

Серия ПК модели MELSEC FX

ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР

FX1S, FX1N, FX2N, FX2NC

К ЭТОМУ РУКОВОДСТВУ

Тексты, рисунки, диаграммы и примеры, находящиеся в этом руководстве, служат исключительно для подключения, обслуживания, программирования и использования программируемого контроллера (ПК) серии FX1S, FX1N, FX2N и FX2NC модели MELSEC.

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 изменения или изменения в этом руководстве без специальных указаний.

© 02/2004

РУКОВОДСТВО ПО ПРОГРАММИРОВАНИЮ
Программируемый контроллер (ПК) семейства MELSEC-FX
FX1S, FX1N, FX2N und FX2NC
Артикул-Nr.: 154315

ВЕРСИЯ	ИЗМЕНЕНИЯ / ДОПОЛНЕНИЯ / КОРРЕКТИРОВКА
A 02/2004	—

УКАЗАНИЯ ПО БЕЗОПАСНОСТИ

ДОПУСК СПЕЦИАЛИСТОВ

Это руководство предназначено исключительно на 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

Программируемый контроллер серии FX1S, FX1N, FX2N и FX2NC предусмотрен только для областей применения, которые описаны в этом руководстве. Внимательно изучите информацию представленную в данном руководстве. Могут использоваться только рекомендованные MITSUBISHI ELECTRIC дополнительные или расширяющие возможности устройства совместно с программируемым контроллером серии FX1S, FX1N, FX2N и FX2NC и A-серии.

Всякие 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, имеющих значение в специальных условиях.

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

- VDE-предписания
 - VDE 0100
Определения для сооружений силовых установок с номинальным напряжением до 1000 В.
 - VDE 0105
Работа силовых установок.
 - VDE 0113
Электрические установки с электронными аппаратами.
 - 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 подключаемых 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 обращения с 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 повреждение изоляции или обрыв. При обнаружении повреждения в кабельных соединениях нужно сейчас же обесточить устройства и кабельные соединения и заменить поврежденные кабельные соединения.
- Перед вводом в эксплуатацию проверьте соответствует ли местное напряжение сети допустимой области питающего напряжения.
- Предпримите требуемые меры, чтобы после отключения или прерывания напряжения прерванную программу можно было в нужном порядке 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 неопределенный 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.

Оглавление

1	Введение	
1.1	Общие положения	1-1
1.2	Применяемые программаторы/аппараты обслуживания	1-2
2	Основы программирования	
2.1	Обработка программы в ПК	2-1
2.1.1	Метод отображения процесса	2-2
2.1.2	Обработка команд в отличие от системы управления с проводным монтажом	2-3
2.2	Управляющие инструкции (команды)	2-4
2.2.1	Структура инструкции управления	2-4
2.2.2	Операнды	2-5
2.2.3	Виды изображения инструкций управления	2-6
2.2.4	Листинг присвоения адресов и присоединен	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	Таймеры	3-7
3.4.1	Адресация таймеров	3-7
3.4.2	Программирование таймеров	3-8
3.4.3	Задание уставок времени с помощью встроенного потенциометра системы управления серий FX1S и FX1N	3-9
3.4.4	Точность отсчета времени	3-9
3.4.5	Таймер с запоминанием	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	Шаговое управление	3-22
3.6.1	Адресация операндов шаговых состояний	3-22
3.7	Десятичные и шестнадцатеричные константы	3-23
3.7.1	Области числовых значений десятичных и шестнадцатеричные констант	3-23
3.8	Регистры	3-24
3.8.1	Классификация регистров	3-24
3.8.2	Структура регистров	3-25
3.8.3	Адресация регистров	3-26
3.8.4	Применение специальных регистров	3-26
3.8.5	Внешне изменяемые регистры	3-27
3.8.6	Введение индекса регистра	3-28
3.8.7	Введение регистра-файла	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	3-37
3.10	Прерывание-указатель	3-38
3.10.1	Адресация прерывания-указателя	3-38
3.11	Разветвление	3-43
3.11.1	Адресация операндов подпрограмм	3-43
4	Базовые команды	
4.1	Общие указания	4-1
4.1.1	Пояснения таблиц базового состава команд	4-1
4.2	Начало логических связей (LD, LDI)	4-5
4.3	Выдача результата логических связей (OUT)	4-6
4.4	Связь И (AND, ANI)	4-8
4.5	Связь ИЛИ (OR, ORI)	4-10
4.6	Импульсное начало связей (LDP, LDF)	4-12

4.7	Импульсная связь И (ANP, ANF)	4-14
4.8	Импульсная связь ИЛИ (ORP, ORF)	4-16
4.9	Связь И-блока (ANB)	4-18
4.10	Связь ИЛИ-блока (ORB)	4-19
4.11	Обработка результата 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	Создание одноразового 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)	4-33
4.18	Пример программы пользователя	4-34
4.18.1	Опрос входа	4-35

5 STL-Инструкция

5.1	Общие указания	5-1
5.1.1	Пример применения с использованием STL-инструкции	5-2
5.1.2	Схематичный процесс пошагового управления	5-3
5.1.3	Представление процесса управления в Fluss диаграмме	5-4
5.2	Программирование STL-инструкций	5-5
5.3	Инициализация статуса шага	5-10
5.4	STL-разветвление	5-11
5.4.1	Простой процесс	5-11
5.4.2	Селективное разветвление	5-12
5.4.3	Параллельное разветвление	5-14
5.4.4	Комбинации селективных и параллельных разветвлений	5-16
5.4.5	Программирование пустого состояния	5-17
5.4.6	Разветвление скачком (переходом)	5-18
5.5	Пример контроля загрузки и разгрузки	5-20
5.6	Пример процессов сортировки и транспортировки	5-22

6	Прикладные инструкции	
6.1	Общие указания	6-1
6.1.1	Пояснения к описанию используем. инструкций.	6-1
6.1.2	Описание операндов	6-2
6.1.3	Обработка слов	6-3
6.1.4	Структура данных	6-5
6.1.5	Исполнение используемых инструкций	6-6
6.1.6	Введение индексов регистра V, Z	6-7
6.1.7	Значения флагов	6-8
6.1.8	Ошибки процесса проектирования при исполнении используемых инструкций	6-8
6.1.9	32-х битовые инструкции	6-8
6.1.10	Обзор используемых инструкций.	6-9
6.2	Инструкции процесса отработки программы	6-11
6.2.1	Переход внутри программы (CJ)	6-12
6.2.2	Вызов подпрограммы (CALL)	6-16
6.2.3	Конец подпрограммы (SRET)	6-17
6.2.4	Ввод прерывания программы (IRET, EI, DI)	6-18
6.2.5	Конец подпрограммы (FEND)	6-22
6.2.6	Обновление таймера контроля времени работы программы (WDT)	6-23
6.2.7	Повторение части программы (FOR, NEXT)	6-25
6.3	Инструкции сравнения и преобразования	6-27
6.3.1	Сравнение числовых данных (CMP, DCMP)	6-28
6.3.2	Сравнение числовых областей данных (ZCP, DZCP)	6-30
6.3.3	Пересылка данных (MOV, DMOV)	6-32
6.3.4	Пересылка со смещением данных (SMOV)	6-33
6.3.5	Пересылка с инверсией (CML)	6-36
6.3.6	Передача блоков (BMOV)	6-37
6.3.7	Специальное копирование (FMOV)	6-38
6.3.8	Обмен данными (XCH)	6-39
6.3.9	Конвертирование BCD данных (BCD, DBCD)	6-41
6.3.10	Конвертирование двоичных данных (BIN, DBIN)	6-43
6.4	Арифметические инструкции	6-45
6.4.1	Суммирование 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	Инкрементирование (приращение) (INC, DINC)	6-53
6.4.6	Декрементирование (уменьшение) (DEC)	6-54
6.4.7	Логическая И-связь двоичных данных (WAND, DAND)	6-55
6.4.8	Логическая ИЛИ-связь двоичных данных (WOR, DOR)	6-56
6.4.9	Логическая исключающая ИЛИ-связь двоичных данных (WXOR, DXOR)	6-57
6.4.10	Инверсия данных (NEG).	6-58
6.5	Инструкции сдвига	6-59
6.5.1	Сдвиг вправо (ROR)	6-60
6.5.2	Сдвиг влево (ROL)	6-61
6.5.3	Кольцевой сдвиг бита вправо (RCR)	6-62
6.5.4	Кольцевой сдвиг бита влево (RCL)	6-63
6.5.5	Сдвиг побитно двоичных данных (SFTR, SFTL)	6-64
6.5.6	Сдвиг побитно двоичных данных вправо (WSFR)	6-66
6.5.7	Сдвиг побитно двоичных данных влево (WSFL)	6-67
6.5.8	Запись в FIFO-накопитель (SFWR)	6-68
6.5.9	Чтение из FIFO-накопителя (SFRD)	6-69
6.6	Операции с данными	6-71
6.6.1	Выключить область операндов (ZRST)	6-72
6.6.2	Декодировка данных (DECO)	6-73
6.6.3	Кодировка данных (ENCO)	6-75
6.6.4	Определение включенного бита (SUM)	6-77
6.6.5	Проверка бита (BON)	6-78
6.6.6	Определение среднего значения (MEAN)	6-79
6.6.7	Пуск интервала времени (ANS)	6-80
6.6.8	Выключение битового показания (ANR)	6-81
6.6.9	Вчисление корня квадратного (SQR)	6-82
6.6.10	Преобразование формата числа (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)	6-88
6.7.4	Включение и отключение с помощью высокоскоростного счетчика (DHSCS, DHSCR)	6-91
6.7.5	Сравнение областей (DHSZ)	6-93
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 [ШИМ] (PWM)	6-103
6.7.9	Выдача определенного числа импульсов (PLSR)	6-105

6.8	Инструкции ориентированные на 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	Функция пульс-пары (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 инструкции FNC

7.1	Общие указания	7-1
7.1.1	Общий обзор специальных инструкций FNC	7-1
7.2	Инструкции входов/выходов	7-4
7.2.1	Десятичная клавиатура (TKY)	7-5
7.2.2	Шестнадцатеричная клавиатура (HKY)	7-7
7.2.3	Двоичный выключатель (DSW)	7-10
7.2.4	7-ми сегментный индикатор (SEGD)	7-12
7.2.5	7-ми сегментный индикатор с запоминанием (SEGL)	7-13
7.2.6	7-ми сегментный индикатор с дополнительными клавишами (ARWS)	7-16
7.2.7	ASCII-конвертирование (ASC)	7-19
7.2.8	Выдача данных по выходам (PR)	7-21
7.2.9	Чтение данных из специального модуля (FROM)	7-23
7.2.10	Запись данных в специальный модуль (TO)	7-25
7.3	Коммуникационные инструкции	7-27
7.3.1	Прием и передача данных (RS)	7-28
7.3.2	Перенос входов или меркеров (PRUN)	7-34
7.3.3	ASCII-преобразование (ASCI)	7-35
7.3.4	Шестнадцатеричная преобразование (HEX)	7-37
7.3.5	Проверка на сумму и паритет (CCD)	7-39
7.3.6	Чтение задаваемых значений из FX1N-8AV-BD и FX2N-8AV-BD (VRRD)	7-41
7.3.7	Чтение установки переключателя из FX1N-8AV-BD и FX2N-8AV-BD (VRSC)	7-42
7.3.8	Программирование замкнутой цепи регулирования (PID)	7-43

7.4	Инструкции чисел с плавающей запятой (FX2N)	7-49
7.4.1	Сравнение чисел с плавающей запятой (DECMP)	7-50
7.4.2	Зонный компаратор чисел с плавающей запятой (DEZCP)	7-51
7.4.3	Преобразование формата чисел с плавающей запятой в натуральную форму числа (DEBCD)	7-53
7.4.4	Преобразование из натуральной формы в формат с плавающей запятой (DEBIN)	7-54
7.4.5	Сложение чисел с плавающей запятой (DEADD)	7-55
7.4.6	Вычитание чисел с плавающей запятой (DESUB)	7-56
7.4.7	Умножение чисел с плавающей запятой (DEMUL)	7-57
7.4.8	Деление чисел с плавающей запятой (DEDIV)	7-58
7.4.9	Корень квадратный из числа с плавающей запятой (DESQR)	7-59
7.4.10	Преобразование формата чисел с плавающей в целое число (INT)	7-60
7.4.11	Расчеты синуса с числами с плавающей запятой (DSIN)	7-61
7.4.12	Расчеты косинуса с числами с плавающей запятой (DCOS)	7-62
7.4.13	Расчеты тангенса с числами с плавающей запятой (DTAN)	7-63
7.5	Инструкции обработки данных	7-64
7.5.1	Обмен старшего и младшего байтов (SWAP)	7-65
7.6	Инструкции позиционирования (FX1S/FX1N)	7-66
7.6.1	Предупреждения при использовании инструкций позиционирования	7-66
7.6.2	Формат выходных импульсов	7-67
7.6.3	Операнды используемые при позиционировании	7-68
7.6.4	Пример подключения сервопривода	7-69
7.6.5	Пример программы	7-70
7.6.6	Чтение абсолютной текущей позиции (DABS)	7-77
7.6.7	Перемещение к нулевой(исходной) точке (ZRN)	7-78
7.6.8	Выдача импульсов с переменной частотой (PLSV)	7-80
7.6.9	Позиционирование по инкрементальным значениям (DRVI)	7-82
7.6.10	Позиционирование по абсолютному значению (DRVA)	7-84
7.7	Инструкции реального времени	7-86
7.7.1	Сравнение данных времени (TCMP)	7-87
7.7.2	Сравнение данных времени с областью (TZCP)	7-89
7.7.3	Сложение данных времени (TADD)	7-91
7.7.4	Вычитание данных времени (TSUB)	7-93
7.7.5	Чтение данных времени (TRD)	7-95
7.7.6	Запись данных времени (TWR)	7-97
7.7.7	Рабочий счетчик часов (HOUR)	7-98

7.8	Инструкции Кода Грея	7-99
7.8.1	Преобразование целого числа в код Грея (GRY)	7-99
7.8.2	Преобразование кода Грея в целое число (GBIN)	7-100
7.9	Обмен данными с аналоговым модулем FX0N-3A (FX1N)	7-101
7.9.1	Чтение аналоговых входных значений (RD3A)	7-101
7.9.2	Запись аналоговых выходных значений (WR3A)	7-102
7.10	Внешняя кассета памяти	7-103
7.9.2	Функции внешней ROM кассеты(EXTR)	7-103
7.11	Инструкция сравнения II	7-124
7.11.1	Загрузка сравниваемого (LDI)	7-125
7.11.2	Сравнение И-связи (ANDI)	7-127
7.11.3	Сравнение ИЛИ-связи (ORI)	7-129

8 Специальные функции

8.1	Содержание данных в режиме СТОП	8-2
8.2	Работа с постоянным временем цикла	8-3
8.3	Функции слова пароля	8-4
8.4	Фиксация импульсов	8-5
8.5	Настройка входного фильтра	8-7
8.6	Интегрированный потенциометр у FX1S и FX1N	8-8
8.7	Функции реального времени	8-9
8.8	Файл-Регистр	8-10
8.9	Переключение Работа/Стоп	8-11
8.10	FX2N-базовые аппараты с питанием 24 V DC	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 нежелательного "доступа"	8-15
8.11.5	Изменение состояния или значения операнда	8-15
8.11.6	Автоматическое выключение освещения отображения	8-16
8.11.7	Запуск и блокировка отображения ошибок	8-16

9	Специальные меркеры, специальные регистры	
9.1	Специальные меркеры (M8000–M8255)	9-1
9.1.1	Статус ПК (M8000–M8009)	9-2
9.1.2	Время такта (M8011–M8019)	9-3
9.1.3	Флаги (M8020–M8029)	9-4
9.1.4	Режим ПК (M8030–M8039)	9-5
9.1.5	STL-состояние (M8040–M8049)	9-6
9.1.6	Прерывание программы (M8050–M8059)	9-7
9.1.7	Фиксация импульсов	9-8
9.1.8	Функции связи и особые функции (M8070–M8198)	9-9
9.1.9	Счетчик вверх/вниз (M8200–M8254)	9-11
9.2	Специальные регистры (D8000–D8195)	9-14
9.2.1	Статус ПК (D8000–D8009)	9-14
9.2.2	Время такта (D8010–D8019)	9-15
9.2.3	Флаги (D8020–D8029)	9-15
9.2.4	Режим ПК (D8030 – D8039)	9-16
9.2.5	STL-состояние (D8040–D8049)	9-16
9.2.6	Регистры функций связи и специальных функций (D8070 – D8099)	9-17
9.2.7	Прочие регистры (D8102 – D8109)	9-17
9.2.8	Регистр для коммуникационных адаптеров (232ADP, 485ADP) (D8120 – D8129)	9-18
9.2.9	Исполнительный регистр для инструкций HSZ и PLSY (D8130 – D8143)	9-19
9.2.10	Специальные Регистры (D8158 – D8164)	9-20
9.2.11	Индекс-регистр (D8182 – D8195)	9-20
10	Ошибки программы	
10.1	Опознание ошибок	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 (6101–6409)	10-3
10.2.2	Коды ошибок (6501–6511)	10-4
10.2.3	Коды ошибок (6601–6609)	10-5
10.2.4	Коды ошибок (6610–6632)	10-6
10.2.5	Коды ошибок (6701–6709)	10-7
10.2.6	Коды ошибок (6730–6752)	10-8

A Технические данные	
A.1	Обзор базовых команд A-1
A.2	Общие системные данные MELSEC FX1S A-4
A.3	Операнды MELSEC FX1S A-4
A.4	Общие системные данные MELSEC FX1N A-6
A.5	Операнды MELSEC FX1N A-6
A.6	Используемые инструкции MELSEC FX1S/FX1N A-8
A.7	Общие системные данные MELSEC FX2N A-10
A.8	Операнды MELSEC FX2N A-10
A.9	Используемые инструкции MELSEC FX2N A-12
B Времена исполнения инструкций	
B.1	Времена исполнения FX1S/FX1N-серии B-1
B.1.1	Базовые команды и инструкции шаговых состояний B-1
B.1.2	Инструкции разветвления программы B-3
B.1.3	Инструкции сравнения и передачи B-3
B.1.4	Арифметические инструкции B-4
B.1.5	Ротационные и сдвигающие инструкции B-4
B.1.6	Операции с данными B-5
B.1.7	Высокоскоростные инструкции B-5
B.1.8	Инструкции ориентированные на пользователя B-6
B.1.9	Специальные FNC-инструкции B-6
B.1.10	Инструкции позиционирования B-7
B.1.11	Инструкции для часов реального времени B-7
B.1.12	Обмен данными с входным аналоговым модулем FX0N-3A B-7
B.1.13	Инструкции сравнения B-8
B.2	Времена исполнения FX2N B-9
B.2.1	Базовые команды и инструкции шаговых состояний B-9
B.2.2	Инструкции разветвления программы B-11
B.2.3	Инструкции сравнения и передачи B-11
B.2.4	Арифметические инструкции B-12
B.2.5	Ротационные и сдвигающие инструкции B-13
B.2.6	Операции с данными B-14
B.2.7	Высокоскоростные инструкции B-14
B.2.8	Инструкции ориентированные на пользователя B-15
B.2.9	Специальные FNC-инструкции B-16

1 ВВЕДЕНИЕ

1.1 ОБЩИЕ ПОЛОЖЕНИЯ

Области применения

Предлагаемое руководство описывает и поясняет все операнды, а также распределение адресов входов и выходов, которое нужно для программируемой системы управления MELSEC серий FX1S, FX1N, FX2N и FX2NC.

Информация для инсталляции, 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:

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.

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.

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):

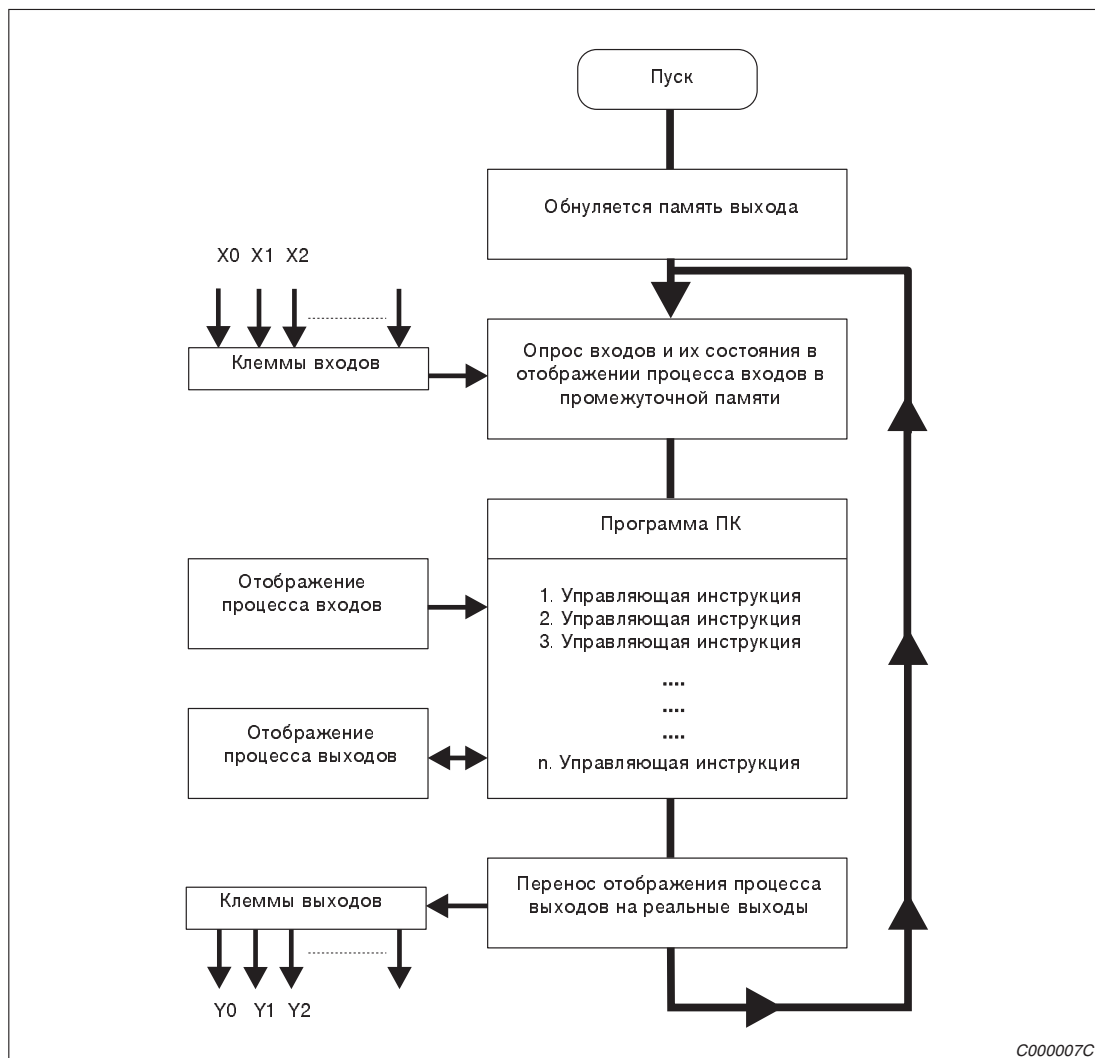


Рис.2-1. Обработка программы по методу отображения процесса

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

Связи, которые касаются выходов, записываются соответственно в промежуточную память 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 поэтому 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“	„Y“	„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

Программирование ПК может выполняться в трех различных видах изображения программы 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.

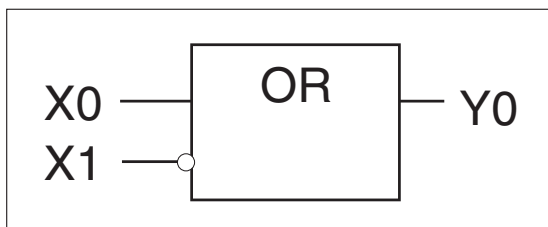


Рис.2-2.
Пример функционального блок

C000381C

КОНТАКТНАЯ СХЕМА

Контактная схема является производной от релейно-контакторной 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

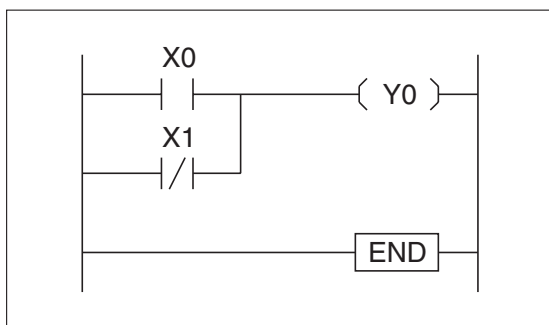


Рис. 2-3.
Пример контактной схемы

C000004C

2.2.4 ЛИСТИНГ РАСПРЕДЕЛЕНИЯ И ПОДКЛЮЧЕНИЕ ПК

Наряду с описание 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.

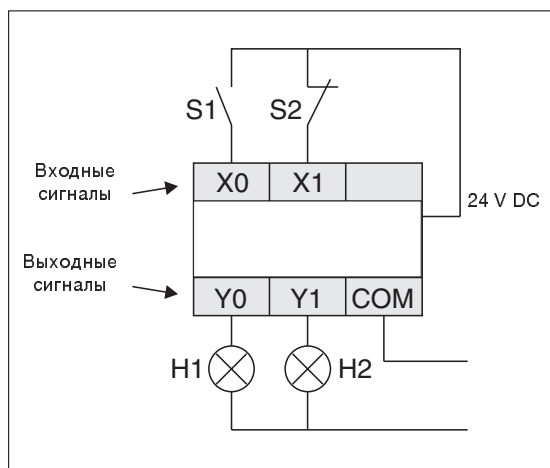


Рис.2-4.
Пример присоединения к ПК

C000005C

3 ОПЕРАНДЫ

3.1 ОБЗОР ОПЕРАНДОВ

Эта глава описывает все имеющиеся операнды и их возможности 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
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	
Таймер	T	64	256	256	
Счетчик	C	53	256	256	
Высокоскоростной счетчик (используются одновременно)	C	21	21	21	
Состояние шага	S	128	1000	1000	
Десятичная константа	K	16/32 бит	16/32 бит	16/32 бит	
шестнадцатеричного константа	H	16/32 бит	16/32 бит	16/32 бит	
Регистр данных	D	256	8000	8000	
Регистр файлов	D	–	7000 (частично)	7000 (частично)	
Регистр индексации	V, Z	16	16	16	
Указатель	P	64	128	128	
Указатель прерывания	I	6	6	6 входов 3 таймера	
Разветвление программы	N	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. 128a
	Выходы	Y0 ... Y157	max. 128a
FX2N	Входы	X0 ... X317	max. 256 b
FX2NC	Выходы	Y0 ... Y317	max. 256 b

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).

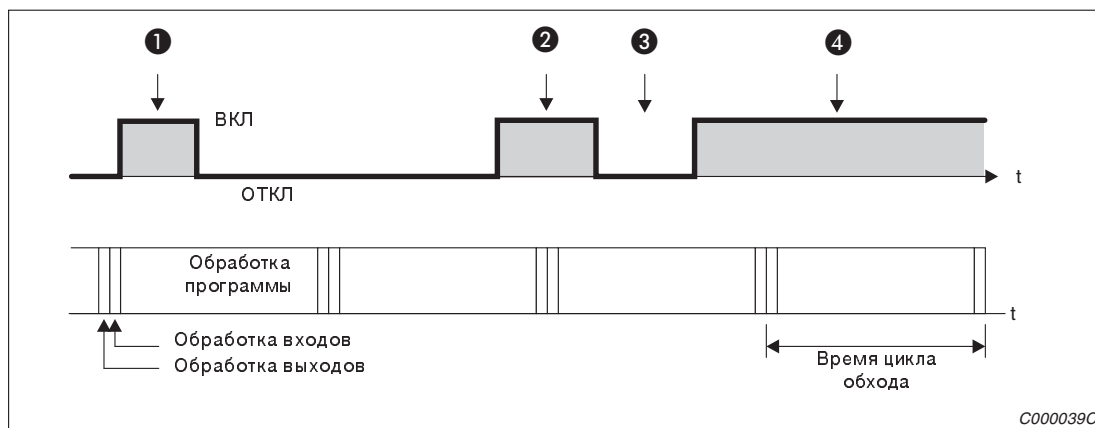


Рис.3-1. Обработка входных сигналов с коротким временем импульса

- ① и ③: это состояние входа не опознается;
- ②: это состояние входа случайно опознали;
- 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

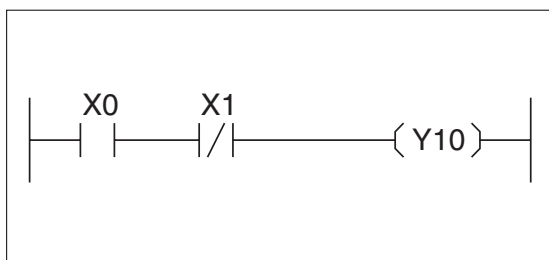


Рис.3-2.
Пример программирования с использованием входов и выходов

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) представляют в распоряжение так называемые меркеры с памятью при снятии напряжения и специальные меркеры.

- Меркеры с памятью сохраняют свою информацию также 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	Меркер1	–	M0 – M3071	3072
	Меркер с памятью	–	M500 – M3071	2572 (частично)
	Специальный меркер	–	M8000 – M8255	256

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.).

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 памяти в рабочей памяти ПК.

ПРИМЕР ▾

Применения меркера

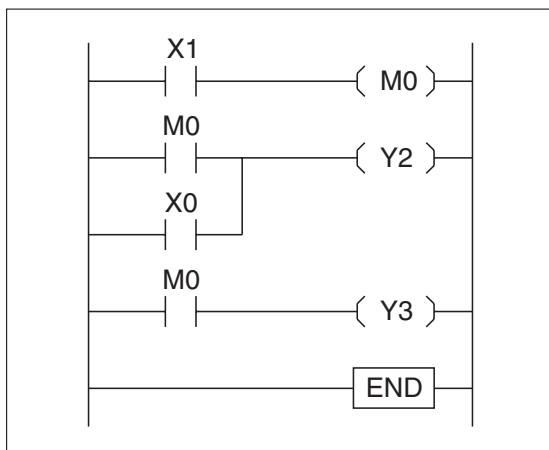


Рис.3-3.
Пример применения меркера

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

Для некоторых процессов управления, как например, зависимое от времени присоединение двигателя вентилятора, необходимы реле времени. В релейной технике для этого 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 меркер M8028
FX1S	T0 – T62	63	100 мс	0 – 3276,7 с	ОТКЛ
	T0 – T31	32	100 мс	0 – 3276,7 с	ВКЛ
	T32 – T62	31	10 мс	0 – 327,67 с	ВКЛ
	T63	1	1 мс	0 – 32,767 с	–
FX1N	T0 – T199	200	100 мс	0 – 3276,7 с	–
	T200 – T245	46	10 мс	0 – 327,67 с	–
	T246 – T249	4	1 мс	0 – 32,767 с	–
	T250 – T255 (с памятью)	6	100 мс	0 – 3276,7 с	–
FX2N FX2NC	T0 – T199	200	100 мс	0 – 3276,7 с	–
	T200 – T245	46	10 мс	0 – 327,67 с	–
	T246 – T249	4	1 мс	0 – 32,767 с	–
	T250 – 255 (с памятью)	6	100 мс	0 – 3276,7 с	–

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.

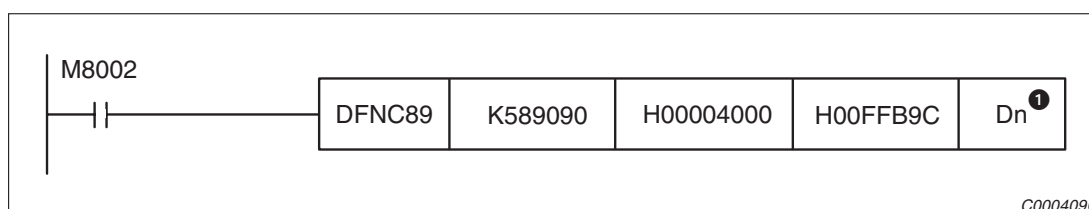


Рис.3-4. Добавление программы при применении таймера T63 для ПК FX1S версии от V1.00 до V1.30

- ❶ 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 K, 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

Задание временной константы может выполняться также косвенно посредством записанного заранее в регистр данных десятичного числа.

ПРИМЕР ▾

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

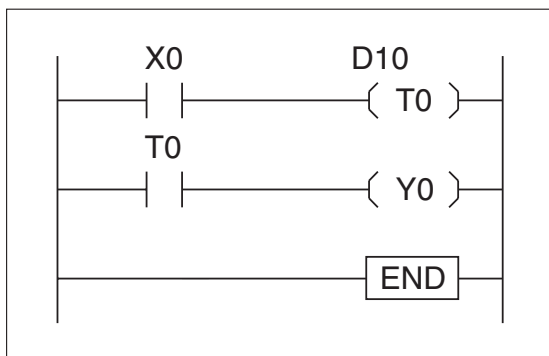


Рис. 3-5.
Пример программирования с использованием таймера

C000201C

Уставка времени записана в регистр данных D10.

△

ПРИМЕР ▾

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

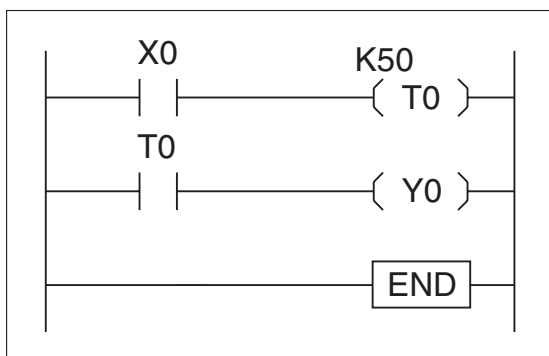


Рис. 3-6.
Указание временной константы с помощью константы

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 вращаемых потенциометров можно вручную 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.

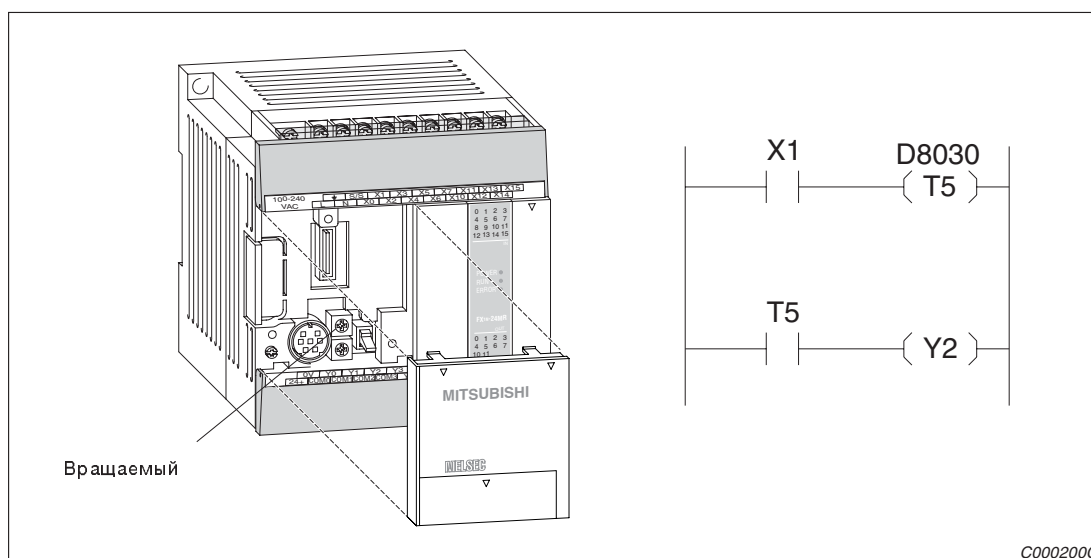


Рис.3-7. Пример настройки задания уставки времени в регистр данных D8030 с помощью потенциометра

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: $(T - \alpha) \leq T \leq (T + T0)$

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. $(T + T0) + T0 = T + 2T0$.

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, содержащее которой сохраняется и при отключении напряжения.

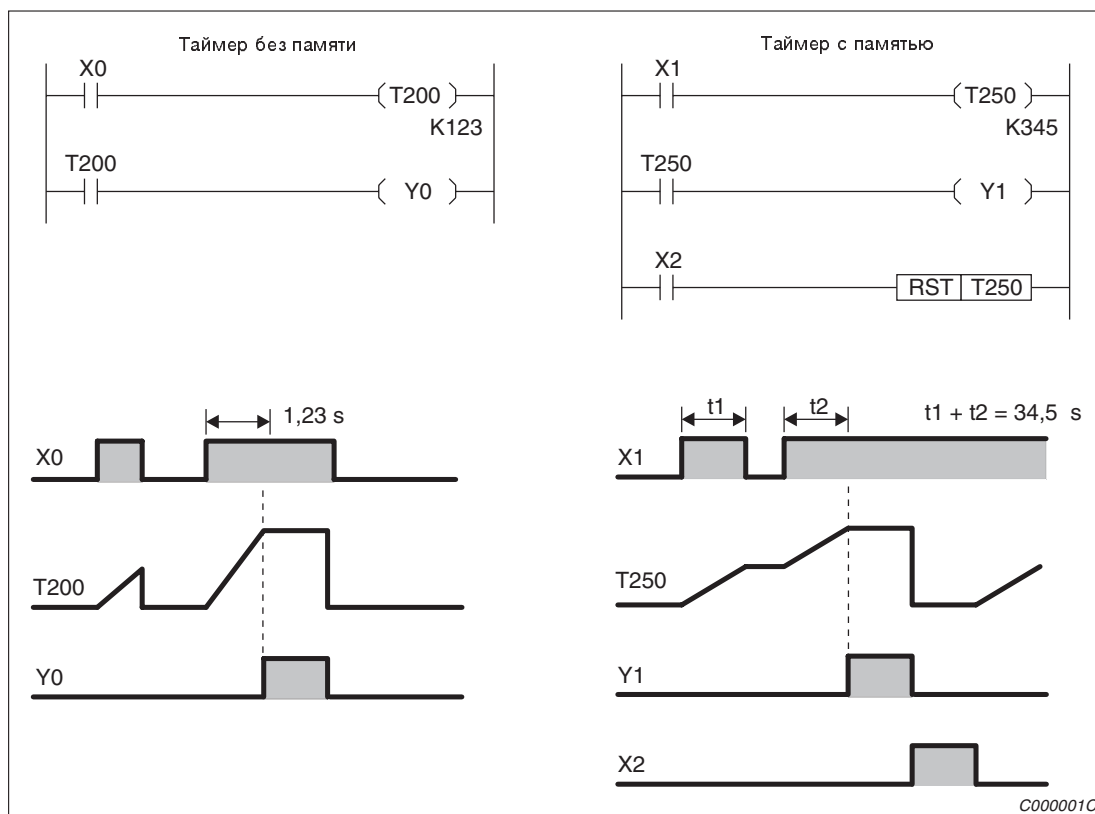


Рис.3-8. Принцип работы таймера с памятью

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	-
	C100 – C199	100	●
FX2NC	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 между +1 и +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

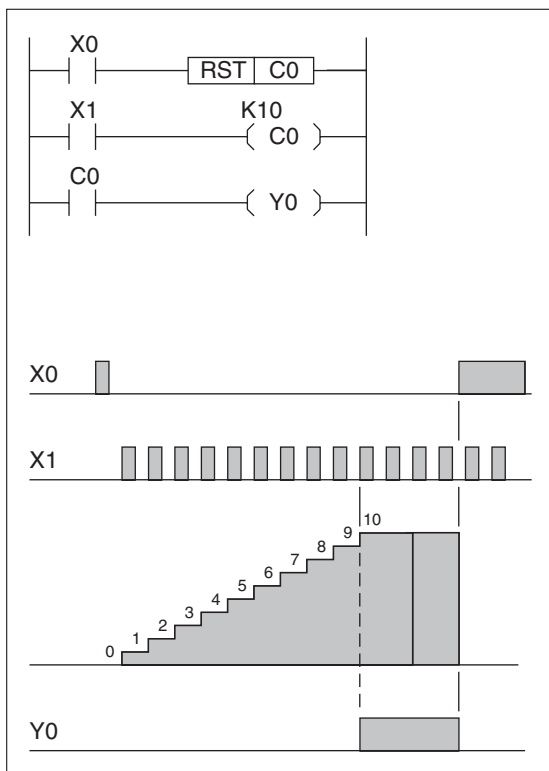


Рис.3-9.
Пример программирования с использованием 16-ти битного счетчика с непосредственно предварительно задаваемым числом счета

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

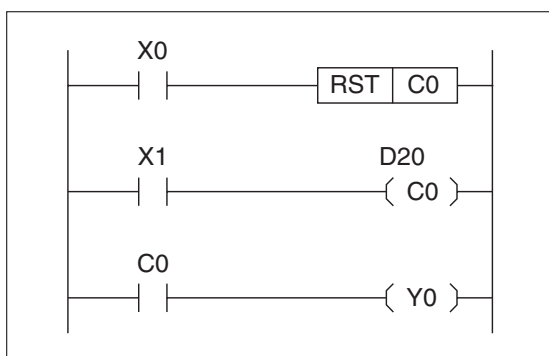


Рис.3-10.
Пример программирования с использованием 16-ти битного счетчика с косвенно предварительно задаваемым числом счета

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	-
	C220 ... C234	15	●
FX2NC	C200 ... C219	20	-
	C220 ... C234	15	⑰

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

Направление счета счетчика определяется по логическому состоянию предписанных 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

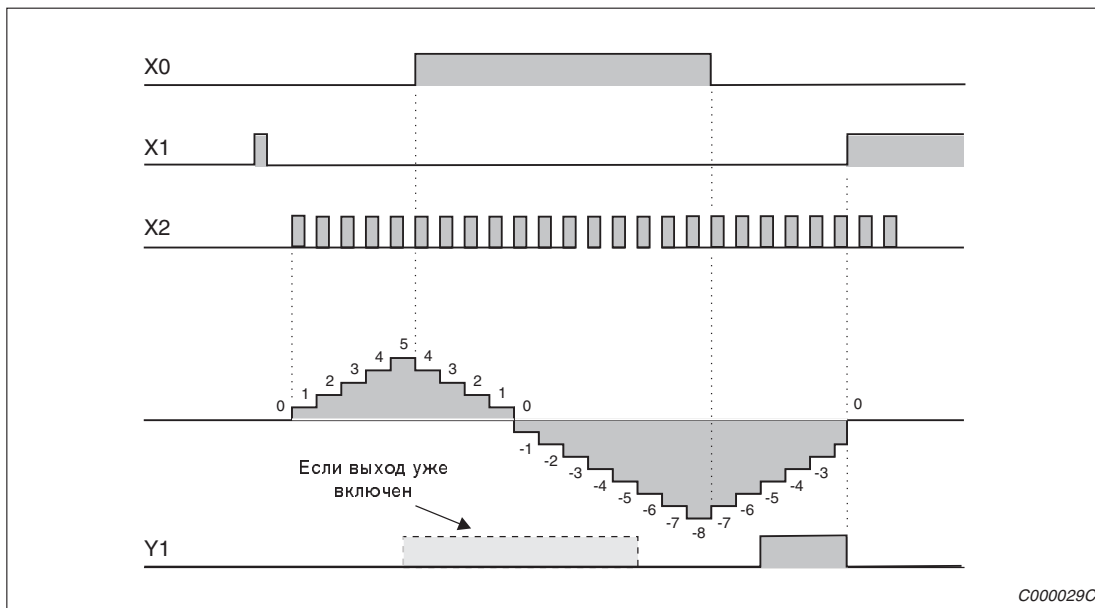


Рис.3-11. Пример программирования с использованием 32-х битного счетчика с прямым предварительно задаваемым числом счета

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

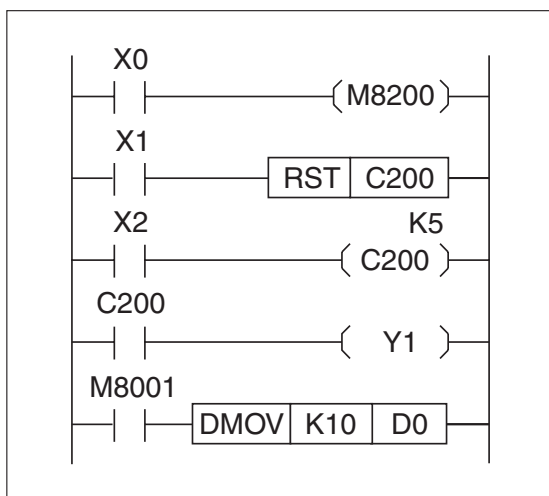


Рис.3-12. Пример программирования с использованием 32-х битного счетчика с косвенным предварительно задаваемым числом счета

C000030C

3.5.3 32-Х БИТНЫЙ ВЫСОКОСКОРОСТНОЙ СЧЕТЧИК

Высокоскоростной счетчик является 32-х битным счетчиком, который обрабатывает быстрые внешние сигналы счета. В качестве входов счета имеются входы от X0 до X7. Двойное присвоение входов не разрешено, поэтому каждый ПК максимально располагает 7-ю высокоскоростными счетчиками.

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 с пуском и сбросом					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	C 244	C 245	C 246	C 247	C 248	C 249	C 250	C 251	C 252	C 253	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			A		A
X4					U/D				U/D					D		D			B		B
X5						U/D			R					R		R			R		R
X6										S					S					S	
X7											S					S					S

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

- 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-го или 2-х фазных счетчиков: макс. 10 кГц
- АВ-фазного счетчика: макс. 5 кГц

Сумма всех частот счета для ПК серий FX1S или FX1N не должна превышать 60 кГц и для ПК серии FX2N или FX2NC 30 кГц. При этом нужно следить, чтобы частота АВ-счетчика в этом расчете согласовывалась с двойным значением.

Входы X0 и X1 ПК серии FX2N благодаря их конструкции в состоянии считать очень высокую частоту:

При применении 1-но фазных счетчиков C235, C236 и 2-х фазного C246 можно считать до 60 кГц. При применении АВ-счетчика C251 можно считать до 30 кГц.

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

Сумма всех частот счета на всех входах ПК должна быть ≤ 60 кГц (АВ-фазный счетчик может считать двойное значение)

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
Специальные 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)

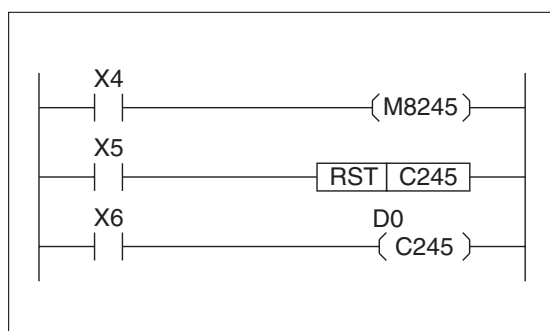


Рис.3-14.

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)

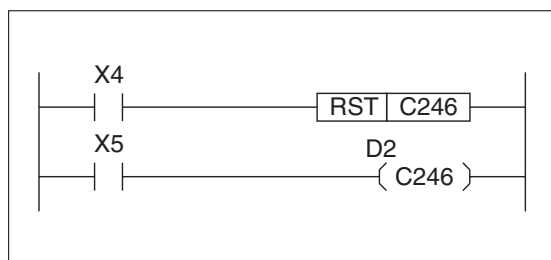


Рис.3-15.

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")

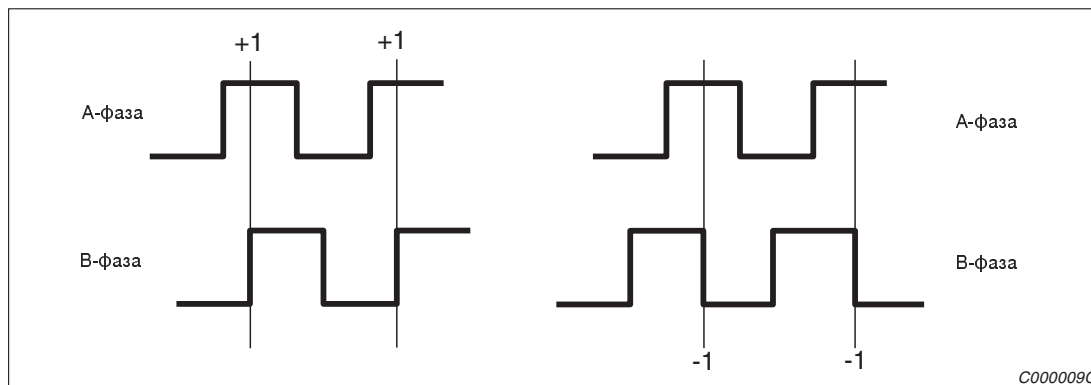


Рис.3-16. АВ-фазный счетчик

Каждая смена сигнала на входе с возрастающим фронтом на В-фазе позволяет счетчику 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)

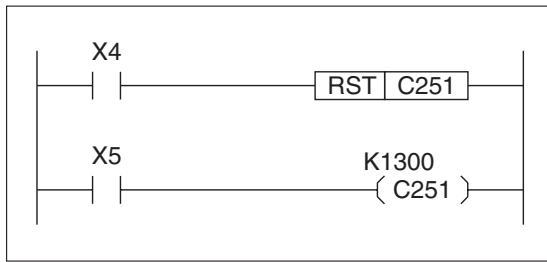


Рис.3-17.

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

Тип ПК	Операнды	Из них буферизированные
FX1S	S0 - S127 (128)	S0 - S127 (128)
FX1N	S0 - S999 (1000)	S0 - S999 (1000)
FX2N	S0 - S999 (1000)	S500 - S999 (500)
FX2NC		

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

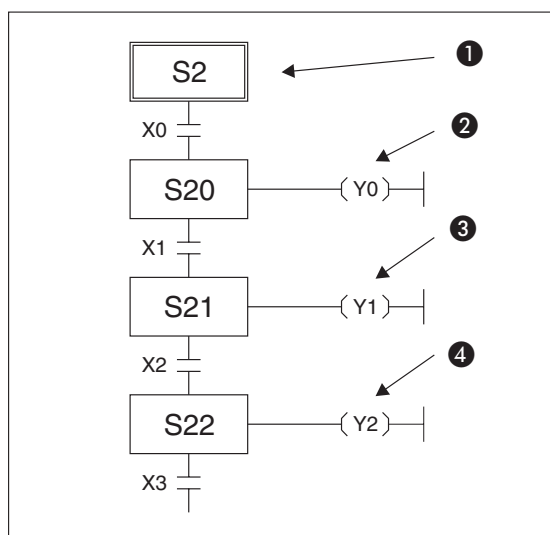


Рис 3-18.

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 (B опускание).

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

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

3.7 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.

3.7.1 ОБЛАСТИ ЧИСЛОВЫХ ЗНАЧЕНИЙ ДЕСЯТИЧНЫХ И ШЕСТНАДЦАТЕРИЧНЫХ КОНСТАНТ

Константы	16 бит	32 бит
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

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 индцирования 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 параметров или рецептуры необходимы регистры файлов. Для ПК серий FX1N, или FX2N или FX2NC области памяти этих регистров устанавливаются пользователем. Эти регистры файлов являются частью регистров файлов с запоминанием.

3.8.2 СТРУКТУРА РЕГИСТРА

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

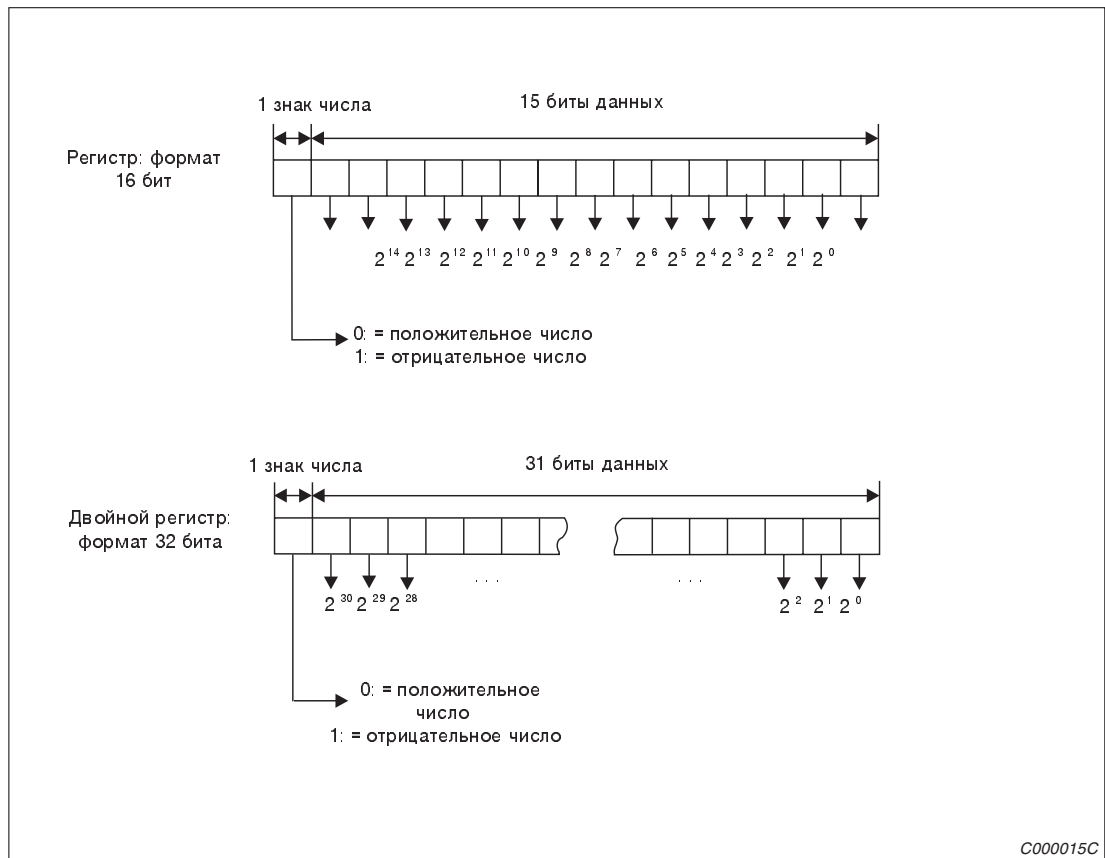


Рис.3-19. Структура регистра (16 бит) и двойного регистра (32 бита)

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)

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

① Оба этих регистра находятся также в специальных регистрах.

УКАЗАНИЕ

Из указанных областей данных регистров ПК серии FX2N и FX2NC пользователем могут определяться буферизированные они или нет только у регистров от D200 до D511. Регистры данных от D512 в FX2N всегда буферизированы.

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.

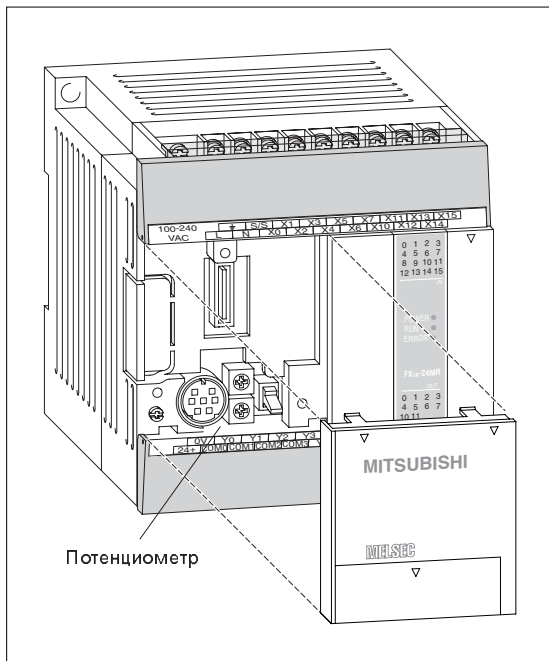


Рис.3-20. Потенциометры на ПК серии FX1S/FX1N

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

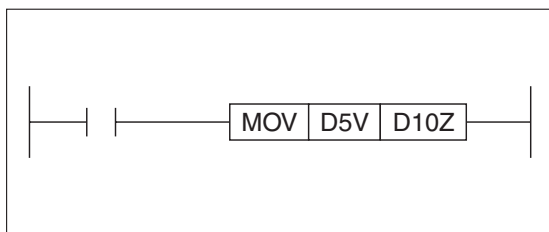


Рис.3-21.

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 РЕГИСТРОВ ФАЙЛОВ

Регистры файлов записываются блоках 500 адресами в программной области памяти (EPROM или EEPROM) ПК. Количество блоков устанавливается в параметрах. Доступ к регистрам файлов возможен через программаторы и терминалы обслуживания.

УКАЗАНИЕ

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

При применении регистров файлов в ПК серии FX1N, FX2N и FX2NC нужно обращать внимание на то, что общая область памяти включает область не разрушаемой памяти.

ЧТЕНИЕ РЕГИСТРОВ ФАЙЛОВ

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

ЗАПИСЬ РЕГИСТРОВ ФАЙЛОВ

В ПК серии FX1S регистры файлов могут записываться только с программатора или с персонального компьютера и соответствующего программного обеспечения.

В ПК серии FX1N, FX2N и FX2NC содержание регистров файлов также могут изменяться в программе ПК при применении MOV-инструкции.

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

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

УКАЗАНИЕ

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

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

Регистры файлов, 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 негативным, если в высшем разряде бита (знак числа) стоит число 1.

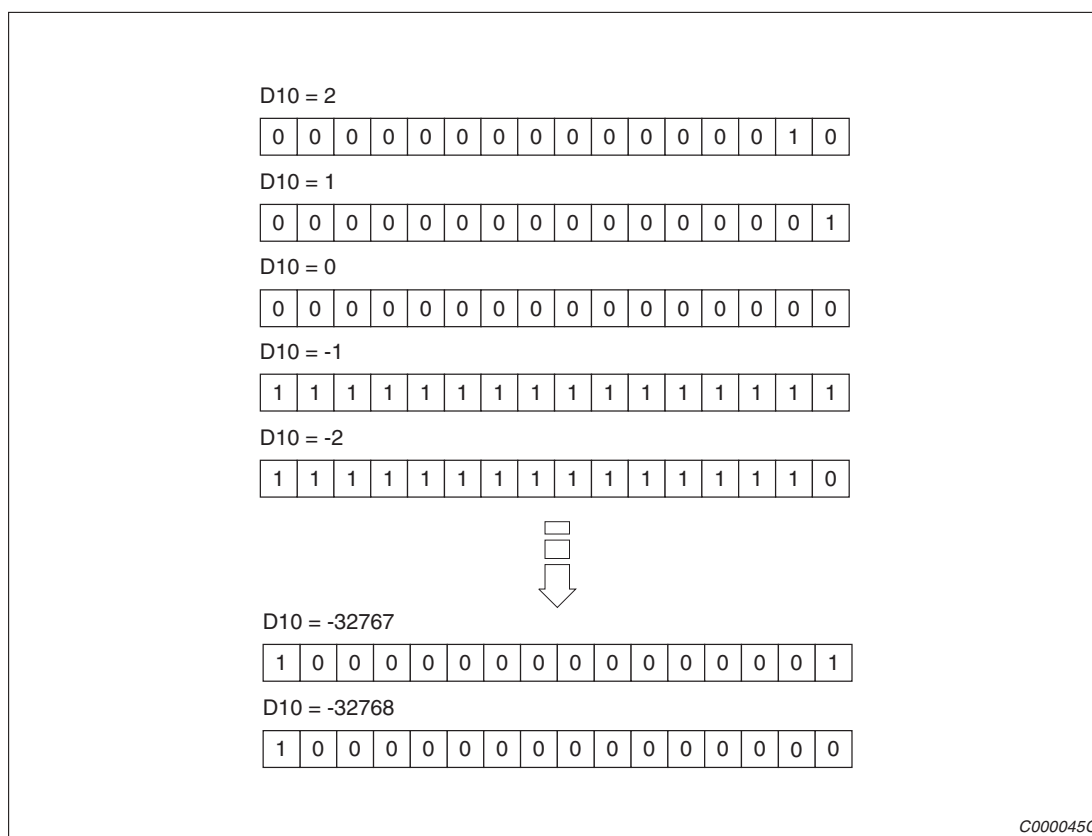


Рис.3-22. Примеры представления отрицательных чисел



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.

ДЕСЯТИЧНАЯ СИСТЕМА ЧИСЕЛ

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 × 10² + 5 × 10¹ + 1 × 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 D 120 $\times 10D^{121}$.

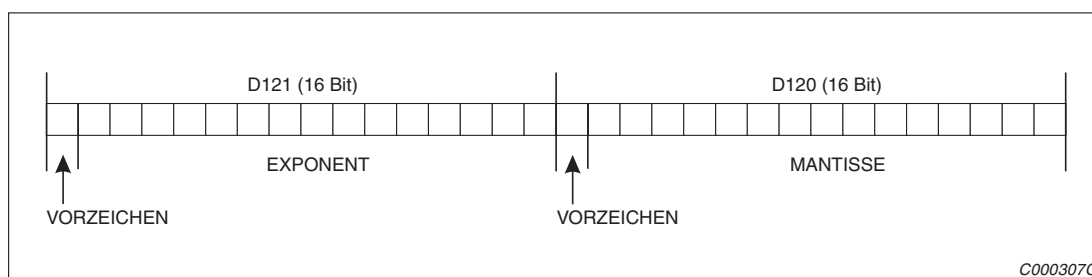


Рис.3-23. Расположение в регистре данных



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.

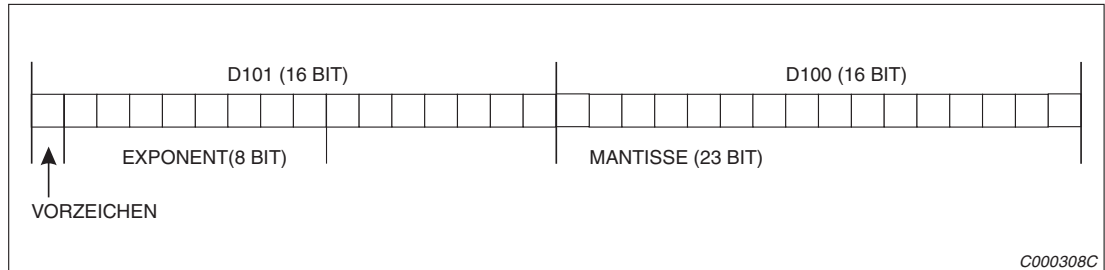


Рис.3-24. Размещение бит в регистре данных

Формат: + - Мантисса x 2^{Экспонента}

Область значений:

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

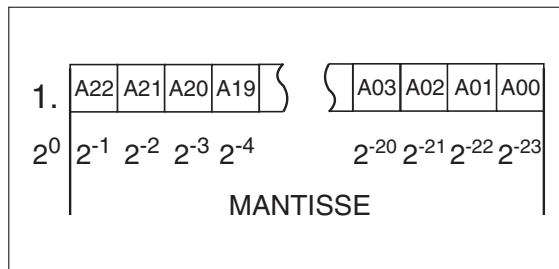


Рис.3-25.
Мантисса

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

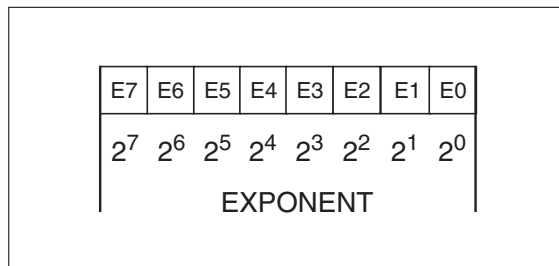
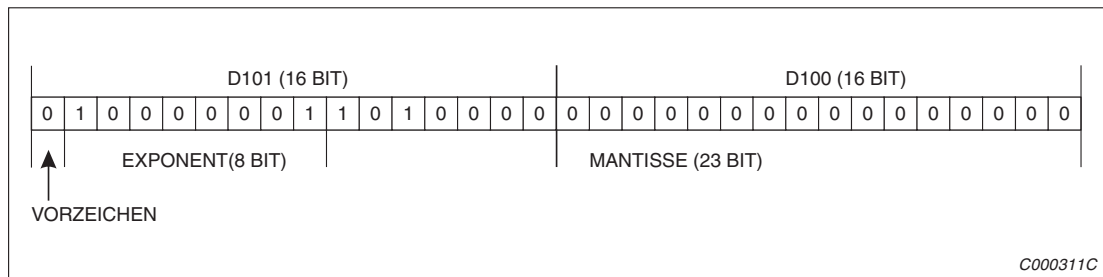


Рис.3-26.
Экспонента

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

$$\begin{aligned} & 1 \times 2^0 + \text{представлена числом } 10100000000000000000000 \text{ это соответствует } 1,101_{\text{BIN}} \text{ и} \\ & 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + \dots + 0 \times 2^{-23} \\ & = 1,625 \end{aligned}$$

В результате число равно + 1,625 x 2² = 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 (восьмеричное)

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

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

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

△

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

30 (десятичное)

$$30 : 8 = 3 \text{ остаток } 6$$

$$3 : 8 = 0 \text{ остаток } 3$$

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

△

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

374 (восьмеричное)

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

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

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

△

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 → ШЕСТНАДЦАТИРИЧНОЕ 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: ШЕСТНАДЦАТИРИЧНОЕ ЧИСЛО → ДЕСЯТИЧНОЕ ЧИСЛО****ПРИМЕР ▾**

7A (шестнадцатеричное)

7A (шестнадцатеричное) = $7 \times 16^1 + 10 \times 16^0$ 7A (шестнадцатеричное) = $112 + 10$ 7A (шестнадцатеричное) = 122 (десятичное) △**VCD - ФОРМАТ (ДВОИЧНО-ДЕСЯТИЧНЫЙ КОД)**

В VCD-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 VCD-формате допустимо кодирование только десятичных цифр от 0 до 9.

КОДИРОВАНИЕ: ДЕСЯТИЧНОЕ ЧИСЛО → VCD - ФОРМАТ**ПРИМЕР ▾**

67 (десятичное) <=> Цифры: 6, 7

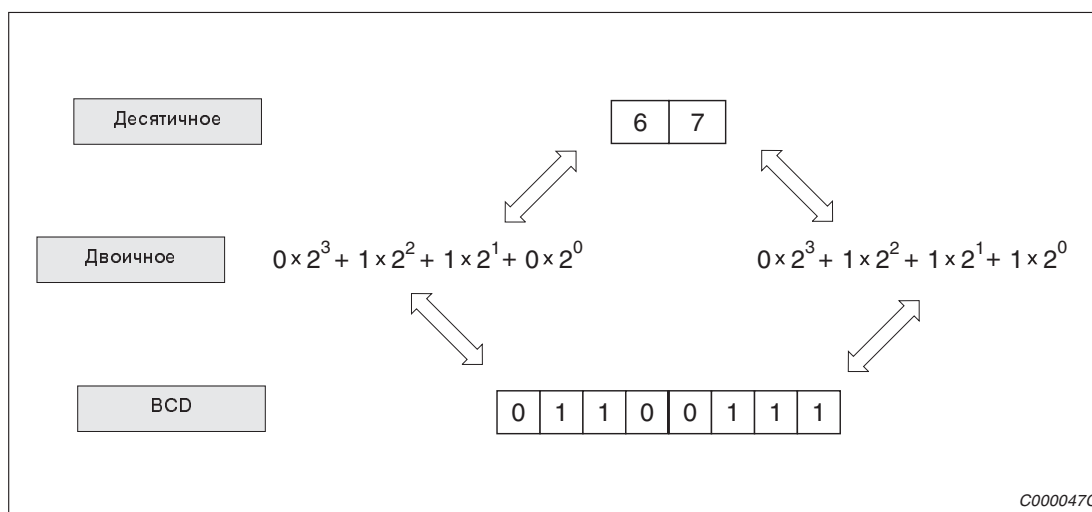


Рис.3-28. Кодирование десятичного числа в VCD-формат и наоборот

3.9 УКАЗАТЕЛЬ

Указатель программируется вместе с инструкциями SJ-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 ПК серии FX1S имеются указатели-маркировки от P0 до P63 (64 адреса). В ПК серии FX1N, FX2N и FX2NC область указателей охватывает 128 адресов (от P0 до P127). В соединении с инструкцией SJ можно перейти в конец программы, если применить адрес указателя P63. Инструкция END не нуждается в этом случае ни в каких указателях - маркировке.

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.

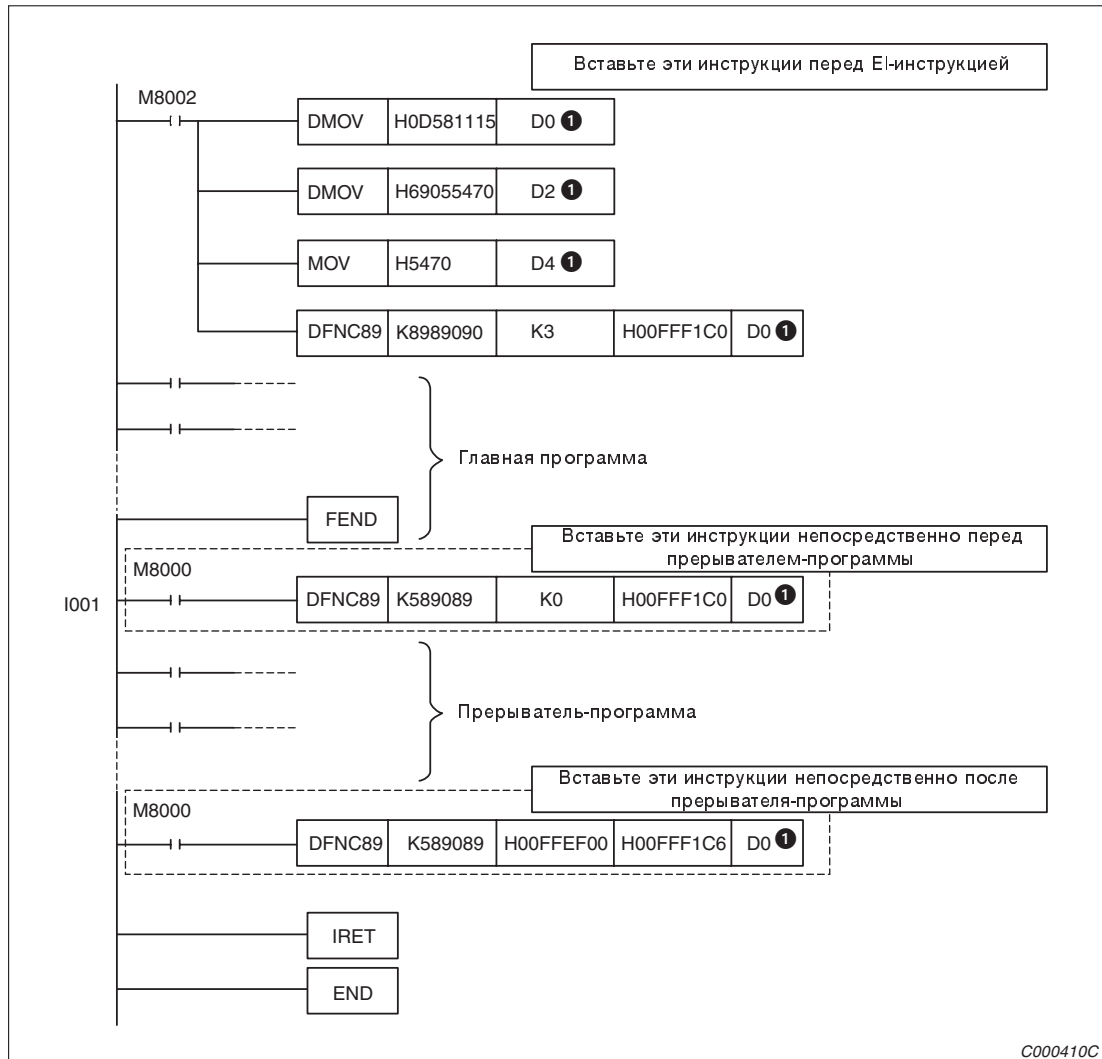


Рис.3-29. Необходимое расширение программы для FX1S ((V1.00 до V1.30) при применении прерывателя-указателя

- ① 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.

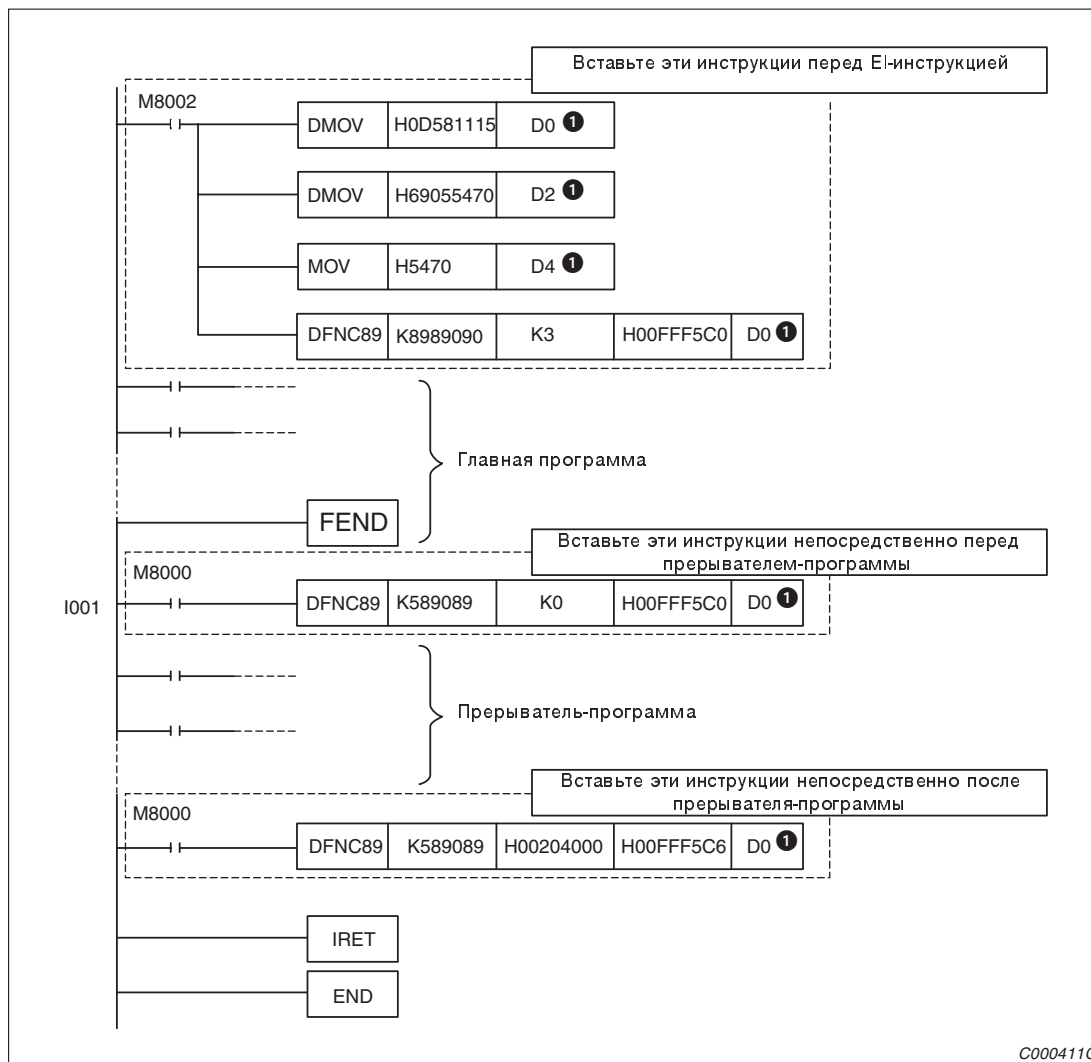


Рис.3-30. Необходимое расширение программы для FX1N (V1.00) при применении прерывателя-указателя

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**MELSEC FX2N и FX2NC:**

Имеется 9 прерывателей-указателей. Адресация прерывателей-указателей должна выполняться согласно следующим указаниям, при этом различаются три группы:

1 Прерыватель-входов: | 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.

**2** Прерыватель-времени: | 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.

3 Прерыватель-счетчик: | 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 включения прерывателя-счетчика включается специальный меркер M8059.

ПРИМЕР ▾

Прерыватель-указатель: I030

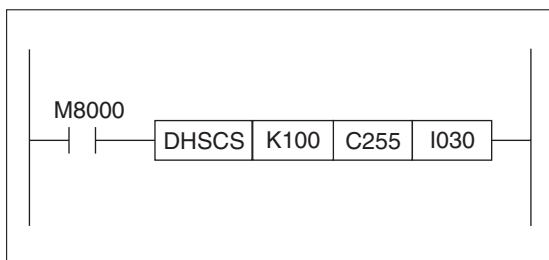


Рис.3-31.
Пример программирования при использовании прерывателя-счетчика

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 I0ab.

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				
		CPU ④	FX1S	FX1N	FX2N	FX2NC
		●	●	●	●	
⑤	⑥	Шаги программы		Примечание		
Операнды						
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(32 Bit): 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) Начало 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)


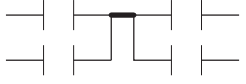

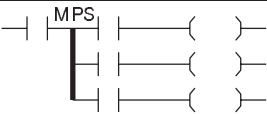
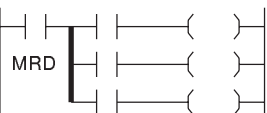
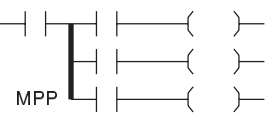
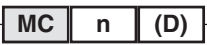



ИНСТРУКЦИЯ	СИМВОЛЫ КОНТАКТНОЙ СХЕМЫ	ЗНАЧЕНИЕ	ОПЕРАНДЫ	ШАГОВ ПРОГРАММЫ	ССЫЛКА НА РАЗДЕЛ
ORF		ИЛИ; (импульсно) ИЛИ - логическая связь с опросом по падающему (заднему) фронту сигнала	X, Y, M, S, T, G	2	Разд. 4.8
ANB		И-БЛОК; Команда связи: последовательное включение параллельных связей	-	1	Разд. 4.9
ORB		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 одnorазового импульса при возрастающем фронте	Y, M	2	Разд. 4.14
PLF		ГЕНЕРАЦИЯ ИМПУЛЬСОВ; Формирование одnorазового 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; SPS-Programmende	-	1	Abs. 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
	●	●	●	●
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

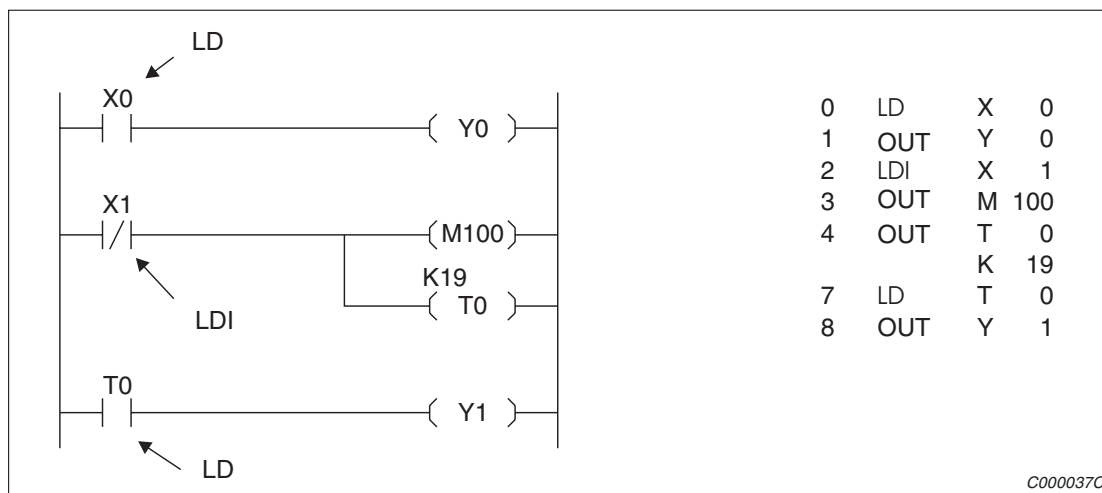


Рис.4-1. Пример программирования с использованием инструкций LD, LDI

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
		●	●	●	●
Операнды		Шаги программы			Примечание
Y, M, S, T, C		Y, M	1	T, C (16 Bit)	3
		S	2	C (32 Bit)	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

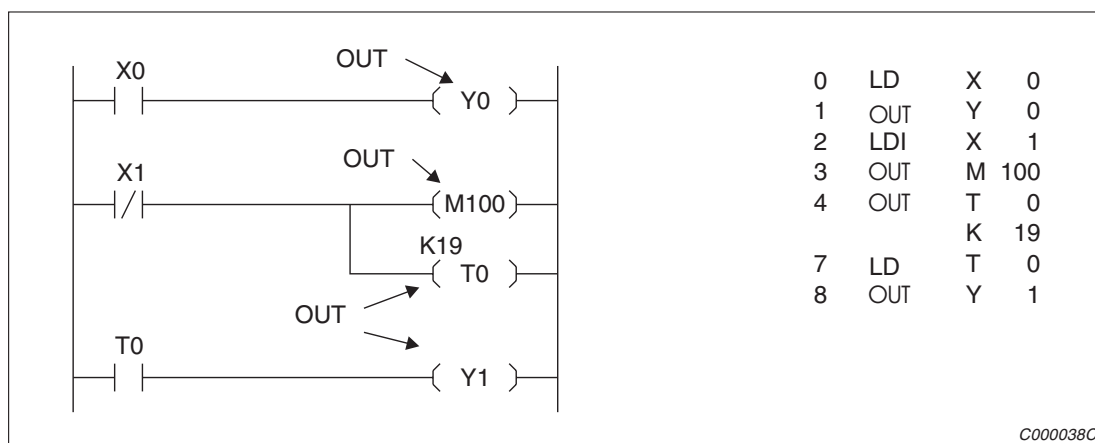


Рис.4-2. Пример программирования с применением инструкции OUT

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 K.

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

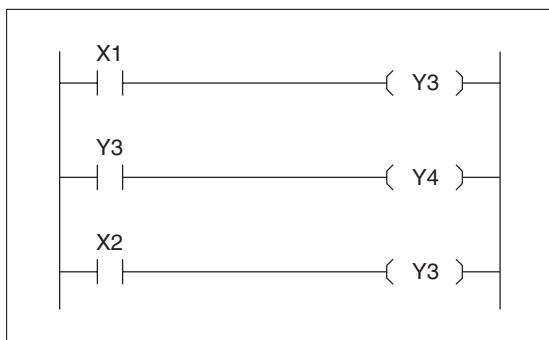


Рис.4-3.
Пример программирования двойной записи выхода

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 ▾

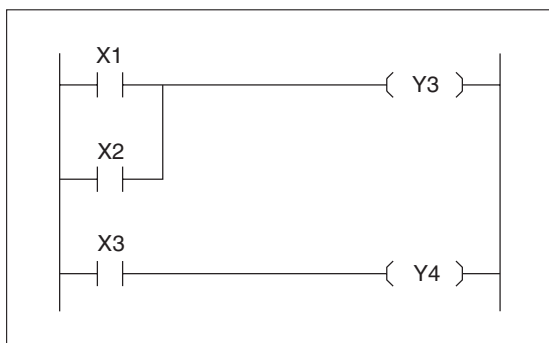



Рис.4-4.
Пример программирования

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
	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC
	●	●	●	●
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 $10 \times 24 = 240$ контактов (аргументов функции).

ПРИМЕР ▾

Применение инструкций AND, ANI

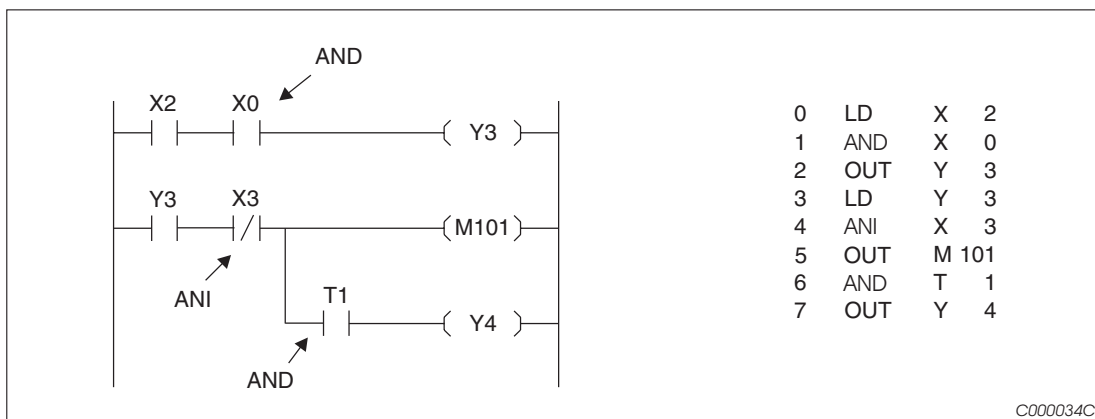


Рис.4-5. Пример программирования с использованием инструкций AND и ANI

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:

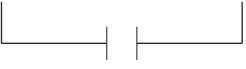

- Выход Y3 приобретет состояние сигнала "1"
- И

- Вход X3 приобретет состояние сигнала "0"
- И

- Контакт таймера T1 приобретет состояние сигнала "0"



4.5 ИЛИ-ЛОГИЧЕСКИЕ СВЯЗИ (OR, ORI)

		OR			
		ИЛИ; ИЛИ - логическая связь с опросом на состояние сигнала "1"			
CPU	FX1S	FX1N	FX2N	FX2NC	
	●	●	●	●	
		ORI			
		ИЛИ НЕ; ИЛИ - логическая связь с опросом на состояние сигнала "0"			
CPU	FX1S	FX1N	FX2N	FX2NC	
	●	●	●	●	
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

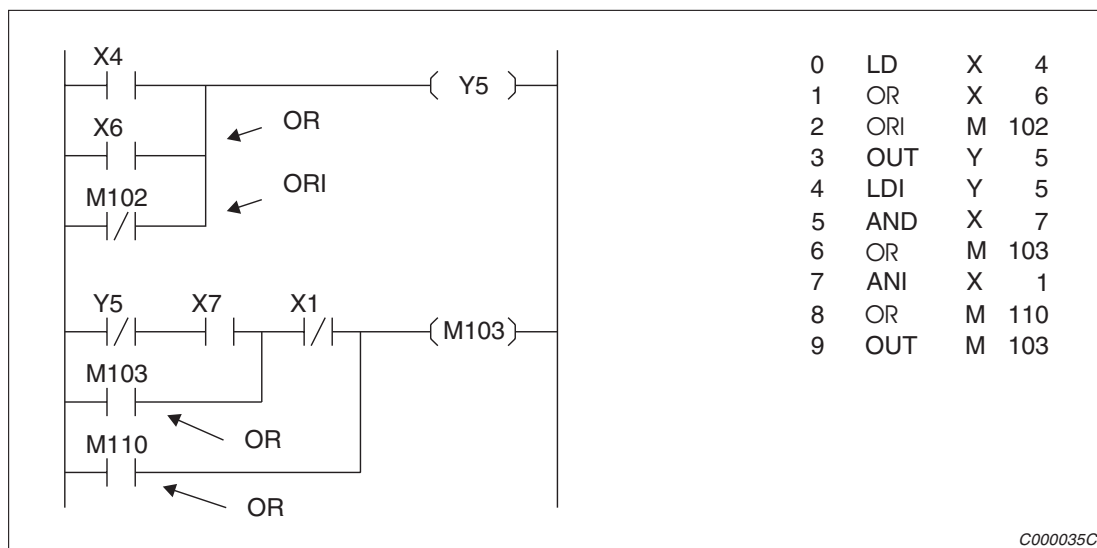


Рис.4-6. Пример программирования с использованием инструкций OR и ORI

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
		●	●	●	●
	CPU	FX1S	FX1N	FX2N	FX2NC
		●	●	●	●
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 заветвлений (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

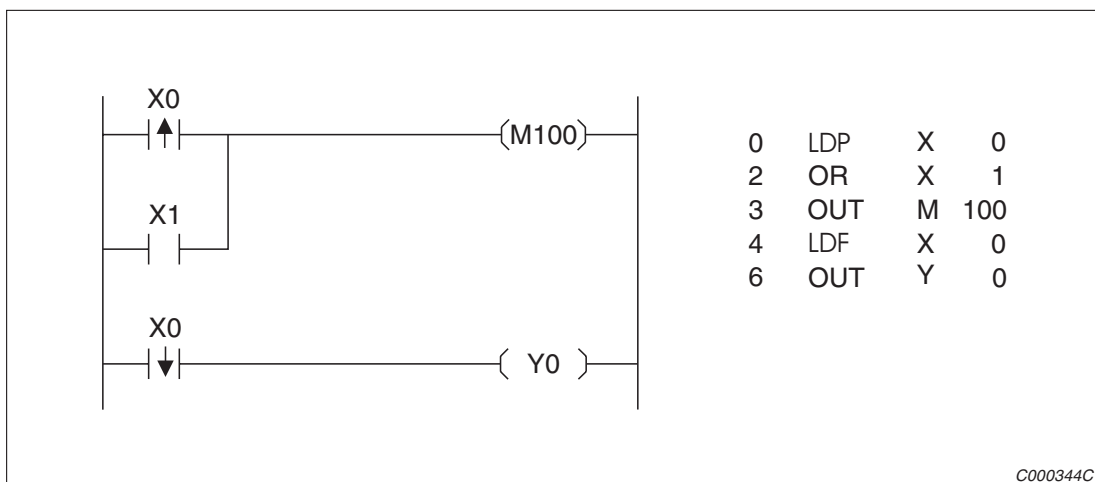




Рис.4-7. Пример программирования с использованием инструкций LDP и LDF

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
		●	●	●	●
	CPU	FX1S	FX1N	FX2N	FX2NC
		●	●	●	●
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

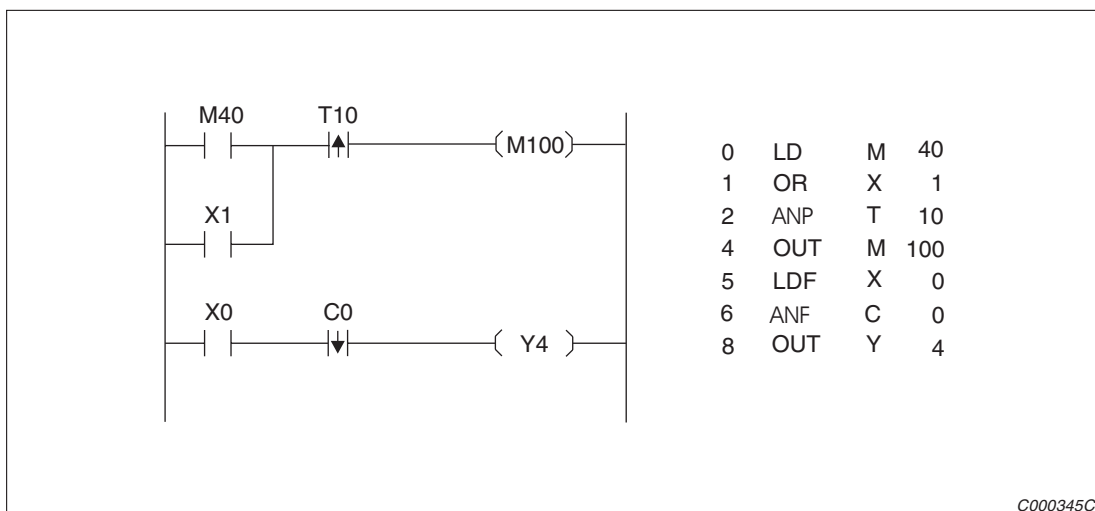




Рис.4-8. Пример программирования с использованием инструкций ANP, ANF

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			
CPU	FX1S	FX1N	FX2N	FX2NC
	●	●	●	●
	ORF			
	ИЛИ (импульсно); ИЛИ - логическая связь с опросом по падающему (заднему) фронту сигнала			
CPU	FX1S	FX1N	FX2N	FX2NC
	●	●	●	●
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

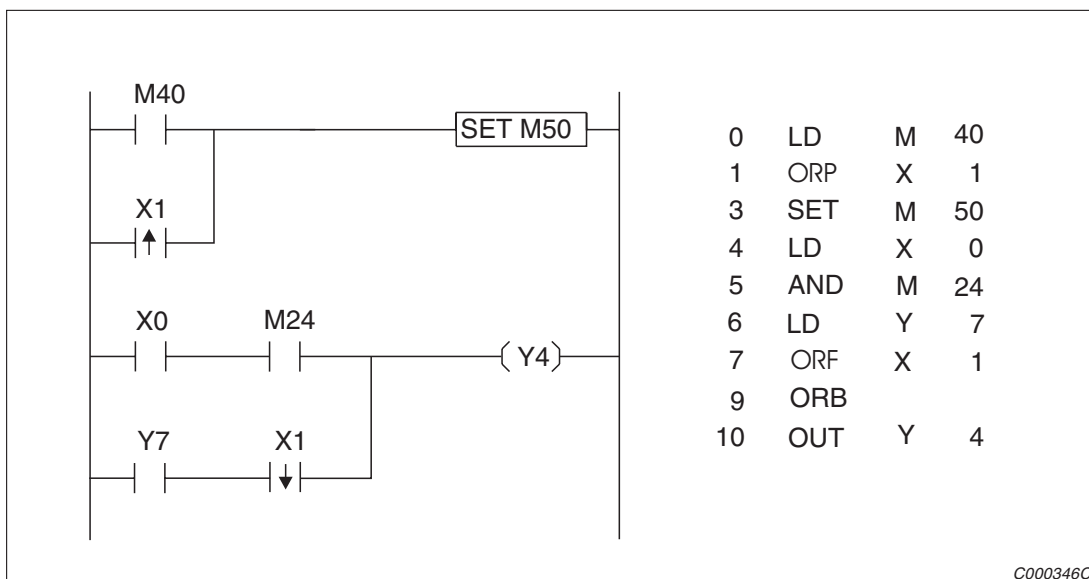


Рис.4-9. Пример программирования с использованием инструкций ORP, ORF

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 иди включенном выходе Y7 и отрицательном (падающем) фронте X1.

△

4.9 “И”-БЛОКИ-ЛОГИЧЕСКИЕ СВЯЗИ (ANB)

	ANB				
	И-блок; Команда связи: последовательное включение параллельных связей				
CPU	FX1S	FX1N	FX2N	FX2NC	
	●	●	●	●	
Операнды		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.

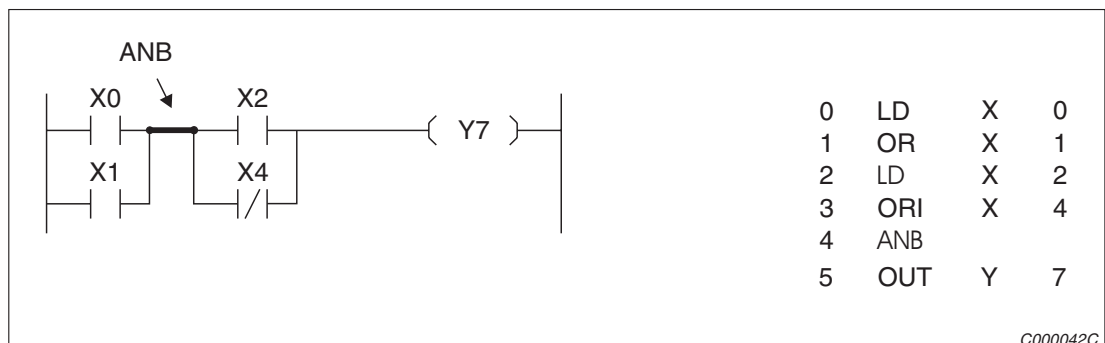


Рис.4-10. Пример программирования с использованием инструкции ANB

4.10 “ИЛИ”-БЛОКИ-ЛОГИЧЕСКИХ СВЯЗЕЙ (ORB)

	ORB			
	ИЛИ-блок; Команда связи: параллельное включение последовательных связей			
CPU	FX1S	FX1N	FX2N	FX2NC
	●	●	●	●
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
	<pre> 0 LD X 0 1 AND X 1 2 LD X 2 3 AND X 3 4 ORB 5 LDI X 4 6 AND X 5 7 ORB 8 OUT Y 6 </pre>	<pre> 0 LD X 0 1 AND X 1 2 LD X 2 3 AND X 3 4 LDI X 4 5 AND X 5 6 ORB 7 ORB 8 OUT Y 6 </pre>

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
		●	●	●	●
	CPU	FX1S	FX1N	FX2N	FX2NC
		●	●	●	●
	CPU	FX1S	FX1N	FX2N	FX2NC
		●	●	●	●
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

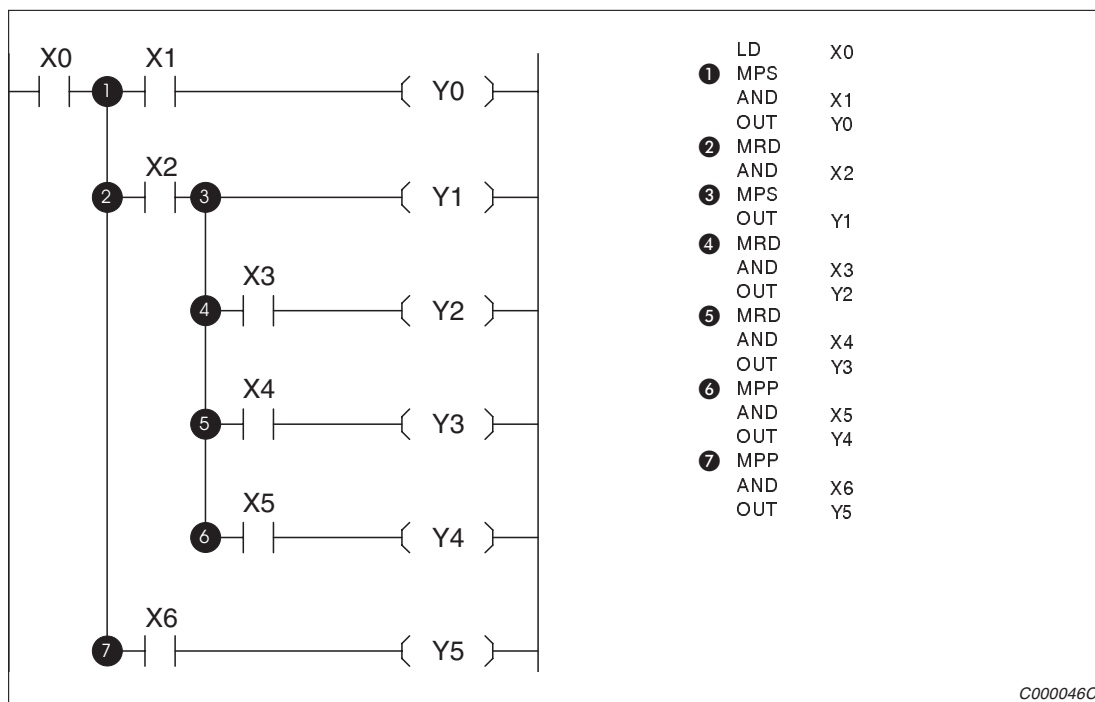


Рис.4-12. Пример программирования инструкций MPS, MRD, MPP

- 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.



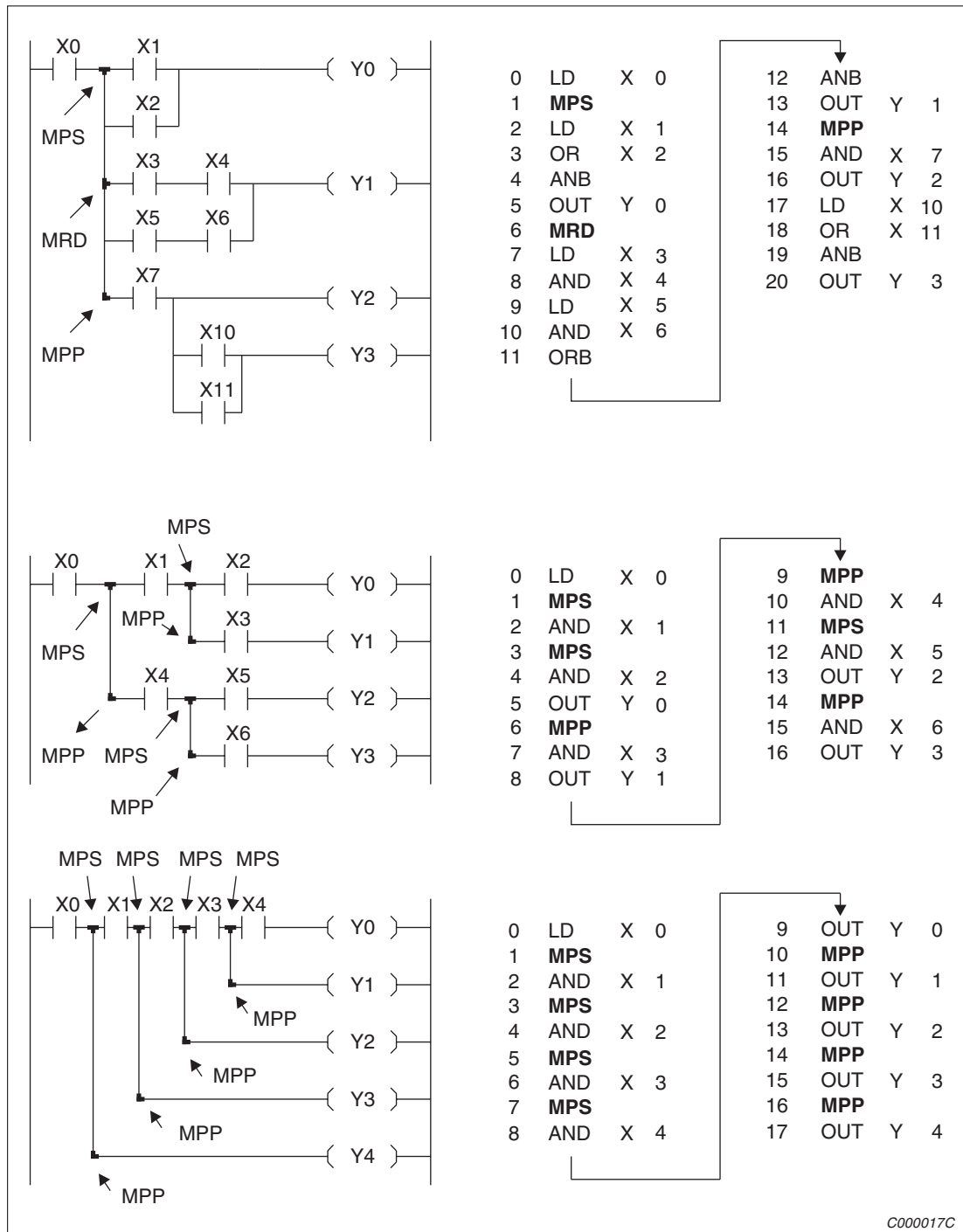


Рис.4-13. Примеры программирования инструкций MPS, MRD и MPP

C000017C

4.12 ВКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ УСЛОВИЙ КОНТРОЛЯ (MC, MCR)

	MC			
	Верхний уровень управления; Включение (запуск) условий контроля			
	MCR			
	Верхний уровень управления сбросат; Отключение (сброс) условий контроля			
CPU	FX1S	FX1N	FX2N	FX2NC
	●	●	●	●
CPU	FX1S	FX1N	FX2N	FX2NC
	●	●	●	●
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 MC и MCR

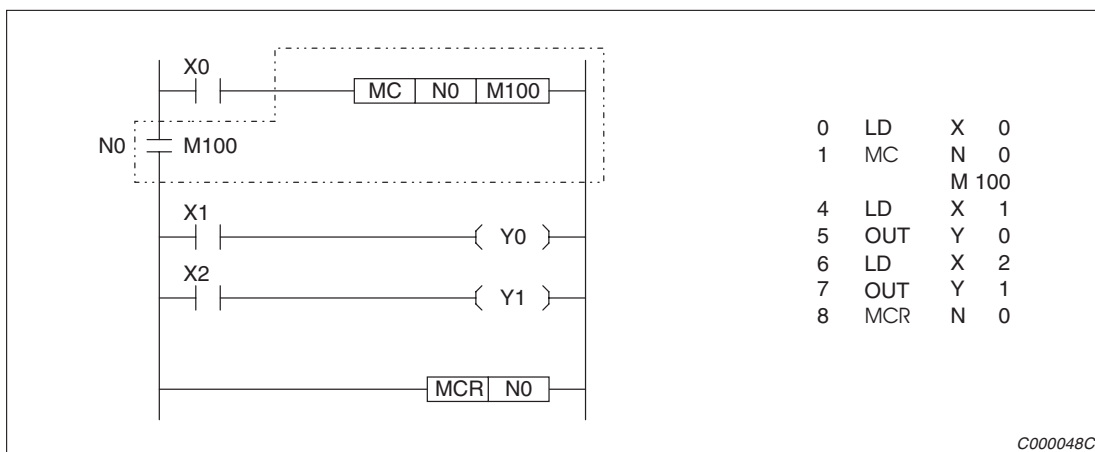


Рис.4-14. Пример программирования инструкций MC и MCR

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 MC инструкции, включается контакт меркера M100 (адрес программы разветвления N0). Все цепи между инструкциями MC и MCR активизируются. Состояние сигнала Y0 или Y1 зависит тогда только от сигналов состояния X1 и X2.



Использование нескольких мс и мсг инструкций внутри одной программы

При программировании нескольких МС и МСR инструкций внутри программы нужно обращать внимание на следующее:

- чтобы 1-АЯ МС-инструкция начиналась с самого низкого адреса N разветвления программы
- чтобы 1-АЯ МСR-инструкция начиналась с самого высокого адреса N разветвления программы.

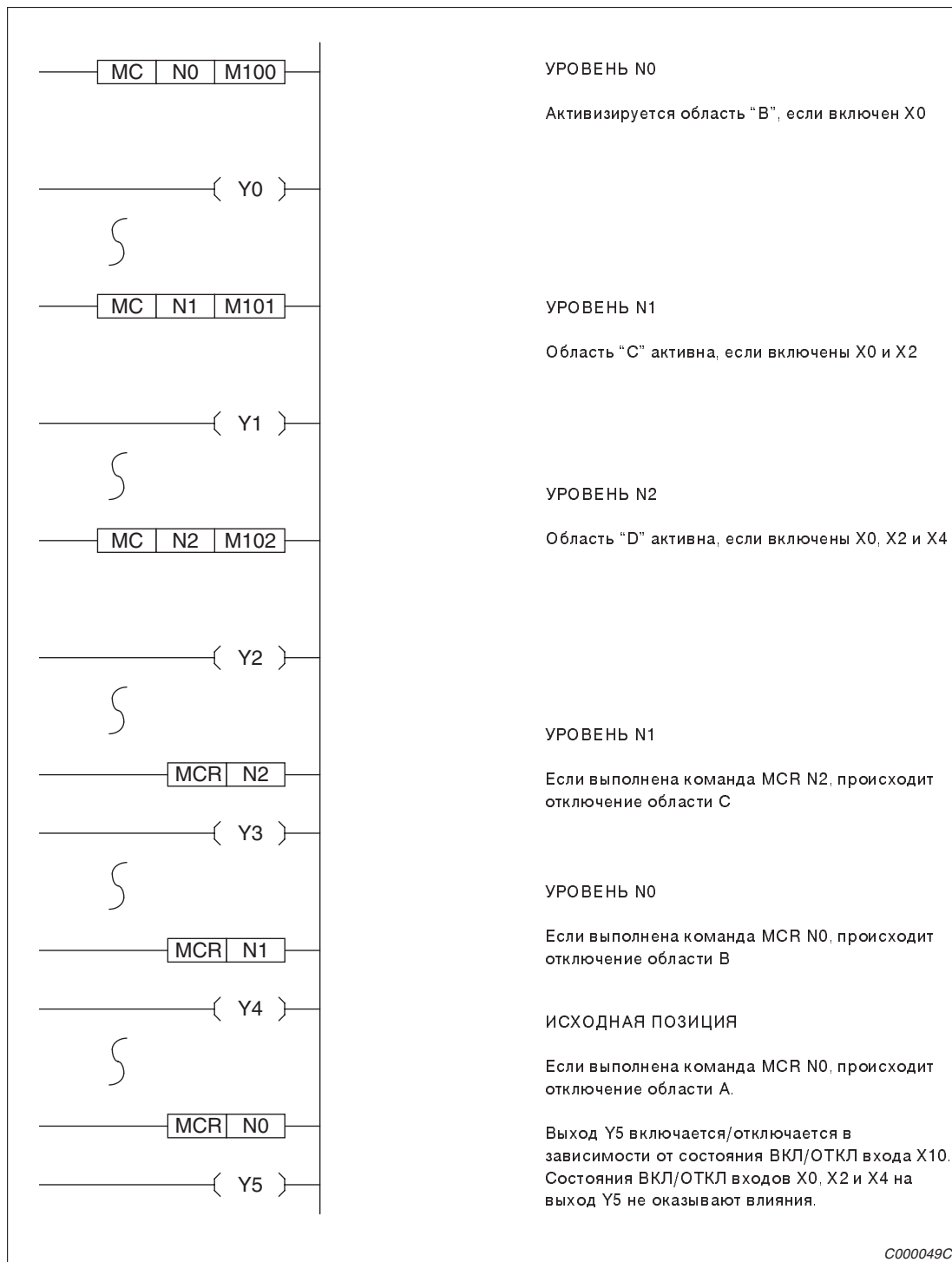
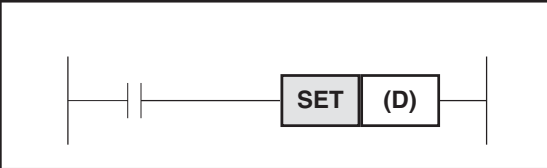


Рис.4-15. Пример программирования нескольких МС и МСR инструкций внутри программы

4.13 ВКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ ОПЕРАНДОВ (SET, RST)

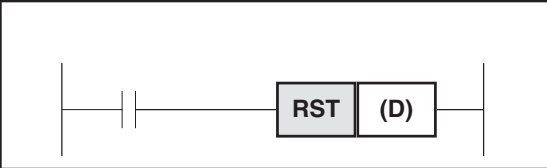
		SET				
		Включение; Включение операнда				
		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
Операнды	Шаги программы			Примечание		
Y, M, S	Y, M	1				
	S, специальный меркер	2				

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
			●	●	●	●
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.

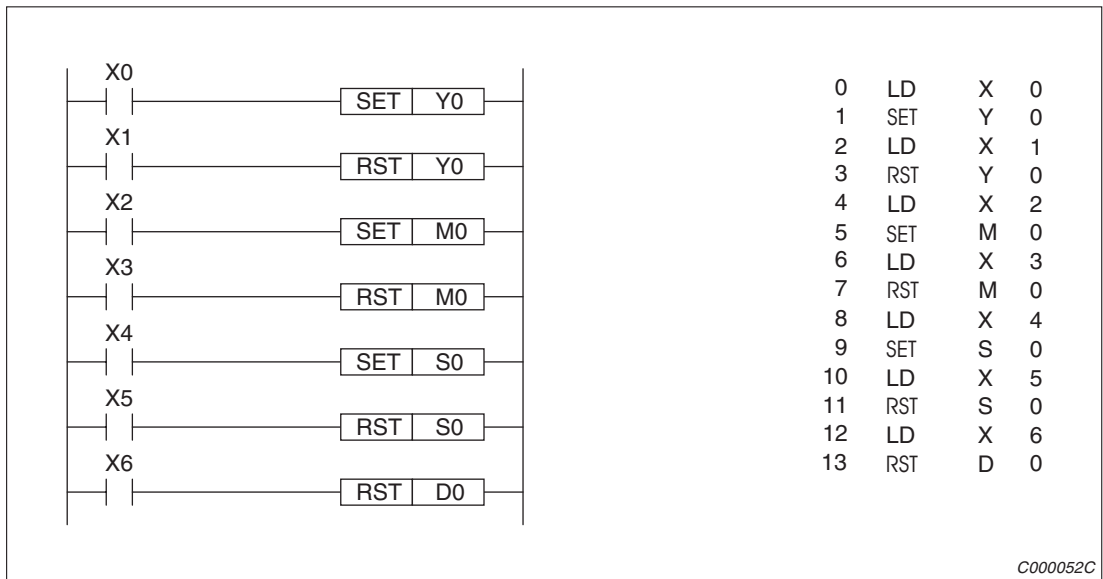


Рис.4-16. Пример программирования инструкций SET и RST

ПРИМЕР ▾

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

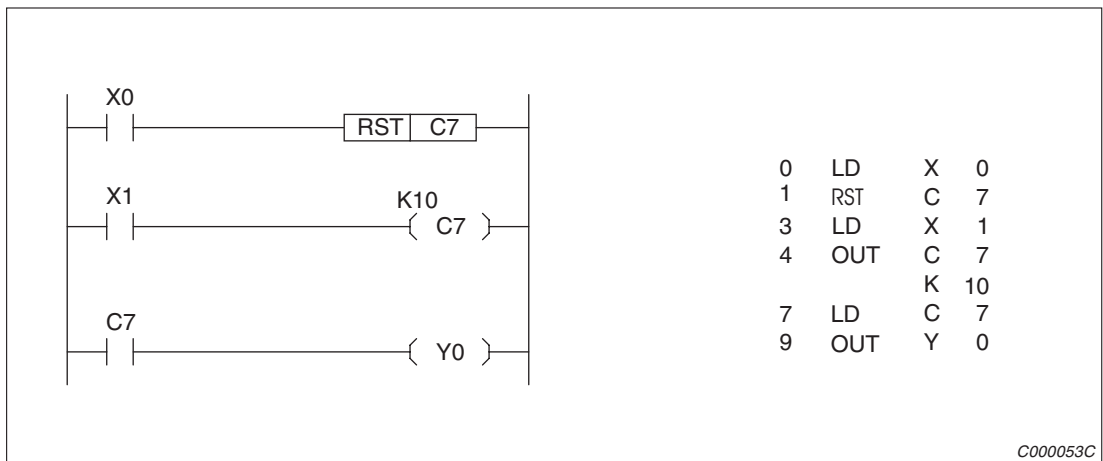
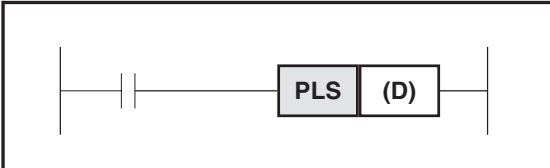
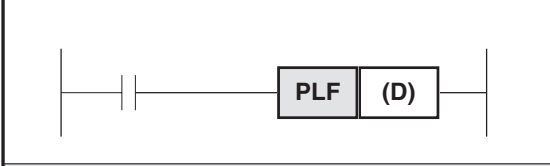


Рис.4-17. Пример программирования отключения 16-ти битного счетчика с помощью RST инструкции

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 СОЗДАНИЕ ОДНОГО 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
	●	●	●	●
Операнды				
Шаги программы		Примечание		
Y, M	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 одноразового импульса PLS, PLF

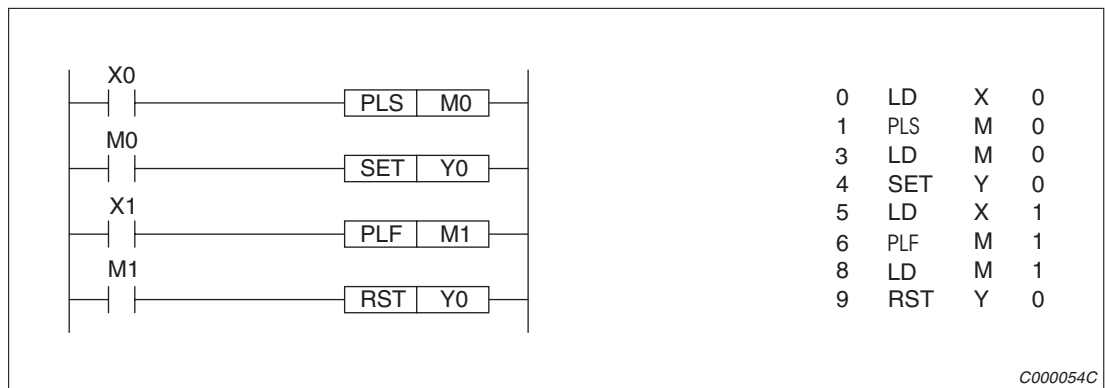


Рис.4-18. Пример программирования инструкций PLS и PLF

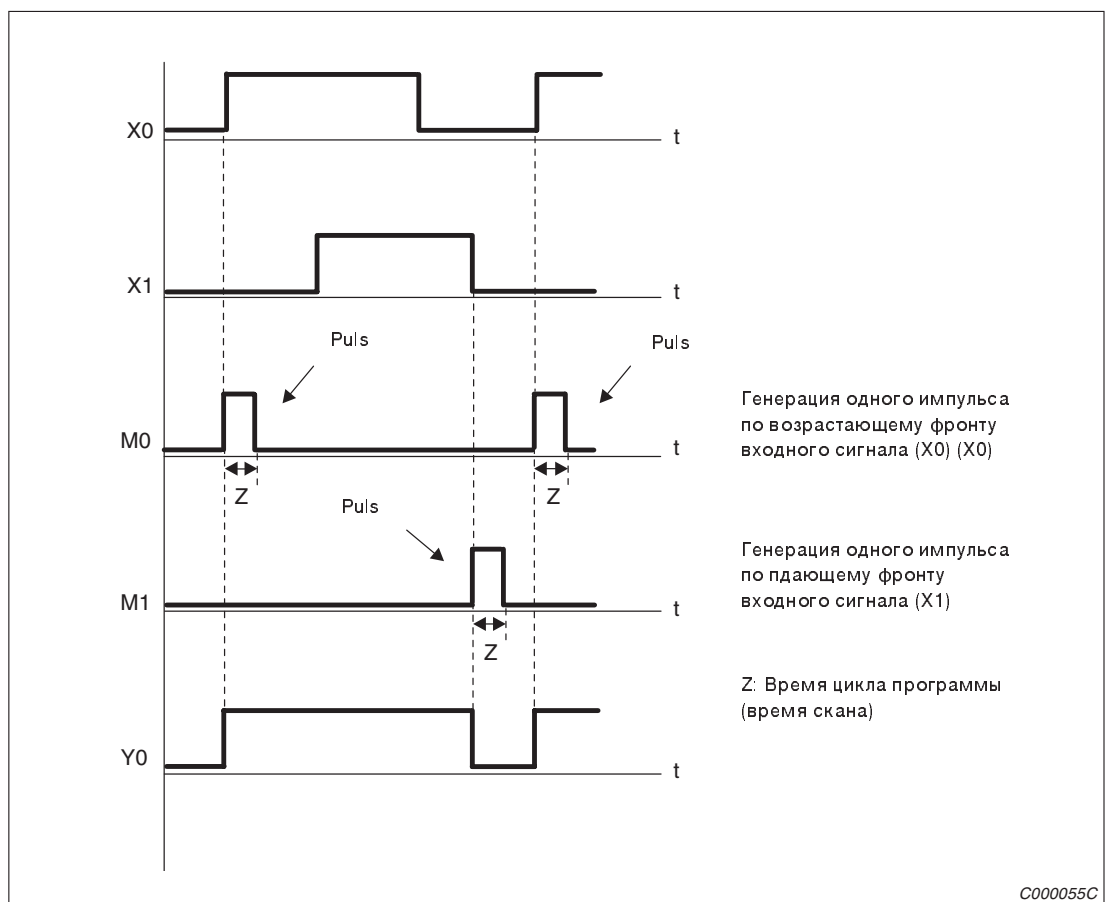


Рис.4-19. Пример программирования, изображение обработки входного сигнала и генерации импульса

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
		●	●	●	●
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

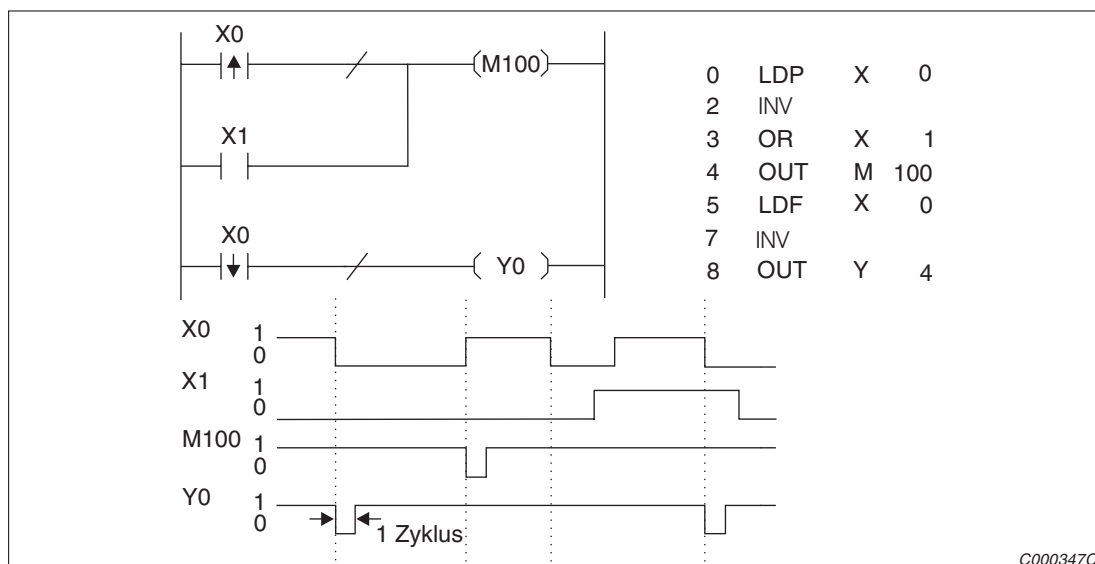


Рис.4-20. Пример программирования инструкции INV

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

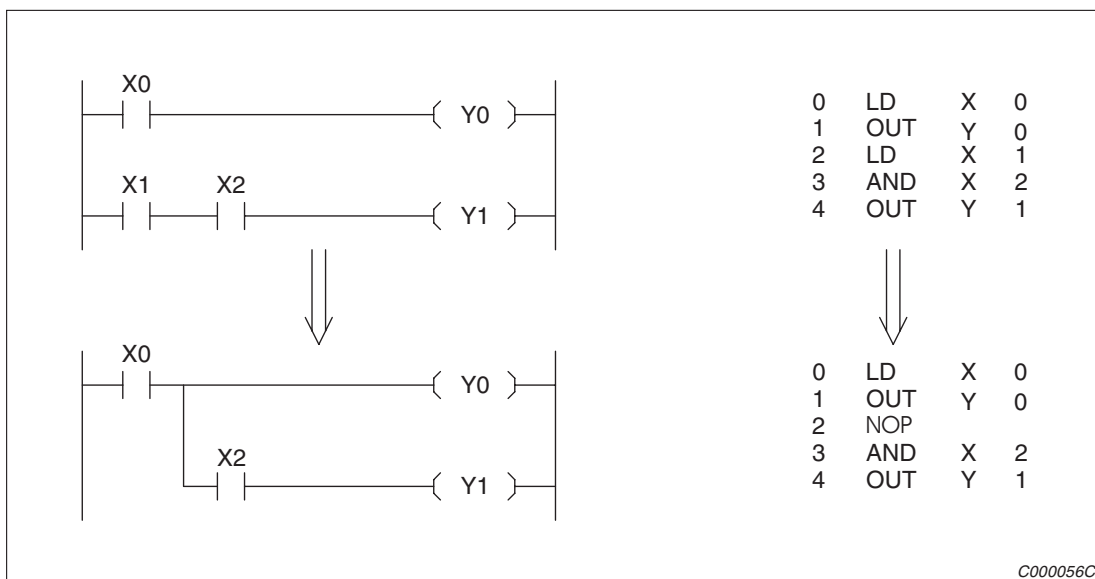



Рис.4-21. Пример программирования инструкции NOP

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	
●	●	●	●	●	
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).

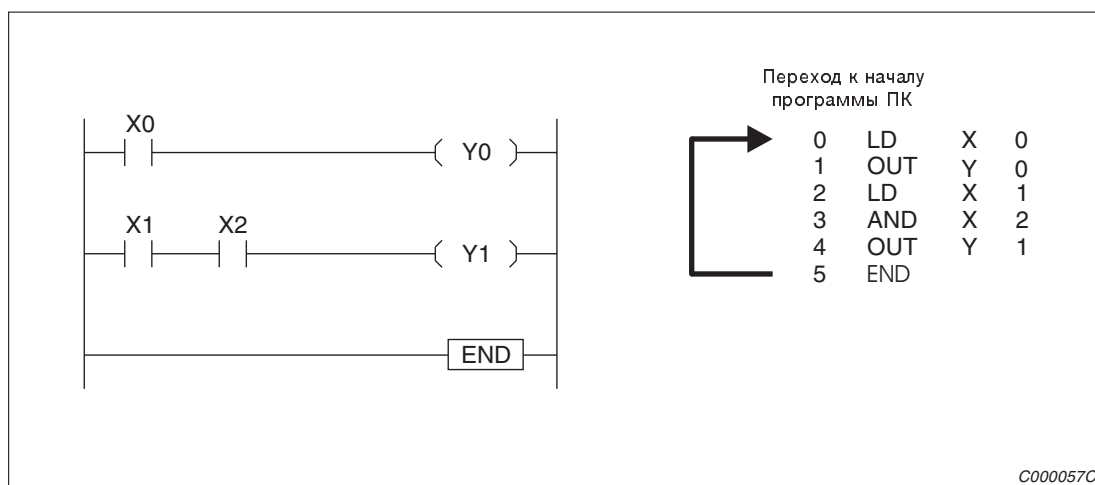


Рис.4-22. Пример программирования END-инструкции

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

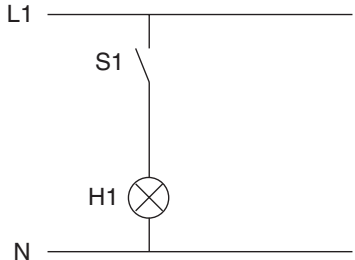
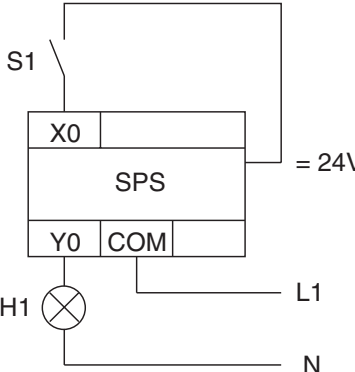
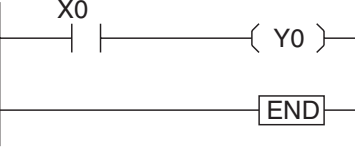
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При замыкании замыкающего контакта S1 должна засветиться лампа H1.</p>	 <p style="text-align: right;"><i>C000007G</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Сигнальная лампа: H1 Y0</p>	 <p style="text-align: right;"><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

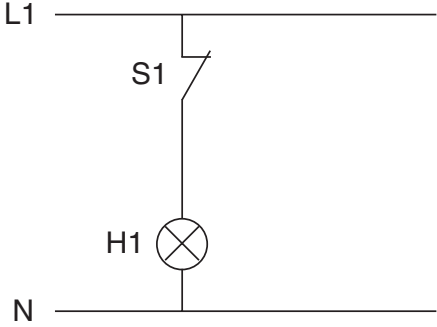
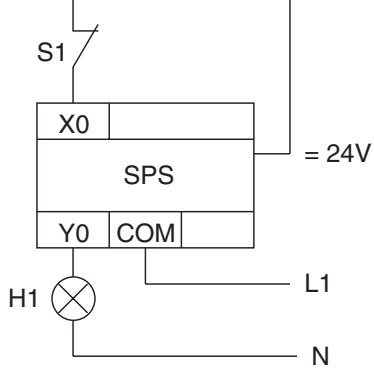
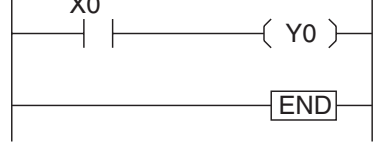
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При задействованном размыкающем контакте S1 должна светиться лампа H1.</p>	 <p style="text-align: right;"><i>C000010G</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Размыкающий контакт: S1 X0</p> <p>Сигнальная лампа: H1 Y0</p>	 <p style="text-align: 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

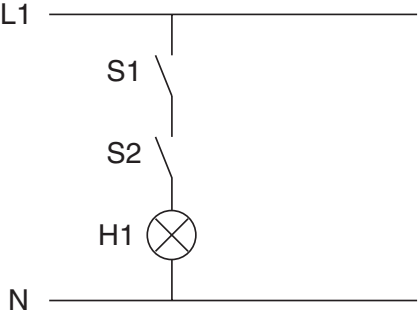
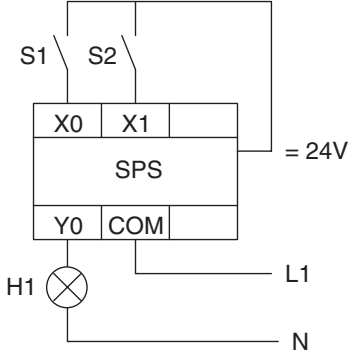
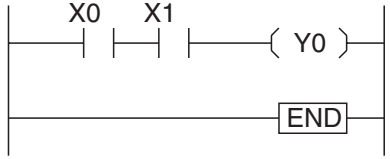
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При задействованных замыкающем контакте S1 "И" замыкающем контакте S2 должна светиться лампа Н1.</p>	 <p style="text-align: right;"><i>C00001</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Замыкающий контакт: S2 X1 Сигнальная 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

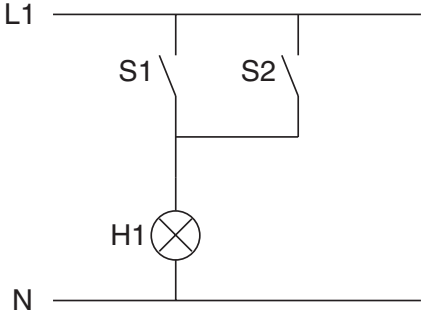
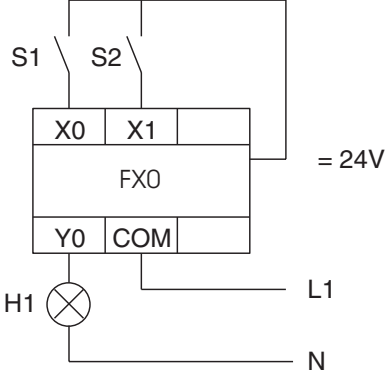
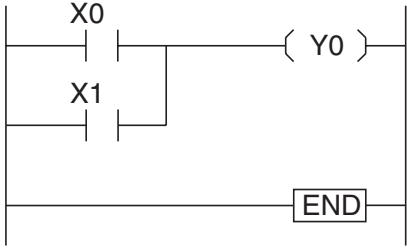
ПРИМЕР	ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРОСХЕМА
<p>При задействованном замыкающем контакте S1 "ИЛИ" замыкающем контакте S2 должна светиться лампа H1.</p>	 <p style="text-align: right;"><i>C000019G</i></p>
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Замыкающий контакт: S2 X1 Сигнальная лампа: H1 Y0</p>	 <p style="text-align: right;"><i>C000020G</i></p>
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (KOP)
<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 X1 есть сигнал "1".</p>	

Табл.4-7. Пример логической связи "ИЛИ"

САМОПИТАНИЕ (I)

Включение выхода с самопитанием

<p>ПРИМЕР</p> <p>При включении замыкающего контакта S1 должна засветиться 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 (активируется замыкающий контакт S1). Выход Y0 отключится (состояние сигнала "0"), если кратковременно задействуется вход X1 (активируется 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 засветиться 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 (активируется замыкающий контакт S1). Выход Y0 отключится (состояние сигнала "0"), если кратковременно задействуется вход X1 (активируется размыкающий контакт S2).</p>	

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

ПРИМЕНЕНИЕ ТАЙМЕРА ДЛЯ ЗАДЕРЖКИ ВКЛЮЧЕНИЯ

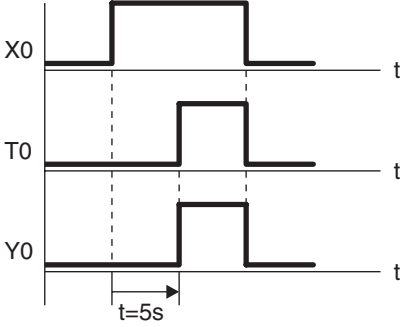
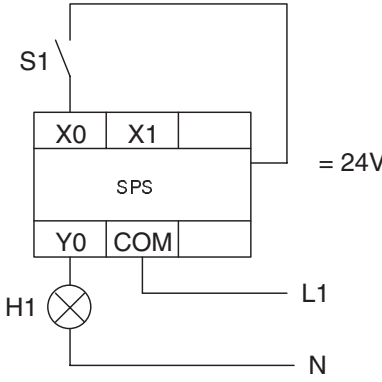
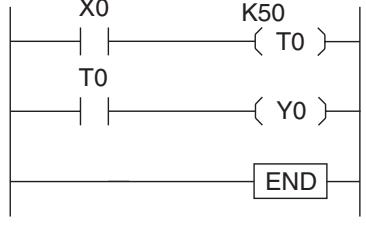
ПРИМЕР	ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК
<p>При задействованном замыкающем контакте S1 лампа H1 не должна загораться до отсчета 5 с.</p>	<p>Замыкающий контакт: S1 X0 Сигнальная лампа: H1 Y0 Таймер : T0 100 мс</p>
ВРЕМЕННАЯ ДИАГРАММА	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
	
ЛИСТИНГ ИНСТРУКЦИЙ (AWL)	КОНТАКТНАЯ СХЕМА (КОР)
<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 действующем замыкающем контакте S1 лампа H1 сейчас же загорается. Однако лампа H1 должна светить на 5 с больше, чем отключится замыкающий контакт S1.</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 (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 Н1. Контактom S2 счетчик C0 должен снова вернуться в исходное состояние (сбросить накопленный счет) и погасить лампу Н1.</p>	-
ЛИСТИНГ ПРИСВОЕНИЯ АДРЕСОВ ПК	СХЕМА ПОДКЛЮЧЕНИЯ К ПК
<p>Замыкающий контакт: S1 X0 Замыкающий контакт: S2 X1 Сигнальная лампа: Н1 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.

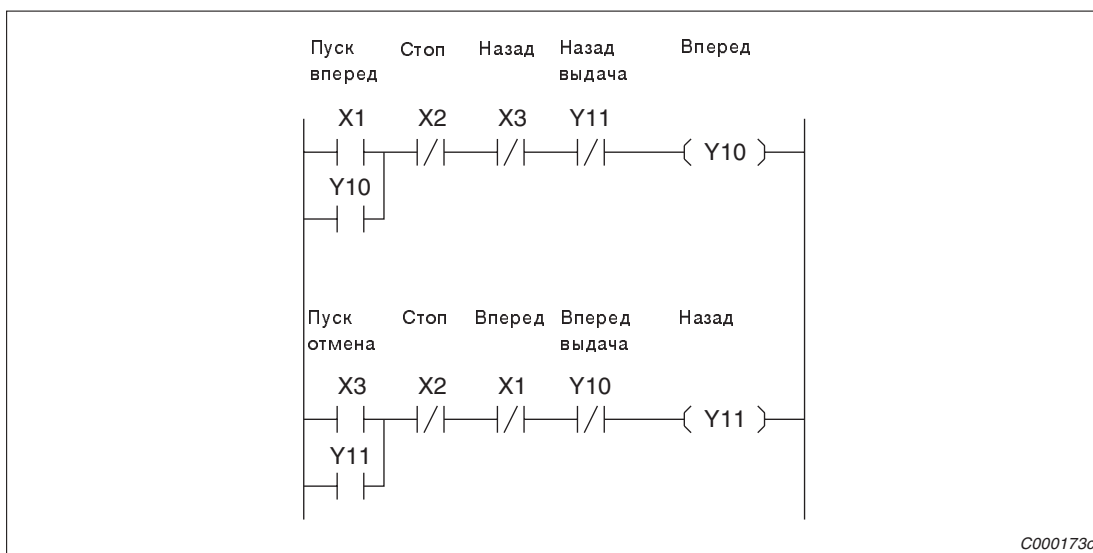


Рис.5-1. Пример применения блокировочных контактов

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.

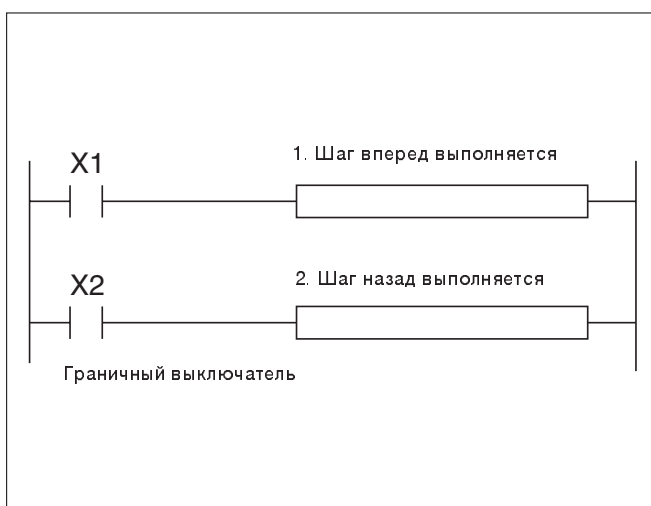


Рис.5-2. Пример схемы без блокировочных контактов

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.

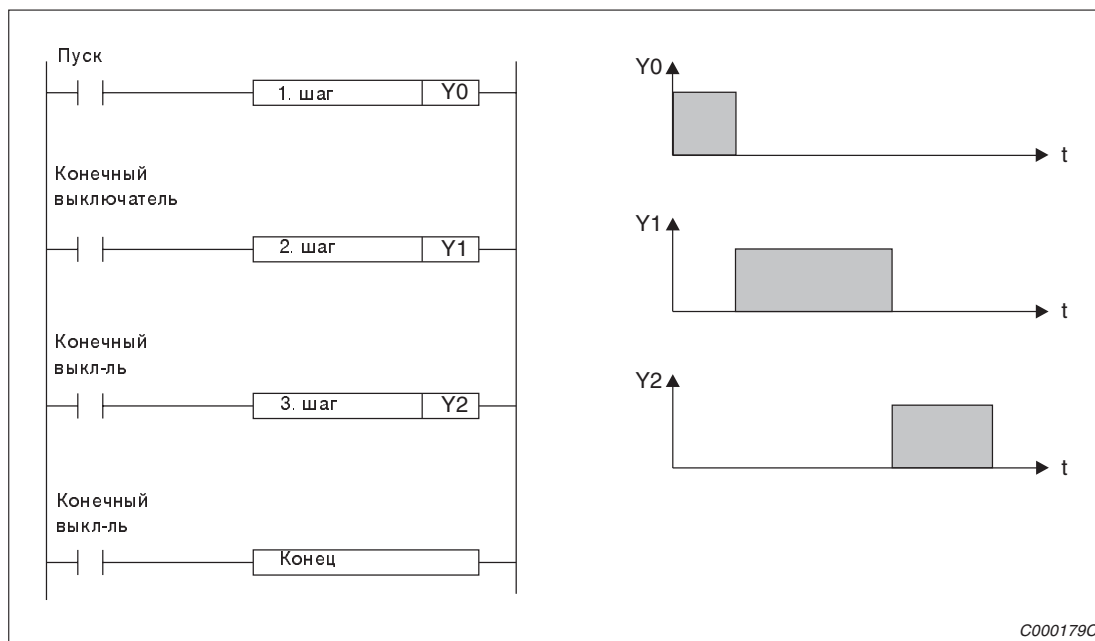


Рис.5-3. Схематичный процесс шагового управления

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.

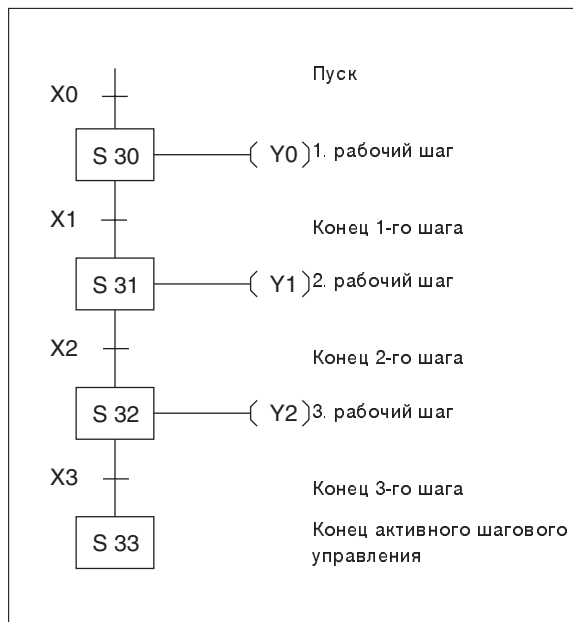


Рис. 5-4.
Диаграмма блоков примера
пользователя

C000148C

5.2 ПРОГРАММИРОВАНИЕ STL-ИНСТРУКЦИИ

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

Тип ПК	Количество операндов	Область адресов
FX1S	128	S0 ... S127
FX1N	1000	S0 ... S999
FX2N		
FX2NC		

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	допустимо	не допустимо
	Условия дальнейшего включения	допустимо	не допустимо	не допустимо

Табл.5-2. Допустимые инструкции внутри состояний шагов

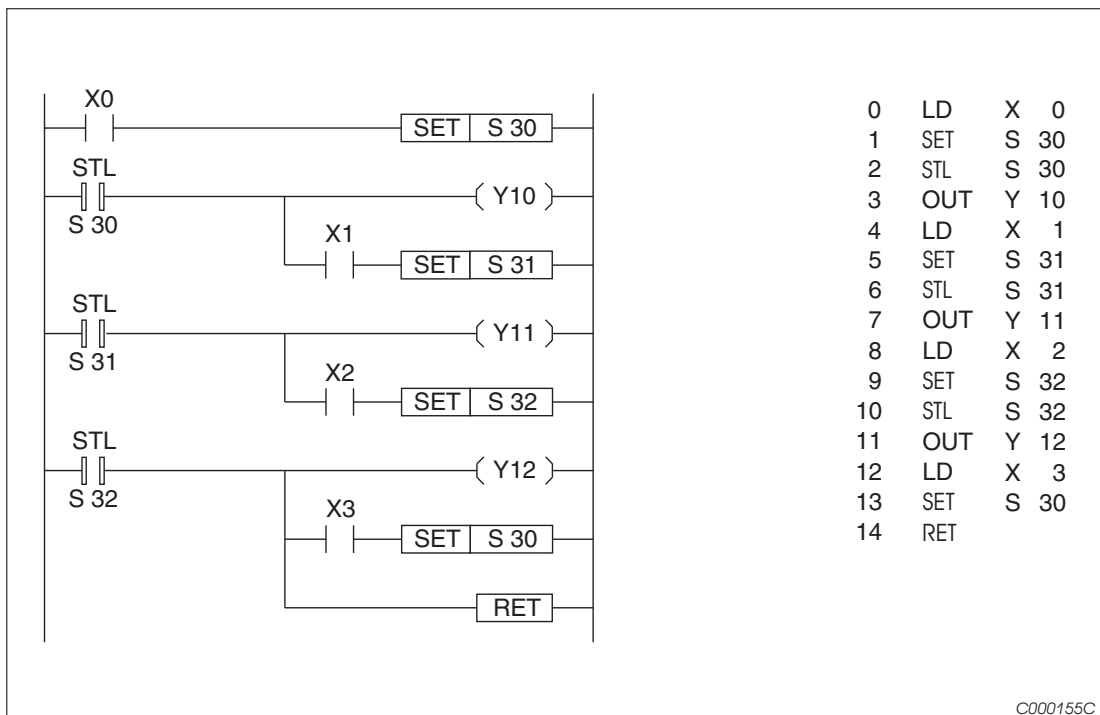


Рис.5-5. Пример программирования STL-, RET-инструкций

ЗАПИСЬ ВЫХОДОВ НЕСКОЛЬКО РАЗ

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

ПРИМЕР ▾

Запись выходов несколько раз

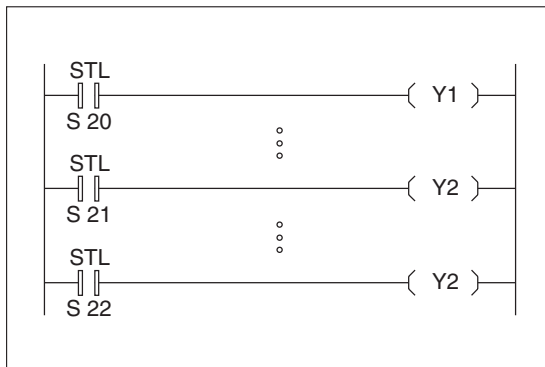


Рис.5-6.

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.

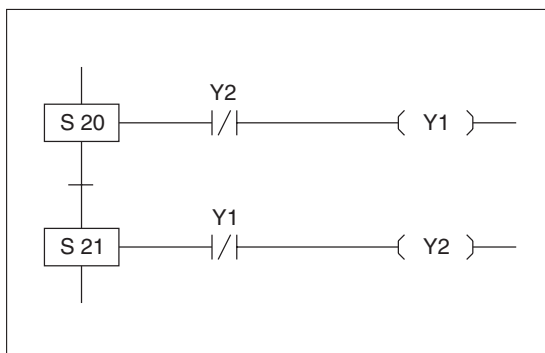


Рис.5-7.

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.

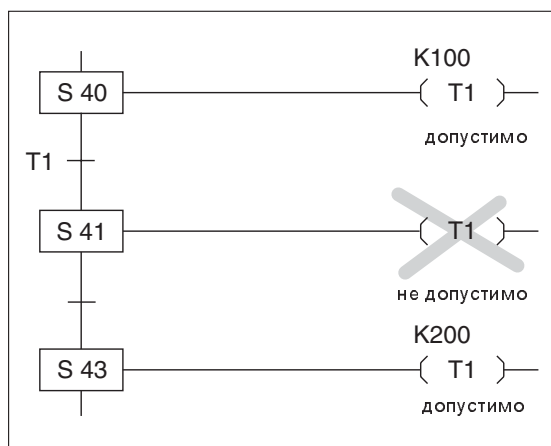


Рис. 5-8. Многократная запись таймеров

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

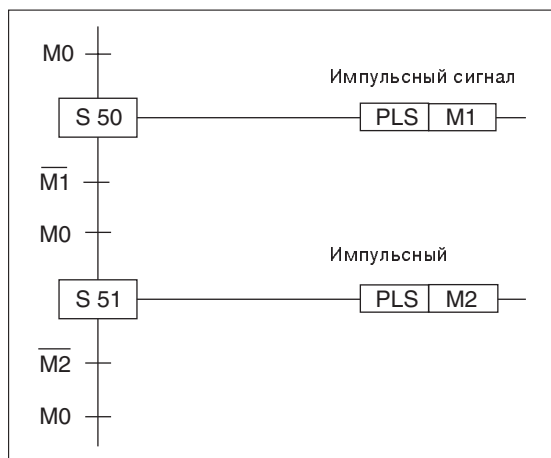


Рис. 5-9. Условие последующего включения по импульсному сигналу

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. △

УСЛОВИЯ ПОСЛЕДУЮЩИХ ВКЛЮЧЕНИЙ ПОСРЕДСТВОМ ИМПУЛЬСНОЙ ИНСТРУКЦИИ (FX2N и FX2NC)

При применении ПК серии FX2N или FX2NC условия последующих включений могут реализовываться посредством импульсных инструкций (LDP, LDF, ANP и т.д.) и импульсных меркеров M2800...M3071.

ПРИМЕР ▾

Условие последующих включений при применении импульсного меркера M2800:

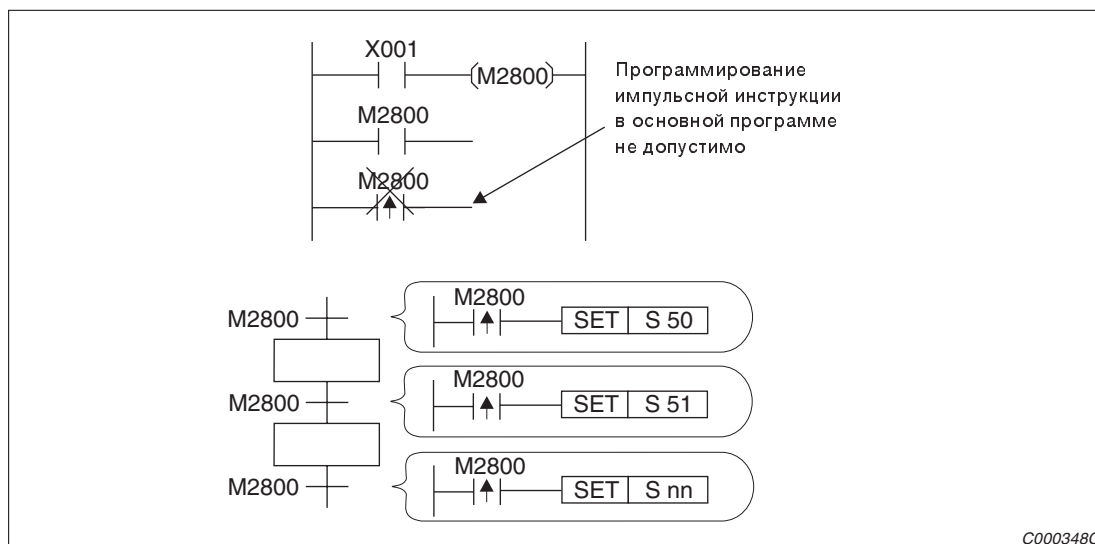


Рис.5-10. Условие последующего включения по импульсному меркеру M2800

Посредством включения меркера 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 меркер 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

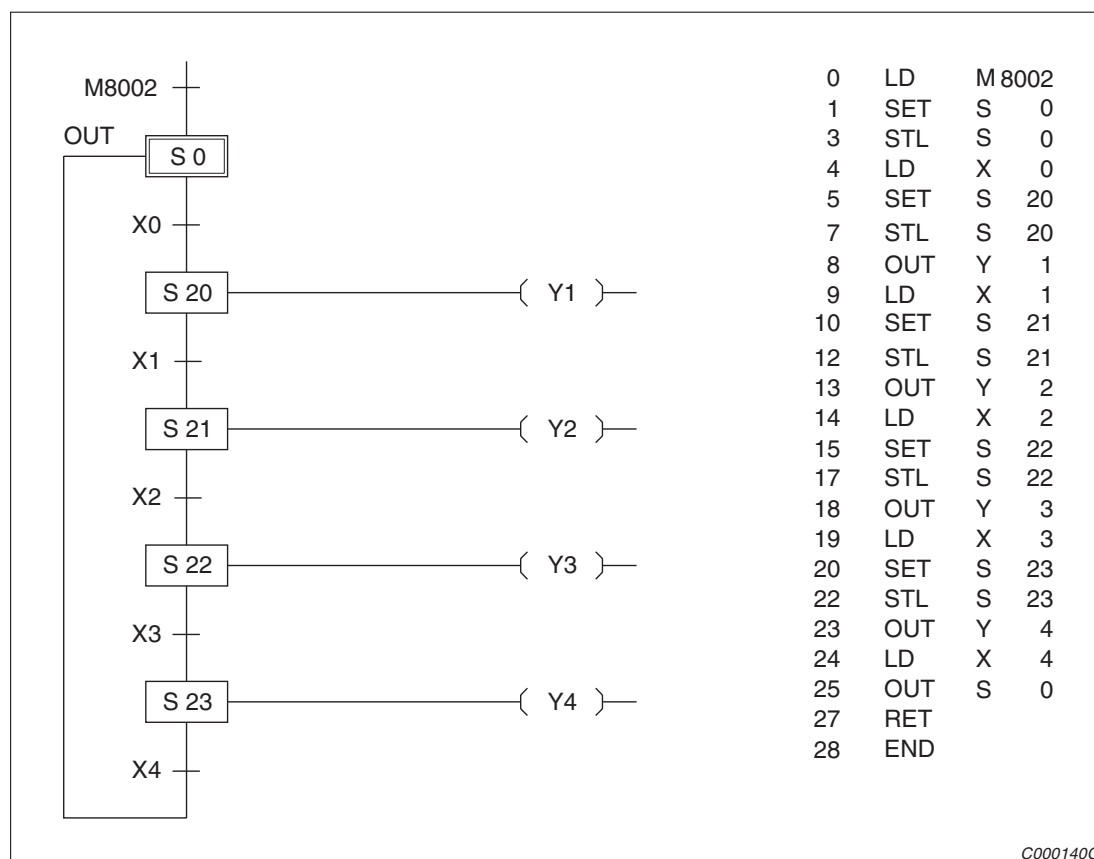


Рис.5-11. Пример программирования инициализации шаговых состояний

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.

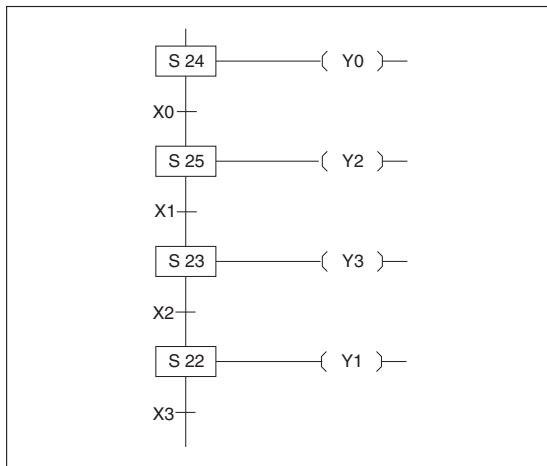
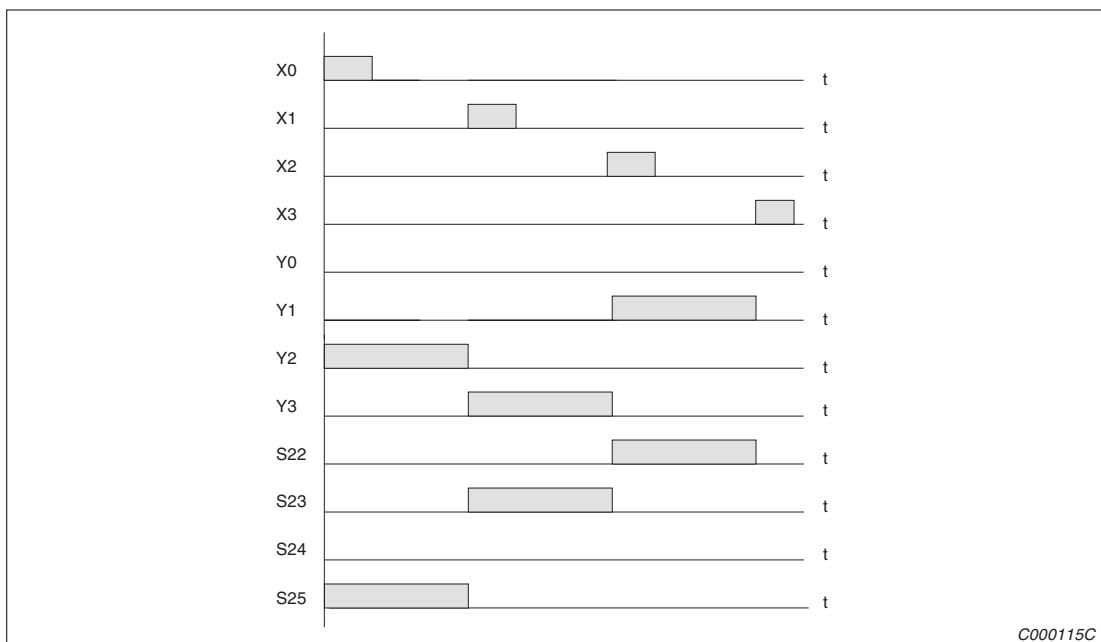


Рис. 5.12.
Пример простого процесса

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.

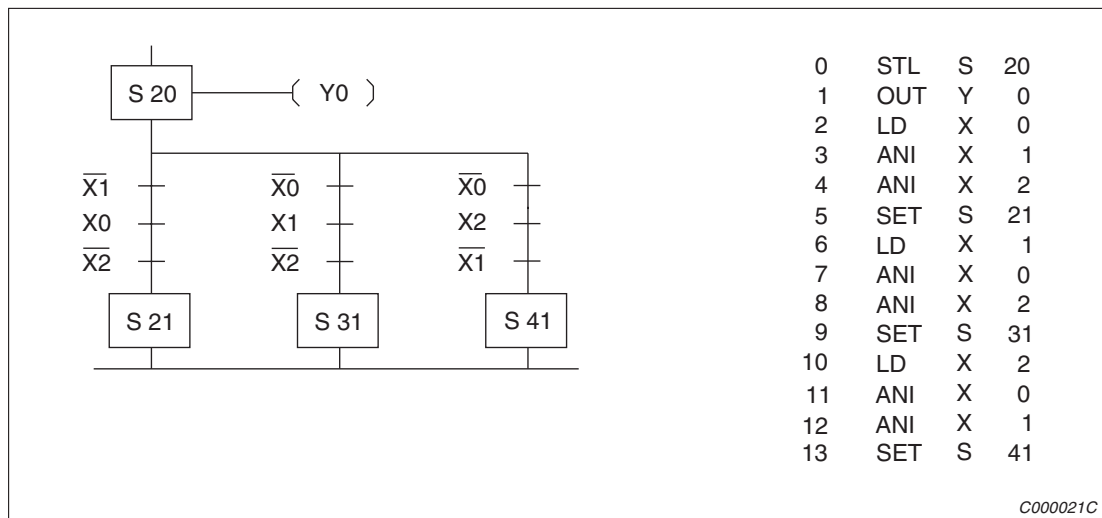


Рис.5-14. Пуск (начало) селективного разветвления

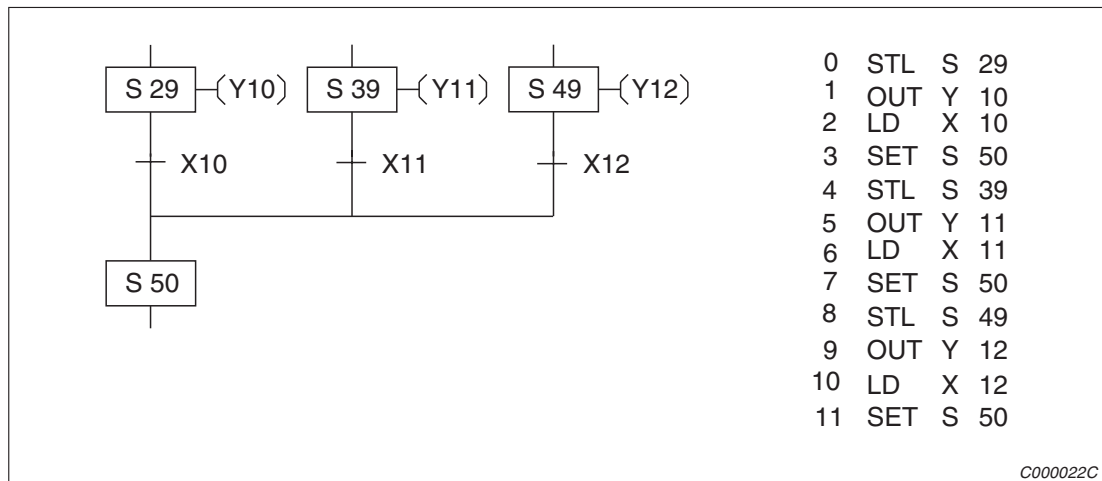


Рис.5-15. Сборка (окончание) селективного разветвления

ПРИМЕР ▾

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

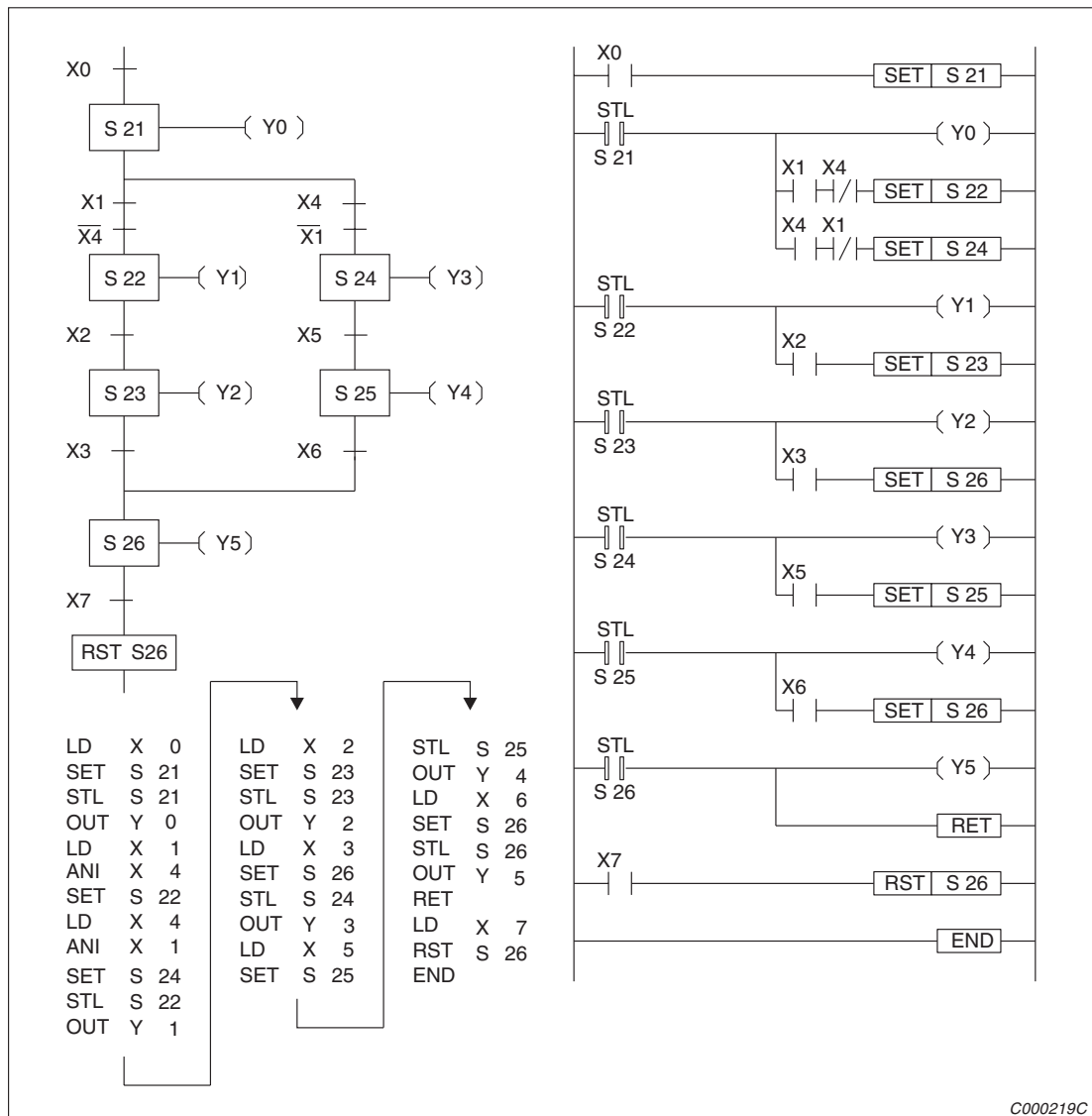


Рис.5-16. Пример программирования селективного разветвления

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.

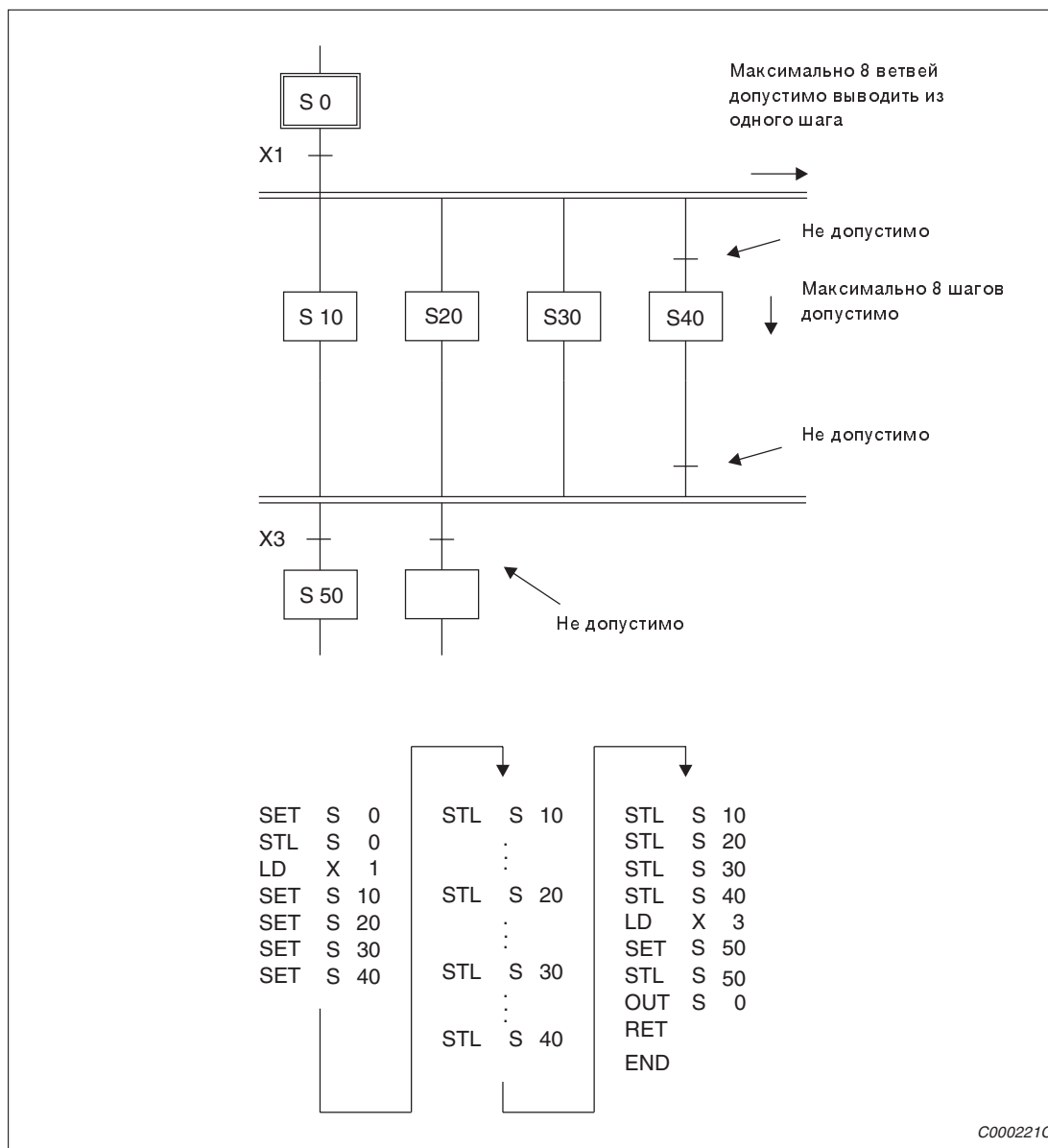


Рис.5-17. Допустимые параллельные разветвления

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 (КОР) и листинг инструкций (AWL) параллельного разветвления

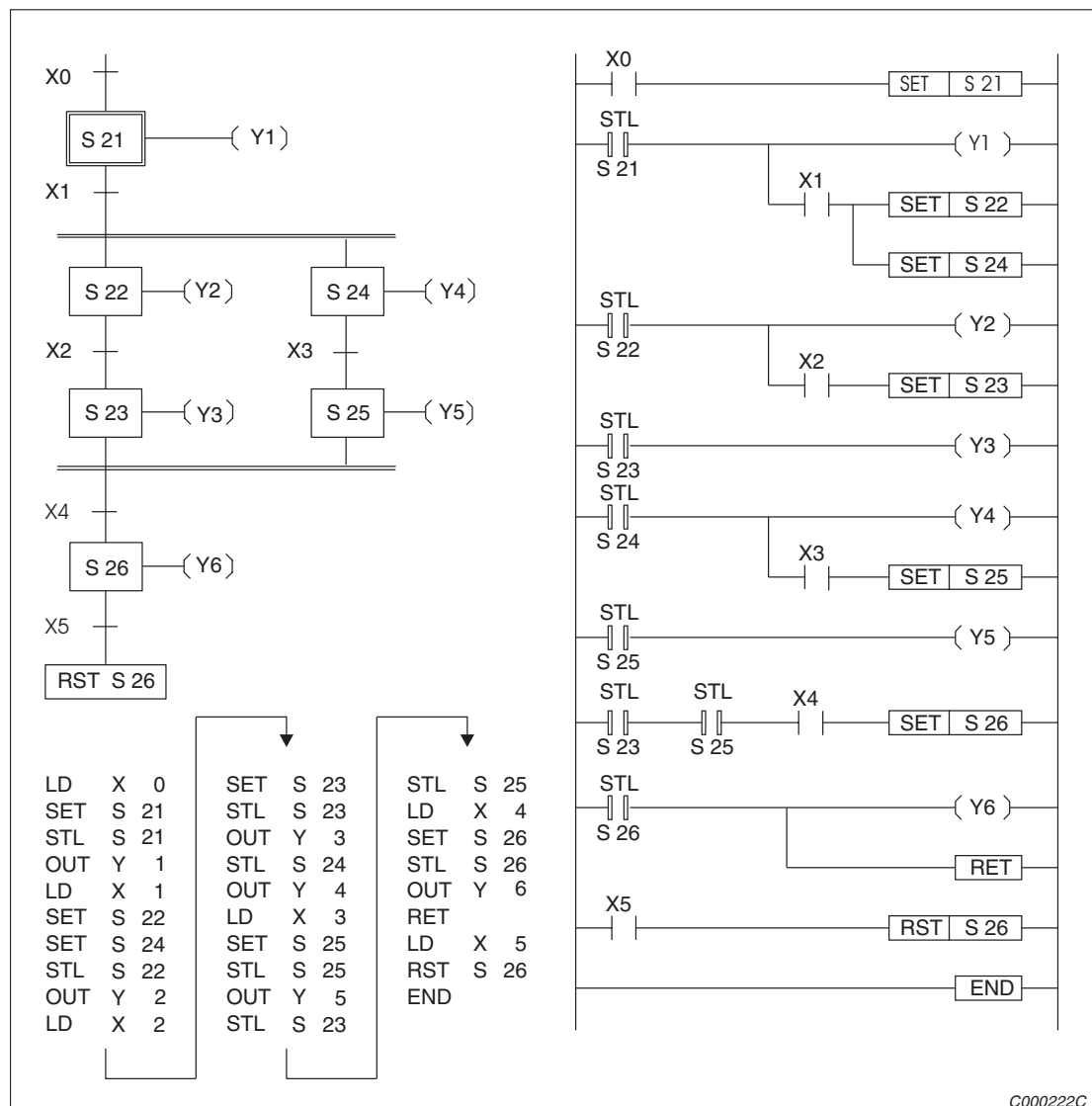


Рис.5-18. Параллельное разветвление

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

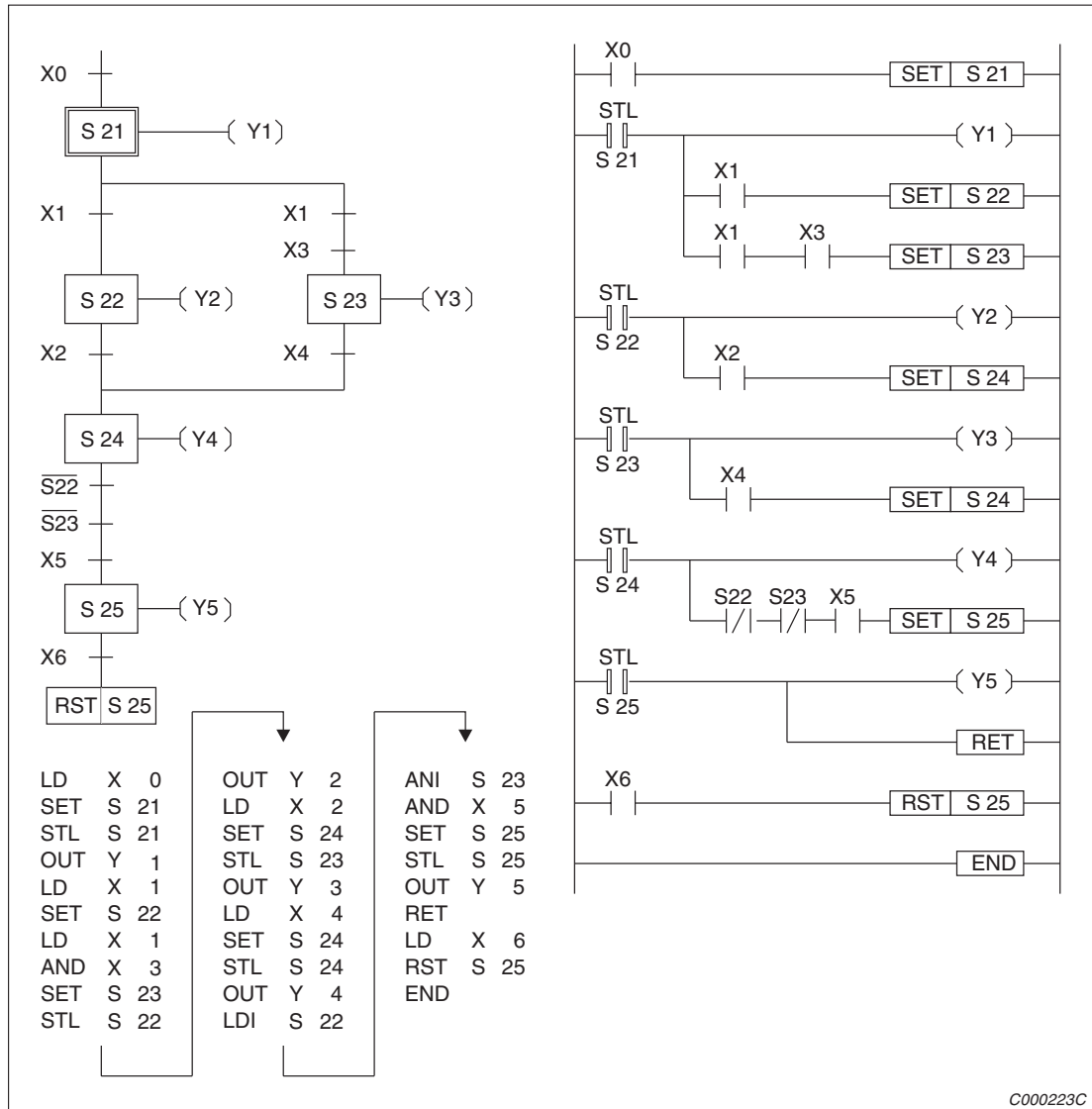


Рис.5-19. Комбинация из селективного и параллельного разветвления

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.

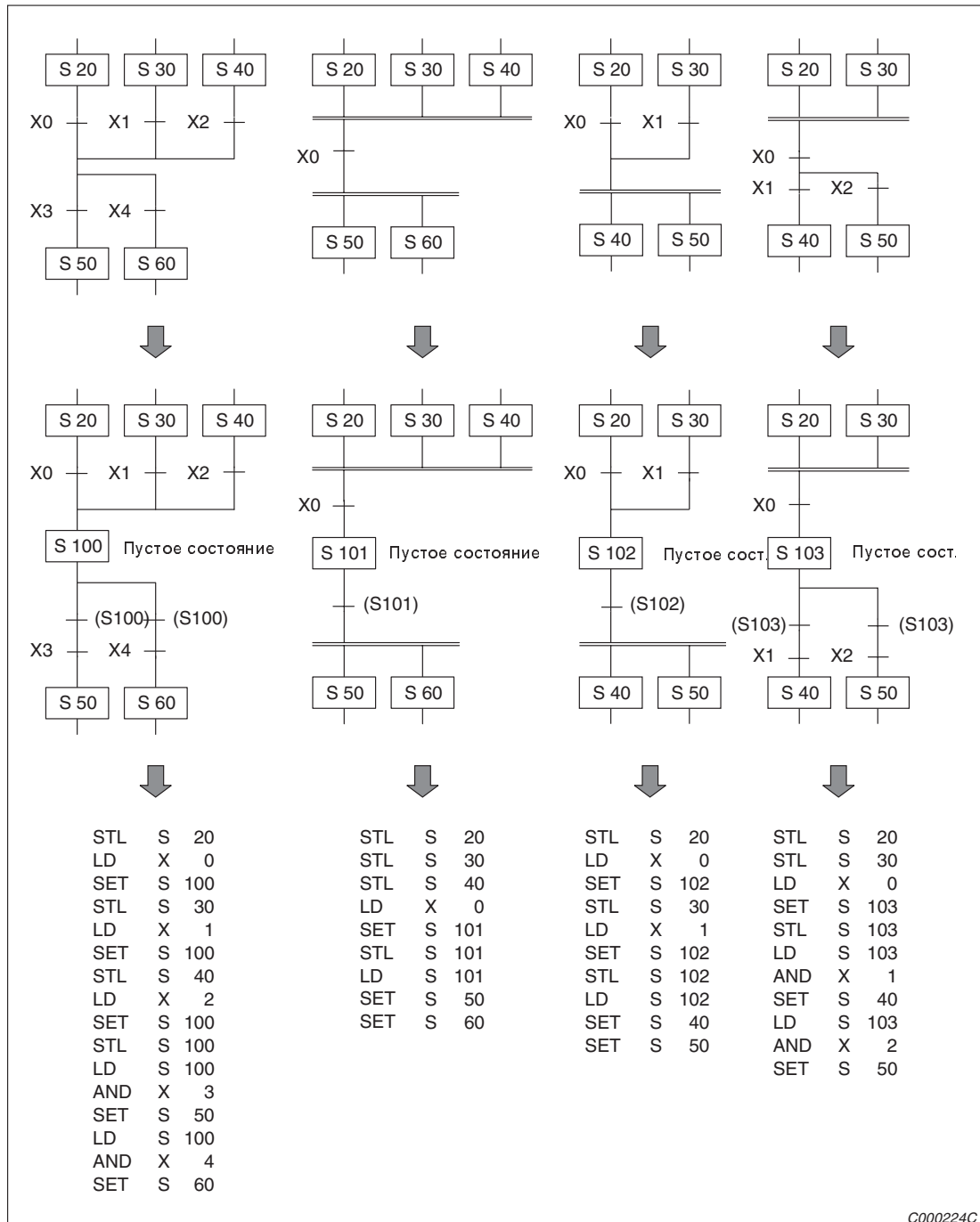
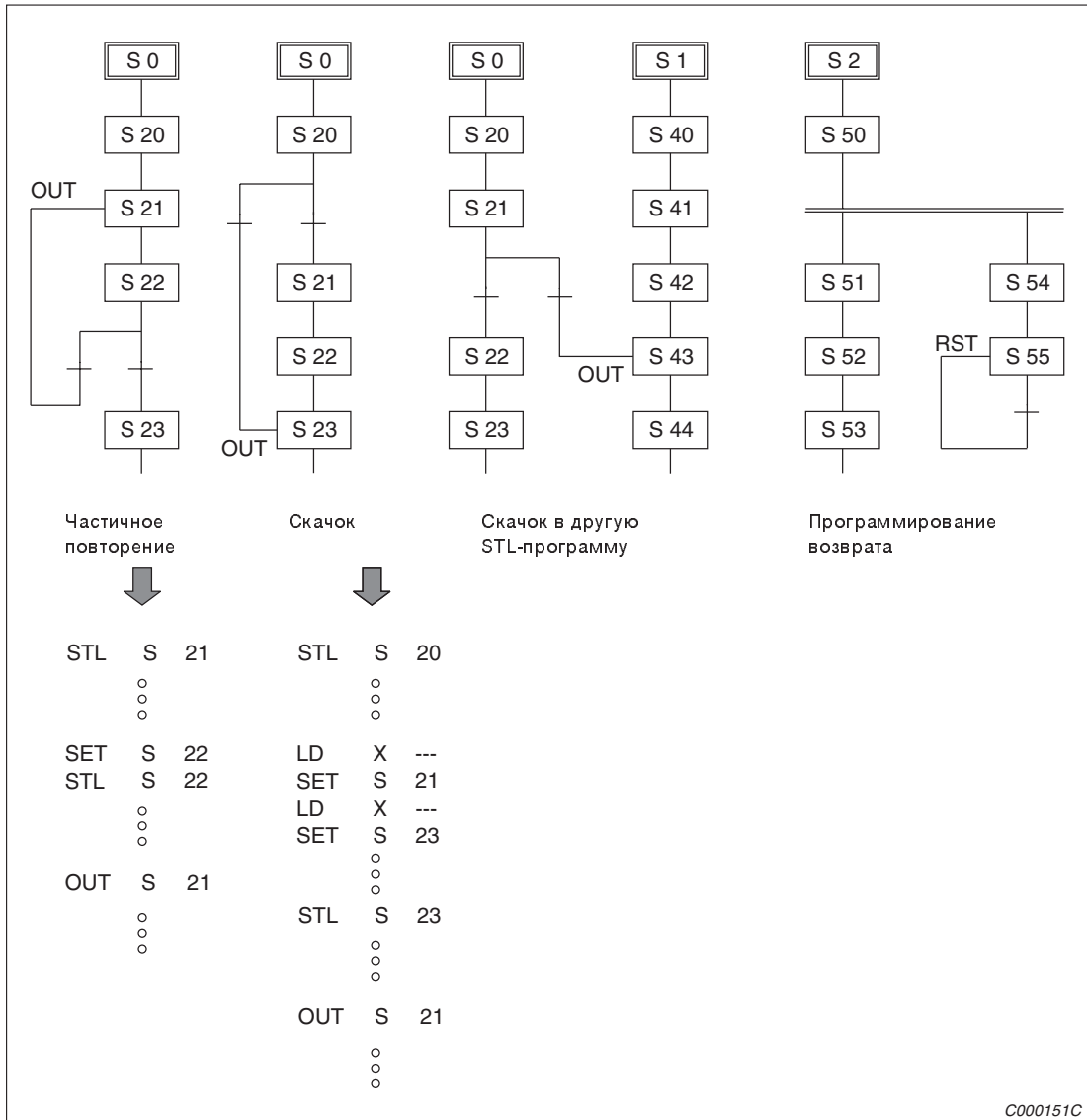


Рис.5-20. Программирование пустого состояния

5.4.6 ПЕРЕХОДНОЕ РАЗВЕТВЛЕНИЕ

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



C000151C

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

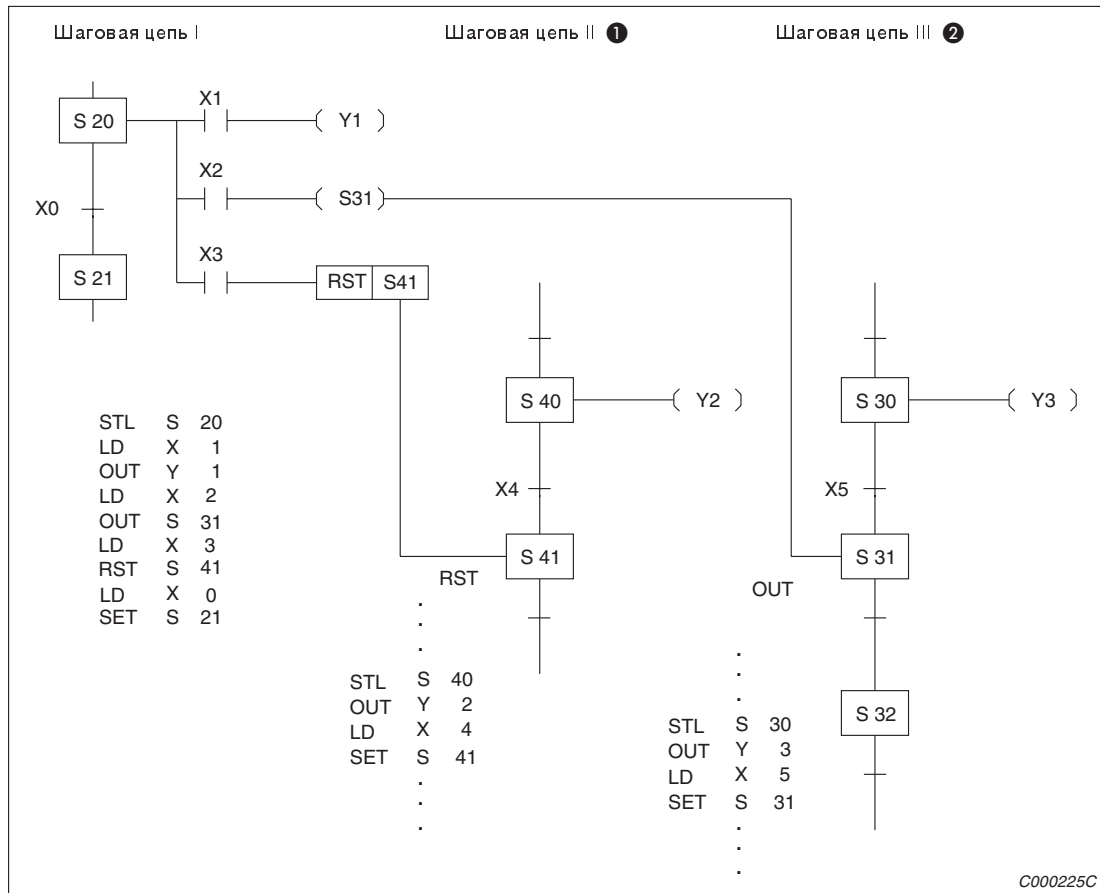


Рис.5-22. Условие последующего включения в другую шаговую цепь

- 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.

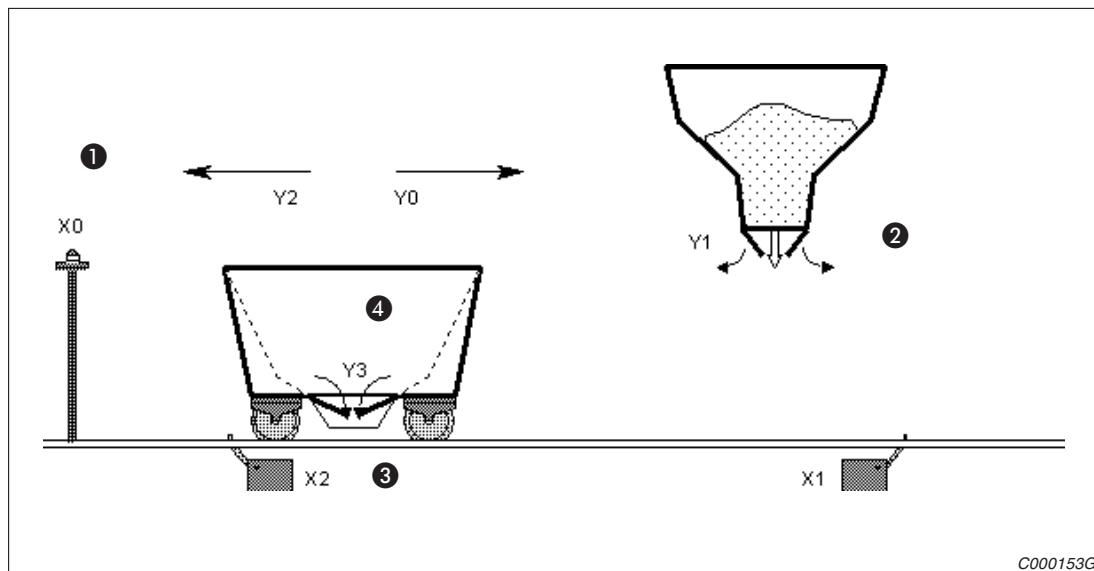


Рис.5-23 Пример контроля загрузки и разгрузки контейнера

- 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 ▾

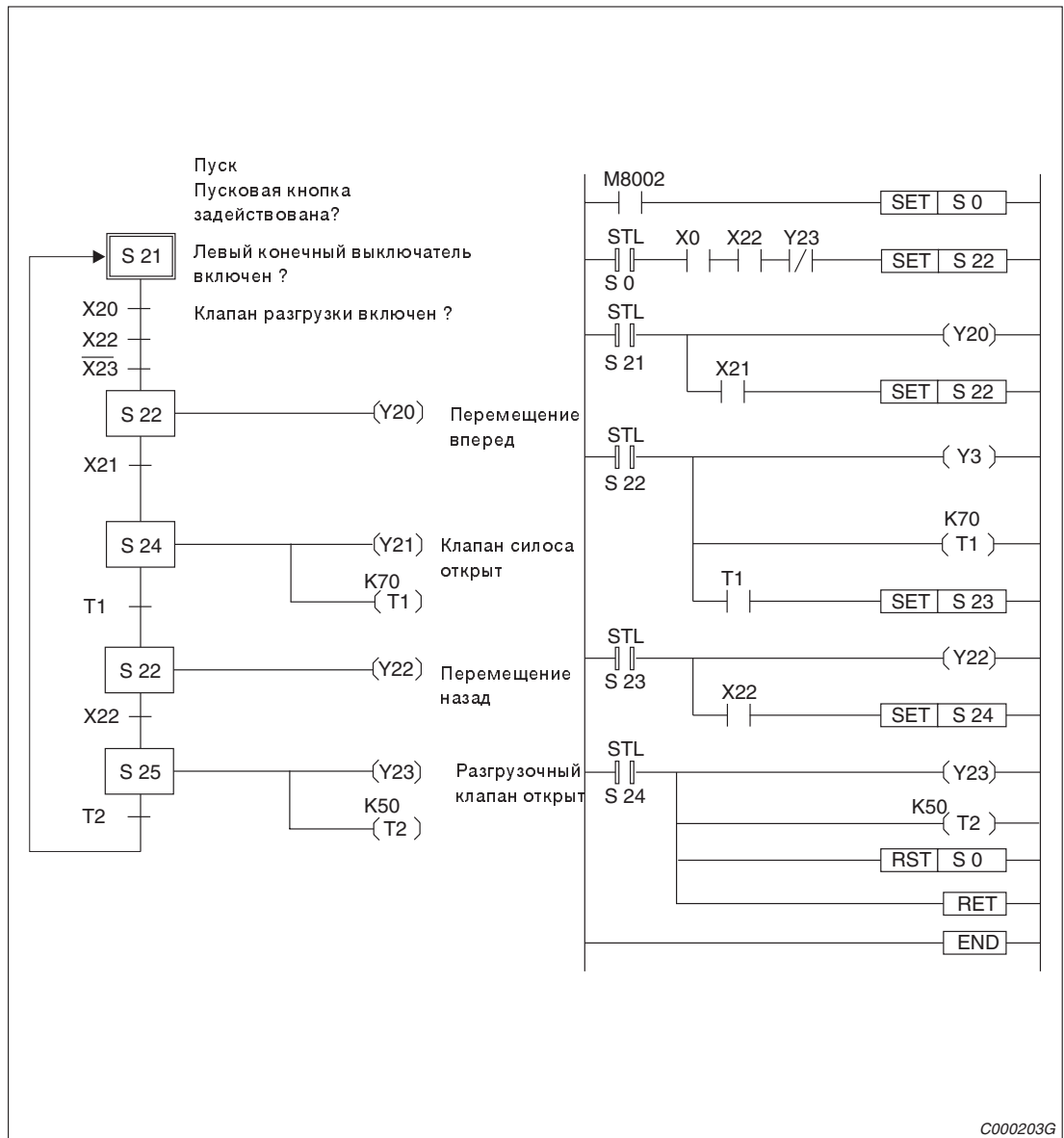


Рис.5-24. Программа процесса контроля загрузки и разгрузки контейнера



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.

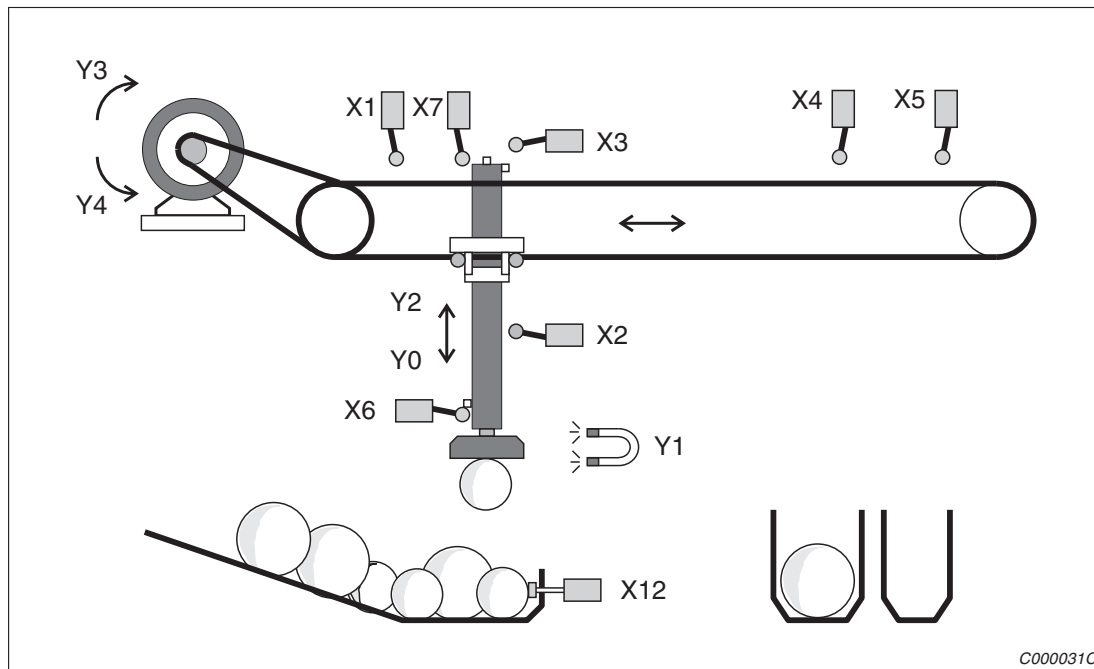


Рис.5-25. Устройство сортировки шаров

- 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).

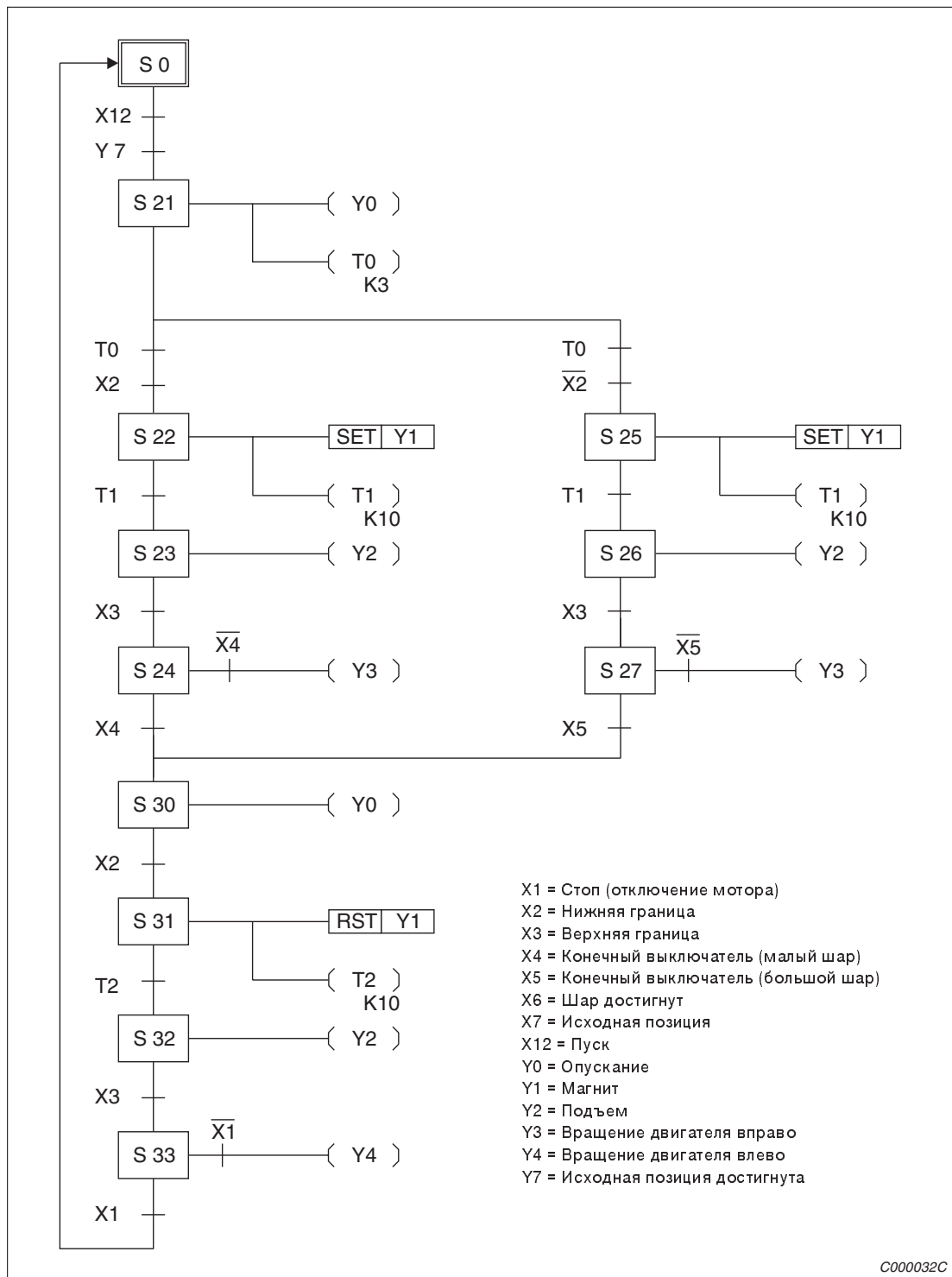


Рис.5-26. Пример программирования приведенного впереди устройства сортировки

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.

		CMP		FNC 10							
		Назначение инструкции (Circled '3' above)									
Операнд S1+, S2+ K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		D+ Y, M, S		Имп. инструкция (P)				Обработка		Шаги программы	
				CPU (Circled '4' above)		16 Bit		32 Bit		CMP	
				●		● ●					

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 ПОСЛОВНАЯ ОБРАБОТКА

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

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

ПРИМЕР ▾ Установка длины блока и начального адреса.

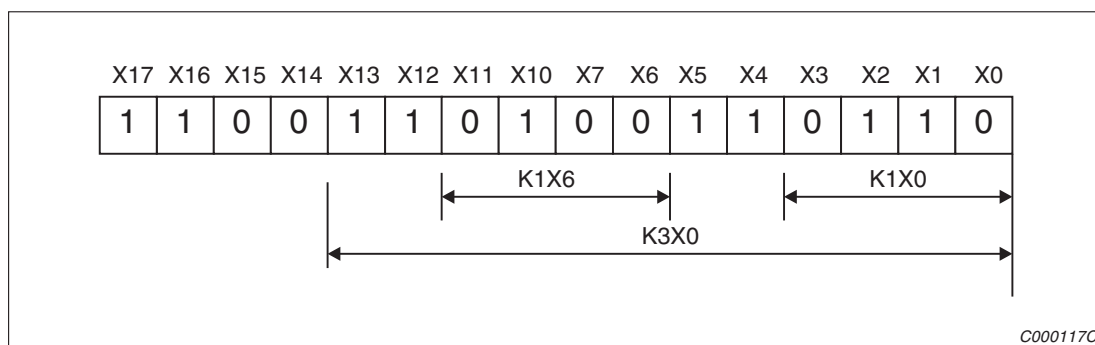


Рис. 6-1. Пример установки длины блока и начального адреса

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)

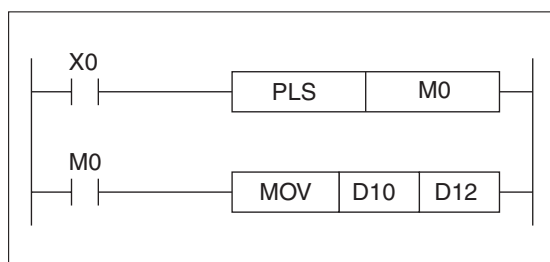


Рис. 6-2.

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.

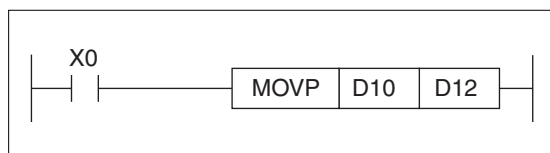


Рис. 6-3.

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

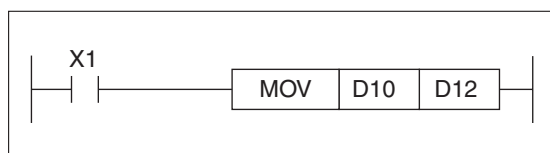


Рис. 6-4.

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 индицирование отмечаются знаком плюс "+": например, (S+) и (D+) (здесь S и D исходные и конечные данные)

ПРИМЕР ▾

Применение индексных регистров V, Z

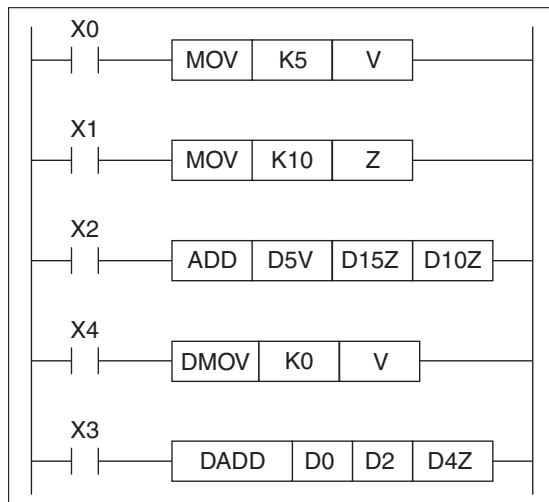


Рис. 6-5. Пример программирования индексных регистров V, Z

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).

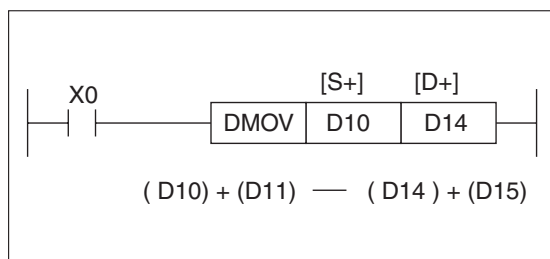


Рис. 6-6.

Пример программирования 32-х битной инструкции DMOV

C000209C

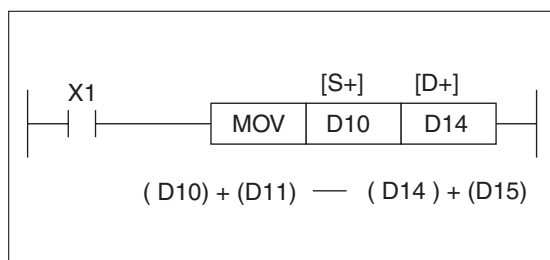


Рис. 6-7.

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
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
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	
Операнды	Имп. инструкция (P)		Обработка		Шаги программы	
Точки P0...P63 (индексирование-отражение допустимо) P63 означает переход к концу инструкции.	●		16 Bit	32 Bit	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 то же время. Это может привести к ошибочной обработке 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

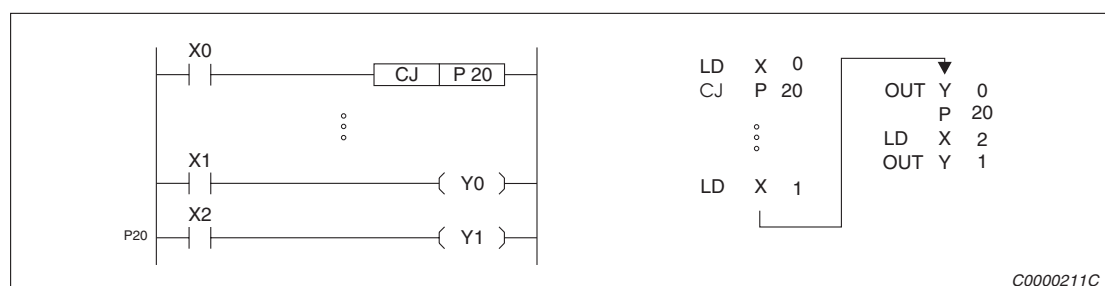


Рис. 6-8. Пример программирования CJ-инструкции

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

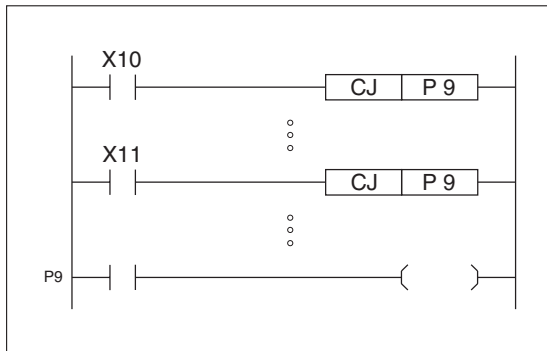


Рис. 6-9.
Пример программирования для двухразовой вставки в программу адреса точки P9

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).

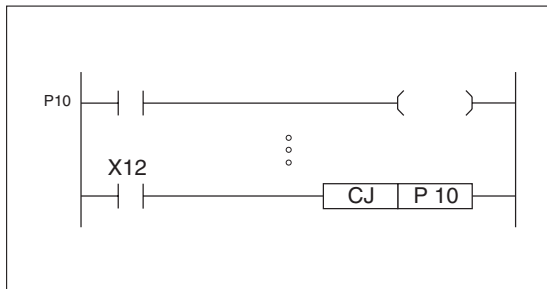


Рис. 6-10.
Пример программирования по установке точки маркировки перед CJ-инструкцией

C000213C

ПЕРЕХОДЫ В ОБЛАСТИ ГЛАВНОГО УПРАВЛЕНИЯ (Master-Control)

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

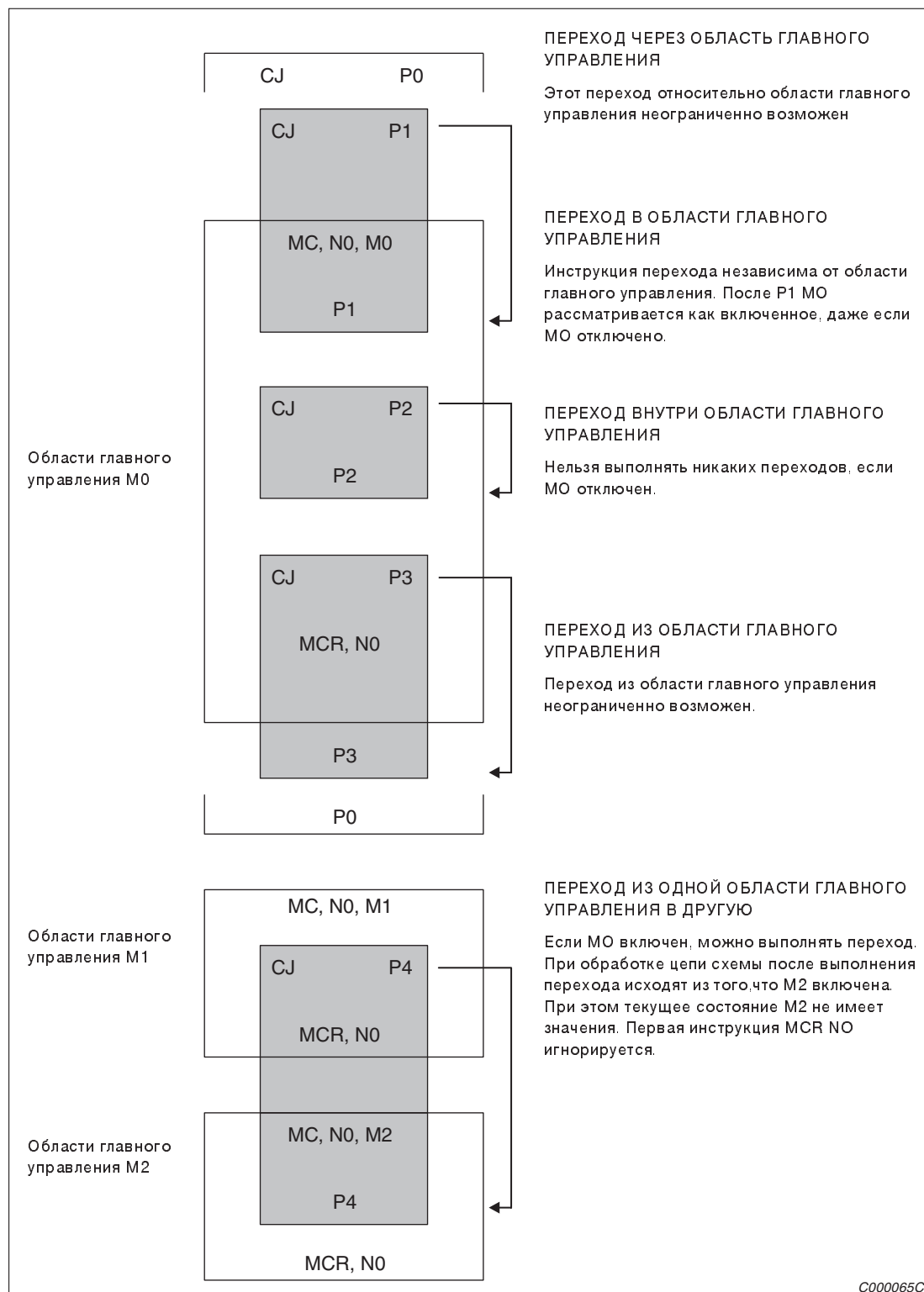


Рис. 6-11. Примеры переходов в области главного управления

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	
Операнды		D					
		Точки P0...P62 (индексирование-отражение допустимо)	●				
		Имп. инструкция (P)	Обработка		Шаги программы		
			16 Bit	32 Bit	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				
		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
Операнды	D	Имп. инструкция (P)	Обработка		Шаги программы	
	—	●	16 Bit	32 Bit	SRET	1

ФУНКЦИЯ

С помощью SRET-инструкции определяется конец подпрограммы.

ОПИСАНИЕ

- Подпрограмма маркируется с помощью точек (P0...P62) и вызывается CALL-инструкцией.
- 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 ▾

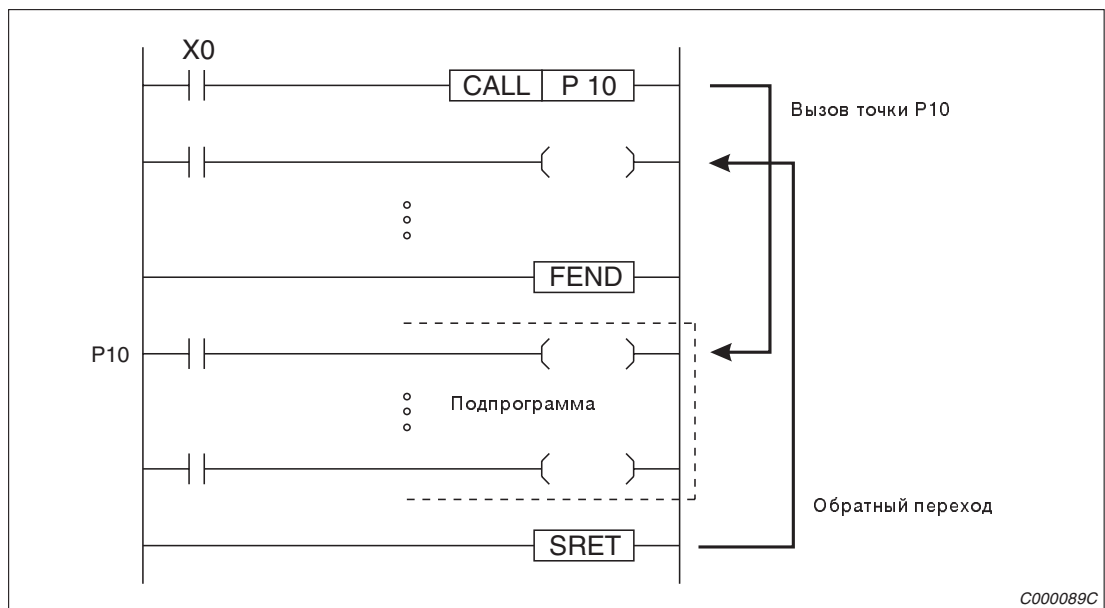


Рис. 6-12. Пример программирования с применением CALL- и SRET- инструкций



6.2.4 ПРИМЕНЕНИЕ ПРЕРЫВАНИЯ ПРОГРАММЫ (IRET, EI, DI)

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

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

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

ПРИНЦИП ФУНКЦИОНИРОВАНИЯ

Вызов, Окончание, Активизация и Деактивизация программы 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.

FX2N

- Допускает прерывание по времени.
- Допускает прерывание по счету.

УКАЗАНИЕ

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

FX2N/FX2NC**ПРЕРЫВАНИЕ ПО ВРЕМЕНИ (только для FX2N и FX2NC)**

- Программа прерывания обрабатывается соответственно после завершения предварительно заданного времени.

Точка прерывания: I ◆◇◇

◇◇: время интервала 10...99 мс

◆: адрес 6...8 (T6...T8)

FX2N/FX2NC**ПРЕРЫВАНИЕ ПО СЧЕТУ (только для FX2N и FX2NC)**

- 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 повременно 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 IO①②.

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

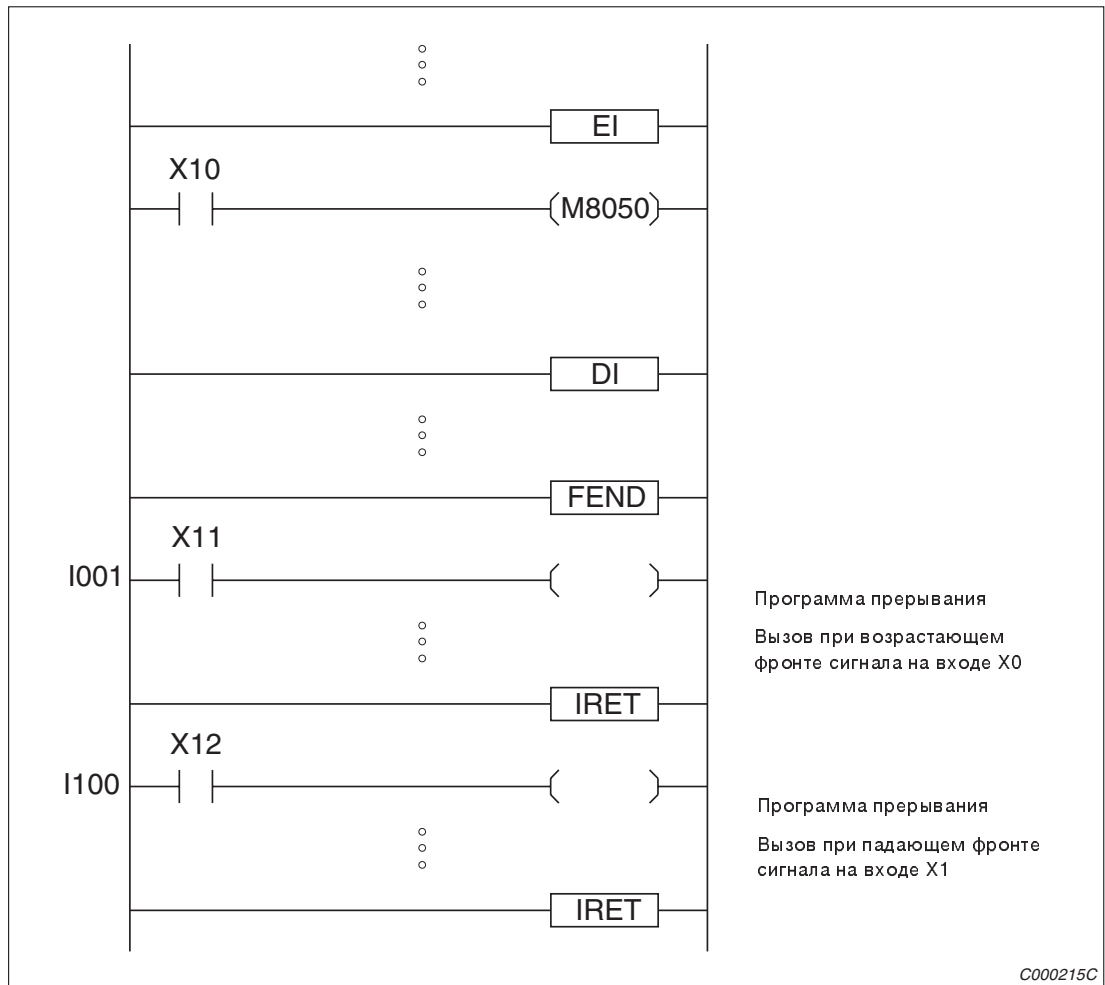


Рис. 6-13. Пример программирования при использовании инструкций EI, DI и IRET

- 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	CPU	FX1S	FX1N	FX2N	FX2NC
		-	●	●	●	●
Имп. инструкция (P)		Обработка		Шаги программы		
		16 Bit	32 Bit	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.

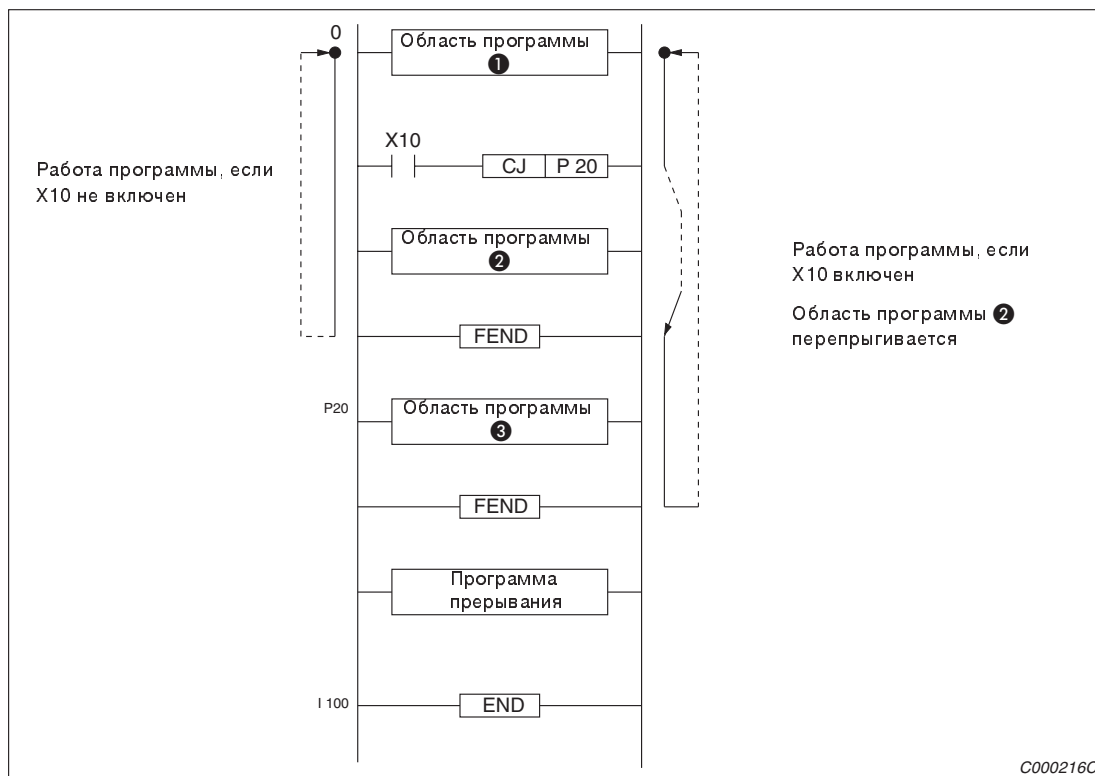


Рис. 6-14. Пример программирования при использовании FEND-инструкции



6.2.6 ОБНОВЛЕНИЕ КОНТРОЛЯ ВРЕМЕНИ ЦИКЛА (WDT)

		WDT		FNC 07			
		Обновление контроля времени цикла					
Операнды	D	Имп. инструкция (P)	Обработка		Шаги программы		
	—		16 Bit	32 Bit	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.

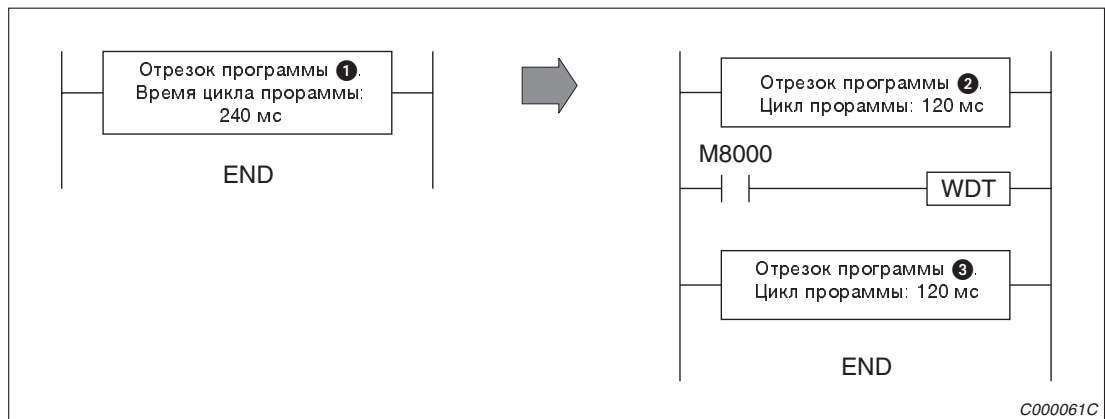


Рис. 6-15. Пример программирования при использовании WDT-инструкции

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.

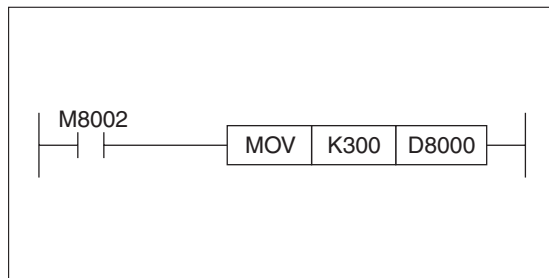


Рис. 6-16.

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					
Операнды		S	Имп. инструкция (P)	Обработка		Шаги программы	
				CPU	FX1S	FX1N	FX2N
				●	●	●	●
		K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		16 Bit	32 Bit		
				●			

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

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.

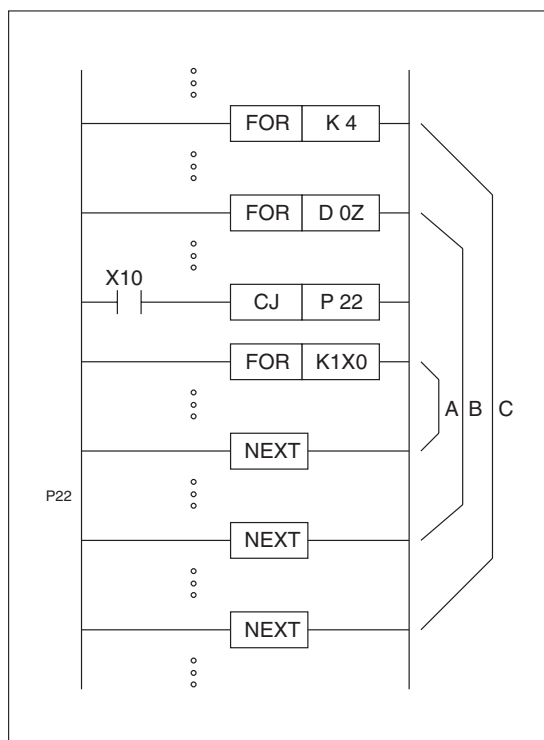
ПРИМЕР ▾ | Применение FOR- и NEXT-инструкций


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 C обрабатывается четыре раза (здесь K4 константа 4). В конце обработки последний программный шаг выполняется после третьей NEXT-инструкции.
- При каждом исполнении отрезка C отрезок программы B обрабатывается шесть раз, если в регистре данных D0Z записано число 6.
- Поэтому отрезок B 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 B отрезок программы обрабатывается семь раз.
- Поэтому отрезок A обрабатывается $168 (6 \times 4 \times 7)$ раз.

△

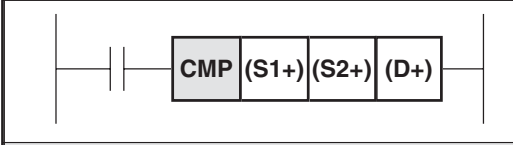
6.3 ИНСТРУКЦИИ СРАВНЕНИЯ И ПЕРЕДАЧИ

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
BCD	18	BCD-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, DCMPP)

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

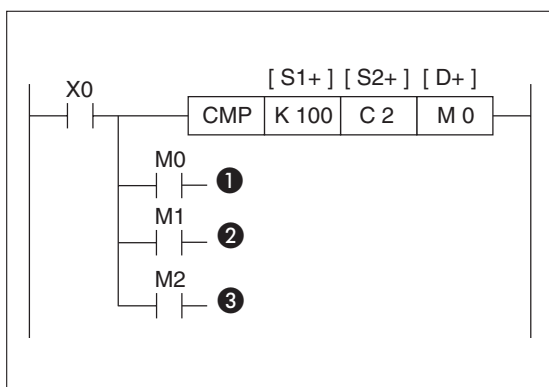


Рис. 6-18.
Пример программирования с использованием CMP-инструкции

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			
		●	●	●	●				
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	●	16 Bit	32 Bit	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 источниках (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

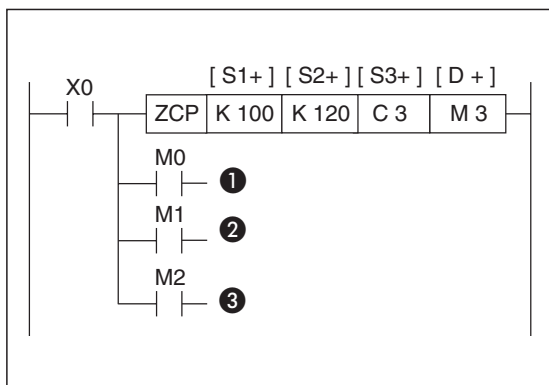


Рис. 6-19.
Пример программирования с использованием ZCP-инструкции

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	
			●	●	●	●	
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		●	16 Bit	32 Bit	MOV/MOVP
				●	●	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

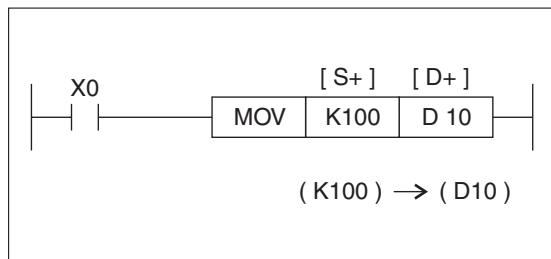


Рис. 6-20.

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				
				CPU	FX1S	FX1N	FX2N	FX2NC
							●	●
Операнды	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		●	16 Bit	32 Bit	SMOV/ SMOVP

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

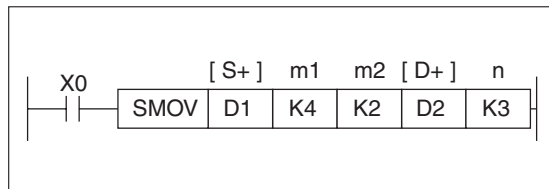


Рис. 6-21. Пример программирования с использованием SMOV-инструкции без специального меркера

C000125C

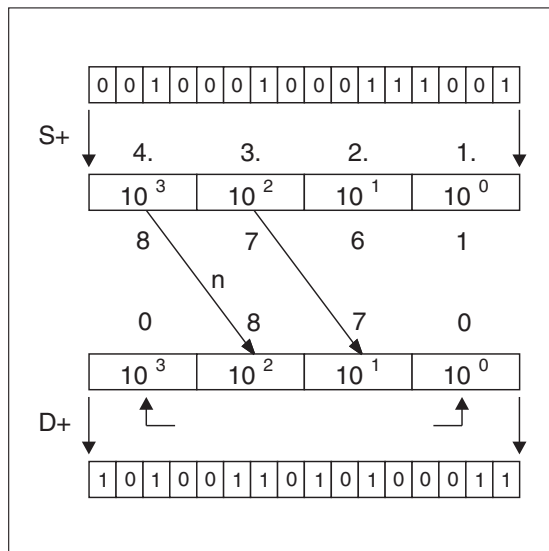


Рис. 6-22. Конвертирование и передача

C000128C

ПРИМЕР ▾

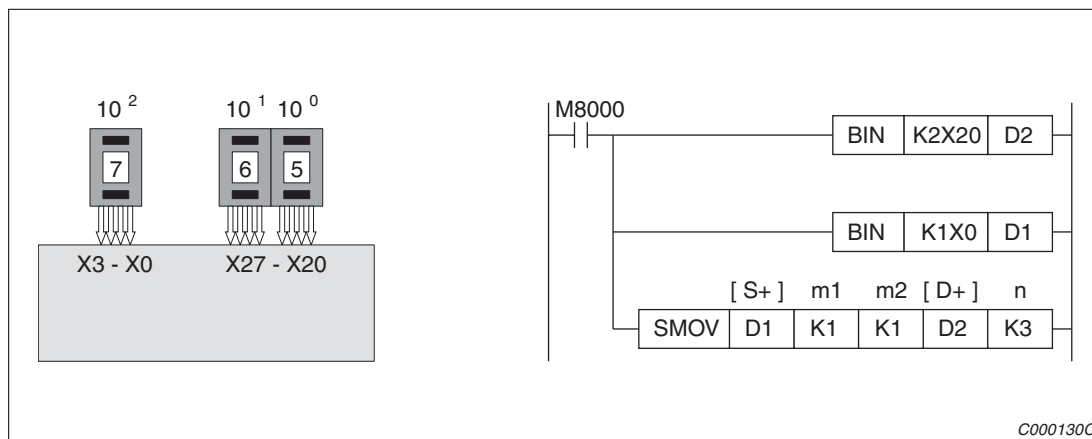


Рис. 6-23. Пример программирования для двоичных данных

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

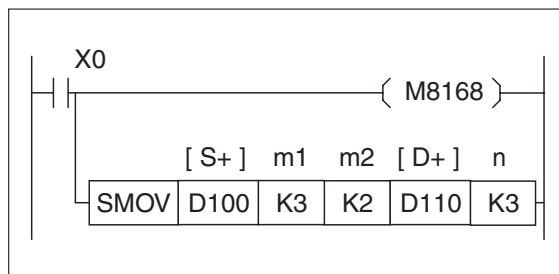


Рис. 6-24.
Пример программирования с использованием SMOV-инструкции

C000312C

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

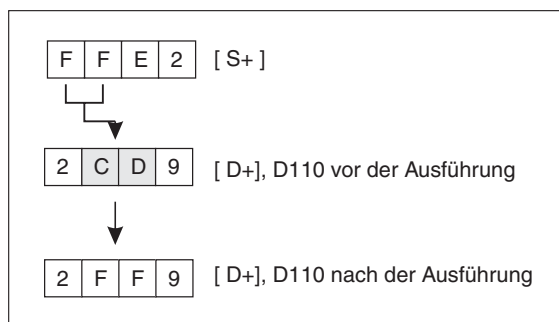


Рис. 6-25.
Сдвиг в регистре данных (D110 перед выполнением / D110 после выполнения)

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					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
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	●	16 Bit	32 Bit	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

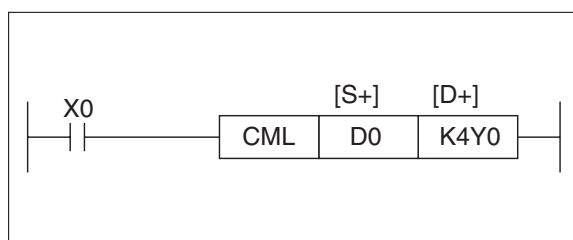
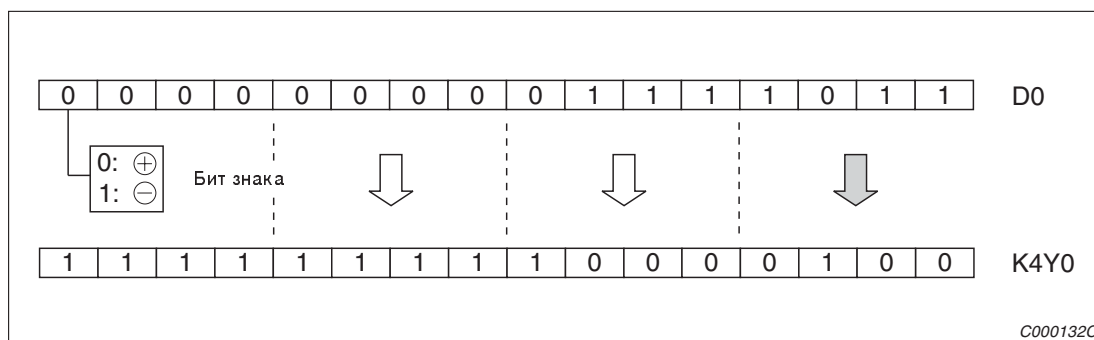


Рис. 6-26. Пример программирования для CML-инструкции

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	
					●	●	●	●	
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-Register	K, H		●	16 Bit	32 Bit	BMOV/ BMOV P

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.

ПРИМЕР ▾

BMOV-инструкции

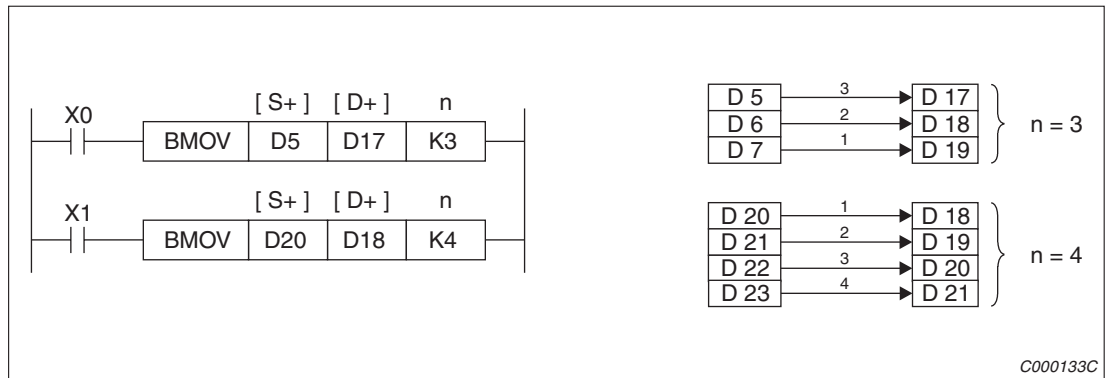


Рис. 6-28. Передача блока



6.3.7 ПЕРЕДАЧА ОДИНАКОВЫХ ДАННЫХ (FMOV)

				FMOV		FNC 16			
				Передача одинаковых данных					
				CPU	FX1S	FX1N	FX2N	FX2NC	
							●	●	
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	●	16 Bit	32 Bit	FMOV/ FMOVP	7
					●	●	DFMOV/ DFMOVP	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

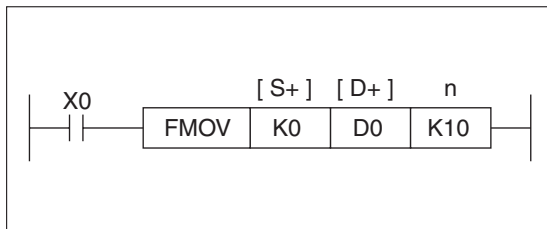


Рис. 6-29. Пример программирования FMOV-инструкции

C000134C

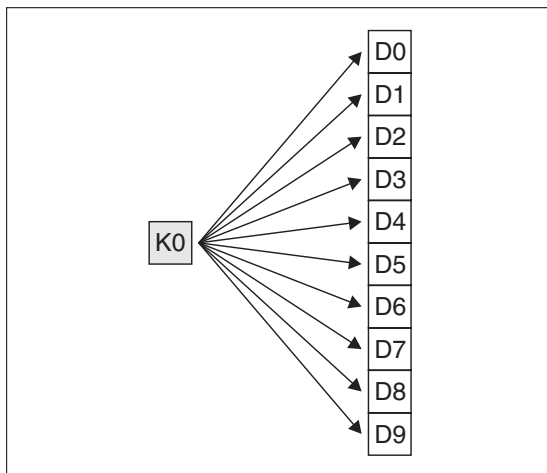


Рис. 6-30. Передача (перенос) значения “0” в регистры данных D0...D9

C000119C



6.3.8 ОБМЕН ДАННЫМИ (XCH)

		XCH		FNC 17		
		Обмен данными				
		CPU	FX1S	FX1N	FX2N	FX2NC
					●	●
Операнды	D1+, D2+	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS T, C, D, V, Z	●	16 Bit	32 Bit	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

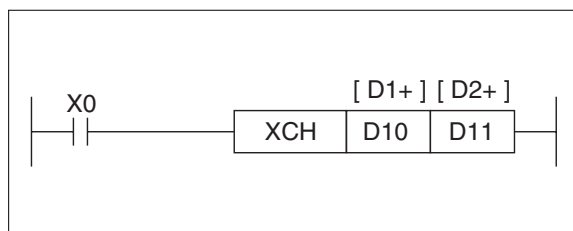


Рис. 6-31.
Пример программирования XCH-инструкции без специального меркера

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

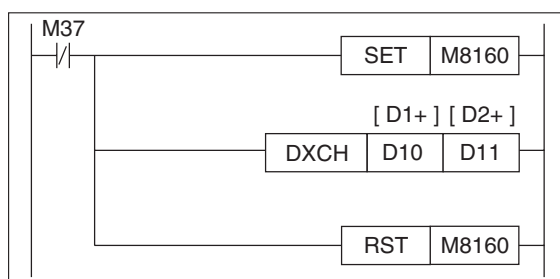


Рис. 6-32.
Пример программирования с DXCH-инструкцией со специальным меркером M8160

C000314C

Процесс обмена может быть представлен следующим образом:

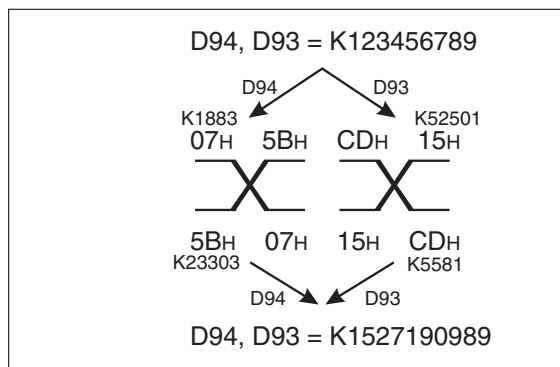


Рис. 6-33.
Представление процесса обмена (DXCH-инструкция обмена)

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	
			●	●	●	●	
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		●	16 Bit	32 Bit	BCD/BCDP
				●	●	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.

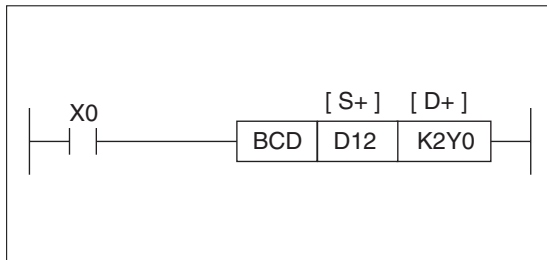
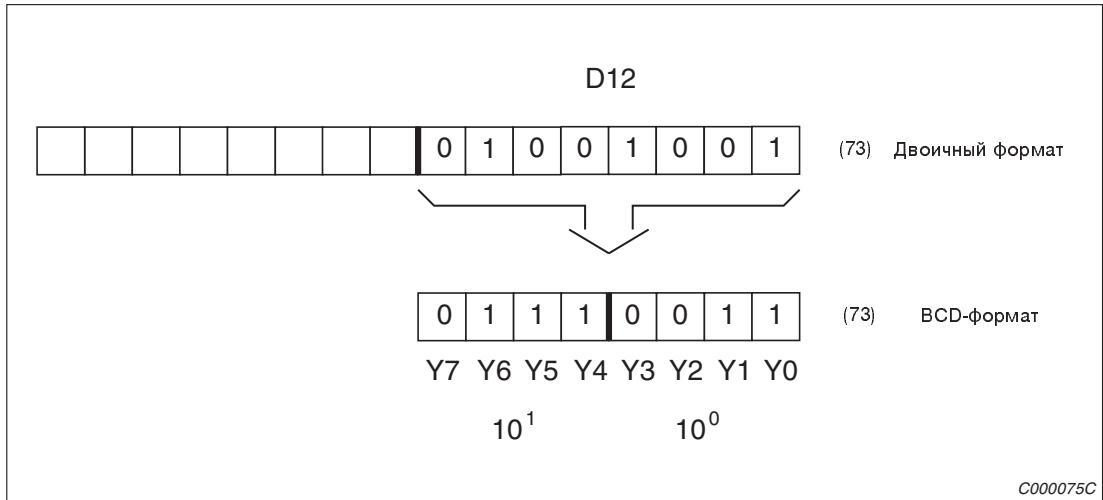


Рис. 6-34.
Пример программирования BCD-инструкции

C000074C

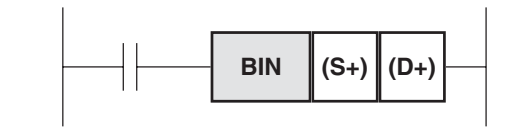


C000075C

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

Двоичные данные из регистра данных D 12 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	
			●	●	●	●	
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	●	16 Bit	32 Bit	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 ▾

ВІN-инструкция может применяться, например, чтобы передать ВCD-данные от цифрового переключателя в ПК.

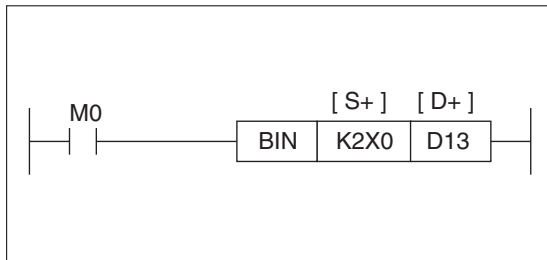
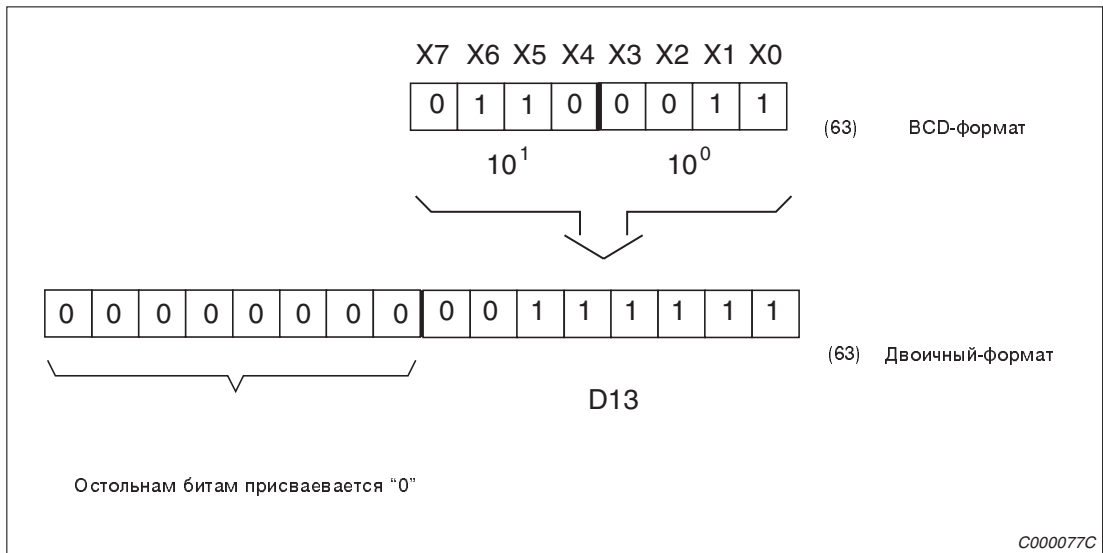


Рис. 6-36.
Пример программирования ВІN-инструкции

C000076C



C000077C

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

ВCD-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	
			●	●	●	●	
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		●	16 Bit	32 Bit	ADD/ADDP
				●	●	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

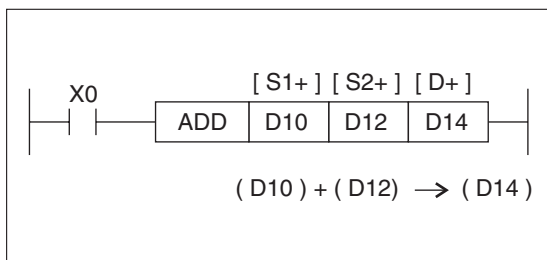


Рис. 6-38.
Пример программирования
ADD-инструкции

C000078C

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

Применение DADD-инструкции

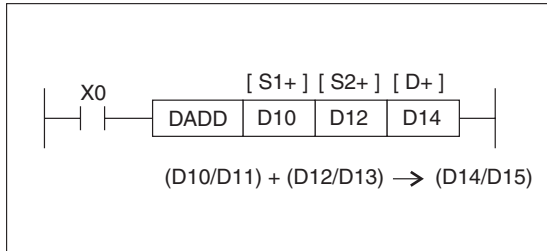


Рис. 6-39.
Пример программирования
DADD-инструкции

C000069C



6.4.2 ВЫЧИТАНИЕ ЧИСЛОВЫХ ДАННЫХ (SUB, DSUB)

		SUB		FNC 21			
		Вычитание числовых данных					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
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	●	16 Bit	32 Bit	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

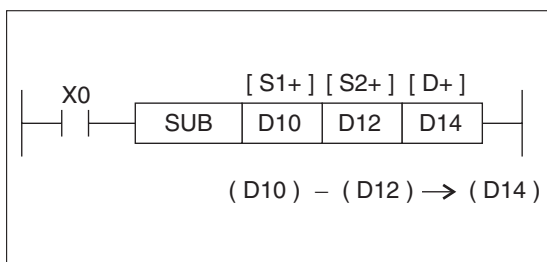


Рис. 6-40.

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

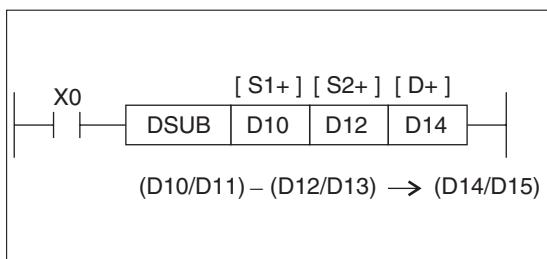


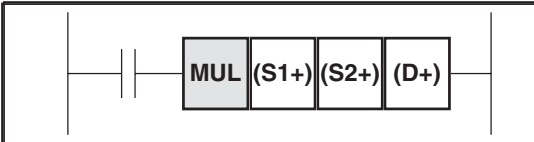
Рис. 6-41.

Пример программирования DSUB-инструкции

C000067C



6.4.3 УМНОЖЕНИЕ ЧИСЛОВЫХ ДАННЫХ (MUL, DMUL)

		MUL		FNC 22			
		Умножение числовых данных					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
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)	●	16 Bit	32 Bit	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

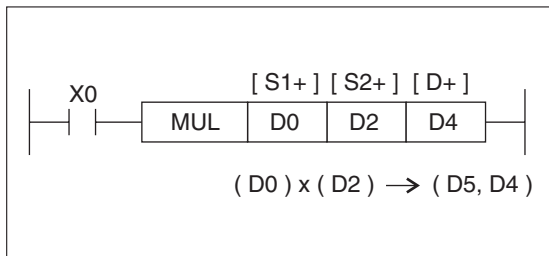


Рис. 6-42.
Пример программирования MUL-инструкции

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

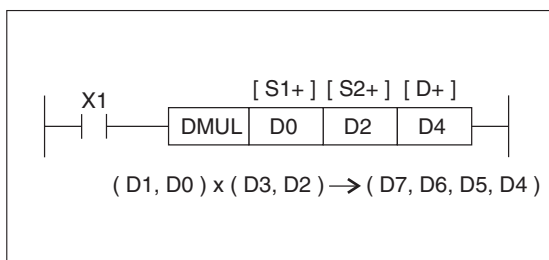


Рис. 6-43.
Пример программирования DMUL-инструкции

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	
			●	●	●	●	
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)		●	16 Bit	32 Bit	DIV/DIVP
				●	●	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

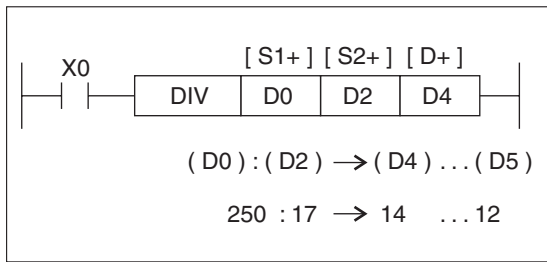


Рис. 6-44.

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

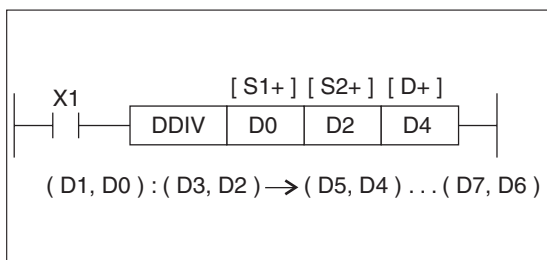


Рис. 6-45.

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				
		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
Операнды	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z	●	16 Bit	32 Bit	INC/INCP	3
			●	●	DINC/DINCP	5

ФУНКЦИЯ

К числовому значению данных прибавляется число 1 (приращивается).

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).

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

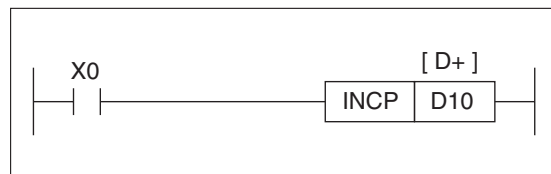


Рис. 6-46.
Пример программирования INCP-инструкции

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				
		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
Операнды	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z	●	16 Bit	32 Bit	DEC/DECP	3
			●	●	DDEC/DDECP	5

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

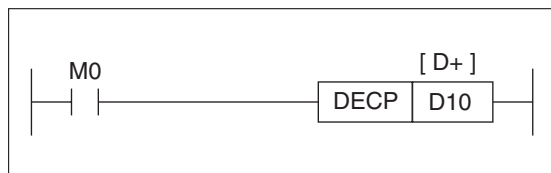


Рис. 6-47.
Пример программирования
DECP-инструкции

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/DAND		FNC 26			
		Логическая связь “И”					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
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	●	16 Bit	32 Bit	WAND/ WANDP DAND/ DANDP	7 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”

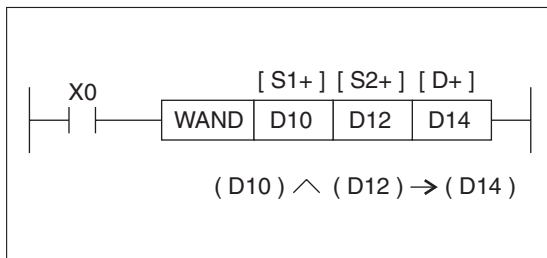
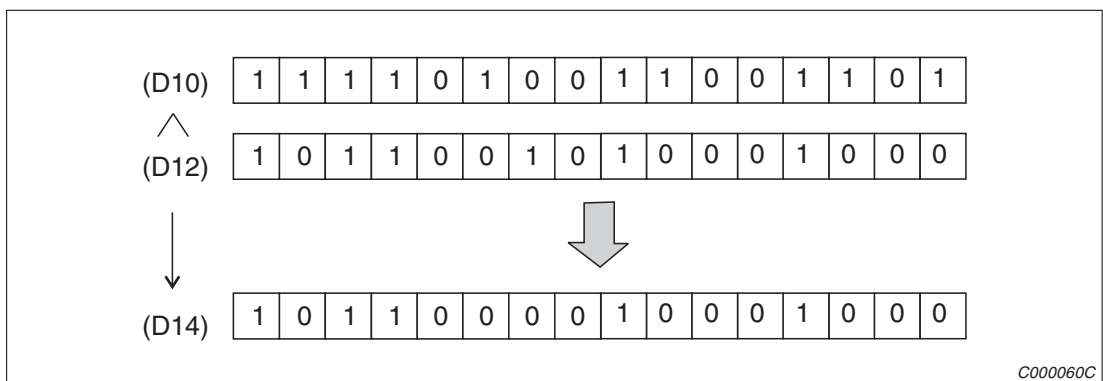


Рис. 6-48.
Пример программирования “И”-связи (выражения)

C000086C



C000060C

Рис. 6-49. Применение WAND-инструкции

6.4.8 ЛОГИЧЕСКАЯ СВЯЗЬ “ИЛИ” ДВОИЧНЫХ ДАННЫХ (WOR, DWOR)

		WOR/DOR		FNC 27		
		Логическая связь “ИЛИ”				
		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
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	●	16 Bit	32 Bit	WOR/WORP DOR/DORP
				●	●	7 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”

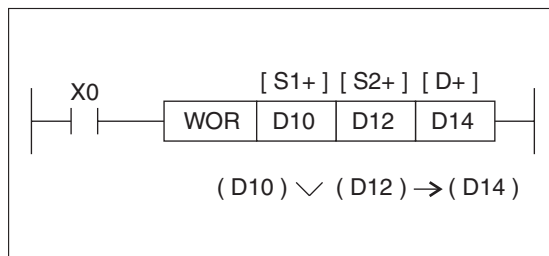
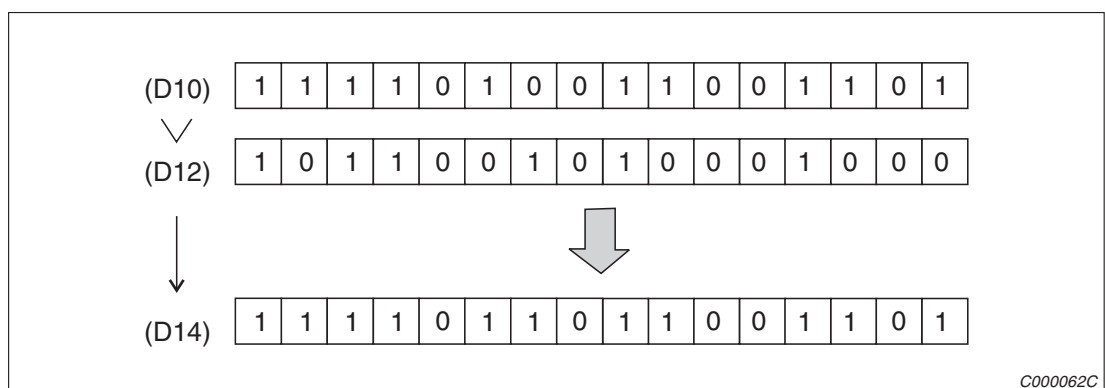


Рис. 6-50. Пример программирования WOR-инструкции (“ИЛИ”-связи)

C000087C



C000062C

Рис. 6-51. Применение WOR-инструкции

6.4.9 ЛОГИЧЕСКАЯ СВЯЗЬ “ИСКЛЮЧАЮЩЕГО ИЛИ” ДВОИЧНЫХ ДАННЫХ (WXOR, DXOR)

		WXOR/DXOR		FNC 28		
		Логическая связь “Исключающего ИЛИ”				
		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
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)		16 Bit	32 Bit	
			●	●	WXOR/ WXORP WXOR/ WXORP	7 13

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”

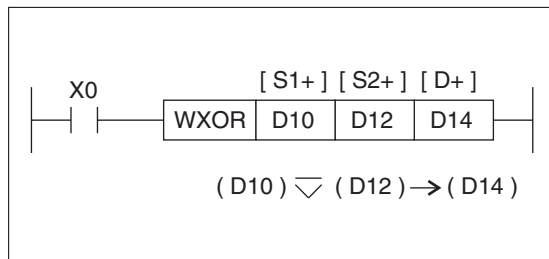
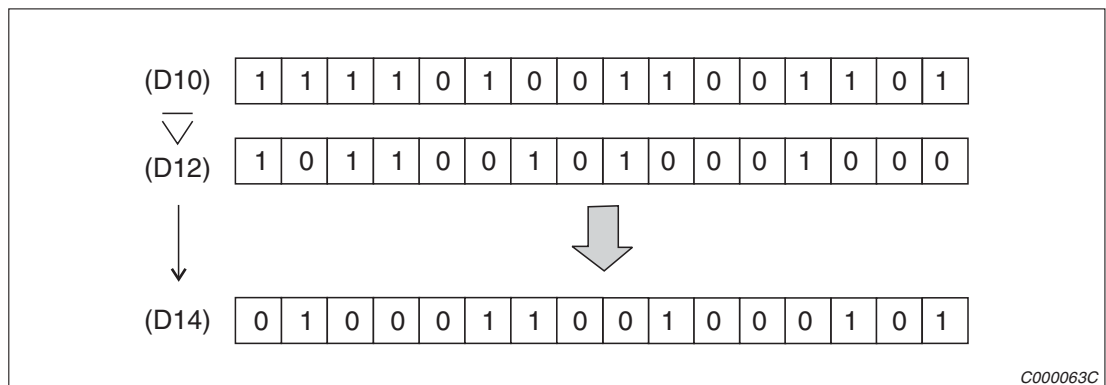


Рис. 6-52.

Пример программирования WXOR-инструкции (“Исключающего ИЛИ”-связи)

C000088C



C000063C

Рис. 6-53 Пример применение WXOR-инструкции

6.4.10 ИНВЕРСИЯ ДАННЫХ (NEG)

		NEG		FNC 29		
		инверсия данных				
		CPU	FX1S	FX1N	FX2N	FX2NC
					●	●
Операнды	D	Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS T, C, D, V, Z	●	16 Bit	32 Bit	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

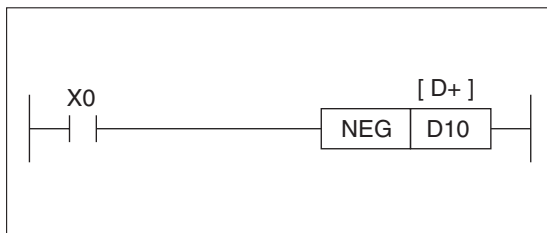
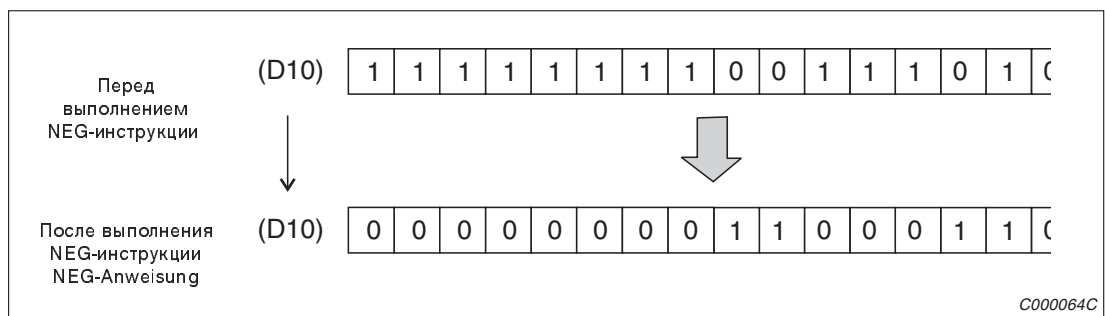


Рис. 6-54.
Пример программирования
NEG-инструкции

C000137C

ФУНКЦИЯ

Двоичное: $\overline{D10} + 1 \rightarrow D10$



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	
					●	●	
	D+	n	Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●	16 Bit	32 Bit	ROR, RORP	5
				●	●	DROR, DRORP	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

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 опознания 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.

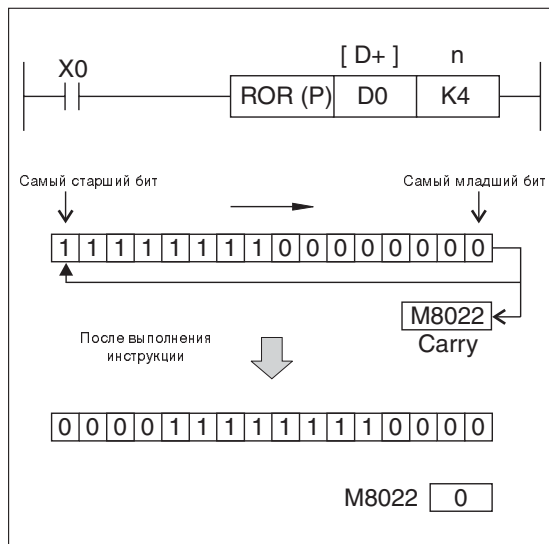


Рис. 6-56.

Пример программирования ротации вправо

C000091C



6.5.2 СДВИГ ВЛЕВО (ROL)

		ROL		FNC 31			
		Ротация влево					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
Операнды	D+	n	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●	16 Bit	32 Bit	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.

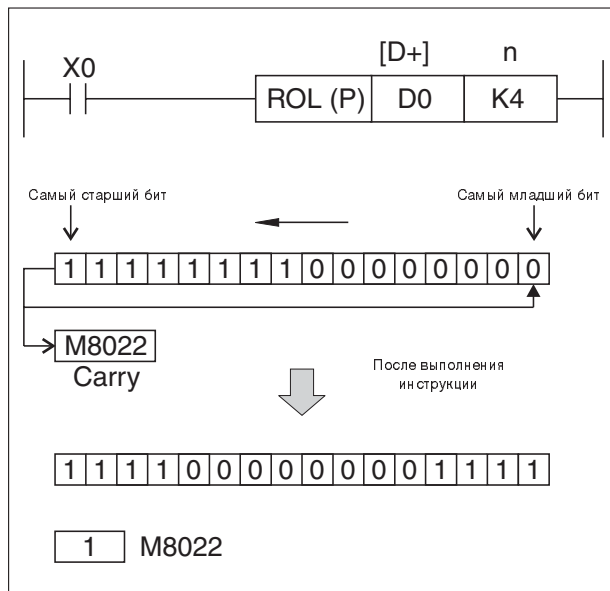


Рис. 6-57. Пример программирования ротации влево

C000092C

6.5.3 КОЛЬЦЕВОЙ СДВИГ ВПРАВО (RCR)

		RCR		FNC 32			
		Ротация бит вправо					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
Операнды	D+	n	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●	16 Bit	32 Bit	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.

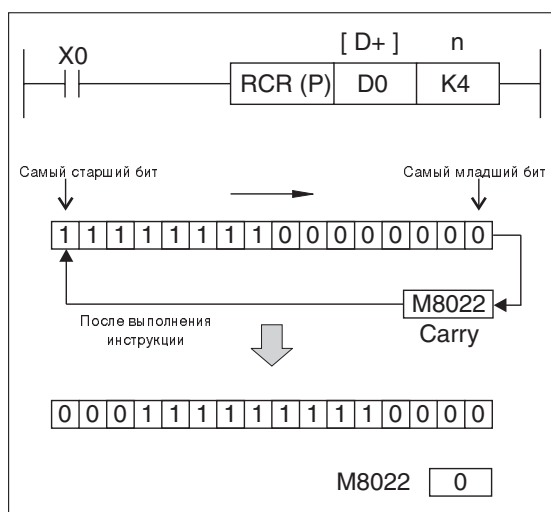


Рис. 6-58.
Пример программирования ротации вправо

C000093C



6.5.4 КОЛЬЦЕВОЙ СДВИГ ВЛЕВО (RCL)

		RCL		FNC 33			
		Ротация бит влево					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
Операнды	D+	n	Имп. инструкция (P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z ^①	K, H ^②	●	16 Bit	32 Bit	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.

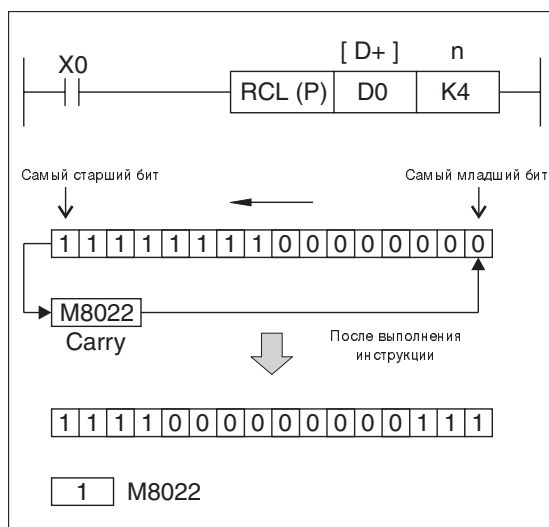


Рис. 6-59. Пример программирования ротации влево

C000094C



6.5.5 ПОБИТНЫЙ СДВИГ ДВОИЧНЫХ ДАННЫХ (SFTR, SFTL)

				SFTR		FNC 34				
				Побитный сдвиг двоичных данных вправо						
				CPU	FX1S	FX1N	FX2N	FX2NC		
				●	●	●	●			
Операнды	S+	D+	n1, n2	Имп. инструкция (P)	Обработка		Шаги программы			
	X, Y, M, S	Y, M, S	K, H		●	16 Bit	32 Bit	SFTR/ SFTRP	9	

				SFTL		FNC 35				
				Побитный сдвиг двоичных данных влево						
				CPU	FX1S	FX1N	FX2N	FX2NC		
				●	●	●	●			
Операнды	S+	D+	n1, n2	Имп. инструкция (P)	Обработка		Шаги программы			
	X, Y, M, S	Y, M, S	K, H		●	16 Bit	32 Bit	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

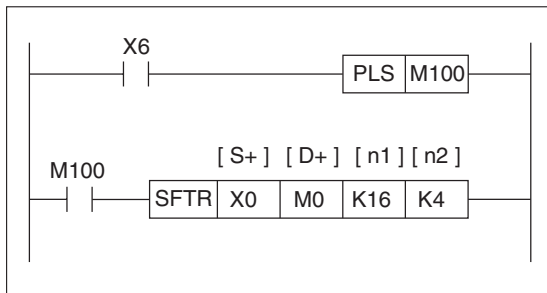
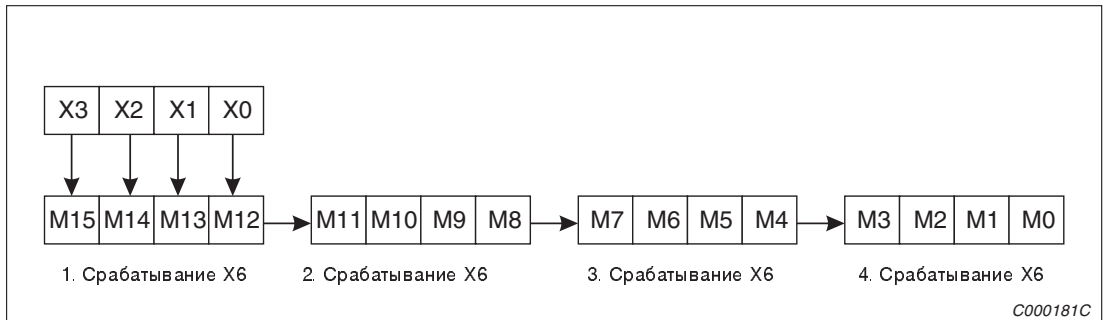


Рис. 6-60.
Пример программирования SFTR-инструкции

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

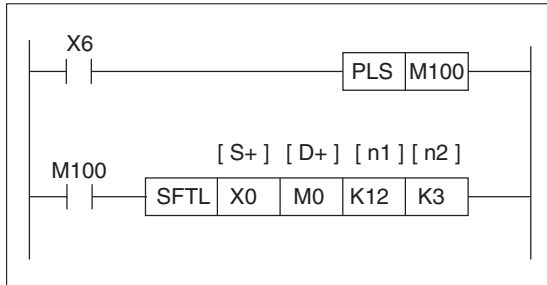
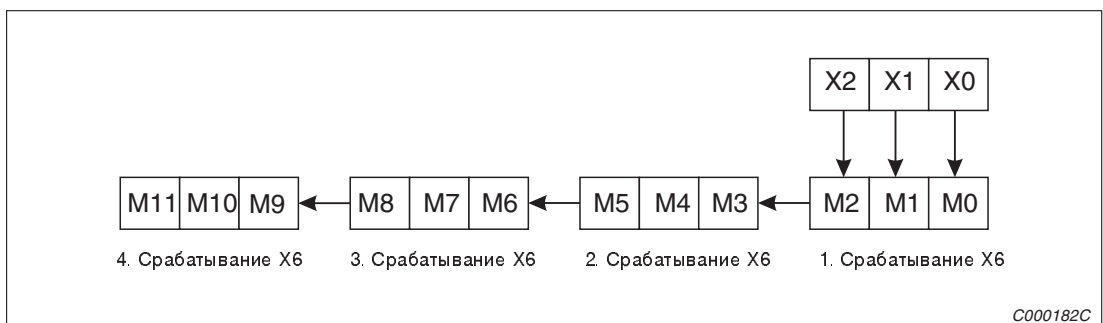


Рис. 6-62.
Пример программирования SFTL-инструкции

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	●	16 Bit	32 Bit	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

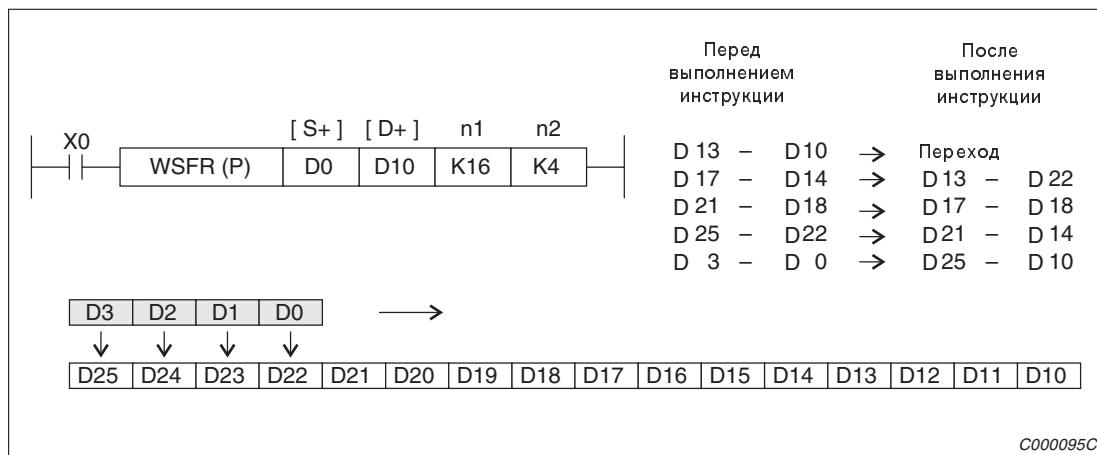


Рис. 6-64. Пример программирования сдвига вправо



6.5.7 ПОСЛОВНЫЙ СДВИГ ДАННЫХ ВЛЕВО (WSFL)

				WSFL		FNC 37		
				Пословный сдвиг данных влево				
				CPU	FX1S	FX1N	FX2N	FX2NC
							●	●
Операнды	S+	D+	n1, n2	Имп. инструкция (P)	Обработка		Шаги программы	
	KnX,KnY,KnM, KnS,T,C,D	KnY,KnM,KnS, T,C,D	К, Н n2mn1m512	●	16 Bit	32 Bit	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

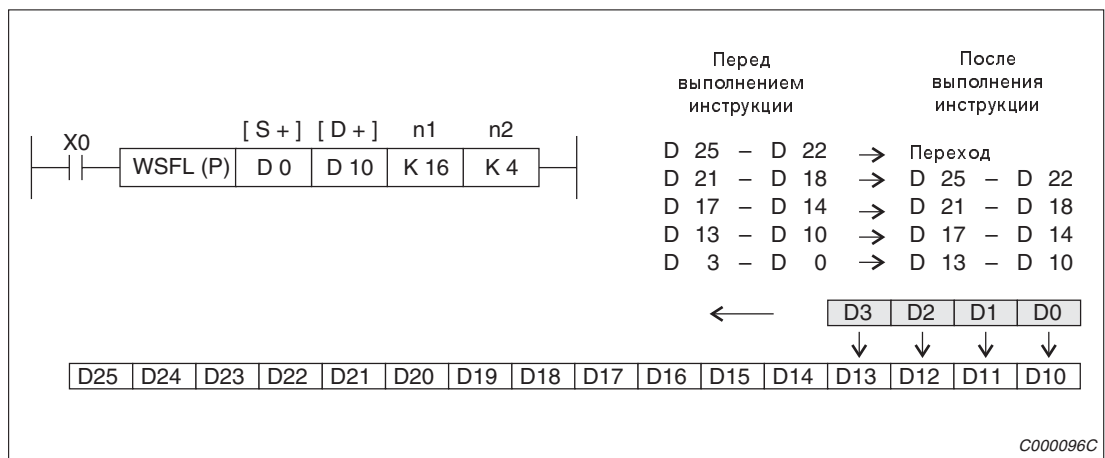



Рис. 6-65. Пример программирования сдвига влево



6.5.8 ЗАПИСЬ В СТЕКОВУЮ ПАМЯТЬ ТИПА FIFO (SFWR)

				SFWR		FNC 38					
				Запись в стековую память типа FIFO							
				CPU	FX1S	FX1N	FX2N	FX2NC			
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 2mn512		●	16 Bit	32 Bit	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

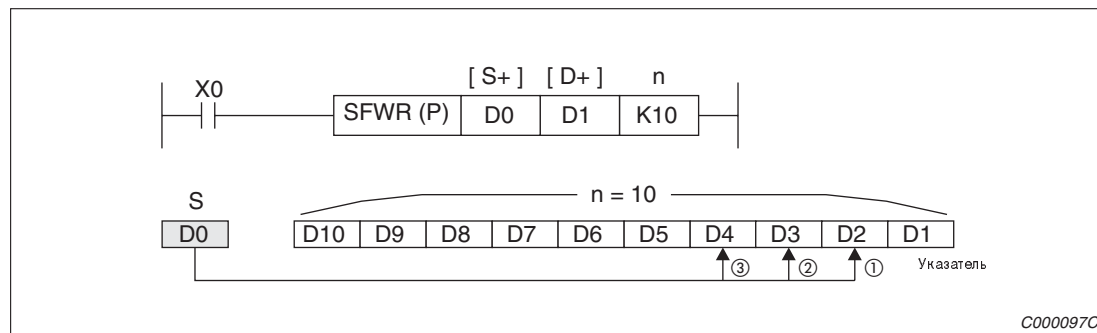


Рис. 6-66. Пример программирования записи в стек FIFO

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		
				●	●	●	●		
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	●	16 Bit	32 Bit	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

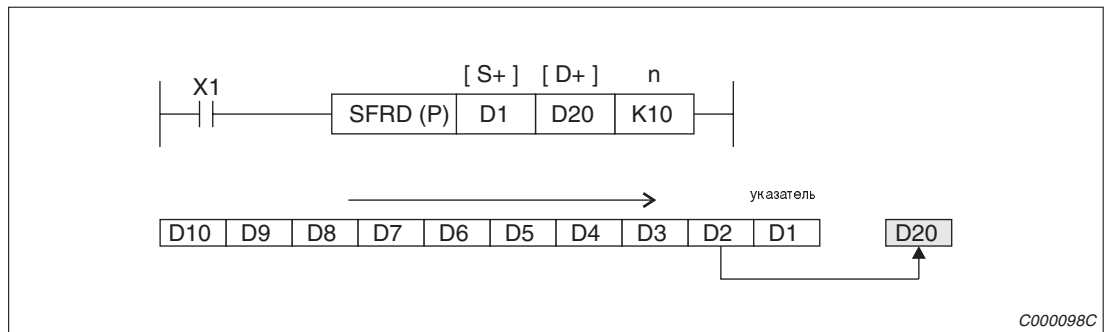


Рис. 6-67. Пример программирования чтения из стека FIFO

Пример использования см. страницу 6-70.



ПРИМЕР ▾

Программирование стекового накопителя FIFO

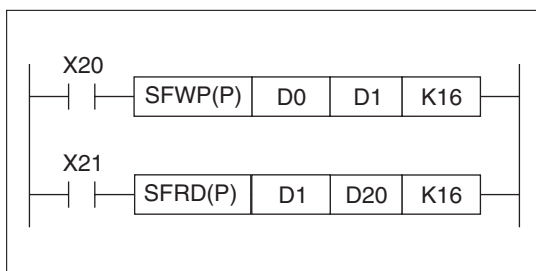


Рис. 6-68.
Пример программирования записи и чтения стека FIFO

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
			●	●	●	●
Операнды	D1+, D2+	Имп. инструкция (P)	Обработка		Шаги программы	
	Y, M, S, T, C, D	●	16 Bit	32 Bit	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

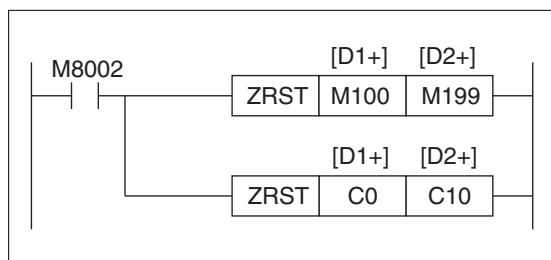


Рис. 6-70.
Пример программирования ZRST-инструкции

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	
					●	●	●	●	
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	●	16 Bit	32 Bit	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)

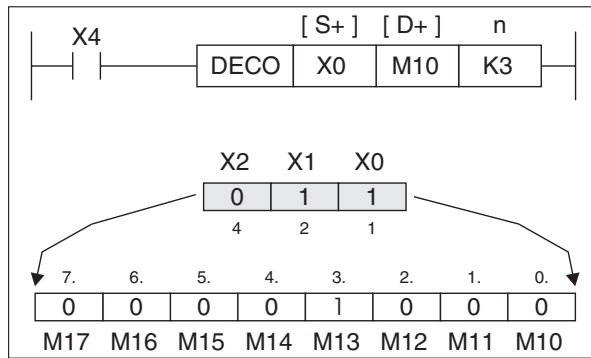


Рис. 6-71.

Пример программирования DECO-инструкции с указанием битовых операндов в D+

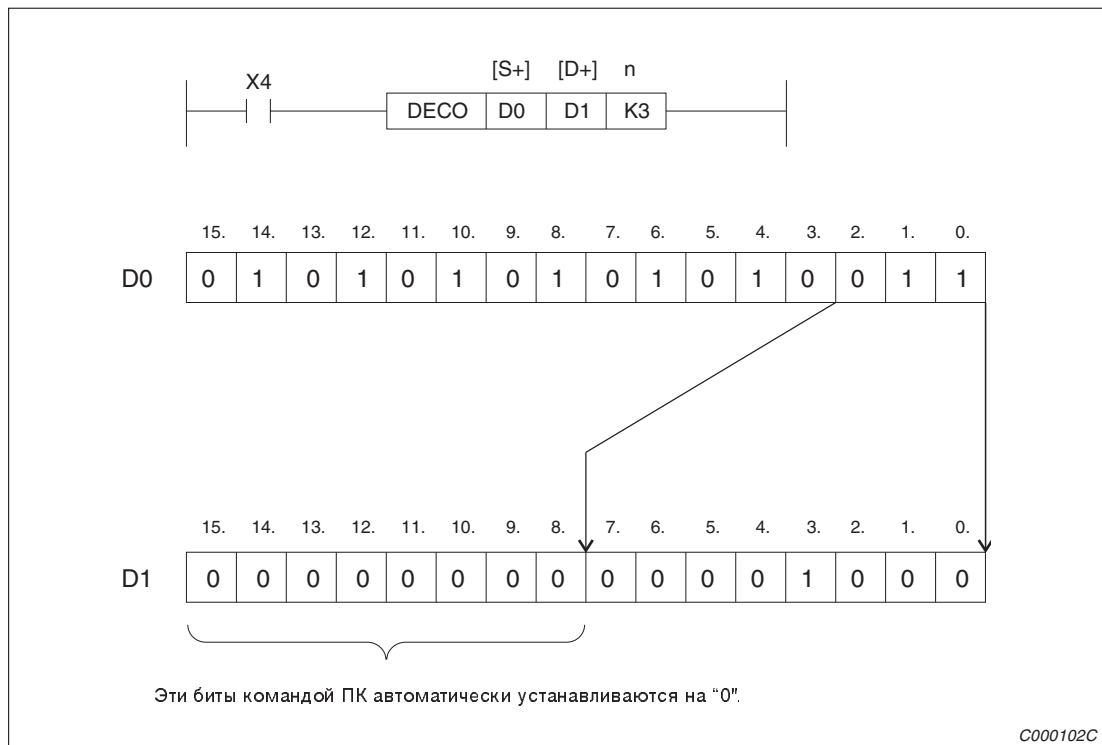
C000101C

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	
					●	●	●	●	
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		●	16 Bit	32 Bit	ENCO/ ENCOP	7

ФУНКЦИЯ

Кодирование данных

ОПИСАНИЕ

Данные в 2ⁿ операндов, начиная со стартового адреса, указанного в (S+), кодируются. В (D+) определяется операнд цели, куда записывается результат кодировки.

2ⁿ: Количество операндов, данные которых должны кодироваться.

n: Число операндов цели

При указании битового операнда в S+ должно соблюдаться: (1 ≤ n ≤ 8)

При указании словного операнда в S+ должно соблюдаться: (1 ≤ n ≤ 4)

(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+) ($1 \leq n \leq 8$)

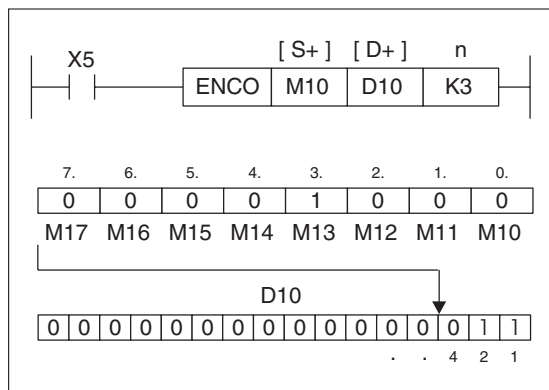


Рис. 6-73.

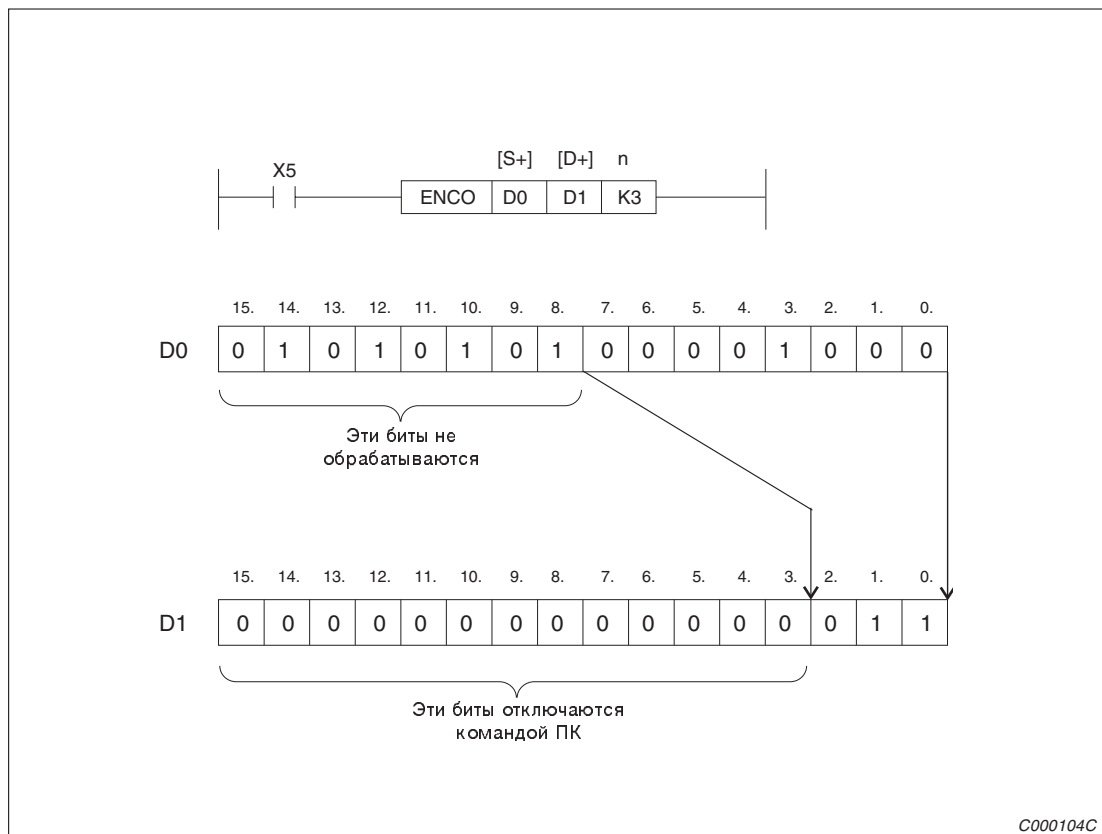
Пример программирования ENCO-инструкции с указанием битовых операндов в (S+)

C000103C

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. ▴

ПРИМЕР ▾

Применение ENCO-инструкции с указанием словных операндов в S+ ($1 \leq n \leq 4$)



C000104C

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. ▴

6.6.4 СУММА АКТИВНЫХ БИТ (SUM)

		SUM		FNC 43			
		Определение используемых битов					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
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	●	16 Bit	32 Bit	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

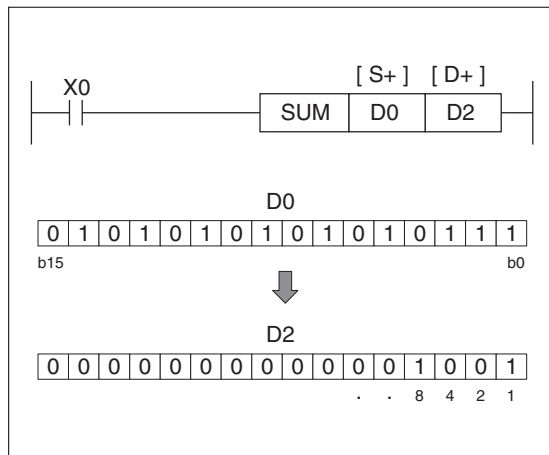


Рис. 6-75.

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	
							●	●	
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	●	16 Bit	32 Bit	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

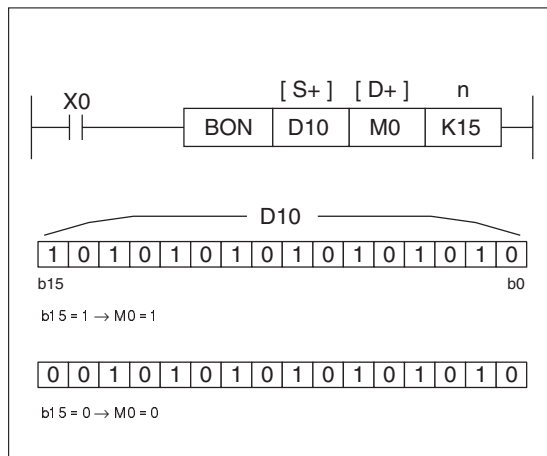


Рис. 6-76.

Пример программирования для проверки бита

C000142C



6.6.6 ВЫЧИСЛЕНИЕ СРЕДНЕГО ЗНАЧЕНИЯ (MEAN)

				MEAN		FNC 45				
				Вычисление среднего значения						
				CPU	FX1S	FX1N	FX2N	FX2NC		
							●	●		
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)	●	16 Bit	32 Bit	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

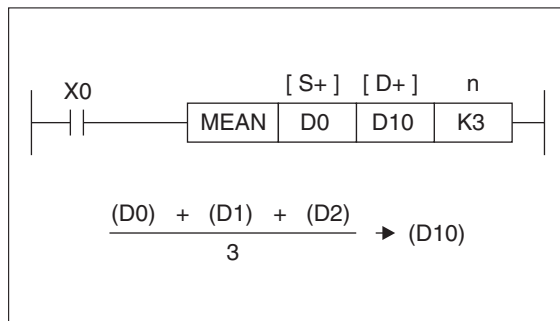


Рис. 6-77.

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		
							●	●		
Операнды	S+	D+	m	Имп. инструкция (P)	Обработка		Шаги программы			
	T T0 ... T199	S S900 ... S999	K (1 ... 32767)		16 Bit	32 Bit	ANS		7	
					●					

ФУНКЦИЯ

Запуск интервалов времени и включение бита отображения

ОПИСАНИЕ

- Запуск интервалов времени и включение бита отображения.
- При выполнении этой инструкции запускается время $m \times 100$ 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

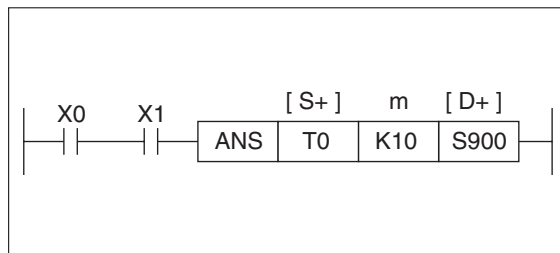
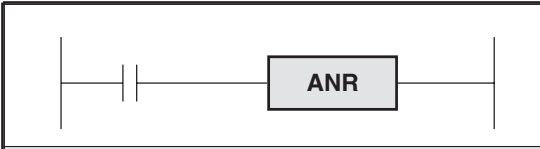


Рис. 6-78.
Пример программирования для запуска временного интервала

C000144C



6.6.8 ОТКЛЮЧЕНИЕ БИТА ОТОБРАЖЕНИЯ (ANR)

		ANR		FNC 47		
		Отключение бита отображения				
		CPU	FX1S	FX1N	FX2N	FX2NC
					●	●
Операнды		—	Имп. инструкция (P)		Шаги программы	
		-	●	16 Bit	32 Bit	ANR/ANRP
			●			

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

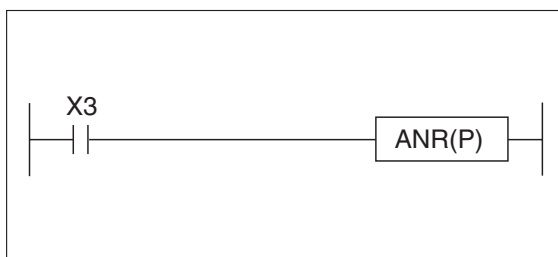


Рис. 6-79.
Пример программирования отключения бита отображения

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							
Операнды		S+	D+	Имп. инструкция (P)		Обработка		Шаги программы	
				K, H, D	D	●	16 Bit	32 Bit	SQR/SQRP
●	●	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

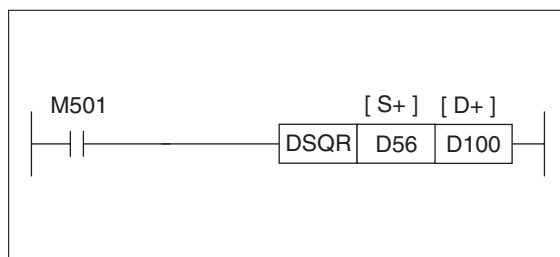


Рис. 6-80. Пример программирования расчета корня квадратного

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						
Операнды		S+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
		D	D		●	16 Bit	32 Bit	FLT/FLTP
					●	●	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

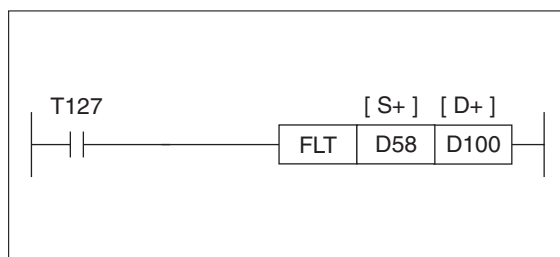


Рис. 6-81.

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	
				●	●	●	●	
Операнды	D	n	Имп. инструкция (P)	Обработка		Шаги программы		
	X, Y ^①	K, H ^②	●	16 Bit	32 Bit	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

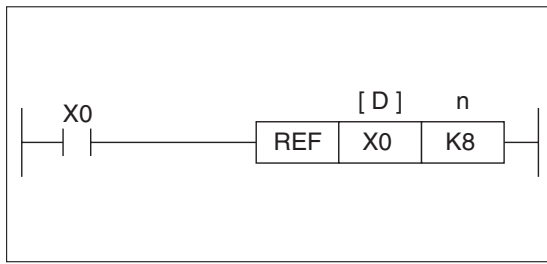


Рис. 6-82.
Пример программирования REF-инструкции; обновление входов

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 замедления входного сигнала может быть изменено в регистрах данных D8020 и D8021 (дальнейшие указания см. раздел 10.2.3).

ПРИМЕР ▾ Применение REF-инструкции, обновление входов

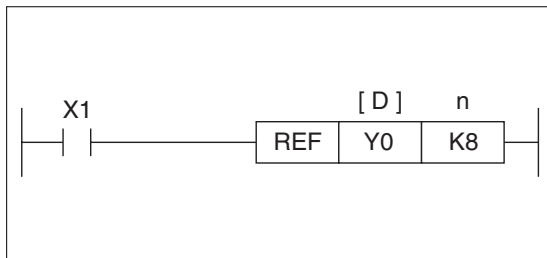


Рис. 6-83.
Пример программирования REF-инструкции; обновление входов

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 Ausgänge auffrischen					
Операнды		n K, H	Имп. инструкция (P)	Обработка		Шаги программы	
				●	●	●	●
				16 Bit	32 Bit	REF, REFP	5

ФУНКЦИЯ

Установка времени фильтра для входов X0 - X7

ОПИСАНИЕ

- REFF-инструкция устанавливает время фильтра равное (n)ms.
- (n) может задаваться от 0 до 60. Уставка 0 равно времени фильтра 50 микросекунд.

УКАЗАНИЕ

Инструкция должна быть включена в каждом цикле, иначе уставка фильтра устанавливается на значение по умолчанию 10ms.

ПРИМЕР ▾

Если активен вход X10, то уставка фильтра примет значение 1ms, при отключенном - 10ms. По команде "REFF K20" устанавливается задержка включения 20ms.

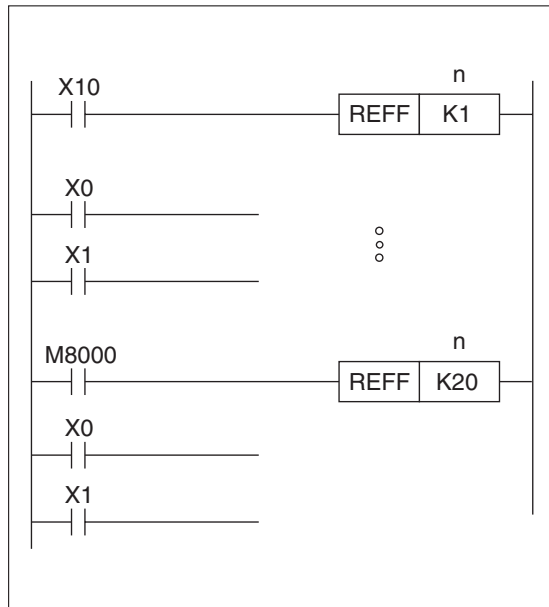


Рис. 6-84. Пример использования REFF-инструкции.

C000150C



6.7.3 ЧТЕНИЕ МАТРИЦЫ (MTR)

				MTR		FNC 52			
				Чтение матрицы					
				CPU	FX1S	FX1N	FX2N	FX2NC	
				●	●	●	●	●	
Операнды	S+, D1+	D2+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	H ^①	Y, M, S	K, H n = 2 ... 8		16 Bit	32 Bit	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.

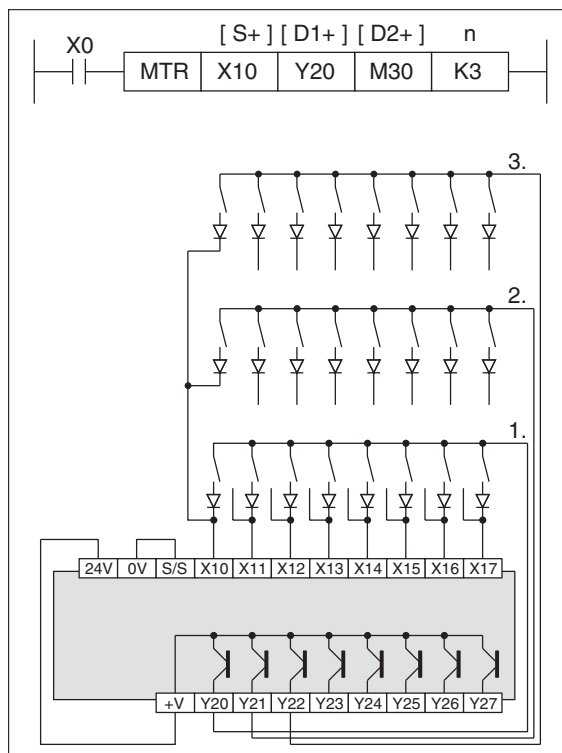


Рис. 6-85.

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.

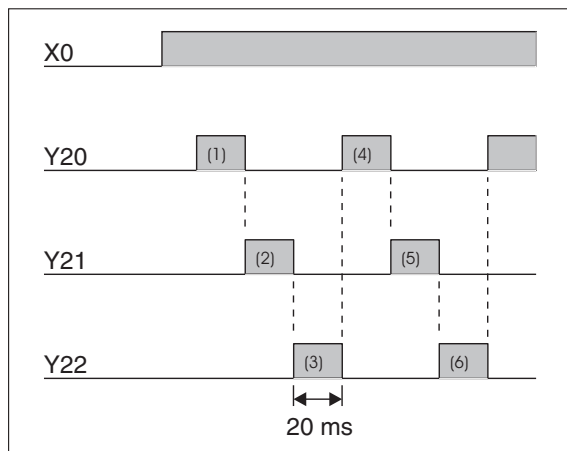


Рис. 6-86.
Пример программирования включения выходов

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

Для исключения конфликтной ситуации в программе в качестве входов по возможности не применять адреса X0...X7.

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.

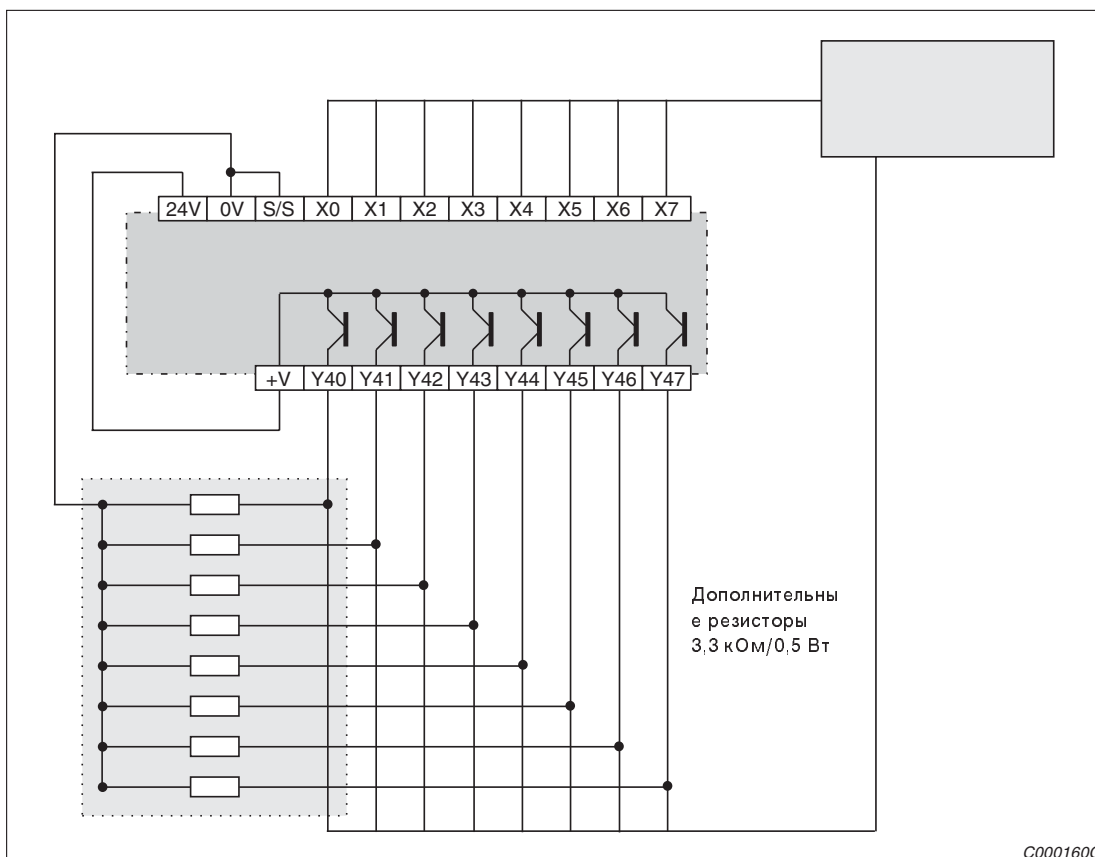


Рис. 6-87. Пример применения Pull-Down-резисторов



6.7.4 СИГНАЛ ВКЛЮЧЕНИЯ И ОТКЛЮЧЕНИЯ ОТ ВЫСОКОСКОРОСТНОГО СЧЕТЧИКА (DHSCS, DHSCR)

				DHSCS		FNC 53			
				Включение от высокоскоростного счетчика					
				CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	●	●	
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		16 Bit	32 Bit	DHSCS	13	
						●			

				DHSCR		FNC 54			
				Отключение от высокоскоростного счетчика					
				CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	●	●	
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		16 Bit	32 Bit	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

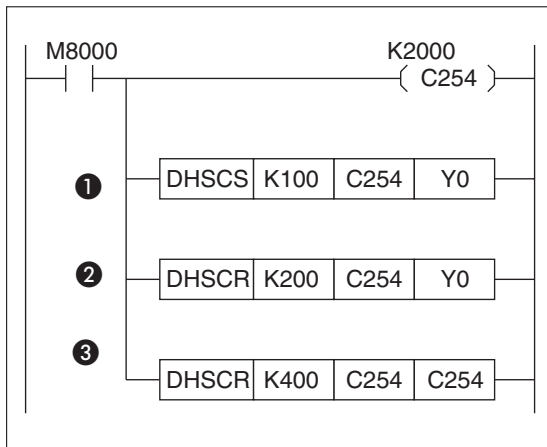


Рис. 6-88.
Пример программирования DHSCS-, DHSCR-инструкций

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

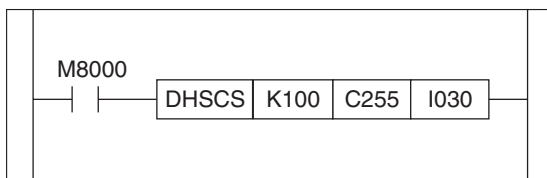


Рис. 6-89.
Пример программирования Счетчика-прерывания

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	
							●	●	
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		16 Bit	32 Bit	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 ▾

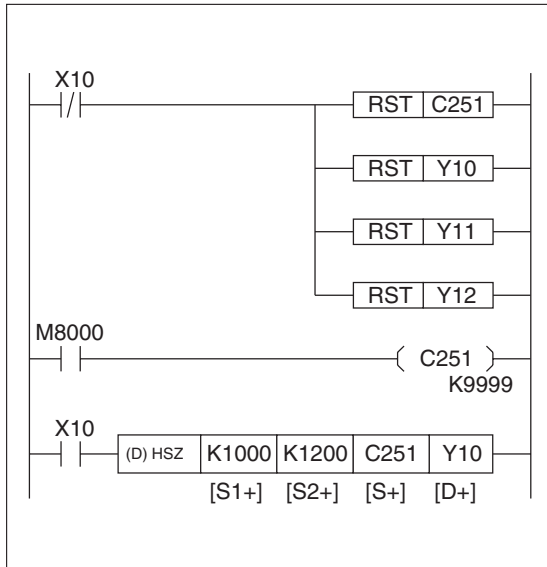


Рис. 6-90.
Пример программирования
DHSZ-инструкции

C000156C

- (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.

ПРИМЕР ▾

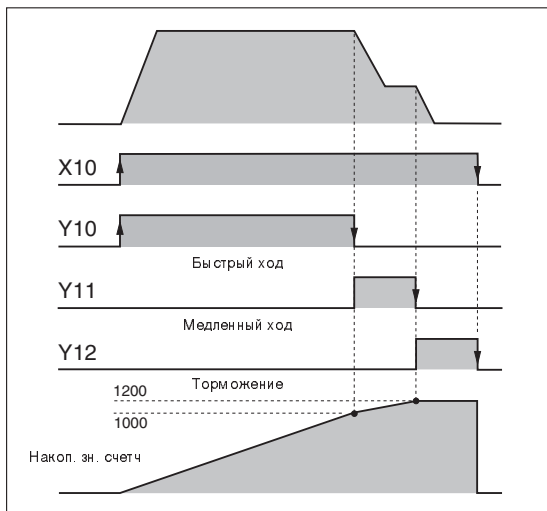


Рис. 6-91.
Временные характеристики включения
выходов Y10,Y11,Y12

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. △

FX2N/FX2NC **ПРИНЦИП ФУНКЦИОНИРОВАНИЯ СО СПЕЦИАЛЬНЫМ МЕРКЕРОМ M8130 (FX2N/FX2NC)**

Сравнение таблиц высокоскоростных счетчиков в областях от S1+ и следующих n1

ОПИСАНИЕ

- DHSZ-инструкция со специальным меркером M8130 сравнивает в режиме прерывания накопленное значение высокоскоростного счетчика со значениями, указанными в табличных областях.
- Операнд (D+) для этой специальной функции задается по специальному меркеру 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 меркера M8130.

Должна быть корректно введена HSC-инструкция.

DHSZ-инструкция со специальным меркером 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

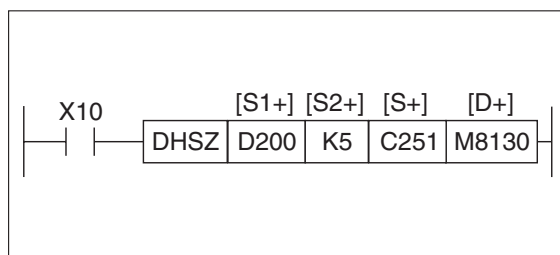


Рис. 6-92.
Пример программирования DHSZ-инструкции со специальным меркером M8130

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 ▾

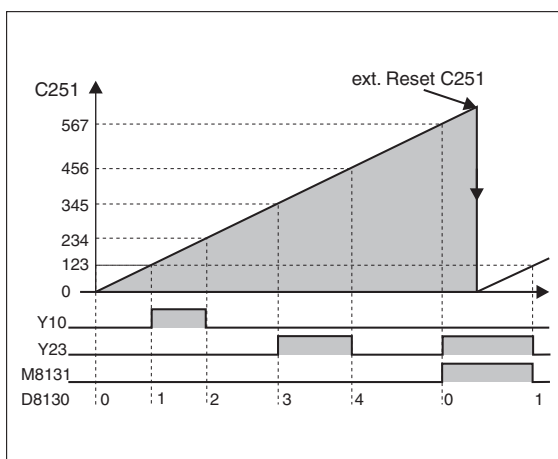


Рис. 6-93.
Временные процессы для включения выходов Y10, Y23

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.

△

FX2N/FX2NC **FX2N ПРИНЦИП ФУНКЦИОНИРОВАНИЯ СО СПЕЦИАЛЬНЫМ МЕРКЕРОМ M8132 (FX2N/FX2NC)**

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

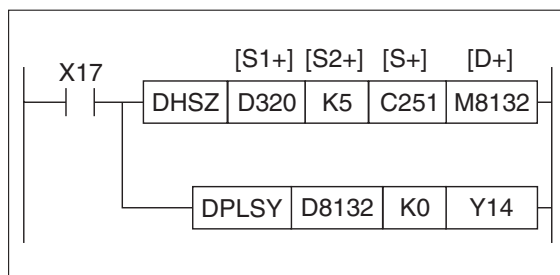


Рис. 6-94.
Пример программирования DHSZ-инструкции со специальным меркером M8132

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 ▾

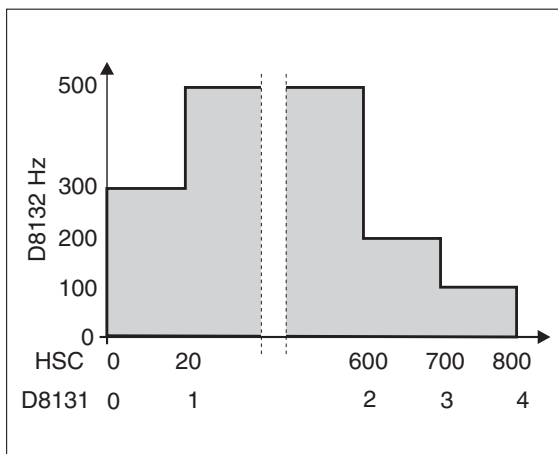


Рис. 6-95.
Изменение частоты при применении DHSZ-инструкции со специальным меркером M8132

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	
					●	●	●	●	
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		16 Bit	32 Bit	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 ▾

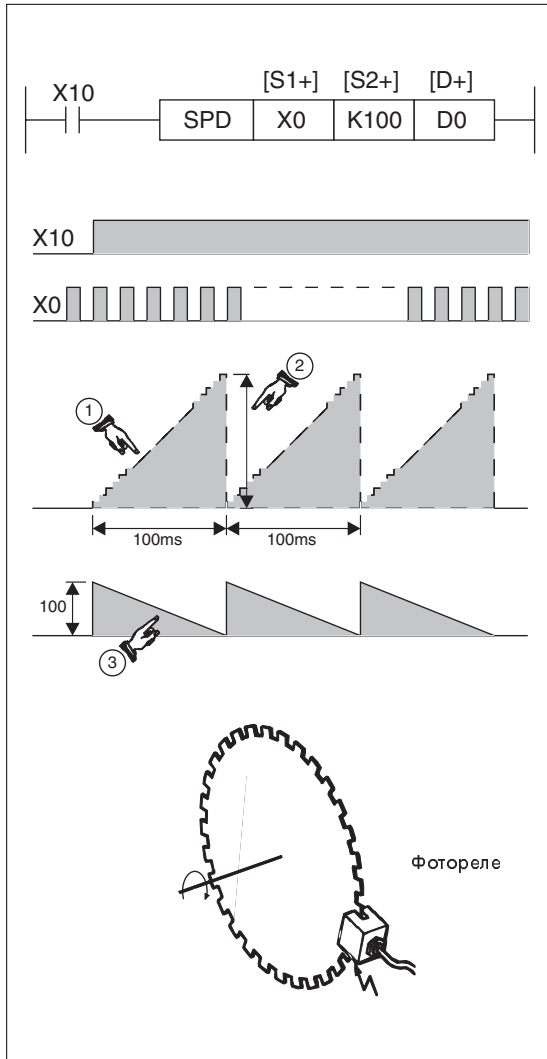


Рис. 6-96.
Пример программирования SPD-инструкции

C000158C

- 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+



6.7.7 ВЫДАЧА ОПРЕДЕЛЕННОГО ЧИСЛА ИМПУЛЬСОВ (PLSY, DPLSY)

		PLSY		FNC 57			
		Выдача определенного числа импульсов					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
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		16 Bit	32 Bit	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+) определяется частота: 1 ... 32767 Гц (FX1S/FX1N, PLSY)
1... 100 кГц(FX1S/FX1N, DPLSY)
2... 20 кГц(FX2N)
- 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 деактивизируется 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 стабильной работы выходов, на максимальных частотах, ток нагрузки для ПК серии FX2N составляет минимум 200mA, для ПК серии FX1S-1N - от 10 до 100mA.
- | Применение DHSZ-инструкции смотри в разделе Рис. 6-97.

ПРИМЕР ▾

Применение PLSY-инструкции формирования группы импульсов

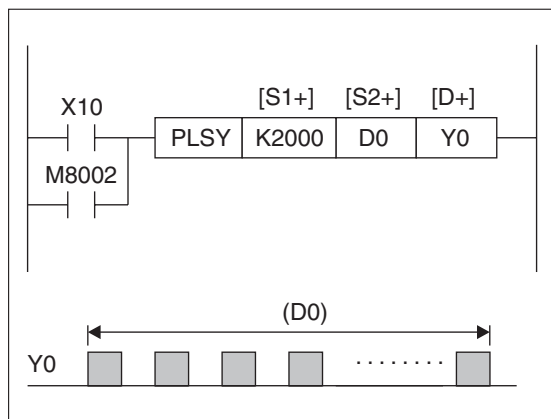


Рис. 6-97.
Пример программирования PLSY-инструкции

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 D8136 и D8137 в ПК серии FX2N заносится сумма импульсов.

6.7.8 ВЫДАЧА ИМПУЛЬСОВ С МОДУЛЯЦИЕЙ ШИРИНЫ ИМПУЛЬСА (PWM)

		PWM		FNC 58			
		Выдача импульсов с модуляцией ширины импульса					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
Операнды	S1+, S2+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z (S1 m S2)	Только Y0 или Y1		16 Bit	32 Bit	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 стабильной работы выходов, на максимальных частотах, ток нагрузки для ПК серии FX2N составляет минимум 200mA, для ПК серии FX1S-1N - от 10 до 100mA.

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

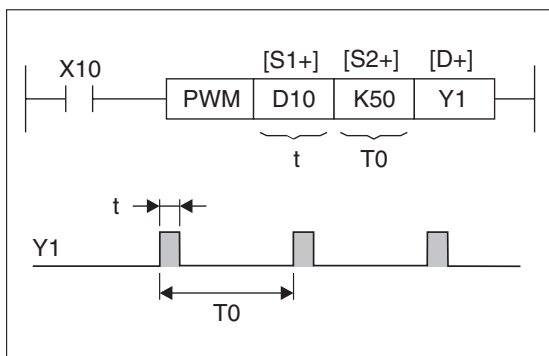


Рис. 6-98.
Пример программирования
PWM-инструкции

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	
			●	●	●	●	
Операнды	S1+, S2+, S3+	D+	Имп. инструкция (P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z (S1 m S2)	Только Y0 или Y1		16 Bit	32 Bit	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+).
- Выдаваемая частота в ПК серии FX2N может находиться в пределах между 10 и 22 000 Гц. Для ПК серий FX1S или FX1N частота может быть в пределах от 10 до 100 000 Гц. Указанная частота должна делиться на 10. Если задаваемая частота не делится на 10, она округляется во соответствующего значения.
- Ширина шага наклонной составляет 1/10 указанной выходной частоты (при применении шагового двигателя это нужно учитывать).
- Максимальное количество импульсов:
16-ти битовые инструкции: от 110 до 32.767 импульсов
32-х битных инструкций: от 110 до 2.147.483.647 импульсов
При задании менее 110 импульсов правильная работа не гарантируется.
- 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 ▾

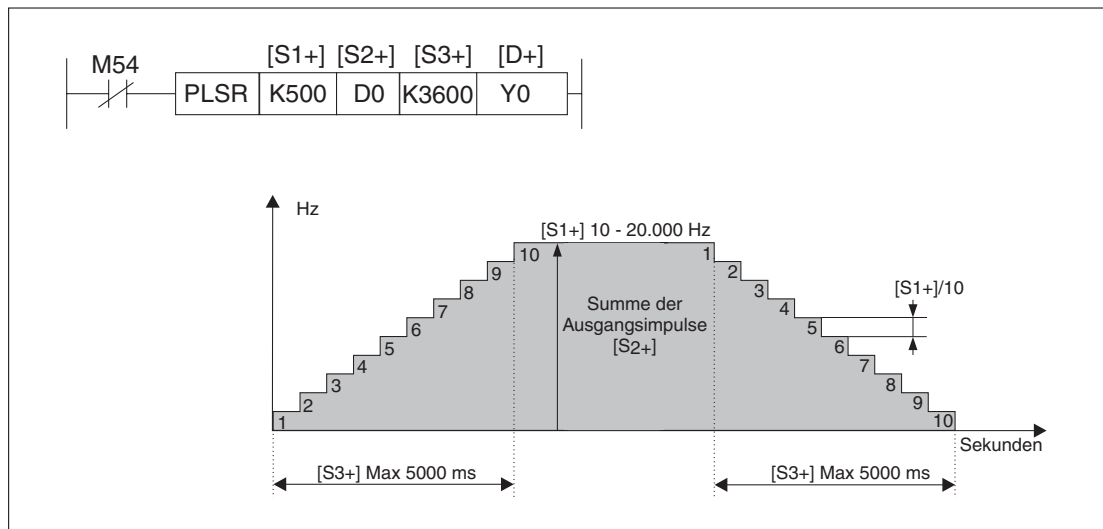


Рис. 6-99. Пример программирования PLSR-инструкции

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									
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+)				16 Bit 32 Bit		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).
- В управлении используются специальные меркеры M8040...M8042 и M8047.
- 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

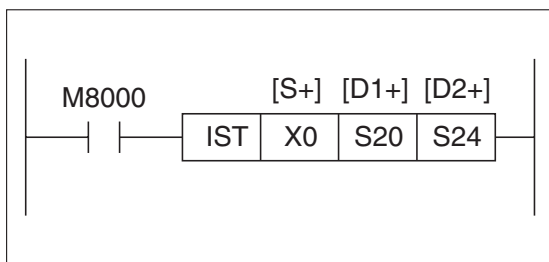


Рис. 6-100.

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: разрешено последующее включение других шагов
- M8041: начало последующего включения
- M8042: импульс запуска
- M8043: нулевая точка достигнута
- M8044: нулевая точка определена

ПРИМЕР ▾

Передвижной контейнер для транспортировки сыпучих материалов в автоматическом режиме должен непрерывно загружаться и разгружаться.

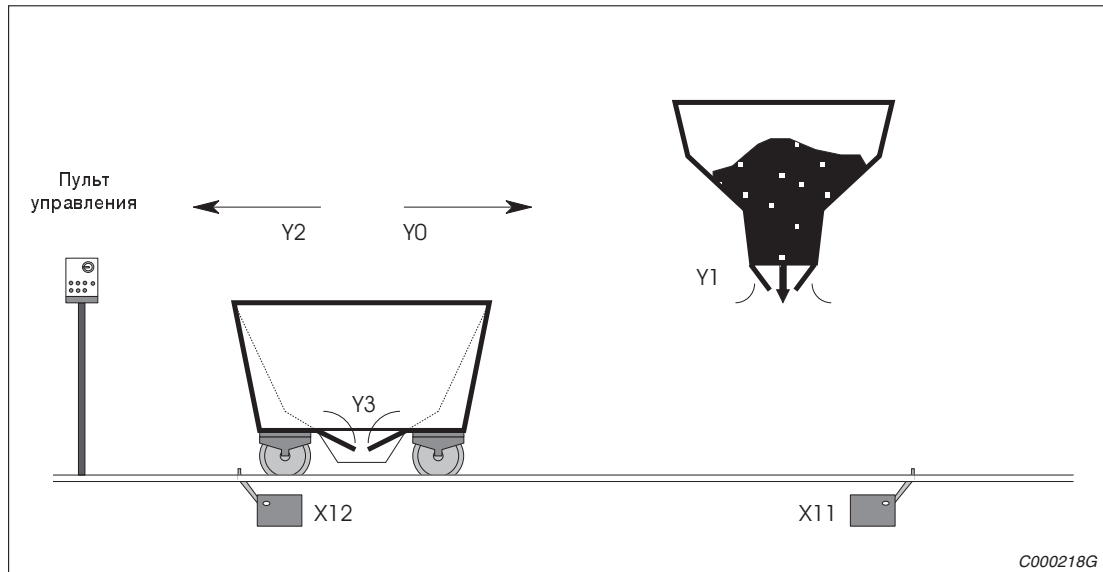


Рис. 6-101. Пример, контроль загрузки и разгрузки контейнера с помощью IST-инструкции

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

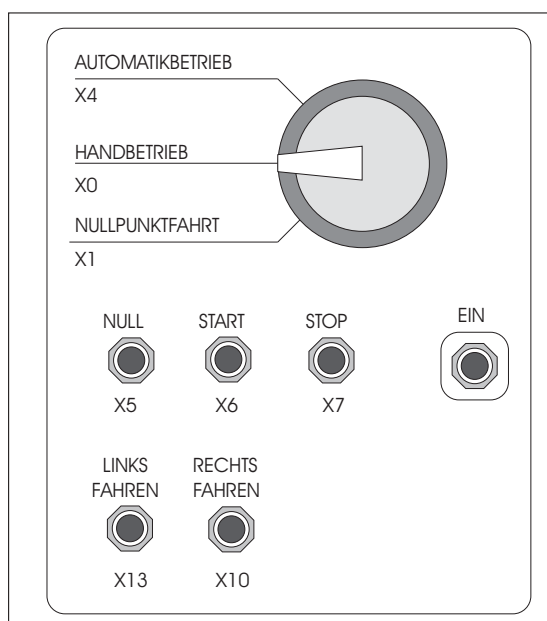


Рис. 6-102.
Пример пульта управления

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 ▾

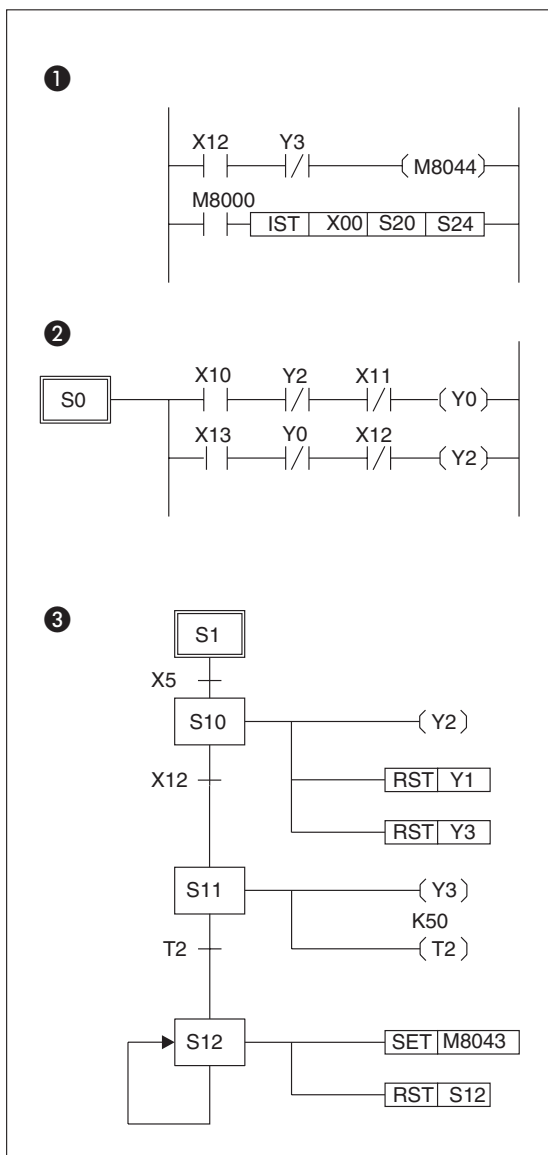


Рис. 6-103.

Пример программирования контроля загрузки и разгрузки перемещаемого контейнера по IST-инструкции

C000213G

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.

ПРИМЕР ▾

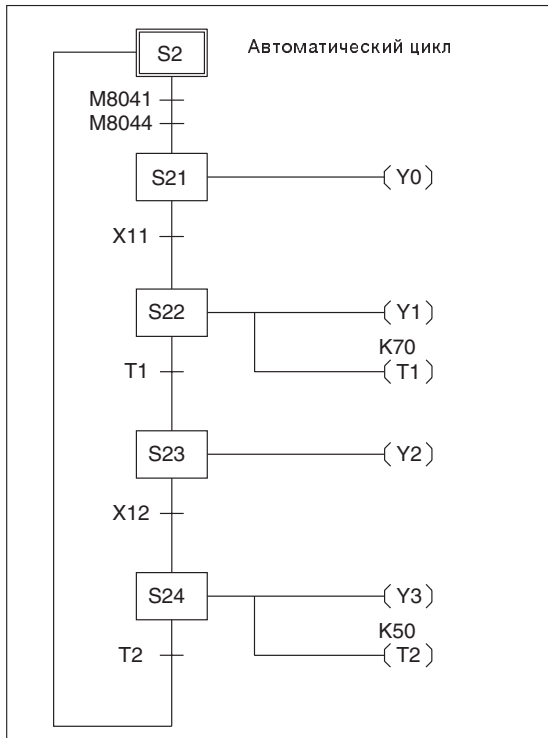


Рис. 6-104.

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 X 0	RST Y 1	STL S 21	LD X 13
S 20	RST Y 3	OUT Y 0	ANI Y 0
S 24	OUT Y 2	LD X 11	OUT Y 2
	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	
								●	●	
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	●	16 Bit	32 Bit	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

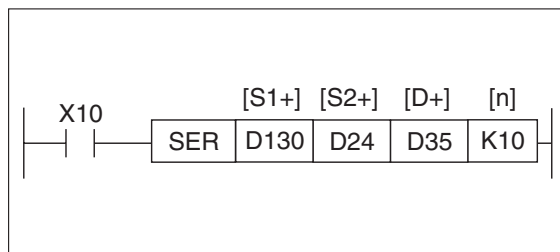


Рис. 6-106.
Пример программирования
SER-инструкции

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	
						●	●	●	●	
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		16 Bit	32 Bit	ABSD	9	
						●	●	DABSD	17	

- ① X, Y, M, S должны быть кратными 8.
- ② Недопустимы высокоскоростные счетчики

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+) + 1	(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 из словных операндов

- Если счетчик (S2+) получил накопленное значение, занесенное в таблицу, включается относящийся к нему операнд.
- Записанное в таблицу значение должно находиться в области счета от 0 до 32 767.
- Значение может, например, быть записанным с помощью MOV-инструкции.

УКАЗАНИЕ

- | Всегда записываются два счетчика, (S2+) и ((S2+)+1). В (S1+) должен применяться четный адрес операнда.
- | ABSD-инструкция может использоваться в программе только один раз.

ПРИМЕР ▾

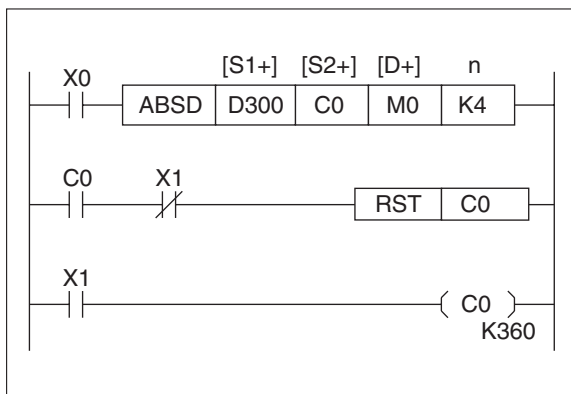


Рис. 6-108.
Пример программирования
ABSD-инструкции

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 включено X0, то меркеры M0...M3 включаются согласно графикам на Рис. 6-112. Если X0 выключен, меркеры не включаются.

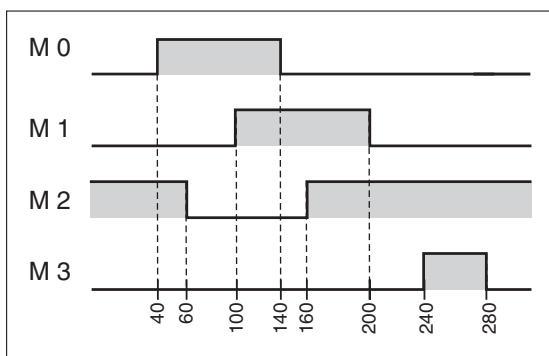


Рис. 6-109.
Временная характеристика
включения/отключения меркеров

C000202C



6.8.4 ИНКРЕМЕНТАЛЬНЫЙ МНОГОУСТАВОЧНЫЙ СЧЕТЧИК (INCD)

					INCD		FNC 63		
					Инкрементальный многоуставочный счетчик				
					CPU	FX1S	FX1N	FX2N	FX2NC
						●	●	●	●
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			16 Bit	32 Bit	INCD
						●			

- 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.

ПРИМЕР ▾

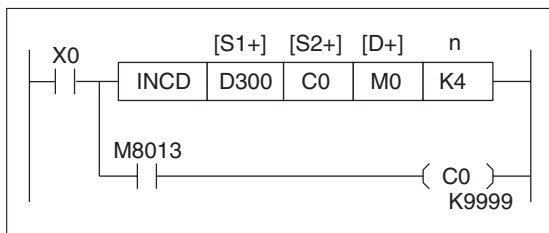


Рис. 6-110.
Пример программирования INCD-инструкции

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

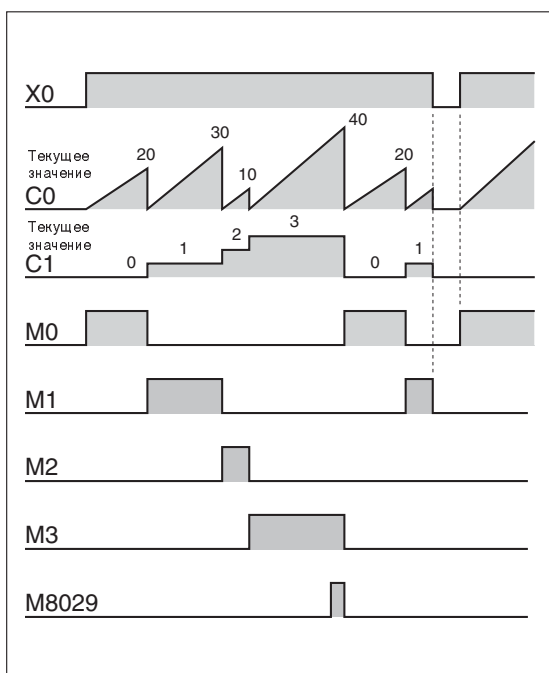


Рис. 6-111.
Временная характеристика соответствует рассмотренному выше примеру включения и отключения меркеров

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 счета. 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
					●	●
Операнды	D	n	Имп. инструкция (P)	Обработка		Шаги программы
	D	K, H n = 0 - 2		16 Bit	32 Bit	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).

ПРИМЕР ▾

Пример формирования времен

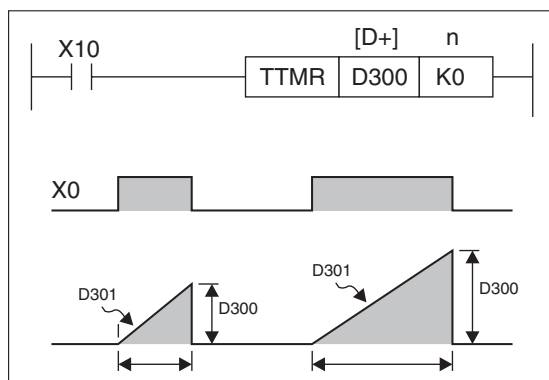


Рис. 6-112.
Пример программирования TTMR-инструкции

При включении X0 измеряется продолжительность воздействия.

D300: Время воздействия в секундах, умноженное на 1

D301: Время воздействия в секундах

C000167C



6.8.6 СПЕЦИАЛЬНЫЙ ТАЙМЕР (STMR)

				STMR		FNC 65			
				Специальный таймер					
				CPU	FX1S	FX1N	FX2N	FX2NC	
							●	●	
Операнды	S+	D+	n	Имп. инструкция (P)	Обработка		Шаги программы		
	T0 - T199	Y, M, S	K, H n=1 - 32767		16 Bit	32 Bit	STMR	7	
					●				

ФУНКЦИЯ

Генерация функции специального таймера

ОПИСАНИЕ

- С помощью STMR-инструкции могут управляться функции специального таймера
 - замедление отключения
 - таймер с отклонением времени
 - датчик мигания
- Таймер в (S+) (с уставкой n) управляет битовым операндом (D+) следующим образом:

- (D+): замедление отключения
- ((D+)+1): таймер с отклонением времени
- ((D+)+2) / ((D+)+3): датчик мигания

Таким образом, начиная с (D+), задействуются четыре последовательных бита.

УКАЗАНИЕ

| Таймер, используемый в этой инструкции, не может применяться еще раз в другом месте.

ПРИМЕР ▾

Применение STMR-инструкции

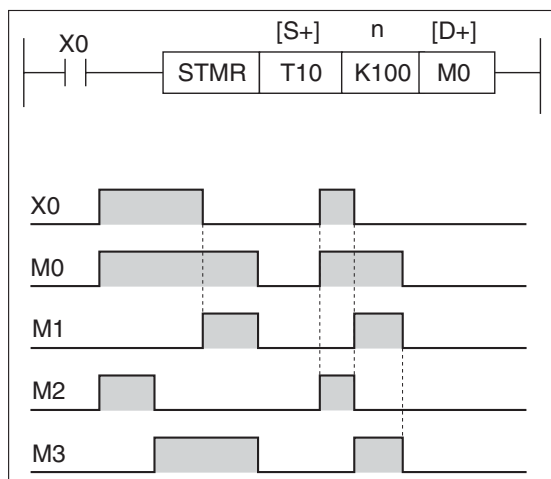


Рис. 6-113.
Пример программирования генерации функции специального таймера

C000161C



6.8.7 ТРИГГЕРНАЯ ФУНКЦИЯ (FLIP-FLOP) (ALT)

		ALT		FNC 66			
		Триггерная функция					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
Операнды	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	Y, M, S	●	16 Bit	32 Bit	ALT/ALTP	3	

ФУНКЦИЯ

Реализация триггерной функции

ОПИСАНИЕ

- С помощью ALT-инструкции может программироваться триггерная функция.
- ALT-инструкция активизируется по управлению сигналом "1" и записывается в операнд, указанный в D+.
- Новым управлением сигнала в "1" операнд отключается.

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

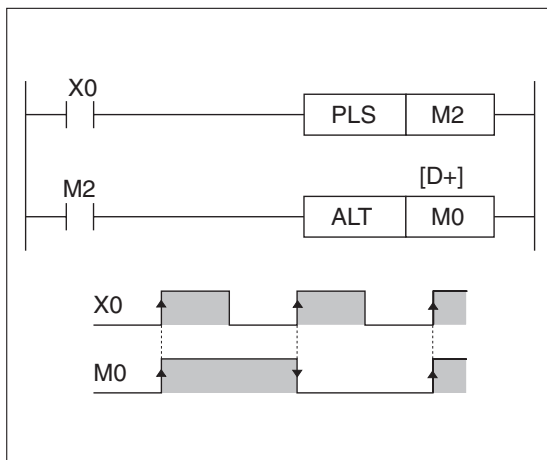


Рис. 6-114.

Пример программирования ALT-инструкции (инвертирование статуса операнда)

C000108C

Состояние меркера M0 каждый раз инвертируется, если на входе устанавливается сигнал "1".



ПРИМЕР ▾

Применение ALT-инструкции. Инвертирование статуса операнда

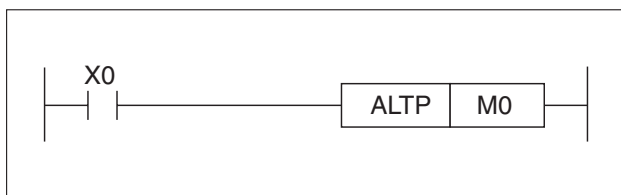


Рис. 6-115.
Пример программирования ALT-инструкции (инвертирование статуса операнда)

Функция идентична, приведенной на рисунке 6-114.



ПРИМЕР ▾

Применение ALT-инструкции. Старт-стоп-функция

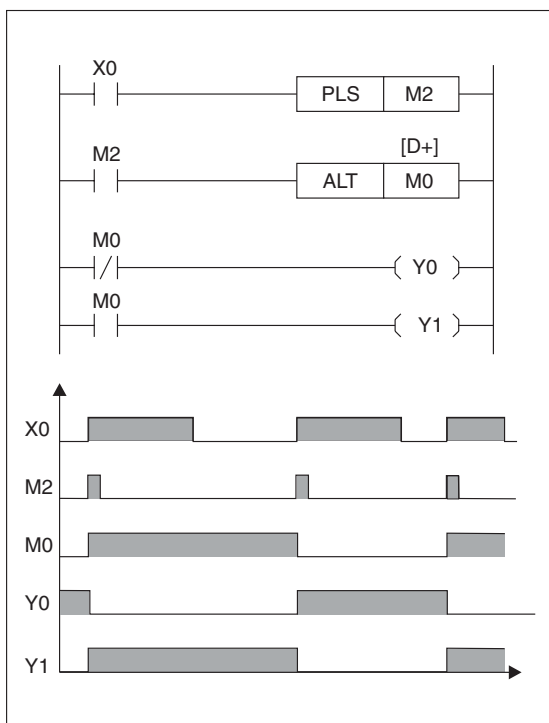


Рис. 6-116.
Пример программирования ALT-инструкции (старт-стоп- функция)

Старт-выход Y1 активизируется нажатием кнопки X0. Стоп-выход Y0 активизируется повторным нажатием кнопки X0.



C000114C

6.8.8 ФУНКЦИЯ НАКЛОННЫХ ХАРАКТЕРИСТИК (RAMP)

		RAMP		FNC 67			
		Рамповая функция					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
Операнды	S1+, S2+, D+	n	Имп. инструкция (P)	Обработка		Шаги программы	
	D	K, H n=1 ... +32 767		16 Bit	32 Bit	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.

У FX2N-CPU можно повлиять на процесс RAMP-функции с помощью специального меркера M8026. Если M8026 не введен в работу, произойдет постоянное повторение RAMP-функции. Это означает, если накопленное значение D+ равно значению в S2+, то RAMP-функция автоматически отключится и снова запустится. При включенном специальном меркере M8026 RAMP-функция сохраняется. Это означает, что до тех пор пока накопленное значение в D+ будет соответствовать значению в S2+ RAMP-функция сохранит мгновенное состояние. M8029 в этом случае остается включенным до тех пор, пока активна RAMP-функция. Значение в D+ не отключится до тех пор, пока не появится деинсталляция инструкции.

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

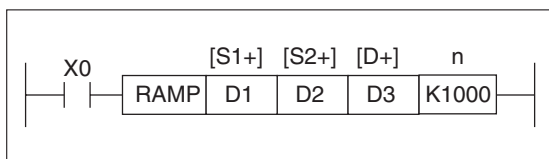
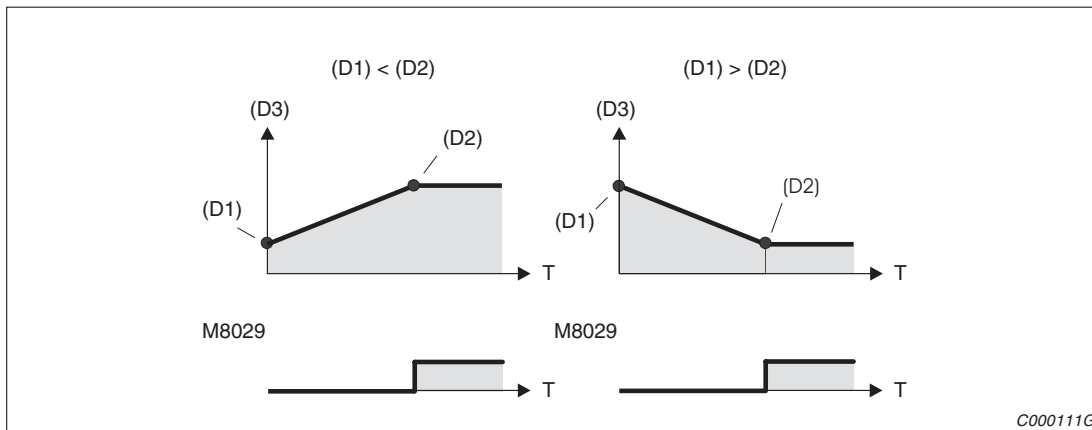


Рис. 6-117.
Пример программирования RAMP-инструкции

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 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, то 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	
Операнды	S+	m1 / m2	D+	Имп. инструкция (P)	Обработка		Шаги программы		
	D ①	K, H ②	Y, M, S ③	●	16 Bit	32 Bit	ROTC	9	

① 3 следующих друг за другом списка данных $((S+1) \leq m1)$

② $m1 =$ от 2 до 32 767; $m2 =$ от 0 до 32 767 ($m1 \leq m2$)

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

ROTC-инструкцией включаются или опрашиваются следующие операнды:

((D+)+0)	A-фазы сигнал счетчика
((D+)+1)	B-фазы сигнал счетчика
((D+)+2)	Нулевой-фазы сигнал счетчика
((D+)+3)	Выход: поворот стола вперед быстро
((D+)+4)	Выход: поворот стола вперед медленно
((D+)+5)	Стоп-Выход
((D+)+6)	Выход: поворот стола назад медленно
((D+)+7)	Выход: поворот стола назад быстро
	m1 Количество импульсов счета на один оборот стола
	m2 Количество импульсов счета на расстоянии, проходящем при медленном перемещении назад
((S+)+0)	Действительная позиция стола (может только считываться)
((S+)+1)	Позиция цели
((S+)+2)	Опознавание детали перемещаемой детали

ПРИМЕР ▾

Прием (фиксация) импульсов счета

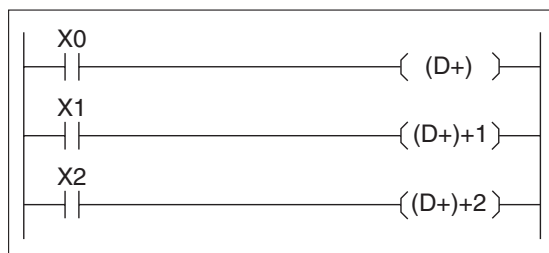


Рис. 6-119.
Пример программирования для определения сигналов

C000238C

Так как датчик и переключатель для нулевого сигнала может восприниматься по входам, то необходимо, записывать эти входы в операндах (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".

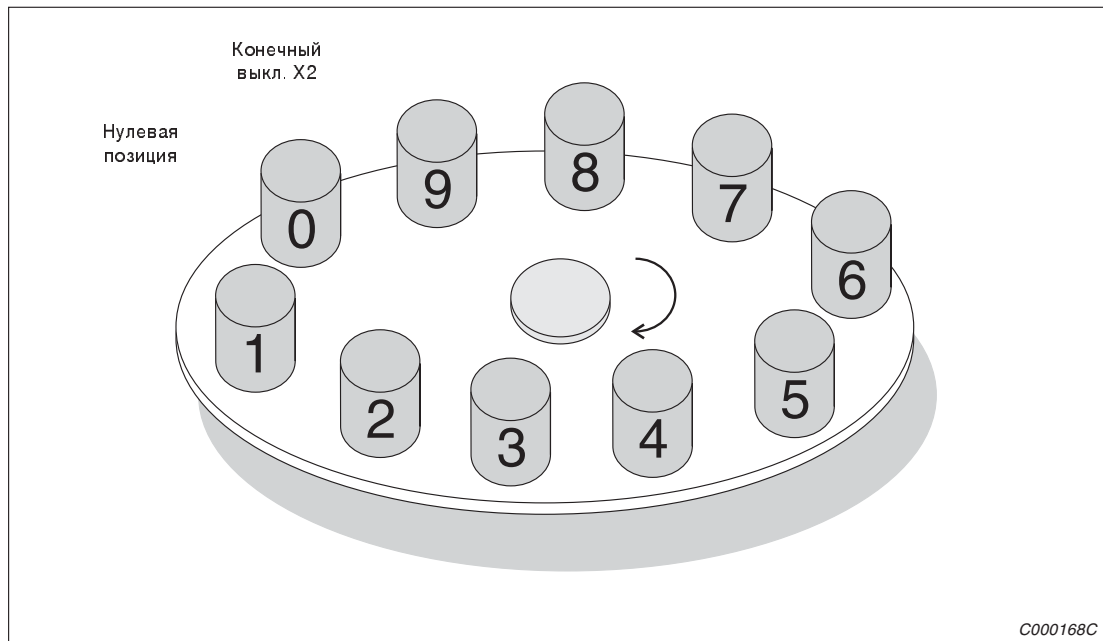


Рис. 6-120. Пример позиционирования поворотного стола

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) = $3 \times 50 = 150$: удаление позиции в импульсах от позиции 0
 Значение должно быть записано в D201 перед выполнением инструкции.
 ((S+)+2) = $7 \times 50 = 350$: 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 ▾

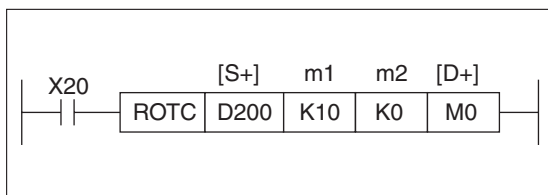


Рис. 6-121.
Пример программирования
ROTC-инструкции

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	
								●	●	
Операнды	S+	n1 / n2	D+	m	Имп. инструкция (P)	Обработка		Шаги программы		
	D ①	K, H ②	D	K, H, D		16 Bit	32 Bit	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 SORT-инструкции внутренняя матрица данных, характеризуемая стартовым регистром данных (S+) с величиной строк n1 и столбцов n2, сортируется по значениям в столбце m и записывается с новой сортировкой в регистр 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.

ПРИМЕР ▾

Применение SORT-инструкции на матрице

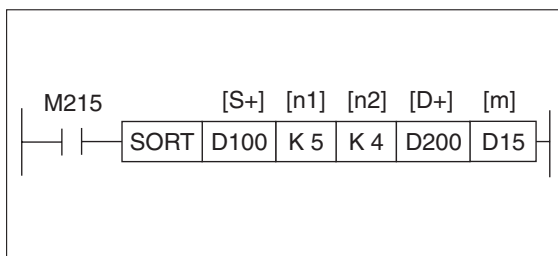


Рис. 6-122.
Пример программирования SORT-инструкции

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 FNC-ИНСТРУКЦИИ

7.1 ОБЩИЕ УКАЗАНИЯ

Эта глава описывает специальные FNC-инструкции семейства FX для особых случаев применения. Эти инструкции позволяют реализовать функции ввода и вывода данных, коммуникации модулей и управления специальными модулями.

Подробное пояснение структуры таблиц инструкций содержится в разделе 6.1.1.

7.1.1 ОБЗОР СПЕЦИАЛЬНЫХ FNC-ИНСТРУКЦИЙ

Классификация	Инструкция	FNC	Назначение	Раздел	Серия типов ПК			
					FX1S	FX1N	FX2N	FX2NC
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			●	●
	FROM	78	Выдача данных по выходам	7.2.9		●	●	●
	TO	79	Запись данных в специальные модули	7.2.10		●	●	●
Инструкции последовательной коммуникации	RS	80	Последовательная передача данных	7.3.1	●	●	●	●
	PRUN	81	Перезапись входов или меркеров	7.3.2	●	●	●	●
	ASCI	82	Перезапись входов или меркеров	7.3.3	●	●	●	●
	HEX	83	Преобразование в шестнадцатеричное значение	7.3.4	●	●	●	●
	CCD	84	Проверка на сумму и паритет	7.3.5	●	●	●	●
	VRRD	85	Чтение заданного значения из FX1N-8AV-BD и FX2N-8AV-BD	7.3.6	●	●	●	●
	VRSC	86	Чтение положения переключателя FX1N-8AV-BD и FX2N-8AV-BD	7.3.7	●	●	●	●
	PID	88	Программирование контуров регулирования	7.3.8	●	●	●	●

Табл. 7-1. Обзор специальных FNC-инструкций

Классификация	Инструкция	FNC	Назначение	Раздел	Серия типов ПК			
					FX1S	FX1N	FX2N	FX2NC
Инструкции чисел с плавающей запятой	DECMP	110	Сравнение чисел с плавающей запятой	7.4.1			●	●
	DEZCP	111	Сравнение чисел с плавающей запятой в области	7.4.2			●	●
	DEBCD	118	Преобразование формата с плавающей запятой в научный формат чисел	7.4.3			●	●
	DEBIN	119	Преобразование научного формата чисел в формат с плавающей запятой	7.4.4			●	●
	DEADD	120	Сложение чисел с плавающей запятой	7.4.5			●	●
	DESUB	121	Вычитание чисел с плавающей запятой	7.4.6			●	●
	DEMUL	122	Умножение чисел с плавающей запятой	7.4.7			●	●
	DEDIV	123	Деление чисел с плавающей запятой	7.4.8			●	●
	DESQR	127	Квадратный корень из числа с плавающей запятой	7.4.9			●	●
	INT	129	Преобразование формата с плавающей запятой в десятичный формат	7.4.10			●	●
	DSIN	130	Расчет синуса числа с плавающей запятой	7.4.11			●	●
	DCOS	131	Расчет косинуса числа с плавающей запятой	7.4.12			●	●
	DTAN	132	Расчет тангенса числа с плавающей запятой	7.4.13			●	●
Инструкции обработки данных	SWAP	147	Обмен старшим и младшим байтами	7.5.1			●	●
Позиционирование с помощью сервоусилителя	ABS	155	Чтение абсолютного действительного значения позиции	7.6.1	●	●		
	ZRN	156	Перемещение в исходную точку	7.6.2	●	●		
	PLSV	157	Выдача импульсов с переменной частотой	7.6.3	●	●		
	DRVI	158	Позиционирование по приращиваемому значению	7.6.4	●	●		
	DRVA	159	Позиционирование по абсолютному значению	7.6.5	●	●		
Инструкции реального времени	TCMP	160	Сравнение данных часов	7.7.1	●	●	●	●
	TZCP	161	Сравнение данных часов в области	7.7.2	●	●	●	●
	TADD	162	Сложение данных часов	7.7.3	●	●	●	●
	TSUB	163	Вычитание данных часов	7.7.4	●	●	●	●
	TRD	166	Чтение данных часов	7.7.5	●	●	●	●
	TWR	167	Запись данных часов	7.7.6	●	●	●	●
	HOURL	169	Счетчик рабочего времени	7.7.7	●	●		
Инструкции кода Грея	GRY	170	Преобразование целых чисел в коде Грея	7.8.1			●	●
	GBIN	171	Преобразование кода Грея в целое число	7.8.2			●	●
Обмен данными с FXON_3A	RD3A	176	Чтение аналогового значения из FXON-3A	7.9.1		●		
	WR3A	177	Передача данных к FXON-3A	7.9.2		●		
Функция внешнего ПЗУ	EXTR	180	Выполнение команды из внешнего ПЗУ	7.10.1			●	●

Табл. 7-2. Обзор специальных FNC-инструкций

Классификация	Инструкция	FNC	Назначение	Раздел	Серия типов ПК			
					FX1S	FX1N	FX2N	FX2NC
ЗАГРУЗКА-логическая связь сравнений	LD=	224	Инструкция сравнения, равно	7.11.1	●	●	●	●
	LD>	225	Инструкция сравнения, больше	7.11.1	●	●	●	●
	LD<	226	Инструкция сравнения, меньше	7.11.1	●	●	●	●
	LD<>	228	Инструкция сравнения, не равно	7.11.1	●	●	●	●
	LD≤	229	Инструкция сравнения, меньше равно	7.11.1	●	●	●	●
	LD≥	230	Инструкция сравнения, больше равно	7.11.1	●	●	●	●
Логическое "И"-логическая связь сравнений	AND=	232	Инструкция сравнения логического "И", равно	7.11.2	●	●	●	●
	AND>	233	Инструкция сравнения логического "И", больше	7.11.2	●	●	●	●
	AND<	234	Инструкция сравнения логического "И", меньше	7.11.2	●	●	●	●
	AND<>	236	Инструкция сравнения логического "И", не равно	7.11.2	●	●	●	●
	AND≤	237	Инструкция сравнения логического "И", меньше равно	7.11.2	●	●	●	●
	AND≥	238	Инструкция сравнения логического "И", больше равно	7.11.2	●	●	●	●
Логическое "ИЛИ"-логическая связь сравнений	OR=	240	Инструкция сравнения логического "ИЛИ", равно	7.11.3	●	●	●	●
	OR>	241	Инструкция сравнения логического "ИЛИ", больше	7.11.3	●	●	●	●
	OR<	242	Инструкция сравнения логического "ИЛИ", меньше	7.11.3	●	●	●	●
	OR<>	244	Инструкция сравнения логического "ИЛИ", не равно	7.11.3	●	●	●	●
	ORJ	245	Инструкция сравнения логического "ИЛИ", меньше равно	7.11.3	●	●	●	●
	ORi	246	Инструкция сравнения логического "ИЛИ", больше равно	7.11.3	●	●	●	●

Табл. 7-3. Обзор специальных FNC-инструкций

7.2 ИНСТРУКЦИИ ВВОДА/ВЫВОДА

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
FROM	78	Чтение данных из специальных модулей	7.2.9
TO	79	Запись данных в специальные модули	7.2.10

Табл. 7-4. Обзор инструкций FNC 70...79

7.2.1 ДЕСЯТИЧНАЯ КЛАВИАТУРА (ТКУ)

				TKY		FNC 70			
				Десятичная клавиатура					
				CPU	FX1S	FX1N	FX2N	FX2NC	
						●		●	
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		16 Bit	32 Bit	TKY	7	
					●	●	DTKY	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 (макс.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 ▾

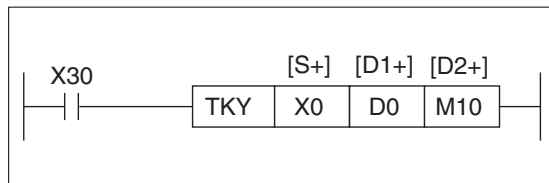


Рис. 7-1. Программирование TKY-инструкции

C000242C

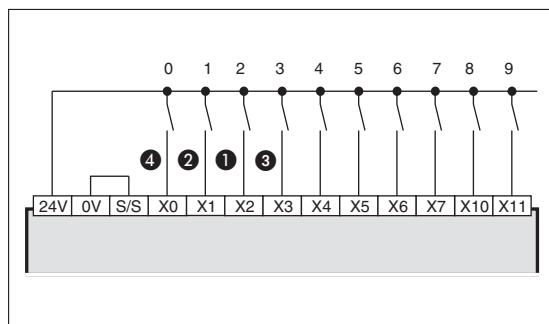


Рис. 7-2. Распределение клавиш

C000241C

В примере цифровые клавиши от 0 до 9 распределены на входы X. В (S+) указан стартовый адрес X0.

ПРИМЕР ▾

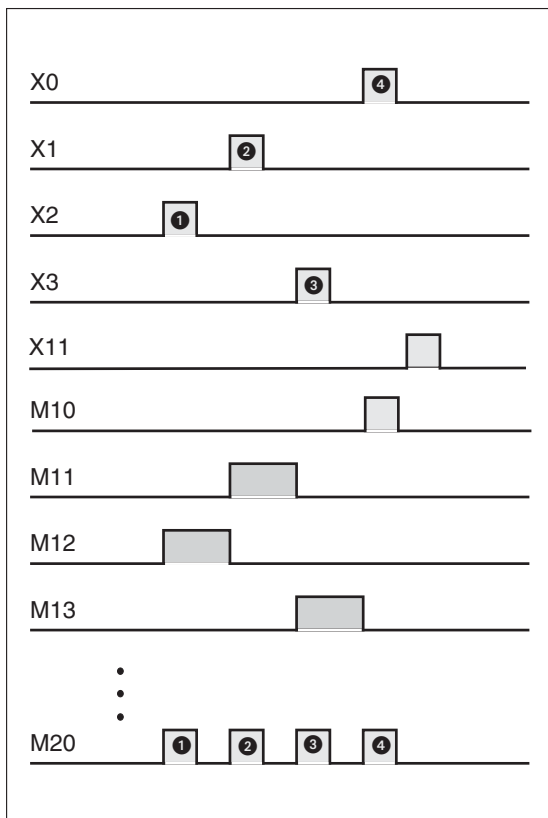


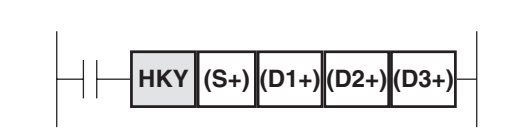
Рис. 7-3.

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 выключатся и новое содержание D0 будет 1 309. △

7.2.2 ШЕСТНАДЦАТИРИЧНАЯ КЛАВИАТУРА (HKY)

					HKY		FNC 71			
					Шестнадцатеричная клавиатура					
					CPU	FX1S	FX1N	FX2N	FX2NC	
								●	●	
Операнды	S+	D1+	D2+	D3+	Имп. инструкция(P)	Обработка		Шаги программы		
	X	Y	T, C, D, V, Z	Y, M, S ①		16 Bit	32 Bit	HKY	9	
						●	●	DHKY	17	

① 8 следующих друг за другом бит

ФУНКЦИЯ

Чтение шестнадцатеричной клавиатуры по входам

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

| HKY-инструкция может применяться в программе только один раз.

| При применении HKY-инструкции ПК должно работать с постоянным временем цикла.

ПРИМЕР ▾

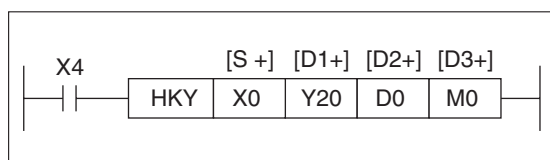


Рис. 7-4.
Пример программирования
HKY-инструкции

C000244C

ПРИМЕР ▾

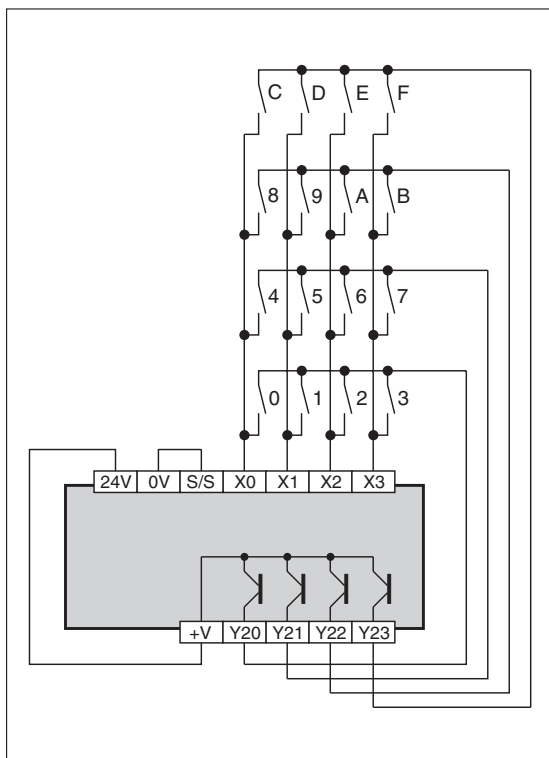


Рис. 7-5.

Пример программирования включения и отключения входов и меркеров

C000245C

ФУНКЦИОНАЛЬНЫЕ КЛАВИШИ

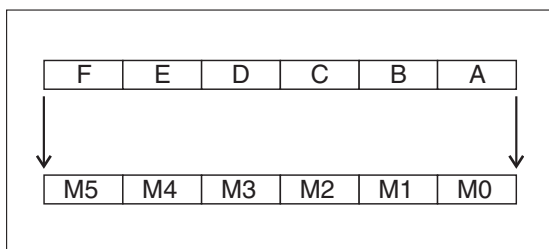


Рис. 7-6.

Пример распределения меркеров

C000246C

Если задействована клавиша А, включается меркер М0. М0 остается включенным до тех пор, пока не будут задействованы другие клавиши. Если в конце будет задействована клавиша D, то М0 отключится, а М3 включится.

Если задействуются две или больше клавиш, то принимается во внимание первая нажатая клавиша.

ВЫХОДЫ

До тех пор, пока удерживаются нажатыми клавиши А...F, М6 остается активным.

До тех пор, пока удерживаются нажатыми клавиши 1...9, М7 остается активным.

Если X4 отключается, данные в D0 не изменяются. Меркеры от М0 до М7 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	
						●	●	●	●	
Операнды	S+	D1+	D2+	n	Имп. инструкция(P)	Обработка		Шаги программы		
	X ^①	Y ^②	T, C, D, V, Z	Y, M, S ^③		16 Bit	32 Bit	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

НКУ-инструкция может применяться в программе только два раза.

Для корректного выполнения DSW-инструкции ПК должно применяться с транзисторными выходами.

ПРИМЕР ▾

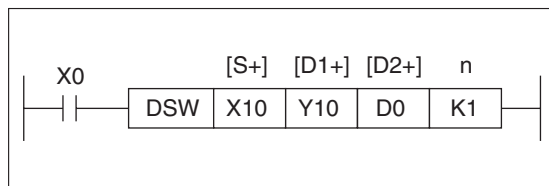


Рис. 7-7.
Пример программирования DSW-инструкции

C000247C



ПРИМЕР ▾

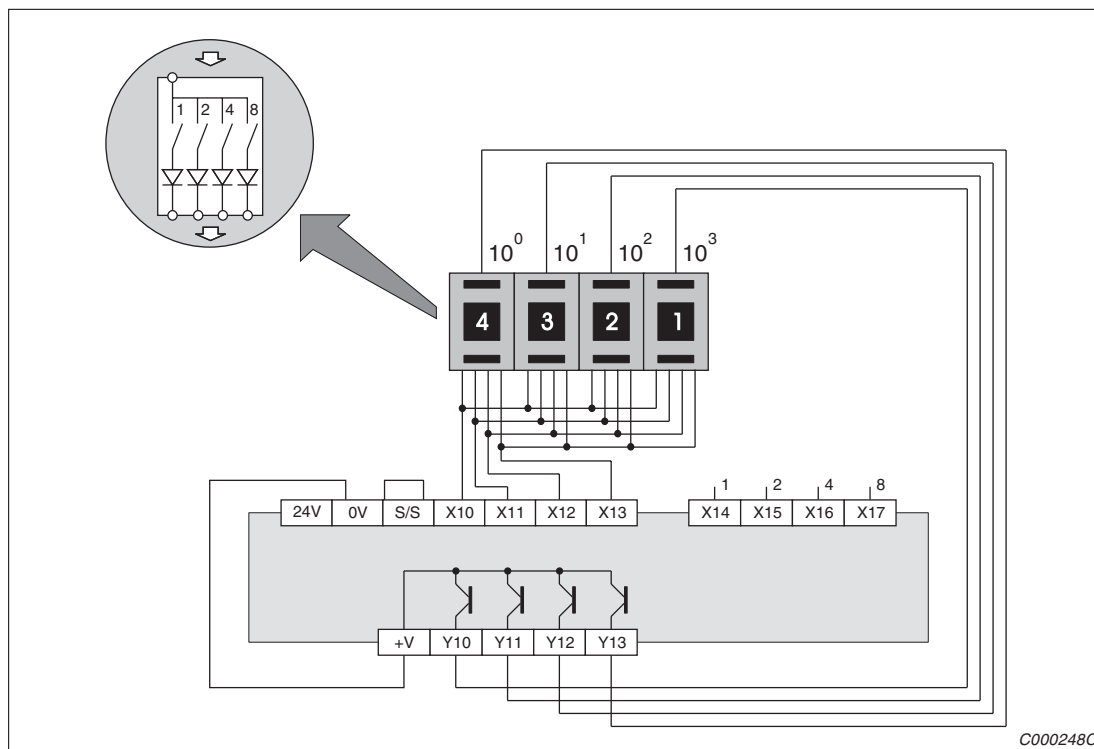


Рис. 7-8. Схема присоединения входов/выходов

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.

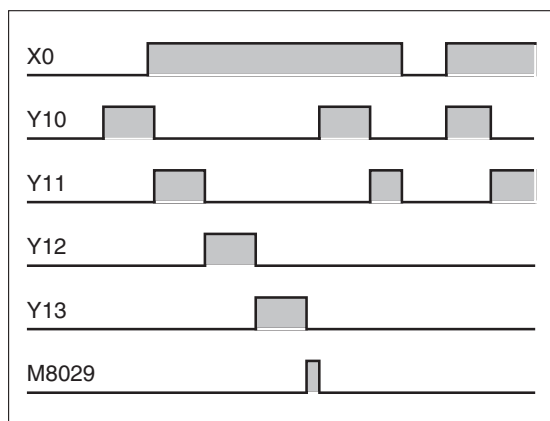


Рис. 7-9.

Временная диаграмма включения выходов

C000249C

7.2.4 7-МИ СЕГМЕНТНЫЙ ЦИФРОВОЙ ИНДИКАТОР (SEGD)

		SEGD		FNC 73				
		7-ми сегментный цифровой индикатор						
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	●	16 Bit	32 Bit	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:

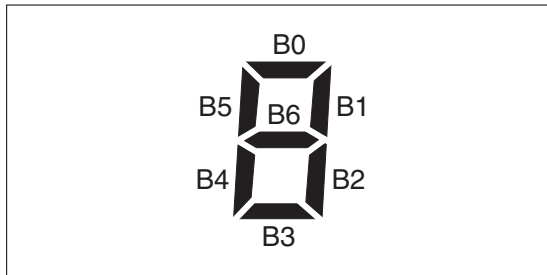


Рис. 7-10.
7-ми сегментный цифровой индикатор

C000251C

ПРИМЕР ▾

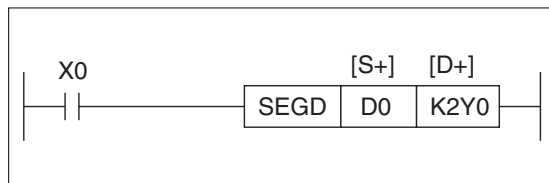


Рис. 7-11.
Пример программирования
SEGD-инструкции

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		
				●	●	●	●		
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		16 Bit	32 Bit	SEGL	7	
				●					

- ① n = 0...3: 8 друг за другом следующих выходов
n = 4...7: 12 друг за другом следующих выходов

ФУНКЦИЯ

Управление четырехзначным 7-ми сегментным индикатором с запоминанием показания

ОПИСАНИЕ

- С помощью этой инструкции можно управлять до двух четырехзначными 7-ми сегментными индикаторами. Управление выполняется по мультиплексному методу. Для работы записываются 4 тактовых выхода и для каждого четырехзначного индикатора еще по 4 выходных данных.
- Числовое значение (макс.9999), содержащееся в (S+), преобразовывается в BCD-код и выдается по выходам (D+)...((D+)+3). Если нужно управлять двумя четырехзначными индикаторами, то выход данных выполняется для двух индикаторов по выходам ((D+)+10)...((D+)+13).
- 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.

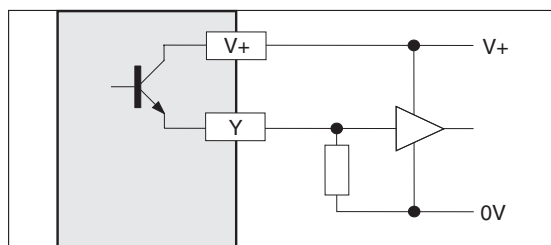


Рис. 7-12.
Положительная логика

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.

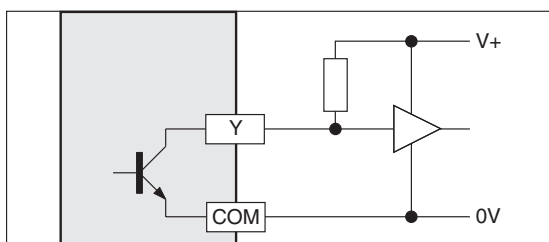


Рис. 7-13.
Отрицательная логика

C000255C

Логический тактовый сигнал: данные запоминаются, если тактовый сигнал младший (LOW).

Логический сигнал данных: активный провод данных является младшим (LOW).

ПК-ЛОГИКА	ТАКОВЫЙ 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)		

Табл. 7-5. 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.

SEGL-инструкция может использоваться только два раза в программе.

ПРИМЕР ▾

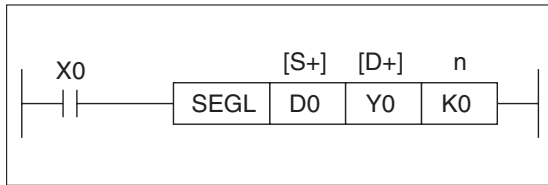
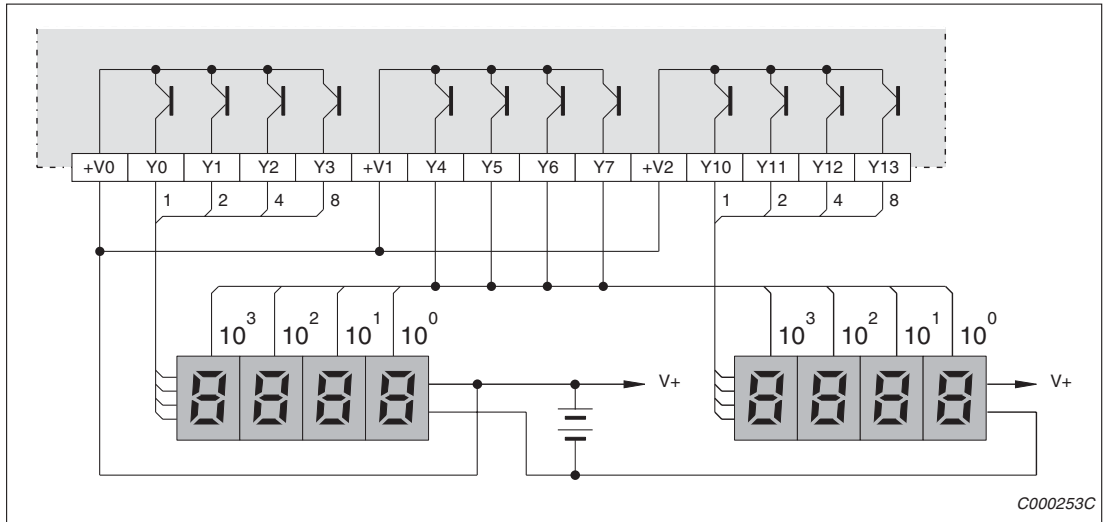


Рис. 7-14.
Пример программирования SEGL-инструкции

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
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		16 Bit	32 Bit	ARWS	

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 ▾

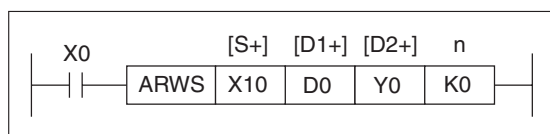


Рис. 7-16.
Пример программирования
ARWS-инструкции

C000256C

ПРИМЕР ▾

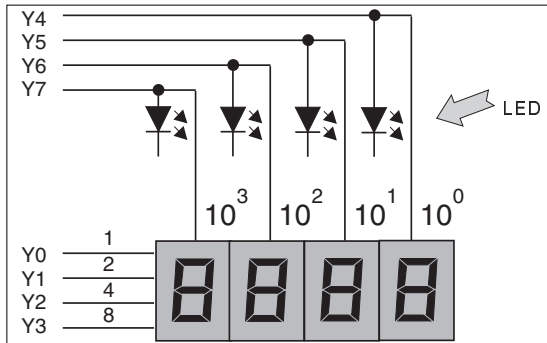


Рис. 7-17.
Пример присоединения 7-ми сегментного индикатора

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

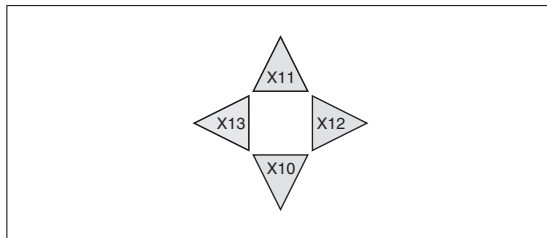


Рис. 7-18.
Пример выходов X10...X13

C000258C

По входам X12 и X13 определяются изменяемые позиции.

Если X0 включено, позиция 103 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):

103 - 102 - 101 - 100 - 103

Воздействие на X13 (сдвиг влево):

103 - 102 - 101 - 100 - 103

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				
Операнды		S+ ASCII-данные z.B. 0-9; A-Z; a-z		D+ T, C, D ①		Имп. инструкция(P)
						Обработка
				16 Bit	32 Bit	Шаги программы
				●		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 ▾

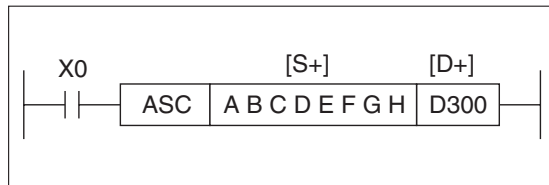


Рис. 7-19.
Пример программирования
ASC-инструкции

C000259C

	старшие 8 бит	младшие 8 бит
D300	42 (B)	41 (A)
D301	44 (D)	43 (C)
D302	46 (F)	45 (E)
D303	48 (H)	47 (G)

Рис. 7-20.
Запись и сохранение данных от A до H

C000260C

ЦИФРЫ / БУКВЫ	ASCII	ЦИФРЫ / БУКВЫ	ASCII	ЦИФРЫ / БУКВЫ	ASCII	ЦИФРЫ / БУКВЫ	ASCII
0	30	G	47	W	57	m	6D
1	31	H	48	X	58	n	6E
2	32	I	49	Y	59	o	6F
3	33	J	4A	Z	5A	p	70
4	34	K	4B	a	61	q	71
5	35	L	4C	b	62	r	72
6	36	M	4D	c	63	s	73
7	37	N	4E	d	64	t	74
8	38	O	4F	e	65	u	75
9	39	P	50	f	66	v	76
A	41	Q	51	g	67	w	77
B	42	R	52	h	68	x	78
C	43	S	53	i	69	y	79
D	44	T	54	j	6A	z	7A
E	45	U	55	k	6B		
F	46	V	56	l	6C		

Табл. 7-6. ASCII-конвертирование

7.2.8 ВЫДАЧА ДАННЫХ ПО ВЫХОДАМ (PR)

			PR		FNC 77			
			Выдача данных по выходам					
			CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●		
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D	Y ①		16 Bit	32 Bit	PR	5	
				●				

① 10 следующих друг за другом выхода

ФУНКЦИЯ

Выдача ASCII-знаков по выходам

ОПИСАНИЕ

- Выдача ASCII-знаков в (S+)...((S+)+3) по выходам.
- Выходы (D+)...((D+)+7) представляют биты b0...b7 сформированные по (S+).
- ((D+)+10) является тактовым сигналом, ((D+)+11) является флагом выполнения инструкции.

УАЗАНИЕ

PR-инструкция может использоваться в программе только два раза.

Для корректного выполнения инструкции ПК должно применяться с транзисторными выходами.

ПРИМЕР ▾

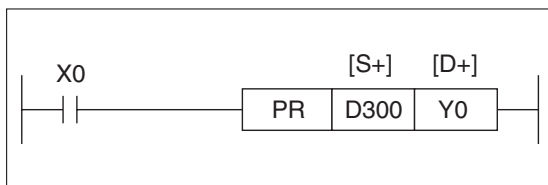


Рис. 7-21.
Пример программирования PR-инструкции

C000261C

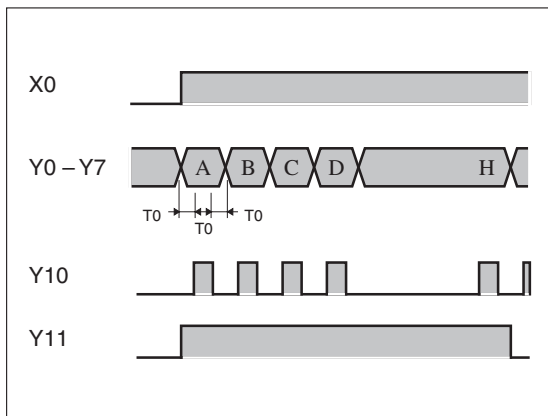


Рис. 7-22
Диаграмма примера программирования
для включения входов/выходов

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.



7.2.9 ЧТЕНИЕ ДАННЫХ ИЗ СПЕЦИАЛЬНОГО МОДУЛЯ (FROM)

		FROM		FNC 78			
		Чтение данных из специального модуля					
		CPU	FX1S	FX1N	FX2N	FX2NC	
				●	●	●	
Операнды	D+	n1, n2, n3	Имп. инструкция(P)	Обработка		Шаги программы	
	KnY, KnM, KnS, T, C, D, V, Z	K, H	●	16 Bit	32 Bit	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 области:

- n1 = 0...7
- n2 = 0...32768
- n3 = 1...32 при 16-ти битной инструкции
- n3 = 1...16 при 32-х битной инструкции

Каждый специальный модуль нумеруется последовательно от 0 до 7. Нумерация начинается с модуля, который первым связан с ПК. Максимум можно присоединить к ПК 8 специальных модулей (FX1N/FX2N). Адресация числовых входов и выходов и специальных модулей показана на рисунке.

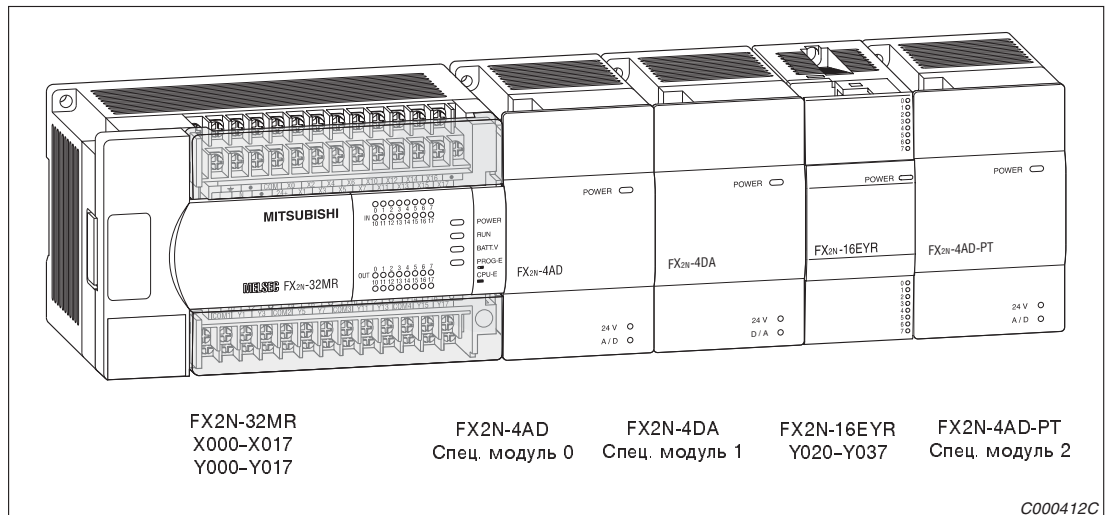


Рис. 7-23. Аналоговые модули присоединенные с другими устройствами ПК серии FX

В ПК имеется возможность прерывания выполнения инструкции посредством специального бита M8028:

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

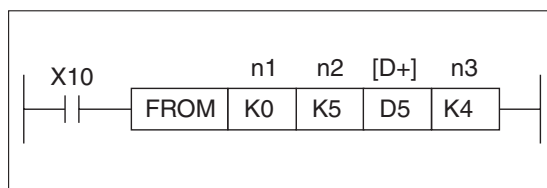


Рис. 7-24.
Пример программирования FROM-инструкции

C000264C

По примеру, показанному на рисунке, содержание адресов буферной памяти #5...#8 специального модуля с позиционного номера 0 передается в D5...D8. Значение адресации представлено по отдельности на следующем рисунке.

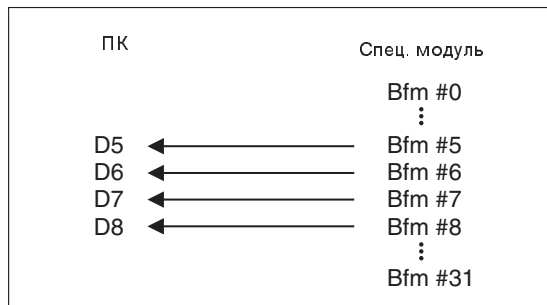


Рис. 7-25.
Адресация при FROM-инструкции

C000265C



УКАЗАНИЕ

Если должно передаваться только содержание адресов буферной памяти, то нужно занести в n3 значение 1.

7.2.10 ЗАПИСЬ ДАННЫХ В СПЕЦИАЛЬНЫЙ МОДУЛЬ (TO)

		TO		FNC 79					
		Запись данных в специальный модуль							
Операнды		S+	n1, n2, n3	Имп. инструкция(P)	CPU	FX1S	FX1N	FX2N	FX2NC
						●	●	●	
		KnY, KnM, KnS, T, C, D, V, Z	K, H		Обработка		Шаги программы		
					16 Bit	32 Bit	TO/TOP	9	
					●	●	DTO/DTOP	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 области:

- n1 = 0 ... 7
- n2 = 0 ... 32767
- n3 = 1...32 при 16-ти битной инструкции
- n3 = 1...16 при 32-ти битной инструкции

Каждый специальный модуль нумеруется последовательно от 0 до 7. Нумерация начинается с модуля, который первым связан с ПК. Максимум можно присоединить к ПК 8 специальных модулей. Адресация числовых входов и выходов и специальных модулей показана на рисунке.

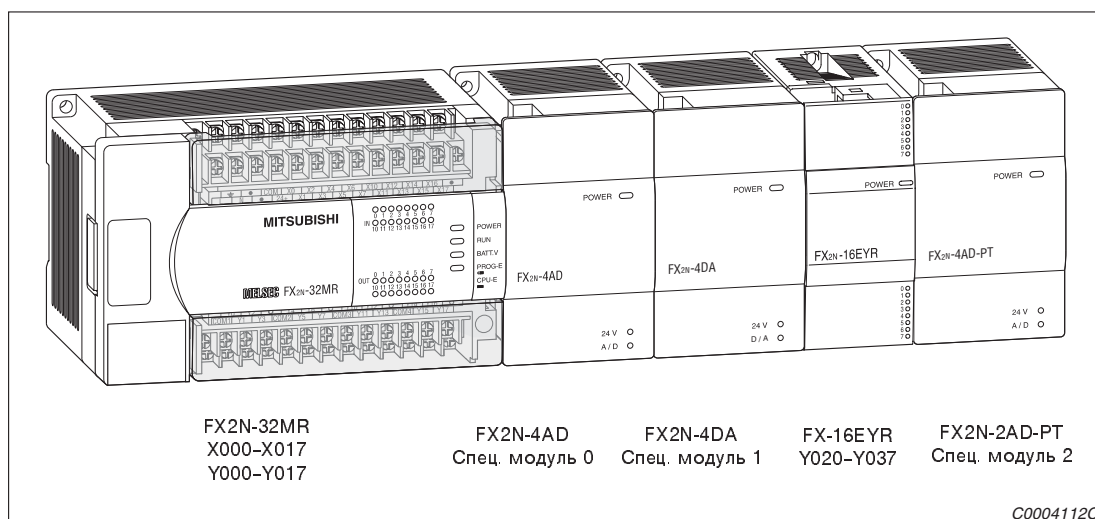


Рис. 7-26. Специальные модули, соединенные с другими устройствами ПК серии FX

В ПК имеется возможность прерывания выполнения инструкции посредством специального бита M8028:

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 (TO)

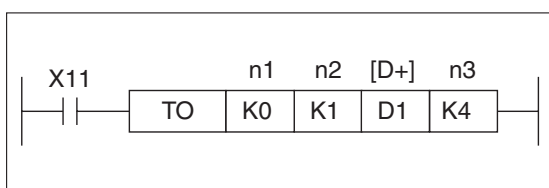


Рис. 7-27.
Пример программирования TO-инструкции

C000266C

По примеру, показанному на рисунке, D1...D4 передаются к адресам буферной памяти #1...#4 специального модуля, начиная с позиционного номера 0. Значение адресации представлено по отдельности следующим образом:

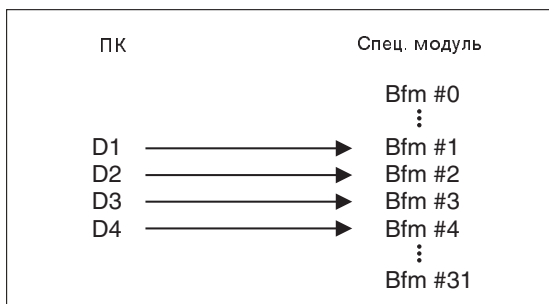


Рис. 7-28.
Адресация при TO-инструкции

C000267C



УКАЗАНИЕ

Если должно передаваться только содержание адресов буферной памяти, то нужно занести в n3 значение 1.

7.3 КОММУНИКАЦИОННЫЕ КОМАНДЫ

ОБЗОР ИНСТРУКЦИЙ FNC 80...89

Символ	FNC	Назначение	Разд.
RS	80	Последовательная передача данных	7.3.1
PRUN	81	Перенос входов или меркеров	7.3.2
ASCII	82	Преобразование в ASCII-знаки	7.3.3
HEX	83	Преобразование в шестнадцатеричный код	7.3.4
CCD	84	Проверка на сумму и паритет	7.3.5
VRRD	85	Чтение задаваемых значений из FX1N-8AV-BD и FX2N-8AV-BD	7.3.6
VRSC	86	Чтение положений переключателя из FX1N-8AV-BD и FX2N-8AV-BD	7.3.7
-	87		
PID	88	Контроль контура регулирования	7.3.8
-	89		

Табл. 7-7. Обзор инструкций FNC 80...89

7.3.1 ПРИЕМ И ПЕРЕДАЧА ДАННЫХ (RS)

				RS		FNC 80			
				Перенос входов или меркеров					
CPU		FX1S	FX1N	FX2N	FX2NC				
		●	●	●	●				
Операнды	S+	D+	n1, n2	Имп. инструкция(P)	Обработка		Шаги программы		
	D	D	K, D		16 Bit	32 Bit	RS	9	
					●				

ФУНКЦИЯ

Передача данных по модулям последовательной коммуникации FX2N-232BD, FX2N-32ADP, FX0N-485ADP и FX2N-485-BD.

ОПИСАНИЕ

С помощью RS-инструкции возможны прием и передача от большого числа приборов с последовательным интерфейсом. При этом коммуникация через последовательный интерфейсный адаптер управляется в четыре частичных шага:

- 1 Настройка коммутационных параметров
- 2 Выдача RS-инструкции, состоящей из:
 - (S+) = адрес старта буфера передачи
 - n1 = длина передаваемого сообщения (FX1S/FX1N макс. 256 байт, FX2N/FX2NC макс. 4096 байт)
 - (D+) = адрес старта буфера приема
 - n2 = длина принимаемого сообщения (FX1S/FX1N макс. 256 байт, FX2N/FX2NC макс. 4096 байт)

Сумма сообщений составляет при FX1S/FX1N макс. 512 байти при FX2N/FX2NC макс. 8000 байт.

- 3 Передача сообщения
- 4 Прием сообщения

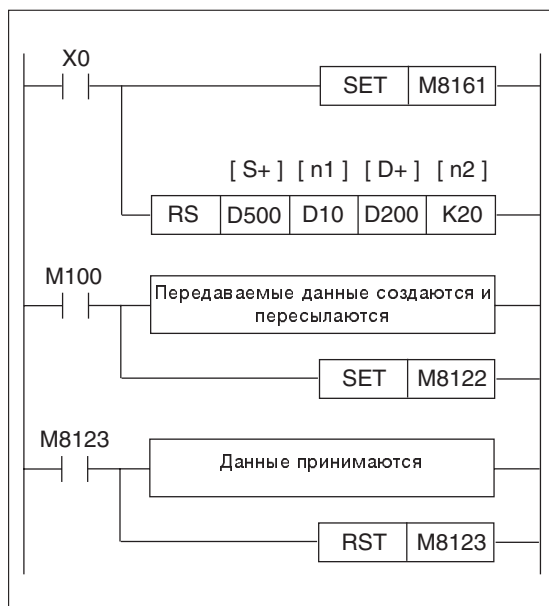


Рис. 7-29. Пример программирования RS-инструкции

C000220C

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, (0101): 1200, (0110): 2400, (0111): 4800, (1000): 9600 (1001): 19200		
b5				
b6				
b7				
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. 7-8.

Биты и их соответственное значение для RS232-коммуникации

Специальный регистр D8124 содержит значение стартового бита, в случае если он был выбран. Базовым значением является ASCII "STX" или 02H. 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 03H. 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. 7-30.
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 изменения 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.

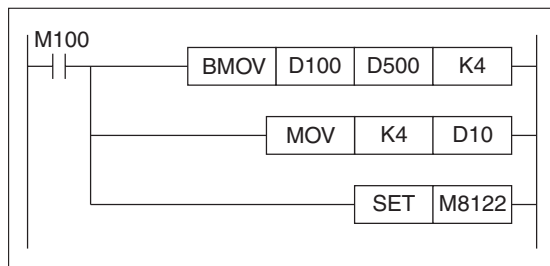


Рис. 7-31.

Пример программирования для передачи сообщения с помощью специального меркера M8122

C000175C

Сначала должны 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.

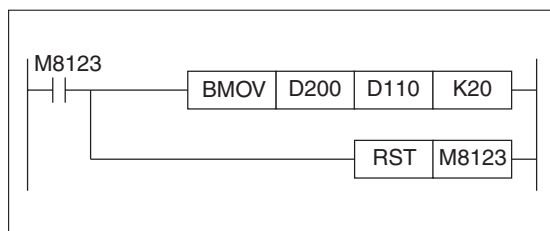


Рис. 7-32.

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.

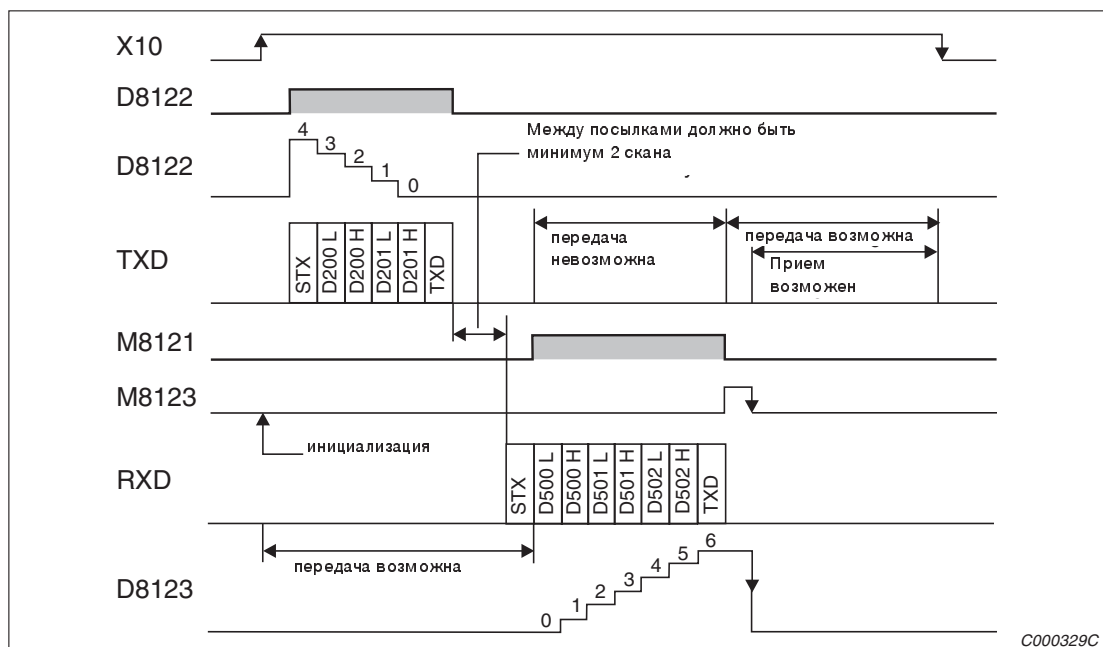


Рис. 7-33. Представление временного процесса

7.3.2 ПЕРЕНОС ВХОДОВ ИЛИ МЕРКЕРОВ (PRUN)

		PRUN		FNC 81				
		Перенос входов или меркеров						
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	5
					●	●	DPRUN/ DPRUNP	9

ФУНКЦИЯ

Перенос входов или меркеров в область данных, автоматически передаваемую в параллельном режиме.

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...

УКАЗАНИЕ

Структура и инициализация параллельной обработки данных поясняется подробнее в приложениях этого руководства.

ПРИМЕР ▾

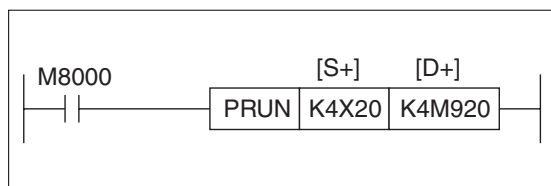
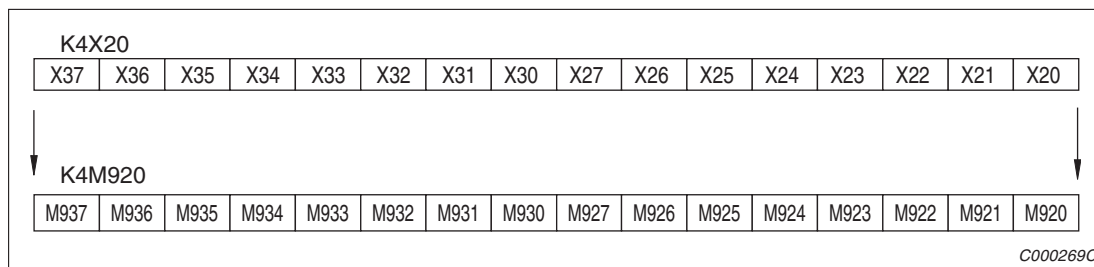


Рис. 7-34.
Пример программирования PRUN-инструкции

C000268C



C000269C

Рис. 7-35. Распределение меркеров

Меркеры M929 и M928 не записаны, так как PRUN-инструкция работает восьмерично.



7.3.3 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	●	16 Bit	32 Bit	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.

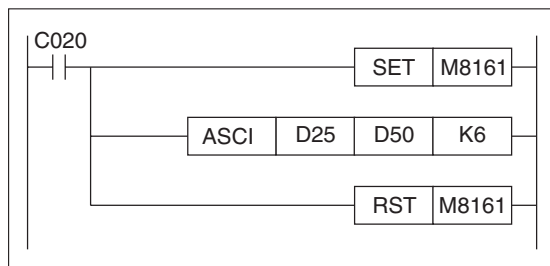


Рис. 7-36.
Пример программирования
ASCII-инструкции

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 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. При этом при ВКЛ. выбран 8-ми битовый режим.

УКАЗАНИЕ

Инструкция "SET M8161" требуется только тогда, если должен применяться 8-ми битовый режим.

СОСТАВЛЕНИЕ ASCII-ИНСТРУКЦИИ

- **Стартовый адрес**
Определенное здесь значение обозначает первый словный операнд, который содержит преобразуемое шестнадцатеричное число. Если нужно преобразовывать больше двух чисел, то считаются 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. 7-37.

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. 7-9. Таблица ASCII-кодов

7.3.4 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		●	16 Bit	32 Bit	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.

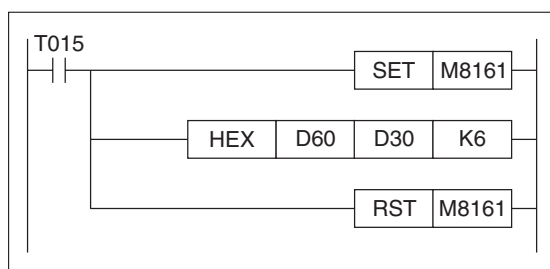


Рис. 7-38.
Пример программирования
HEX-инструкции

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.

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	„9“	„8“	–	„8“
D61	„A“	„0“	–	„9“
D62	„C“	„B“	–	„0“
			–	„A“
			–	„B“
			–	„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		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. 7-10. Таблица ASCII-кодов

7.3.5 ПРОВЕРКА НА СУММУ И ПАРИТЕТ (CCD)

				CCD		FNC 84					
				Проверка на сумму и паритет							
				CPU	FX1S	FX1N	FX2N	FX2NC	Обработка		Шаги программы
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	16 Bit	32 Bit	CCD/CCDP	7
							●				

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.

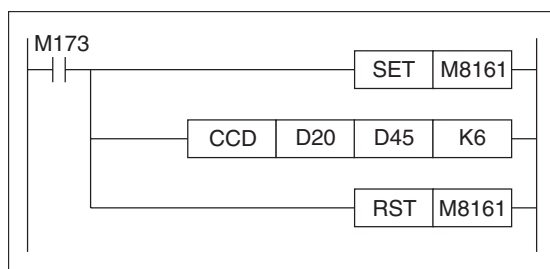


Рис. 7-40.
Пример программирования
CCD-инструкции

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.

СОСТАВЛЕНИЕ CSD-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			
D20	5 A	9 3	D20	-	9 3
	0101101	1001001			10010011
	0	1	D21	-	5 A
D21	7 4	0 F			01011010
	0111010	0000111	D22	-	0 F
	0	1			00001111
D22	B 2	4 D	D23	-	7 4
	1011001	0100110			01110100
	0	1	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. 7-41.
Графическое представление 16-ти битового и 8-ми битового форматов

7.3.6 ЧТЕНИЕ ЗНАЧЕНИЯ С FX1N-8AV-BD И FX2N-8AV-BD (VRRD)

		VRRD		FNC 85			
		Чтение значения настройки у FX-8AV					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H S = 0 ... 7	KnY, KnM, KnS, T, C, D, V, Z	●	16 Bit	32 Bit	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 ▾

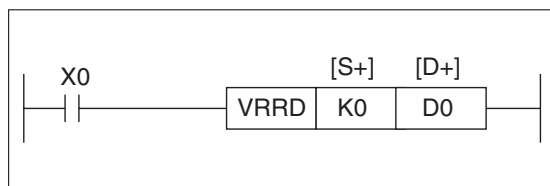


Рис. 7-42.

Пример программирования для VRRD-инструкции

C000270C

Считываются показания потенциометра "0" и установленное значение запоминается в регистре данных D0. △

7.3.7 ЧТЕНИЕ ПОЛОЖЕНИЯ ПЕРЕКЛЮЧАТЕЛЯ У FX1N-8AV-BD И FX2N-8AV-BD (VRSC)

		VRSC		FNC 86		
		Чтение положения переключателя у FX-8AV				
		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
Операнды	S+	K, H S = 0 ... 7				
	D+	KnY, KnM, KnS, T, C, D, V, Z	Имп. инструкция(P) ●	Обработка 16 Bit ●	32 Bit	Шаги программы VRSC/ VRSCP

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 платоместо расширения FX1S/FX1N- или FX2N-CPU.

ПРИМЕР ▾

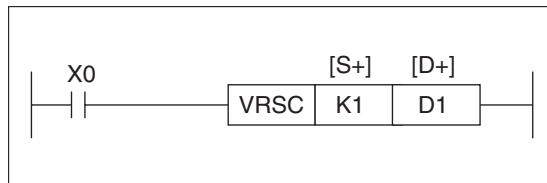


Рис. 7-43.

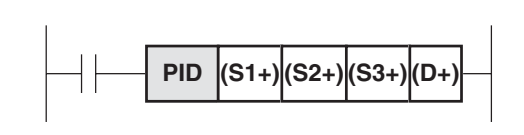
Пример программирования для VRSC-инструкции

C000271C

Положение переключения переключателя "1" запоминается в регистре данных D1.



7.3.8 ПРОГРАММИРОВАНИЕ ЗАМКНУТОГО КОНТУРА РЕГУЛИРОВАНИЯ (PID)

				PID		FNC 88			
				Контроль контура регулирования					
CPU				FX1S	FX1N	FX2N	FX2NC		
				● ^①	●	●	●		
Операнды	S1+, S2+	S3+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	D	D [Ⓜ]	D		16 Bit	32 Bit	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+) является начальным адресом области регистра, в которой запоминаются параметры контура регулирования. Для этого имеется 25 один за другим следующих регистров данных. Таблица 7-11 содержит обзор параметров.
- 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	
+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. 7-11. Обзор параметров регулирования

- ① Серия ПК FX1S/FX1N или FX2N/FX2NC располагает функцией автоматической настройки (Auto-Tuning). Эта функция определяет стартовое значение параметров регулирования K_p ((S3+)+3), T_i ((S3+)+4), T_D ((S3+)+6) и направление регулирования ((S3+)+1), бит (0). Все другие параметры должны указываться пользователем. С включением бита 4 в ((S3+)+1) активизируется функция Auto-Tuning. Выходные значения (MV) содержатся в области стартовых значений и контролируются ответом регулируемой системы (PV). Если оно достигнет 1/3 значения задаваемого значения, то функция Auto-Tuning отключается и бит 4 в ((S3+)+1) тоже отключается.

НЕОБХОДИМО СЛЕДИТЬ ЗА СЛЕДУЮЩИМИ ТОЧКАМИ ПРИ ПРИМЕНЕНИИ ФУНКЦИИ AUTO-TURNING.

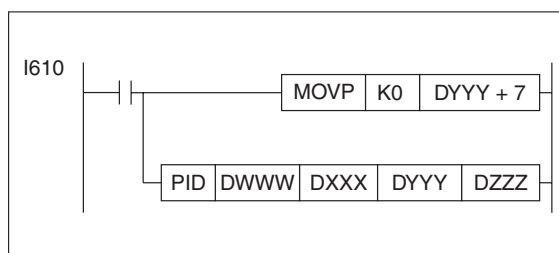
- Различие между текущим значением (CV) и заданным значением (SV) должно составлять 150%.
- Время (интервал) проверки должен быть кратным времени цикла (скана) и больше 1 с.
- Перед пуском функции Auto-Tuning система должна быть стабильной.

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.

**Рси.7-44.**

Пример программирования для PID-инструкции с предварительно установленной MOVP-инструкцией

C000330C

Интервал времени проверки (Sampling-Zeit) (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-Zeit) (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 выполняться 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 должно подводить данные для регулирования по последовательным связям.

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) $< =$ 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 ($-32768 - +32767$).	Выполнение заканчивается
K6750	Заданное значение - текущее значение < 150 или контур регулирования нестабилен (Разность между заданным и текущим значениями колеблется очень сильно и очень быстро)	
K6751	Заданное значение значительно завышено	
K6752	Большие колебания при функции автоматической настройки (Auto-Tuning)	

Табл. 7-12. Обзор сообщений об ошибках, запоминаемых в регистре D8067

ПРИМЕР ▾

Ввод PID-инструкции

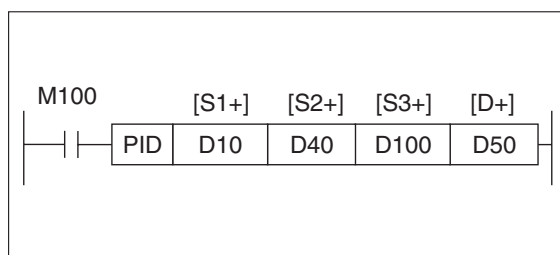


Рис. 7-45.

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.

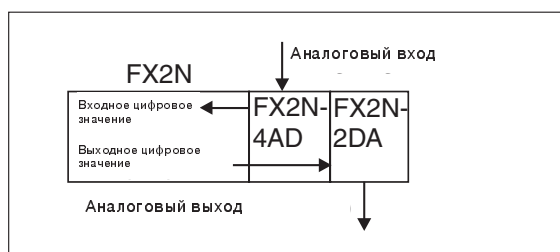


Рис. 7-46.

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. 7-13. 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+)).

Следующий рисунок представляет систему координат, в которой оси координат представлены для заданного значения контура регулирования и для текущего значения. На Рис. 7-47 штриховая линия представляет процесс движения вперед сплошная линия процесс движения назад.

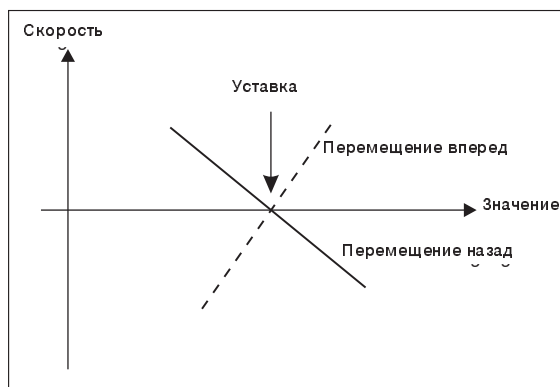


Рис. 7-47.

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.

7.4 ИНСТРУКЦИИ НАД ЧИСЛАМИ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (FX2N)

ОБЗОР ИНСТРУКЦИЙ FNC 110...132

Символ	FNC	Назначение	Разд.
DECMP	110	Сравнение чисел с плавающей запятой	7.4.1
DEZCP	111	Сравнение чисел с плавающей запятой в области	7.4.2
DEBCD	118	Преобразование чисел с плавающей запятой в научный формат чисел	7.4.3
DEBIN	119	Преобразование научного формата чисел в числа с плавающей запятой	7.4.4
DEADD	120	Сложение чисел с плавающей запятой	7.4.5
DESUB	121	Вычитание чисел с плавающей запятой	7.4.6
DEMUL	122	Умножение чисел с плавающей запятой	7.4.7
DEDIV	123	Деление чисел с плавающей запятой	7.4.8
DESQR	127	Корень квадратный из числа с плавающей запятой	7.4.9
INT	129	Преобразование формата с плавающей запятой в десятичный формат	7.4.10
SIN	130	Расчет синуса числа с плавающей запятой	7.4.11
COS	131	Расчет косинуса числа с плавающей запятой	7.4.12
TAN	132	Расчет тангенса числа с плавающей запятой	7.4.13

Табл. 7-14. Обзор инструкций FNC 110...132

7.4.1 СРАВНЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DECMP)

			DECMP		FNC 110			
			Сравнение чисел с плавающей запятой					
			CPU	FX1S	FX1N	FX2N	FX2NC	
						●	●	
Операнды	S1+	S2+	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		●	16 Bit	32 Bit	DECMP
						●	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 ▾

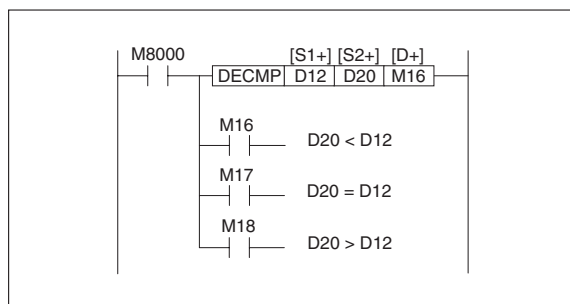


Рис. 7-48.

Пример программирования DECMP-инструкции

C000350C

При включении меркера M8000 сравниваются число с плавающей запятой, указанное в D20 (S2+), с числом с плавающей запятой, указанным в 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.

7.4.2 ЗОННЫЙ КОМПОРАТОР (DEZCP)

					DEZCP		FNC 111		
					Сравнение числа с плавающей запятой с областью				
CPU					FX1S	FX1N	FX2N	FX2NC	
							●	●	
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		●	16 Bit	32 Bit	DEZCP
						●		DEZCPP	13

ФУНКЦИЯ

Сравнение числа 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 ▾

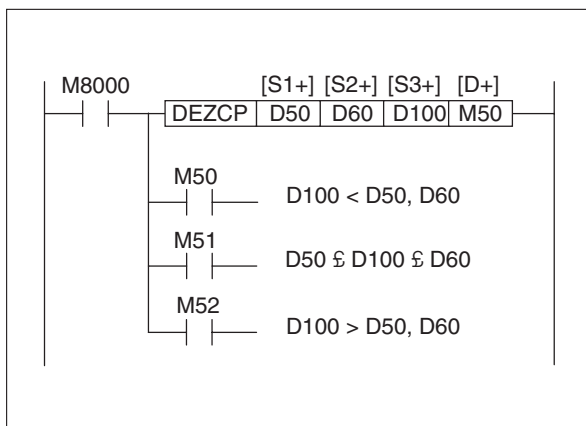


Рис. 7-49.

Пример программирования
DEZCP-инструкции

C000351C

При включении меркера M8000 сравниваются число с плавающей запятой D100 (S3+) с числами в области между D50 (S1+) и D60 (S2+).

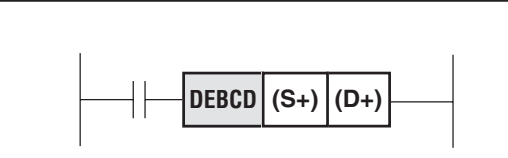
Если число в D100 меньше числа между D50 и D60, то включается меркер M50.

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.

△

7.4.3 ПРЕОБРАЗОВАНИЕ ФОРМАТА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В НАТУРАЛЬНУЮ ФОРМУ (DEBCD)

		DEBCD		FNC 118			
		Преобразование формата с плавающей запятой в научный формат чисел					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
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		●	16 Bit	32 Bit	DEBCD
					●	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 ▾

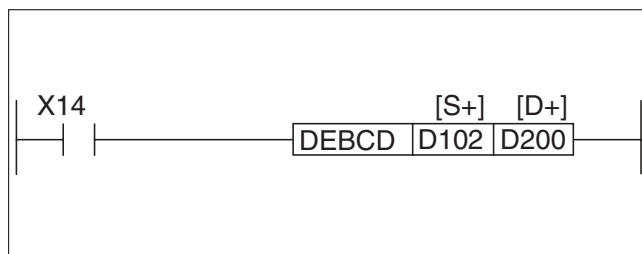


Рис. 7-50.
Пример программирования DEBCD-инструкция

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.

△

7.4.4 ПРЕОБРАЗОВАНИЕ ИЗ НАТУРАЛЬНОЙ ФОРМЫ В ФОРМАТ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEBIN)

		DEBIN		FNC 119			
		Преобразование научного формата чисел в формат с плавающей запятой					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
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))		●	16 Bit	32 Bit	DEBIN
					●	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 x 10⁹ (D+), (D+)+1) в формате с плавающей запятой.

ПРИМЕР ▾

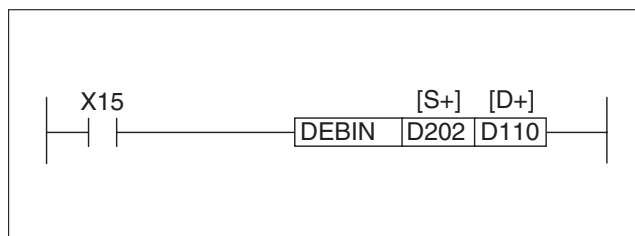


Рис. 7-51.
Пример программирования
DEBIN-инструкции

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.



7.4.5 СЛОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEADD)

				DEADD		FNC 120					
				Сложение чисел с плавающей запятой							
				CPU	FX1S	FX1N	FX2N	FX2NC			
							●		●		
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))	●	16 Bit	32 Bit	DEADD	13	DEADDP

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 ▾

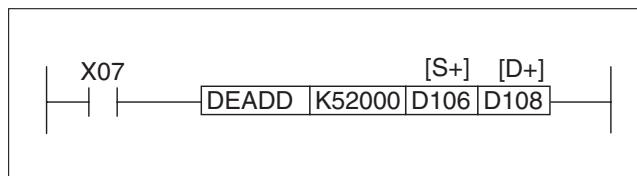


Рис. 7-52.
Пример программирования
DEADD-инструкции

C000354C

При включении входа X7 к константе K52000 прибавится число с плавающей запятой, записанное в D106.

Результат сохранится в D108.



7.4.6 ВЫЧИТАНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DESUB)

			DESUB		FNC 121			
			Вычитание чисел с плавающей запятой					
			CPU	FX1S	FX1N	FX2N	FX2NC	
						●	●	
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))		●	16 Bit	32 Bit	DESUB
						●	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 ▾

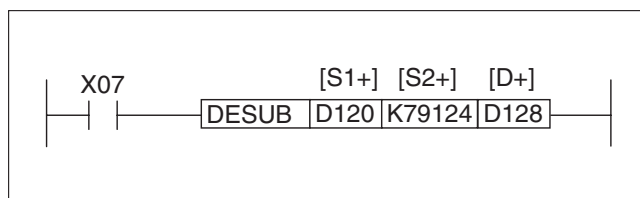


Рис. 7-53.
Пример программирования
DESUB-инструкции

C000355C

При включении входа X07 из числа с плавающей запятой D 120 вычитается константа K79124. Результат сохранится в D128.

△

7.4.7 УМНОЖЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEMUL)

			DEMUL		FNC 122				
			Умножение чисел с плавающей запятой						
			CPU	FX1S	FX1N	FX2N	FX2NC		
						●		●	
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))		●	16 Bit	32 Bit	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 ▾

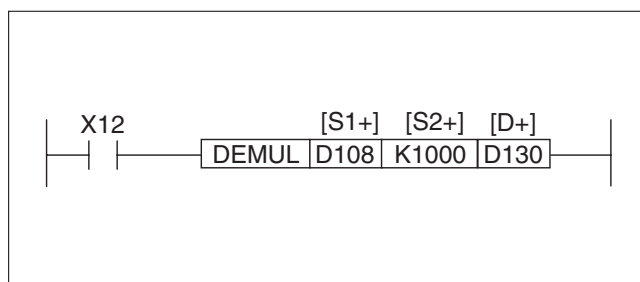


Рис. 7-54.
Пример программирования
DEMUL-инструкции

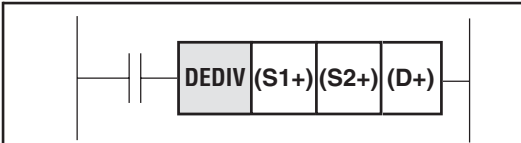
C000356C

При включении меркера M12 число с плавающей запятой в D108 перемножается с константой K1000.

Результат сохранится в D130.

△

7.4.8 ДЕЛЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DEDIV)

				DEDIV		FNC 123			
				Деление чисел с плавающей запятой					
				CPU	FX1S	FX1N	FX2N	FX2NC	
							●	●	
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))		●	16 Bit	32 Bit	DEDIV	13
						●	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 ▾

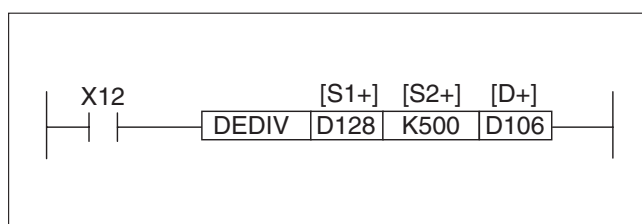


Рис. 7-55.
Пример программирования DE-DIV-инструкции

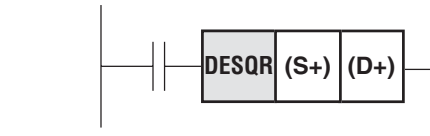
C000357C

При включении входа X10 число с плавающей запятой в D128 делится на константу K500.

Результат сохранится в D106.



7.4.9 КОРЕНЬ КВАДРАТНЫЙ ИЗ ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DESQR)

		DESQR		FNC 127			
		Корень квадратный из числа с плавающей запятой					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
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))		●	16 Bit	32 Bit	DESQR
					●	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 M8067t.

ПРИМЕР ▾

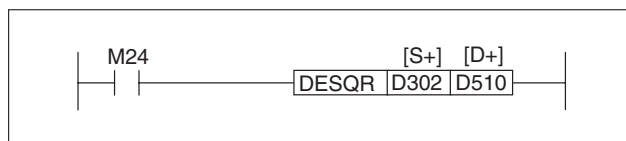


Рис. 7-56.
Пример программирования
DESQR-инструкции

C000358C

При включении меркера M24 вычисляется корень квадратный из числа с плавающей запятой в D302.

Результат сохранится в D510.



7.4.10 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		
					●	●		
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)		16 Bit	32 Bit	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 (zero) 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 ▾

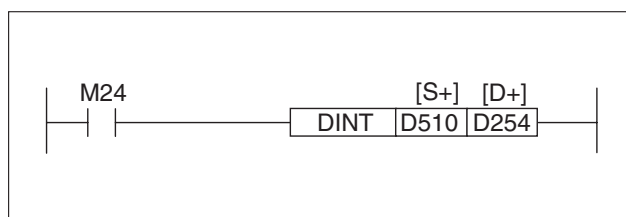


Рис. 7-57.
Пример программирования DINT-инструкции

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.

△

7.4.11 РАСЧЕТ СИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DSIN)

		DSIN		FNC 130		
		Расчет синуса числа с плавающей запятой				
CPU		FX1S	FX1N	FX2N	FX2NC	
				●	●	
Операнды	S+	D+	Имп. инструкция(P)		Шаги программы	
	D (числа с плавающей запятой в радианах (32 бита))	D (число с плавающей запятой (32 бита))	●	16 Bit	32 Bit	DSIN
				●	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, указанного в (S+), должно быть углом между 0 и 360 градусами (0 и 2 π). Указание значения угла должно выполняться в радианах (градус $\times \pi / 180$ [rad]).

ПРИМЕР ▾

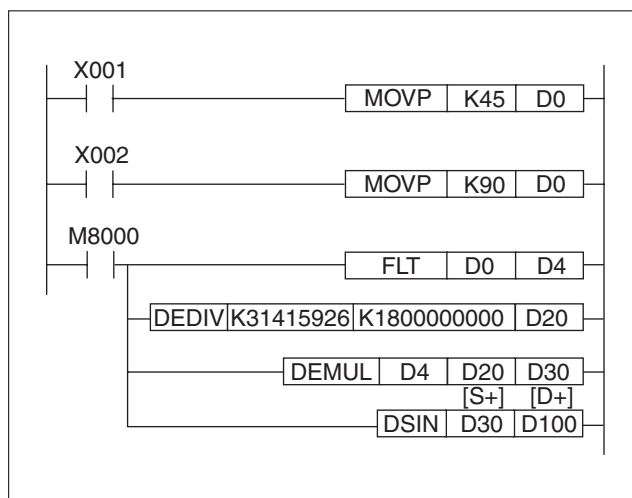


Рис. 7-58.

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.

△

7.4.12 РАСЧЕТ КОСИНУСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DCOS)

		DCOS		FNC 131				
		Расчет косинуса числа с плавающей запятой						
Операнды		S+		Имп. инструкция(P)	Обработка		Шаги программы	
		D (числа с плавающей запятой в радианх (32 бита))			D (число с плавающей запятой (32 бита))		16 Bit	32 Bit
				●		●	DCOSP	9

ФУНКЦИЯ

Рассчитывается косинуса 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, указанного в (S+), должно быть углом между 0 и 360 градусами (0 и 2 π). Указание значения угла должно выполняться в радианах (градус × π / 180 [rad]).

ПРИМЕР ▾

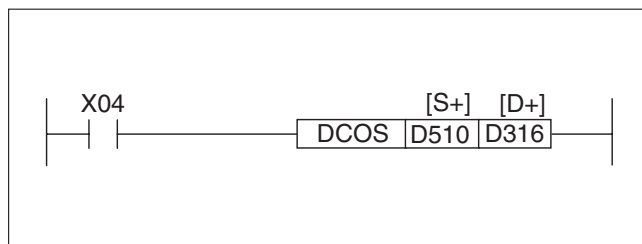


Рис. 7-59.
Пример программирования DCOS-инструкции

C000361C

С включением входа X4 рассчитывается значение косинуса угла, указанного в радианах (D510 и D511).

Результат записывается в D316 и D317.



7.4.13 РАСЧЕТ ТАНГЕНСА ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ (DTAN)

		DTAN		FNC 132		
		Расчет тангенса числа с плавающей запятой				
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))	●	16 Bit 32 Bit	DTAN DTANP

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, указанного в (S+), должно быть углом между 0 и 360 градусами (0 и 2 pi). Указание значения угла должно выполняться в радианах (Градус x pi / 180 [rad]).

ПРИМЕР ▾

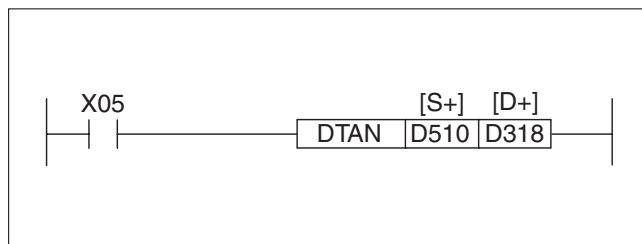


Рис. 7-60.
Пример программирования DTAN-инструкции

C000362C

С включением входа X5 рассчитывается значение тангенса угла, указанного в радианах (D510 и D511).

Результат записывается в D318 и D319.



7.5 ИНСТРУКЦИИ ПО ОБРАБОТКЕ ДАННЫХ

ОБЗОР ИНСТРУКЦИЙ FNC 147 ... 149 (FX2N)

Символ	FNC	Назначение	Разд.
–	140 ... 146		
SWAP	147	Обмен старшими и младшими байтами	7.5.1
–	148 149		

Табл. 7-15. Обзор инструкции FNC 140 ... 149

7.5.1 ОБМЕН СТАРШИМИ И МЛАДШИМИ БАЙТАМИ (SWAP)

		SWAP		FNC 147			
		Обмен старшими и младшими байтами					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
Операнды	S+	Имп. инструкция(P)	Обработка		Шаги программы		
	KnY, KnM, KnS, T, C, D, V, Z	●	16 Bit	32 Bit	SWAP, SWAPP	5	
			●	●	DSWAP, DSWAPP	9	

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.

ПРИМЕР ▾

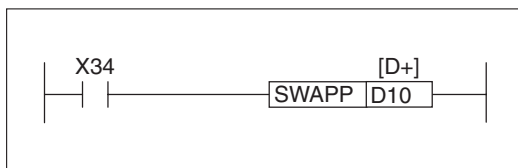


Рис. 7-61.
Пример программирования SWAPP-инструкции

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. 7-16.
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. 7-17.
Обмен байтами при DSWAPP-инструкции

7.6 ИНСТРУКЦИИ ПОЗИЦИОНИРОВАНИЯ (FX1S/FX1N)

ОБЗОР ИНСТРУКЦИЙ FNC 155...159

Символ	FNC	Назначение	Разд.
ABS	155	Чтение абсолютной текущей позиции	7.6.1
ZRN	156	Перемещение к нулевой (исходной) точке	7.6.2
PLSV	157	Выдача импульсов переменной частоты	7.6.3
DRVI	158	Позиционирование по инкрементальным (приращиваемым) значениям	7.6.4
DRVA	159	Позиционирование по абсолютным значениям	7.6.5

Табл. 7-18. Обзор инструкции FNC 155...159

7.6.1 Предупреждение при использовании инструкций позиционирования.

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.

7.6.2 Формат выходных импульсов

При реализации позиционирования с помощью выходов ПК используется формат сигналов “Последовательность импульсов + Знак”, как показано на рисунке.

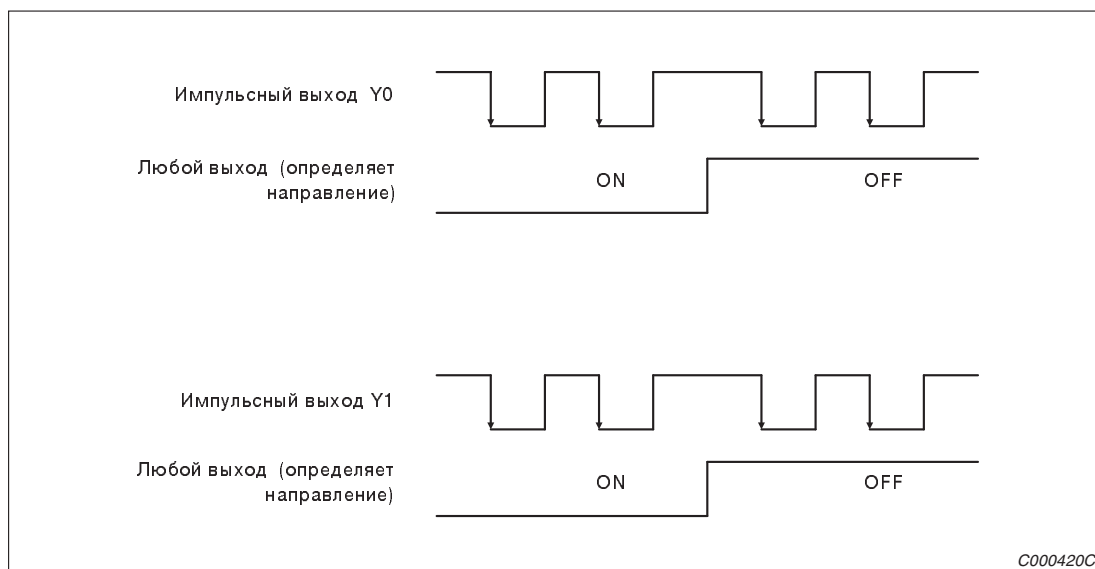


Рис. 7-62: Формат выходных сигналов

Обязательно установите в сервоусилителе входные данные:

Формат входа: Последовательность импульсов + знак

Логика: Негативная

7.6.3 Операнды используемые при позиционировании

Операнд	Доступ	Описание
M8145	Могут устанавливаться в программе	Стоп выдачи импульсов выход Y0
M8146		Стоп выдачи импульсов выход Y1
M8147	Только чтение	Заданные импульсы выданы Y0
M8148		Заданные импульсы выданы Y1

Табл. 7-19. Специальные меркеры для инструкций позиционирования

Операнд		Значение по умолчанию	Описание
D8140	Двойное слово	0	Регистр текущего значения Y0 (32 Bit) ①
D8141	Двойное слово		
D8142	Двойное слово	0	Регистр текущего значения Y1(32 Bit) 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 (32 Bit) Диапазон: от 100 Hz до 100 kHz
D8147	Двойное слово		
D8148		100	Время разгона/торможения до максимальной скорости для инструкций FNC156, FNC158 и FNC159 Диапазон: от 50 до 5000 мсек

Табл. 7-20. Специальные регистры для инструкций позиционирования

- 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)

7.6.4 Пример подключения серводвижителя

На рисунке показана схема подключения к ПК FX1S-30MT серводвижителя MELSERVO MR-J2-mA:

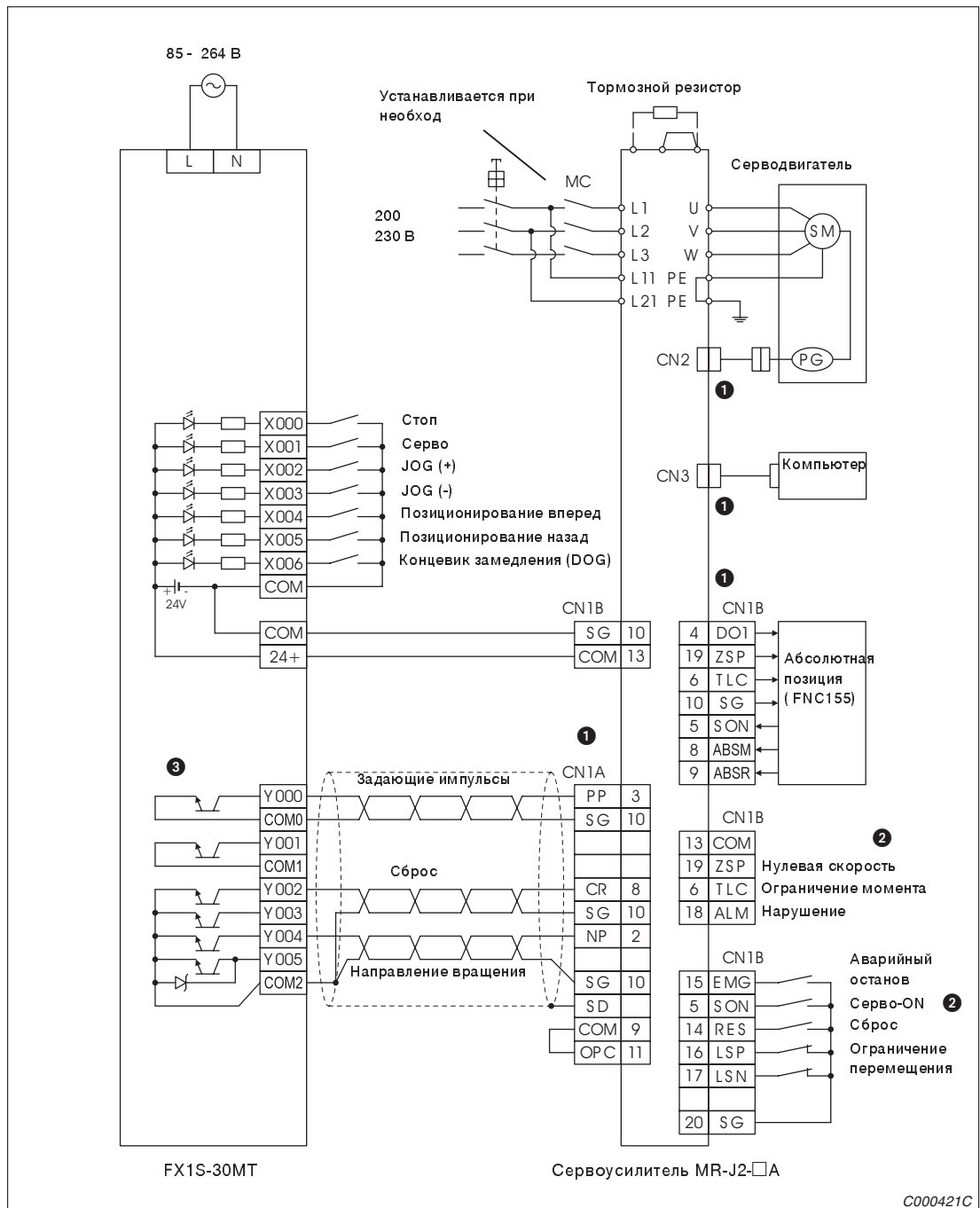


Рис. 7-63. Принципиальная схема.

- 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.

7.6.5 Пример программы

При работе, позиционирование выполняется абсолютным способом.

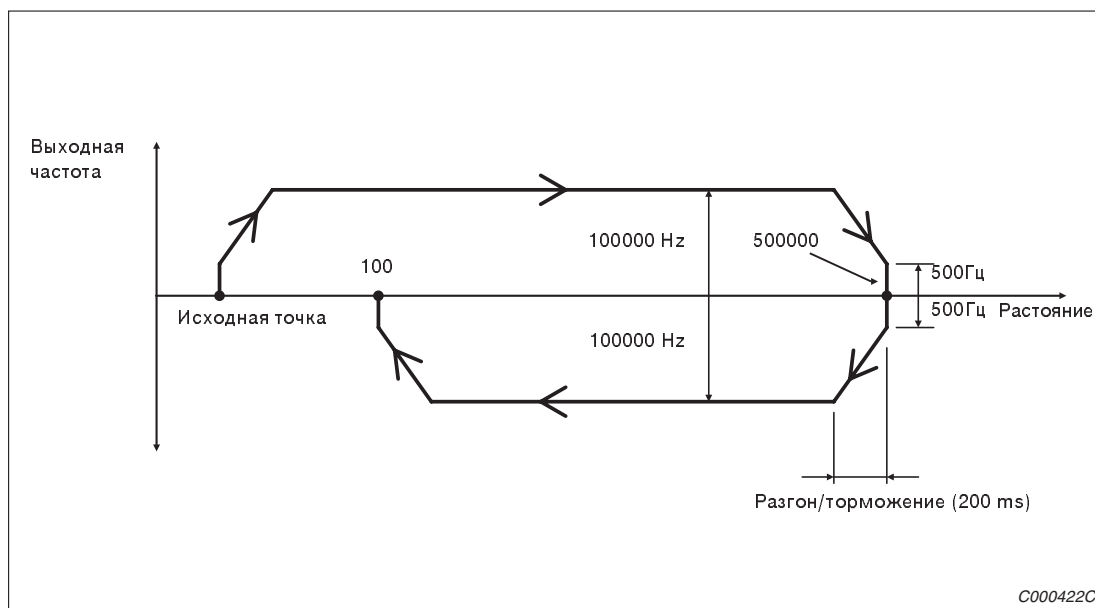


Рис. 7-64. Пример программы

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 -

$$f_{\min} = \sqrt{\frac{100000}{2 \times 0,2}} = 500 \text{ Гц}$$

SPS подключение показано в гл 7.6.4

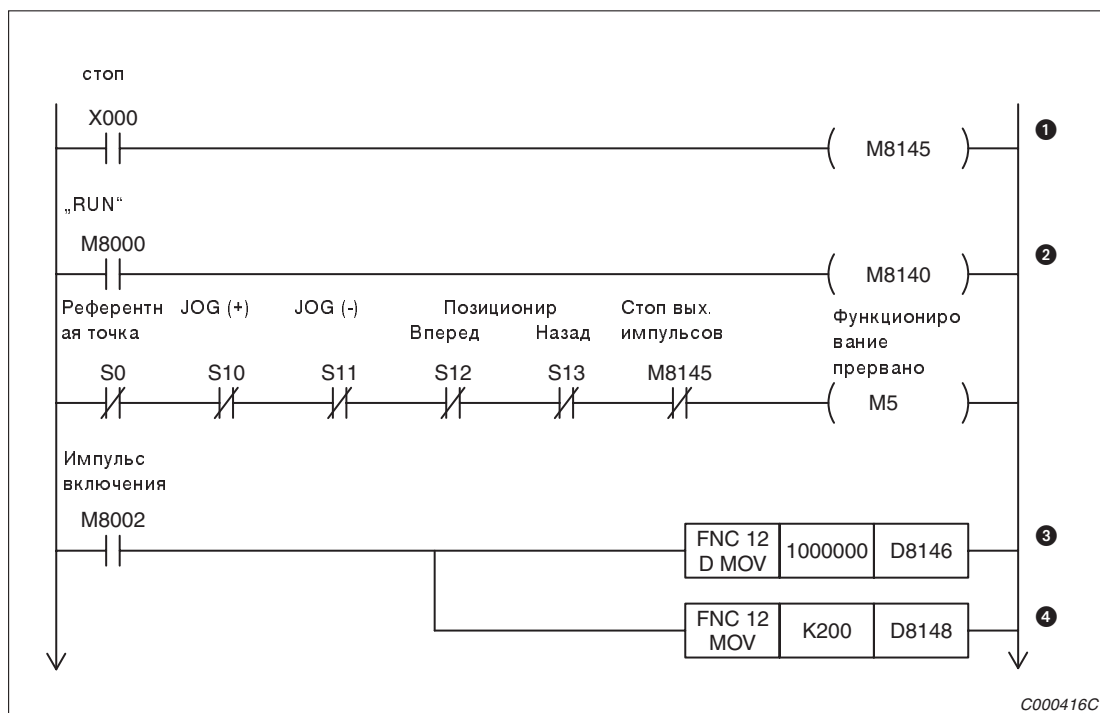


Рис. 7-65. Пример программы движения

- 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.

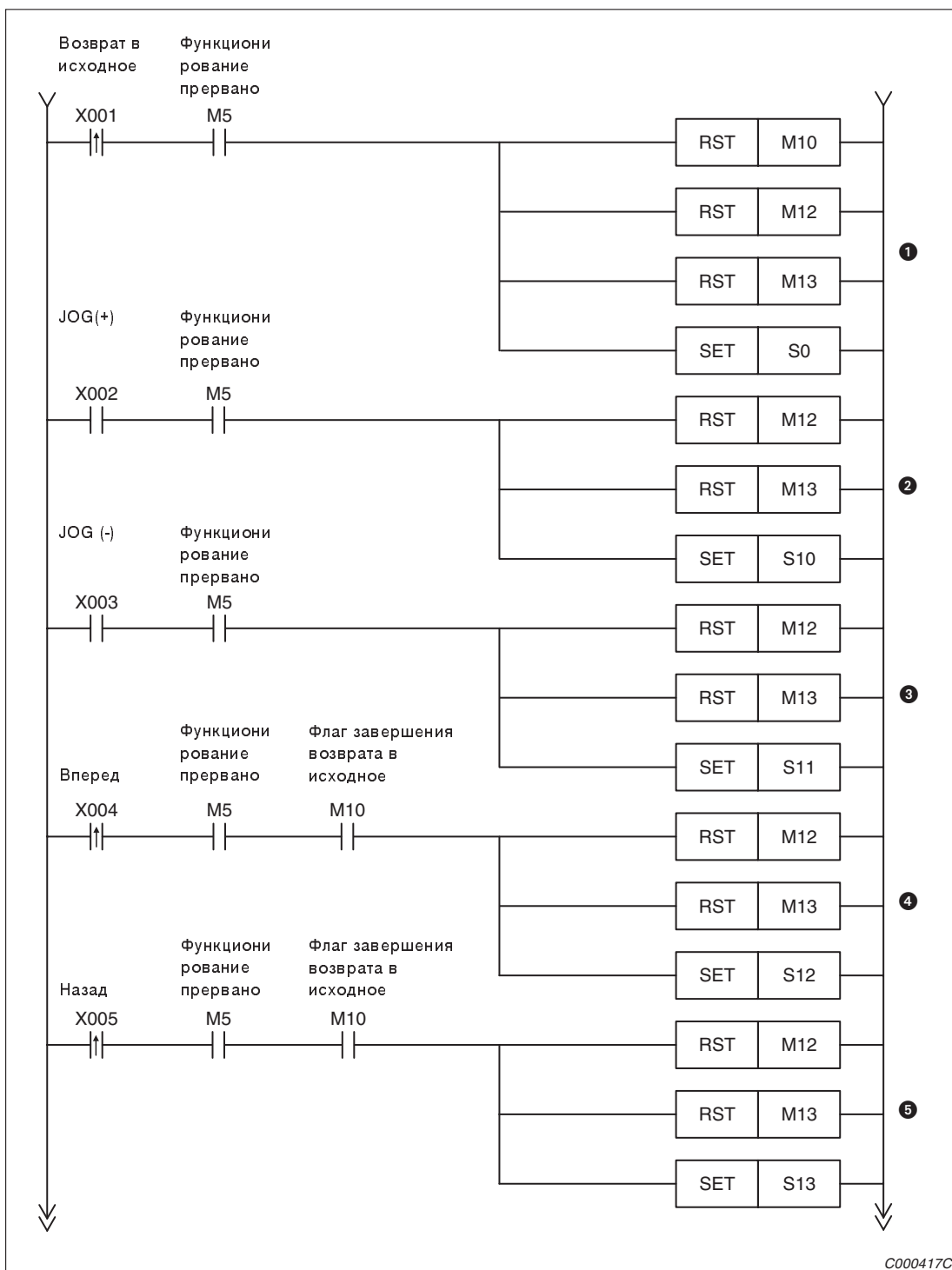


Рис. 7-66. Пример программы

- 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.

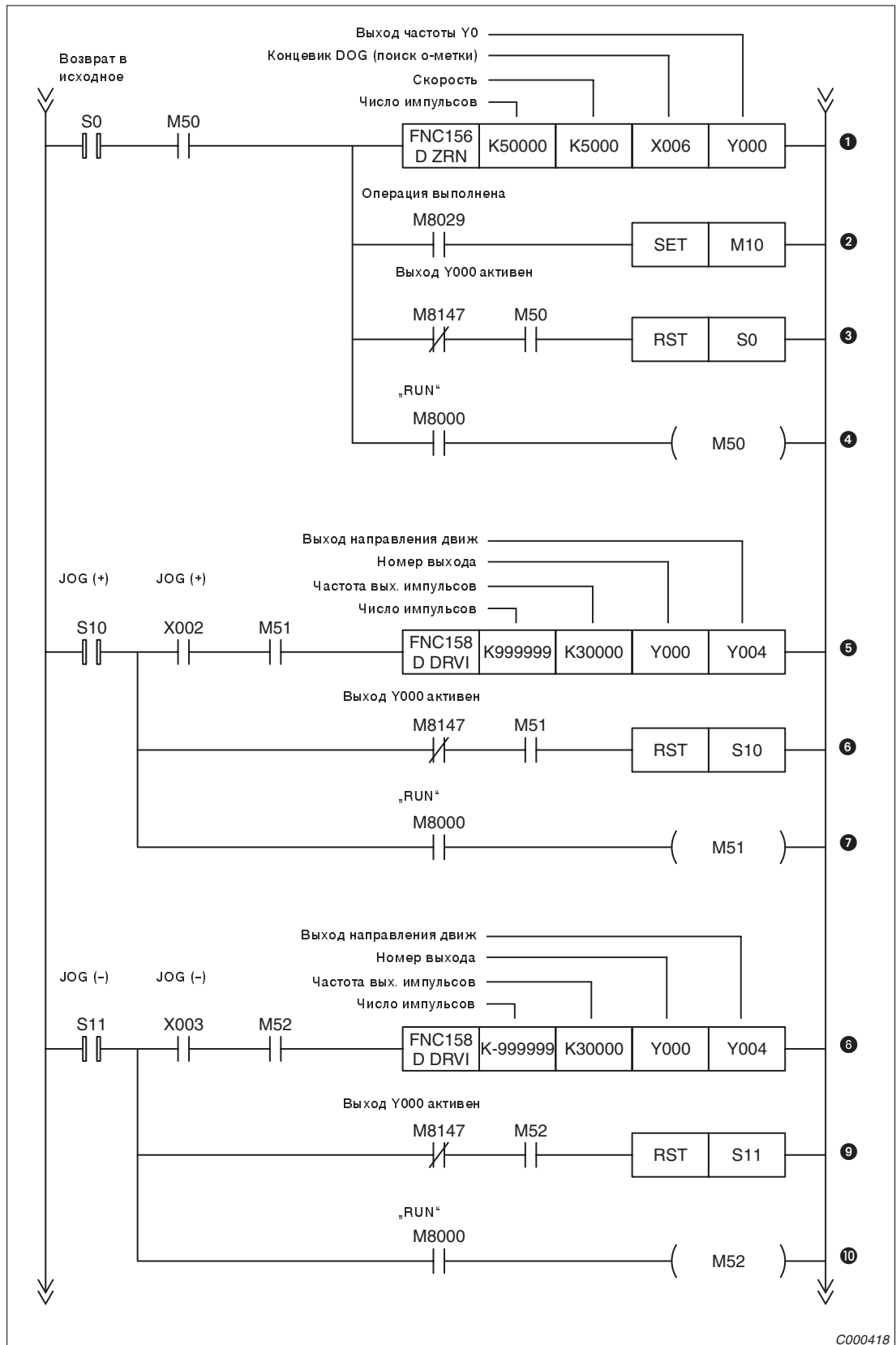


Рис. 7-67. Пример программы

- 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.

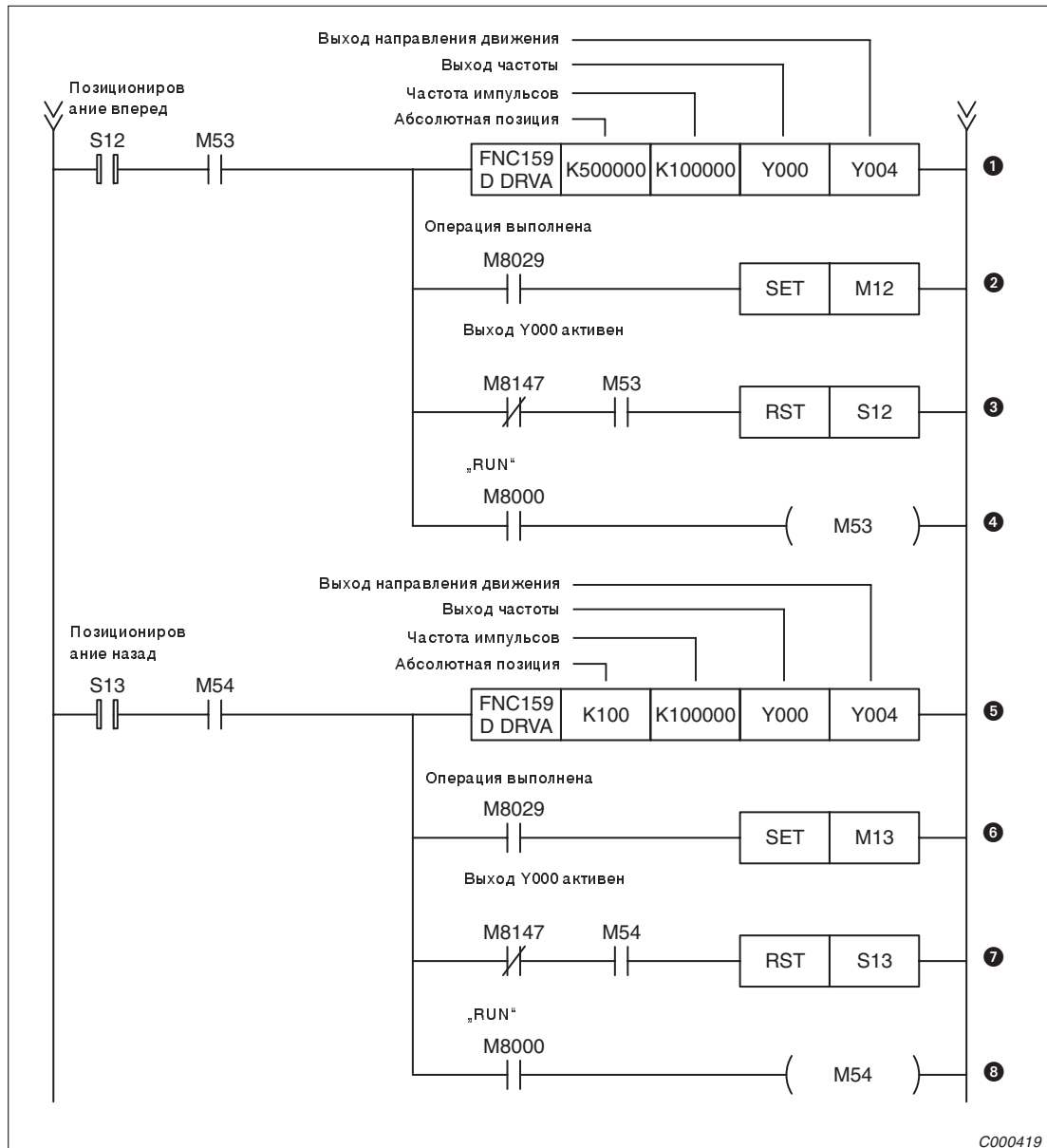
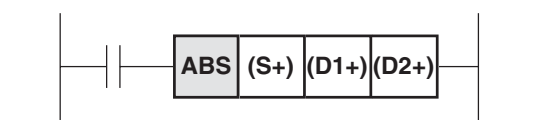


Рис. 7-68. Пример программы

- ❶ Перемещение в абсолютную позицию 500,000 используя инструкцию позиционирования в абс. координатах (Y004 включен).
- ❷ Флаг завершения позиционирования в направлении +.
- ❸ Позиционирование в направлении + завершено (самосброс).
- ❹ Задержка выполнения инструкции на время одного цикла предотвращает одновременное выполнение инструкций позиционирования.
- ❺ Перемещение в абсолютную позицию 100 используя инструкцию позиционирования в абс. координатах (Y004 выключен).
- ❻ Флаг завершения позиционирования в направлении -.
- ❼ Позиционирование в направлении - завершено (самосброс).
- ❽ Задержка выполнения инструкции на время одного цикла предотвращает одновременное выполнение инструкций позиционирования.

7.6.6 ЧТЕНИЕ АБСОЛЮТНОЙ ТЕКУЩЕЙ ПОЗИЦИИ (DABS)

				DABS		FNC 155		
				Чтение абсолютной текущей позиции				
CPU				FX1S	FX1N	FX2N	FX2NC	
				●	●			
Операнды	S+	D1+	D2+	Имп. инструкция (P)	Обработка		Шаги программы	
	X, Y, M, S	Y, M, S	KnY, KnM, KnS, T, C, D, V, Z			16 Bit	32 Bit	DABS
						●		

ФУНКЦИЯ

С помощью этой инструкции может восприниматься абсолютная позиция MITSUBISHI сервоусилителей типа MR-H или MR-J2. Сервопривод должен быть настроен на работу с абсолютным датчиком.

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

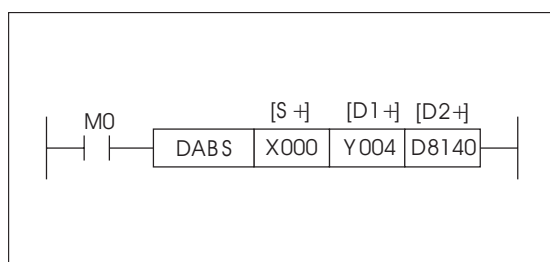


Рис. 7-69.
Пример программирования
ABS-инструкции

C000400C



7.6.7 ПЕРЕМЕЩЕНИЕ К НУЛЕВОЙ (ИСХОДНОЙ) ТОЧКЕ (ZRN)

					ZRN		FNC 156			
					Перемещение к нулевой (исходной) точке					
					CPU	FX1S	FX1N	FX2N	FX2NC	
						●	●			
Операнды	S1+	S2+	S3+	D	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z,			X, Y, M, S		Только Y0 или Y1	16 Bit	32 Bit	ZRN	9
						●	●	DZRN	17	

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 должен быть выполнен возврат к нулевой точке. При словном операнде возможно от 10 до 32,767 Гц, при 32-х битовом операнде возможна область от 10 до 100 кГц.
- 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 выдачи импульсов могут применяться только выходы Y0 и Y1. Так как импульсы выдаются с большой частотой, то ПК должно применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны. Для получения крутого по фронту выходного сигнала нагрузочный ток транзисторного выхода минимум должен составлять 200 мА. Иногда необходимо применять предварительно включенные резисторы.
- 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. 7-21. Операнды для позиционирования

- У MITSUBISHI сервоусилителей типа MR-H или MR-J2, которые в состоянии воспринимать абсолютную позицию, текущая позиция может считываться с помощью ABS-инструкции (FNC 155).

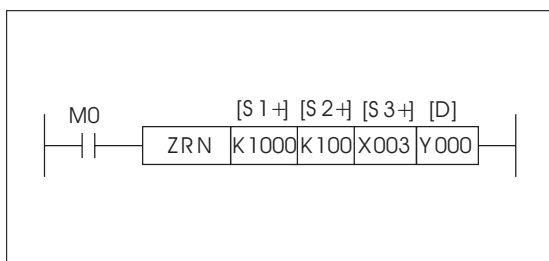
УКАЗАНИЕ

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.

ПРИМЕР ▾

ZRN-инструкции

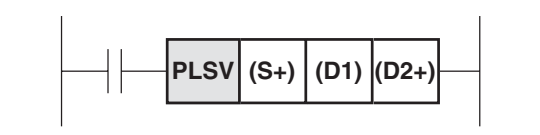
**Рис. 7-70.**

Пример программирования ZRN-инструкции

C000401C



7.6.8 ВЫДАЧА ИМПУЛЬСОВ С ПЕРЕМЕННОЙ ЧАСТОТОЙ (PLSV)

				PLSV		FNC 157			
				Выдача импульсов					
CPU				FX1S	FX1N	FX2N	FX2NC		
				●	●				
Операнды	S+	D1	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Только Y0 или Y1	Y, M, S		16 Bit	32 Bit	PLSV	9	
				●	●	DPLSV	17		

ФУНКЦИЯ

Выдача импульсов с переменной частотой на Y0 или Y1 и сигнала для определения направления вращения

ОПИСАНИЕ

- В S1+ указывается частота выходных импульсов. При словном операнде охватывается область от 10 до 32,767 Гц, при 32-х битовом операнде выдается частота от 10 до 100 кГц.
- Для выдачи импульсов могут применяться только выходы Y0 и Y1. Так как импульсы выдаются с большой частотой, то ПК должно применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны. Для получения крутого по фронту выходного сигнала нагрузочный ток транзисторного выхода минимум должен составлять 200 мА. Иногда необходимо применять предварительно включенные резисторы.
- 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).

- Следующие операнды представлены 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. 7-22. Операнды для позиционирования

УКАЗАНИЕ | При программировании стартовых условий обратите внимание на временные процессы.

ПРИМЕР ▾ PLSV-инструкция

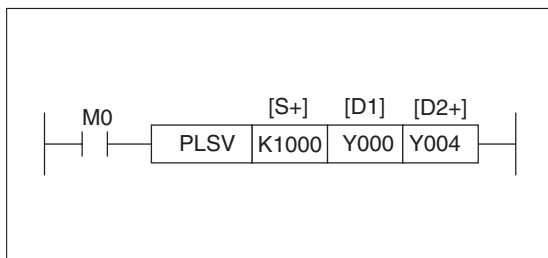
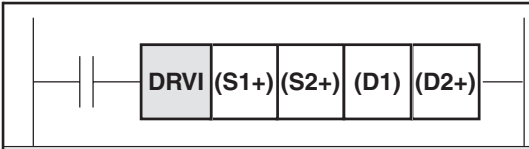


Рис. 7-71.
Пример программирования
PLSV-инструкции

C000402C



7.6.9 ПОЗИЦИОНИРОВАНИЕ ПО ИНКРЕМЕНТАЛЬНЫМ ЗНАЧЕНИЯМ (DRVI)

					DRVI		FNC 158			
					Позиционирование по инкрементальным значениям					
					CPU	FX1S	FX1N	FX2N	FX2NC	
						●	●			
Операнды	S1+	S2+	D1	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z,		Только Y0 или Y1	X, M, S		16 Bit	32 Bit	DRVI	9	
						●	●	DDRVI	17	

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 импульсов выхода. При словном операнде охватывается область от 10 до 32,767 Гц, при 32-х битном операнде возможна выдача частоты от 10 до 100 кГц.
- Для выдачи импульсов могут применяться только выходы Y0 и Y1. Так как импульсы выдаются с большой частотой, то ПК должно применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны.
Для получения крутого по фронту выходного сигнала нагрузочный ток транзисторного выхода минимум должен составлять 200 мА. Иногда необходимо применять предварительно включенные резисторы.
- 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 отработывается и привод останавливается. В этом случае специальный меркер M8029, который показывает конец обработки, не включается.
- После отключения стартовых условий DRVI-инструкция запускается вновь тогда, если специальные меркеры M8147 (контроль Y0) и M8148 (контроль Y1) больше не будут включены.
- Минимальная выходная частота может быть рассчитана по следующей формуле:

$$f_{\min} = \sqrt{\frac{f_{\max}}{2 \times t_B}}$$

f_{\max} является максимальной частотой выходных импульсов, запомненной в D8146 и D8147.

t_B является временем ускорения и замедления в секундах.

- Следующие операнды представляются в распоряжение для задач позиционирования:

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. 7-23. Операнды для позиционирования

УКАЗАНИЕ | При программировании стартовых условий следите за временными процессами.

ПРИМЕР ▾ DRVl-инструкции

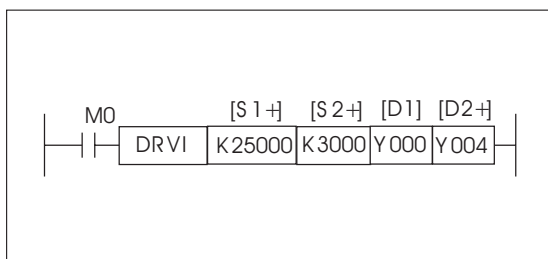
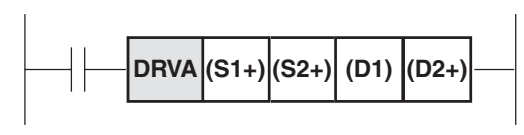


Рис. 7-72.
Пример программирования
DRVl-инструкции

C000403C



7.6.10 ПОЗИЦИОНИРОВАНИЕ ПО АБСОЛЮТНЫМ ЗНАЧЕНИЯМ (DRVA)

					DRVA		FNC 159			
					Позиционирование по абсолютным значениям					
					CPU	FX1S	FX1N	FX2N	FX2NC	
		●	●							
Операнды	S1+	S2+	D1	D2+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z,		Только Y0 или Y1	X, M, S			16 Bit	32 Bit	DRVA	9
						●	●	DDRVA	17	

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 импульсов выхода. При словном операнде охватывается область от 10 до 32,767 Гц, при 32-х битном операнде возможна выдача частоты от 10 до 100 кГц.
- Для выдачи импульсов могут применяться только выходы Y0 и Y1. Так как импульсы выдаются с большой частотой, то ПК должно применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны. Для получения крутого по фронту выходного сигнала нагрузочный ток транзисторного выхода минимум должен составлять 200 мА. Иногда необходимо применять предварительно включенные резисторы.
- 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.
- После отключения стартовых условий DRVA-инструкция запускается вновь тогда, если специальные меркеры M8147 (контроль Y0) и M8148 (контроль Y1) больше не будут включены.
- Минимальная выходная частота может быть рассчитана по следующей формуле:

$$f_{\min} = \sqrt{\frac{f_{\max}}{2 \times t_B}}$$

f_{\max} является максимальной частотой выходных импульсов, запомненной в D8146 и D8147.

t_B является временем 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. 7-24. Операнды для позиционирования

УКАЗАНИЕ | При программировании стартовых условий следите за временными процессами.

ПРИМЕР ▾ DRVA-инструкция

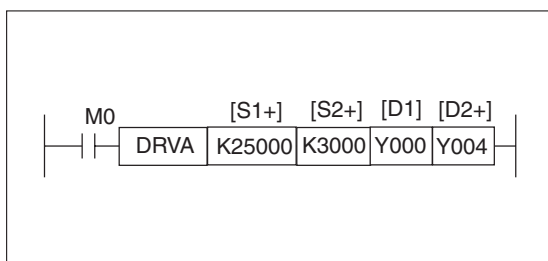


Рис. 7-73.
Пример программирования
DRVA-инструкции

C000404C



7.7 ИНСТРУКЦИИ РЕАЛЬНОГО ВРЕМЕНИ

ОБЗОР ИНСТРУКЦИЙ FNC 160...167

Символ	FNC	Назначение	Разд.
TCMP	160	Сравнение данных времени	7.7.1
TZCP	161	Сравнение данных времени в области	7.7.2
TADD	162	Сложение данных времени	7.7.3
TSUB	163	Вычитание данных времени	7.7.4
TRD	166	Чтение данных времени	7.7.5
TRW	167	Запись данных времени	7.7.6
HOUR	169	Счет рабочих часов	7.7.7

Табл. 7-25. Обзор инструкции FNC 160...167

7.7.1 СРАВНЕНИЕ ДАННЫХ ВРЕМЕНИ (TCMP)

						TCMP		FNC 160			
						Сравнение данных времени					
						CPU	FX1S	FX1N	FX2N	FX2NC	
						●	●	●	●		
Операнды	S1+	S2+	S3+	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
				T, C, D	Y, M, S		●	16 Bit	32 Bit	TCMP	11
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z			Соответственно применяется 3 следующих друг за другом адреса операндов				●			TCMPP

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 времени реального времени быть использованы регистры D8015 (часы), D8014 (минуты), D8013 (секунды) в качестве исходных данных (S1+), (S2+) и (S3+).

ПРИМЕР ▾

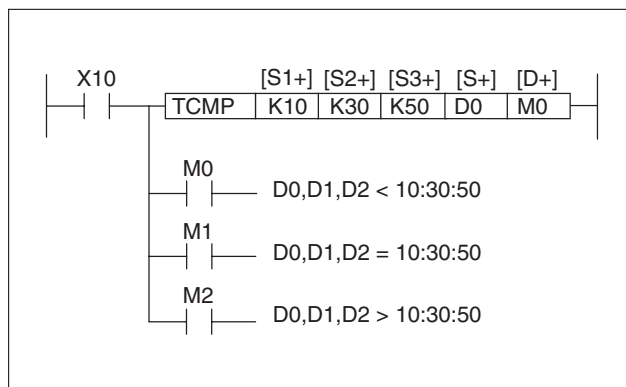


Рис. 7-74.
Пример программирования
TCMR-инструкции

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.

△

7.7.2 СРАВНЕНИЕ ДАННЫХ ВРЕМЕНИ В ОБЛАСТИ (TZCP)

					TZCP		FNC 161			
					Сравнение данных времени в области					
					CPU	FX1S	FX1N	FX2N	FX2NC	
					Операнды					
S1+	S2+	S+	D+	Имп. инструкция(P)						Обработка
	T, C, D ((S1+) J (S2+))		Y, M, S			16 Bit	32 Bit	TZCP 9		
	Соответственно применяется 3 следующих друг за другом адреса операндов				●	●		TZCPP 9		

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).

ПРИМЕР ▾

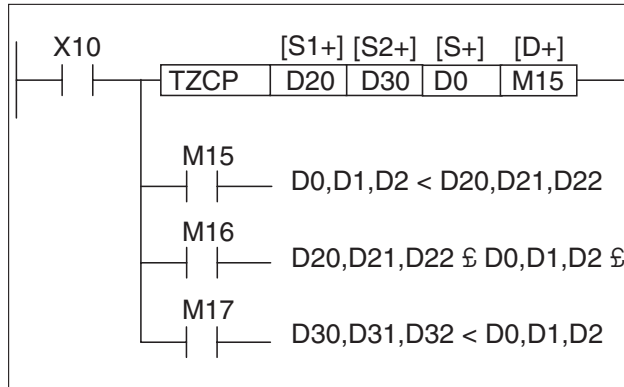


Рис. 7-75.

Пример программирования
TZCP-инструкции

C000365C

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.

△

7.7.3 СЛОЖЕНИЕ ДАННЫХ ВРЕМЕНИ (TADD)

				TADD		FNC 162			
				Сложение данных времени					
				CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	●	●	
Операнды	S1+	S2+	D+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D. Соответственно применяется 3 следующих друг за другом адреса операндов				●	16 Bit	32 Bit	TADD	7
				●			TADDP	7	

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).

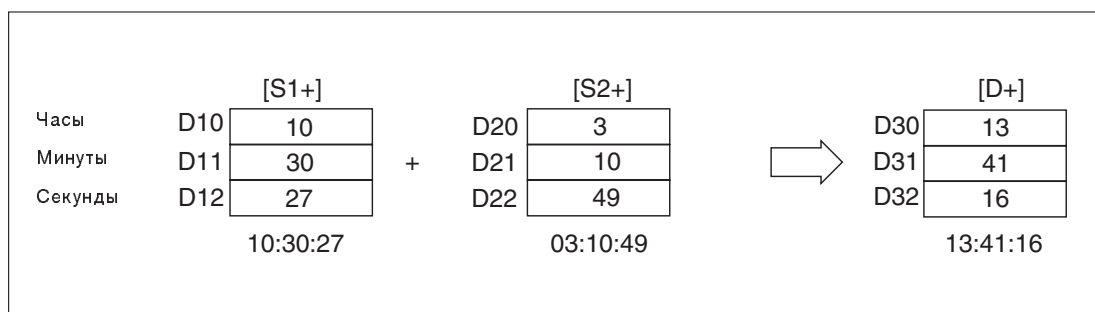


Рис. 7-76. Сложение данных времени

C000366C

- Если результат сложения больше, чем 24 часа, выполняется переключение на 0 часов ("следующий день"), и включается флаг передачи (carry) M8022.

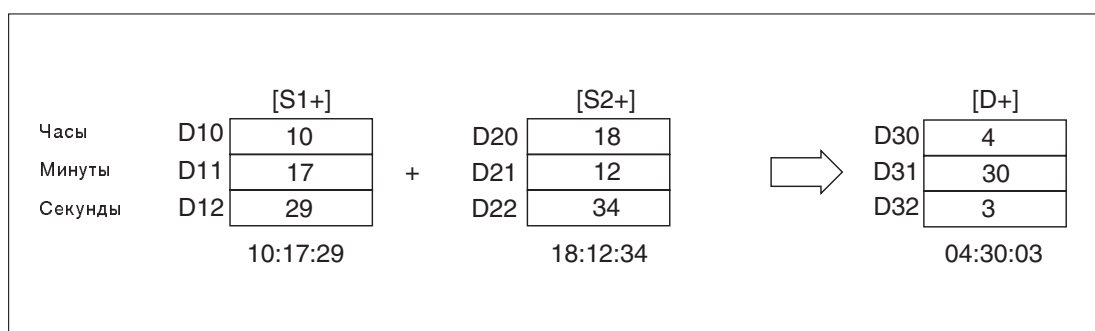


Рис. 7-77. Сложение данных времени с переходом часов

- 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.

ПРИМЕР ▾

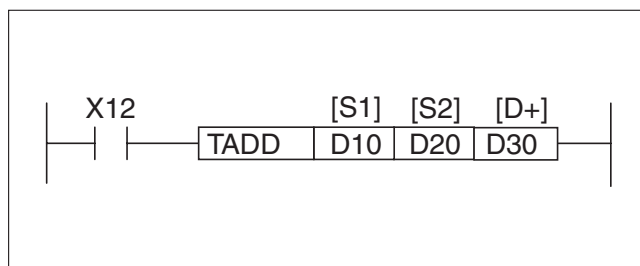


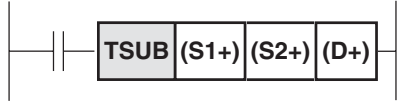
Рис. 7-78.
Пример программирования
TADD-инструкции

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.

△

7.7.4 ВЫЧИТАНИЕ ДАННЫХ ВРЕМЕНИ (TSUB)

				TSUB		FNC 163			
				Вычитание данных времени					
				CPU	FX1S	FX1N	FX2N	FX2NC	
				Операнды S1+ S2+ D+ T, C, D, Соответственно применяется 3 следующих друг за другом адреса операндов				Имп. инструкция(P)	●
16 Bit	32 Bit	TSUB	7						
					●		TSUBP	7	

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)

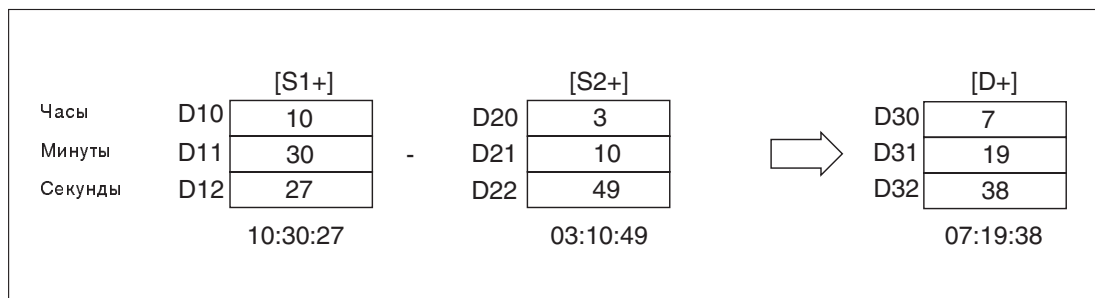


Рис. 7-79. Вычитание данных времени

- Если результат вычитания меньше, чем 0 часов (00:00:00), то остаток от 24 часов заимствуется ("предыдущий день"), и включается флаг заимствования (borrow) M8021.

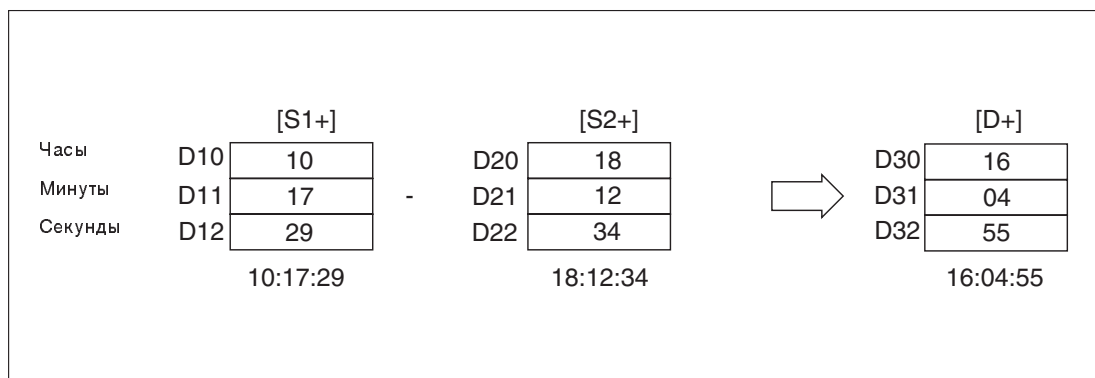


Рис. 7-80. Вычитание данных времени с заимствованием часов

- 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 блокировка.

ПРИМЕР ▾

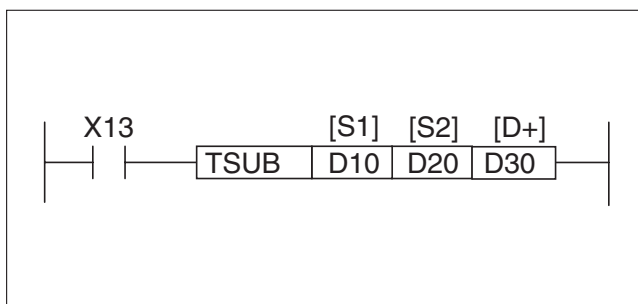


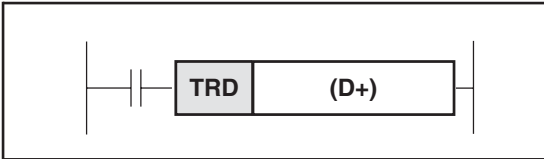
Рис. 7-81.
Пример программирования
TSUB-инструкции

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.

△

7.7.5 ЧТЕНИЕ ДАННЫХ ВРЕМЕНИ (TRD)

		TRD		FNC 166		
		Чтение данных времени				
Операнды		CPU	FX1S	FX1N	FX2N	FX2NC
			●	●	●	●
D+		Имп. инструкция(P)	Обработка		Шаги программы	
T, C, D. Соответственно применяется 7 следующих друг за другом адреса операндов			●	16 Bit	32 Bit	TRD
				●		TRDP

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. 7-26. Чтение данных времени

УКАЗАНИЕ

Число лет читается как двухзначное число. Четырехзначное представление достигается запоминанием значения 2000 в регистре D8018 (см. следующие рисунки).

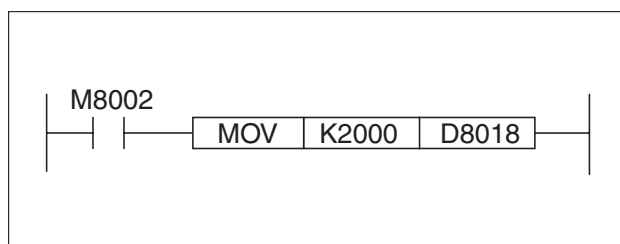


Рис. 7-82.

Пример программирования для описания регистра D8018

C000372C

Много знаковое представление числа лет активизируется только после END-отработки первого цикла программы.

Устройства отображения FX-10DU-E и FX-20DU-E поддерживают двух знаковое представление числа лет.

ПРИМЕР ▾

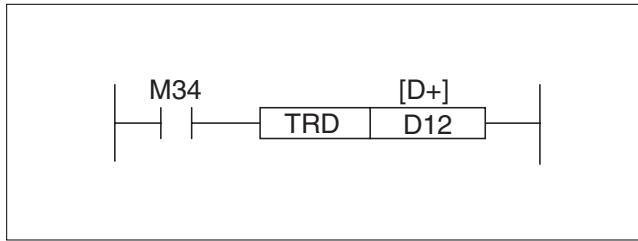


Рис. 7-83.
Пример программирования
TRD-инструкции

C000373C

С включением меркера M34 считываются данные реального времени и запоминаются в регистрах D12...D18.



7.7.6 ЗАПИСЬ ДАННЫХ ВРЕМЕНИ (TWR)

		TWR		FNC 167			
		Запись данных времени					
		CPU	FX1S	FX1N	FX2N	FX2NC	
			●	●	●	●	
Операнды	S+	Имп. инструкция(P)	Обработка		Шаги программы		
	T, C, D. Соответственно применяется 7 следующих друг за другом адреса операндов	●	16 Bit	32 Bit	TWR	5	
			●		TWRP	5	

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. 7-27. Запись данных времени

УКАЗАНИЕ

При применении TWR-инструкции не требуется включение меркера M8015 (сохранение реального времени).

ПРИМЕР ▾

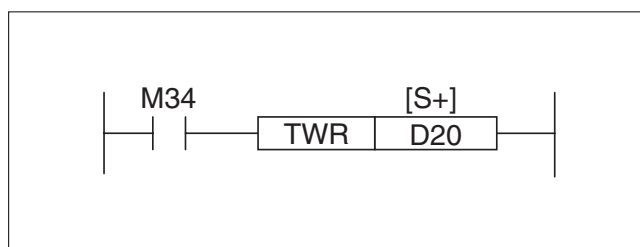


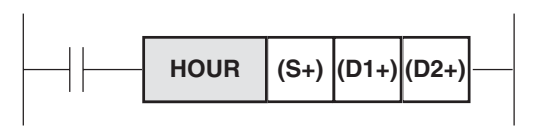
Рис. 7-84.
Пример программирования
TWR-инструкции

C000374C

С включением меркера M34 записываются данные времени из регистра D20...D26 как реальное время.

△

7.7.7 СЧЕТ РАБОЧИХ ЧАСОВ (HOUR)

				HOUR		FNC 169				
				Счет рабочих часов						
				CPU	FX1S	FX1N	FX2N	FX2NC		
Операнды	S+	D1+	D2+	Имп. инструкция(P)	Обработка		Шаги программы			
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z,	D	Z, Y, M, S		16 Bit	32 Bit				
					●	●				

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-х битовая инструкция: (D1+) и (D1+1) содержит данные полных часов. В ((D1+)+2) заносится оставшееся время в секундах.
- Операнд, указанный в (D2+), включается, если текущее (действительное) значение в (D1+) больше, чем заданное значение в (S+).

УКАЗАНИЕ

Чтобы действительное прошедшее время сохранить и при отключении напряжения, для (D1+) должен быть применен регистр с памятью.

Счет также продолжится, после того как операнд, указанный в (D2+) будет включен. Счет остановится, если максимальная область значений 16-ти или 32-х битового операнда в (D1+) будет превышена.

Если сигнал в (D2+) должен включиться в жестком интервале, то действительное значение в регистрах (D1+) до ((D1+)+1) (16-бит) или (D1+) до ((D1+)+2) (32-бита) стираются, если будет получено заданное значение.

ПРИМЕР ▾

HOUR-инструкции

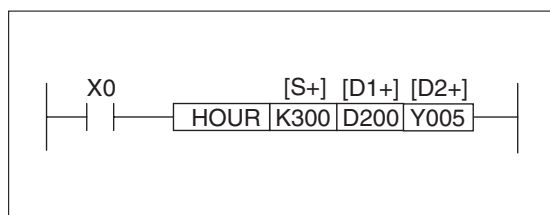


Рис. 7-85.
Пример программирования
HOUR-инструкции

C000405C

После отсчета 300 часов и секунд, после того как будет включен вход X), включится Y5.

△

ПРИМЕР ▾

DHOUR-инструкции

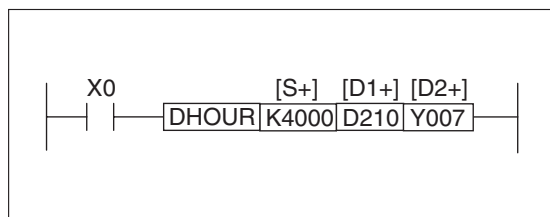


Рис. 7-86.
Пример программирования
DHOUR-инструкции

C000406C

Между включением X0 и Y7 проходит 4000 часов и секунда.

7.8 ИНСТРУКЦИИ КОДА ГРЕЯ

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. 7-28. Обзор инструкции FNC 170...171

7.8.1 ПРЕОБРАЗОВАНИЕ ЦЕЛОГО ЧИСЛА В КОД ГРЕЯ (GRY)

		GRY		FNC 170			
		Преобразование целого числа в код Грея					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
Операнды	S+	D+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V	KnY, KnM, KnS, T, C, D, V, Z	●	16 Bit	32 Bit	GRY, GRYP	5
				●	●	DGRY, DGRYP	9

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 (Stroboskop-Signal).

ПРИМЕР ▾

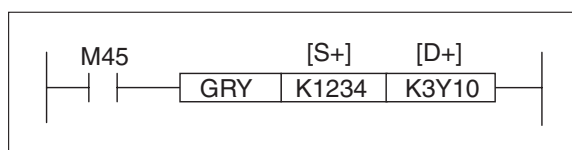


Рис. 7-87.
Пример программирования GRY-инструкции

C000375C

При включении меркера M45 целое значение K1234 конвертируется в код Грея. Результат выдается на выходы Y10...Y23.

△

7.8.2 ПРЕОБРАЗОВАНИЕ КОДА ГРЕЯ В ЦЕЛОЕ ЧИСЛО (GBIN)

		GBIN		FNC 171				
		(S+)	(D+)	Преобразование кода Грея в целое число				
CPU						●	●	
Операнды	S+	K, H, KnX, KnY, KnM, KnS, T, C, D, V	KnY, KnM, KnS, T, C, D, V, Z	Имп. инструкция(P)	Обработка		Шаги программы	
				●	16 Bit	32 Bit	GBIN, GBINP	5
					●	●	DGBIN, DBBINP	9

ФУНКЦИЯ

Конвертирование значения в коде Грея в целое число

ОПИСАНИЕ

- С помощью GBIN-инструкции выполняется конвертирование значения, кодированного в коде Грея в (S+), в целое число.
- Результат заносится в (D+).

УКАЗАНИЕ

Эта GBIN-инструкция может применяться для чтения данных кодированных в коде Грея.

Если для операндов источника применяются входы X0...X17, то время чтения может быть сокращено посредством настройки фильтров активизации (FNC51, REFF)

ПРИМЕР ▾

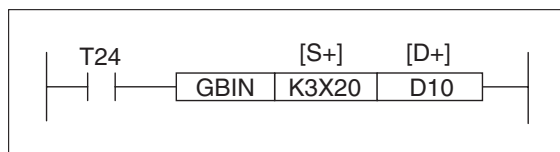


Рис. 7-88.

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.

△

7.9 ОБМЕН ДАННЫМИ С АНАЛОГОВЫМИ МОДУЛЯМИ

Символ	FNC	Назначение	Разд.
RD3A	176	Чтение аналоговых входных значений	7.9.1
WR3A	177	Передача аналоговых выходных значений	7.9.2

Табл. 7-29. Обзор инструкций по обмену данными с аналоговыми модулями

7.9.1 ЧТЕНИЕ АНАЛОГОВЫХ ВХОДНЫХ ЗНАЧЕНИЙ (RD3A)

				RD3A	FNC 176						
				Чтение аналоговых входных значений							
				CPU	FX1S	FX1N	FX2N	FX2NC			
						●	●	●			
Операнды	n1+	n2+	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		●	16 Bit	32 Bit	RD3A	7		
					●						

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

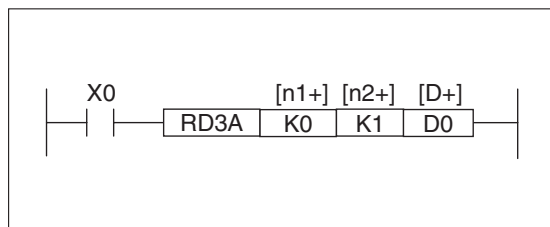



Рис. 7-89.

Пример программирования RD3A-инструкции

C000407C

Аналоговое значение аналогового модуля, установленное в канале 1, передается по адресу 0 в регистр D0.

7.9.2 ЗАПИСЬ АНАЛОГОВЫХ ВЫХОДНЫХ ЗНАЧЕНИЙ (WR3A)

				WR3A		FNC 177			
				Запись аналоговых выходных значений					
				CPU	FX1S	FX1N	FX2N	FX2NC	
						●	●	●	
Операнды	n1+	n2+	S+	Имп. инструкция(P)	Обработка		Шаги программы		
	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	●	16 Bit	32 Bit	WR3A	7	

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

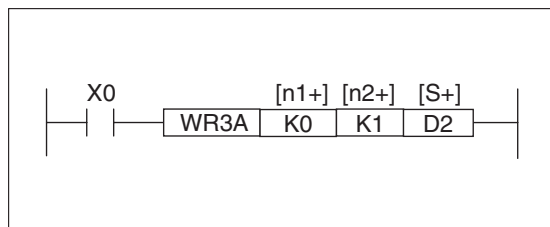


Рис. 7-90.
Пример программирования WR3A-инструкции

C000408C

Содержание регистра D2 передается для выдачи его как аналогового значения к аналоговому модулю с адресом 0.



7.10 ВНЕШНЯЯ КАССЕТА ПАМЯТИ

Символ	FNC	Назначение	Разд.
EXTR	180	команды внешней кассеты памяти ROM	7.10.1

Табл. 7-30. команды внешней кассеты памяти ROM

7.10.1 Функция внешней ROM кассеты (EXTR)

		EXTR	FNC 180				
		команды внешней кассеты памяти					
		CPU	FX1S	FX1N	FX2N	FX2NC	
					●	●	
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	●	16 Bit	32 Bit	EXTR	9
				●	●	DEXTR DEXTRP	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, взводится бит M8065 и вносится в D8065 записывается код ошибки 6512.

EXTR K10 и EXTR K13 (Связь с инверторами)**ФУНКЦИЯ**

Внешние кассеты ROM содержат инструкции 10-13, обеспечивающие считывание и запись данных в инверторы 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 инверторов.
EXTR K13	запись параметров	PLC в INV	

Табл. 7-31. Команды для обмена данных с инвертор MITSUBISHI

УКАЗАНИЕ

Шестиразрядные команды, поддерживаемые инверторами серий E500 и S500, не поддерживаются инструкцией EXTR.

ОПИСАНИЕ

- Если условие работы функции EXTR выполнено, то идет постоянный обмен с инвертором.
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 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 CJ-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 частотным преобразователем

При подключении преобразователя частоты требуются следующие настройки порта:

Параметр		Установка
Стандарт связи		RS 485
Число инверторов 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		Устанавливается параметрами связи

Табл. 7-32. Параметры интерфейса связи инвертора 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 инверторов в сети более 2х, необходимо установить адрес в сети каждого инвертора.	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	0-1/8, 1-2/8, 10-1/7, 11-2/7	Выберите 10
120	Проверка четности	0-отсутствует, 1-Присутствует (нечет), Присутствует (чет)	Выберите 2
121	Число попыток восстановления связи	Установите число попыток восстановления связи после ошибки. Если число попыток превышено инвертор генерирует сбой и останавливается.	При пробном пуске выберите 9999 и выполните настройку. При работе выберите значение в соответствии с характеристиками системы.
122	Интервал времени проверки связи	Устанавливается интервал времени проверки связи. Если связь отсутствует в течение контрольного времени инвертор генерирует сбой и останавливается. См. стр. 195.	
123	Время ожидания	Установите время ожидания между передачей данных и ответом	Выберите 9999
124	Выбор CR, LF	0-CR&LF отсутствуют, 1-CR присутствует, 2-CR&LF присутствуют	Выберите значение в соответствии с параметрами системы

Табл. 7-33. Параметры для инверторов A-500 и E-500

параметр	Описание	Содержимое	Установки для контроллера
331	Номер станции	Соответствует номеру станции, установленному с устройства программирования, подключенного в разъем PU. Если инверторов в сети более 2х, необходимо установить адрес в сети каждого инвертора.	Установите соответствующие номера станций в программе контроллера.
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	0-1/8, 1-2/8, 10-1/7, 11-2/7	Выберите 10
334	Проверка четности	0-отсутствует, 1-Присутствует (нечет), Присутствует (чет)	Выберите 2
335	Число попыток восстановления связи	Установите число попыток восстановления связи после ошибки. Если число попыток превышено инвертор генерирует сбой и останавливается.	При пробном пуске выберите 9999 и выполните настройку. При работе выберите значение в соответствии с
336	Интервал времени проверки связи	Устанавливается интервал времени проверки связи. Если связь отсутствует в течение контрольного времени инвертор генерирует сбой и останавливается. См. стр. 195.	
337	Время ожидания	Установите время ожидания между передачей данных и ответом	
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	0, 1 или 2	Выберите 1 (только CR)
342	Выбор записи в EEPROM	0 запись в EEPROM 1 запись в RAM	Выберите значение в соответствии с параметрами системы

Табл. 7-34. Параметры для инверторов A500 с опциональным модулем A5NR

- Временные установки должны быть минимальны во избежание проблем при сбое связи. Инвертор будет продолжать вращение двигателя в течении установленного времени, что может вызвать аварию. При обнаружении проблем со связью отключайте инвертор.

параметр	Описание	Содержимое	Установки для контроллера
n1	Номер станции	Соответствует номеру станции, установленному с устройства программирования, подключенного в разъем PU. Если инверторов в сети более 2х, необходимо установить адрес в сети каждого инвертора.	Установите соответствующие номера станций в программе контроллера.
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	0-1/8, 1-2/8, 10-1/7, 11-2/7	Выберите 10
n4	Проверка четности	0-отсутствует, 1-Присутствует (нечет), 2-Присутствует (чет)	Выберите 2
n5	Число попыток восстановления связи	Установите число попыток восстановления связи после ошибки. Если число попыток превышено инвертор генерирует сбой и останавливается.	При пробном пуске выберите 9999 и выполните настройку. При работе выберите значение в соответствии с характеристиками системы.
n6	Интервал времени проверки связи	Устанавливается интервал времени проверки связи. Если связь отсутствует в течение контрольного времени инвертор генерирует сбой и останавливается. См. стр. 195.	
n7	Время ожидания	Установите время ожидания между передачей данных и ответом	Выберите 9999
n11	Выбор использования CR/LF	0, 1 или 2	Выберите 1 (только CR)

Табл. 7-35. Параметры для приводов S500-Serie

Формат данных

Пример формата передачи при записи данных из контроллера в инвертор

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

Табл. 7-36. Формат передачи при записи данных из контроллера в инвертор

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 ▾

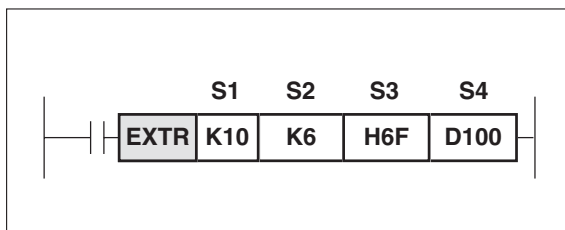


Рис. 7-91.

Пример команды EXTR K10

В этом примере выходная частота инвертора с номером станции 6 читается и сохраняется в регистре D100.

△

Функция	Описание	Обработка		
		16 Bit	32 Bit	Импульсная (P)
EXTR K10	Мониторинг состояния	●		

Операнды		Диапазон параметров
S1	K, H	K10: номер отображаемой функции
S2	K, H, D	Номер станции инвертора (0..31)
S3	K, H, D	Код инструкции инвертора (зависит от модели)
S4	D, KnY, KnM, KnS	Определение области хранения считанного значения

код инструкции	Описание кода инструкции	Возможность использования инструкции в данной серии инвертора			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 инвертора	●	●	●	2	B ⇒ E F
H6E	Считывание заданной частоты (EEPROM)			●	4	B ⇒ E F
H6D	Считывание заданной частоты (RAM)			●	4	B ⇒ E F

Табл. 7-37. Подробности по S3

УКАЗАНИЕ

Вы найдете подробные сведения о кодах инструкции и форматах данных в руководстве по эксплуатации частотного преобразователя.

ПРИМЕР ▾

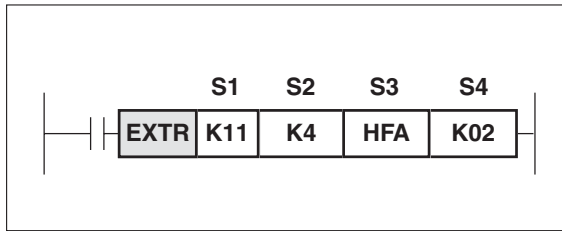


Рис. 7-92.

Пример команды EXTR K11

В этом примере в инвертор с номером станции 4 посылается команда вращения вперед.



Функция	Описание	Обработка		
		16 Bit	32 Bit	Импульсная (P)
EXTR K11	Управление	●		

Операнды		Диапазон параметров
S1	K, H	K11: номер функции для управления инвертором
S2	K, H, D	Номер станции инвертора (0..31)
S3	K, H, D	Код инструкции инвертора (зависит от модели)
S4	K, H, D, KnX, KnY, KnM, KnS	Значение, записываемое в инвертор

код инструкции	Описание кода инструкции	Возможность использования инструкции в данной серии инвертора			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 инвертора	●	●	●	4	A (нет ответа)
HF4	Сброс предыстории аварий	●		●	4	A ⇒ C D
HFC	Сброс всех параметров	●	●	●	4	A ⇒ C D
HFC	Сброс пользователя	●			4	A ⇒ C D

Табл. 7-38. Подробности по S3

УКАЗАНИЕ

Вы найдете подробные сведения о кодах инструкции и форматах данных в руководстве по эксплуатации частотного преобразователя.

ПРИМЕР ▾

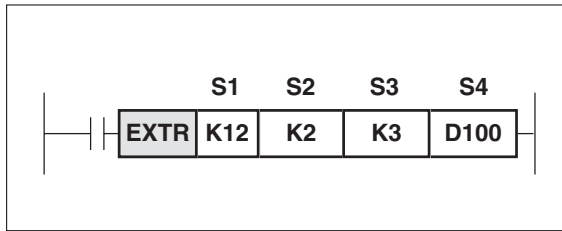


Рис. 7-93.

Пример команды EXTR K12

Считывание базовой частоты (параметр 3) с инвертора станционный номер 2 и сохранение в регистре D100.



Функция	Описание	Обработка		
		16 Bit	32 Bit	Импульсная (P)
EXTR K12	Считывание параметров	●		

Операнды		Диапазон параметров
S1	K, H	K12: номер функции для считывания параметров инвертора
S2	K, H, D	Номер станции инвертора (0..31)
S3	K, H, D	Код инструкции инвертора (зависит от модели)
S4	D, KnY, KnM, KnS	Значение, записываемое в инвертор

ПРИМЕР ▾

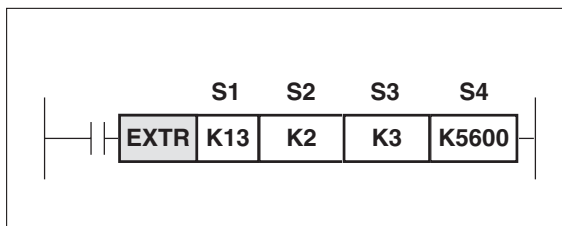


Рис. 7-94.

Пример команды EXTR K13

При этом примере значение 5 600 записывается, как базовая частота (параметр 3) в инвертор с номером станции 2.



Функция	Описание	Обработка		
		16 Bit	32 Bit	Импульсная (P)
EXTR K13	Запись параметров	●		

Операнды		Диапазон параметров
S1	K, H	K13: номер функции для записи параметров в инвертор
S2	K, H, D	Номер станции инвертора (0..31)
S3	K, H, D	Код инструкции инвертора (зависит от модели)
S4	K, H, D, KnX, KnY, KnM, KnS	Значение, записываемое в инвертор

УКАЗАНИЕ

Вы найдете подробные сведения о кодах инструкции и форматах данных в руководстве по эксплуатации частотного преобразователя.

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. 7-39. 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

Табл. 7-40. Считывание и запись смещения/наклона

Определение спец. D-регистров и спец. M-выходов

Операнд	Описание
M8154	используется для отладки
M8155	включен во время связи, выключен после завершения связи
M8156	Включается при возникновении ошибки связи. Эффективен только после выполнения инструкции EXTR. Если выполняется следующая инструкция EXTR, M8156 сбрасывается.
M8157	Активируется при возникновении ошибки связи (с фиксацией). Сбрасывается при переходе из STOP в RUN.
D8154	время ожидания ответа от инвертора
D8155	Номер шага инструкции, выполняющей функцию связи с инвертором. Сохраняет '-1' во время отсутствия связи.
D8156	Код ошибки команды связи. Обновляется, если сбой происходит при последующем выполнении инструкции EXTR. Инициализируется в '-1' при переходе из STOP в RUN.
D8157	номер шага в котором происходит сбой команды связи (фиксируется номер шага для первой ошибки связи после пуска). Инициализируется в '-1' при переходе из STOP в RUN.

Табл. 7-41. Определение спец. D-регистров и спец. M-выходов

Коды ошибок связи

D8156	содержание ошибки	Реакция инвертора
0000H	Связь завершена нормально (ошибок нет)	
0001H	Инвертор не ответил	
0002H	ошибка timeout взаимосвязана с M8129. Ошибка происходит если передача из инвертора 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	Инвертор передал код ошибки H0 - computer NAK error. Данные в запросе содержат ошибку за пределом допустимого числа попыток передачи.	инвертор не принимает данные, не генерирует сбой и останавливается.
0101H	Инвертор передал ошибку H1 -- контроль четности.	
0102H	Инвертор передал ошибку H2 - ошибка контрольной суммы.	
0103H	Инвертор передал ошибку H3 - ошибка протокола. Синтаксическая ошибка в полученных инвертором данных, передача данных не завершена вовремя или ошибка в CR/LF.	
0104H	Инвертор передал ошибку H4 - структура посылки нарушена. Ошибка в числе стоповых битов.	
0105H	Инвертор передал ошибку H5 - переполнение.	
0106H	Инвертор передал ошибку H6.	
0107H	Инвертор передал ошибку H7 - ошибка символов. Получен символ, отличный от 0...9, A...F.	инвертор не принимает данные, не генерирует сбой и не останавливается.
0108H	Инвертор передал ошибку H8.	
0109H	Инвертор передал ошибку H9.	

Табл. 7-42. Коды ошибок связи

D8155	содержание ошибки	Реакция инвертора
010AH	Инвертор передал ошибку HA. Ошибка режима. Не выбран режим связи по интерфейсу.	инвертор не принимает данные, не генерирует сбой и не останавливается.
010BH	Инвертор передал ошибку HB- ошибка кода инструкции. Несуществующая инструкция.	
010CH	Инвертор передал ошибку HC- ошибка данных. При записи использованы данные вне допустимого диапазона.	
010DH	Инвертор передал ошибку HD.	
010EH	Инвертор передал ошибку HE.	
010FH	Инвертор передал ошибку HF.	

Табл. 7-43. 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 инвертора	●	●	●
Если инвертор 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 инвертором не соответствует норме.		●	

Табл. 7-44. Таблица коммуникационных ошибок

ПРИМЕР ▾

Пример программы 1.

Программа считывает параметры 0..99 станции №6 в D1000..D1099 контроллера.

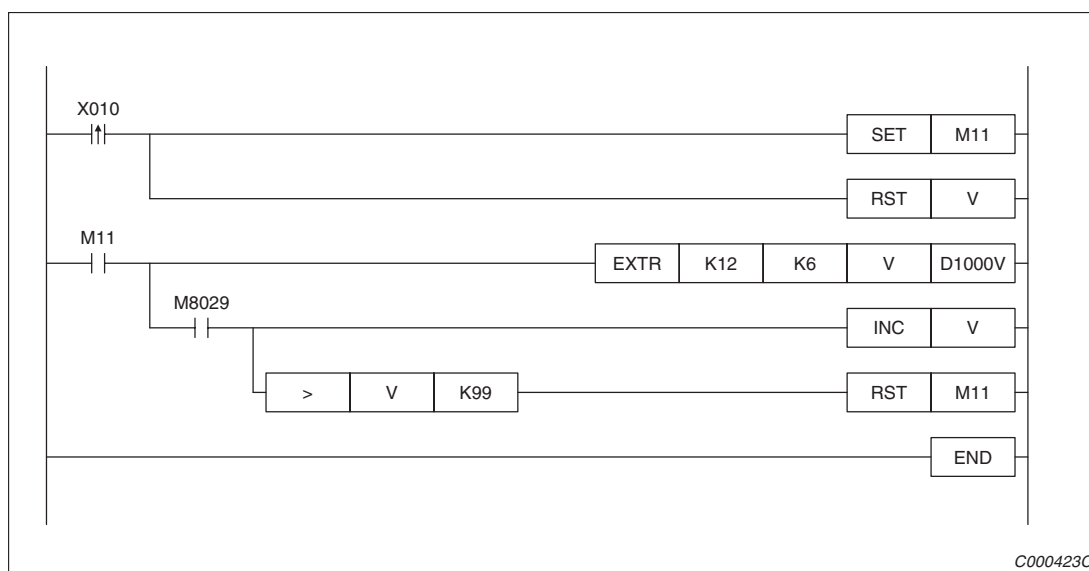


Рис. 7-95. Примерная программа чтения параметров



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.

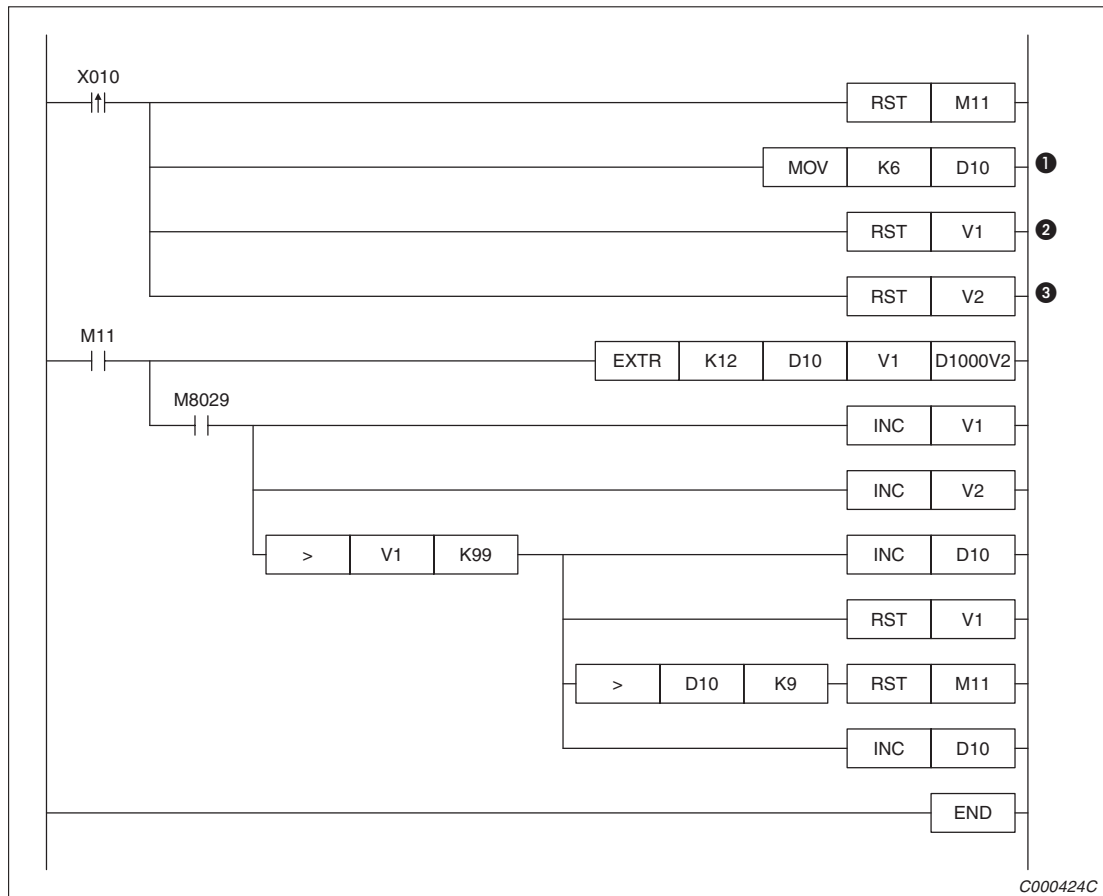


Рис. 7-96. Пример программу чтения параметров нескольких преобразователей

- ① Регистр D10 - номер станции
- ② Номер параметра
- ③ Значение параметра



ПРИМЕР ▾

Пример программы 3

Программа записывает параметр скорости из контроллера в инвертор, выполняет вращение вперед по входу X1 и реверс по входу X2.

Частота инвертора может быть изменена перезаписью D10.

Программа также отображает частоту и выходной ток инвертора.

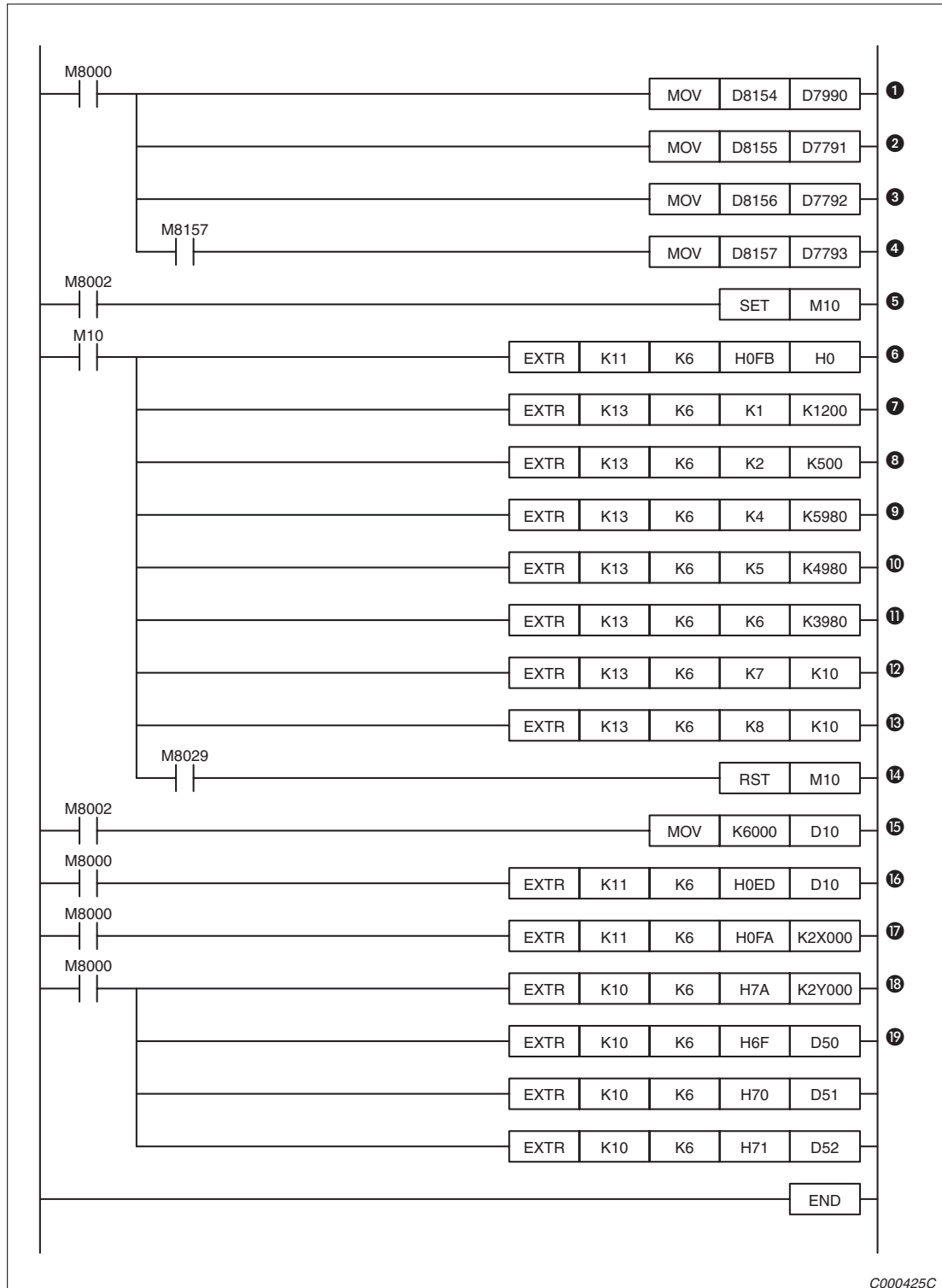


Рис. 7-97. Пример программы по управлению и диагностике преобразователя частоты

В цепочках программы с 1 по 4, сохраняются коды ошибок для более поздней оценки в списке данных.

- ① Ожидание ответа от инвертора
- ② Номер шага текущей выполняемой инструкции
- ③ Код ошибки
- ④ Шаг в котором ошибка произошла впервые
- ⑤ Запись параметров в инвертор 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 инвертор
- 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 инвертор. После окончания передачи в данных в инвертор функция мониторинга продолжается.

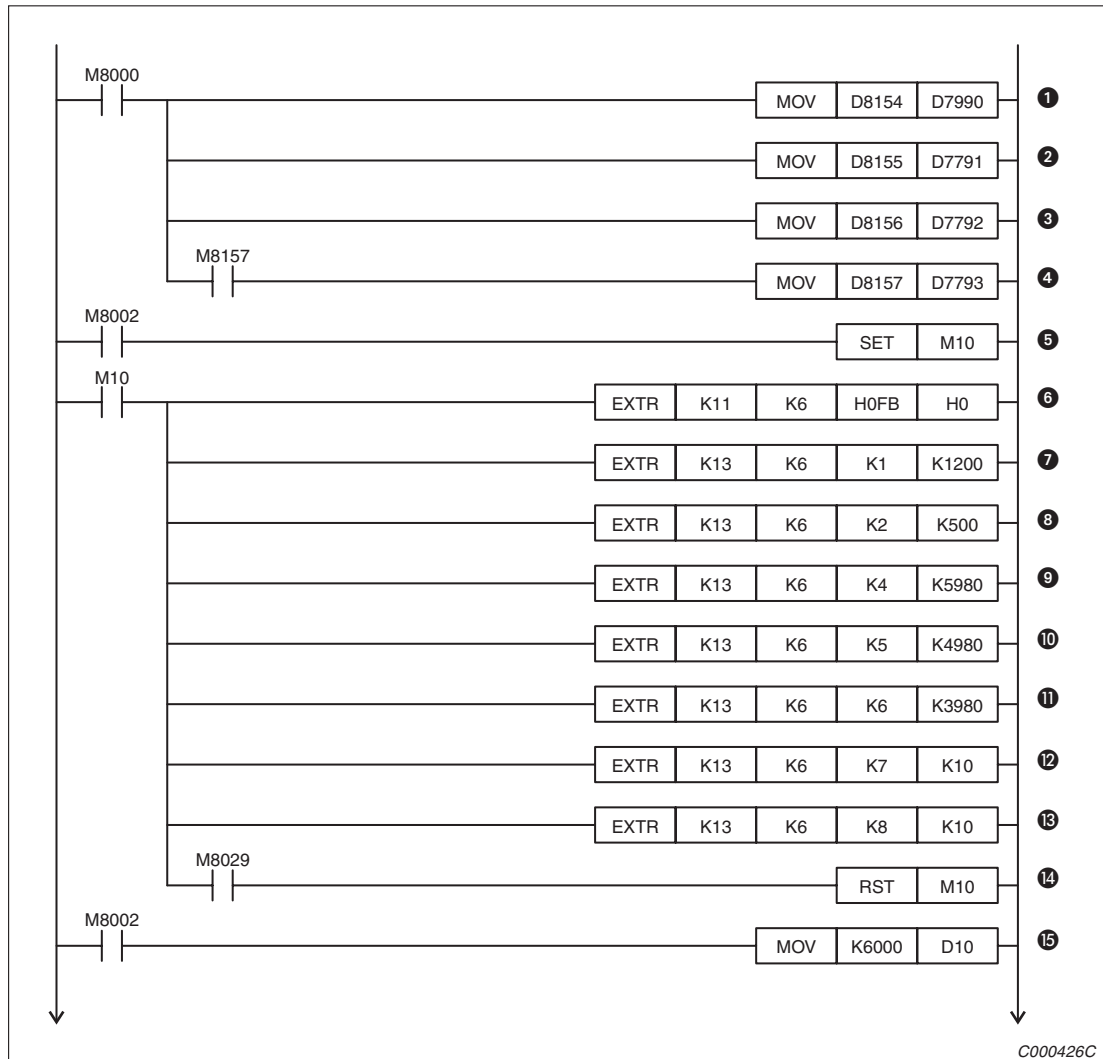


Рис. 7-98. Пример программы с приостановкой функции мониторинга (часть 1)

В цепочках программы с 1 по 4, сохраняются коды ошибок для более поздней оценки в списке данных.

- ① Ожидание ответа от инвертора
- ② Номер шага текущей выполняемой инструкции
- ③ Код ошибки
- ④ Шаг в котором ошибка произошла впервые
- ⑤ Запись параметров в инвертор 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

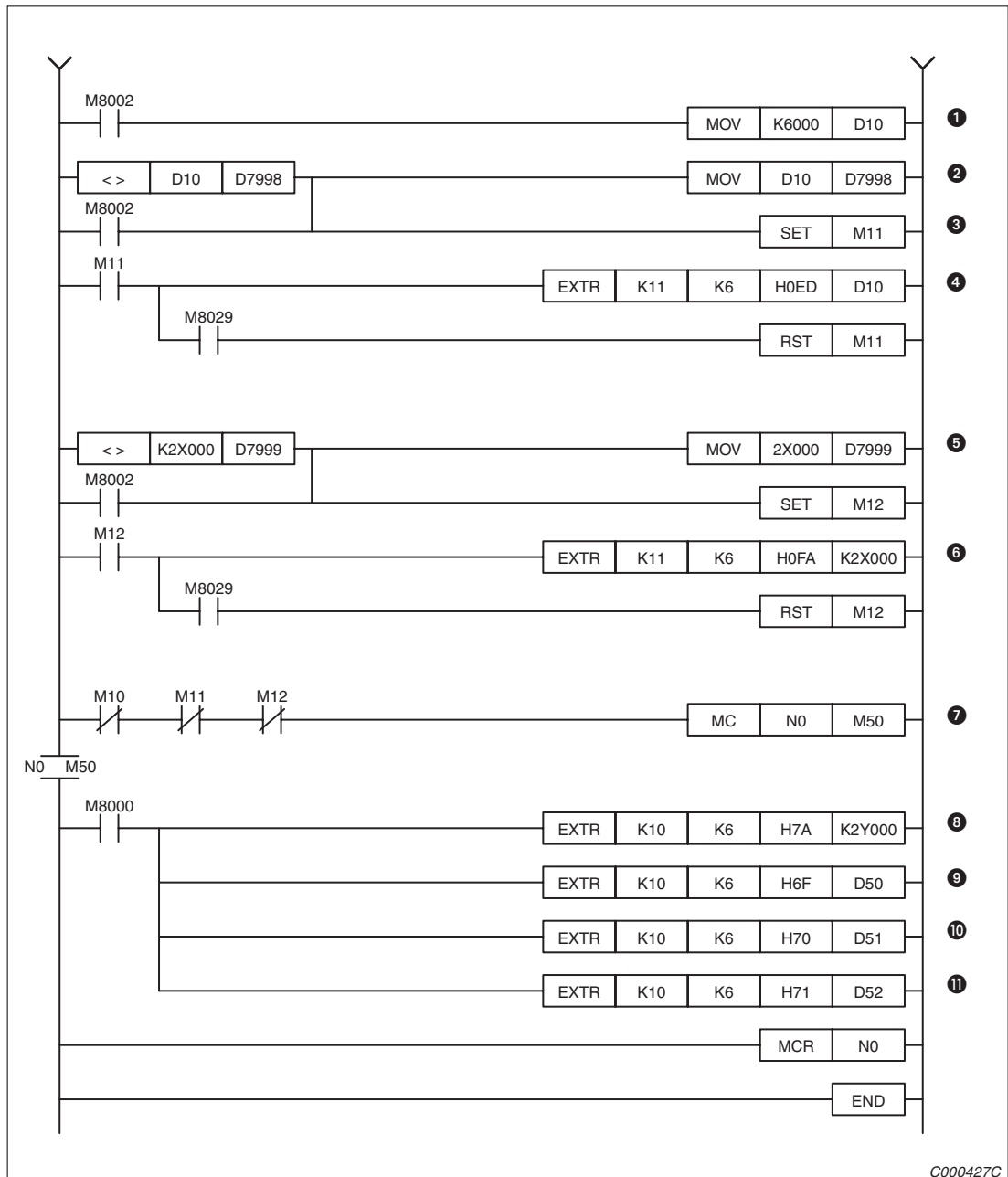


Рис. 7-99. Пример программы с приостановкой функции мониторинга (часть 2)

- ① Выполняет запись только если D10 изменен
- ② Извлечение для определения изменений
- ③ Запись частоты
- ④ Выполняется запись в инвертор, только если состояние одного из X0..X7 изменено.
- ⑤ Извлечение для определения изменений
- ⑥ Отображение состояния, если в инвертор 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

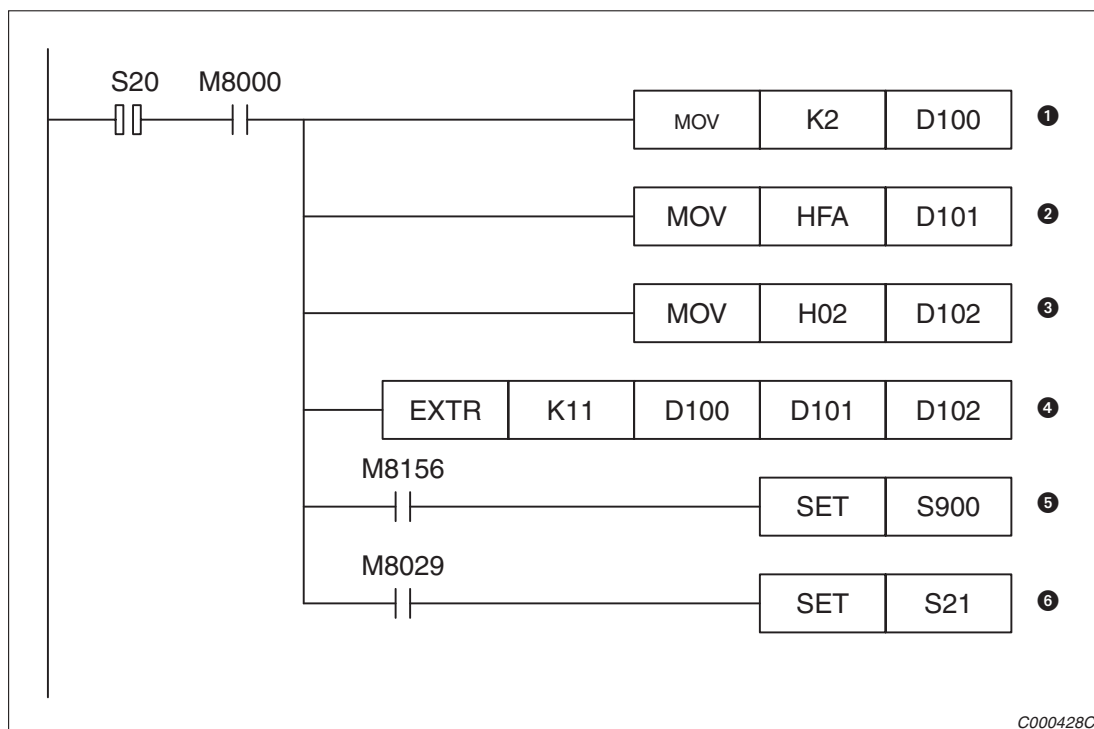


Рис. 7-100. Пример с вызовом функции EXTR в пошаговой программе

- ① Определение станции №2
- ② Код инструкции для рабочей команды
- ③ Команда вперед
- ④ Передача/прием команды в/из инвертора.
- ⑤ Переход в состояние обработки ошибок при возникновении ошибки
- ⑥ Переход в следующее состояние при нормальном получении данных.



7.11 ИНСТРУКЦИИ СРАВНЕНИЯ II

ОБЗОР ИНСТРУКЦИЙ FNC 224...246

Символ	FNC	Назначение	Разд.
LD =	224	Инструкция сравнения, равно	7.11.1
LD >	225	Инструкция сравнения, больше	7.11.1
LD <	226	Инструкция сравнения, меньше	7.11.1
LD <>	228	Инструкция сравнения, не равно	7.11.1
LD ≤	229	Инструкция сравнения, меньше или равно	7.11.1
LD ≥	230	Инструкция сравнения, больше или равно	7.11.1
AND =	232	И-функция инструкции сравнения, равно	7.11.2
AND >	233	И-функция инструкции сравнения, больше	7.11.2
AND <	234	И-функция инструкции сравнения, меньше	7.11.2
AND <>	236	И-функция инструкции сравнения, не равно	7.11.2
AND ≤	237	И-функция инструкции сравнения, меньше или равно	7.11.2
AND ≥	238	И-функция инструкции сравнения, больше или равно	7.11.2
OR =	240	ИЛИ-функция инструкции сравнения, равно	7.11.3
OR >	241	ИЛИ-функция инструкции сравнения, больше	7.11.3
OR <	242	ИЛИ-функция инструкции сравнения, меньше	7.11.3
OR <>	244	ИЛИ-функция инструкции сравнения, не равно	7.11.3
OR ≤	245	ИЛИ- функция инструкции сравнения, меньше или равно	7.11.3
OR ≥	246	ИЛИ- функция инструкции сравнения, больше или равно	7.11.3

Табл. 7-45. Обзор инструкций FNC 224...246

7.11.1 ЗАГРУЗКА СРАВНЕНИЯ (LDI)

		LD□		FNC 224 – 230			
		Загрузка сравнения					
		CPU		FX1S	FX1N	FX2N	FX2NC
				●	●	●	●
Операнды	S1+	S2+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		16 Bit	32 Bit	LD□	5
				●	●	DLD□	9

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
	16 Bit	32 Bit		
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. 7-46. Обзор LD□-инструкций

УКАЗАНИЕ

LD□-инструкция может применяться как одна LD-инструкция (см. пример).

ПРИМЕР ▾

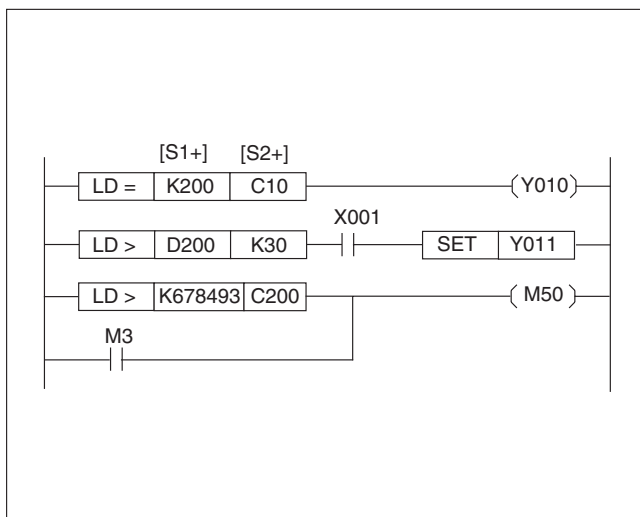


Рис. 7-101.
Пример программирования
LD□-инструкции

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.

△

7.11.2 ЗАГРУЗКА СРАВНЕНИЯ (ANDI)

		AND□		FNC 230 – 238					
		Загрузка сравнения функции "И"							
		CPU		FX1S	FX1N	FX2N	FX2NC		
				●	●	●	●		
Операнды	S1+		S2+		Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V		K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z			16 Bit	32 Bit	AND□	
						●	●	DAND□	

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
	16 Bit	32 Bit		
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. 7-47. Обзор AND□-инструкций

УКАЗАНИЕ

| AND□-инструкция может применяться как одна AND-инструкция

ПРИМЕР ▾

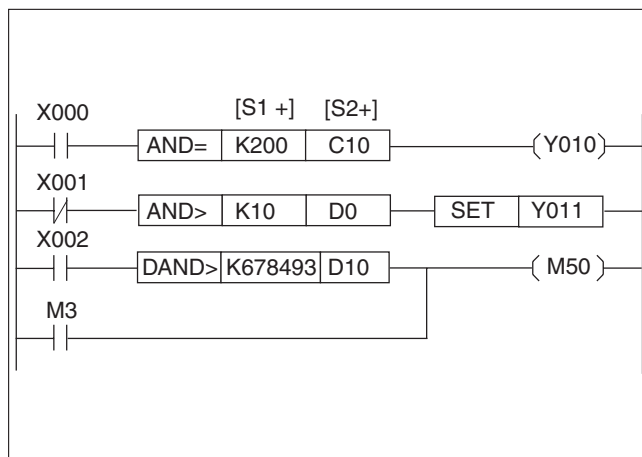


Рис. 7-102.
Пример программирования
AND□-инструкции

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.

△

7.11.3 ЗАГРУЗКА СРАВНЕНИЯ (ORI)

		OR□		FNC 240 – 246			
		Загрузка сравнения функции "ИЛИ"					
CPU		FX1S	FX1N	FX2N	FX2NC		
		●	●	●	●		
Операнды	S1+	S2+	Имп. инструкция(P)	Обработка		Шаги программы	
	K, H, KnX, KnY, KnM, KnS, T, C, D, V	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		16 Bit	32 Bit	OR□	5
	●	●		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
	16 Bit	32 Bit		
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. 7-48. Обзор OR□-инструкций

УКАЗАНИЕ

OR□-инструкция может применяться как одна OR-инструкция (см.примеры).

ПРИМЕР ▾

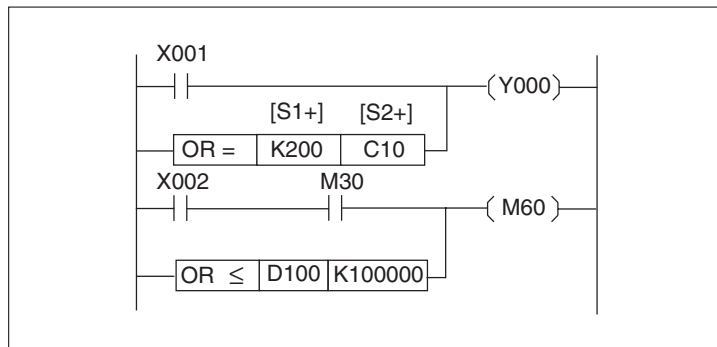


Рис. 7-103.

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.

8 СПЕЦИАЛЬНЫЕ ФУНКЦИИ

Программируемые контроллеры - ПК серии FX представляют в распоряжение пользователя 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 у ПК серии FX1S и FX1N
- Функции реального времени
- Регистры файлов
- Переключение 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.

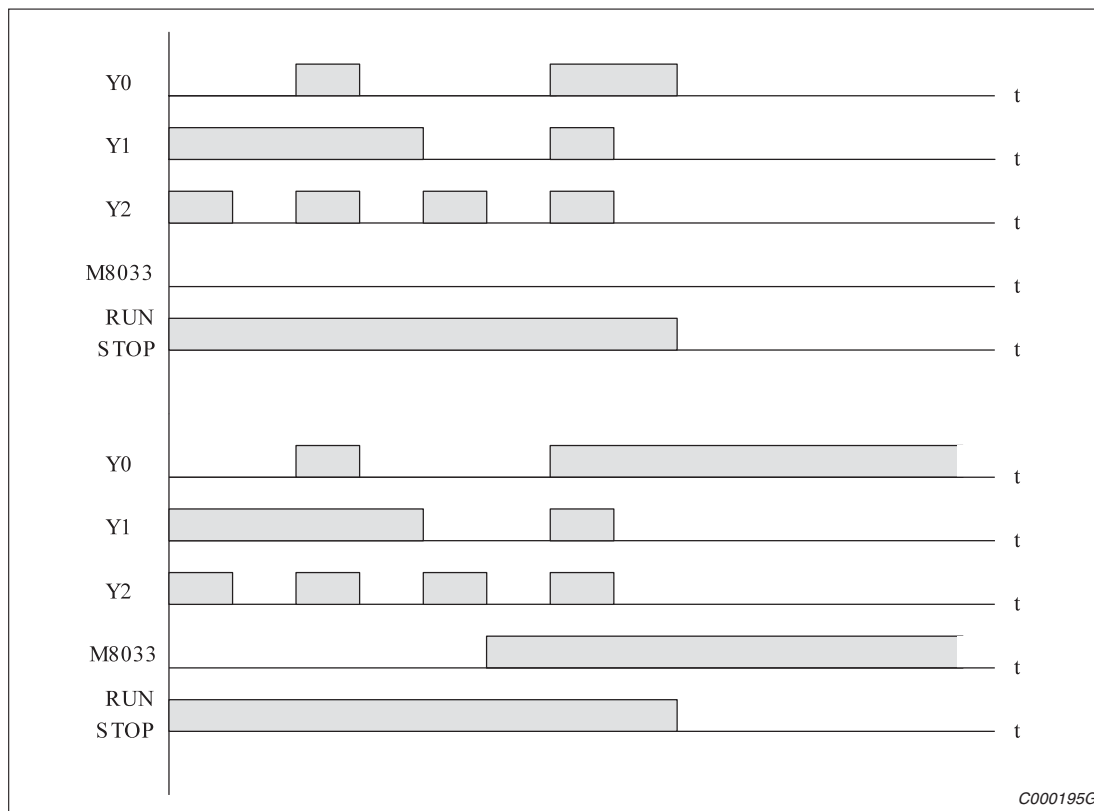


Рис. 8-1. Пример сохранения данных в режиме СТОП

Следующий пример отображает нужную для этого программу:

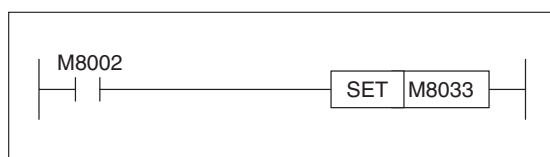


Рис. 8-2.
Пример программирования

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.

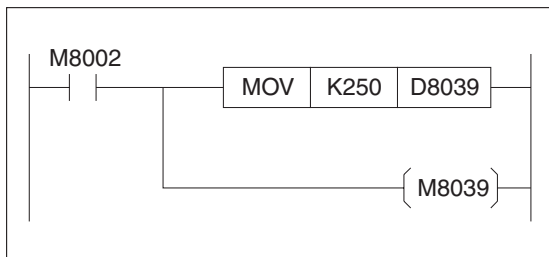


Рис. 8-3.

Определение постоянного времени цикла программы значением 250 мс в регистре данных D8039

C000194C

8.3 ФУНКЦИИ КОДА РАЗРЕШЕНИЯ РАБОТЫ

Благодаря применению кода разрешения работы хранимая 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.

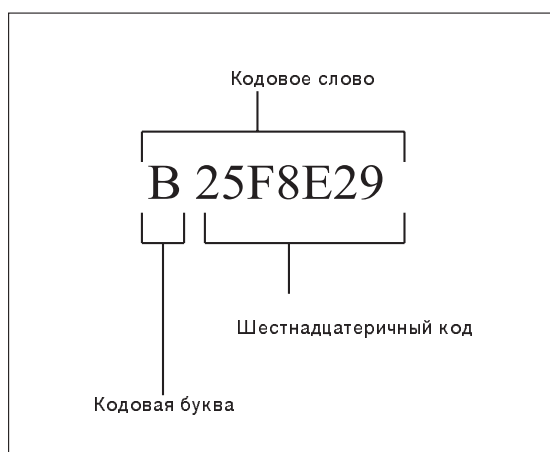


Рис. 8-4. Задание кодового слова

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

С помощью функции фиксации импульсов ПК может обрабатывать очень короткие импульсы, 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

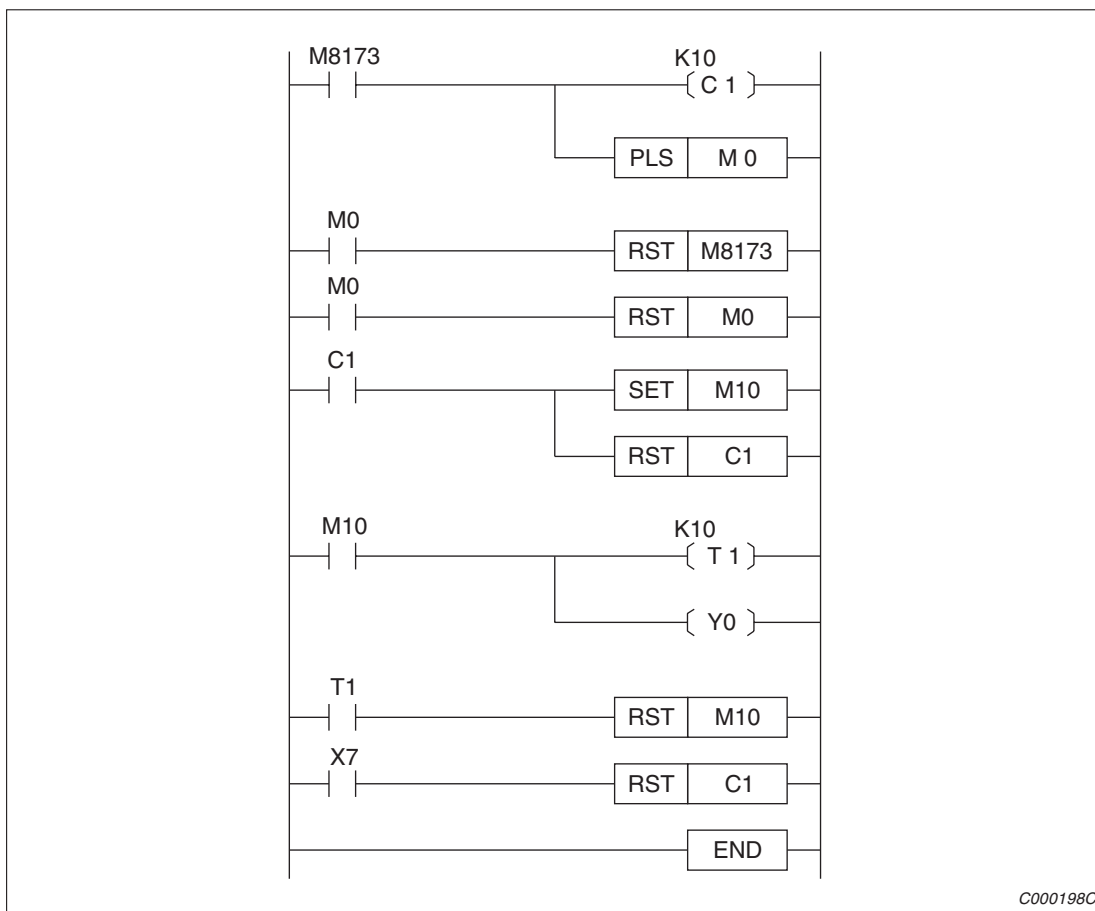
Функция фиксации импульсов не является высокоскоростной функцией (счетчиком). В каждом цикле программы может обрабатываться только один входной импульс.

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).



C000198C

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

FX1S/FX1N

ПК 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.

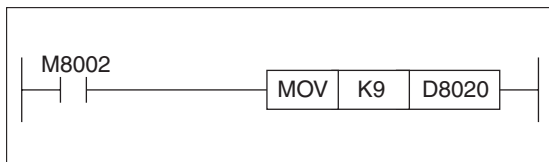


Рис. 8-6.

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 замедление 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.

FX2N/FX2NC

ПК серии FX2N и FX2NC

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 FX1N	Потенциометр 1 (VR1)	D8030
	Потенциометр 2 (VR2)	D8031

Табл. 8-3.
Регистры данных потенциометров

ПРИМЕР ▾

100 мс- таймер с переменным временем

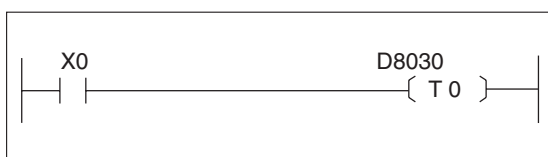


Рис. 8-7.
Таймер со значением уставки, которая может изменяться с помощью потенциометра VR 1

C000196C

Если D8030 = 200, то отсчитывается время 20 с.

Если нужно время более короткое или малыми ступенями, то можно применить таймер на 10 мс.

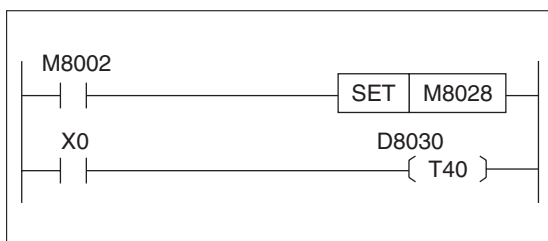


Рис. 8-8.
Таймер на 10 мс с переменным временем (FX1S)

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

ПК серий FX1S, FX1N, FX2N и FX2NC располагают специальными командами для простой установки и использования часов реального времени (более подробная информация содержится в разделе 7.7).

8.8 ФАЙЛОВЫЕ РЕГИСТРЫ

Под файловыми регистрами подразумевают регистры, которые могут применяться как дополнительные, защищенные от отключения напряжения накопителя данных.

Все ПК MELSEC серии FX1N, FX2N и FX2NC 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

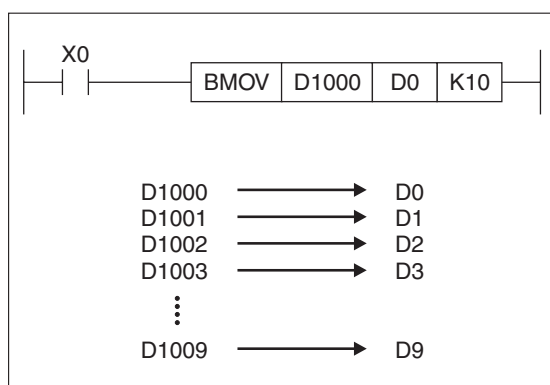


Рис. 8-9.
Чтение файловых регистров

C000195C

8.9 ПЕРЕКЛЮЧЕНИЕ 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 переключкой 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.

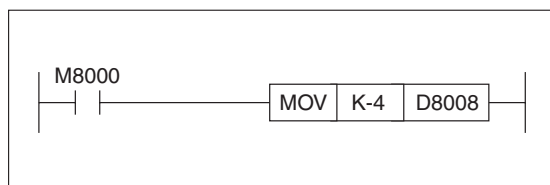


Рис. 8-10.
Пример программирования

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	Freigabe zur Änderung von Operanden
		Mn + 1		Rückmeldung: Änderung abgeschlossen
		Mn + 2	Автоматическое отключение освещения отображения	Automatisches Ausschalten sperren, Beleuchtung einschalten
		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 „-“-Taste
		Mn + 6		Status der „+“-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 ОУТ-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 записываться 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...M8255)

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	Отобразить состояние ПК: RUN (РАБОТА)	
M8001		–		Отобразить состояние ПК: RUN (РАБОТА)	
M8002		–		Импульс инициализации	
M8003		–		Импульс инициализации	
M8004		–		Ошибки ПК	
M8005		–	FX2N FX2NC	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 разрешено
	—	—	FX2N (ab V3.0) FX2NC (ab V3.0)	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	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	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-состояний

- 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	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	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 ФУНКЦИЯ ФИКСАЦИИ ИМПУЛЬСА (ТОЛЬКО FX2N) (M8170...M8175)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8170	●	—	FX1S FX1N FX2N FX2NC	Фиксация импульса X0 ③
M8171	●	—		Фиксация импульса X1 ③
M8172	●	—		Фиксация импульса X2 ③
M8173	●	—		Фиксация импульса X3 ③
M8174	●	—		Фиксация импульса X4 ③
M8175	●	—		Фиксация импульса X5 ③

Табл. 9-7. Специальные меркеры для функции фиксации импульса

- 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 на одном из входов X0 до X5 определен Импульс-сигнал, то включаются выполняемые здесь соответствующие меркеры. После отключения специального меркера, меркер автоматически включается снова с помощью следующего Импульс-сигнала. Таким способом могут опознаваться и запоминаться быстрые входные импульсы.

9.1.8 ФУНКЦИЯ СВЯЗИ И СПЕЦИАЛЬНЫЕ (M8070...M8198)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния 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	●	—		Опознавание, 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	—	—		
M8099	●	●	FX2N/FX2NC	Свободная работа высокоскоростного счетчика
M8109	●	—	FX2N/FX2NC	Ошибка активизированного выхода
M8121	●	—	FX1S FX1N FX2N FX2NC	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	Выбор таблицы сравнения, которая применяется с HSZ-инструкцией
M8131	●	—		Опознавание окончания обработки HSZ-сравнения
M8132	●	●		Определение значения PLSY-инструкции с помощью таблицы HSZ-сравнения
M8133	●	—		Опознавание окончания обработки HSZ-сравнения (при применении PLSY-инструкции)
M8140	●	—	FX1S/FX1N	Очищает импульсный выход в инструкции FNC(156)
M8141- M8144	Зарезервировано			
M8145	—	●	FX1S FX1N	Комманда останова импульсного выхода Y000
M8146	—	●		Комманда останова импульсного выхода Y001
M8147	●	—	FX1S FX1N	Отображение состояния импульсного выхода Y000 (Занят/Готов)
M8148	●	—		Отображение состояния импульсного выхода Y001 (Занят/Готов)
M8160	●	●	FX2N/FX2NC	XCH-инструкция как функция обмена байтами
M8161	●	●	FX1S/FX1N FX2N/FX2NC	Флаг для 8-ми битного режима (RS, ASC, ASCI, HEX, CCD)
M8162	●	●		Высокоскоростной режим параллельного обхода (32-х битный для каждого направления движения)

Табл. 9-8. Специальные меркеры для связи и специальные (1)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8164	●	●	FX2N/FX2NC (ab V 2.00)	Если включен, значение в D8164 используется как число точек обмена для FROM/TO. (начиная с версии CPU 2.00 FX2N/2NC)
M8167	●	●	FX2N FX2NC	Шестнадцатеричный формат для НКУ-инструкции
M8168	●	●		BCD-формат для SMOV-инструкции
M8169	Зарезервировано			
M8170– M8175	●	●	FX1S/FX1N FX2N/FX2NC	Флаг фиксации импульса для входов X0 до X5
M8176– M8182	Зарезервировано			
M8183 M504 bei FX1S	●	—	FX1S FX1N FX2N (ab V2.00) FX2NC (ab V2.00)	Включен при возникновении ошибки в мастер-станции
M8184 M505 bei FX1S	●	—		Включен при возникновении ошибки в 1-ой слэив-станции
M8185 M506 bei FX1S	●	—		Включен при возникновении ошибки в 2-ой слэив-станции
M8186 M507 bei FX1S	●	—		Включен при возникновении ошибки в 3-ей слэив-станции
M8187 M508 bei FX1S	●	—		Включен при возникновении ошибки в 4-ой слэив-станции
M8188 M509 bei FX1S	●	—		Включен при возникновении ошибки в 5-ой слэив-станции
M8189 M510 bei FX1S	●	—		Включен при возникновении ошибки в 6-ой слэив-станции
M8190 M511 bei FX1S	●	—		Включен при возникновении ошибки в 7-ой слэив-станции
M8191 M503 bei FX1S	●	—	Включен при связи с другими станциями	

Табл. 9-9. Специальные меркеры для связи и специальные (2)

- 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.

9.1.9 СЧЕТЧИКИ СЧЕТА ВВЕРХ-ВНИЗ (M8200...M8254)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8200	—	●	FX1N FX2N FX2NC	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. 9-10. 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	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. 9-11. Специальные меркеры для 1-но фазных счетчиков со счетным входом

- 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.

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	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. 9-12. Специальные меркеры для двухфазных счетчиков с двумя счетными входами

А/В-ФАЗНЫЙ СЧЕТЧИК С ДВУМЯ СЧЕТНЫМИ ВХОДАМИ (M8251...M8255)

Специальные меркеры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
M8251	●	—	FX1S FX1N FX2N FX2NC	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. 9-13. 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.

9.2 СПЕЦИАЛЬНЫЕ РЕГИСТРЫ(D8000...D8195)

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:

- ❶ Специальные регистры, значения данных которых программой ПК могут только считываться.
- ❷ Специальные регистры, значения данных которых программой ПК могут считываться и изменяться.

9.2.1 СОСТОЯНИЕ ПК (D8000...D8009)

Специальные регистры адреса операндов	❶ Чтение значений данных	❷ Изменения значений данных	CPU	Значение
D8000	●	●	FX1S FX1N FX2N FX2NC	Таймер контроля времени цикла (Watch-Dog) настраивается в единицах 1 мс. Стандартное значение 200 мс
D8001	●	—		Номер версии FX1S: 22V _{VV} FX1N: 26V _{VV} FX2N: 24V _{VV} (Например, FX1N версия 1.00 → 26100)
D8002	●	—		Емкость памяти: 0002 → 2к-шагов (только у FX1S) 0004 → 4к-шагов (FX2N) 0008 → 8к- или 16к-шагов (FX1N,FX2N)
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	—	—	FX2N FX2NC	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. 9-14. Специальные регистры состояния ПК

- ❶ Специальный регистр, значение данных которого в программе ПК может только читаться.
- ❶ Специальный регистр, значение данных которого в программе ПК может читаться и изменяться.

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	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	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. 9-15. 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	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	Текущее значение данных в индексных регистрах Z0
D8029	●	—		Текущее значение данных в индексных регистрах V0

Табл. 9-16. 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	Установлено постоянное время цикла программы в единицах 1 мс.

Табл. 9-17. 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	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	Последний меркер ошибки. Регистр запоминает последний активный меркер ошибки из области S900 до S999.

Табл. 9-18. 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	Контроль превышения времени отработки цикла программы (Watch-Dog) 500 мс для параллельной связи
D8071 – D8073	—	—	—	Зарезервировано
D8074 bis D8098	—	—	FX2N FX2NC	При выполнении Sampling Trace эти регистры используются контроллером
D8099	—	●	FX2N FX2NC	Кольцевой таймер, устанавливается от 0 до 32767 шагами в 0,1 мс

Табл. 9-19. 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 (D8102...D8109)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8102	●	—	FX1S FX1N FX2N FX2NC	Емкость накопления: 0002 → 2к-шагов (только у FX1S) 0004 → 4к-шагов (FX2N, FX2NC) 0008 → 8к-шагов (FX1N, FX2N, FX2NC) 0016 → 16к-шагов (FX2N, FX2NC)
D8109	●	—	FX2N FX1S/FX1N FX2N/FX2NC	Адрес операнда, у которого появилась ошибка активизации выхода.

Табл. 9-20. 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 (232ADP, 485ADP) (D8120...D8129)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8120	—	●	FX1S FX1N FX2N FX2NC	Формат коммуникации
D8121	—	●		Номер локальной станции (485-сеть)
D8122	●	—		RS, количество переданных остаточных данных
D8123	●	—		RS, количество принимаемых данных
D8124	—	●		RS, головная телеграмма (STX(02 _H))
D8125	—	●		232 ADP, конец телеграммы (ETX(03 _H))
D8127	—	●		RS485, головной адрес требуемой станции
D8128	—	●		RS485, длина данных требуемых данных
D8129	—	●		RS485, установка времени времени выхода таймера сети

Табл. 9-21. 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	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	Количество выданных импульсов посредством PLSY- и PLSR-инструкций (32 бита)
D8138, D8139	●	—	—	Зарезервировано
D8140, D8141	●	—	FX1S FX1N FX2N FX2NC	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. 9-22. 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 (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	Число точек обмена для FROM/TO (начиная с версии 2.00 CPU FX2N/2NC)

Табл. 9-23. Специальные 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.

9.2.11 ИНДЕКС-РЕГИСТРЫ (D8182...D8195)

Специальные регистры адреса операндов	① Чтение значений данных	② Изменения значений данных	CPU	Значение
D8182	●	—	FX1S FX1N FX2N FX2NC	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.

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	SPS-Modus
M8060	●	—	FX2N FX2NC	Ошибка конфигурации входов/выходов	Вых.	RUN
M8061	●	—	FX1S/FX1N FX2N/FX2NC	Ошибка технических средств ПК	Вх.	STOP
M8062	●	—	FX2N FX2NC	Ошибка PC/HPP-коммуникации	Вых.	RUN
M8063	●	—	FX1S FX1N FX2N FX2NC	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	Ошибка шин входов/выходов C	—	—

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 регистры адреса операндов	① Опрос состояния сигналов	② Определение состояния сигналов	CPU	Значение
D8060		–	FX2N FX2NC	Адреса входов/выходов ошибочных базовых и расширенных устройств
D8061		–	FX1S/FX1N FX2N/FX2NC	Номер кода ошибки технических средств ПК записывается в D8061
D8062		–	FX2N FX2NC	Код ошибки для ошибки коммуникации между ПК и программатором (см. соответствующий код ошибки в соответствующей таблице)
D8063		–	FX1S FX1N FX2N FX2NC	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.
D8069		–		Адрес шага ошибки M8065...M8067 запоминается в D8069

Табл. 10-2. Специальные регистры для опознания ошибок

- 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 (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=BKL)	
		6104	Ошибка источника питания 24 В пост.тока (M8069=BKL)	
		6105	Ошибка контроля времени цикла (Watch-Dog)	Время цикла программы больше, чем заданное значение в D8000.
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	
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)	
		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	Остановите 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	
		6409	Другие ошибочные параметры	

Табл. 10-3. Коды ошибок (6101...6409)

10.2.2 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. 10-4. Коды ошибок (6501...6511)

10.2.3 КОДЫ ОШИБОК (6601...6609)

Ошибка	Специальный регистр	Код ошибки	Значение	Устранение ошибки
Ошибка программирования	D8066	0000	Нет ошибки	<p>Ошибки программы появляются, если возникают 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. 10-5. Коды ошибок (6601...6609)

10.2.4 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 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-область: 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)

10.2.5 КОДЫ ОШИБОК (6701...6709)

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.).			

Табл. 10-7. Коды ошибок (6701...6709)

10.2.6 КОДЫ ОШИБОК (6730...6752)

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.
		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)	Команды PID-регулирования должны отключаться раньше, чем продолжиться выполнение.
		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	
		6750	Заданное значение - действительное значение < 150 или регулировочный контур нестабилен (Разность между заданным и действительным значениями очень сильно колеблется и очень быстро).	Колебания не могут выравниваться PID-регулятором. Выполнение PID-команды окончено. PID-команда должна отключиться.
6751	Заданное значение значительно намного больше.			
6752	Большие колебания при задании функции автоматической настройки (Auto-Tuning-Funktion)			

Табл. 10-8. Коды ошибок (6730...6752)

УКАЗАНИЕ

| Более подробные детали по кодовым ошибкам PID-инструкции содержит раздел 7.3.8.

А ТЕХНИЧЕСКИЕ ДАННЫЕ

А.1 ОБЗОР БАЗОВЫХ КОМАНД

Инструкция	Символ релейной схемы	Значение	Операнды	Количество шагов программы	Раздел
LD		LADE; Начало логического выражения (связей) с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.2
LDI		LADE NICHT; Начало логического выражения (связей) с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.2
OUT		AUSGABE; Выход. Присвоение результата логического выражения	Y, M, S, T, C	Y, M: 1 S, Спец. меркер: 2 T: 3, C: 3 C(32 Bit): 5	Разд. 4.3
AND		UND; Логическое выражение "И" с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.4
ANI		UND Nicht; Логическое выражение "И" с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.4
OR		ODER; Логическое выражение "ИЛИ" с опросом на состояние сигнала "1"	X, Y, M, S, T, C	1	Разд. 4.5
ORI		ODER Nicht; Логическое выражение "ИЛИ" с опросом на состояние сигнала "0"	X, Y, M, S, T, C	1	Разд. 4.5
LDP		LADE; (импульсно) Начало логического выражения с опросом по переднему фронту	X, Y, M, S, T, G	2	Разд. 4.6
LDF		LADE; (импульсно) Начало логического выражения с опросом по заднему фронту	X, Y, M, S, T, G	2	Разд. 4.6
ANP		UND; (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)

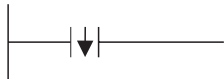

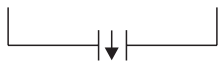
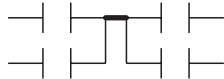

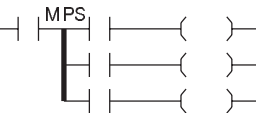
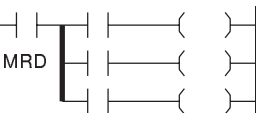
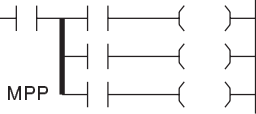
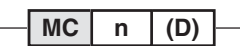



Инструкция	Символ релейной схемы	Значение	Операнды	Количество шагов программы	Раздел
ANF		UND; (импульсно) Логическое выражение "И" с опросом по заднему фронту	X, Y, M, S, T, G	2	Разд. 4.7
ORP		ODER; Логическое выражение "ИЛИ" с опросом по переднему фронту	X, Y, M, S, T, G	2	Разд. 4.8
ORF		ODER; Логическое выражение "ИЛИ" с опросом по заднему фронту	X, Y, M, S, T, G	2	Разд. 4.8
ANB		UND-Block; Команда связи; Последовательное соединение параллельных соединений	–	1	Разд. 4.9
ORB		ODER-Block; Команда связи; Параллельное соединение последовательных соединений	–	1	Разд. 4.10
MPS		Push Down Stack; Запоминание результатов логических связей	–	1	Разд. 4.11
MRD		Read Down Stack; Чтение результатов логических связей	–	1	Разд. 4.11
MPP		Pop Up Stack; Чтение и стирание записанных логических связей	–	1	Разд. 4.11
MC		Master Control; Включение контрольных условий	Y, M, без специальных меркеров	3	Разд. 4.12
MCR		Master Control Reset; Отключение контрольных условий	N	2	Разд. 4.12
SET		Setzen; Включение операнда	Y, M, S	Y, M: 1 S, Спец. меркеры: 2	Разд. 4.13
RST		Rücksetzen; 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		Impulserzeugung; Генерация одного одноразового импульса по переднему фронту	Y, M	2	Разд. 4.14
PLF		Impulserzeugung; Генерация одного одноразового импульса по заднему фронту	Y, M	2	Разд. 4.14
INV		Inversion; Инверсирование результата логического выражения	—	1	Разд. 4.15
NOP		Leerzeile; Пустая строка без выполнения функции	—	1	Разд. 4.16
END		Ende; 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	6 входов		
	FX1S-14MR-ES/UL FX1S-14MR-DS FX1S-14MT-DSS	Y0 - Y3	4 выхода		
	FX1S-14MR-ES/UL FX1S-14MR-DS FX1S-14MT-DSS	X0 - X7	8 входов		
	FX1S-14MR-ES/UL FX1S-14MR-DS FX1S-14MT-DSS	Y0 - Y5	6 выходов		
Входы / Выходы	FX1S-20MR-ES/UL FX1S-20MR-DS FX1S-20MT-DSS	X0 - X13	12 входов		
	FX1S-20MR-ES/UL FX1S-20MR-DS FX1S-20MT-DSS	Y0 - Y7	8 выходов		
	FX1S-30MR-ES FX1S-30MR-DS FX1S-30MT-DSS	X0 - X17	16 входов		
	FX1S-30MR-ES FX1S-30MR-DS FX1S-30MT-DSS	Y0 - Y15	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 MELESEC 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/V 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. А-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	32 Bit	Общее значение	C200 – C219	16 адресов
			Действительное значение занесено в EEPROM	C220 – C234	15 адресов

Табл. А-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	
	Шестнадцатичная	16 бит		0 ... FFFF _H	
		32 бит		0 ... FFFFFFFF _H	

Табл. А-9. Операнды MELESEC 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		
	Меркеры с памятью	M500 – M3071	2572 адреса (разделенные)		
Состояния шагов	Специальные меркеры	M8000 – M8255	256 адресов		
	Инициализация	S0 – S9	10 адресов (разделенные)		
	Общее значение	S0 – S999	1000 адресов		
	Меркеры с памятью	S500 – S999	500 адресов (разделенные)		
Таймер	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	
	1 мс, с памятью	0 – 32,767 с	T246 – T249	4 адреса	
Счетчик	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	D0 – D7999	8000 адресов
			С памятью	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 bis +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. A-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.

В ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ

В.1 ВРЕМЯ ВЫПОЛНЕНИЯ ИНСТРУКЦИЙ ДЛЯ ПК СЕРИЙ FX1S/FX1N

Описание сносок находится в конце раздела.

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				

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
MPS	Сохранение результата логического выражения	—	1	0,5	
MRD	Чтение результата логического выражения			0,55	
MPP	Чтение и стирание результата логического выражения			0,5	
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	□□□	1	0,45	

Табл. В-2. Базовые команды и инструкции шаговых состояний для ПК серии FX1S/FX1N (2)

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+	B	39,5	43,8	—
		16	82,6	22,8	—
PLSY	FNC 57 /S1+ /S2+ /D+	32	100,6	34,9	—
PWM	FNC 58 /S1+ /S2+ /D+	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-11. 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	—
HOUR	FNC 169 /S1+ /D1+ /D2+	16	39,7	38,7	
		32	41,9	40,6	

Табл. В-12. 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-13. 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
LDm	FNC 221 – 230 /S1+ /S2+	16	27,6		—
		32	28,2		
ANDm	FNC 232 – 238 /S1+ /S2+	16	27,6		—
		32	28,2		
ORm	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 включены статические инструкции и нет импульсной инструкции, изменяется значение адреса цели.
- 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	
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. Инструкции 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): M8021
 Cy (Carry): 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	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 ⑩	16	97 + 487n	6,4	—
		32	99 + 962n	6,4	
TO	FNC 79 /n1 /n2 /S+ /n3 ⑩	16	94 + 557n	6,4	—
		32	96 + 1099n	6,4	
RS	FNC 80	16	117,6	18	—
		32			
PRUN	FNC 81 /S+ /D+ ⑪	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	
LDm	FNC 221 – 230 /S1+ /S2+	16	1,52		—
		32	1,84		
ANDm	FNC 232 – 238 /S1+ /S2+	16	1,52		—
		32	1,84		
ORm	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 включены статические инструкции и нет импульсной инструкции, изменяется значение адреса цели.
- ④ “n” указывает число сдвигаемых регистров ($n \leq 512$).
- ⑤ “n” указывает число обрабатываемых бит-операндов ($n \leq 16$ в режиме 16 бит, $n \leq 32$ в режиме 32 бит).
- 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.
- ⑨ “n” указывает адреса операндов, которые обрабатываются MEAN-инструкцией ($1 \leq n \leq 64$).
- ⑩ “n” указывает активизированную область операндов ($8 \leq n \leq 128$ в восьми шагах).
- ⑪ “n” указывает константу времени входного фильтра ($0 \leq n \leq 60$ мс).
- ⑫ Эта инструкция может быть активной максимум 6 раз в течении одного и того же времени.
- ⑬ “n” указывает число штабелируемых элементов ($n \leq 256$ для 16-ти битной обработки, $n \leq 128$ для 32-х битной обработки)
- ⑭ “n” указывает число адресов выходов ($n \leq 64$).
- ⑮ “n” указывает число элементов таблицы данных ($1 \leq m \leq 32$). Для полной SORT-обработки SORT-инструкция выполняется m 1 раз).
- 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.

