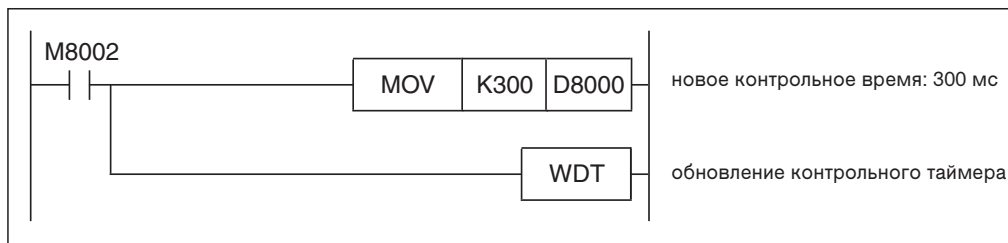


Рис. 7-320: Пример изменения настройки контрольного таймера: в первом цикле программы после запуска контроллера на один цикл устанавливается маркер M8002 и изменяется настройка контрольного таймера..

- Обновление контрольного таймера перед и после выполнения команды INITER

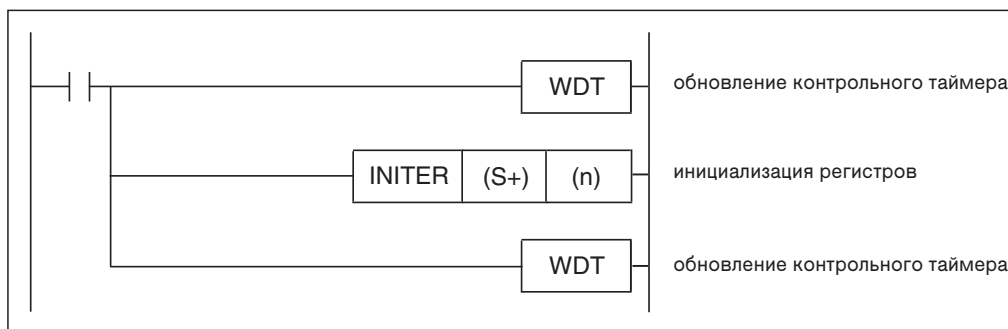


Рис. 7-321: Благодаря актуализации контрольного таймера предотвращается возникновение ошибки при инициализации.

Источники ошибок

В следующих случаях возникает ошибка обработки, устанавливается специальный маркер M8067 и в специальный регистр D8067 записывается код ошибки:

- В (S+) указаны иные операнды, а не начальные адреса секторов. (код ошибки „6706“)
- Адрес инициализируемых регистров превышает значение „32767“. (код ошибки “6706“). В этом случае передаются регистры до ER32767.
- Активирована защита от записи на кассету памяти. (код ошибки „6770“)
- Кассета памяти не установлена. (код ошибки „6771“)

ПРИМЕР ▾

В следующем примере программы инициализируются расширенные файловые регистры с ER0 по ER2048 (сектор 0).

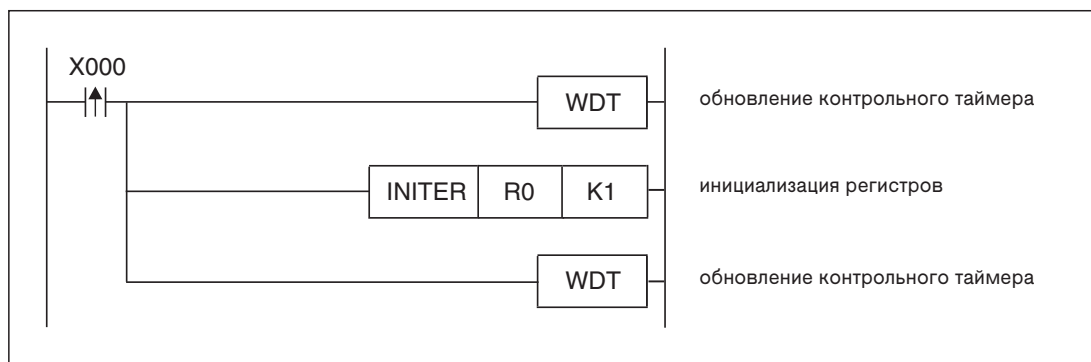


Рис. 7-322: При включении входа X0 инициализируются расширенные файловые регистры с ER0 по ER2048.



8 СПЕЦИАЛЬНЫЕ ФУНКЦИИ

Несколько специальных функций, которые позволяют расширить возможности применения ПК. Эти специальные функции собраны в отдельный раздел, поскольку они не выполняются непосредственно определенными инструкциями.

В этом разделе описаны следующие специальные функции:

- Сохранение данных в режиме СТОП
- Работа с постоянным временем цикла программы (скана)
- Функции кода разрешения работы
- Функции фиксации импульсов
- Установка входного фильтра
- Интегрированные потенциометры ПК серии FX1S и FX1N
- Функции реального времени
- Регистры файлов
- Переключение режима РАБОТА/СТОП
- Базовые аппараты 24 В постоянного тока
- Модули отображения FX1N-5DM для FX1S и FX1N

8.1 СОХРАНЕНИЕ ДАННЫХ В РЕЖИМЕ “СТОП”

В “нормальном” процессе работы ПК серии FX переключают состояния всех выходных сигналов на “0”, как только система управления переключится с режима РАБОТА в режим СТОП. Для некоторых управляющих систем имеет однако смысл и в режиме СТОП сохранить последнее состояние выходных сигналов. Это можно выполнить если включить в программу ПК специальный меркер M8033. При этом накопленные значения таймеров и счетчиков также будут сохранены.

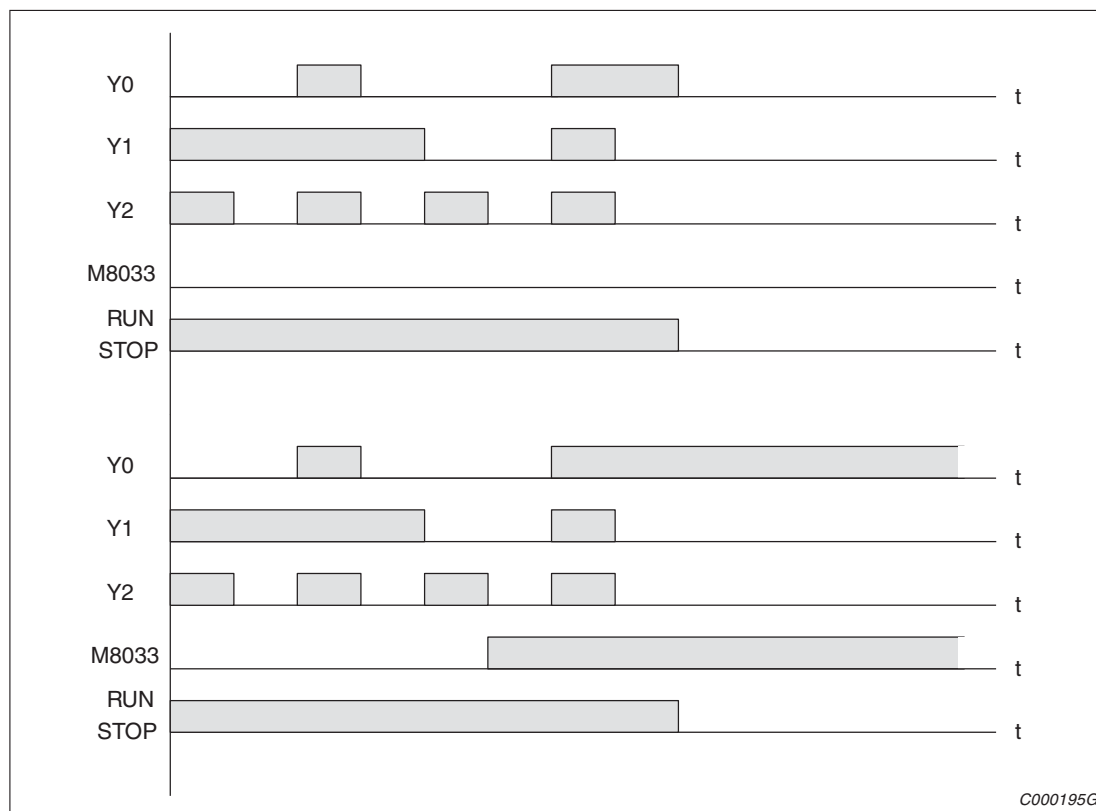


Рис. 8-1. Пример сохранения данных в режиме СТОП

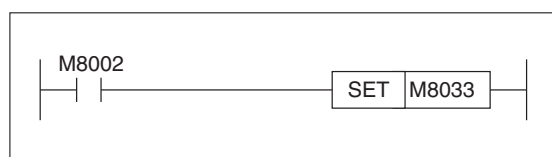


Рис. 8-2.
Пример программирования

C000193C

8.2 РАБОТА С ПОСТОЯННЫМ ВРЕМЕНЕМ ЦИКЛА ПРОГРАММЫ (СКАНОМ)

ПК серии FX могут работать с постоянным временем цикла программы независимо от самой программы ПК. Это необходимо, например, при применении RAMP-инструкции.

Чтобы получить постоянное время цикла программы, в программу ПК должен быть включен специальный меркер M8039. Время цикла программы может устанавливаться, начиная от 1 мс. Выбранное время скана должно быть записано в регистр данных D8039. Выберите время скана больше чем среднее значение скана программы. Среднее значение времени скана программы хранится в ПК автоматически в регистре данных D8010.

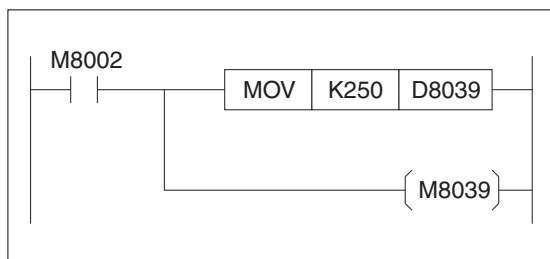


Рис. 8-3.

Определение постоянного времени цикла программы значением 250 мс в регистре данных D8039

C000194C

8.3 ФУНКЦИИ КОДА РАЗРЕШЕНИЯ РАБОТЫ

Благодаря применению кода разрешения работы хранящаяся программа ПК может быть защищена от нежелательного доступа.

В таблице 9-1 представлены 3 уровня защиты:

Кодовая буква уровня защиты	Значение	Возможный доступ				
		Проверка программы	Функции монитора	Чтение программы	Запись программы	Изменение задаваемых значений
A	Защита от любого доступа	—	—	—	—	—
B	Защита от копирования	●	●	—	—	—
C	Защита данных	●	●	●	—	●

Табл. 8-1. Уровни защиты кодов разрешения работы

Кодовое слово задается по соответствующей системе программирования. Кодовое слово состоит из кодовой буквы и семизначного шестнадцатеричного кода.

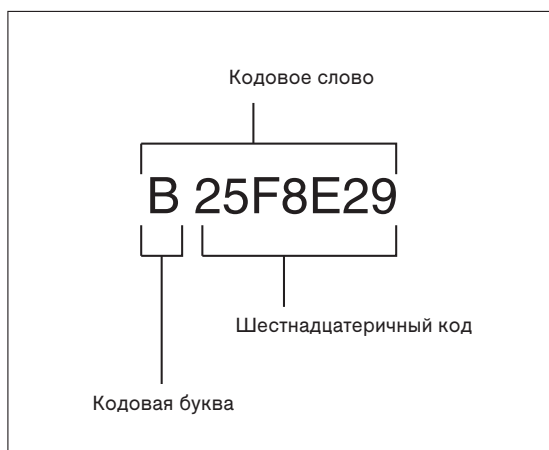


Рис. 8-4. Задание кодового слова

C000197G

УКАЗАНИЕ

Обратите внимание, что у программы ПК, кодовое слово которой не известно, нельзя выполнить никакие изменения. Программу ПК можно только полностью стереть.

Для кодового слова с уровнем защиты “А” запрещен доступ и с помощью устройства обслуживания, например, FX-20 DU. Если нужен доступ к программе с этого устройства обслуживания, выберите уровень защиты “В”.

8.4 ФУНКЦИИ ФИКСАЦИИ ИМПУЛЬСОВ

Например, сигналы фотореле. Минимальная продолжительность импульса, которую может обработать ПК, составляет 300 мкс.

Функция фиксации импульсов может использоваться только для сигналов, которые поступают в ПК только по входам X0, X1, X2, X3, X4 и X6. В каждом цикле программы может обрабатываться только 1 импульс.

При каждом пробегающем импульсе на одном из входов ПК автоматически включает специальный меркер. Этот специальный меркер может затем в программе обрабатываться дальше. Чтобы ПК мог определить новый импульс на входе, прежде в программе должен отключиться соответствующий этому входу специальный меркер.

Вход	X0	X1	X2	X3	X4	X5
Специальный меркер	M8170	M8171	M8172	M8173	M8174	M8175

Табл. 8-2. Входы и соответствующие им специальные меркеры

УКАЗАНИЕ

Функция фиксации импульсов не является высокоскоростной функцией (счетчиком). В каждом цикле программы может обрабатываться только один входной импульс.

Обратите внимание, что входы X0 до X5 не могут одновременно применяться и как входы прерывания и как функция фиксации импульсов и как входы цели для высокоскоростных счетчиков. Двойное распределение входов недопустимо.

Функции фиксации импульсов необходима активная EI-команда (FNC04).

ПРИМЕР ▾

Применение функции фиксации импульсов для счета импульсов с фотореле по входу X3 (специальный меркер M8173).

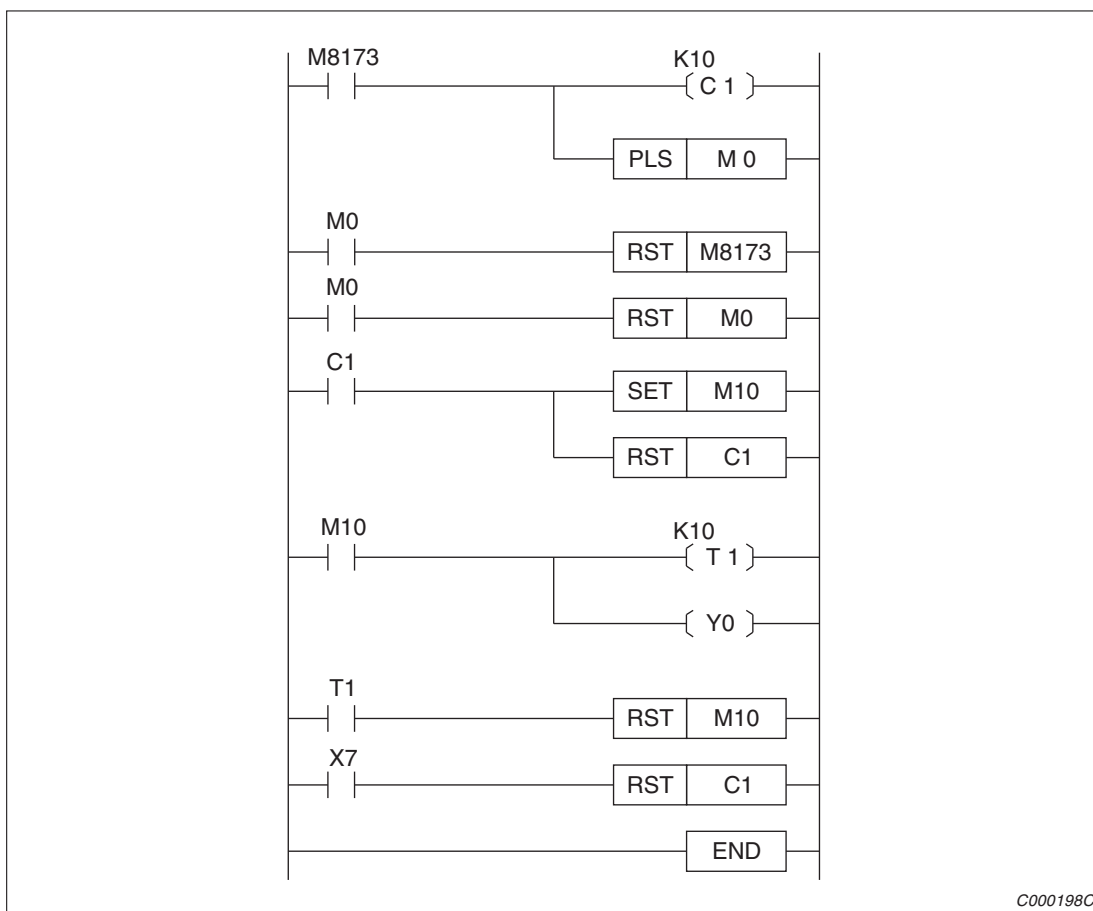


Рис. 8-5. Пример программирования Функции фиксации импульсов и счета импульсов по фотореле и по входу X3

После 10 импульсов (K10) включается выход Y0 на 1 секунду (T1/K10). Счетчик C1 сбрасывается по входу X7.



8.5 НАСТРОЙКА ВХОДНОГО ФИЛЬТРА

8.5.1 СЕРИИ FX1S И FX1N

Время замедления обработки входных сигналов определяются в ПК входными фильтрами. Время замедления устанавливается шагами в 1 мс от 0 до 15 мс. Стандартное значение составляет 10 мс.

Время фильтрации входов X0 до X7 определяется в регистре данных D8020.

При каждом переключении со СТОП-режима в РАБОТА-режим в регистр данных D8020 заносится стандартное значение в 10 мс.

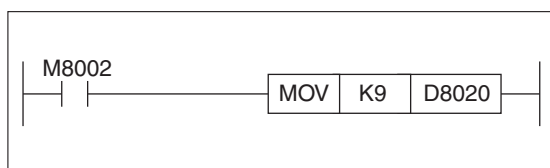


Рис. 8-6.

Установка времени замедления обработки входных сигналов X0 до X7 на значение в 9 мс в регистре данных D8020

C000199C

УКАЗАНИЕ

Время замедления может устанавливаться только блоком для входов X0 до X7.

Если занесено число "0", то время замедления устанавливается в 10 мкс.

Если в программе входы X0 до X3 определены как счетные входы для высокоскоростных счетчиков, то время замедления автоматически устанавливается на значение 50 мкс.

Если установлено очень кратковременное время замедления (≤ 5 мс), то нужно гарантировать, что входные сигналы не будут подвержены помехам. Это может привести к ошибкам при отработке программы.

8.5.2 СЕРИИ FX2N, FX2NC И FX3U

Входной фильтр устанавливается по REFF-инструкции (FNC51).

8.6 ИНТЕГРИРОВАННЫЕ ПОТЕНЦИОМЕТРЫ В FX1S И FX1N

ПК MELSEC серии FX1S и FX1N имеют аналоговые потенциометры для задания значений счета от 0 до 255.

FX1S	Потенциометр 1 (VR1)	D8030
FX1N	Потенциометр 2 (VR2)	D8031

Табл. 8-3.
Регистры данных потенциометров

ПРИМЕР ▾

100 мс- таймер с переменным временем

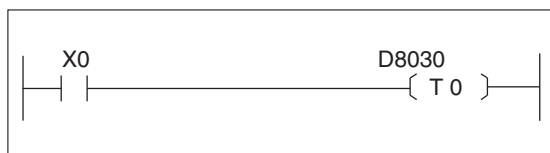


Рис. 8-7.
Таймер со значением уставки, которая может изменяться с помощью потенциометра VR1

C000196C

Если D8030 = 200, то отсчитывается время 20 с.

Если нужно время более короткое или малыми ступенями, то можно применить таймер на 10 мс.

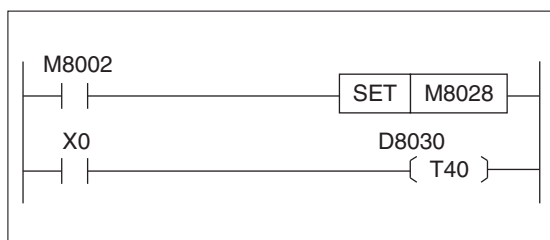


Рис. 8-8.
Таймер на 10 мс с переменным временем (FX1S)

C000197C



8.7 ФУНКЦИИ РЕАЛЬНОГО ВРЕМЕНИ

Для ПК серий FX встроены часы реального времени. Часы подпитываются батареей и продолжают работать, если отключено питающее напряжение.

ФУНКЦИИ

Подпитанные батареей часы реального времени предоставляют информацию по дате и по времени.

Года учитываются в пределах от 1980 до 2079 года.

Специальные регистры	Время	Установка
D8013	Секунды	0 - 59
D8014	Минуты	0 - 59
D8015	Часы	0 - 23
D8016	День	1 - 31
D8017	Месяц	1 - 12
D8018	Год	0 - 99 (1980 - 1999; 2000 - 2079)
D8019	День недели	0-6 (Воскресенье - Суббота)

Табл. 8-4.

Регистры для часов реального времени

ТОЧНОСТЬ

Колебания точности при применении ПК лежат в области температуры 25°C +/- 45 в течении месяца.

СПЕЦИАЛЬНЫЕ МЕРКЕРЫ ДЛЯ РАБОТЫ ЧАСОВ РЕАЛЬНОГО ВРЕМЕНИ

Специальный меркер	Значение	Описание
M8015	Установка времени	Если включен M8015, часы останавливаются. Время на часах можно задавать с помощью программатора или редактироваться. Если M8015 включится снова начинают действовать функции часов.
M8016	Хранение данных	Если включен M8016, то данные сохраняются в соответствующем регистре данных.
M8017	Округление минут	При включении M8017, то указание минут соответственно округляются в большую или меньшую сторону.
M8018	Активизация часов	M8018 включается автоматически, чтобы показать, что функции часов активизированы.
M8019	Ошибка записи	M8019 включается, если указанное значение находится вне допустимой области.

Табл. 8-5. Значение специальных меркеров

УКАЗАНИЕ

В контроллерах MELSEC семейства FX можно использовать специальные команды для настройки и опроса внутренних часов. Более подробную информацию см. в разделе 7.10.

8.8 ФАЙЛОВЫЕ РЕГИСТРЫ

Под файловыми регистрами подразумевают регистры, которые могут применяться как дополнительные, защищенные от отключения напряжения накопителя данных.

Все контроллеры серий FX1N, FX2N, FX2NC и FX3U имеют эти регистры.

Дополнительная информация находится в описании набора операндов для файловых регистров в приложении к этому руководству.

СОЗДАНИЕ ФАЙЛОВЫХ РЕГИСТРОВ

Файловые регистры записываются в блоки по соответственно 500 регистров в наборе параметров ПК. Это выполняется с помощью системы программирования, как например, GX IEC Developer.

ЧТЕНИЕ ФАЙЛОВЫХ РЕГИСТРОВ ПОСРЕДСТВОМ ПРОГРАММЫ ПК

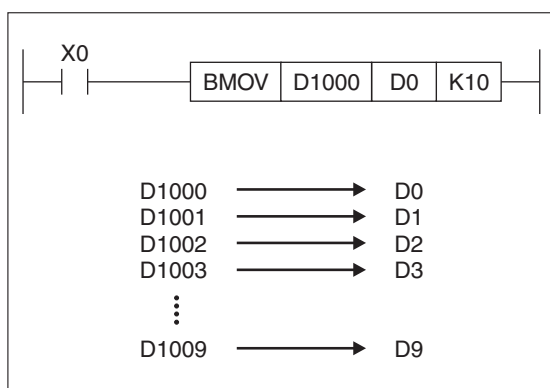


Рис. 8-9.
Чтение файловых регистров

C000195C

8.9 ПЕРЕКЛЮЧЕНИЕ РЕЖИМА РАБОТА/СТОП (RUN/STOP)

Для ПК серии FX2N и FX2NC можно параметризовать входы X0 до X17 (X0 до X7 для FX2N-16M) как клеммы РАБОТЫ (RUN-клеммы).

ПК может включаться в режим РАБОТЫ либо перемычкой между клеммами питания 24 В и РАБОТА либо переключателем РАБОТА/СТОП в режим РАБОТА.

Дополнительно имеется возможность при неподключенной RUN-клемме или при включении режима СТОП на переключателе РАБОТА/СТОП установить управление в режимы РАБОТА или СТОП по 3 специальным меркерам.

МЕРКЕР	РАБОТА	СТОП
M8035	1	0
M8036	1	0
M8037	0	1

Табл. 8-6.
Назначение меркеров

Если включен M8037, то меркеры M8035 и M8036 отключаются.

Меркеры могут включаться по системе программирования или с устройства обслуживания.

Для ПК серий FX1N, FX2N и FX2NC переключение в режим СТОП возможно также, если режим РАБОТА активизирован посредством терминала (на клемме) или переключателем РАБОТА/СТОП. Переключение происходит благодаря включению меркера M8037. После отключения меркера CPU возвращается в режим РАБОТА.

8.10 БАЗОВЫЕ АППАРАТЫ FX2N С ПИТАНИЕМ 24 В ПОСТОЯННОГО ТОКА

Чтобы гарантировать безупречную работу этого ПК, нужно ввести следующую программную строку в начале программы ПК.

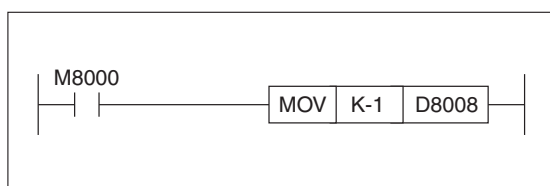


Рис. 8-10.
Пример программирования

C000191C

При питании базового прибора серии FX2N переменным током (AC) можно определить отключение напряжения посредством предварительно указанной записи желаемого отрезка времени опознания в специальный регистр D8008.

При питании модулей постоянным током (DC) этот отрезок времени опознания должен устанавливаться в 5 мс. Это достигается записью в D8008 значения -4.

Без этой записи при отключении напряжения питания постоянного тока может произойти ошибочное определение данных.

8.11 МОДУЛЬ ОТОБРАЖЕНИЯ FX1N-5DM ДЛЯ FX1S И FX1N

Модуль отображения FX1N-5DM может применяться для определения состояния операндов и для ввода данных для ПК серий FX1S и FX1N.

8.11.1 ФУНКЦИИ

ФУНКЦИИ ДЛЯ ОБСЛУЖИВАНИЯ

Следующие функции могут быть выбраны и выполнены только с помощью клавиш на модуле отображения FX1N-5DM. Более подробные указания по использованию функциональных клавиш можно найти в руководстве к модулю отображения.

ФУНКЦИЯ		ОПИСАНИЕ
Часы	Показание	Отображение часов реального времени, встроенных в FX1S/FX1N
	Положение	Настройка даты и времени
Просмотр операндов	Битовый операнд	Отображение состояния входов (X), выходов (Y), меркеров (M) и специальных меркеров (S)
	Словный операнд (16 бит)	Отображение действительного значения таймера (T), счетчика (C) и регистра (D). Установка таймера и счетчика может задаваться.
	Словный операнд (32 бита)	Отображение действительного значения 32-х битового счетчика и регистра (D)
Просмотр буферной памяти (только для FX1N)		Отображение содержания буферной памяти специальных модулей
Показ ошибок		Отображение кода ошибки и номера шага ошибки
Форсировка отображения		Принудительное включение и отключение битовых операндов (X,M,S)
Отключение таймеров и счетчиков		Стирание действительного значения таймеров и счетчиков
Изменение данных	Действительного значения	Изменение действительного значения таймеров, счетчиков и регистров
	Заданного значения	Изменение задаваемого значения таймеров и счетчиков

Табл. 8-7. Функции, которые могут быть выбраны функциональными клавишами

ФУНКЦИИ УПРАВЛЕНИЯ МОДУЛЕМ ОТОБРАЖЕНИЯ

Модуль отображения может управляться программой ПК.

ФУНКЦИИ	ОПИСАНИЕ
Защита от несанкционированного доступа	Выборочно можно по дате и времени запустить все функции, только функции просмотра или только отображение
Определение отображаемых операндов	Пользователь может определить вид и адрес операнда, который отображается
Включение и отключение отображения ошибок	Выборочно можно включить или отключить отображение кодов ошибок
Автоматическое отключение подсветки отображаемых сообщений	Установите время, после которого освещение подсветки (фон) отображения сообщения автоматически выключится
Восприятие функциональных клавиш	Может оцениваться состояние (ВКЛ или ОТКЛ) функциональных клавиш режимов отображения

Табл. 8-8. Функции, которые могут выполняться программой ПК

УКАЗАНИЕ

Если программа ПК защищена кодовым словом от несанкционированного чтения или записи, на модуле отображения показываются только дата и время. Если в этом случае выбираются другие функции, отображение мигает около 5 секунд.

8.11.2 ОПЕРАНДЫ ДЛЯ УПРАВЛЕНИЯ МОДУЛЕМ ОТОБРАЖЕНИЯ

Специальный регистр D8158 и D8159 служат для управления модулем отображения. В эти регистры в виде констант заносятся начальные адреса областей регистров или меркеров. Обе эти области (5 регистров и 15 меркеров) содержат настройку для управления модулем отображения.

Специальный регистр		Операнд для управления	Функция	Описание
Адрес	Содержание			
D8158	Kn	Dn	Установка отображаемого операнда	Вид операнда
		Dn + 1		Адрес операнда
		Dn + 2	Автоматическое отключение освещения отображения	Время до отключения освещения (минуты)
		Dn + 3	Защита от несанкционированного "Доступа"	Защитные функции
		Dn + 4	Зарезервировано	
D8159	Kn	Mn	Установка отображаемого операнда	Снятие блокировки для изменения операндов
		Mn + 1		Возвратное сообщение: Изменение совершено
		Mn + 2	Автоматическое отключение освещения отображения	Время до отключения подсветки (минуты)
		Mn + 3	Включение отображения ошибок	
		Mn + 4	Опрос функциональных клавиш модуля отображения. До тех пор пока клавиша нажата, опрашиваемый меркер включен.	Status der „ESC“-Taste
		Mn + 5		Status der „s“-Taste
		Mn + 6		Status der „r“-Taste
		Mn + 7		Status der „OK“-Taste
		Mn + 8	Зарезервированно	
		Mn + 9		
		Mn + 10		
		Mn + 11		
		Mn + 12		
		Mn + 13		
Mn + 14				

Табл. 8-9. Области регистров и меркеров для управления отображением

Если в D8158 или D8159 записано отрицательное значение или значение, которое находится вне областей регистров или меркеров, управляющая команда не выполняется. После включения управления содержание регистров D8158 и D8159 равно "-1", функции управления отключаются.

8.11.3 ОПРЕДЕЛЕНИЕ ОТОБРАЖАЕМЫХ ОПЕРАНДОВ

Благодаря занесению в первый регистр областей регистров, указанных в D8158, могут определяться операнды, которые изображаются с помощью модуля отображения.

Значение в Dn	Выбранный операнд
1	Входы (X)
2	Выходы (Y)
3	Меркеры (M)
4	Шаговые меркер (S)
5	Таймеры (T)
6	Счетчики (C), задаваемые и действительные значения 16-ти битных счетчиков и задаваемые значения 32-х битных счетчиков
7	16-ти битный регистр (D)
8	32-х битный регистр (D)
9	Дата и время
10	Счетчики (C), задаваемые и действительные значения 16-ти битных счетчиков и действительные значения 32-х битных счетчиков

Табл. 8-10.

Выбор операндов для занесения значений в Dn

УКАЗАНИЕ

Если в регистр заносится значение, которое лежит вне области от 1 до 10, то никакой операнд не может быть выбран и все функции обслуживания выполняются.

Если при определении таймера и счетчика указывается операнд, который не применен в программе, показывается операнд, адрес которого находится наиболее близко к желаемому операнду и который используется в программе. Если указанный адрес лежит вне допустимой области, то указывается операнд с наиболее высоким адресом, который применяется в программе. Если для выбранного таймера или счетчика в программе отсутствует OUT-инструкция, указывается знак "----".

С помощью клавиш "+" или "-" на модуле отображения могут выбираться друг за другом следующие операнды.

Если операнды не следуют друг за другом и несмотря на это должен выполняться выбор клавишами "+" и "-", должна быть расширена программа ПК. Адрес операнда, который должен быть отображен, заносится при этом в индекс-регистр. Воздействуя на клавиши "+" и "-" программой определяется, повышено или понижено содержание регистра и поэтому изменен адрес операнда.

Содержание области регистров переключается на "0", если отключается питающее напряжение ПК и регистр не находится в буферной области. Из-за этого больше не выбирается никакой операнд и все функции обслуживания разрешены. По этой причине выбирайте для функций управления область регистров, которая и при отключении напряжения не теряет своих данных.

8.11.4 ЗАЩИТА ОТ НЕСАНКЦИОНИРОВАННОГО “ДОСТУПА”

В четвертом регистре (Dn +3) области регистров, указанных в D8158, может заноситься значение, с помощью которого могут ограничиваться функции.

Значение в Dn + 3	Описание
0	Защита отсутствует, все функции разрешены
1	Отображаются только дата и время, дата и время могут не изменяться
2	Возможно наблюдение только за операндами, значения операндов могут не изменяться
Все другие значения	Защита отсутствует, все функции разрешены

Табл. 8-11. Защитные функции

8.11.5 ИЗМЕНЕНИЕ СОСТОЯНИЯ ИЛИ ЗНАЧЕНИЯ ОПЕРАНДОВ

Эта функция позволяет оператору изменять отображенные операнды с помощью функциональных клавиш.

Специальные регистры		Операнд управления	Описание
Адрес	Содержание		
D8159	Kn	Mn	Снятие блокировки для изменения операндов
		Mn + 1	Возвратное сообщение: Изменение совершено

Табл. 8-12. Меркеры для изменения операндов

Чтобы изменить значение операнда, должен быть включен меркер Mn. При отключении меркера Mn функция блокируется. Поэтому рекомендуется использовать SET-инструкцию. При включенном меркере Mn могут включаться или отключаться выходы (Y), меркеры (M) и шаговые меркеры (S) и изменяться задаваемые или действительные значения слов-операндов D, T и C.

Чтобы изменить бит-операнд под операндом размещается пульсирующий курсор. Воздействием на “OK”-клавишу включается или отключается операнд. С помощью “ESC”-клавиши выполняется выход из режима редактирования. Меркер (Mn + 1) исключается и Mn отключается.

После воздействия на “OK”-клавишу у словных операндов текущие значения отображаются пульсирующими. С помощью “+” и “-”-клавиш значения могут изменяться. После повторного воздействия на “OK”-клавишу значение воспринимается. Если воздействовать на “ESC”-клавишу перед “OK”-клавишей, то изменение прервется (не выполнится). После приема измененного значения с помощью клавиши “ESC” выходят из режима редактирования. Меркер (Mn + 1) исключается и Mn отключается.



Если для выбора операнда воздействие на “+” и “-”-клавиши воспримется программой ПК, то должна быть предусмотрена блокировка для случая, при которой обе кнопки изменяют состояние или значение операндов.

8.11.6 АВТОМАТИЧЕСКОЕ ОТКЛЮЧЕНИЕ ПОДСВЕТКИ ОТОБРАЖЕНИЙ

Подсветка отображения выключается автоматически после определенного времени. Это время задается в регистре. Дополнительно подсветка может постоянно включаться или отключаться.

Специальные регистры		Операнд управления	Функции	Описание
Адрес	Содержание			
D8158	Kn	Dn + 2	Время до отключения подсветки (минуты)	0 (предварительная установка): 10 минут 1...240 минут: от 1 до 240 минут > 240: 240 минут Отрицательное значение: постоянно ОТКЛ.
D8159	Kn	Mn + 2	Автоматическое отключение блокировано, подсветка включена	0: Автоматическое отключение разрешено 1: Автоматическое отключение блокировано, свет всегда включен

Табл. 8-13. Регистр и меркер для управления подсветкой

УКАЗАНИЕ

После погасания подсветки она включится вновь, как только произойдет воздействие на любую клавишу модуля отображения. Отображение покажет то же самое, что было до отключения подсветки. Это первое воздействие на клавишу служит для включения подсветки и не имеет никаких других функций.

8.11.7 ДЕБЛОКИРОВКА И БЛОКИРОВКА ОТОБРАЖЕНИЯ ОШИБОК

Посредством включения или отключения меркера (Mn + 3) можно выбрать, какое сообщение об ошибке следует отобразить.

Специальные регистры		Операнд управления	Функции	Описание
Адрес	Содержание			
D8159	Kn	Mn + 3	Включение отображения ошибок	0: Отображение ошибок технических средств ПК, синтаксиса программы и программирования 1: Дополнительное отображение ошибок конфигурации, параллельных связей и ошибок при работе

Табл. 8-14. Области регистров и меркеров для управления отображениями

При воздействии на любую клавишу модуля отображения или если ошибка больше не появляется, исчезает отображение ошибок.

Если одновременно появляется несколько ошибок, то предпочтение имеет отображение кода ошибки, которая отображается безусловно (даже если (Mn + 3) = 0). Дополнительно ошибка с самым низким кодом ошибки имеет более высокий приоритет.

9 СПЕЦИАЛЬНЫЕ МЕРКЕРЫ, СПЕЦИАЛЬНЫЕ РЕГИСТРЫ

9.1 СПЕЦИАЛЬНЫЕ МЕРКЕРЫ (M8000...M8467)

Благодаря применению специальных меркеров могут опрашиваться определенные состояния работы ПК в программе ПК или включаться или отключаться.

Специальные меркеры делятся на две группы:

- ❶ Специальные меркеры, у которых в программе ПК могут опрашиваться только состояния сигналов контактных инструкций (например, LD- или LDI-инструкции).
- ❷ Специальные меркеры, которые в программе ПК с помощью инструкций могут непосредственно включаться или отключаться.

УКАЗАНИЕ

Специальные меркеры, которые отображают ошибки технических средств или отработки программы, описаны в следующей главе 10.

9.1.1 СОСТОЯНИЕ ПК (M8000–M8009)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение	
M8000	●	–	FX1S FX1N FX2N FX2NC FX3U	Отобразить состояние ПК: RUN (РАБОТА)	
M8001	●	–		Отобразить состояние ПК: RUN (РАБОТА)	
M8002	●	–		Импульс инициализации	
M8003	●	–		Импульс инициализации	
M8004	●	–		Ошибки ПК	
M8005	●	–	FX2N FX2NC FX3U	Включение меркера, если напряжение батареи, значение которого задано в D8006, понижено.	
M8006	●	–		Запоминание ошибки "Низкое напряжение батареи".	
M8007	●	–		Включается при кратковременном отключении напряжения	
M8008	●	–		Сообщение об отключении напряжения	
M8009	●	–		Сообщение об отключении напряжения питания 24 V DC	

Табл. 9-1. Специальные меркеры состояния ПК

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК включается или отключается непосредственно по инструкции.

УКАЗАНИЕ

Описание специальных меркеров M8005...M8009 можно получить в соответствующем руководстве (описание технических средств) по CPU.

9.1.2 ВРЕМЕННЫЕ ТАКТЫ (M8011...M8019)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8010	—	—	—	Зарезервировано
M8011	●	—	FX1S FX1N FX2N FX2NC	Датчик тактов: 10 мс. Один такт составляет 10 мс.
M8012	●	—		Датчик тактов: 100 мс. Один такт составляет 100 мс.
M8013	●	—		Датчик тактов: 1 с. Один такт составляет 1 секунду.
M8014	●	—		Датчик тактов: 1 мин. Один такт составляет 1 минуту.
M8015	●	●		Установка времени. Если меркер включен, часы останавливаются. Часы пойдут снова, когда меркер отключится.
M8016	●	—		Регистр данных. Если меркер включен, содержимое D8013...D8019 сохраняется, но часы продолжают идти.
M8017	●	●		Округление минут. При импульсном сигнале часы реального времени (RTC) округляются на полном значении минут.
M8018	●	—		Часы реального времени активны. Как только меркер включится активизируются часы реального времени.
M8019	●	—		Установка ошибки. Установка данных времени выполнена вне разрешенной области.

Табл. 9-2. Специальные меркеры для временных тактов и для реального времени

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

9.1.3 ФЛАГИ (M8020...M8029)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8020	●	—	FX1S FX1N FX2N FX2NC	Флаг нуля (zero) включается, если результат сложения или вычитания равен нулю.
M8021	●	—		Флаг заема (borrow) включается, если результат вычитания (FNC21) меньше самого малого отрицательного значения.
M8022	●	—		Флаг переноса (carry) включается при передаче значения числа, при суммировании или при передаче данных, при выполнении инструкции сдвига.
M8024	●	●	FX2N FX2NC	Если используется с BMOV-то: M8024 = 0: передача из (S+) в (D+) M8024 = 1: передача из (D+) в (S+)
M8025	—	●		При включении меркера обрабатываются все HSC-инструкции (FNC 53...55), как только активизируются внешние входы HSC-возврата.
M8026	—	—		RAMP-инструкция (FNC 67) останавливается.
M8027	—	—		Строка данных для 16 элементов в PR-инструкции (FNC 77).
M8028	—	—	FX1S	Использование 10ms-таймеров с T32 до T62 разрешенно
	—	—	FX2N (начиная с 3.0) FX2NC (начиная с 3.0)	M8028 = 0: КОМАНДЫ FROM/TO не прерываются при активизации программы прерывания M8028 = 1: КОМАНДЫ FROM/TO будут прерваны при активизации программы прерывания, а после завершения прерывания, будут продолжены
M8029	●	—	FX1S/FX1N FX2N/FX2NC	Инструкция полностью отработана.

Табл. 9-3. Специальные меркеры для флагов

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

9.1.4 РЕЖИМЫ ПК (M8030...M8039)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8030	●	—	FX2N/FX2NC	Напряжение батареи слишком низко, хотя светодиод батареи (BATT.LED) не светится.
M8031	●	●	FX1S/FX1N FX2N/FX2NC	Все операнды отключены, значения данных которых НЕ хранятся в неразрушающейся памяти
M8032	●	●		Все операнды отключены, значения данных которых хранятся в неразрушающейся памяти
M8033	●	●		Содержание значений данных режима СТОП. Содержание регистра отображения и память данных сохраняются, если ПК переключается из режима РАБОТА (RUN) в режим СТОП.
M8034	●	●		Препятствие выходам. Все выходы заблокированы, однако программа обрабатывается дальше.
M8035	●	●		Режимы РАБОТА/СТОП принудительно включены.
M8036	●	●		Принудительный режим РАБОТА благодаря включению в программе ПК специального меркера.
M8037	●	●		Используя режим принудительного управления (M8035 включен), можно выполнять операции РАБОТА/СТОП или импульсный РАБОТА/СТОП.
M8038	—	●		Установка устройства для сети n-zu-n.
M8039	●	●		Программа ПК с постоянным циклом программы. Если включен M8039, то ПК обрабатывает с постоянным циклом программы, значение которого записано в D8039.

Табл. 9-4. Специальные меркеры режимов ПК

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

9.1.5 STL-СОСТОЯНИЯ (M8040...M8049)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8040	●	●	FX1S FX1N FX2N FX2NC FX3U	Продолжение работы невозможно (нет условия продолжения работы). Если M8040 включен, продолжение работы невозможно.
M8041	●	●		Начало условия продолжения работы. Условие продолжения работы возможно во время автоматической работы.
M8042	●	●		Стартовый импульс. При соответствующих входных условиях задается стартовый импульс.
M8043	●	●		Выполняется возврат в исходную позицию. При достижении исходной позиции включается M8043.
M8044	●	●		Включается M8044 - условие возврата в исходную позицию, если исходная позиция опознана.
M8045	●	●		Отключение всех выходов невозможно. Если включается M8045, то отключение всех выходов невозможно.
M8046	●	—		Включен статус STL. M8046 включается, если операнды состояния шагов от S0 до S899 и M8047 включены.
M8047	●	●		Указание статуса STL. Если M8047 включен, то в регистре данных D8040 до D8047 состояния шагов отображаются первые 8 операндов состояния шагов.
M8048	●	—	FX2N FX2NC FX3U	Указан меркер ошибки. Меркер включен, если активизирован контроль меркера ошибки (M8049) и включен меркер ошибки.
M8049	—	●		Отображение меркера ошибки возможно. При включенном меркере может по D8049 активизироваться отображение меркера ошибки.

Табл. 9-5: Специальные меркеры для STL-Состояния

* Эти маркеры сбрасываются при останове контроллера.

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

9.1.6 ПРОГРАММА ПРЕРЫВАНИЯ (M8050...M8059)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8050	●	●	FX1S FX1N FX2N FX2NC FX3U	Программа прерывания I00** не выполняется ③
M8051	●	●		Программа прерывания I10** не выполняется ③
M8052	●	●		Программа прерывания I20** не выполняется ③
M8053	●	●		Программа прерывания I30** не выполняется ③
M8054	●	●		Программа прерывания I40** не выполняется ③
M8055	●	●		Программа прерывания I50** не выполняется ③
M8056	●	●	FX2N FX2NC FX3U	Программа прерывания I6** не выполняется ③
M8057	●	●		Программа прерывания I7** не выполняется ③
M8058	●	●		Программа прерывания I8** не выполняется ③
M8059	●	●		Отключено прерывание I010 до I060

Табл. 9-6. Специальные меркеры для программы прерывания

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.
- ③ Если в программе пользователя выполняется EI-инструкция (FNC 04), то все прерывания разрешены до тех пор, пока включен один из выполняемых специальных меркеров. В этом случае для каждого включенного специального меркера блокируется относящееся к нему прерывание, т.е. оно не может быть активизировано.

9.1.7 СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ СООБЩЕНИЙ О НЕИСПРАВНОСТИ (M8060-M8069)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8060	●	—	FX2N/FX2NC FX3U	Ошибка конфигурации ввода-вывода
M8061	●	—	FX1S/FX1N FX2N/FX2NC FX3U	Аппаратная неисправность контроллера
M8062	●	—	FX2N/FX2NC	Ошибка коммуникации между контроллером и программатором
M8063 ^③	●	—	FX1S FX1N FX2N FX2NC FX3U	Ошибка при последовательной коммуникации
M8064	●	—		Ошибка параметра
M8065	●	—		Синтаксическая ошибка программы
M8066	●	—		Ошибка программирования
M8067 ^④	●	—		Ошибка выполнения
M8068	—	●		Ошибка выполнения (сохраненная)
M8069	—	●		FX2N FX2NC FX3U

Табл. 9-7: Специальные маркеры для индикации ошибок

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.
- ③ У контроллеров серий FX1S, FX1N, FX2N и FX2NC этот маркер сбрасывается при переключении контроллера из режима "STOP" в режим "RUN". У контроллеров серии FX3U маркер M8063 в этом случае не сбрасывается.
О сбое при последовательной коммуникации по каналу 2 в контроллере FX3U сигнализирует маркер M8438.
- ④ Этот маркер сбрасывается, если контроллер переключается из режима "STOP" в режим "RUN".
- ⑤ После установки M8069 выполняется проверка шины ввода-вывода. Если при этом распознается неисправность, в специальный регистр D8069 записывается код ошибки 6130 и устанавливается специальный маркер M8061.

9.1.8 ФУНКЦИЯ ПЕРЕХВАТА ИМПУЛЬСОВ

Если на входах с X0 по X7 распознается импульсный сигнал, устанавливается один из указанных здесь маркеров. После сброса специального маркера маркер снова автоматически устанавливается при следующем импульсном сигнале. Таким образом, возможно распознавание и сохранение быстрых входных импульсов.

В контроллерах серий FX2N, FX2NC и FX3U для использования функции перехвата импульсов должны быть деблокированы прерывания (с помощью команды EI). В контроллерах серий FX1S и FX1N в этом нет необходимости.

При переходе контроллера из режима „STOP“ в режим „RUN“ специальные маркеры с M8170 по M8177 сбрасываются.

Специальные маркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8170	●	–	FX1S FX1N FX2N FX2NC FX3U	Фиксация импульса X0
M8171	●	–		Фиксация импульса X1
M8172	●	–		Фиксация импульса X2
M8173	●	–		Фиксация импульса X3
M8174	●	–		Фиксация импульса X4
M8175	●	–		Фиксация импульса X5
M8176	●	–	FX3U	Фиксация импульса X6
M8177	●	–		Фиксация импульса X7

Табл. 9-8. Специальные маркеры для функции фиксации импульса

- ① Специальный маркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный маркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

9.1.9 ФУНКЦИЯ СВЯЗИ И СПЕЦИАЛЬНЫЕ МЕРКЕРЫ (M8070–M8191)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определе-ние состоя-ния сигналов	CPU	Значение
M8070	●	●	FX1S FX1N FX2N	Меркер включается, если в ПК Мастер-станция включена в работу по параллельной связи
M8071	●	●		Меркер включается, если в ПК Подчиненная-станция включена в работу по параллельной связи
M8072	●	—		Определение работы по параллельной связи ПК
M8073	●	—		Опознание, что M8070 или M8071 в работе по параллельной связи неправильно включены
M8074	Зарезервировано			
M8075	—	—	FX2N FX2NC	При выполнении Sampling Trace эти биты используются контроллером
M8076	—	—		Активен при выполнении Sampling Trace
M8077	●	—		Активен при завершении Sampling Trace
M8078	●	—		При выполнении Sampling Trace эти биты используются контроллером
M8079	—	—		При выполнении Sampling Trace эти биты используются контроллером
M8090	●	—	FX3U	При команде ВКСМР (FNC194 ... FNC199) все сравнения дают результат "истинно".
M8091	●	●		Содержимое последнего знака в команде COMRD или BINDA
M8099	●	●	FX2N/FX2NC FX3U	Активация кольцевого счетчика *
M8104	●	—	FX2N/FX2NC	Расширение памяти установлено (начиная с версии 3)
M8105	●	—	FX3U	Происходит запись во флэш-EPROM.
M8107	●	—		Проверяется наличие комментария к операнду.
M8109	●	—	FX2N/FX2NC FX3U	Ошибка активизированного выхода
M8112	●	●	FX1S/FX1N	Расширительный адаптер FX1N-4EX-BD: вход ВХ0
				Расширительный адаптер FX1N-2AD-BD: изменить режим входа для канала 1
				Расширительный адаптер FX1N-1DA-BD: изменить режим выхода
M8113	●	●		Расширительный адаптер FX1N-4EX-BD: вход ВХ1
				Расширительный адаптер FX1N-2AD-BD: изменить режим входа для канала 2
M8114	●	●		Расширительный адаптер FX1N-4EX-BD: вход ВХ2
M8115	●	●		Расширительный адаптер FX1N-4EX-BD: вход ВХ3
M8116	●	●		Расширительный адаптер FX1N-2EYT-BD: выход ВУ0
M8117	●	●		Расширительный адаптер FX1N-2EYT-BD: выход ВУ1

Табл. 9-9. Специальные меркеры для связи и специальные (1)

* В контроллерах FX2N или FX2NC кольцевой счетчик (специальный регистр D8099) активируется в конце программного цикла, в котором был установлен M8099. В контроллерах FX3U кольцевой счетчик активируется сразу после установки M8099.

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определе-ние состоя-ния сигналов	CPU	Значение
M8121	●	—	FX1S FX1N FX2N FX2NC FX3U	Замедление передачи данных RS
M8122	●	●		Флаг передачи данных RS
M8123	●	●		Объем данных окончен RS
M8124	●	—		Опознан carrier-флаг RS
M8126	●	—		Флаг RS485
M8127	●	—		Затребование флага Handshake (RS485)
M8128	●	●		Ошибка затребованного флага (RS485)
M8129	●	●		Затребование флага байта/слова: включен = байт, не включен = слово (RS485)
M8130	●	●	FX2N FX2NC FX3U	Выбор таблицы сравнения, которая применяется с HSZ-инструкцией
M8131	●	—		Опознание окончания обработки HSZ-сравнения
M8132	●	●		Определение значения PLSY-инструкции с помощью таблицы HSZ-сравнения
M8133	●	—		Опознание окончания обработки HSZ-сравнения (при применении PLSY-инструкции)
M8138	●	—	FX3U	Обработка команды DHSCT окончена
M8139	●	—		Выполняется команда для высокоскоростного счетчика (DHSCS, DHSCR, DHSZ, DHSCT)
M8140	●	—	FX1S/FX1N	Очищает импульсный выход в инструкции FNC(156)
M8145	—	●	FX1S	Команда останова импульсного выхода Y000
M8146	—	●	FX1N	Команда останова импульсного выхода Y001
M8147	●	—	FX1S FX1N	Отображение состояния импульсного выхода Y000 (Занят/Готов)
M8148	●	—		Отображение состояния импульсного выхода Y001 (Занят/Готов)
M8151	●	—	FX3U	Обмен данными с преобразователем частоты по каналу 1
M8152	●	—		Ошибка при соединении с преобразователем частоты (канал 1)
M8153	●	—		Ошибка при соединении с преобразователем частоты по каналу 1 (сохраненная ошибка)
M8154	●	—	FX3U	Ошибка при выполнении команды IVBWR (канал 1)
		—	FX2N/FX2NC (начиная с 3.00)	Время ожидания для соединения (команда EXTR)
M8155	●	—	FX2N/FX2NC (начиная с 3.00)	Происходит соединение (команда EXTR)
M8156	●	—	FX3U	Обмен данными с преобразователем частоты по каналу 2
		—	FX2N/FX2NC (начиная с 3.00)	Ошибка соединения или параметра (команда EXTR)
M8157	●	—	FX3U	Ошибка соединения с преобразователем частоты (канал 2)
		—	FX2N/FX2NC (начиная с 3.00)	Сохраненная ошибка соединения (команда EXTR)
M8158	●	—	FX3U	Ошибка соединения с преобразователем частоты по каналу 1 (сохраненная ошибка)
M8159	●	—		Ошибка при выполнении команды IVBWR (канал 2)
M8160	●	●	FX2N/FX2NC FX3U	XCH-инструкция как функция обмена байтами

Табл. 9-10: Специальные меркеры для связи и специальные регистры (2)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определе-ние состоя-ния сигналов	CPU	Значение
M8161	●	●	FX1S/FX1N	8-битный режим (RS, ASC, ASCII, HEX, CCD, CRC)
M8162	●	●	FX2N/FX2NC FX3U	Высокоскоростной режим параллельного обхода (32-х битный для каждого направления движения)
M8164	●	●	FX2N/FX2NC (начиная с 2.00)	Если включен, значение в D8164 используется как число точек обмена для FROM/TO. (начиная с версии CPU 2.00 FX2N/2NC)
M8165	●	●	FX3U	При команде SORT2 сортировать в убывающей последовательности.
M8167	●	●	FX2N/FX2NC	Шестнадцатеричный формат для НКУ-инструкции
M8168	●	●	FX3U	Шестнадцатеричный формат для SMOV-инструкции
M8170 - M8177	●	●	FX1S/FX1N FX2N/FX2NC FX3U	Флаги перехвата импульсов для входов с X0 по X5 (см. раздел 9.1.8)
M8178	●	●	FX3U	Выбор канала при параллельном соединении (M8178 = 0: канал 1, M8178 = 1: канал 2)
M8179	●	●		Выбор канала при сети типа n:n (M8179 = 0: канал 1, M8179 = 1: канал 2)
M8183 M504 В FX1S	●	—	FX1S FX1N FX2N (начиная с 2.00) FX2NC (начиная с 2.00) FX3U	Включен при возникновении ошибки в мастер-станции
M8184 M505 В FX1S	●	—		Включен при возникновении ошибки в 1-ой слэив-станции
M8185 M506 В FX1S	●	—		Включен при возникновении ошибки в 2-ой слэив-станции
M8186 M507 В FX1S	●	—		Включен при возникновении ошибки в 3-ей слэив-станции
M8187 M508 В FX1S	●	—		Включен при возникновении ошибки в 4-ой слэив-станции
M8188 M509 В FX1S	●	—		Включен при возникновении ошибки в 5-ой слэив-станции
M8189 M510 В FX1S	●	—		Включен при возникновении ошибки в 6-ой слэив-станции
M8190 M511 В FX1S	●	—		Включен при возникновении ошибки в 7-ой слэив-станции
M8191 M503 В FX1S	●	—		Включен при связи с другими станциями
M8198	●	●	FX3U	C251, C252, C254: засчитывать каждый фронт фаз А и В
M8199	●	●		C253, C255: засчитывать каждый фронт фаз А и В

Табл. 9-11. Специальные меркеры для связи и специальные (3)

- ① специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

УКАЗАНИЕ

Указание по специальным меркерам M8060 до M8069 можно прочесть также в главе 11 (Ошибки программирования).

В серии FX1S меркеры с M504 до M511 зарезервированны для работы по сети n:n. Эти меркеры не могут использоваться для других функций.

9.1.10 СЧЕТЧИКИ СЧЕТА ВВЕРХ-ВНИЗ (M8200...M8254)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8200	–	●	FX1N FX2N FX2NC FX3U	Если включается один из рядом стоящих специальных меркеров, принадлежащий к нему счетчик определяется как счетчик счета вверх (M8200 = C200, M8201 = C201 и т.д.). Счетчик считает вверх столько, пока не включится относящийся к нему специальный меркер.
M8201	–	●		
M8203	–	●		
M8204	–	●		
M8205	–	●		
M8206	–	●		
M8207	–	●		
M8208	–	●		
M8209	–	●		
M8210	–	●		
M8211	–	●		
M8212	–	●		
M8213	–	●		
M8214	–	●		
M8215	–	●		
M8216	–	●		
M8217	–	●		
M8218	–	●		
M8219	–	●		
M8220	–	●		
M8221	–	●		
M8222	–	●		
M8223	–	●		
M8224	–	●		
M8225	–	●		
M8226	–	●		
M8227	–	●		
M8228	–	●		
M8229	–	●		
M8230	–	●		
M8231	–	●		
M8232	–	●		
M8233	–	●		
M8234	–	●		

Табл. 9-12. Специальные меркеры для счетчиков вверх/вниз

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

ОДНОФАЗНЫЙ СЧЕТЧИК СО СЧЕТНЫМ ВХОДОМ (M8235...M8245)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8235	●	●	FX1S FX1N FX2N FX2NC FX3U	Если включен M8235, C235 считает вверх
M8236	●	●		Если включен M8236, C236 считает вверх
M8237	●	●		Если включен M8237, C237 считает вверх
M8238	●	●		Если включен M8238, C238 считает вверх
M8239	●	●		Если включен M8239, C239 считает вверх
M8240	●	●		Если включен M8240, C240 считает вверх
M8241	●	●		Если включен M8241, C241 считает вверх
M8242	●	●		Если включен M8242, C242 считает вверх
M8243	●	●		Если включен M8243, C243 считает вверх
M8244	●	●		Если включен M8244, C244 считает вверх
M8245	●	●		Если включен M8245, C245 считает вверх

Табл. 9-13: Специальные меркеры для 1-но фазных счетчиков со счетным входом**ДВУХФАЗНЫЙ СЧЕТЧИК С ДВУМЯ СЧЕТНЫМИ ВХОДАМИ (M8246–M8250)**

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8246	●	–	FX1S FX1N FX2N FX2NC FX3U	Если M8246 включен, C246 считает вниз. Если M8246 отключен, C246 считает вверх.
M8247	●	–		Если M8247 включен, C247 считает вниз. Если M8247 отключен, C247 считает вверх.
M8248	●	–		Если M8248 включен, C248 считает вниз. Если M8248 отключен, C248 считает вверх.
M8249	●	–		Если M8249 включен, C249 считает вниз. Если M8249 отключен, C249 считает вверх.
M8250	●	–		Если M8250 включен, C250 считает вниз. Если M8250 отключен, C250 считает вверх.

Табл. 9-14: Специальные меркеры для двухфазных счетчиков с двумя счетными входами

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

А/В-ФАЗНЫЙ СЧЕТЧИК С ДВУМЯ СЧЕТНЫМИ ВХОДАМИ (M8251–M8255)

Адреса операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	CPU	Значение
M8251	●	–	FX1S FX1N FX2N FX2NC FX3U	Если M8251 включен, C251 считает вниз. Если M8251 отключен, C251 считает вверх.
M8252	●	–		Если M8252 включен, C252 считает вниз. Если M8252 отключен, C252 считает вверх.
M8253	●	–		Если M8253 включен, C253 считает вниз. Если M8253 отключен, C253 считает вверх.
M8254	●	–		Если M8254 включен, C254 считает вниз. Если M8254 отключен, C254 считает вверх.
M8255	●	–		Если M8255 включен, C255 считает вниз. Если M8255 отключен, C255 считает вверх.

Табл. 9-15: Специальные меркеры для А/В-фазных счетчиков с двумя счетными входами

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК может включаться или отключаться непосредственно по инструкции.

9.1.11**СПЕЦИАЛЬНЫЕ АНАЛОГОВЫЕ МОДУЛИ (M8260–M8299)**

Адреса операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	CPU	Значение
M8260 ... M8269	●	–	FX3U	специальный маркер для 1-го специального модуля
M8270 ... M8279	●	–		специальный маркер для 2-го специального модуля
M8280 ... M8289	●	–		специальный маркер для 3-го специального модуля
M8290 ... M8299	●	–		специальный маркер для 4-го специального модуля

Табл. 9-16: Специальные маркеры для подключенных аналоговых модулей

- ① Специальные маркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI).
- ② Специальные маркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды.

УКАЗАНИЕ

Функция специальных маркеров M8260...M8299 описана в руководствах по аналоговым модулям.

9.1.12 СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ДИАГНОСТИКИ НЕИСПРАВНОСТЕЙ (M8316-M8329)

Адреса операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	CPU	Значение
M8316	●	–	FX3U	Обращение к несуществующему адресу ввода-вывода. M8316 устанавливается, если соответствующий адрес ввода-вывода используется непосредственно (например, в команде LD, AND, OR или OUT) или если обращение к несуществующему входу или выходу произошло косвенно, через индексный регистр.
M8318	●	–		Маркер ошибки при инициализации буферной памяти (маркер M8318) устанавливается, если возникла ошибка при доступе к буферной памяти специального модуля с помощью команды FROM или TO. В этом случае регистр D8318 содержит номер специального модуля, а D8319 - адрес в буферной памяти.
M8328	●	–		Команда не может быть выполнена.
M8329	●	–		При выполнении команды возникла ошибка.

Табл. 9-17:Дополнительные специальные маркеры для диагностики неисправностей в контроллере FX3U

- ① специальные маркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI).
- ② специальные маркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды.

9.1.13 СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ВЫВОДА ИМПУЛЬСОВ И ПОЗИЦИОНИРОВАНИЯ (M8330-M8467)

Адреса операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	CPU	Значение			
M8331	●	—	FX3U	Команда DUTY	выходной сигнал 1		
M8332	●	—			выходной сигнал 2		
M8333	●	—			выходной сигнал 3		
M8336*	●	●		Команда DVIT	выходной сигнал 4		
M8338	●	●		Команда PLSV	деблокировка прерывания		
M8340	●	—	FX3U	Выход Y000	разгон/замедление		
M8341*	●	●			деблокировка сброса выхода		
M8342*	●	●			форма сигнала выключателя приближения к базовой точке (размыкающий / замыкающий контакт)		
M8343	●	●			ограничение при движении вперед		
M8344	●	●			ограничение при движении назад		
M8345*	●	●			форма сигнала выключателя приближения к базовой точке (размыкающий / замыкающий контакт)		
M8346*	●	●			форма сигнала нулевой точки (размыкающий / замыкающий контакт)		
M8347*	●	●			форма сигнала прерывания (размыкающий / замыкающий контакт)		
M8348	●	—			действует команда позиционирования		
M8349*	●	●			останов вывода импульсов		
M8350	●	—			FX3U	Выход Y001	контроль импульса (0: готов, 1: активен)
M8351*	●	●					деблокировка сброса выхода
M8352*	●	●					направление вращения для движения референцирования
M8353	●	●	ограничение при движении вперед				
M8354	●	●	ограничение при движении назад				
M8355*	●	●	форма сигнала выключателя приближения к базовой точке (размыкающий / замыкающий контакт)				
M8356*	●	●	форма сигнала нулевой точки (размыкающий / замыкающий контакт)				
M8357*	●	●	форма сигнала прерывания (размыкающий / замыкающий контакт)				
M8358	●	—	действует команда позиционирования				
M8359*	●	●	останов вывода импульсов				

Табл. 9-18: Специальные маркеры для вывода сигналов и позиционирования (1)

* эти маркеры сбрасываются при STOP контроллера

- ① специальные маркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI).
- ② специальные маркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды.

Адреса операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	CPU	Значение	
M8360	●	—	FX3U	Выход Y002	контроль импульса (0: готов, 1: активен)
M8361*	●	●			деблокировка сброса выхода
M8362*	●	●			направление вращения для движения референцирования
M8363	●	●			ограничение при движении вперед
M8364	●	●			ограничение при движении назад
M8365*	●	●			форма сигнала выключателя приближения к базовой точке (размыкающий / замыкающий контакт)
M8366*	●	●			форма сигнала нулевой точки (размыкающий / замыкающий контакт)
M8367*	●	●			форма сигнала прерывания (размыкающий / замыкающий контакт)
M8368	●	—			действует команда позиционирования
M8369*	●	●			останов вывода импульсов
M8370	●	—			FX3U
M8371*	●	●	деблокировка сброса выхода		
M8372*	●	●	направление вращения для движения референцирования		
M8373	●	●	ограничение при движении вперед		
M8374	●	●	ограничение при движении назад		
M8375*	●	●	форма сигнала выключателя приближения к базовой точке (размыкающий / замыкающий контакт)		
M8376*	●	●	форма сигнала нулевой точки (размыкающий / замыкающий контакт)		
M8377*	●	●	форма сигнала прерывания (размыкающий / замыкающий контакт)		
M8378	●	—	действует команда позиционирования		
M8379*	●	●	останов вывода импульсов		
M8460	●	●	FX3U	Команда DVIT: пользовательские прерывания	
M8461	●	●			выход Y001
M8462	●	●			выход Y002
M8463	●	●			выход Y003
M8464	●	●	FX3U	Команда DSZR и ZRN: деблокировка сигнала сброса	выход Y000
M8465	●	●			выход Y001
M8466	●	●			выход Y002
M8467	●	●			выход Y003

Табл. 9-19: Специальные маркеры для вывода сигналов и позиционирования (2)

* эти маркеры сбрасываются при STOP контроллера.

- ① специальные маркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI).
- ② специальные маркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды.

9.1.14 СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА (M8380-M8392)

Адрес операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	Процессор	Значение
M8380*	●	–	FX3U	состояние счетчиков C235, C241, C244, C246, C247, C249, C251, C252 и C254
M8381*	●	–		состояние счетчика C236
M8382*	●	–		состояние C237, C242 и C245
M8383*	●	–		состояние C238, C248, C248 (OP), C250, C253 и C255
M8384*	●	–		состояние C239 и C243
M8385*	●	–		состояние C240
M8386*	●	–		состояние C244 (OP)
M8387*	●	–		состояние C245 (OP)
M8388	●	–		изменение функции высокоскоростного счетчика
M8389	●	●		форма сигнала внешнего сбрасывающего входа (размыкающий / замыкающий контакт)
M8390	●	●		переключение функций для C244
M8391	●	●		переключение функций для C245
M8392	●	●		переключение функций для C248 и C253

Табл. 9-20: Специальные меркеры режимов ПК

* эти меркеры сбрасываются при STOP контроллера.

- ① Специальные меркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI).
- ② специальные меркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды.

9.1.15 СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ ПРОГРАММ ПРЕРЫВАНИЯ (M8393 И M8394)

Адрес операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	Процессор	Значение
M8393*	●	–	FX3U	настройка времени задержки
M8394*	●	–		операнд для вызова программы прерывания с помощью команды DHCMOV

Табл. 9-21: Специальные меркеры для программ прерывания

* Эти меркеры сбрасываются при STOP контроллера.

- ① специальные меркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI).
- ② специальные меркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды.

9.1.16 СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ КОЛЬЦЕВОГО СЧЕТЧИКА (M8398)

Адрес операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	Процессор	Значение
M8398	●	●	FX3U	Активация кольцевого счетчика (32 бита, 1 мс) Кольцевой счетчик (D8399 и D8398) активируется после установки M8398.

Табл. 9-22: Специальные маркеры для кольцевого счетчика

- ① специальные маркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI).
- ② специальные маркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды.

9.1.17 СПЕЦИАЛЬНЫЕ МАРКЕРЫ ДЛЯ КОММУНИКАЦИЙ (M8401-M8449)

Адрес операнда специального маркера	① Опрос состояния сигнала	② Установка состояния сигнала	Процессор	Значение	
M8401	●	–	FX3U	Команда RS2 (канал 1)	ожидание передачи
M8402*	●	●			запрос передачи
M8403*	●	●			прием завершен
M8404	●	–			несущий сигнал распознан
M8409	●	●			контрольное время истекло
M8421	●	–	FX3U	Команда RS2 (канал 2)	ожидание передачи
M8422*	●	●			запрос передачи
M8423*	●	●			прием завершен
M8424	●	–			несущий сигнал распознан
M8426	●	–		Сеть RS485 (канал 2)	глобальный режим
M8427	●	–			передача данных по запросу
M8428	●	●			ошибка при передаче данных по запросу
M8429	●	●	FX3U	Команда RS2 (канал 2)	контрольное время истекло (RS2)
				Сеть RS485 (канал 2)	переключение между "байтом" и "словом" при передаче данных по запросу
M8438	●	●	FX3U		ошибка при последовательной коммуникации по каналу 2
M8449	●	–			ошибка специального модуля

Табл. 9-23: Специальные маркеры для команд RS2, сети RS485 и диагностики ошибок

* эти маркеры сбрасываются, если контроллер остановлен или команда RS2 более не выполняется

- ① специальные маркеры, у которых в программе контроллера можно только опросить состояние сигнала (с помощью команды контакта, например, LD или LDI)
- ② специальные маркеры, которые в программе контроллера могут устанавливаться или сбрасываться непосредственно, с помощью команды

9.2 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ (D8000–D8511)

В специальных регистрах сохраняются значения данных по ПК рабочим состояниям. Значения данных могут программой ПК считываться, а также изменяться.

Специальные регистры можно разделить на две группы:

- ① Специальные регистры, значения данных которых программой ПК могут только считываться.
- ② Специальные регистры, значения данных которых программой ПК могут считываться и изменяться.

9.2.1 СОСТОЯНИЕ ПК (D8000–D8009)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8000	●	●	FX1S FX1N FX2N FX2NC FX3U	Таймер контроля времени цикла (Watch-Dog) настраивается в единицах 1 мс. Стандартное значение 200 мс
D8001	●	–		Номер версии FX1S: 22V _W FX1N: 26V _W FX2N: 24V _W (Например, FX1N версия 1.00 → 26100)
D8002	●	–		Емкость памяти: 0002 ? 2 тысячи шагов (только у FX1S) 0004 ? 4 тысячи шагов (FX2N/FX2NC) 0008 ? 8 тысяч или более шагов (кроме FX1S) При более 16 тысяч шагов в D8002 записывается значение "8", а в D8102 - значение "16" или "64".
D8003	●	–		Тип памяти: 00 _H → RAM (optional) 01 _H → EPROM (optional) 02 _H → EEPROM (optional) 0A _H → EEPROM (optional, защита записи) 10 _H → CPU-память
D8004	●	–		Адрес меркера ошибок
D8005	–	–		Напряжение батареи: например, значение 36 означает 3,6 В
D8006	–	–		Память напряжения батареи, при котором должна опознаться ошибка "низкое напряжение батареи"
D8007	–	–		Количество кратковременных отключений напряжения
D8008	–	–		Память времени задержки при отключении напряжения до отключения CPU (стандартно 10 мс)
D8009	–	–	Память самого низкого адреса аппарата, который касается отключению напряжения 24 В постоянного тока.	

Табл. 9-24: Специальные регистры состояния ПК

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.2 ТАКТЫ ВРЕМЕНИ (D8010...D8019)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8010	●	–	FX1S FX1N FX2N FX2NC FX3U	Текущее время отработки цикла программы в единицах 0,1 мс
D8011	●	–		Минимальное время отработки цикла программы в единицах 0,1 мс
D8012	●	–		Максимальное время отработки цикла программы в единицах 0,1 мс
D8013	●	●	FX1S FX1N FX2N FX2NC FX3U	Счетчик секунд (0...59) реального времени
D8014	●	●		Счетчик минут (0...59) реального времени
D8015	●	●		Счетчик часов (0...23) реального времени
D8016	●	●		Счетчик дней (1...31) реального времени
D8017	●	●		Счетчик месяцев (1...12) реального времени
D8018	●	●		Счетчик лет (0...99) реального времени
D8019	●	●		Счетчик дней недели (0...6) реального времени

Табл. 9-26: Специальные регистры тактов времени и реального времени

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.3 ФЛАГИ (D8020–D8029)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8020	—	●	FX1S FX1N	Входной фильтр для входов X0...X7. Настраиваемое значение времени между 0 мс и 15 мс в единицах 1 мс (стандартное значение 10 мс)
			FX2N FX2NC FX3U	Входной фильтр для входов X0...X17. Настраиваемое значение времени между 0 мс и 15 мс в единицах 1 мс (стандартное значение 10 мс)
D8021	–	●	FX1S	Входной фильтр для входов X10...X17. Настраиваемое значение времени между 0 мс и 15 мс в единицах 1 мс (стандартное значение 10 мс)
D8022 – D8027	–	–	–	Зарезервировано
D8028	●	–	FX1S FX1N FX2N FX2NC FX3U	Текущее значение данных в индексных регистрах Z0
D8029	●	–		Текущее значение данных в индексных регистрах V0

Табл. 9-25: Специальные регистры флагов

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.4 РЕЖИМЫ РАБОТЫ ПК (D8030 – D8039)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8030	●	–	FX1S FX1N	Считываемое значение с потенциометра VR 1 (0 ... 255).
D8031	●	–		Считываемое значение с потенциометра VR 2 (0 ... 255).
D8032 – D8038	–	–	–	Зарезервировано
D8039	—	●	FX1S FX1N FX2N FX2NC FX3U	Установлено постоянное время цикла программы в единицах 1 мс.

Табл. 9-27: Специальные регистры для режимов работы ПК

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.5 СОСТОЯНИЕ STL (D8040–D8049)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8040	●	–	FX1S FX1N FX2N FX3U	Номер 1-го активированного состояния шага
D8041	●	–		Номер 2-го активированного состояния шага
D8042	●	–		Номер 3-го активированного состояния шага
D8043	●	–		Номер 4-го активированного состояния шага
D8044	●	–		Номер 5-го активированного состояния шага
D8045	●	–		Номер 6-го активированного состояния шага
D8046	●	–		Номер 7-го активированного состояния шага
D8047	●	–		Номер 8-го активированного состояния шага
D8048	—	–	–	Зарезервировано
D8049	●	–	FX2N FX2NC FX3U	Последний меркер ошибки. Регистр запоминает последний активный меркер ошибки из области S900 до S999.

Табл. 9-28: Состояние STL (шаговых инструкций)

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.6 РЕГИСТРЫ СВЯЗИ И СПЕЦИАЛЬНЫЕ (D8070 – D8099)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8070	●	–	FX1S/FX1N FX2N/FX2NC FX3U	Контроль превышения времени отработки цикла программы (Watch-Dog) 500 мс для параллельной связи
D8071 – D8073	—	—	–	Зарезервировано
D8074 ... D8098	—	—	FX2N FX2NC FX3U	При выполнении Sampling Trace эти регистры используются контроллером
D8099	—	●	FX2N FX2NC FX3U	Кольцевой таймер, устанавливается от 0 до 32767 шагами в 0,1 мс

Табл. 9-30: Специальные регистры для связи и специальные

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.7 ПРОЧИЕ РЕГИСТРЫ (D8101 – D8114)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8101	●	–	FX3U	номер версии в формате "16VVV" (например, FX3U версии 1.00 ?16100)
D8102	●	–	FX1S FX1N FX2N FX2NC FX3U	емкость памяти: 0002 ? 2 тысячи шагов (только у FX1S) 0004 ? 4 тысячи шагов (FX2N, FX2NC) 0008 ? 8 тысяч шагов (FX1N, FX2N, FX2NC) 0016 ? 16 тысяч шагов (FX2N, FX2NC) 0064 ? 64 тысячи шагов (только у FX3U)
D8104	●	–	FX2N/FX2NC	идентификационный код для расширения памяти
D8105	●	–		номер версии расширения памяти (например, версия 1.00 ? 100)
D8107	●	–	FX3U	количество сохраненных комментариев к операндам
D8108	●	–		количество подключенных специальных модулей
D8109	●	–	FX2N/FX2NC FX3U	Адрес операнда, у которого появилась ошибка активизации выхода.
D8112	●	–	FX1S FX1N	адаптер FX1N-2AD-BD: цифровое входное значение канала 1
D8113	●	–		адаптер FX1N-2AD-BD: цифровое входное значение канала 1
D8114	●	●		адаптер FX1N-1DA-BD: цифровое выходное значение

Табл. 9-29: Прочие регистры

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.8 РЕГИСТРЫ КОММУТАЦИОННЫХ АДАПТЕРОВ (D8120 – D8129)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8120	—	●	FX1S FX1N FX2N FX2NC FX3U	Формат коммуникации
D8121	—	●		Номер локальной станции (485-сеть)
D8122	●	—		RS, количество переданных остаточных данных
D8123	●	—		RS, количество принимаемых данных
D8124	—	●		RS, головная телеграмма (STX(02H))
D8125	—	●		232 ADP, конец телеграммы (ETX(03H))
D8127	—	●		RS485, головной адрес требуемой станции
D8128	—	●		RS485, длина данных требуемых данных
D8129	—	●		RS485, установка времени времени выхода таймера сети

Табл. 9-31: Специальные регистры для коммуникационных адаптеров

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.9 РЕГИСТРЫ ИСПОЛНЕНИЯ ДЛЯ HSZ- И PLSY-ИНСТРУКЦИИ (D8130 – D8148)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8130	●	—	FX2N FX2NC FX3U	Текущий цикл сравнения HSZ-инструкции
D8131	●	—		Текущий цикл сравнения HSZ-инструкции при активизированной PLSY-инструкции
D8132, D8133	●	—		Выходная частота для PLSY-инструкции
D8134, D8135	●	—		Копия значения для операции сравнения при применении HSY-инструкции совместно с PLSY-инструкцией (32 бита)
D8136, D8137	●	—	FX1S/FX1N FX2N/FX2NC FX3U	Количество выданных импульсов посредством PLSY- и PLSR-инструкций (32 бита)
D8138	●	—	FX3U	табличный счетчик команды DHSCT
D8139	●	—		количество выполненных команд DHSCS, DHSCR, DHSZ и DHSCT
D8140, D8141	●	—	FX1S FX1N FX2N FX2NC FX3U	Количество выданных на Y0 импульсов посредством PLSY- и PLSR-инструкций (32 бита)
D8142, D8143	●	—		Количество выданных на Y1 импульсов посредством PLSY- и PLSR-инструкций (32 бита)
D8145	—	●	FX1S FX1N	установка величины смещения (по умолчанию:0)
D8146, D8147	—	●		установка макс. скорости (по умолчанию: 100,000)
D8148	—	●		Установка времени разгона/замедления (по умолчанию: 100)

Табл. 9-32: Специальные регистры для HSZ- И PLSY-инструкций

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.10 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ КОММУНИКАЦИЙ С ПРЕОБРАЗОВАТЕЛЯМИ ЧАСТОТЫ

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение	
D8150	–	●	FX3U	коммуникация по каналу 1	время ожидания ответа от преобразователя частоты
D8151	●	–			номер шага команды при коммуникации с преобразователем частоты (предустановка: -1)
D8152	●	–			код ошибки
D8153	●	–			номер шага команды, при которой во время коммуникации возникла ошибка (сохраненное значение; предустановка: -1)
D8154	●	–	FX3U	коммуникация по каналу 1	номер параметра, при котором во время выполнения команды IVBWR возникла ошибка (предустановка: -1) коммуникация по каналу 1
	●	–	FX2N/FX2NC (начиная с 3.0)		время ожидания ответа при команде EXTR
D8155	–	●	FX3U	Kommunikation über Kanal 2	время ожидания ответа от преобразователя частоты
	●	–	FX2N/FX2NC (начиная с 3.0)		номер шага команды при коммуникации с помощью команды EXTR
D8156	●	–	FX3U	коммуникация по каналу 2	номер шага команды при коммуникации с преобразователем частоты (предустановка: -1)
	●	–	FX2N/FX2NC (начиная с 3.0)		код ошибки при команде EXTR
D8157	●	–	FX3U	коммуникация по каналу 2	код ошибки
	●	–	FX2N/FX2NC (начиная с 3.0)		сохраненный код ошибки при команде EXTR (предустановка: -1)
D8158	●	–	FX3U	коммуникация по каналу 2	номер шага команды, при которой во время коммуникации возникла ошибка (сохраненное значение; предустановка: -1)
D8159	●	–			номер параметра, при котором во время выполнения команды IVBWR возникла ошибка (предустановка: -1)

Табл. 9-33: Специальные регистры для коммуникации с преобразователями частоты

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.11 РАЗЛИЧНЫЕ ФУНКЦИИ (D8158 – D8164)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8158	—	●	FX1S FX1N	Регистр управления для FX1N-5DM*1 По умолчанию: k-1
D8159	—	●		Регистр управления для FX1N-5DM*1 По умолчанию: k-1
D8164	—	●	FX2N/FX2NC (начиная с 2.00)	Число точек обмена для FROM/TO (начиная с версии 2.00 CPU FX2N/2NC)
D8169	●	—	FX3U	вид ограничения доступа с помощью второго пароля 0000H: без ограничения, 2-й пароль не создан 0010H: защита от записи для программы 0011H: защита от чтения и записи для программы 0012H: все операции в режиме он-лайн заблокированы 0020H: пароль снят, ограничений не имеется

Табл. 9-34: Индекс-регистры

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.12 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ СЕТИ ТИПА N:N (D8173 - D8180, D820)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение	
D8173	●	–	FX1S FX1N FX2N FX2NC FX3U	номер станции	
D8174	●	–		общее количество подчиненных станций	
D8175	●	–		обновленная область	
D8176	–	●		Настройки	номер станции
D8177	–	●			количество подчиненных станций
D8178	–	●			обновленная область
D8179	–	●			попытки повторения
D8180	–	●			контрольное время
D8201 (D201 в FX1S)	●	–			текущее время опроса
D8202 (D202 в FX1S)	●	–		максимальное время опроса	
D8203 (D203 в FX1S)	●	–		количество ошибок коммуникации у главной станции	
D8204 (D204 в FX1S)	●	–		количество ошибок коммуникации у подчиненной станции 1	
D8205 (D205 в FX1S)	●	–		количество ошибок коммуникации у подчиненной станции 2	
D8206 (D206 в FX1S)	●	–		количество ошибок коммуникации у подчиненной станции 3	
D8207 (D207 в FX1S)	●	–		количество ошибок коммуникации у подчиненной станции 4	
D8208 (D208 в FX1S)	●	–		количество ошибок коммуникации у подчиненной станции 5	
D8209 (D209 в FX1S)	●	–		количество ошибок коммуникации у подчиненной станции 6	
D8210 (D210 в FX1S)	●	–		количество ошибок коммуникации у подчиненной станции 7	
D8211 (D211 в FX1S)	●	–		код ошибки коммуникации у главной станции	
D8212 (D212 в FX1S)	●	–		код ошибки коммуникации у подчиненной станции 1	
D8213 (D213 в FX1S)	●	–		код ошибки коммуникации у подчиненной станции 2	
D8214 (D214 в FX1S)	●	–		код ошибки коммуникации у подчиненной станции 3	
D8215 (D215 для FX1S)	●	–		код ошибки коммуникации у подчиненной станции 4	
D8216 (D216 в FX1S)	●	–		код ошибки коммуникации у подчиненной станции 5	
D8217 (D217 в FX1S)	●	–		код ошибки коммуникации у подчиненной станции 6	
D8218 (D218 в FX1S)	●	–		код ошибки коммуникации у подчиненной станции 7	

Табл. 9-35: Специальные регистры для сети типа n:n

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.13 ИНДЕКС-РЕГИСТРЫ (D8182 – D8195)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8182	●	–	FX1S FX1N FX2N FX2NC FX3U	Индекс-регистр Z1
D8183	●	–		Индекс-регистр V1
D8184	●	–		Индекс-регистр Z2
D8185	●	–		Индекс-регистр V2
D8186	●	–		Индекс-регистр Z3
D8187	●	–		Индекс-регистр V3
D8188	●	–		Индекс-регистр Z4
D8189	●	–		Индекс-регистр V4
D8190	●	–		Индекс-регистр Z5
D8191	●	–		Индекс-регистр V5
D8192	●	–		Индекс-регистр Z6
D8193	●	–		Индекс-регистр V6
D8194	●	–		Индекс-регистр Z7
D8195	●	–		Индекс-регистр V7

Табл. 9-24: Индекс-регистры

- ① Специальный регистр, значение данных которого в программе ПК может только читаться.
- ② Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

9.2.14 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ АНАЛОГОВЫХ СПЕЦИАЛЬНЫХ МОДУЛЕЙ (D8260 - D8299)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8260 ... D8269	–	●	FX3U	специальный регистр для 1-го специального модуля
D8270 ... D8279	–	●		специальный регистр для 2-го специального модуля
D8280 ... D8289	–	●		специальный регистр для 3-го специального модуля
D8290 ... D8299	–	●		специальный регистр для 4-го специального модуля

Табл. 9-36: Специальные регистры для подключенных аналоговых модулей

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

УКАЗАНИЕ

Функция специальных маркеров M8260...M8299 описана в руководствах по аналоговым модулям.

9.2.15 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ МОДУЛЯ ИНДИКАЦИИ FX3U-7DM (D8300 - D8303)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8300	—	●	FX3U	управляющий регистр для модуля индикации FX3U-7DM; предустановка: K-1
D8301	—	●		управляющий регистр для модуля индикации FX3U-7DM; предустановка: K-1
D8302	●	—		настройка языка*: "0": японский иные значения кроме "0": английский
D8303	●	—		настройка контраста жидкокристаллического дисплея предустановка: "0"

Табл. 9-37: Специальные регистры для модуля индикации FX3U-7DM

* * Эта настройка сохраняется также при выключении напряжения питания

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.16 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ КОМАНДЫ RND (D8310 И D8311)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8310	●	—	FX3U	данные для выработки случайного числа (младшее слово)
D8311	●	—		данные для выработки случайного числа (старшее слово)

Табл. 9-38: Специальные регистры для выработки случайного числа с помощью команды RND

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.17 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ ДИАГНОСТИКИ ОШИБОК (D8312 - D8319)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение	
				описание	биты
D8312	–	●	FX3U	сохраненный номер шага, в котором возникла ошибка (индикация с помощью M8068, 32 бита)	младшее слово
D8313	–	●			старшее слово
D8314*	●	–	FX3U	номер шага, в котором возникла ошибка (индикация с помощью M8065...M8065, 32 бита)	младшее слово
D8315*	●	–			старшее слово
D8316	●	–	FX3U	номер шага, в котором используется недействительный адрес ввода-вывода (индикация с помощью M8316, 32 бита)	младшее слово
D8317	●	–			старшее слово
D8318	●	–	FX3U	ошибка при инициализации буферной памяти	№ специального модуля
D8319	●	–			адрес в буферной памяти

Табл. 9-39: Специальные регистры, содержащие номера шагов программы или информацию для диагностики ошибок

* содержимое D8114 и D8115 стирается при переключении контроллера в режим "RUN"

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.18 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ ВЫВОДА ИМПУЛЬСОВ И ПОЗИЦИОНИРОВАНИЯ

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение		
				описание	биты	
D8330	●	–	FX3U	команда DUTY: счетчик циклов для выхода 1		
D8331	●	–		команда DUTY: счетчик циклов для выхода 2		
D8332	●	–		команда DUTY: счетчик циклов для выхода 3		
D8333	●	–		команда DUTY: счетчик циклов для выхода 4		
D8334	●	–		команда DUTY: счетчик циклов для выхода 5		
D8336	●	–	FX3U	вход прерывания для команды DVIT		
D8340	–	–	FX3U	текущее значение	биты 15 - 0	
D8341	–	–			биты 31 - 16	
D8342	–	●		Y000	минимальная скорость; предустановка: 0	
D8343	–	●			максимальная скорость; предустановка: 100000	биты 15 - 0
D8344	–	●				биты 31 - 16
D8345	–	●			ползучая скорость; предустановка: 1000	
D8346	–	●			скорость для движения референцирования предустановка: 50000	биты 15 - 0
D8347	–	●				биты 31 - 16
D8348	–	●			время разгона; предустановка: 100	
D8349	–	●			время замедления; предустановка: 100	

Табл. 9-40: Специальные регистры для вывода импульсов и позиционирования

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение		
D8350	–	–	FX3U	Y001	текущее значение	биты 15 - 0
D8351	–	–			биты 31 - 16	
D8352	–	●			минимальная скорость; предустановка: 0	
D8353	–	●			максимальная скорость; предустановка: 100000	биты 15 - 0
D8354	–	●				биты 31 - 16
D8355	–	●			ползучая скорость; предустановка: 1000	
D8356	–	●			скорость движения референцирования; предустановка: 50000	биты 15 - 0
D8357	–	●				биты 31 - 16
D8358	–	●			время разгона; предустановка: 100	
D8359	–	●			время замедления; предустановка: 100	
D8360	–	–	FX3U	Y002	текущее значение для Y002	биты 15 - 0
D8361	–	–				биты 31 - 16
D8362	–	●			минимальная скорость; предустановка: 0	
D8363	–	●			максимальная скорость; предустановка: 100000	биты 15 - 0
D8364	–	●				биты 31 - 16
D8365	–	●			ползучая скорость; предустановка: 1000	
D8366	–	●			скорость движения референцирования; предустановка: 50000	биты 15 - 0
D8367	–	●				биты 31 - 16
D8368	–	●			время ускорения; предустановка: 100	
D8369	–	●			время замедления; предустановка: 100	
D8370	–	–	FX3U	Y003*	текущее значение	биты 15 - 0
D8371	–	–				биты 31 - 16
D8372	–	●			минимальная скорость; предустановка: 0	
D8373	–	●			максимальная скорость; предустановка: 100000	биты 15 - 0
D8374	–	●				биты 31 - 16
D8375	–	●			ползучая скорость; предустановка: 1000	
D8376	–	●			скорость движения референцирования; предустановка: 50000	биты 15 - 0
D8377	–	●				биты 31 - 16
D8378	–	●			время ускорения; предустановка: 100	
D8379	–	●			время замедления; предустановка: 100	
D8464	●	–	FX3U	Y000	команда DSZR и ZRN: содержимые этих регистров стираются при достижении нулевой точки, если установлен соответствующий маркер M8464...M8467.	
D8465	●	–		Y001		
D8466	●	–		Y002		
D8467	●	–		Y003*		

Табл. 9-41: Специальные регистры для позиционирования

* выход Y003 имеется только в том случае, если подключены два адаптера FX3U-2HSY-ADP.

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.19 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЯ ПРОГРАММ ПРЕРЫВАНИЯ (D8393)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8393	–	●	FX3U	время задержки

Табл. 9-42: Специальные регистры для программ прерывания

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.20 КОЛЬЦЕВОЙ СЧЕТЧИК (D8398 И D8399)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8398	–	●	FX3U	32-битный кольцевой счетчик, диапазон счета: от 0 до 2147483647
D8399	–	●		Кольцевой счетчик активируется после установки M8398.

Табл. 9-43: Регистры D8398 и D8399 сохраняют состояние кольцевого счетчика

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

9.2.21 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ ДЛЖОММУНИКАЦИЙ (D8400 - D8437)

Специальные регистры адреса операндов	① Чтение значений данных	В Изменения значений данных	CPU	Значение	
D8400	–	●	FX3U	Команда RS2 (канал 1)	настройка формата передачи
D8402*	●	–			количество данных, которые еще требуется передать
D8403*	●	–			количество принятых данных
D8405	●	–			индикация параметров коммуникации
D8409	–	●			контрольное время
D8410	–	●			метка начала 1 и 2; предустановка: STX
D8411	–	●			метка начала 3 и 4
D8412	–	●			метка конца 1 и 2 предустановка: ETX
D8413	–	●			метка конца 3 и 4
D8414	●	–			принятая контрольная сумма
D8415	●	–			рассчитанная контрольная сумма для принятых данных
D8416	●	–			переданная контрольная сумма
D8419	●	–			индикация режима

Табл. 9-44: Специальные регистры для коммуникации с помощью команды RS2

Специальные регистры адреса операндов	① Чтение значений данных	В Изменения значений данных	CPU	Значение		
D8420	–	●	FX3U	Команда RS2 (канал 2)	настройка формата передачи	
D8421	–	●		Сеть RS485	настройка номера станции	
D8422*	●	–		Команда RS2 (канал 2)	количество данных, которые еще требуется передать	
D8423*	●	–			количество принятых данных	
D8425	●	–			индикация параметров коммуникации	
D8427	–	●		Сеть RS485	первый регистр при передаче по запросу	
D8428	–	●			количество регистров при передаче по запросу	
D8429	–	●		Команда RS2 (канал 2)	контрольное время	
D8430	–	●			метка начала 1 и 2; предустановка: STX	
D8431	–	●			метка начала 3 и 4	
D8432	–	●			метка конца 1 и 2 предустановка: ETX	
D8433	–	●			метка конца 3 и 4	
D8434	●	–			Empfangene Prüfsumme	
D8435	●	–			рассчитанная контрольная сумма для принятых данных	
D8436	●	–			переданная контрольная сумма	
D8438	●	–			Код ошибки при последовательной коммуникации по каналу 2	
D8439	●	–			Команда RS2 (канал 2)	индикация режима

Табл. 9-45: Специальные регистры для коммуникации с помощью команды RS2 или через сеть RS485

* содержимое этих специальных регистров стирается при останове контроллера

- ① специальные регистры, значения которых программа контроллера может только считывать
- ② специальные регистры, значения которых программа контроллера может считывать и изменять

10 ОШИБКИ ПРОГРАММЫ

10.1 ОПОЗНАНИЕ ОШИБКИ

10.1.1 СПЕЦИАЛЬНЫЕ МЕРКЕРЫ (M8060...M8069)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение	(„PROG-E“/„ERROR“)-LED	режимы работы ПК
M8060	●	—	FX2N/FX2NC FX3U	Ошибка конфигурации входов/выходов	Вых.	RUN
M8061	●	—	FX1S/FX1N FX2N/FX2NC FX3U	Ошибка технических средств ПК	Вх.	STOP
M8062	●	—	FX2N FX2NC	Ошибка PC/HPP-коммуникации	Вых.	RUN
M8063	●	—	FX1S FX1N FX2N FX2NC FX3U	Ошибка параллельной коммуникации	Вых.	RUN
M8064	●	—		Ошибка параметров	Сигнал	STOP
M8065	●	—		Ошибка синтаксиса программы	Сигнал	STOP
M8066	●	—		Ошибка программирования	Сигнал	STOP
M8067	●	—		Ошибка выполнения в области операндов	Вых.	RUN
M8068	—	●		Ошибка выполнения в области операндов с памятью	Вых.	RUN
M8069	—	●		FX2N/FX2NC FX3U	Ошибка шин входов/выходов ③	—

Табл. 10-1: Специальные меркеры для опознания ошибок

- ① Специальный меркер, у которого в программе ПК может опрашиваться только сигнал состояния по контактной инструкции (например, LD- или LDI-инструкции).
- ② Специальный меркер, который в программе ПК включается или отключается непосредственно по инструкции.
- ③ После включения M8069 выполняется контроль шин входов/выходов. Если здесь определена ошибка, то код ошибки из 6130 записывается в специальный регистр D8069 и включается специальный меркер M8061.

10.1.2 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ (D8060–D8069)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
D8060	●	–	FX2N/FX2NC FX3U	Адреса входов/выходов ошибочных базовых и расширенных устройств
D8061	●	–	FX1S/FX1N FX2N/FX2NC FX3U	Номер кода ошибки технических средств ПК записывается в D8061
D8062	●	–	FX2N/FX2NC FX3U	Код ошибки для ошибки коммуникации между ПК и программатором (см.соответствующий код ошибки в соответствующей таблице)
D8063	●	–	FX1S FX1N FX2N FX2NC FX3U	Код ошибки для ошибки параллельной связи (см.описание технических средств FX)
D8064	●	–		Номер кода ошибки параметров запоминается в D8064.
D8065	●	–		Номер кода ошибки синтаксиса программы запоминается в D8065.
D8066	●	–		Номер кода ошибки программирования запоминается в D8066.
D8067	●	–		Номер кода ошибки выполнения запоминается в D8067.
D8068*	—	●		Ошибка выполнения адреса шага запоминается в D8068. При количестве шагов более 32 тысяч номер шага сохраняется в D8313 и D8312
D8069*	●	–		Адрес шага ошибки M8065...M8067 При количестве шагов более 32 тысяч номер шага сохраняется в D8315 и D8314.
D8449	●	–		FX3U

Табл. 10-2: Специальные регистры для опознания ошибок

* При переводе контроллера в режим RUN эти регистры стираются.

- ① Специальный регистр, который в программе ПК может только читаться.
- ② Специальный регистр, значения данных которого в программе ПК могут читаться и изменяться.

10.2 КОДЫ ОШИБОК

10.2.1 КОДЫ ОШИБОК (3801-3820)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка при последовательной коммуникации	D8438	0000	ошибок нет	Проверьте электропитание и проводку. Более подробную информацию см. в руководстве по коммуникации для FX.
		3801	ошибка четности/переполнения/рамок	
		3802	ошибочный знак	
		3803	ошибка контрольной суммы	
		3804	ошибка формата	
		3805	недействительная команда	
		3806	контрольное время истекло	
		3807	ошибка при инициализации модема	
		3808	ошибка параметра при сети типа n:n	
		3812	ошибочный знак при параллельном соединении	
		3813	ошибка контрольной суммы при параллельном соединении	
		3814	ошибка формата при параллельном соединении	
3820	ошибка при обмене данными с преобразователем частоты			

Табл. 10-3: Коды ошибок FX3U при сбое последовательной коммуникации

10.2.2 КОДЫ ОШИБОК (6101–6409)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка технических средств ПК	D8061	0000	Нет ошибки	Проверить связь между программатором и ПК. Устраните в нужном случае ошибку цепи тока.
		6101	Ошибка памяти RAM	
		6102	Ошибочная токовая цепь	
		6103	Ошибка входов/выходов (M8069=ВКЛ)	
		6104	Ошибка источника питания 24 В пост. тока (M8069=ВКЛ)	Время цикла программы больше, чем заданное значение в D8000.
		6105	Ошибка контроля времени цикла (Watch-Dog)	
		6106	Ошибка при выработке таблицы ввода-вывода (ошибка центрального процессора). После включения контроллера не был включено электропитание расширительного модуля со встроенным электропитанием. Или не удалось сопоставить входы-выходы в сети CC-Link.	
6107	Ошибка при конфигурации системы	Проверьте количество подключенных специальных модулей		
Ошибка коммуникации между ПК и Программатором (только у FX2N и FX2NC)	D8062	0000	Нет ошибки	Устраните причину ошибки и повторите передачу.
		6201	Ошибка паритета, перехода, наклона (рампы)	
		6202	Ошибочный знак коммуникации	
		6203	Ошибка контроля суммы при передаче данных	
		6204	Ошибочный формат данных	
6205	Ошибка инструкции			

Табл. 10-4: Коды ошибок (6101...6409)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка в коммуникации второго параллельного адаптера FX-40AV/AP	D8063	0000	Нет ошибки	Проверьте напряжение питания и монтажные провода обоих параллельных адаптеров. (В 485-сети ошибки не могут передаваться по сети и должны контролироваться мастер-модулем)
		6301	Ошибка паритета, перехода, наклона (рампы)	
		6302	Ошибка знака	
		6303	Ошибка контроля суммы	
		6304	Ошибка формата	
		6305	Ошибочная команда (485-сеть)	
		6306	Ошибка контроля времени цикла (Watch-Dog)	
		6307	Ошибка при инициализации модема	
		6308	Ошибка параметра при сети типа n:n	
		6312	Ошибка знака при параллельной связи	
		6313	Ошибка контрольной суммы при параллельной связи	
		6314	Ошибка формата данных при параллельной связи	
Ошибка параметров	D8064	0000	Нет ошибки	становите ПК и откорректируйте ошибочные данные
		6401	Ошибка программы по контрольной сумме	
		6402	Ошибочная настройка емкости памяти	
		6403	Ошибочная настройка для операнда с памятью	
		6404	Ошибочная настройка области комментариев	
		6405	Ошибочная настройка файловых регистров	
		6406	Ошибка при записи заданных значений в буферную память специальных модулей или ошибка контрольной суммы при команде позиционирования	
		6409	Другие ошибочные параметры	

Табл. 10-5: Коды ошибок (6301–6409)

10.2.3 КОДЫ ОШИБОК (6501–6511)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка синтаксиса программы	D8065	0000	Нет ошибки	Во время программирования каждый раз контролируйте инструкцию. Если встретилась синтаксическая ошибка устраните ее в режиме программирования.
		6501	Инструкция. Символ операнда или адрес операнда запрограммирован ошибочно.	
		6502	Нельзя задавать OUT-T-инструкцию или OUT-C-инструкцию перед программированием соответствующего задаваемого значения.	
		6503	1) Одной из OUT-T-инструкции или OUT-C-инструкции не предшествует никакого задаваемого значения.2) Недостаточно количество операндов для применяемых инструкций	
		6504	1) Часто применяется одна и та же маркировка точек.2) Часто применено одно и тоже входное условие для программы прерывания и высокоскоростного счетчика.	
		6505	Недопустимые адреса операндов	
		6506	Несоответствующая инструкция	
		6507	Несоответствующая передача точки	
		6508	Несоответствующая передача точки прерывания	
		6509	Другие ошибки	
		6510	Ошибочный номер уровня MC-слоя (N)	
6511	Перекрыт входной адрес прерывания и высокоскоростного счетчика.			

Табл. 10-6: Коды ошибок (6501...6511)

10.2.4 КОДЫ ОШИБОК (6601–6609)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка программирования	D8066	0000	Нет ошибки	<p>Ошибки программы появляются, если возникают ошибочные комбинации инструкций или ошибочные соотношения между парными взаимозависимыми инструкциями.</p> <p>Опознанная ошибка должна исправляться в режиме программирования.</p>
		6601	LD- или LDI-инструкции были запрограммированы один после другого 9 раз или чаще.	
		6602	1) Ни одной LD- или LDI-инструкции. 2) LD-, LDI-, AND-, ANI-инструкции неправильно включены. 3) Следующие инструкции связаны некорректно: STL, RET, MCR, P, I, EI, DI, IRET, FOR, NEXT, FEND, SRET, END. 4) Отсутствует MPP-инструкция.	
		6603	MPS-инструкция была запрограммирована один за другим 12 раз или чаще.	
		6604	MPS-, MRD-, и MPP-инструкции неправильно включены.	
		6605	1) LD- или LDI-инструкции были запрограммированы один за другим 9 раз или чаще. 2) MC-, MCR- или прерывания-инструкции были запрограммированы внутри шага. 3) RET-инструкция была запрограммирована вне шага состояния или отсутствует в программе.	
		6606	1) Отсутствуют точка P или точка прерывания I. 2) Отсутствуют IRET-/SRET-инструкции. 3) IRET-/SRET- и прерывания-инструкции были запрограммированы в главной программе. 4) STL/RET/MC или MCR были запрограммированы в подпрограмме или в программе прерывания.	
		6607	1) Недопустимые FOR-/NEXT-инструкции, 6 или больше уровней. 2) Следующие инструкции были запрограммированы в FOR-/NEXT-петле: STL, RET, MC, MCR, IRET, FEND, SRET, END.	
		6608	1) Недопустимые MC-, MCR-инструкции. 2) Отсутствует MCR NO. 3) IRET-/SRET- или прерывания-инструкции были запрограммированы между MC и MCR блоками.	
6609	Другие ошибки			

Табл. 10-7: Коды ошибок (6601–6609)

10.2.5 КОДЫ ОШИБОК (6610–6632)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка программирования	D8066	6610	LD и LDI используются непрерывно 9 или более раз подряд	Ошибки программы появляются, если возникают ошибочные комбинации инструкций или ошибочные соотношения между парными взаимозависимыми инструкциями. Опознанная ошибка должна исправляться в режиме программирования.
		6611	Число инструкций LD/LDI больше числа инструкций ANB/ORB	
		6612	Число инструкций LD/LDI меньше числа инструкций ANB/ORB	
		6613	MPS используются непрерывно 12 или более раз	
		6614	Инструкция MPS отсутствует	
		6615	Инструкция MPP отсутствует	
		6616	Неразрешенное использование инструкций MPS/MRD/MPP, возможно отсутствует выход (катушка).	
		6617	Одна из следующих инструкции не связана с активной шиной: STL, RET, MCR, Pointer (P), Interrupt (I), EI, DI, SRET, IRET, FOR, NEXT, FEND und END.	
		6618	STL, RET, MC или MCR использованы в подпрограмме или алгоритме прерывания.	
		6619	Неверное использование следующих инструкций в цикле FOR-NEXT: STL.....END.	
		6620	Превышено число (5) вложений для инструкций FOR-NEXT.	
		6621	Число инструкций FOR и NEXT не соответствует.	
		6622	Иструкция NEXT не обнаружена.	
		6623	Иструкция MC не обнаружена.	
		6624	Иструкция MCR не обнаружена.	
		6625	STL используется непрерывно 9 или более раз	
		6626	Неверное использование следующих инструкций в STL-, RET-областях: MC, MCP, I, IRET, SRET.	
		6627	Иструкция RET не обнаружена.	
		6628	Неправильно запрограммированы инструкции I-, IRET- и SRET-	
		6629	Иструкции Pointer (P)- и Interrupt (I)-Label не обнаружены.	
6630	Иструкции SRET или IRET не обнаружены.			
6631	Инструкция SRET-неправильно расположена в программе			
6632	Инструкция IRET-неправильно расположена в программе			

Табл. 10-6. Коды ошибок (6610...6632)

10.2.6 КОДЫ ОШИБОК (6701–6710)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка исполнения	D8067	0000	Нет ошибки	Эти ошибки появляются во время обработки инструкции. В случае ошибки нужно остановить ПК и устранить ошибку в режиме программирования. Исполнение ошибки может появиться самостоятельно, если не сообщается ни о синтаксической ни о программной ошибке. (Так например, D500Z является действенным отображением. Однако, если Z имеет значение 100, то выполняется попытка доступа к регистру данных D600. В этом случае возникает ошибка, так как D600 не существует).
		6701	1) Для CJ-инструкции не указывается цель перехода. 2) Маркировка точки программируется в одном блоке, который обрабатывается только после END-инструкции. 3) Независимая метка перехода была определена внутри FOR/NEXT-петли или в подпрограмме.	
		6702	6 или больше CALL-инструкций.	
		6703	3 или больше уровней прерывания.	
		6704	6 или больше FOR/NEXT уровней.	
		6705	В применяемую инструкцию введен ошибочный операнд.	
		6706	Область операндов или область данных, которая была запрограммирована в применяемой инструкции, находится вне допустимой области.	
		6707	Доступ выполнен к файловому регистру, который находится вне допустимой области адресов.	
		6708	Ошибка при присоединении с FROM-/TO-инструкцией.	
		6709	Другие ошибки (например, падающая IRET-инструкция, недопустимое соотношение между FOR-/NEXT и т.д.).	
6710	Ошибочное указание операндов команды (например, в команде передачи один и тот же операнд указан в качестве цели и в качестве источника)			

Табл. 10-7: Коды ошибок (6701-6710)

10.2.7 КОДЫ ОШИБОК (6730–6773)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка исполнения команд PID (пропорционального, интегрального, дифференциального регулирования)	D8067	6730	Цикл ощупывания T_s ($T_s < 0$ или > 32767)	Указанные значения параметров находятся вне допустимой области. Команды PID-регулирования должны отключаться раньше, чем продолжится выполнение.
		6732	Коэффициент фильтрации α ($\alpha < 0$ или ≥ 101)	
		6733	Константа пропорциональности K_p ($K_p < 0$ или > 32767)	
		6734	Константа интегральности T_i ($T_i < 0$ или > 32767)	
		6735	Дифференциальный усилитель K_D ($K_D < 0$ или ≥ 101)	
		6736	Дифференциальная константа T_D ($T_D < 0$ или > 32767)	
		6740	Время ощупывания T_s меньше, чем время цикла программы	Время ощупывания включается одновременно с временем цикла.
		6742	Действительное значение Δ очень большое	Относящиеся сюда данные отключаются на следующем граничном значении. При всех кодах ошибок (кроме 6745) имеет место или -32768 или +32767. Выполнение продолжается, но PID-команда должна отключиться.
		6743	Расчетное отклонение ϵ очень большое	
		6744	Результат интеграции очень большой	
		6745	Дифференциальное значение очень большое или разностное значение превысило допустимую область	
		6746	Результат дифференциальности очень большой	
		6747	Общий результат PID очень большой	Откорректируйте настройки
		6748	Верхний предел для значений, выдаваемых ПИД-регулятором, меньше нижнего предела.	
		6749	Неправильные настройки для сигнализации о фактическом значении или выходном значении (например, введены отрицательные значения)	Колебания не могут выравниваться PID-регулятором. Выполнение PID-команды окончено. PID-команда должна отключиться.
		6750	Заданное значение - действительное значение < 150 или регулировочный контур нестабилен (Разность между заданным и действительным значениями очень сильно колеблется и очень быстро).	
6751	Заданное значение значительно намного больше.			
6752	Большие колебания при задании функции автоматической настройки (Auto-Tuning-Funktion)			
6753	Недопустимое выходное значение при автонастройке (верхний предел $<$ нижний предел)	Откорректируйте настройки		
6754	Недопустимый гистерезис фактического значения при автонастройке ($SHPV < 0$)			
6755	При автонастройке были перезаписаны используемые операнды.	Убедитесь в том, что используемые ПИД-регулятором операнды не перезаписываются в программе.		

Табл. 10-10: : Коды ошибок (6730-6755)

УКАЗАНИЕ

| Более подробная информация по кодам ошибок команды PID имеется в разделе 7.3.8.

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка выполнения PID	D8067	6756	Ошибочный результат при автонастройке из-за слишком большого времени измерения	Время для автонастройки больше требуемого. Увеличьте диапазон между верхним и нижним пределом выходной величины, уменьшите коэффициент фильтра α или уменьшите порог для фактического значения (SHPV).
		6757	При автонастройке превышает допустимый диапазон для коэффициента пропорциональности KP. (KP может находиться между 0 и 32767.)	По сравнению с фактическим значением изменение коэффициента пропорциональности KP мало. Умножьте фактическое значение на 10, чтобы во время автонастройки это значение изменялось сильнее
		6758	При автонастройке превышает допустимый диапазон для постоянной интегрирования TI. (TI может находиться между 0 и 32767.)	Время для автонастройки больше требуемого. Увеличьте диапазон между верхним и нижним пределом выходной величины, уменьшите коэффициент фильтра α или уменьшите порог для фактического значения (SHPV).
		6759	При автонастройке превышает допустимый диапазон для постоянной дифференцирования TD. (TD может находиться между 0 и 32767.)	
Ошибка выполнения		6760	Ошибка контрольной суммы при считывании абсолютного положения из сервоусилителя	Проверьте подключение сервоусилителя и параметров. Проверьте также команду ABS.
		6762	Интерфейс, указанный для коммуникации с преобразователем частоты, уже применяется для другого обмена данными.	Проверьте и откорректируйте настройки.
		6763	Вход, указанный для команды DSZR, DVIT или ZRN, уже используется другой командой.	Проверьте, не применяется ли вход команды DSZR, DVIT или ZRN для прерывания, высокоскоростного счетчика, функции перехвата импульсов или команды SPD.
			Операнд для прерывания в команде DVIT превышает допустимую область.	Проверьте содержимое D8336.
6764	Выход для вывода импульсов уже применяется в команде позиционирования или команде вывода импульсов.	Проверьте и откорректируйте настройки.		

Табл. 10-11: Коды ошибок (6756-6764)

УКАЗАНИЕ

| Более подробные детали по кодовым ошибкам PID-инструкции содержит раздел 7.3.8

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка выполнения	D8067	6765	Запрограммировано слишком много прикладных команд.	Уменьшите количество прикладных команд в программе
		6770	Ошибка при записи в кассету памяти (флэш-EPROM)	Замените кассету памяти.
		6771	Не установлена кассета памяти типа флэш-EPROM.	Проверьте, правильно ли установлена кассета памяти.
		6772	Активирована защита от записи на кассету памяти типа флэш-EPROM.	Прежде чем передавать данные на кассету памяти, выключите защиту от записи на кассету.
		6773	Во время передачи программы в режиме RUN не удалось обратиться к кассете памяти типа флэш-EPROM.	Во время изменений в режиме RUN не должна происходить передача данных на кассету памяти или считывание данных из кассеты.

Табл. 10-13: Коды ошибок (6765-6773)

10.2.8 КОДЫ ОШИБОК ПРИ НЕИСПРАВНОСТЯХ В СПЕЦИАЛЬНЫХ МОДУЛЯХ

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка, относящаяся к специальному модулю	D8449	0000	Ошибок нет	—
		□020*	Ошибка суммы данных	Проверьте, правильно ли подключены специальные модули.
		□021*	Ошибка в данных	
		□080*	Ошибка при выполнении команды FROM или TO.	Проверьте параметры команд. Убедитесь в том, что области буферной памяти, к которым требуется обратиться, имеются в специальном модуле. Проверьте, правильно ли подключены специальные модули.
		□090*	Ошибка при обращении к периферийной аппаратуре	Проверьте соединение между программатором и контроллером. Проверьте, правильно ли подключены специальные модули.

Табл. 10-12: Коды ошибок в D8449 при неисправностях в специальных модулях

* Символ "□" заменяет номер специального модуля, который может быть в диапазоне между 0 и 7 (см. раздел 7.3)

А ТЕХНИЧЕСКИЕ ДАННЫЕ

А.1 ОБЗОР БАЗОВЫХ КОМАНД

Инструкция	Символ релейной схемы	Значение	Операнды	Количество шагов программы	Раздел
LD		ЗАГРУЗКА; Начало логического выражения (связей) с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.2
LDI		НЕТ ЗАГРУЗКИ; Начало логического выражения (связей) с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.2
OUT		ВЫХОД; Выход. Присвоение результата логического выражения	Y, M, S, T, C	Y, M: 1 S, Спец. меркер: 2 T: 3, C: 3 C(32бита): 5	Разд. 4.3
AND		И; Логическое выражение "И" с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.4
ANI		НЕ И; Логическое выражение "И" с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.4
OR		ИЛИ; Логическое выражение "ИЛИ" с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.5
ORI		НЕ ИЛИ; Логическое выражение "ИЛИ" с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.5
LDP		ЗАГРУЗКА; (импульсно) Начало логического выражения с опросом по переднему фронту	X, Y, M, S, T, G	2	Разд. 4.6
LDF		ЗАГРУЗКА; (импульсно) Начало логического выражения с опросом по заднему фронту	X, Y, M, S, T, G	2	Разд. 4.6
ANP		И; (импульсно) Начало логического выражения с опросом по заднему фронту	X, Y, M, S, T, G	2	Разд. 4.7

Табл. А-1: Обзор базовых команд (часть 1)

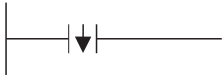


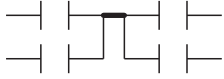
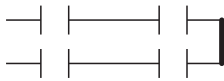
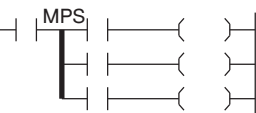
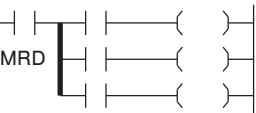
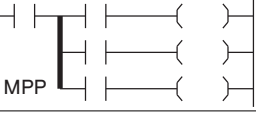
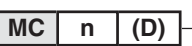



Инструкция	Символ релейной схемы	Значение	Операнды	Количество шагов программы	Раздел
ANF		И; (импульсно) Логическое выражение "И" с опросом по заднему фронту	X, Y, M, S, T, G	2	Разд. 4.7
ORP		ИЛИ; Логическое выражение "ИЛИ" с опросом по переднему фронту	X, Y, M, S, T, G	2	Разд. 4.8
ORF		ИЛИ; Логическое выражение "ИЛИ" с опросом по заднему фронту	X, Y, M, S, T, G	2	Разд. 4.8
ANB		И-БЛОК; Команда связи; Последовательное соединение параллельных соединений	—	1	Разд. 4.9
ORB		ИЛИ-БЛОК; Команда связи; Параллельное соединение последовательных соединений	—	1	Разд. 4.10
MPS		ДВИЖЕНИЕ ВНИЗ ПО СТЕКУ; Запоминание результатов логических связей	—	1	Разд. 4.11
MRD		ЧТЕНИЕ ВНИЗ ПО СТЕКУ; Чтение результатов логических связей	—	1	Разд. 4.11
MPP		ВСПЛЫВАЮЩИЙ СТЕК; Чтение и стирание записанных логических связей	—	1	Разд. 4.11
MC		МАСТЕР КОНТРОЛЬ; Включение контрольных условий	Y, M, без специальных меркеров	3	Разд. 4.12
MCR		МАСТЕР КОНТРОЛЬ СБРОС; Отключение контрольных условий	N	2	Разд. 4.12
SET		ВКЛЮЧЕНИЕ; Включение операнда	Y, M, S	Y, M, без специальных меркеров	Разд. 4.13
RST		ОТКЛЮЧЕНИЕ; Отключение операнда	Y, M, S, D V, Z, T, C	Y, M: 1 D, V, Z, Спец. меркеры: 3 T, C: 2	Разд. 4.13

Табл. А-2: Обзор базовых команд (часть 2)

Инструкция	Символ релейной схемы	Значение	Операнды	Количество шагов программы	Раздел
PLS		ГЕНЕРАЦИЯ ИМПУЛЬСОВ; Генерация одного одноразового импульса по переднему фронту	Y, M	2	Разд. 4.14
PLF		ГЕНЕРАЦИЯ ИМПУЛЬСОВ; Генерация одного одноразового импульса по заднему фронту	Y, M	2	Разд. 4.14
INV		ИНВЕРСИЯ; Инверсирование результата логического выражения	–	1	Разд. 4.15
NOP	–	ПУСТАЯ СТРОКА; Пустая строка без выполнения функции	–	1	Разд. 4.16
END		END; Окончание программы пользователя ПК	–	1	Разд. 4.17

Табл. А-3:Обзор базовых команд (часть 3)

A.2 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELESEC FX1S

Признак	Технические данные
Обработка программы	Циклическая обработка записанной программы
Обработка входов/выходов	Обработка отображения процесса Имеется непосредственная обработка инструкций Входной фильтр настраивается от 0 до 15 мс
Язык программирования	Листинг инструкций и релейно-контактная схема по DIN 19 239 Шаговые инструкции
Время выполнения инструкции	Базовые команды: от 0,55 до 0,7 мкс Применяемые инструкции: см. приложение "B"
Объем программы	2к шагов, EEPROM-модуль
Количество инструкций	Набор базовых команд: 29 Инструкций управляющих шагов: 2 Применяемых инструкций: 85

Табл. А-4: Общие системные данные MELESEC FX1S

A.3 ОПЕРАНДЫ MELESEC FX1S

Признак	Технические данные			
Входы / Выходы	FX1S-10MR-ES/UL FX1S-10MR-DS FX1S-10MT-DSS	X0 – X5 Y0 – Y3	6 входов 4 выходы	
	FX1S-14MR-ES/UL FX1S-14MR-DS FX1S-14MT-DSS	X0 – X7 Y0 – Y5	8 входов 6 выходов	
	FX1S-20MR-ES/UL FX1S-20MR-DS FX1S-20MT-DSS	X0 – X13 Y0 – Y7	12 входов 8 выходов	
	FX1S-30MR-ES FX1S-30MR-DS FX1S-30MT-DSS	X0 – X17 Y0 – Y15	16 входов 14 выходов	
Меркеры	Меркеры	M0 – M383	384 адреса	
	Меркеры с памятью	M384 – M511	Действительное значение занесено в EEPROM	128 адресов
	Специальные меркеры	M8000 – M8254	256 адресов	
Состояния шагов	Инициализация	S0 – S9	10 адресов	
	Общее значение	S10 – S127	118 адресов	
Таймер	100 мс	0,1 – 3 276,7с	T0 – T62	63 адреса
	10 мс	0,1 – 327,67с	T32 – T62	31 адрес, если активен M8028
	1 мс	0,1 – 32,767с	T63	1 адрес
Счетчик	Счет вверх	16 бит +1 ... +32 767	Общее значение	C0 – C15 16 адресов
			Действительное значение занесено в EEPROM	C16 – C31 16 адресов

Табл. А-5: Операнды MELSEC FX1s (1)

Признак		Технические данные			
Высоко-скоростной счетчик	1-но фазный счетчик без пуска и возврата, счет вверх и вниз	32 бит	Действительное значение занесено в EEPROM	C235 – C240	6 счетчиков
	1-но фазный счетчик с пуском и возвратом, счет вверх и вниз	32 бит		C241 – C245	5 счетчиков
	2-х фазный счетчик, счет вверх и вниз	32 бит		C246– C250	5 счетчиков
	A/B фазный счетчик	32 бит		C251– C255	5 счетчиков
Регистр	Регистр данных	16 бит	Общее значение	D0 – D127	128 адресов
			Действительное значение занесено в EEPROM	D128 – D255	128 адресов
	Файловый регистр	Определяется параметром в трех блоках по 500 шагов программы		D1000 – D2499	1500 адресов
	Специальный регистр	16 бит		D8000 – D8255	256 адресов
	Внешне изменяемый регистр	16 бит, область значений от 0 до 255, выставка по VR1 и VR2		D8030, D8031	2 адреса
	Индексный регистр	16 бит		V, Z	16 адресов
Точка	Точка инструкции перехода			P0 – P63	64 адреса
	Точка прерывания	Входы-прерывания: X0...X3		I00* – I130*	6 адресов
Разветвление	Разветвление программы, главный контакт			N0 – N7	8 адресов
Константа	Десятичная	16 бит	-32 768 – +32 767		
		32 бит	-2 147 483 648 – +2 147 438 647		
	Шестнадцатеричная	16 бит	0 – FFFF _H		
		32 бит	0 – FFFFFFFF _H		

Табл. А-6:Операнды MELESEC FX1S (2)

A.4 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELESEC FX1N

Признак	Технические данные
Обработка программы	Циклическая обработка записанной программы
Обработка входов/выходов	Обработка отображения процесса Имеется непосредственная обработка инструкций Входной фильтр настраивается от 0 до 15 мс
Язык программирования	Листинг инструкций и релейно-контактная схема по DIN 19 239 Шаговые инструкции
Время выполнения инструкции	Базовые команды: от 0,55 до 0,7 мкс Применяемые инструкции: см. приложение "B"
Объем программы	8к шагов, EEPROM-модуль
Количество инструкций	Набор базовых команд: 29 Инструкций управляющих шагов: 2 Применяемых инструкций: 89

Табл. A-7: Общие системные данные MELESEC FX1N

A.5 ОПЕРАНДЫ MELESEC FX1N

Признак	Технические данные				
Входы / Выходы	FX1N-□□□-MR-DS FX1N-□□□-MR-ES/UL FX1N-□□□-MT-ESS/UL FX1N-□□□-MT-DSS	Максимальная конфигурация технических средств составляет 128 выходов/выходов в сумме. Программно можно адресовать максимально 128 входов и 128 выходов.			
Меркеры	Меркеры	M0 – M383		384 адреса	
	Меркеры с памятью	M384 – M1535	Действительное значение занесено в EEPROM	1152 адреса	
	Специальные меркеры	M8000 – M8255		256 адресов	
Состояния шагов	Инициализация	S0 – S9		10 адресов	
	Общее значение	S10 – S999		990 адресов	
Таймер	100 мс	0,1 – 3 276,7с	T0 – T199	200 адресов	
	10 мс	0,1 – 327,67с	T200 – T245	46 адресов	
	1 мс	0,001 – 32.767	T246 – T249	4 адреса	
	100 мс, с памятью	0 – 25,5с	T250 – T255	6 адресов	
Счетчик	Счет вверх	16 бит +1 ... +32 767	Общее значение	C0 – C15	16 адресов
			Действительное значение занесено в EEPROM	C16 – C199	184 адреса
	Счет вверх/вниз	32бита	Общее значение	C200 – C219	16 адресов
			Действительное значение занесено в EEPROM	C220 – C234	15 адресов

Табл. A-8: Операнды MELESEC FX1N (1)

Признак		Технические данные			
Высоко-скоростной счетчик	1-но фазный счетчик без пуска и возврата, счет вверх и вниз	32 бит	Действительное значение занесено в EEPROM	C235 – C240	6 адресов
	1-но фазный счетчик с пуском и возвратом, счет вверх и вниз	32 бит		C241 – C245	5 адресов
	2-х фазный счетчик счет вверх и вниз	32 бит		C246 – C250	5 адресов
	A/B фазный счетчик	32 бит		C251 – C255	5 адресов
Регистр	Регистр данных	16 бит	Общее значение	D0 – D127 D1000 – D7999	7128 адресов
			Действительное значение занесено в EEPROM	D128 – D999	872 адреса
	Файловый регистр	16 бит	Определяется параметром в 14-ти блоках по 500 шагов программы	D1000 – D7999	7000 адресов
	Специальный регистр	16 бит		D8000 – D8255	256 адресов
	Внешне изменяемый регистр	16 бит, область значений от 0 до 255, выставка по VR1 и VR2		D8030, D8031	2 адреса
	Индексный регистр	16 бит		V, Z	2 адреса
Точка	Точка инструкции перехода			P0 – P127	128 адресов
	Точка прерывания	Входы-прерывания: X0...X3		I00* – I130*	6 адресов
Разветвление	Разветвление программы, главный контакт			N0 – N7	8 адресов
Константа	Десятичная	16 бит		-32 768 ... +32 767	
		32 бит		-2 147 483 648 ... +2 147 438 647	
	Шестнадцатиричная	16 бит		0 ... FFFF _H	
		32 бит		0 ... FFFFFFFF _H	

Табл. А-9: Операнды MELSEC FX1N (2)

A.6 ИСПОЛЬЗУЕМЫЕ ИНСТРУКЦИИ MELSEC FX1S/FX1N

Классификация	Инструкция	FNC	Назначение	Раздел
Инструкции обработки программы	CJ	00	Переход внутри программы	6.2.1
	CALL	01	Вызов подпрограммы	6.2.2
	SRET	02	Конец подпрограммы	6.2.3
	IRET	03	Закончить прерывание программы	6.2.4
	EI	04	Активизировать прерывание программы	6.2.4
	DI	05	Деактивизировать прерывание программы	6.2.4
	FEND	06	Конец области программы	6.2.5
	WDT	07	Обновить значение таймера контроля скана	6.2.6
	FOR	08	Начало повторения программы	6.2.7
	NEXT	09	Конец повторения программы	6.2.7
Инструкции сравнения и переноса	CMP	10	Сравнение числовых данных	6.3.1
	ZCP	11	Сравнение числовых областей данных	6.3.2
	MOV	12	Передача данных	6.3.3
	BMOV	15	Передача блоков	6.3.6
	BCD	18	BCD-конвертирование	6.3.9
	BIN	19	BIN-конвертирование	6.3.10
Арифметические инструкции	ADD	20	Сложение числовых данных	6.4.1
	SUB	21	Вычитание числовых данных	6.4.2
	MUL	22	Умножение числовых данных	6.4.3
	DIV	23	Деление числовых данных	6.4.4
	INC	24	Инкрементация (приращение)	6.4.5
	DEC	25	Декрементация (уменьшение)	6.4.6
	WAND	26	Логическая связь "И" (UND)	6.4.7
	WOR	27	Логическая связь "ИЛИ" (ODER)	6.4.8
Инструкции сдвига	WXOR	28	Логическая связь "НЕ ИЛИ" (Exklusiv-ODER)	6.4.9
	SFTR	34	Сдвиг побитно двоичных данных вправо	6.5.5
	SFTL	35	Сдвиг побитно двоичных данных влево	6.5.5
	SFWR	38	Запись в накопитель типа FIFO	6.5.8
Операции с данными	SFRD	39	Чтение из накопителя типа FIFO	6.5.9
	ZRST	40	Отключить область операндов	6.6.1
	DECO	41	Декодирование данных	6.6.2
	ENCO	42	Кодирование данных	6.6.3

Табл. А-10: Обзор используемых инструкций FX1S/FX1N (1)

Классификация	Инструкция	FNC	Значение	Раздел
Высокоскоростные инструкции	REF	50	Обновление входов и выходов	6.7.1
	MTR	52	Чтение матрицы	6.7.3
	DHSCS	53	Включение по высокоскоростному счетчику	6.7.4
	DHSCR	54	Выключение по высокоскоростному счетчику	6.7.4
	SPD	56	Определение скорости	6.7.6
	PLSY	57	Выдача заданного числа импульсов	6.7.7
	PWM	58	Выдача импульсов с модуляцией их ширины	6.7.8
	PLSR	59	Выдача определенного числа импульсов	6.7.9
Инструкции ориентированные на пользователя	IST	60	Инициализация состояния шагов	6.8.1
	ABSD	62	Сравнение абсолютного счетчика	6.8.3
	INCD	63	Сравнение инкрементального счетчика	6.8.4
	ALT	66	Функция пульс-пары (Flip-Flop)	6.8.6
	RAMP	67	Функция рампы (наклона)	6.8.7

Табл. А-11: Обзор используемых инструкций FX1S/FX1N (2)

УКАЗАНИЕ

| Используемые инструкции FNC 70 до FNC 246 описаны в главе 7.

A.7 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELSEC FX2N/FX2NC

Признак	Технические данные
Обработка программы	Циклическая обработка записанной программы
Обработка входов/выходов	Обработка отображения процесса Имеется непосредственная обработка инструкций Входной фильтр настраивается от 0 до 15 мс
Язык программирования	Листинг инструкций и релейно-контактная схема по DIN 19 239
Время выполнения инструкции	Базовые команды: от 0,08 мкс Применяемые инструкции: см. приложение "B"
Объем программы	8к шагов: внутренняя память RAM 16к шагов: RAM-, EEPROM-кассета (опция)
Количество инструкций	Набор базовых команд: 29 Инструкций управляющих шагов STL: 2 Применяемых инструкций: 125

Табл. A-12: Общие системные данные MELSEC FX2N и FX2NC

A.8 ОПЕРАНДЫ MELSEC FX2N/FX2NC

Признак		Технические данные			
Входы / Выходы	FX2N-□□□-MR-DS	Максимальная конфигурация технических средств составляет 255 адресов входов/выходов в сумме. Программно можно адресовать максимально 255 входов и 255 выходов.			
	FX2N-□□□-MR-ES/UL				
Меркеры	FX2N-□□□-MT-ESS/UL				
	FX2N-□□□-MT-DSS				
	Меркеры	M0 – M3071	3072 адреса		
Состояния шагов	Меркеры с памятью	M500 – M3071	2572 адреса (разделенные)		
	Специальные меркеры	M8000 – M8255	256 адресов		
	Инициализация	S0 – S9	10 адресов (разделенные)		
	Общее значение	S0 – S999	1000 адресов		
Таймер	Меркеры с памятью	S500 – S999	500 адресов (разделенные)		
	Меркеры ошибки	S900 – S999	100 адресов		
	100 мс	0 – 3 276,7 с	T0 – T199	200 адресов	
	10 мс	0 – 327,67 с	T200 – T245	46 адресов	
Счетчик	1 мс, с памятью	0 – 32.767 с	T246 – T249	4 адреса	
	100 мс, с памятью	0 – 3276,7 с	T250 – T255	6 адресов	
	Счет вверх, 16 бит	+1 – +32 767	Общее значение	C0 – C199	200 адресов
			Действительное значение занесено в EEPROM	C100 – C199	100 адресов (разделенные)
Счет вверх, 32 бит	+1 – +214748367	Общее значение	C200 – C234	35 адресов	
		Действительное значение занесено в EEPROM	C219 – C234	15 адресов (разделенные)	

Табл. A-13: Операнды MELSEC FX2N и FX2NC (1)

Признак		Технические данные			
Высоко-скоростной счетчик	1-но фазный счетчик	-2147483648 – +2147483647	Действительное значение занесено в EEPROM. Частота счета всех счетчиков ≤ 20 kHz	C235 – C240	6 адресов
	1-но фазный счетчик с входом пуска и возврата			C241 – C245	5 адресов
	2-х фазный счетчик			C246 – C250	5 адресов
	A/B фазный счетчик			C251 – C255	5 адресов
Регистр	Регистр данных	16 бит	Общее значение С памятью	D0 – D7999	8000 адресов
				D200 – D7999	7800 адресов (разделенные)
	Файловый регистр	16 бит	Определяется параметром в 14-ти блоках по 500 шагов программы	D1000 – D7999	7000 адресов
	Специальный регистр	16 бит		D8000 – D8255	256 адресов
	Индексный регистр	16 бит		V0 – V7, Z0 – Z7	16 адресов
Точка	Точка инструкции перехода			P0 – P63	128 адресов
	Точка прерывания □ = 1 (передний фронт) □ = 0 (задний фронт) ** = время в мс	Входы-прерывания: X0-X3		I00□ – I50□	6 адресов
		Прерывание-время		I6** – I8**	3 адреса
		Прерывание-счет		I010 – I060	6 адресов
Разветвление	Разветвление программы, главный контакт			N0 – N7	8 адресов
Константа	Десятичная	16 бит		-32 768 ... +32 767	
		32 бит		-2 147 483 648 ... +2 147 438 647	
	Шестнадцатеричная	16 бит		0 ... FFFF _H	
		32 бит		0 ... FFFFFFFF _H	

Табл. А-14: Операнды MELSEC FX2N и FX2NC (2)

A.9 ИСПОЛЬЗУЕМЫЕ ИНСТРУКЦИИ MELSEC FX2N/FX2NC

Классификация	Инструкция	FNC	Значение	Раздел
Инструкции обработки программы	CJ	00	Переход внутри программы	6.2.1
	CALL	01	Вызов подпрограммы	6.2.2
	SRET	02	Конец подпрограммы	6.2.3
	IRET	03	Закончить прерывание программы	6.2.4
	EI	04	Активизировать прерывание программы	6.2.4
	DI	05	Деактивизировать прерывание программы	6.2.4
	FEND	06	Конец области программы	6.2.5
	WDT	07	Обновить значение таймера контроля скана	6.2.6
	FOR	08	Начало повторения программы	6.2.7
	NEXT	09	Конец повторения программы	6.2.7
Инструкции сравнения и переноса	CMP	10	Сравнение числовых данных	6.3.1
	ZCP	11	Сравнение числовых областей данных	6.3.2
	MOV	12	Передача данных	6.3.3
	SMOV	13	Сдвиг и передача данных	6.3.4
	CML	14	Копирование и инвертирование	6.3.5
	BMOV	15	Передача блоков	6.3.6
	FMOV	16	Передача одинаковых данных	6.3.7
	XCH	17	Обмен данными	6.3.8
	BCD	18	BCD-конвертирование	6.3.9
	BIN	19	BIN-конвертирование	6.3.10
Арифметические инструкции	ADD	20	Сложение числовых данных	6.4.1
	SUB	21	Вычитание числовых данных	6.4.2
	MUL	22	Умножение числовых данных	6.4.3
	DIV	23	Деление числовых данных	6.4.4
	INC	24	Инкрементация (приращение)	6.4.5
	DEC	25	Декрементация (уменьшение)	6.4.6
	AND	26	Логическая связь "И" (UND)	6.4.7
	OR	27	Логическая связь "ИЛИ" (ODER)	6.4.8
	XOR	28	Логическая связь "НЕ ИЛИ" (Exklusiv-ODER)	6.4.9
	NEG	29	Отрицание данных	6.4.10
Инструкции сдвига	ROR	30	Ротация (циклический сдвиг) вправо	6.5.1
	ROL	31	Ротация (циклический сдвиг) влево	6.5.2
	RCR	32	Ротация побитно вправо	6.5.3
	RCL	33	Ротация побитно влево	6.5.4
	SFTR	34	Сдвиг побитно двоичных данных вправо	6.5.5
	SFTL	35	Сдвиг побитно двоичных данных влево	6.5.5
	WSFR	36	Сдвиг данных пословно вправо	6.5.6
	WSFL	37	Сдвиг данных пословно влево	6.5.7
	SFWR	38	Запись в накопитель типа FIFO	6.5.8
	SFRD	39	Чтение из накопителя типа FIFO	6.5.9

Табл. А-15: Обзор используемых инструкций FX2N и FX2NC (1)

Классификация	Инструкция	FNC	Значение	Раздел
Операции с данными	ZRST	40	Отключить область операндов	6.6.1
	DECO	41	Декодирование данных	6.6.2
	ENCO	42	Кодирование данных	6.6.3
	SUM	43	Определение использованного бита	6.6.4
	BON	44	Проверка бита	6.6.5
	MEAN	45	Расчет среднего значения числа	6.6.6
	ANS	46	Запуск интервала времени	6.6.7
	ANR	47	Отключение бита отображения	6.6.8
	SQR	48	Вычисление корня квадратного	6.6.9
	FLT	49	Преобразование формата числа	6.6.10
Высокоскоростные инструкции	REF	50	Обновление входов и выходов	6.7.1
	REFF	51	Установка входного фильтра	6.7.2
	MTR	52	Чтение матрицы (MTR)	6.7.3
	DHSCS	53	Включение по высокоскоростному счетчику	6.7.4
	DHSCR	54	Выключение по высокоскоростному счетчику	6.7.4
	DHSZ	55	Сравнение областей	6.7.5
	SPD	56	Определение скорости	6.7.6
	PLSY	57	Выдача заданного числа импульсов	6.7.7
	PWM	58	Выдача импульсов с модуляцией их ширины	6.7.8
	PLSR	59	Выдача определенного числа импульсов	6.7.9
Инструкции ориентированные на пользователя	IST	60	Инициализация состояния шагов	6.8.1
	SER	61	Инструкция поиска	6.8.2
	ABSD	62	Сравнение абсолютного счетчика	6.8.3
	INCD	63	Сравнение инкрементального счетчика	6.8.4
	TTMR	64	Таймер обучения	6.8.5
	STMR	65	Специальный таймер	6.8.6
	ALT	66	Функция пульс-пары (Flip-Flop)	6.8.7
	RAMP	67	Функция рампы (наклона)	6.8.8
	ROTC	68	Позиционирование поворотного стола	6.8.9
	SORT	69	Инструкция сортировки	6.8.10

Табл. А-16: Обзор используемых инструкций FX2N и FX2NC (2)

УКАЗАНИЕ

| Используемые инструкции FNC 70 до FNC 246 описаны в главе 7.

A.10 ОБЩИЕ СИСТЕМНЫЕ ДАННЫЕ MELSEC FX3U

Признак	Технические данные
Обработка программы	циклическая обработка хранящейся в памяти программы
Обработка входов-выходов	обработка отображения процесса имеются команды непосредственной обработки входной фильтр настраивается в диапазоне от 0 до 15 мс
Язык программирования	набор команд контроллера по стандарту DIN 19239
Время выполнения команд	базовая команда: 0,065 мкс прикладная команда: см. приложение В
Объем программы	64 тысячи шагов: внутренняя память RAM 64 тысячи шагов: кассета RAM, EEPROM (опция)
Количество команд	базовый набор команд: 27 команды STL: 2 прикладные команды: 209

Табл. А-17:Общие данные MELSEC FX3U

A.11 ОПЕРАНДЫ MELSEC FX3U

Признак	Технические данные				
Входы и выходы	В базовом блоке и расширительных модулях возможна адресация максимум к 248 входам и 248 выходам (с X000 по X367 и с Y000 по Y367). Однако сумма входов и выходов в базовых и расширительных блоках не должна превышать 256. Дополнительно можно обращаться к 224 входам-выходам в сети CC-Link или 248 входам-выходам в сети ASI. Сумма входов и выходов в базовых блоках, расширительных блоках и сети не должна превышать 384.				
Маркеры	маркеры	M0 – M7679	7680 адресов		
	фиксируемые маркеры	M500 – M7679	7180 адресов (по частям)		
	специальные маркеры	M8000 – M8511	512 адресов		
Состояние шага	инициализация	S0 – S9	10 адресов (по частям)		
	общие регистры	S10 – S499	490 адресов		
	фиксируемые маркеры (переменные)	S500 – S899	400 адресов (по частям)		
	маркеры ошибок	S900 – S999	100 адресов		
Таймеры	100 ms	0 – 3276,7 s	T0 – T199	200 адресов	
	10 ms	0 – 327,67 s	T200 – T245	46 адресов	
	1 мс (остаточный)	0 – 32,767 s	T246 – T249	4 адреса	
	100 мс (остаточный)	0 – 3276,7 s	T250 – T255	6 адресов	
	1 ms	0 – 32,767 s	T256 – T511	256 адресов	
Счетчики	возрастающий счет, 16 битов	диапазон счета: от +1 до +32 767	общие	C0 – C199	200 адресов
			фактическое значение сохраняется в EEPROM	C100 – C199	100 адресов (по частям)
	с возрастающим и убывающим счетом, 32 бита	диапазон счета: от -2147483648 до +2147483647	общие	C200 – C234	35 адресов
			фактическое значение сохраняется в EEPROM	C219 – C234	15 адресов (по частям)

Табл. А-18:Операнды MELSEC FX3U (1)

Признак		Технические данные			
Высокоскоростные счетчики	1-фазный счетчик	диапазон счета: -2147483648 ... +2147483647	фактическое значение сохраняется в EEPROM	C235 – C240	6 адресов
	1-фазный счетчик с запускающим и сбрасывающим входом			C241 – C245	5 адресов
	2-фазный счетчик			C246 – C250	5 адресов
	счетчик фаз A/B			C251 – C255	5 адресов
Регистры (по 2 регистра можно объединять в 32-битный регистр)	регистры данных	16 битов	16 битов	D0 – D7999	8000 адресов
			общие	D200 – D7999	7800 адресов (по частям)
	регистры файлов	16 битов	устанавливаются с помощью параметров блоками по 500 адресов	D512 – D7999	7488 адресов
	специальные регистры	16 битов		D8000 – D8511	512 адресов
	индексные регистры	16 битов		V0 – V7, Z0 – Z7	16 адресов
Расширенные регистры		16 битов	содержимое сохраняется при исчезновении напряжения	R0 ... R32767	32768 адресов
Расширенные регистры файлов		16 битов	использование возможно только при установленной кассете памяти	ER0 ... ER32767	32768 адресов
Указатели	указатель команды перехода			P0 – P4095	4096 адресов
	указатель прерывания =1 (положительный фронт) =0 (отрицательный фронт) **= время в мс	прерывание по входу: X0 - X5		I00□ – I50□	6 адресов
		прерывание по таймеру		I6** – I8**	3 адреса
	прерывание по счетчику		I010 – I060	6 адресов	
Вложения	разветвление программы, главный контакт			N0 – N7	8 адресов
Константы	десятичные	16 битов		-32 768 ... +32 767	
		32 бита		-2 147 483 648 ... +2 147 438 647	
	шестнадцатеричные	16 битов		0 ... FFFF _H	
		32 бита		0 ... FFFFFFFF _H	
	числа с плавающей запятой	32 бита		-1,0 x 2 ¹²⁸ ... -1,0 x 2 ⁻¹²⁶ 0 1,0 x 2 ⁻¹²⁶ ... -1,0 x 2 ⁺¹²⁸	
	строковые величины	Строковые величины в программе обозначаются кавычками (например, "MITSUBISHI") Можно указать до 32 знаков, из которых каждый занимает один байт.			

Табл. А-19: Операнды MELSEC FX3U (2)

A.12 ПРИКЛАДНЫЕ КОМАНДЫ MELSEC FX3U

Классификация	Инструкция	FNC	Значение	Раздел
Инструкции обработки программы	CJ	00	Переход внутри программы	6.2.1
	CALL	01	Вызов подпрограммы	6.2.2
	SRET	02	Конец подпрограммы	6.2.3
	IRET	03	Закончить прерывание программы	6.2.4
	EI	04	Активизировать прерывание программы	6.2.4
	DI	05	Деактивизировать прерывание программы	6.2.4
	FEND	06	Конец области программы	6.2.5
	WDT	07	Обновить значение таймера контроля скана	6.2.6
	FOR	08	Начало повторения программы	6.2.7
	NEXT	09	Конец повторения программы	6.2.7
Инструкции сравнения и переноса	CMP	10	Сравнение числовых данных	6.3.1
	ZCP	11	Сравнение числовых областей данных	6.3.2
	MOV	12	Передача данных	6.3.3
	SMOV	13	Сдвиг и передача данных	6.3.4
	CML	14	Копирование и инвертирование	6.3.5
	BMOV	15	Передача блоков	6.3.6
	FMOV	16	Передача одинаковых данных	6.3.7
	XCH	17	Обмен данными	6.3.8
	BCD	18	BCD-конвертирование	6.3.9
	BIN	19	BIN-конвертирование	6.3.10
Арифметические инструкции	ADD	20	Сложение числовых данных	6.4.1
	SUB	21	Вычитание числовых данных	6.4.2
	MUL	22	Умножение числовых данных	6.4.3
	DIV	23	Деление числовых данных	6.4.4
	INC	24	Инкрементация (приращение)	6.4.5
	DEC	25	Декрементация (уменьшение)	6.4.6
	AND	26	Логическая связь "И" (UND)	6.4.7
	OR	27	Логическая связь "ИЛИ" (ODER)	6.4.8
	XOR	28	Логическая связь "НЕ ИЛИ" (Exklusiv-ODER)	6.4.9
	NEG	29	Отрицание данных	6.4.10
Инструкции сдвига	ROR	30	Ротация (циклический сдвиг) вправо	6.5.1
	ROL	31	Ротация (циклический сдвиг) влево	6.5.2
	RCR	32	Ротация побитно вправо	6.5.3
	RCL	33	Ротация побитно влево	6.5.4
	SFTR	34	Сдвиг побитно двоичных данных вправо	6.5.5
	SFTL	35	Сдвиг побитно двоичных данных влево	6.5.5
	WSFR	36	Сдвиг данных пословно вправо	6.5.6
	WSFL	37	Сдвиг данных пословно влево	6.5.7
	SFWR	38	Запись в накопитель типа FIFO	6.5.8
	SFRD	39	Чтение из накопителя типа FIFO	6.5.9

Табл. A-20: Обзор используемых инструкций FX2N и FX2NC (1)

Классификация	Инструкция	FNC	Значение	Раздел
Операции с данными	ZRST	40	Отключить область операндов	6.6.1
	DECO	41	Декодирование данных	6.6.2
	ENCO	42	Кодирование данных	6.6.3
	SUM	43	Определение использованного бита	6.6.4
	BON	44	Проверка бита	6.6.5
	MEAN	45	Расчет среднего значения числа	6.6.6
	ANS	46	Запуск интервала времени	6.6.7
	ANR	47	Отключение бита отображения	6.6.8
	SQR	48	Вычисление корня квадратного	6.6.9
	FLT	49	Преобразование формата числа	6.6.10
Высокоскоростные инструкции	REF	50	Обновление входов и выходов	6.7.1
	REFF	51	Установка входного фильтра	6.7.2
	MTR	52	Чтение матрицы (MTR)	6.7.3
	DHSCS	53	Включение по высокоскоростному счетчику	6.7.4
	DHSCR	54	Выключение по высокоскоростному счетчику	6.7.4
	DHSZ	55	Сравнение областей	6.7.5
	SPD	56	Определение скорости	6.7.6
	PLSY	57	Выдача заданного числа импульсов	6.7.7
	PWM	58	Выдача импульсов с модуляцией их ширины	6.7.8
	PLSR	59	Выдача определенного числа импульсов	6.7.9
Инструкции ориентированные на пользователя	IST	60	Инициализация состояния шагов	6.8.1
	SER	61	Инструкция поиска	6.8.2
	ABSD	62	Сравнение абсолютного счетчика	6.8.3
	INCD	63	Сравнение инкрементального счетчика	6.8.4
	TTMR	64	Таймер обучения	6.8.5
	STMR	65	Специальный таймер	6.8.6
	ALT	66	Функция пульс-пары (Flip-Flop)	6.8.7
	RAMP	67	Функция рампы (наклона)	6.8.8
	ROTC	68	Позиционирование поворотного стола	6.8.9
SORT	69	Инструкция сортировки	6.8.10	

Табл. А-21:Общий обзор прикладных команд FX3U (2)

УКАЗАНИЕ

| Прикладные команды с FNC 70 по FNC 299 описаны в разделе 7.

В ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ

В.1 ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ ДЛЯ ПК СЕРИЙ FX1S/FX1N

ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ ДЛЯ ПК СЕРИЙ FX1S/FX1N

В.1.1 БАЗОВЫЕ КОМАНДЫ И ИНСТРУКЦИИ ШАГОВЫХ СОСТОЯНИЙ

Инструкция	Значение	Операнды	Количество шагов программы	Время выполнения [мкс]	
				Время включения	Время отключения
LD	Начало логического выражения (связей) с опросом на состояние сигнала "1"	X, Y, M, S, T, C, Спец. меркеры	1	0,7	
LDI	Начало логического выражения (связей) с опросом на состояние сигнала "0"				
AND	Логическое выражение "И" с опросом на состояние сигнала "1"		1	0,65	
ANI	Логическое выражение "И" с опросом на состояние сигнала "0"				
OR	Логическое выражение "ИЛИ" с опросом на состояние сигнала "1"				
ORI	Логическое выражение "ИЛИ" с опросом на состояние сигнала "0"				
LDP	Начало логического выражения с опросом по переднему фронту (импульс)	X, Y, M, S, T, C	1	11,7	
LDF	Начало логического выражения с опросом по заднему фронту (импульс)				
ANDP	Логическое выражение "И" с опросом по переднему фронту (импульс)				
ANDF	Логическое выражение "И" с опросом по заднему фронту (импульс)				
ORP	Логическое выражение "ИЛИ" с опросом по переднему фронту (импульс)				
ORF	Логическое выражение "ИЛИ" с опросом по заднему фронту (импульс)				
ANB	Блок "И", последовательное включение параллельных цепей	-	1	0,55	
ORB	Блок "ИЛИ", параллельное включение последовательных цепей	-	1	0,55	
MPS	Сохранение результата логического выражения	-	1	0,5	
MRD	Чтение результата логического выражения	-		0,55	
MPP	Чтение и стирание результата логического выражения	-		0,5	

Табл. В-1: Базовые команды и инструкции шаговых состояний для ПК серии FX1S/FX1N (1)

Инструкция	Значение	Операнды	Количество шагов программы	Время выполнения [мкс]	
				Время включения	Время отключения
INV	Инверсия. Изменение на противоположный результат логического выражения	–	1	0,08	
MC	Включение условий контроля	N - Y, M	3	8,6	8,0
MCR	Отключение условий контроля	N (Разветвление)	2	4,1	
NOP	Пустая строка	–	1	0,45	
END	Конец программы		1	450	
STL	Выполнение шагового состояния	S	1	15,8 + 8,2n ^①	
RET	Окончание шагового состояния	–	1	4,8	
OUT	Выдача. Присвоение результата логического выражения	Y, M	1	0,7	
		S	2	4,4	
		Спец. меркер	2	2,8	
		T-K	3	11,2	10,2
		T-D	3	12,2	11,2
		C-K (16 бит)	3	8,1	6,9
		C-D (16 бит)	3	9,5	8,0
		C-K (32 бит)	5	8,1	6,8
C-D (32 бит)	5	9,5	8,0		
SET	Включение. Включение операнда	Y, M	1	0,85	
		S	2	4,2	2,4
		S при применении в ступени STL ^①		18,6 + 6,8n	2,4
		Спец. меркер	2	2,8	
RST	Отключение. Отключение операнда	Y, M	1	0,85	
		S	2	3,8	2,4
		Спец. меркер	2	2,8	
		T, C	2	8,7	7,3
		D, V, Z, Спец. регистр	3	3,8	1,1
PLS	Создание импульса по переднему фронту	Y, M	2	10,8	
PLF	Создание импульса по заднему фронту	Y, M	2	10,8	
P	Адресация точки	0 ... 63	1	0,45	
I	Адресация точки прерывания	I□□□	1	0,45	

Табл. В-2: Базовые команды и инструкции шаговых состояний для ПК серии FX1S/FX1N (2)

0

В.1.2 ИНСТРУКЦИИ РАЗВЕТВЛЕНИЯ

Инструкция		Время выполнения [мкс]		
Символ	FNC-инструкция	Время включения	Время отключения	Специальная функция FNC
CJ	FNC 00 /S+	7,1	6,4	—
CALL	FNC 01 /S+	9,3	3,2	—
SRET	FNC 02	8,3	—	—
IRET	FNC 03 ②	8,1	—	—
EI	FNC 04 ②	6,0	—	—
DI	FNC 05 ②	5,3	—	—
FEND	FNC 06 ②	450		—
WDT	FNC 07	3,7	2,7	—
FOR	FNC 08 /S+ ②	7,5	—	—
NEXT	FNC 09 ②	4,6	—	—

Табл. В-3: Инструкции разветвления программы FX1S/FX1N

В.1.3 ИНСТРУКЦИИ СРАВНЕНИЯ И ПЕРЕНОСА

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
CMP	FNC 10 /S1+ /S2+ /D+	16	40	2,5	—
		32	41	4,5	—
ZCP	FNC 11 /S1+ /S2+ /S+ /D+	16	45	2,5	—
		32	47	4,5	—
MOV	FNC 12 /S+ /D+	16	19	2,5	—
		32	22	3,0	—
BMOV	FNC 15 /S+ /D+ /n ④	16	78 + 22n	2,5	—
BCD	FNC 18 /S+ /D+	16	30	2,5	—
		32	38,6	3,0	—
BIN	FNC 19 /S+ /D+	16	30	2,5	—
		32	35,5	3,0	—

Табл. В-4: Инструкции сравнения и переноса FX1S/FX1N

В.1.4 АРИФМЕТИЧЕСКИЕ ИНСТРУКЦИИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	37,5	2,5	—
		32	40,2	4,5	224
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	37,5	2,5	—
		32	40,5	4,5	232
MUL	FNC 22 /S1+ /S2+ /D+	16	38,2	2,5	—
		32	50,3	4,5	162
DIV	FNC 23 /S1+ /S2+ /D+	16	39,2	2,5	—
		32	63,5	4,5	197
INC	FNC 24 /D+ ③	16	14,5	2,5	—
		32	16,7	4,5	—
DEC	FNC 25 /D+ ③	16	14,5	2,5	—
		32	16,7	4,5	—
WAND	FNC 26 /S1+ /S2+ /D+	16	35,7	2,5	—
		32	37,3	4,5	—
WOR	FNC 27 /S1+ /S2+ /D+	16	35,7	2,5	—
		32	37,3	4,5	—
WXOR	FNC 28 /S1+ /S2+ /D+	16	35,7	2,5	—
		32	37,3	4,5	—

Табл. В-5: Арифметические инструкции FX1S/FX1N

Br (Borrow): M8021
 Cy (Carry): M8022
 F (Инструкция полностью обработана): M8029

В.1.5 ИНСТРУКЦИИ РОТАЦИИ И СДВИГА

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
SFTR	FNC 34 /S+ /D+ /n1 /n2 ③ ⑥	16	55 + 1,25n	2,5	—
SFTL	FNC 35 /S+ /D+ /n1 /n2 ③ ⑥	16	56,1 + 1,25n	2,5	—
SFWR	FNC 38 /S+ /D+ /n ③ ⑦	16	41,6	2,5	—
SFRD	FNC 39 /S+ /D+ /n ③ ⑦	16	52,3	2,5	—

Табл. В-6: Инструкции ротации и сдвига FX1S/FX1N

В.1.6 ОПЕРАЦИИ С ДАННЫМИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	32,4 + 0,5n	2,5	—
		16 (S)	37,8 + 0,9n		
		16 (C)			
		16 (T)			
		16 (M)	51,8 + 0,8n		
16 (Y)					
DECO	FNC 41 /S+ /D+ /n	16	65,6	2,5	—
ENCO	FNC 42 /S+ /D+ /n	16	46,7	2,5	—

Табл. В-7: Операции с данными FX1S/FX1N

В.1.7 ВЫСОКОСКОРОСТНЫЕ ИНСТРУКЦИИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
REF	FNC 50 /D /n ⑩	16	19,5 + 4,3n	2,5	—
MTR	FNC 52 /S /D1 /D2 /n	16	22,6	9,8	—
HSCS	FNC 53 /S1+ /S2+ /D+ ⑫	32	46,8	4,5	—
HSCR	FNC 54 /S1+ /S2+ /D+ ⑫	32	46,8	4,5	—
SPD	FNC 56 /S1+ /S2+ /D+	②	39,5	43,8	—
		16	82,6	22,8	—
PLSY	FNC 57 /S1+ /S2+ /D+	32	100,6	34,9	—
		16	38,7	42,6	—
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	91,6	27,8	—
		32	113,7	41,6	

Табл. В-8:Высокоскоростные инструкции FX1S/FX1N

В.1.8 ИНСТРУКЦИЙ ОРИЕНТИРОВАННЫЕ НА ПОЛЬЗОВАТЕЛЯ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
IST	FNC 60 /S /D1 /D2	16	81,7	2,5	—
ABSD	FNC 62 /S1+ /S2+ /D+/ n ¹⁴	16	56,5 + 6,3n	2,5	—
		32	62,7 + 11n	2,5	
INCD	FNC 63 /S1+ /S2+ /D+/ n	16	60,5	52,7	—
ALT	FNC 66 /D+	16	21,8	2,5	—
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	52,5	44,8	—

Табл. В-9: Инструкции ориентированные на пользователя FX1S/FX1N

В.1.9 СПЕЦИАЛЬНЫЕ FNC-ИНСТРУКЦИИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	95,0	92,6	—
SEGL	FNC 74 /S+ /D+ /n	16 1Set	84,5	40,7	—
FROM (только FX1N)	FNC 78 /n1 /n2 /D+ /n3 ¹⁶	16	87 + 483n	2,5	—
		32	102 + 973n	4,5	
TO (только FX1N)	FNC 79 /n1 /n2 /S+ /n3 ¹⁶	16	85 + 542n	2,5	—
		32	98 + 1121n	4,5	
RS	FNC 80	16	56,3	9,2	—
PRUN	FNC 81 /S+ /D+ ¹⁷	16	46,7 + 1,0n	2,5	—
		32	47,7 + 1,0n	3,0	—
ASCI	FNC 82	16	52,8 + 5,8n	2,5	—
HEX	FNC 83	16	54 + 8,9n	2,5	—
CCD	FNC 84	16	54,3 + 4,5n	2,5	—
VRRD	FNC 85 /S+ /D+	16	142,7	8,9	—
VRSC	FNC 86 /S+ /D+	16	142,7	8,9	—
PID	FNC 88	16	65,5	8,5	—

Табл. В-10: Специальные FNC-инструкции FX1S/FX1N

В.1.10 ИНСТРУКЦИИ ПОЗИЦИОНИРОВАНИЯ

Инструкция		Бит	Время выполнения [мкс]	
Символ	FNC-инструкция		Время включения	Время отключения
ABS	FNC 155 /S+ /D1+ /D2+	32	86,7	85,7
ZRN	FNC 156 /S1+ /S2+ /S3+ /D	16	107,8	27,8
		32	130,5	40,8
PLSV	FNC 157 /S+ /D1 /D2+	16	79,6	22,7
		32	97,8	33,5
DRVI	FNC 158 /S1+ /S2+ /D1 /D2+	16	87,7	26,8
		32	110,6	40,7
DRVA	FNC 158 /S1+ /S2+ /D1 /D2+	16	89,6	26,8
		32	112,7	40,7

Табл. В-13: Инструкции позиционирования FX1S/FX1N

В.1.11 ИНСТРУКЦИИ РЕАЛЬНОГО ВРЕМЕНИ

Инструкция		Бит	Время выполнения [мкс]		
Символ	FNC-инструкция		Время включения	Время отключения	Специальная функция FNC
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	52,6	2,5	—
TZCP	FNC 161 /S1+ /S2+ /S+ /D+	16	64,7	2,5	—
TADD	FNC 162 /S1+ /S2+ /D+	16	42,9	2,5	—
TSUB	FNC 163 /S1+ /S2+ /D+	16	42,9	2,5	—
TRD	FNC 166 /D+	16	29,7	2,5	—
TWR	FNC 167 /S+	16	633,5	2,5	—
		32	41,9	40,6	—
HOUR	FNC 169 /S1+ /D1+ /D2+	16	39,7	38,7	—

Табл. В-11: Инструкции реального времени

В.1.12 ОБМЕН ДАННЫМИ С ВХОДНЫМИ АНАЛОГОВЫМИ МОДУЛЯМИ FXON-3A

Инструкция		Бит	Время выполнения [мкс]		
Символ	FNC-инструкция		Время включения	Время отключения	Специальная функция FNC
RD3A (только FX1N)	FNC 176 /M1+ /M2+ /D+	16	1248,3	7,5	—
WR3A (только FX1N)	FNC 177 /M1+ /M2+ /D+	16	1263,7	7,5	—

Табл. В-12: Инструкции FX1N для коммуникации с FXON-3A

В.1.13 ИНСТРУКЦИИ СРАВНЕНИЯ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
LD□	FNC 221 – 230 /S1+ /S2+	16	27,6		—
		32	28,2		
AND□	FNC 232 – 238 /S1+ /S2+	16	27,6		—
		32	28,2		
OR□	FNC 240 – 246 /S1+ /S2+	16	27,6		—
		32	28,2		

Табл. В-14: Инструкции сравнения

- ① “n” указывает число статических STL-инструкций (число параллельных/сходящихся инструкций).
- ② Характеристика инструкций, не требующих никаких контактов.
- ③ Если применяется статическая команда, а не импульсная, значение целевого адреса изменяется циклически.
- ④ “n” указывает число сдвигаемых регистров ($n \leq 512$).
- ⑥ “n” указывает число обрабатываемых бит-операндов.
- ⑦ “n” указывает число обрабатываемых операндов ($2 \leq n \leq 512$).
- ⑧ “n” указывает отключаемую область операндов.
Тип операнда указывается в скобках.
- ⑩ “n” указывает активизированную область операндов ($8 \leq n \leq 128$ в восьми шагах).
- ⑫ Эта инструкция может быть активной максимум 6 раз в течении одного и того же времени.
- ⑭ “n” указывает число адресов выходов ($n \leq 64$).
- ⑯ “n” указывает число записываемых или читаемых слов данных в или из специального модуля.
- ⑰ “n” указывает число записываемых или читаемых байтовых сигнальных слов данных (8 бит) при параллельной работе двух ПК серии FX.

B.2 ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ ДЛЯ ПК СЕРИИ FX2N/FX2NC

Описание сносок находится в конце раздела

B.2.1 БАЗОВЫЕ КОМАНДЫ И ИНСТРУКЦИИ ШАГОВЫХ СОСТОЯНИЙ

Инструкция	Значение	Операнды	Количество шагов программы	Время выполнения [мкс]	
				Время включения	Время отключения
LD	Начало логического выражения (связей) с опросом на состояние сигнала "1"	X, Y, M, S, T, C, Спец. меркеры	1	0,08	
LDI	Начало логического выражения (связей) с опросом на состояние сигнала "0"				
AND	Логическое выражение "И" с опросом на состояние сигнала "1"				
ANI	Логическое выражение "И" с опросом на состояние сигнала "0"				
OR	Логическое выражение "ИЛИ" с опросом на состояние сигнала "1"				
ORI	Логическое выражение "ИЛИ" с опросом на состояние сигнала "0"				
LDP	Начало логического выражения с опросом по переднему фронту (импульс)	X, Y, M, S, T, C	1	43,2	37,4
LDF	Начало логического выражения с опросом по заднему фронту (импульс)				
ANDP	Логическое выражение "И" с опросом по переднему фронту (импульс)				
ANDF	Логическое выражение "И" с опросом по заднему фронту (импульс)				
ORP	Логическое выражение "ИЛИ" с опросом по переднему фронту (импульс)				
ORF	Логическое выражение "ИЛИ" с опросом по заднему фронту (импульс)				
ANB	Блок "И", последовательное включение параллельных цепей	—	1	0,08	
ORB	Блок "ИЛИ", параллельное включение последовательных цепей				
MPS	Запоминание результата обработки логического выражения				
MRD	Чтение результата обработки логического выражения				
MPP	Чтение и стирание результата обработки логического выражения				

Табл. B-15: Базовые команды и инструкции шаговых состояний для ПК серии FX2N и FX2NC (1)

Инструкция	Значение	Операнды	Количество шагов программы	Время выполнения [мкс]	
				Время включения	Время отключения
INV	Инверсия. Изменение на противоположный результат логического выражения	—	1	0,08	
MC	Включение условий контроля	N - Y, M	3	24,8	27,5
MCR	Отключение условий контроля	N (Разветвление)	2	20,8	
NOP	Пустая строка	—	1	0,08	
END	Конец программы		1	508	
STL	Выполнение шагового состояния	S	1	27,3 + 12,6n ①	
RET	Окончание шагового состояния	—	1	21,6	
OUT	Выдача. Присвоение результата логического выражения	Y, M	1	0,08	
		S	2	24,4	24,3
		Спец. меркер	2	0,16	0,16
		T-K	3	42,3	37,4
		T-D	3	42,2	37,2
		C-K (16 бит)	3	25,5	24,9
		C-D (16 бит)	3	25,3	25,0
		C-K (32 бит)	5	25,3	24,9
		C-D (32 бит)	5	25,2	24,9
SET	Включение. Включение операнда	Y, M	1	0,08	
		S	2	23,7	17,2
		S при применении в ступени STL ①		27,3 + 12,6n	
		Спец. меркер	2	0,16	0,16
RST	Отключение. Отключение операнда	Y, M	1	0,16	
		S	2	23,1	17,3
		Спец. меркер	2	0,16	0,16
		T, C	2	27	25
		D, V, Z, Спец. регистр	3	21,9	17,1
PLS	Создание импульса по переднему фронту	Y, M	2	0,32	0,32
PLF	Создание импульса по заднему фронту	Y, M	2	0,32	
P	Адресация точки	0 ... 63	1	0,08	
I	Адресация точки прерывания	□□□	1	0,08	

Табл. В-16: Базовые команды и инструкции шаговых состояний для ПК серии FX2N и FX2NC (2)

В.2.2 ИНСТРУКЦИИ РАЗВЕТВЛЕНИЯ

Инструкция		Время выполнения [мкс]		
Символ	FNC-инструкция	Время включения	Время отключения	Специальная функция FNC
CJ	FNC 00 /S+	29	6,4	—
CALL	FNC 01 /S+	32,2	6,4	—
SRET	FNC 02 ②	21,2	21,2	—
IRET	FNC 03 ②	18,8	18,1	—
EI	FNC 04 ②	55,8	55,8	—
DI	FNC 05 ②	18,5	18,5	—
FEND	FNC 06 ②	508		—
WDT	FNC 07	26,3	6,4	—
FOR	FNC 08 /S+ ②	27,6	27,6	—
NEXT	FNC 09 ②	5,2	5,2	—

Табл. В-17: Команды разветвления программы FX2N и FX2NC

В.2.3 ИНСТРУКЦИИ СРАВНЕНИЯ И ПЕРЕНОСА

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
CMP	FNC 10 /S1+ /S2+ /D+	16	87,6	6,4	—
		32	91,9	6,4	—
ZCP	FNC 11 /S1+ /S2+ /S+ /D+	16	103,2	6,4	—
		32	108,9	6,4	—
MOV	FNC 12 /S+ /D+	16	1,52	1,52	—
		32	1,84	1,84	—
SMOV	FNC 13 /S+ /m1 /m2 /D+ /n	16	155,2	6,4	—
CML	FNC 14 /S+ /D+	16	51,4	6,4	—
		32	55,9	6,4	—
BMOV	FNC 15 /S+ /D+ /n ④	16	97 + 1,7n	6,4	—
FMOV	FNC 16 /S+ /D+ /n ④	16	69,1 + 2,8n	6,4	—
		32	73,2 + 5,2n	6,4	—
XCH	FNC 17 /D1+ /D2+ ③	16	57,2	6,4	—
		32	64	6,4	—
BCD	FNC 18 /S+ /D+	16	37,9	6,4	—
		32	57,6	6,4	—
BIN	FNC 19 /S+ /D+	16	32,4	6,4	—
		32	44,5	6,4	—

Табл. В-18: Инструкции сравнения и переноса FX2N и FX2NC

В.2.4 АРИФМЕТИЧЕСКИЕ ИНСТРУКЦИИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	27,6	6,4	—
		32	28,9	6,4	224
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	27,6	6,4	—
		32	28,9	6,4	232
MUL	FNC 22 /S1+ /S2+ /D+	16	25,2	6,4	—
		32	31,4	6,4	162
DIV	FNC 23 /S1+ /S2+ /D+	16	32	6,4	—
		32	36,4	6,4	197
INC	FNC 24 /D+ ③	16	18,8	6,4	—
		32	20,2	6,4	—
DEC	FNC 25 /D+ ③	16	18,9	6,4	—
		32	20	6,4	—
WAND	FNC 26 /S1+ /S2+ /D+	16	23,4	6,4	—
		32	24,8	6,4	—
WOR	FNC 27 /S1+ /S2+ /D+	16	23,5	6,4	—
		32	24,7	6,4	—
WXOR	FNC 28 /S1+ /S2+ /D+	16	23,5	6,4	—
		32	25,0	6,4	—
NEG	FNC 29 /D+ ③	16	35,3	6,4	—
		32	38,4	6,4	—

Табл. В-19: Арифметические инструкции FX2N и FX2NC

Br (Borrow, заем): M8021
 Cy (Carry, перенос): M8022
 F (Инструкция полностью обработана): M8029

В.2.5 ИНСТРУКЦИИ РОТАЦИИ И СДВИГА

Инструкция			Время выполнения [мкс]			
Символ	FNC-инструкция		Бит	Время включения	Время отключения	Специальная функция FNC
ROR	FNC 30 /D+ /n Cy	③ ⑤	16	61,7	6,4	—
			32	65,3	6,4	—
ROL	FNC 31 /D+ /n Cy	③ ⑤	16	61,2	6,4	—
			32	65,2	6,4	—
RCR	FNC 32 /D+ /n Cy	③ ⑤	16	66,3 + 2,2n	6,4	—
			32	69,7 + 2,6n	6,4	—
RCL	FNC 33 /D+ /n Cy	③ ⑤	16	65,8 + 2,2n	6,4	—
			32	69,5 + 2,6n	6,4	—
SFTR	FNC 34 /S+ /D+ /n1 /n2	③ ⑥	16	107 + 53,8n	6,4	—
SFTL	FNC 35 /S+ /D+ /n1 /n2	③ ⑥	16	105 + 53,8n	6,4	—
WSFR	FNC 36 /S+ /D+ /n1 /n2	③ ④	16	126 + 11,7n	6,4	—
WSFL	FNC 37 /S+ /D+ /n1 /n2	③ ④	16	125 + 11,8n	6,4	—
SFWR	FNC 38 /S+ /D+ /n	③ ⑦	16	83,9	6,4	—
SFRD	FNC 39 /S+ /D+ /n	③ ⑦	16	80,2	6,4	—

Табл. В-20: Инструкции ротации и сдвига FX2N и FX2NC

В.2.6 ОПЕРАЦИИ С ДАННЫМИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	77 + 1,7n	6,4	—
		16 (S)	83 + 11,1n		
		16 (C)			
		16 (T)			
		16 (M)	89,2 + 9,4n		
16 (Y)					
DECO	FNC 41 /S+ /D+ /n	16	76	6,4	—
ENCO	FNC 42 /S+ /D+ /n	16	81,8	6,4	—
SUM	FNC 43 /S+ /D+	16	72,8	6,4	—
		32	94,6	6,4	—
BON	FNC 44 /S+ /D+ /n	16	78,2	6,4	—
		32	82,3	6,4	—
MEAN	FNC 45 /S+ /D+ /n ⑨	16	83,8 + 3,4n	6,4	—
		32	90,9 + 6,7n	6,4	—
ANS	FNC 46 /S+ /m /D+	16	100,8	6,4	—
ANR	FNC 47 ③	16	37,7	6,4	—
SQR	FNC 48	16	150,2	6,4	—
		32	154,8	6,4	344
FLT	FNC 49	16	66,8	6,4	—
		32	66,8	6,4	—

Табл. В-21: Операции с данными FX2N и FX2NC

В.2.7 ВЫСОКОСКОРОСТНЫЕ ИНСТРУКЦИИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
REF	FNC 50 /D /n ⑩	16	99,6 + 0,6n	6,4	—
REFF	FNC 51 /n ⑪	16	65,3 + 1,7n	6,4	—
MTR	FNC 52 /S /D1 /D2 /n	16	39,1	23,6	—
HSCS	FNC 53 /S1+ /S2+ /D+ ⑫	32	87,8	6,4	—
HSCR	FNC 54 /S1+ /S2+ /D+ ⑫	32	88,6	6,4	—
HSZ	FNC 55 /S1+ /S2+ /S+ /D+ ⑫	32	100,6	6,4	—
SPD	FNC 56 /S1+ /S2+ /D+	②	80,2	80,2	—
PLSY	FNC 57 /S1+ /S2+ /D+	16	85	73,3	—
		32	86,6	75,8	—
PWM	FNC 58 /S1+ /S2+ /D+	16	70,4	73,3	—
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	122,6	87,5	—
		32	125,6	90,5	

Табл. В-22:Высокоскоростные инструкции FX2N и FX2NC

В.2.8 ИНСТРУКЦИЙ ОРИЕНТИРОВАННЫЕ НА ПОЛЬЗОВАТЕЛЯ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
IST	FNC 60 /S /D1 /D2	16	114,3	6,4	—
SER	FNC 61 ⑬	16	129,2 + 8,6n	22,9	—
		32	147 + 9n	29	
ABSD	FNC 62 /S1+ /S2+ /D+ / n ⑭	16	91,8 + 20,2n	6,4	—
		32	97,5 + 21,5n	6,4	
INCD	FNC 63 /S1+ /S2+ /D+ / n	16	110,5	19,5	—
TTMR	FNC 64 /D+ /n	16	54,9	44,9	—
STMR	FNC 65 /S+ /m /D+	16	84,4	84,4	—
ALT	FNC 66 /D+	16	50,1	6,4	—
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	98,1	81,6	—
ROTC	FNC 68 /S+ /m1 /m2 /D+	16	118,4	107,2	—
SORT	FNC 69 ⑮	16	50,5	19,5	—

Табл. В-23:Инструкции ориентированные на пользователя FX2N и FX2NC

В.2.9 СПЕЦИАЛЬНЫЕ FNC-ИНСТРУКЦИИ

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
TKY	FNC 70 /S+ /D1+ /D2+	16	97,2	22,2	—
		32	98,7	22,2	—
HKY	FNC 71 /S+ /D1+ /D2+ /D3+	16	92,2	27,4	—
		32	65,0	6,4	—
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	92,2	27,4	—
SEGD	FNC 73 /S1+ /D+	16	65	6,4	—
SEGL	FNC 74 /S+ /D+ /n	16 1Set	105,9	26,5	—
ARWS	FNC 75 /S1+ /D+ /D2+ /n	16	134,4	22,1	—
ASC	FNC 76 /S+ /D+	16	49,5	6,4	—
PR	FNC 77 /S+ /D+	16	в процессе нажима: 114,8 при окончании нажима: 88	88,5	—
FROM	FNC 78 /n1 /n2 /D+ /n3 16	16	97 + 487n	6,4	—
		32	99 + 962n	6,4	
TO	FNC 79 /n1 /n2 /S+ /n3 16	16	94 + 557n	6,4	—
		32	96 + 1099n	6,4	
RS	FNC 80	16	117,6	18	—
		32			
PRUN	FNC 81 /S+ /D+ 17	16	65,6 + 17n	6,4	—
		32	67 + 17,7n	6,4	—
ASCI	FNC 82	16	88,2 + 10,8n	6,4	—
HEX	FNC 83	16	89,7 + 20n	6,4	—
CCD	FNC 84	16	90,5 + 4,8n	6,4	—
VRRD	FNC 85 /S+ /D+	16	209,7	27,3	—
VRSC	FNC 86 /S+ /D+	16	202,4	27,3	—
PID	FNC 88	16	155	89	—

Табл. В-24: Специальные FNC-инструкции FX2N и FX2NC (1)

Инструкция		Время выполнения [мкс]			
Символ	FNC-инструкция	Бит	Время включения	Время отключения	Специальная функция FNC
ECMP	FNC 110 /S1+ /S2+ /D+	32	104,4	6,4	—
EZCP	FNC 111 /S1+ /S2+ /S3+ /D+	32	124,5	6,4	—
EBCD	FNC 118 /S+ /D+	32	106,9	6,4	—
EBIN	FNC 119 /S+ /D+	32	81,3	6,4	—
EADD	FNC 120 /S1+ /S2+ /D+	32	117,4	6,4	—
ESUB	FNC 121 /S1+ /S2+ /D+	32	117,4	6,4	—
EMUL	FNC 122 /S1+ /S2+ /D+	32	96,4	6,4	—
EDIV	FNC 123 /S1+ /S2+ /D+	32	100,4	6,4	—
ESQR	FNC 127 /S+ /D+	32	152,1	6,4	—
INT	FNC 129 /S+ /D+	16	67,5	6,4	—
		32	70,4	6,4	
SIN	FNC 130 /S+ /D+	32	199,5	6,4	—
COS	FNC 131 /S+ /D+	32	262,5	6,4	—
TAN	FNC 132 /S+ /D+	32	425,3	6,4	—
SWAP	FNC 147 /S+	16	36,1	6,4	—
		32	41,2	6,4	
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	134,2	6,4	—
TZCP	FNC 161 /S1+ /S2+ /S+ /D+	16	140,2	6,4	—
TADD	FNC 162 /S1+ /S2+ /D+	16	118,8	6,4	—
TSUB	FNC 163 /S1+ /S2+ /D+	16	109,4	6,4	—
TRD	FNC 166 /D+	16	46,2	6,4	—
TWR	FNC 167 /S+	16	112	6,4	—
GRY	FNC 170 /S+	16	102,5	6,4	—
		32	107,1	6,4	
GBIN	FNC 171 /D+	16	103,4	6,4	—
		32	107,5	6,4	
RD3A	FNC 176 /M1+ /M2+ /D+	16	1248,3	7,5	
WR3A	FNC 177 /M1+ /M2+ /D+	16	1263,7	7,5	
LD□	FNC 221 – 230 /S1+ /S2+	16	1,52		—
		32	1,84		
AND□	FNC 232 – 238 /S1+ /S2+	16	1,52		—
		32	1,84		
OR□	FNC 240 – 246 /S1+ /S2+	16	1,52		—
		32	1,84		

Табл. В-25: Специальные FNC-инструкции FX2N и FX2NC (2)

- ① “n” указывает число статических STL-инструкций (число параллельных/сходящихся инструкций).
- ② Характеристика инструкций, не требующих никаких контактов.
- ③ Если применяется статическая команда, а не импульсная, значение целевого адреса изменяется циклически.
- ④ Если включены статические инструкции и нет импульсной инструкции, изменяется значение адреса цели.
- ⑤ “n” указывает число сдвигаемых регистров ($n \leq 512$).
- ⑥ “n” указывает число обрабатываемых бит-операндов ($n \leq 16$ в режиме 16 бит, $n \leq 32$ в режиме 32 бит).
- ⑦ “n” указывает число обрабатываемых бит-операндов.
- ⑧ “n” указывает число обрабатываемых операндов ($2 \leq n \leq 512$).
- ⑨ “n” указывает отключаемую область операндов. Тип операнда указывается в скобках.
- ⑩ “n” указывает адреса операндов, которые обрабатываются MEAN-инструкцией ($1 \leq n \leq 64$).
- ⑪ “n” указывает активизированную область операндов ($8 \leq n \leq 128$ в восьми шагах).
- ⑫ “n” указывает константу времени входного фильтра ($0 \leq n \leq 60$ мс).
- ⑬ Эта инструкция может быть активной максимум 6 раз в течении одного и того же времени.
- ⑭ “n” указывает число штабелируемых элементов ($n \leq 256$ для 16-ти битной обработки, $n \leq 128$ для 32-х битной обработки)
- ⑮ “n” указывает число адресов выходов ($n \leq 64$).
- ⑯ “n” указывает число элементов таблицы данных ($1 \leq m \leq 32$). Для полной SORT-обработки SORT-инструкция выполняется m1 раз).
- ⑰ “n” указывает число записываемых или читаемых слов данных в или из специального модуля.
- ⑱ “n” указывает число записываемых или читаемых байтовых сигнальных слов данных (8 бит) при параллельной работе двух ПК серии FX.

В.3 ВРЕМЯ ВЫПОЛНЕНИЯ В СЕРИИ FX3U

Сноски находятся в конце раздела.

В.3.1 БАЗОВЫЕ КОМАНДЫ И КОМАНДЫ СОСТОЯНИЯ ШАГА

Инструкция	Значение	Операнды	Количество шагов программы	Время выполнения [мкс]	
				Время включения	Время отключения
LD	Начало логической связи, опрос на сигнал "1"	X, Y, M, S, T, C, специальные маркеры	1	0,065 (0,129 мкс при обращении к операндам M1536...M3528, M8256...M8511 и S1024...S4095) (0,193 мкс при обращении к маркерам M3584...M7679)	
LDI	Начало логической связи, опрос на сигнал „0“				
AND	Связь И, опрос на сигнал „1“				
ANI	Связь И, опрос на сигнал „0“				
OR	Связь ИЛИ, опрос на сигнал "1"				
ORI	Связь ИЛИ, опрос на сигнал "0"				
LDP	Загрузка (импульсная операция); начало логической связи с опросом положительного фронта	X, Y, M, S, T, C	1	7,8	
LDF	Загрузка (импульсная операция); начало логической связи с опросом отрицательного фронта				
ANDP	Связь И (импульсная); связь И с опросом положительного фронта				
ANDF	Связь И (импульсная); связь И с опросом отрицательного фронта				
ORP	Связь ИЛИ (импульсная); связь ИЛИ с опросом положительного фронта				
ORF	Связь ИЛИ (импульсная); связь ИЛИ с опросом отрицательного фронта				
ANB	Блок И, последовательное соединение параллельных соединений	-	1	0,065	
ORB	Блок ИЛИ, параллельное соединение последовательных соединений				
MPS	Сохранение результата логической операции				
MRD	Считывание результата логической операции				
MPP	Считывание и стирание памяти логической операции				
INV	Инверсия, изменение результатов обработки на противоположные	-	1	0,065	
MC	Установка контрольного условия	N - Y, M	3	4,3	4,7
MCR	Сброс контрольного условия	N (вложения)	2	3,9	

Табл. В-26:Время выполнения базовых команд и команд состояния шага в контроллерах серии FX3U (1)

Инструкция	Значение	Операнды	Количество шагов программы	Время выполнения [мкс]	
				Время включения	Время отключения
NOP	Пустая строка		1	0,065	
END	Конец программы	–	1	$113,9 + (2,13 \times X) + (3,25 \times Y)$ X: количество входов Y: количество выходов Если команды FEND и END используются вместе, значение имеет только время выполнения команды END.	
STL	Выполнить команду состояния шага	S	1	$5,1 + 1,6n$ ①	
RET	Закончить команду состояния шага	–	1	2,9	
OUT	Вывод, присвоение результата логической операции	Y	1	0,065	
		M		0,065 ((0,129 мкс для операндов M1536...M3528 и M8256...M8511) (0,193 мкс для маркеров M3584...M7679)	
		S	2	4,8	4,8
		T-K	3	0,71	0,71
		T-K (T192 – T199, T246 – T511)		11,6	8,2
		T-D	3	0,71	0,71
		T-K (T192 – T199, T246 – T511)		11,6	8,2
		C-K (16бита)	3	0,71	0,71
		C-K (32бита)	5	6,1	6,1
		C-K (C235 – C255)		9,5	9,0
		C-D (16бита)	3	0,71	0,71
		C-D (32бита)	5	6,1	6,1
		C-D (C235 – C255)		9,5	9,0
		SET	Установка, установка операндов	Y,	1
M	1			(0,129 мкс для операндов M1536...M3528 и M8256...M8511) (0,193 мкс для маркеров M3584...M7679) 0,065	
S	2			4,7	0,13
S при применении в ступени STL ①				$6,6 + 0,9n$	

Табл. В-27:Время выполнения базовых команд и команд состояния шага в контроллерах серии FX3U(2)

Инструкция	Значение	Операнды	Количество шагов программы	Время выполнения [мкс]	
				Время включения	Время отключения
RST	Сброс, сброс операндов	Y	1	0,065	
		M	1	0,065 (0,129 мкс для операндов M1536 ... M3528 und M8256 ... M8511) (0,193 мкс для маркеров M3584 ... M7679)	
		S	2	4,6	0,13
		T	2	0,45	0,45
		C (16бита)	2	0,45	0,45
		C (32бита)	2	5,8	4,8
		D, V, Z, Спец. регистр	3	5,4	0,195
PLS	Выработка импульса при положительном фронте	Y, M	2	0,257 (0,321 μs для M3584 ... M7679)	
PLF	Выработка импульса при отрицательном фронте	Y, M	2	0,257 (0,321 μs для M3584 ... M7679)	

Табл. В-29: А время выполнения базовых команд и команд состояния шага в контроллерах серии FX3U (3)

В.3.2

2 КОМАНДЫ РАЗВЕТВЛЕНИЯ ПРОГРАММЫ

Команда			Время выполнения [мкс]	
Символ	Команда FNC		Время включения	Время выключения
CJ	FNC 00 /S+		8	0,195
CALL	FNC 01 /S+		13,5	0,195
SRET	FNC 02	②		–
IRET	FNC 03	②	4,4	–
EI	FNC 04	②	3,8	–
DI	FNC 05	②	3,7	–
FEND	FNC 06	②	113,9 + (2,13 x X) + (3,25 x Y) X: количество входов Y: количество выходов Если команды FEND и END используются вместе, значение имеет только время выполнения команды END.	
WDT	FNC 07		5,4	0,065
FOR	FNC 08 /S+	②	11,6	–
NEXT	FNC 09	②		–

Табл. В-28: Время выполнения команд разветвления программы в контроллерах FX3U

В.3.3 КОМАНДЫ СРАВНЕНИЯ И ПЕРЕДАЧИ

Команда		Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения
CMP	FNC 10 /S1+ /S2+ /D+	16	15,5	0,455
		32	16,0	0,845
ZCP	FNC 11 /S1+ /S2+ /S+ /D+	16	18,9	0,585
		32	19,7	1,105
MOV	FNC 12 /S+ /D+	16	0,64	0,32
		32	1,48	1,48
SMOV	FNC 13 /S+ /m1 /m2 /D+ /n	16	22,9	0,715
CML	FNC 14 /S+ /D+	16	10,6	0,325
		32	10,2	0,585
BMOV	FNC 15 /S+ /D+ /n ④	16	13,9 + 0,44n	0,455
FMOV	FNC 16 /S+ /D+ /n ④	16	14,2 + 0,19n	0,455
		32	14,0 + 0,38n	0,845
XCH	FNC 17 /D1+ /D2+ ③	16	10,7	0,325
		32	11,4	0,585
BCD	FNC 18 /S+ /D+	16	11,0	0,325
		32	15,5	0,585
BIN	FNC 19 /S+ /D+	16	7,5	0,325
		32	8,4	0,585

Табл. В-30:Время выполнения команд сравнения и передачи в контроллерах FX3U

В.3.4 АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

Команда		Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения
ADD	FNC 20 /S1+ /S2+ /D+ Z, Cy, Br	16	8,1	0,455
		32	8,5	0,845
SUB	FNC 21 /S1+ /S2+ /D+ Z, Cy, Br	16	8,1	0,455
		32	8,5	0,845
MUL	FNC 22 /S1+ /S2+ /D+	16	7,0	0,455
		32	7,3	0,845
DIV	FNC 23 /S1+ /S2+ /D+	16	8,2	0,455
		32	8,6	0,845
INC	FNC 24 /D+ ③	16	6,2	0,195
		32	6,4	0,325
DEC	FNC 25 /D+ ③	16	6,2	0,195
		32	6,4	0,325
WAND	FNC 26 /S1+ /S2+ /D+	16	7,0	0,455
		32	7,3	0,845
WOR	FNC 27 /S1+ /S2+ /D+	16	7,0	0,455
		32	7,3	0,845
WXOR	FNC 28 /S1+ /S2+ /D+	16	7,0	0,455
		32	7,3	0,845
NEG	FNC 29 /D+ ③	16	7,6	0,195
		32	8,0	0,325

Табл. В-31:: Время выполнения арифметических операций в контроллерах FX3U

Br (Borrow, заем):

M8021

Cy (Carry, перенос):

M8022

F (команда отработана полностью):

M8029

В.3.5 КОМАНДЫ ВРАЩЕНИЯ И СДВИГА

Команда			Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения	
ROR	FNC 30 /D+ /n Cy	③ ⑤	16	10,5	0,325
			32	11,5	0,585
ROL	FNC 31 /D+ /n Cy	③ ⑤	16	10,5	0,325
			32	11,5	0,585
RCR	FNC 32 /D+ /n Cy	③ ⑤	16	10,9	0,325
			32	11,8	0,585
RCL	FNC 33 /D+ /n Cy	③ ⑤	16	10,9	0,325
			32	11,8	0,585
SFTR	FNC 34 /S+ /D+ /n1 /n2	③ ⑥	16	23,2 + 0,08n	0,585
SFTL	FNC 35 /S+ /D+ /n1 /n2	③ ⑥	16	23,2 + 0,08n	0,585
WSFR	FNC 36 /S+ /D+ /n1 /n2	③ ④	16	7,5 + 0,44n	0,585
WSFL	FNC 37 /S+ /D+ /n1 /n2	③ ④	16	7,5 + 0,44n	0,585
SFWR	FNC 38 /S+ /D+ /n	③ ⑦	16	8,1	0,455
SFRD	FNC 39 /S+ /D+ /n	③ ⑦	16	7,7	0,455

Табл. В-33:Время выполнения команд вращения и сдвига в контроллерах FX3U

В.3.6 ОПЕРАЦИИ С ДАННЫМИ

Команда			Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения	
ZRST	FNC 40 /D1+ /D2+ ⑧	16 (D)	11,1 + 0,19n	0,325	
		16 (T)	17,1 + 0,23n		
		16 (M)	20,7 + 0,02n		
DECO	FNC 41 /S+ /D+ /n	16	13,5	0,455	
ENCO	FNC 42 /S+ /D+ /n	16	18,0	0,455	
SUM	FNC 43 /S+ /D+	16	12,7	0,325	
		32	16,9	0,585	
BON	FNC 44 /S+ /D+ /n	16	14,4	0,455	
		32	15,1	0,845	
MEAN	FNC 45 /S+ /D+ /n	⑨	16	11,8 + 0,41n	0,455
			32	17,8 + 2,13n	0,845
ANS	FNC 46 /S+ /m /D+	16	20,4	19,7	
ANR	FNC 47	⑩	16	7,0	0,065
SQR	FNC 48	16	9,7	0,325	
		32	12,1	0,585	
FLT	FNC 49	16	9,8	0,325	
		32	9,5	0,585	

Табл. В-32:Время выполнения операций с данными в контроллерах FX3U

В.3.7 ВЫСОКОСКОРОСТНЫЕ КОМАНДЫ

Команда		Время выполнения [мкс]			
Символ	Команда FNC	бит	Время включения	Время выключения	
REF	FNC 50 /D /n ⑩	16	4,5 + 1,39n	0,325	
REFF	FNC 51 /n ⑪	16	14,4 + 0,24n	0,195	
MTR	FNC 52 /S /D1 /D2 /n	16	5,9	5,5	
HSCS	FNC 53 /S1+ /S2+ /D+ ⑫	32	20,0	0,845	
HSCR	FNC 54 /S1+ /S2+ /D+ ⑫	32	88,6	6,4	
HSZ	FNC 55 /S1+ /S2+ /S+ /D+ ⑫	32	22,0	1,105	
SPD	FNC 56 /S1+ /S2+ /D+	②	16	12,6	
PLSY	FNC 57 /S1+ /S2+ /D+	16	20,0	6,9	
		32	13,6		
PWM	FNC 58 /S1+ /S2+ /D+	16	10,6	6,2	
PLSR	FNC 59 /S1+ /S2+ /S3+ /D+	16	11,2	7,0	
		32			

Табл. В-35:Время выполнения высокоскоростных команд в контроллерах FX3U

В.3.8 ПРИКЛАДНЫЕ КОМАНДЫ

Команда		Время выполнения [мкс]			
Символ	Команда FNC	бит	Время включения	Время выключения	
IST	FNC 60 /S /D1 /D2	16	28,5	0,455	
SER	FNC 61 ⑬	16	16,4 + 1,4n	0,585	
		32	18,5 + 2,13n	1,105	
ABSD	FNC 62 /S1+ /S2+ /D+ / n ⑭	16	19 + 0,85n	0,585	
		32	20,0 + 1,23n	1,105	
INCD	FNC 63 /S1+ /S2+ /D+ / n	16	23,7	6,5	
TTMR	FNC 64 /D+ /n	16	10,4	9,2	
STMR	FNC 65 /S+ /m /D+	16	19,0	21,0	
ALT	FNC 66 /D+	16	11,6	0,2	
RAMP	FNC 67 /S1+ /S2+ /D+ /n	16	15,0	7,5	
ROTC	FNC 68 /S+ /m1 /m2 /D+	16	25,8	24,8	
SORT	FNC 69 ⑮	16	18,4	6,6	

Табл. В-34:Время выполнения прикладных команд в контроллерах FX3U

В.3.9 СПЕЦИАЛЬНЫЕ КОМАНДЫ

Команда		Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения
TKY	FNC 70 /S+ /D1+ /D2+	16	21,5	5,2
		32	21,8	
HKY	FNC 71 /S+ /D1+ /D2+ /D3+	16	32,0	5,7
		32	32,3	
DSW	FNC 72 /S+ /D1+ /D2+ /n	16	26,8	22,1
SEGD	FNC 73 /S1+ /D+	16	10,8	0,325
SEGL	FNC 74 /S+ /D+ /n	16 1Set	22,3	7,5
ARWS	FNC 75 /S1+ /D+ /D2+ /n	16	28,8	5,2
ASC	FNC 76 /S+ /D+	16	19,8	0,715
PR	FNC 77 /S+ /D+	16	24,0	13,6
FROM ^①		16	24,8 + 529,5n	0,585
		32	15,9 + 1056n	1,105
FROM ^②	FNC 78 /n1 /n2 /D+ /n3 ^⑬	16	17,3 + 1078n	0,585
		32	20,30 + 2152n	1,105
FROM ^③		16	27,9 + 108n	0,585
		32	17,6 + 187,4n	1,105
TO ^①		16	14,4 + 552n	0,585
		32	19,5 + 1101n	1,105
TO ^②	FNC 79 /n1 /n2 /S+ /n3 ^⑬	16	16,9 + 1110n	0,585
		32	18,4 + 2197n	1,105
TO ^③		16	96,7 + 119,2n	0,585
		32	17,3 + 297,7	1,105
RS	FNC 80	16	15,6	5,7
		32		
PRUN	FNC 81 /S+ /D+ ^⑰	16	17,1 + 1,67n	0,325
		32	18,2 + 2,9n	0,585
ASCI	FNC 82	16	13,5 + 1,45n	0,455
HEX	FNC 83	16	13,6 + 1,89n	0,455
CCD	FNC 84	16	13,6 + 1,63n	0,455
VRRD	FNC 85 /S+ /D+	16	Эти команды не может выполнять базовый блок серии FX3U.	
VRSC	FNC 86 /S+ /D+	16		
PID	FNC 88	16	20,0	8,9

Табл. В-36:Время выполнения специальных команд в контроллерах FX3U (1)

- ① Действительны эти значения времени, если эта команда обращается к адресам буферной памяти с 0 по 31 специального модуля серий FX0N, FX2N или FX2NC.
- ② Действительны эти значения времени, если эта команда обращается к буферной памяти специального модуля серий FX0N, FX2N или FX2NC, начиная с адреса 32.
- ③ Действительны эти значения времени, если с помощью этой команды осуществляется доступ к буферной памяти специального модуля серии FX3U.

Команда		Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения
ZPUSH	FNC 102 /D+	16	16,0	0,195
ZPOP	FNC 103 /D+	16	16,0	0,195
ECMP	FNC 110 /S1+ /S2+ /D+	32	18,2	0,845
EZCP	FNC 111 /S1+ /S2+ /S3+ /D+	32	21,6	1,105
EMOV	FNC 112 /S+ /D+	32	10,0	0,585
ESTR	FNC 116 /S1+ /S2+ /D+	32	27 + 1,7n + 1,2m п: число знаков m: количество разрядов дробной части	0,845
EVAL	FNC 117 /S+ /D+		26 + 3,8n	0,585
EBCD	FNC 118 /S+ /D+	32	10,0	0,585
EBIN	FNC 119 /S+ /D+	32	11,9	0,585
EADD	FNC 120 /S1+ /S2+ /D+	32	14,2	0,845
ESUB	FNC 121 /S1+ /S2+ /D+	32	14,2	0,845
EMUL	FNC 122 /S1+ /S2+ /D+	32	14,1	0,845
EDIV	FNC 123 /S1+ /S2+ /D+	32	17,7	0,845
EXP	FNC 124 /S+ /D+	32	11,9	0,585
LOGE	FNC 125 /S+ /D+	32	24,0	0,585
LOG10	FNC 126 /S+ /D+	32	24,3	0,585
ESQR	FNC 127 /S+ /D+	32	10,6	0,585
INT	FNC 129 /S+ /D+	16	13,2	0,325
		32	13,0	0,585
SIN	FNC 130 /S+ /D+	32	12,0	0,585
COS	FNC 131 /S+ /D+	32	23,2	0,585
TAN	FNC 132 /S+ /D+	32	12,0	0,585
ASIN	FNC 133 /S+ /D+	32	13,5	0,585
ACOS	FNC 134 /S+ /D+	32	13,5	0,585
ATAN	FNC 135 /S+ /D+	32	12,0	0,585
RAD	FNC 136 /S+ /D+	32	14,9	0,585
DEG	FNC 137 /S+ /D+	32	14,9	0,585
WSUM	FNC 140 /S+ /D+ /n	16	11,7 + 0,38n	0,455
		32	14,1 + 1,94n	0,845
WTOB	FNC 141 /S+ /D+ /n	16	12,6 + 1,43n	0,455
BTOW	FNC 142 /S+ /D+ /n	16	12,6 + 0,92n	0,455
UNI	FNC 142 /S+ /D+ /n	16	11,6 + 0,4n	0,455
DIS	FNC 143 /S+ /D+ /n	16	10,6 + 0,2n	0,455
SWAP	FNC 147 /S+	16	7,7	0,195
		32	8,0	0,325
SORT2	FNC 149 /S+ /n1 /n2 /D+ /m	16	13,2	6,5
		32	15,2	7,7
DSZR	FNC 150 /S1+ /S2+ /D1+ /D2+	16	170,0	7,0
DVIT	FNC 151 /S1+ /S2+ /D1+ /D2+	16	178,0	7,1
		32		

Табл. В-37:Время выполнения специальных команд в контроллерах FX3U (2)

Команда		Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения
TBL	FNC 152 /D+ /n	–	Время выполнения зависит от того, с какой командой позиционирования сочетается команда TBL: команда DDVIT: 178,0 команда DPLSV: 144,0 команда DDRVI: 178,0 команда DRRVA: 178,0	
ABS	FNC 155 /S+ /D1+ /D2+	32	25,4	22,2
ZRN	FNC 156 /S1+ /S2+ /S3+ /D+	16	58,0	7,1
		32	62,0	
PLSV	FNC 157 /S+ /D1+ /D2+	16	144,0	7,1
		32		
DRVI	FNC 158 /S1+ /S2+ /D1+ /D2+	16	178,0	7,1
		32		
DRVA	FNC 159 /S1+ /S2+ /D1+ /D2+	16	178,0	7,1
		32		
TCMP	FNC 160 /S1+ /S2+ /S3+ /S+ /D+	16	21,3	0,715
TZCP	FNC 161 /S1+ /S2+ /S3+ /D+	16	22,6	0,585
TADD	FNC 162 /S1+ /S2+ /D+	16	13,4	0,455
TSUB	FNC 163 /S1+ /S2+ /D+	16	13,4	0,455
HTOS	FNC 164 /S+ /D+	16	10,8	0,325
		32	11,0	0,585
STOH	FNC 165 /S+ /D+	16	11,4	0,325
		32	11,6	0,585
TRD	FNC 166 /D+	16	10,0	0,195
TWR	FNC 167 /S+	16	344,4	0,195
HOUR	FNC 169 /S+ /D1+ /D2+	16	15,5	15,2
		32	16,1	15,9
GRY	FNC 170 /S+	16	10,2	0,325
		32	10,7	0,585
GBIN	FNC 171 /D+	16	15,4	0,325
		32	16,0	0,585
RD3A	FNC 176 /M1+ /M2+ /D+	16	693,1(FX0N-3A) 906,0 (FX2N-2AD)	0,455
WR3A	FNC 177 /M1+ /M2+ /D+	16	713,9 (FX0N-3A) 1415,9 (FX2N-2AD)	0,455
COMRD	FNC 182 /S+ /D+	16	33,7	0,325
RND	FNC 183 /D+	16	8,5	0,195
DUTY	FNC 186 /n1 /n2 /D+	16	6,0	6,0
CRC	FNC 188 /S+ /D+ /n	16	12,6 + 0,82n	0,455
DHCMOV	FNC 189 /S+ /D+ /n	32	14,8	0,845
BK+	FNC 192 /S1+ /S2+ /D+ /n	16	13,1 + 0,66n	0,585
		32	13,9 + 1,23n	1,105
BK-	FNC 193 /S1+ /S2+ /D+ /n	16	13,1 + 0,66n	0,585
		32	13,9 + 1,23n	1,105
BKCMPO	FNC 194 – 199 /S1+ /S2+ /D+ /n	16	19,6 + 1,88n	0,585
		32	20,3 + 2,26n	1,105
STR	FNC 200 /S1+ /S2+ /D+	16	34,6	0,455
		32	47,0	0,845
VAL	FNC 201 /S+ /D1+ /D2+	16	20,7	0,455
		32	29,2	0,845

Табл. В-38:Время выполнения специальных команд в контроллерах FX3U (3)

Команда		Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения
\$+	FNC 202 /S1+ /S2+ /D+	16	24,8 + 1,5m m: число знаков	0,455
LEN	FNC 203 /S+ /D+	16	12 + 0,44m m: число знаков	0,325
RIGHT	FNC 204 /S+ /D+ /n	16	18,1 + 1,06n + 0,47m n: число знаков m: число извлеченных знаков	0,455
LEFT	FNC 205 /S+ /D+ /n	16	18,1 + 1,06n + 0,47m n: число знаков m: число извлеченных знаков	0,455
MIDR	FNC 206 /S1+ /D+ /S2+	16	25 + 0,59n + 0,68m n: позиция знака m: число знаков	0,455
MIDW	FNC 207 /S1+ /D+ /S2+	16	25,8 + 0,3m + 0,44n n: позиция знака m: число сохраненных знаков	0,455
INSTR	FNC 208 /S1+ /S2+ /D+ /n	16	6,64 + 2,36n m: число искомых знаков	0,585
\$MOV	FNC 209 /S+ /D+	16	16 + 1,52n	0,325
FDEL	FNC 210 /S+ /D+ /n	16	43 + 0,95m ¹⁸	–
FINS	FNC 211 /S+ /D+ /n	16	63 + 0,98m ¹⁹	–
POP	FNC 212 /S+ /D+ /n	16	7,8	0,455
SFR	FNC 213 /D+ /n	16	9,3	0,325
SFL	FNC 214 /D+ /n	16	9,3	0,325
LD □	FNC 224 – 230 /S1+ /S2+	16	1,22	
		32	1,48	
AND □	FNC 232 – 238 /S1+ /S2+	16	1,22	
		32	1,48	
OR □	FNC 240 – 246 /S1+ /S2+	16	1,22	
		32	1,48	
LIMIT	FNC 256 /S1+ /S2+ /S3+ /D+	16	8,1	0,585
		32	8,6	1,105
BAND	FNC 257 /S1+ /S2+ /S3+ /D+	16	8,1	0,585
		32	8,6	1,105
ZONE	FNC 258 /S1+ /S2+ /S3+ /D+	16	7,9	0,585
		32	8,5	1,105
SCL	FNC 259 /S1+ /S2+ /D+	16	15,9	0,455
		32	16,8	0,845
DABIN	FNC 260 /S+ /D+	16	13,7	0,325
		32	19,5	0,585
BINDA	FNC 261 /S+ /D+	16	16,7	0,325
		32	23,1	0,585
SCL2	FNC 269 /S1+ /S2+ /D+	16	2,79 + 5,21n n: количество координатных точек	0,455
		32	29,06 + 7,94n n: количество координатных точек	0,845
IVCK	FNC 270 /S1+ /S2+ /D+ /n	16	14,1	6,5

Табл. В-39:Время выполнения специальных команд в контроллерах FX3U (4)

Команда		Время выполнения [мкс]		
Символ	Команда FNC	бит	Время включения	Время выключения
IVDR	FNC 271 /S1+ /S2+ /S3+ /n	16	14,1	6,5
IVRD	FNC 272 /S1+ /S2+ /D+ /n	16	16,2	6,5
IVWR	FNC 273 /S1+ /S2+ /S3+ /n	16	16,2	6,5
IVBWR	FNC 274 /S1+ /S2+ /S3+ /n	16	20,0	6,5
RBFM ①	FNC 278 /n1 /n2 /D3+ /n3 /n4	16	1086n n: количество данных, переданных в одном программном цикле	0,715
RBFM ②		16	244 + 103n n: количество данных, переданных в одном программном цикле	
WBFM ①	FNC 279 /n1 /n2 /D3+ /n3 /n4	16	1099n n: количество данных, переданных в одном программном цикле	0,715
WBFM ②		16	292 + 116n n: количество данных, переданных в одном программном цикле	
DHSCT	FNC 280 /S1+ /n1 /S2+ /D+ /n2	32	30,0	1,365
LOADR	FNC 290 /S+ /n	16	13,2 + 0,44n	0,325
SAVER	FNC 291 /S+ /n /D+	16	4300 + 215 x (2048/n)	6,4
INTR	FNC 292 /S+ /n	16	17600n	0,325
LOGR	FNC 293 /S+ /n1 /D1+ /n2 /D2+	16	244 + 17,9(n2)	0,715
RWER	FNC 294 /S+ /n	16	46700n n: количество секторов	0,325
INITER	FNC 295 /S+ /n	16	24 + 950n	0,325

Табл. В-40:Время выполнения специальных команд в контроллерах FX3U (5)

- ① Действительны эти значения времени, если эта команда обращается к буферной памяти специального модуля серии FX0N, FX2N или FX2NC.
- ② Действительны эти значения времени, если с помощью этой команды осуществляется доступ к буферной памяти специального модуля серии FX3U.

- ① "n" означает количество статических команд STL (количество параллельных / объединяющих команд).
- ② Обозначенные команды не нуждаются в контактах.
- ③ Если применяется статическая команда, а не импульсная, значение целевого адреса изменяется циклически.
- ④ "n" указывает количество перемещаемых регистров ($n \leq 512$).
- ⑤ "n" указывает количество обрабатываемых битовых операндов ($n \leq 16$ в 16-битном режиме, $n \leq 32$ в 32-битном режиме).
- ⑥ "n" указывает количество обрабатываемых битовых операндов.
- ⑦ "n" указывает количество обрабатываемых операндов ($2 \leq n \leq 512$).
- ⑧ "n" указывает сбрасываемую область операндов. Тип операнда указан в скобках.
- ⑨ "n" указывает адреса операндов, обрабатываемых командой MEAN ($1 \leq n \leq 64$).
- ⑩ "n" указывает обновляемую область операндов ($8 \leq n \leq 128$ с шагом 8).
- ⑪ "n" указывает постоянную времени входного фильтра ($0 \leq n \leq 60$ мс).
- ⑫ Эта команда может быть одновременно активирована максимум 6 раз.
- ⑬ "n" указывает количество элементов стека ($n \leq 256$ для 16-битной обработки, $n \leq 128$ для 32-битной обработки).
- ⑭ "n" указывает количество выходных адресов ($n \leq 64$).
- ⑮ "n" указывает количество элементов таблицы данных ($1 \leq m1 \leq 32$). Для полной обработки SORT команда SORT выполняется $m1$ раз.
- ⑯ "n" указывает количество слов данных, записываемых в специальный модуль или считываемых из специального модуля.
- ⑰ "n" указывает количество записываемых или считываемых однобайтовых слов данных (8 битов) при параллельной работе двух контроллеров FX.
- ⑱ $m =$ (количество строк таблицы) - (позиция стертых данных)
Количество строк таблицы соответствует содержимому (D+), а позиция стертых данных указывается в этой команде с помощью (n).
- ⑲ $m =$ (количество строк таблицы) - (позиция вставленных данных)
Количество строк таблицы соответствует содержимому (D+), а позиция вставленных данных указывается в этой команде с помощью (n).

В.3.10 ВЛИЯНИЕ НА ВРЕМЯ ВЫПОЛНЕНИЯ В СЛУЧАЕ ИМПУЛЬСНОЙ ВЕРСИИ КОМАНДЫ (FX3U)

Если прикладные команды запрограммированы в качестве импульсных команд, выполняемых по положительному фронту входного сигнала, время выполнения удлиняется:

- Команда MOV_P: при положительном фронте входного сигнала для выполнения необходимо время, указанное в таблице В-30 в столбце „Время включения“.
Если команда не выполняется, то действуют значения времени, которые в таблице В-30 указаны в столбце „Время выключения“.
- Все прочие прикладные команды: при импульсном варианте (с буквой „P“ в обозначении команды) табличные значения из раздела В.3, столбцы „Время включения“ и „Время выключения“, увеличиваются на 0,45 мкс.

В.4 КОДИРОВКА ASCII

Bits 3 ... 0		Bits 6 ... 4							
		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	'	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	!!	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	VS	/	?	O	←	o	DEL

Табл. В-41:ASCII-Code

Примеры кодировок ASCII:

00110100 = 34H: „3“

01000111 = 47H: „G“

00001101 = 0DH: CR (**C**arriage **R**eturn = возврат каретки)

MITSUBISHI ELECTRIC EUROPE B.V. /// РОССИЯ /// Москва /// Космодамианская наб., 52, стр. 5
Тел.: +7 495 721 20 70 /// Факс: +7 495 721 20 71 /// automation@mitsubishielectric.ru /// www.mitsubishi-automation.ru