

ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ

DVP-SS / SA / SX / ES / EX / EH

Руководство по программированию

Содержание

1. ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ ПЛК	4
Введение	4
1.1. Принципы работы релейно-контактных схем в ПЛК	4
1.2. Различия между релейно-контактной логикой в ПЛК и физическими релейно-контактными электросхемами	6
1.3. Операнды	7
1.4. Виды изображения управляющих инструкций	9
1.5. Преобразование релейно-контактных схем в мнемокод	11
1.6. Оптимизация релейно-контактных схем	15
1.7. Примеры написания программ	17
2. ОПЕРАНДЫ В DVP-PLC	24
2.1. Обзор операндов	24
2.2. Форматы чисел, Константы [K] и [H]	29
2.3. Адресация и назначение входов [X] и выходов [Y]	31
2.4. Адресация и назначение внутренних реле [M]	33
2.5. Адресация и назначение шаговых реле [S]	34
2.6. Адресация и назначение таймеров [T]	35
2.7. Адресация и назначение счетчиков [C]	37
2.8. Адресация и назначение регистров [D], [E], [F]	49
2.9. Указатели [N], [P], [I]	51
2.10. Специальные реле и регистры	54
2.10.1. Специальные реле	54
2.10.2. Специальные регистры	67
2.11. Описание специальных реле и регистров	80
2.11.1. Флаги состояния ПЛК (M1000 – M1003)	80
2.11.2. Сторожевой таймер (D1000)	80
2.11.3. Объем памяти программы (D1002)	81
2.11.4. Проверка грамматических ошибок в программе (M1004, D1004, D1137)	81
2.11.5. Память резервирования данных (M1005 ~ M1007)	81
2.11.6. Флаги сторожевого таймера (M1008, D1008)	81
2.11.7. Монитор времени сканирования (D1010~D1012)	81
2.11.8. Внутренние тактовые импульсы (M1011~M1014)	82
2.11.9. Высокоскоростной таймер (M1015, D1015)	82
2.11.10. Часы реального времени (M1016, M1017, M1076, D1313~D1319)	83
2.11.11. Число π (D1018, D1019)	83
2.11.12. Корректировка времени реакции дискретных входов (M1019, D1020, D1021)	83
2.11.13. Флаги завершения выполнения (M1029, M1030)	84
2.11.14. Код ошибки коммуникации (D1025)	85
2.11.15. Команда очистки памяти данных (M1031, M1032)	85
2.11.16. Сохранение состояния выходов в режиме СТОП (M1033)	85
2.11.17. Запрещение включения выходов Y (M1034)	85
2.11.18. Переключатель режимов РАБОТА/СТОП (M1035, D1035)	85
2.11.19. Настройка коммуникационных портов (M1120, M1136, M1138, M1139, M1143, D1036, D1109, D1120)	85
2.11.20. Задержка коммуникационного отклика (D1038)	89
2.11.21. Постоянная величина времени сканирования (M1039, D1039)	89
2.11.22. Функции аналоговых входов и выходов (D1056~D1059, D1110~D1113, D1116~D1118)	90
2.11.23. Флаги ошибок алгоритма программы (M1067~M1068, D1067~D1068)	90
2.11.24. Низкое напряжение (M1087, D1100)	91
2.11.25. Файловые регистры (M1101, D1101 – D1103)	91
2.11.26. Функциональная карта микропереключателей (M1104 – M1111)	91
2.11.27. Функциональная карта расширения транзисторных выходов (M1112, M1113)	91
2.11.28. Импульсный выход с ускорением/замедлением (M1115 – M1119, D1104)	91
2.11.29. Специальный высокоскоростной импульсный выход (M1133 – M1135, D1133)	94
2.11.30. Детектирование подключения специальных модулей расширения (D1139, D1140, D1142, D1143, D1145, D1146)	94

2.11.31.	BCD-модуль (D1139, D1381 – D1384).....	95
2.11.32.	KEY-модуль (D1145, D1375 – D1380).....	95
2.11.33.	DISP-модуль (D1146, D1385 – D1393)	97
2.11.34.	Описание функции многосекционного импульсного выхода с корректировкой разгона/замедления (M1144 – M1149, M1154, D1032, D1033, D1144, D1154, D1155)	99
2.11.35.	Функция пошагового выполнения программы (M1170, M1171, D1170)	105
2.11.36.	Функция двухфазного импульсного выхода (M1172 – M1174, D1172 - D1177)	105
2.11.37.	Значение потенциометра VR (M1178 – M1179, D1178 - D1179)	106
2.11.38.	Функции модемного соединения (M1184~M1188)	106
2.11.39.	Установка диапазона энергонезависимых реле и регистров (D1200 – D1219).....	107
2.11.40.	Разрешение управления входами X от WPLSoft или программатора (M1304)	108
2.11.41.	Номера специальных модулей расширения (D1320 – D1327)	108
2.11.42.	Параметры режима PLC Link (M1350 – M1354, M1360 – M1519, D1355 – D1370, D1415 – D1465, D1480 – D1491).....	108
2.12.	Коды ошибок	123
3.	БАЗОВЫЕ КОМАНДЫ	125
3.1.	Перечень базовых команд	125
3.2.	Описание базовых команд	127
4.	КОМАНДЫ ПОШАГОВОГО УПРАВЛЕНИЯ	146
4.1.	Инструкции [STL], [RET].....	146
4.2.	Последовательные функциональные диаграммы (SFC)	147
4.3.	Описание инструкций пошагового управления	149
4.4.	Особенности программирования пошагового управления.....	154
4.5.	Типы процессов пошагового управления	155
4.6.	Команда IST	163
5.	ПРИКЛАДНЫЕ ИНСТРУКЦИИ	169
5.1.	Перечень прикладных инструкций	169
5.2.	Структура прикладных инструкций.....	177
5.2.1.	Структура таблиц описания прикладных инструкций.	177
5.2.2.	Ввод прикладных инструкций.	178
5.2.3.	32-х битные инструкции.....	179
5.2.4.	Непрерывное и импульсное выполнение инструкции.....	179
5.2.5.	Типы операндов	180
5.2.6.	Флаги.....	180
5.2.7.	Ограничение числа использования инструкции в программе	180
5.2.8.	Ограничение одновременного выполнения инструкции в программе.....	181
5.3.	Обработка прикладных инструкций	182
5.3.1.	Пословная обработка битовых операндов	182
5.3.2.	Обработка чисел с плавающей запятой.....	183
5.4.	Индексные регистры E, F	185
6.	ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 00-49	187
7.	ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 50-99	251
8.	ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 100-149	353
9.	ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 150-199	408
10.	ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 215-246	464
11.	ПАРАМЕТРЫ КОММУНИКАЦИИ.....	472
11.1.	Встроенные коммуникационные порты	472
11.2.	Описание протоколов коммуникации.....	472

1. ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ ПЛК

Введение

Программируемые логические контроллеры (далее по тексту ПЛК) серии DVP являются идеальным средством для построения высокоэффективных систем автоматического управления при минимальных затратах на приобретение оборудования и разработку системы.

ПЛК работают следующим образом:

шаг 1: чтение состояния внешних входных устройств (переключатели, датчики, клавиатура);

шаг 2: обработка процессором предварительно заданной программы и установка нового состояния выходов.

Программа состоит из последовательности отдельных управляющих инструкций, которые определяют функции управления. ПЛК обрабатывает инструкции последовательно, т.е. одну за другой. Общий проход программы непрерывно повторяется. Время, необходимое для прохода программы называется временем цикла, а проходы программы – циклическим сканированием.

Контроллеры способны работать в реальном масштабе времени и могут быть использованы как для построения узлов локальной автоматики, так и систем распределенного ввода-вывода с организацией обмена данными по RS-485 интерфейсу.

Для удобства отладки и написания программ разработчики предусмотрели пакет программирования, который не требует существенных ресурсов компьютера и является простым инструментом для всех категорий специалистов. Используются три языка программирования: LAD (релейно-контактная логика или лестничные диаграммы), IL (список инструкций), SFC (последовательные функциональные диаграммы).

Настоящее руководство описывает и поясняет все инструкции, операнды, распределение адресов ввода/вывода, которые нужны для написания программы контроллеров DVP серий SS / SA / SX / ES / EX / EN.

Информация по установке, монтажу, вводу в эксплуатацию, обслуживанию и устранению ошибок есть в соответствующих руководствах по эксплуатации.

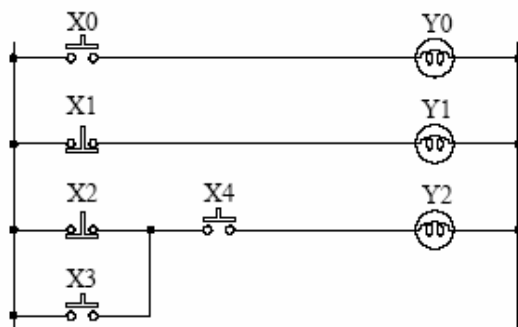
1.1. Принципы работы релейно-контактных схем в ПЛК

Язык релейно-контактной логики в ПЛК (или лестничные диаграммы) является производной от релейно-контактной принципиальной электросхемы в упрощенном представлении. Релейно-контактные схемы в ПЛК имеют набор базовых компонентов, таких как нормально-открытый контакт, нормально-закрытый контакт, катушка (выход), таймер, счетчик и т.д., а также прикладные инструкции: математические функции, команды передачи, обработки данных и большое количество специальных функций и команд. Можно считать, что ПЛК - это сотни или тысячи отдельных реле, счетчиков, таймеров и память. Все эти счетчики, таймеры, и т.д. физически не существуют, а моделируются процессором и предназначены для обмена данными между встроенными функциями, счетчиками, таймерами и др.

Язык релейно-контактной логики в ПЛК по используемой символике очень похож на принципиальные релейно-контактные электросхемы. В релейно-контактных схемах могут быть два типа логики: комбинационная, т.е. схема, состоящая из независимых друг от друга фрагментов, и последовательная логика, когда все шаги программы взаимосвязаны и схема не поддается распараллеливанию.

1. Комбинационная логика:

Релейно-контактная электросхема



Релейно-контактная логика в ПЛК

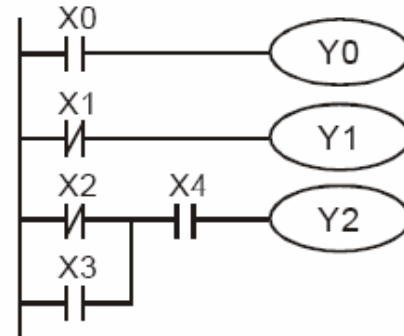


Схема 1 состоит из одного нормально-открытого контакта X0 и катушки Y0, определяющей состояние выхода Y0. При разомкнутом состоянии (логический "0") контакта X0, выход Y0 также будет разомкнут (логический "0"). При замыкании контакта X0 выход Y0 также изменит свое состояние на замкнутое (логическая "1").

Схема 2 состоит из одного нормально-закрытого контакта X1 и катушки Y1, определяющей состояние выхода Y1. В нормальном состоянии контакта X1, выход Y1 будет замкнут (логическая "1"). При изменении состояния контакта X1 на разомкнутое, выход Y1 также изменит свое состояние на разомкнутое.

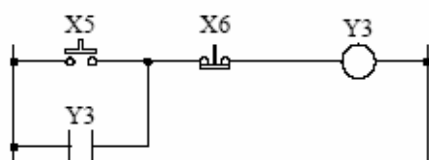
На схеме 3 состояние выхода Y2 зависит от комбинации состояний трех входных контактов X2, X3 и X4. Выход Y2 будет замкнут, когда X2 выключен и X4 включен или когда X3 и X4 включены.

Общая схема является комбинацией трех схем, работающих независимо друг от друга.

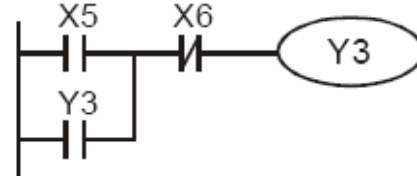
2. Последовательная логика:

В схемах с последовательной логикой результат выполнения предыдущего шага является начальным условием для последующего шага, т.е. выход в предыдущем шаге является входом в следующем шаге.

Релейно-контактная электросхема



Релейно-контактная логика в ПЛК



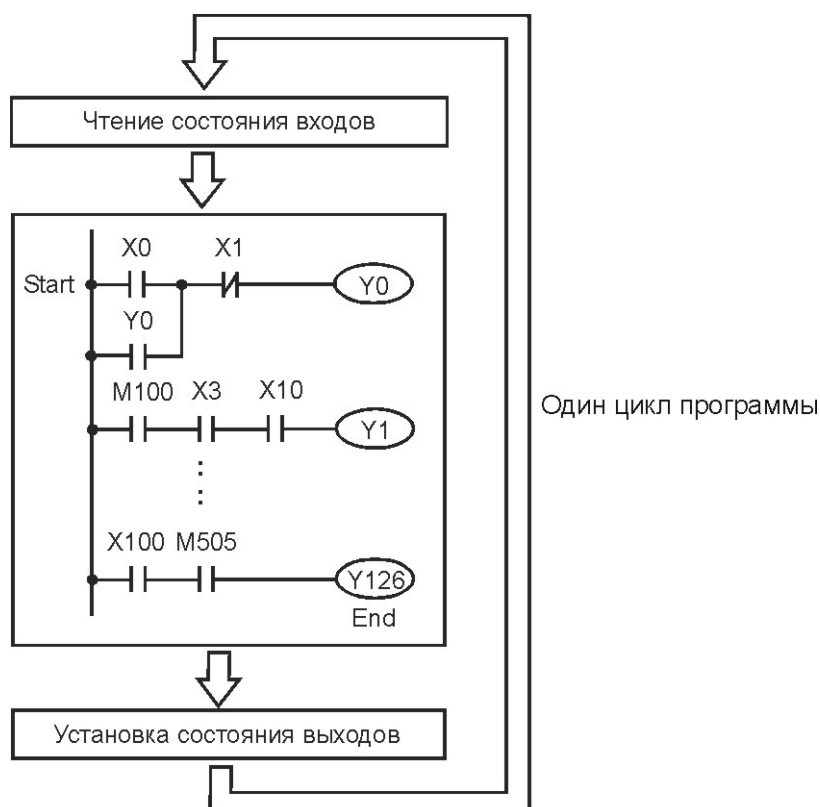
При замыкании контакта X5, выход Y3 изменит свое состояние на замкнутое, однако при размыкании контакта X5, выход Y3 сохранит свое замкнутое состояние до тех пор, пока не будет включен вход X6. Контакт Y3 является самоблокировочным.

1.2. Различия между релейно-контактной логикой в ПЛК и физическими релейно-контактными электросхемами

В обычных релейно-контактных электросхемах все задаваемые управляющие процессы выполняются одновременно (параллельно). Каждое изменение состояние входных сигналов сразу же действует на изменение состояния выходных сигналов.

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

Время выполнения одного цикла программы зависит от количества выполняемых инструкций в программе и от типа используемых инструкций.



В процессе работы ПЛК непрерывно опрашивает текущее состояние входов и в соответствии с требованиями к производственному процессу изменяет состояние выходов(Вкл./Выкл).

1. Проверка текущего состояния входов: ПЛК проверяет текущее состояние входов и в зависимости от значения(Вкл./Выкл) выполняет последовательные действия. Состояние любого из входов сохраняется в памяти (в области данных).

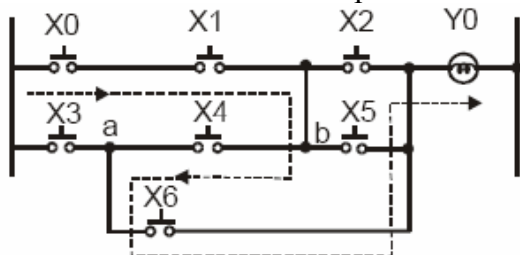
2. Выполнение программы: Будем считать, что в ходе технологического процесса вход X0 переключился с выключено на включено, и в соответствии с технологическим процессом нам необходимо изменить текущее состояние выхода(Y0) с выключено на включено. Так как ЦП опросил текущее состояния всех входов и хранит их текущее состояние в памяти, то выбор последующего действия обусловлен только ходом технологического процесса.

3. Изменение текущего состояния выхода. ПЛК - изменяет текущее состояние выходов в зависимости от того, какие входы являются выключенными, а какие включенными исходя из хода вашей программы. То есть контроллер, физически переключил выход(Y0)

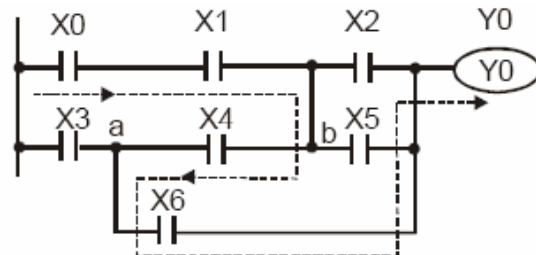
и включились исполнительные механизмы лампочка, двигатель и т.д. После этого следует возврат на первый шаг.

Еще одним отличием релейно-контактной логики ПЛК от обычных релейно-контактных электросхем заключается в том, что выполнение программ в строках идет только слева на право, а схема с "реверсивным направлением тока" (участок a-b на рис.) при компиляции вызовет ошибку.

Релейно-контактная электросхема



Релейно-контактная логика в ПЛК



Ошибка в строке 3

1.3. Операнды

Все внутренние объекты ПЛК, или операнды, подразделяются на различные типы и имеют адреса. Каждый тип имеет свое обозначение и свой формат, который определяет количество занимаемого места в памяти контроллера. Так, например, входные реле обозначаются "X" имеют однобитный формат, а регистры данных общего назначения обозначаются "D" и имеют 16-ти битный (1 слово) или 32-х битный (2 слова) формат.

При указании операнда определяется, с какой операцией (инструкцией) производится работа.

Тип и обозначение операнда		Описание
Вход	X	Входные реле. Определяют состояние внешних битовых устройств, подключенных к входным клеммам ПЛК. Могут принимать одно из двух состояний: 0 или 1. Адресация ведется в восьмеричной системе: X0, X1, ... X7, X10, X11, ...
Выход	Y	Выходные реле. Определяют состояние выходных клемм ПЛК, к которым подключается нагрузка. В программе могут быть как контактами, так и катушками, и принимать одно из двух состояний: 0 или 1. Адресация ведется в восьмеричной системе: Y0, Y1, ... Y7, Y10, Y11, ...
Меркер	M	Внутренние (вспомогательные) реле. Память для двоичных промежуточных результатов. В программе могут быть как контактами, так и катушками, и принимать одно из двух состояний: 0 или 1. Адресация ведется в десятичной системе: M0, M1, ... M7, M8, M9, ...
Состояние шага	S	Управляющие шаговые реле. Используются для

		программирования последовательного управляющего процесса. Могут принимать одно из двух состояний: 0 или 1. Адресация ведется в десятичной системе: S0, S1, ... , S1023
Таймер	T	Реле времени. В программе могут использоваться для хранения текущего значения таймера и иметь 16-ти битный формат, а также могут быть контактами, и принимать одно из двух состояний: 0 или 1. Адресация ведется в десятичной системе: T0, T1, ..., T255
Счетчик	C	Используются для реализации счета. В программе могут использоваться для хранения текущего значения счетчика и иметь 16-ти или 32-х битный формат, а также могут быть контактами, и принимать одно из двух состояний: 0 или 1. Адресация ведется в десятичной системе: C0, C1, ..., C255
Десятичная константа	K	Определение числа в десятичной системе отсчета
Шестнадцатеричная константа	H	Определение числа в шестнадцатеричной системе отсчета
Регистр данных	D	Память данных. 16-ти или 32-х битный формат. Адресация ведется в десятичной системе: D0, D1, ..., D9999. В 32-х битном формате один регистр занимает две ячейки, например при обращении к D10, данные будут прочитаны из ячеек D10 и D11.
Файловый регистр		Используются для хранения данных, когда не хватает регистров данных. Для чтения и записи необходимо использовать специальные инструкции MEMR и MEMW. Операнд не имеет своего символа, а адресация ведется с помощью десятичных констант: K0, K1, ..., K9999.
Индексный регистр	E, F	Память данных для промежуточных результатов и индексной идентификации. 16-ти битный формат. Адресация: E0 – E7, F0 – F7
Указатель	P	Адрес для перехода к подпрограмме.
Указатель прерывания	I	Адрес обработки прерывания.
Номера вложенности	N	Используются для нумерации вложенных схем исключения. N0 – N7.

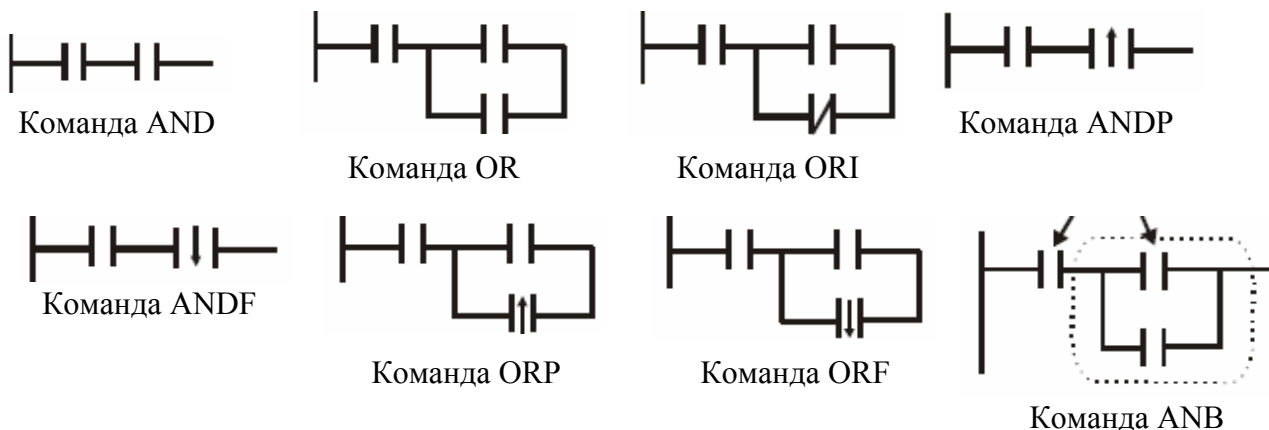
1.4. Виды изображения управляющих инструкций

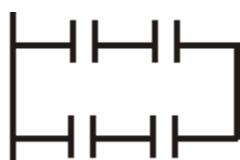
Релейно-контактная схема состоит из одной вертикальной линии, расположенной слева и горизонтальных линий, отходящих вправо. Вертикальная линия называется шиной, а горизонтальная – командной линией или ступенькой. На командной линии располагаются символы условий, ведущие к командам (инструкциям), расположенным справа. Логические комбинации этих условий определяют, когда и как выполняются правосторонние команды. Командные линии могут разветвляться и снова соединяться.

В релейно-контактных схемах в основном применяется следующая символика:

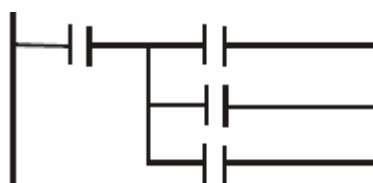
Символ	Пояснение	Команда	Операнд
	Символ для входного сигнала (нормально-открытого контакта а)	LD	X, Y, M, S, T, C
	Символ для входного сигнала (нормально-закрытого контакта b)	LDI	X, Y, M, S, T, C
	Символ для входного импульсного сигнала. (с опросом по переднему фронту)	LDP	X, Y, M, S, T, C
	Символ для входного импульсного сигнала. (с опросом по заднему фронту)	LDF	X, Y, M, S, T, C
	Символ для входного сигнала (шаговый управляющий контакт)	STL	S
	Символ для выходного сигнала (катушки)	OUT	Y, M, S
	Символ для прикладных инструкций	см. главы 3, 5	см. главы 3, 5
	Символ логической инверсии	INV	нет

Входные релейные контакты могут объединяться в последовательные, параллельные и комбинированные схемы:



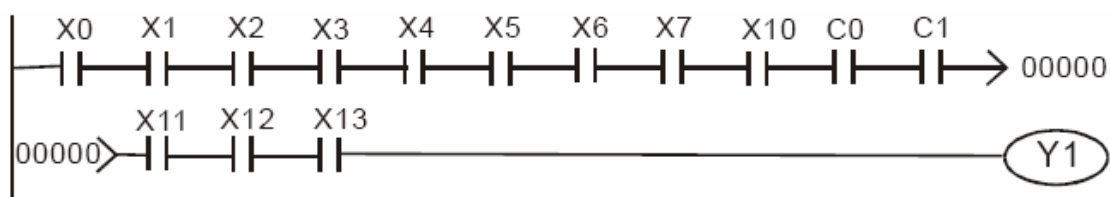


Команда ORB

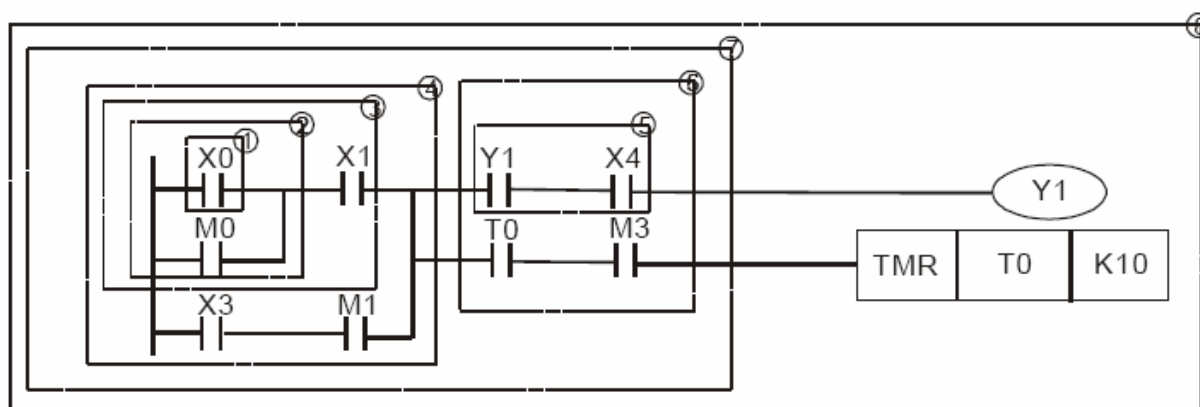


Команды MPS, MRD, MPP

Максимальное количество последовательных контактов в строке – 11. При необходимости использования большего количества, они будут автоматически перенесены на следующую строку:

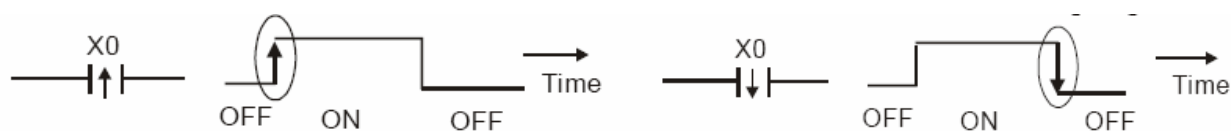


Сканирование программы начинается от левого верхнего угла схемы и заканчивается в правом нижнем углу. Следующий пример иллюстрирует последовательность выполнения программы:

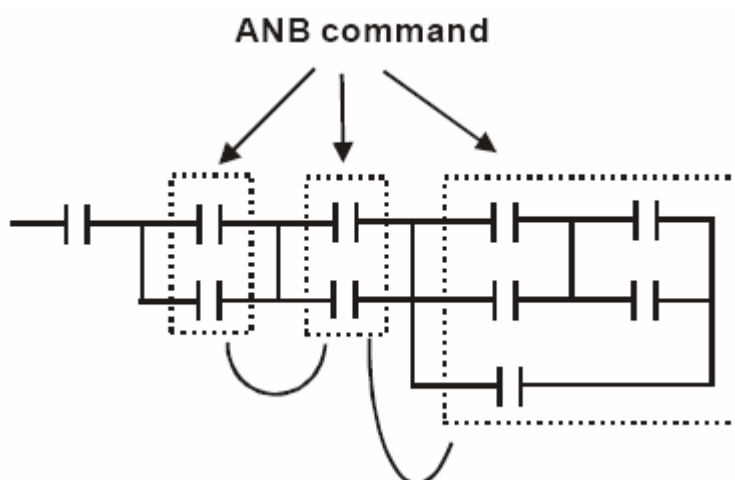


1	LD	X0
2	OR	M0
3	AND	X1
4	LD	X3
	AND	M1
	ORB	
5	LD	Y1
	AND	X4
6	LD	T0
	AND	M3
	ORB	
7	ANB	
8	OUT	Y1
	TMR	T0 K10

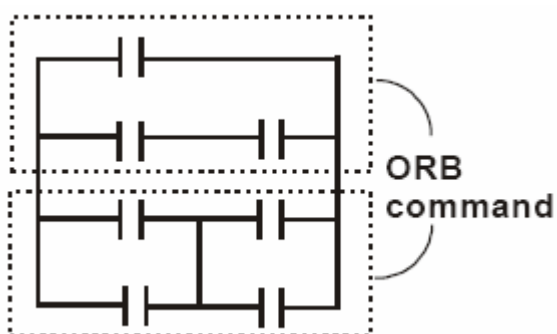
Символы входных сигналов с опросом по переднему фронту (при переходе сигнала с 0 на 1) и с опросом по заднему фронту (при переходе сигнала с 1 на 0) поясняются ниже:



Команды логического блока ANB и ORB не соответствуют конкретным условиям на релейно-контактной схеме, а описывают отношения между блоками. Команда ANB производит операцию ЛОГОЧЕСКОЕ И над условиями исполнения, произведенными двумя логическими блоками.



Команда ORB производит операцию ЛОГОЧЕСКОЕ ИЛИ над условиями исполнения, произведенными двумя логическими блоками.

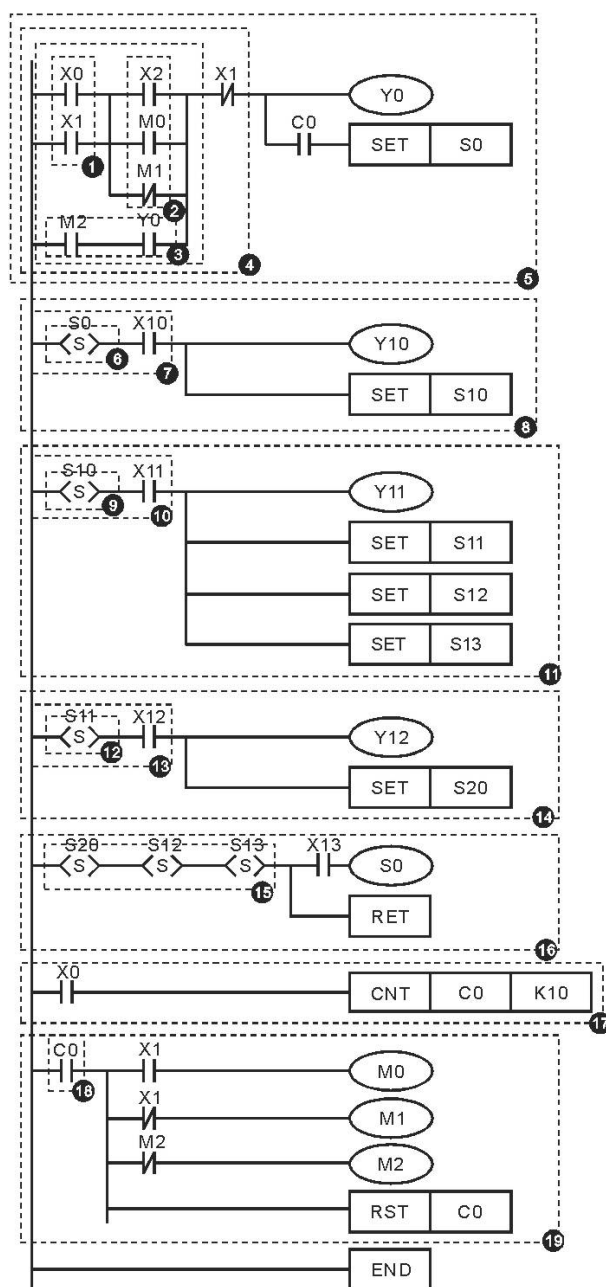


1.5. Преобразование релейно-контактных схем в мнемокод

На ниже приведенном рисунке показана программа, представленная в виде релейно-контактной символики и виде списка инструкций (мнемокода). На рисунке виден порядок преобразования лестничной диаграммы в код, исполняемый ПЛК.

При написании релейно-контактных схем всегда нужно помнить о количестве команд, которые потребуются для её ввода.

Релейно-контактная схема



```

LD    X0
OR    X1
LD    X2
OR    M0
ORI   M1
ANB   ←
LD    M2
AND   Y0
ORB   ←
ANI   X1
OUT   Y0
AND   C0
SET   S0
STL   S0
LD    X10
OUT   Y10
SET   S10
STL   S10
LD    X11
OUT   Y11
SET   S11
SET   S12
SET   S13
STL   S11
LD    X12
OUT   Y12
SET   S20
STL   S20
STL   S12
STL   S13
LD    X13
OUT   S0
RET
LD    S0
CNT   C0 K10
LD    C0
MPS
AND   X1
OUT   M0
MRD
ANI   X1
OUT   M1
MPP
ANI   M2
OUT   M2
END

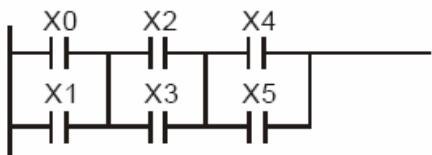
```

1 Блок ИЛИ
 2 Блок ИЛИ
 3 Блок И
 4 Парал. соедин. блоков
 5 Состояние выхода будет установлено в соответств. с сост. выполн. программы
 6 Разветвленные выходы
 7 Старт пошагового выполнения
 8 Установка Y10 и переход к след. шагу
 9 S10 установка сост-я
 10 Установка сост-я X11
 11 Установка Y11 и переход к след. шагу
 12 S11 уст-ка сост-я
 13 Уст-ка сост-я X12
 14 Установка Y12 и переход к след. шагу
 15 Одновременное соответствие
 16 Установка и переход к след. шагу
 17 Возвращение
 18 Чтение C0
 Разветвл. выходы
 Конец пошагового управления
 Конец программы

Процесс обработки релейно-контактной схемы идет с верхнего левого угла и заканчивается в правом нижнем, однако могут быть исключения и различные варианты преобразования в мнемocode, как показано в следующих примерах:

Пример 1. Ниже приведенную схему можно кодировать двумя различными методами, однако результат будет тождественным.

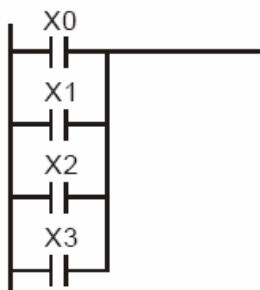
Первый метод кодирования является наиболее предпочтительным.



Метод 1		Метод 2	
LD	X0	LD	X0
OR	X1	OR	X1
LD	X2	LD	X2
OR	X3	OR	X3
ANB		LD	X4
LD	X4	OR	X5
OR	X5	ANB	
ANB		ANB	

Методом 2 можно объединить максимум 8 логических блоков. Для первого метода число логических блоков не ограничено.

Пример 2. Различное кодирование параллельно соединенных контактов.

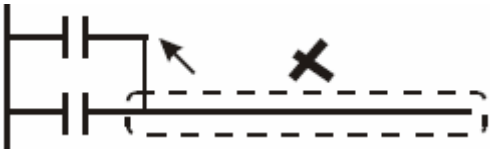
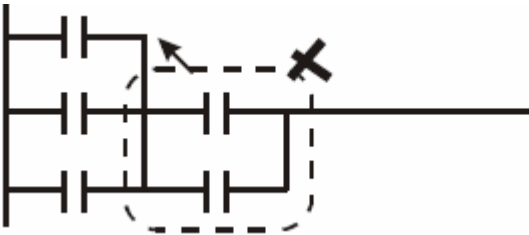
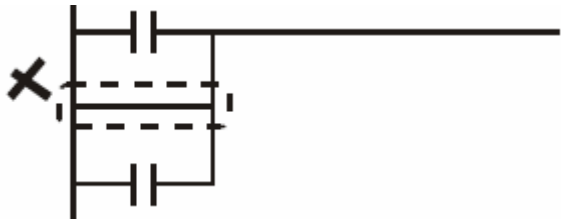
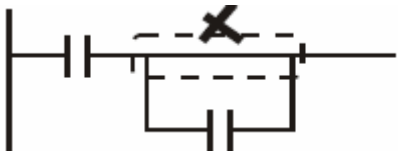
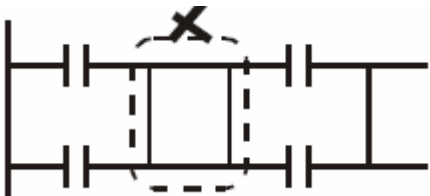
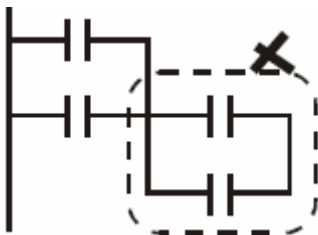
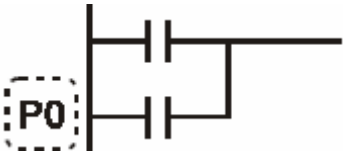
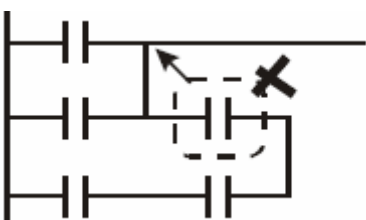


Метод 1		Метод 2	
LD	X0	LD	X0
OR	X1	LD	X1
OR	X2	LD	X2
OR	X3	LD	X3
		ORB	
		ORB	
		ORB	

Первый метод кодирования является наиболее предпочтительным с точки зрения использования оперативной памяти ПЛК.

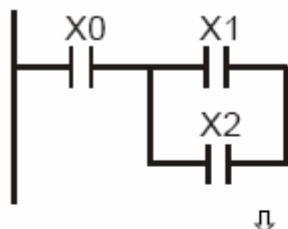
Ошибки при написании релейно-контактных схем

	Нельзя подключать параллельный контакт сверху.
	"Ток" через контакты должен протекать только слева на право. Реверсивное направление не допускается.

	Командная линия должна располагаться выше.
	Блок ИЛИ должен быть расположен выше
	Нельзя выполнить операцию параллельно пустой линии.
	Нельзя выполнить операцию параллельно пустой линии.
	В среднем блоке отсутствуют объекты.
	Блоки должны располагаться на одном уровне.
	Указатель должен быть расположен напротив верхнего устройства командной строки.
	Блок, который соединен последовательно, должен располагаться параллельно.

1.6. Оптимизация релейно-контактных схем

- Если поставить блок вначале командной линии, можно избежать использования команды ANB для последовательной связки блока.



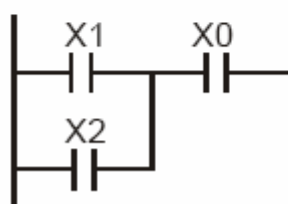
Command

LD X0

LD X1

OR X2

ANB



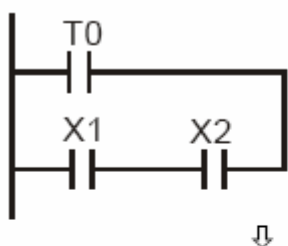
Command

LD X1

OR X2

AND X0

- Если поставить блок выше параллельного контакта, можно избежать использования команды ORB для параллельной связки блока.



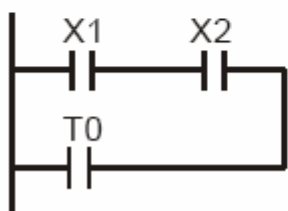
Command

LD T0

LD X1

AND X2

ORB



Command

LD X1

AND X2

OR T0

- В схеме с реверсивным направлением тока можно поменять верхнюю и нижнюю командную линию между собой местами, и избежать, тем самым, ошибки не нарушив логику работы.

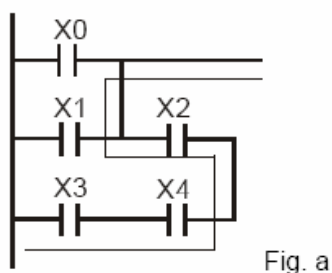


Fig. a

command

```
LD      X0
OR      X1
AND     X2
LD      X3
AND     X4
ORB
```

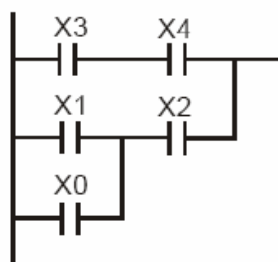
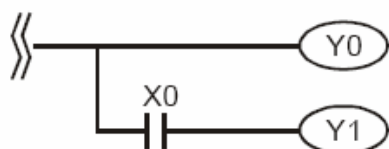
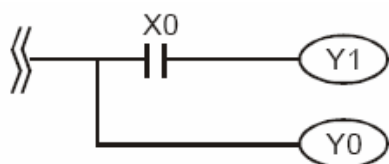


Fig. b

command

```
LD      X3
AND     X4
LD      X1
OR      X0
AND     X2
ORB
```

- Можно избежать использования команд MPS, MPP при разветвлении выходов поменяв командные линии следующим образом:



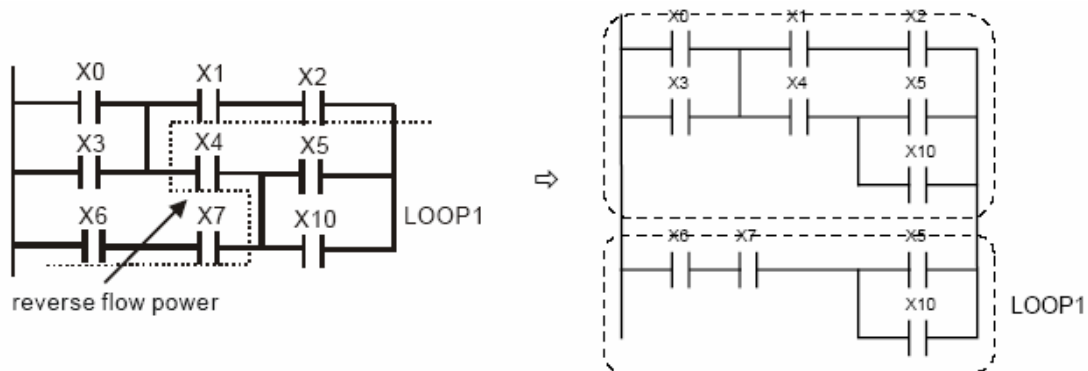
command

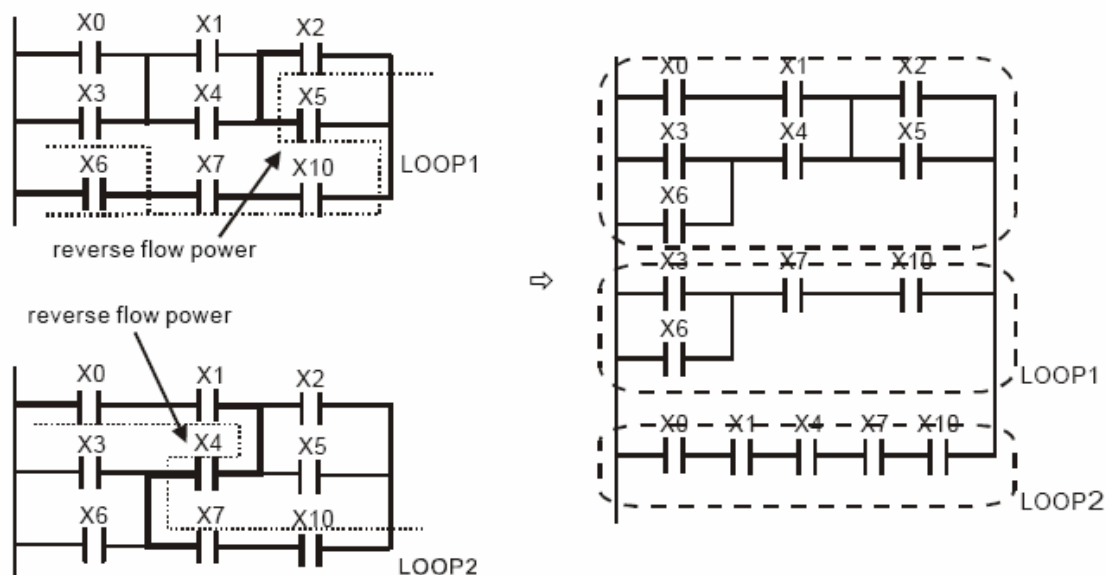
```
MPS
AND     X0
OUT     Y1
MPP
OUT     Y0
```

command

```
OUT     Y0
AND     X0
OUT     Y1
```

- В следующих примерах показаны примеры оптимизации сложных схем:



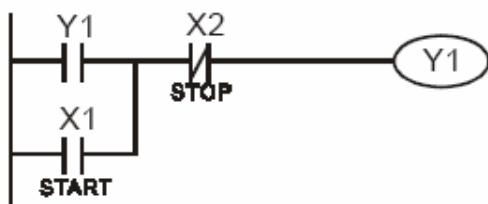


1.7. Примеры написания программ

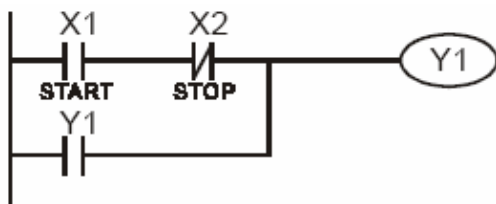
• Старт, стоп и самоблокировка

Часто бывает необходимо использовать для старта и стопа кнопки без фиксации, но с самоблокировкой выхода. Примеры реализации таких схем представлены ниже:

Пример 1: Самоблокировка выхода с приоритетом Стопа



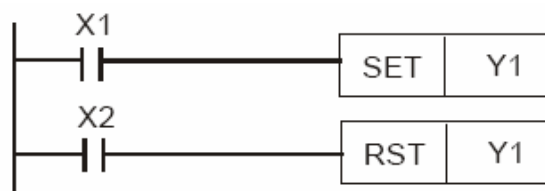
Пример 2: Самоблокировка выхода с приоритетом Старта



Пример 3: Самоблокировка выхода с использованием команд SET и RESET

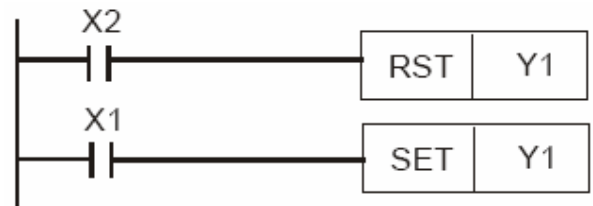
Приоритет Стопа

ПЛК выполняет программу сверху вниз, и следовательно приоритетом будет обладать команда расположенная ниже.

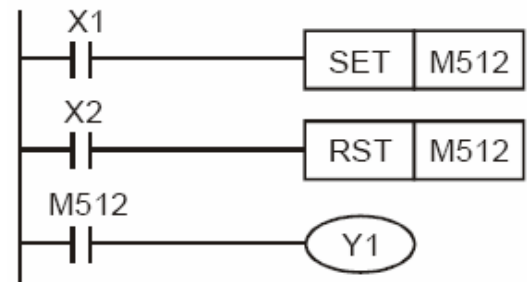


Приоритет Старта

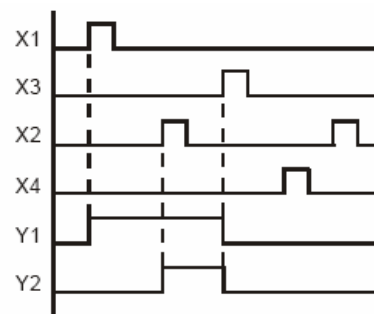
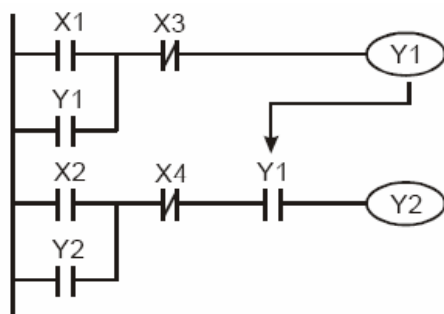
Если одновременно замкнуты оба контакта X1 и X2, то в верхней схеме выход Y1=0, а в нижней - Y1=1.

**Пример 4: Энергонезависимая самоблокировка выхода**

Внутреннее реле M512 является энергонезависимым (см. руководство по эксплуатации), и, следовательно, выход Y1 после включения ПЛК сохранит состояние, которое было до отключения питания.



- Схемы общего применения**

Пример 5: Условное управление

Контакты X1 и X3 включают/выключают выход Y1 автономно, а X2 и X4 могут управлять состоянием выхода Y1 только при условии, что Y1 включен, т.е. выход Y1 является последовательным контактом (логическим И) для нижней схемы.

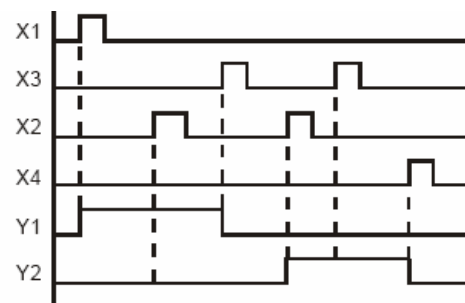
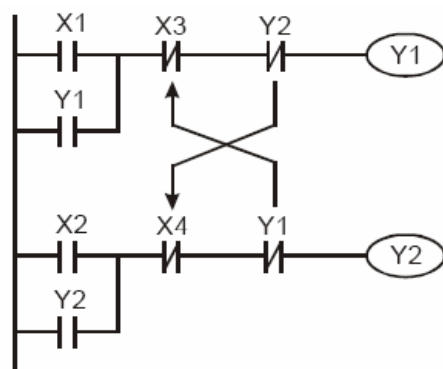
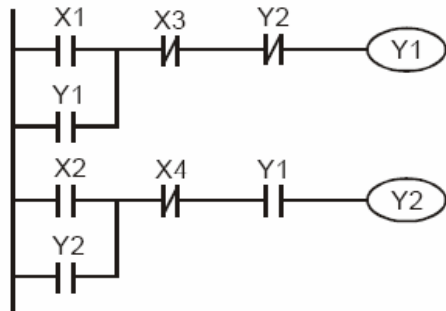
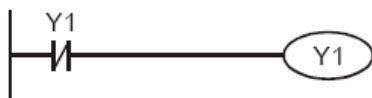
Пример 6: Схема с взаимоблокировкой

Схема исключает одновременное включение двух выходов. Когда включен один выход, второй будет заблокирован. При одновременном замыкании контактов X1 и X2 приоритет будет иметь Y1.

Пример 7: Последовательное управление

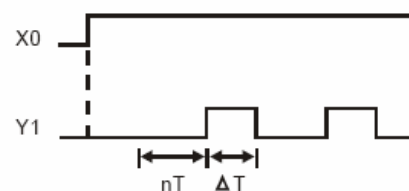
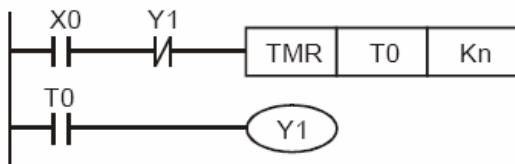
Выход Y2 может быть включен, только если включен Y1, однако при включении Y2 выход Y1 будет отключен.

Пример 8: Колебательные схемы

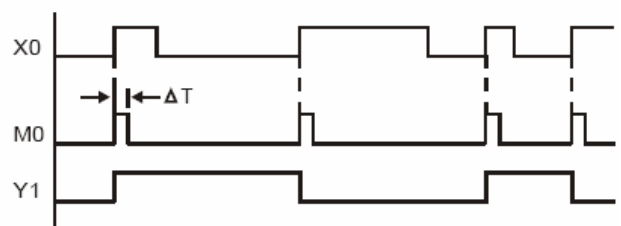
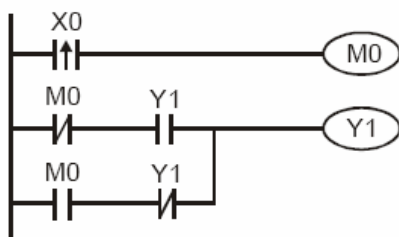
ΔT – время сканирования (время выполнения одного цикла программы)

В первом цикле сканирования выход Y1 будет включен, а во втором – выключен, и т.д.

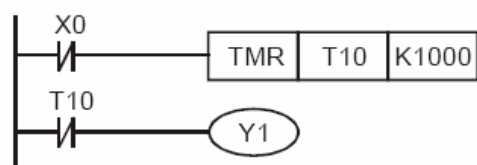
Период колебаний = $\Delta T + \Delta T$



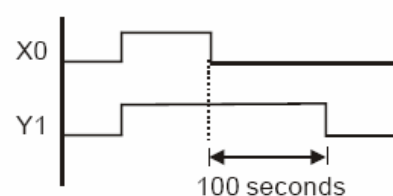
Период колебаний = $nT + \Delta T$

Пример 9: Триггерная схема

Каждое замыкание контакта X0 изменяет состояние выхода Y1 на противоположное. Эта схема еще называется импульсным реле.

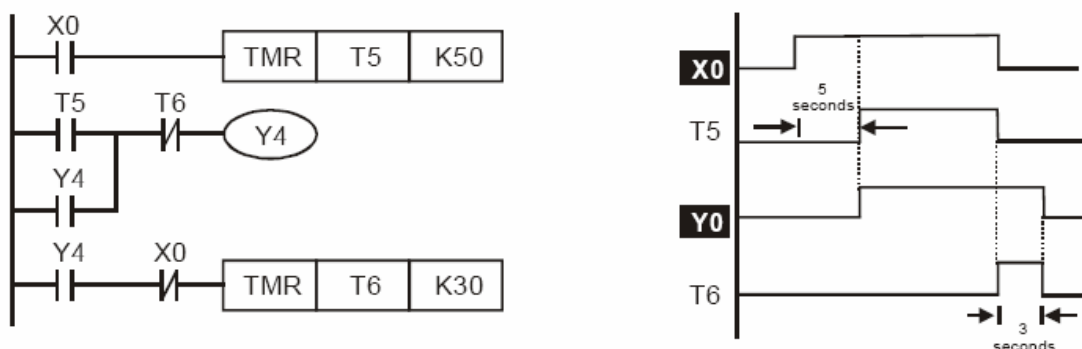
Пример 9: Таймер с задержкой на выключение

TB = 0.1 sec

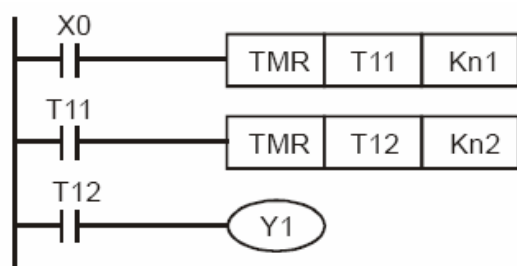


Когда $X0 = 1$, выход $Y1$ включен. При выключении $X0$, выход $Y1$ выключится через 100 секунд, т.к. операнд заданного значения таймера $T0$ имеет дискретность 0.1 сек, т.е. $K1000 = 100$ сек.

Пример 9: Таймер с задержкой на включение и выключение

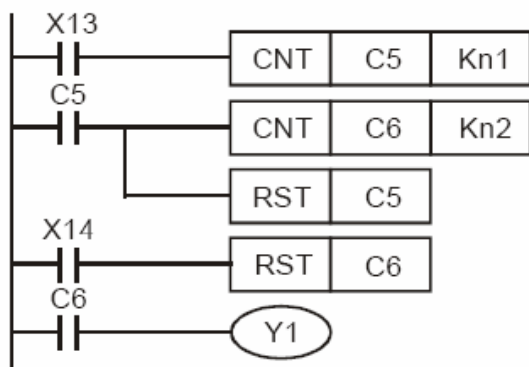


Пример 10: Таймер с 2-х ступенчатой задержкой на включение



Выход $Y1$ будет включен через время $n1+n2$ после замыкания контакта $X0$.

Пример 11: Расширение диапазона счетчика



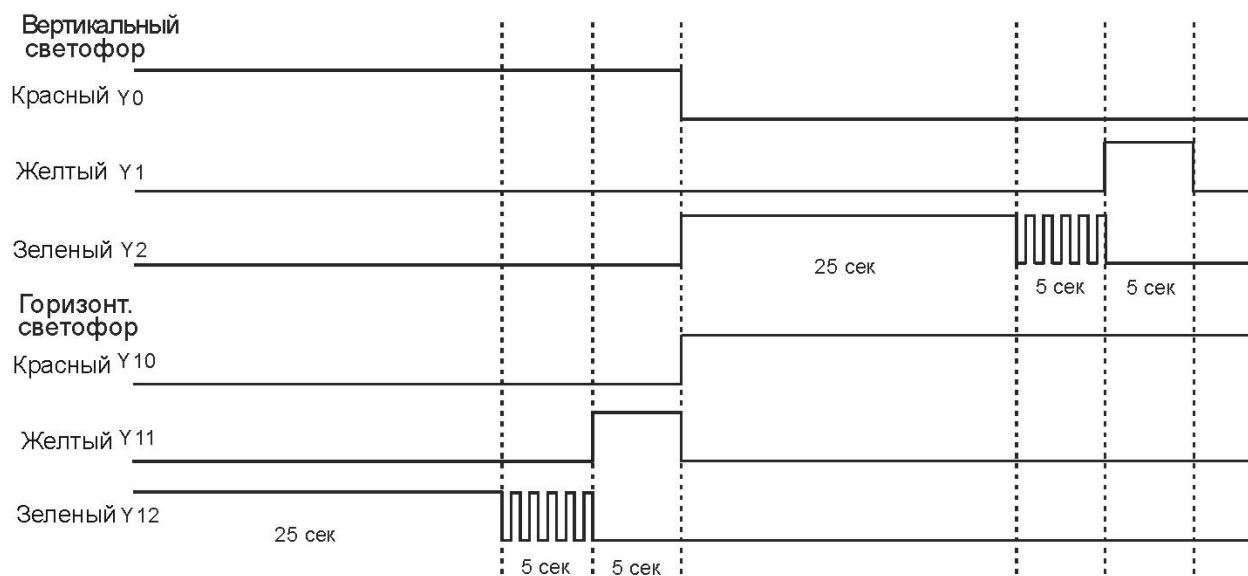
Диапазон 16-ти битного счетчика равен: $0 \dots 32676$. Применение второго счетчика позволит увеличить общий диапазон счета, который будет равен произведению операндов $n1 \times n2$.

Пример 12: Программа управления светофором (с использованием инструкций пошагового управления)

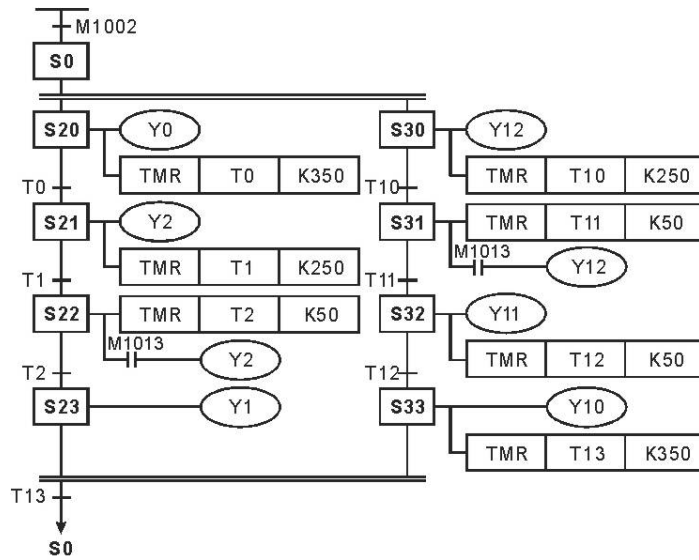


	Красный	Желтый	Зеленый	Зеленый мигает
Вертикальн. светофор	$Y0$	$Y1$	$Y2$	$Y2$
Горизонт. светофор	$Y10$	$Y11$	$Y12$	$Y12$
Время	35 сек	5 сек	25 сек	5 сек

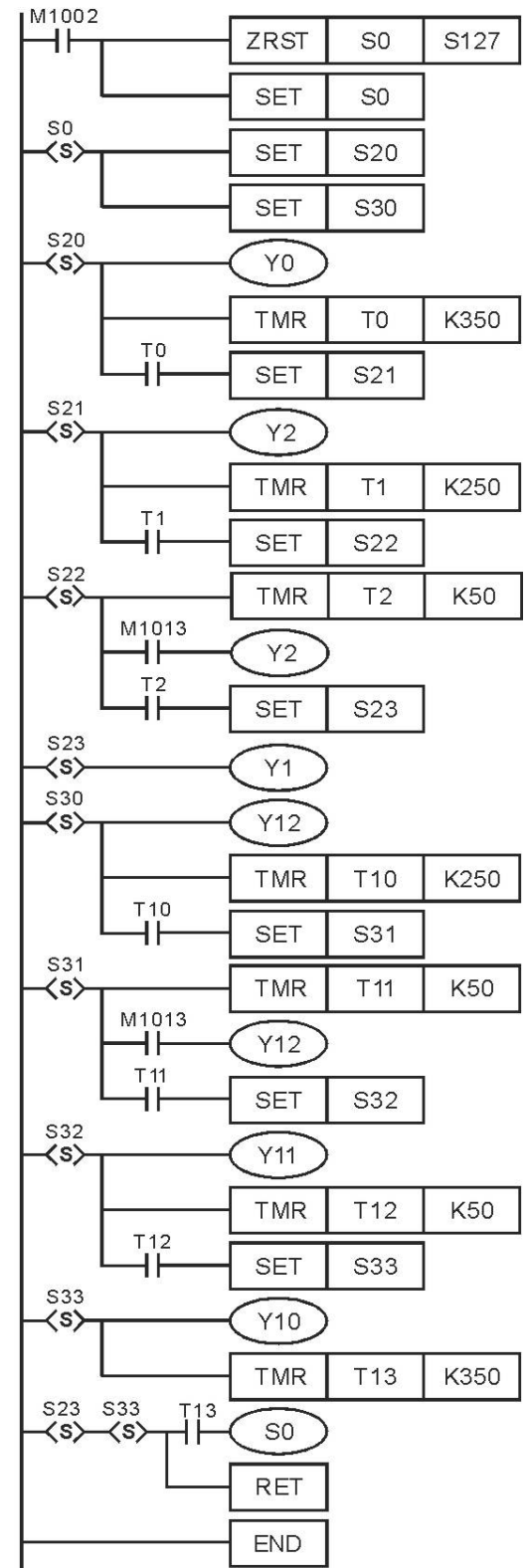
Временная диаграмма:



SFC диаграмма:



Релейно-контактная схема:



- Программирование на языке SFC

Схема SFC	Вид внутренних релейно-контактных схем
<p>The SFC diagram shows a sequence of states and transitions. State S0 is the initial state. Transition 0 leads to a parallel branch with S20 and S30. From S20, transition 1 leads to S21. From S30, transition 5 leads to S31. From S21, transition 2 leads to S22. From S31, transition 6 leads to S32. From S22, transition 3 leads to S23. From S32, transition 7 leads to S33. From S23, transition 4 leads back to S0. States S0, S22, and transition 4 are circled in the diagram.</p>	<p>□ LAD-0</p> <p>□ Условный переход 1</p> <p>□ S22</p> <p>□ Условный переход 4</p> <p>□ Условный переход 7</p>

2. ОПЕРАНДЫ В DVP-PLC

2.1. Обзор операндов

DVP-ES/EX/SS

Тип	Операнд			Диапазон адресов		Назначение	
Реле (1-но битная память)	X	Внешние входные реле		128 точек (X0 – X177)		Макс.256 точек	Входы ПЛК
	Y	Внешние выходные реле		128 точек (Y0 – Y177)			Выходы ПЛК
	M	Внутренние реле (меркеры)	Общие	744(M0-M511, M768-M999)		Макс. 1280 точек	Промежуточная двоичная память. Соответствуют промежуточным реле в электросхемах
			Энергонезав.	256 точек (M512-M767)			
			Специальные	280 точек (M1000-M1279)			
	T	Таймеры	Дискр. 100мс	64 точки (T0-T63)		Макс. 128 точек	Используются в качестве контактов (Т), которые замыкаются при достижении соотв. таймером (команда TMR) своего заданного значения
			Дискр. 10мс	63 точки (T64-T126) M1028=ON – дискретн. 10мс M1028=OFF-дискретн. 100мс			
				Дискр. 1мс	1 точка (T127)		
	C	Счетчики	Инкрементный (16 бит)	112 точек (C0-C111)		Макс. 128 точек	Используются в качестве контактов (С), которые замыкаются при достижении соотв. счетчиком (команда CNT) своего заданного значения
			Инкр. (16 бит) энергонезав.	16 точек (C112-C127)			
			Инкр./декрем. 32 бит быстродействующий энергонезав.	7 точек (C235-C244) 1 фаза, 1 вх		Макс. 13 точек	
				3 точки (C246, C247, C249) 1 фаза, 2 вх 3 точки (C251, C252, C254) 2 фазы, 2 вх			
	S	Шаговые реле (энергонез.)	Начальные	10 точек (S0-S9)		Макс. 128 точек	Используются в режиме пошагового управления
			Возвращ. в нулев. точку	10 точек (S10-S19) используются с инстр. IST			
			Общие	108 точек (S20-S127)			
Регистр (16-ти битная память)	T	Текущее значение таймера		128 точек (T0-T127)		Регистры для хранения текущ. знач. таймеров	
	C	Текущее значение счетчика		128 счетчиков 16-ти бит (C0-C127) 13 счетчиков 32-х бит (C235-C254)		Регистры для хранения текущ. знач. счетчиков	
	D	Регистры данных	Общие	408 точек (D0-D407)		Макс.600 точек	Используются для хранения данных. Е и F используются для индексной идентификации операндов.
			Энергонезав.	192 точек (D408-D599)			
			Специальные	312 точек (D1000-D1311)		Макс.312 точек	
Индексные			2 точки: E(=D1028), F(=D1029)				
Указатели	N	Номера вложенности для инструкций MC, MCR		8 точек (N0-N7).		Используются для нумерации вложенных схем исключения.	
	P	Указатели для инструкций CJ, CALL		64 точек (P0-P63)		Метки для операторов перехода, подпрограмм	
	I	Прерывания	Внешние	4 точки (I001, I101, I201, I301)		Метки для подпрограмм обработки прерываний	
			Временные	1 точка (I6xx = 10-99, дискр. 1 мс)			
			Коммуникац.	1 точка (I150)			
Константы	K	Десятичные константы		K-32768 ...K32767 (16 битные операции) K-2147483648 ...K2147483647 (32 битные операции)			
	H	Шестнадцатеричные константы		H0000...HFFFF (16-ти битные операции) H00000000...HFFFFFFFF (32-х битные операции)			

DVP- SA/SX

Тип	Операнд		Диапазон адресов		Назначение
Реле (1-но битная память)	X	Внешние входные реле	128 точек (X0 – X177)	Макс.256	Входы ПЛК
	Y	Внешние выходные реле	128 точек (Y0 – Y177)	точек	Выходы ПЛК
	M	Внутренние реле (меркеры)	Общие	512 точек (M0 - M511)	Макс. 4096 точек Промежуточная двоичная память. Соответствуют промежуточным реле в электросхемах
			Энергонезав.*	488 точек (M512-M999) 2096тчк(M2000-M4095)	
			Специальные	1000 точек (M1000-M1999)	
	T	Таймеры	Дискр. 100мс	200 точек (T0-T199), T192-T199 для подпрограмм, 6 точек аккумулятивн. типа (T250-T255)	Макс. 256 точек Используются в качестве контактов (Т), которые замыкаются при достижении соотв. таймером (команда TMR) своего заданного значения
			Дискр. 10мс	40 точек (T200-T239), 6 точек аккумулятивн. типа (T240-T245)	
			Дискр. 1мс	4 точки аккумулятивн. типа (T246-T249)	
	C	Счетчики	Инкрементный (16 бит)	96 точек (C0-C95)	Макс. 250 точек Используются в качестве контактов (С), которые замыкаются при достижении соотв. счетчиком (команда CNT) своего заданного значения
			Инкр. (16 бит) энергонезав.*	104 точек (C96-C199)	
			Инкр./декрем. 32 бит	16 точек (C200-C215)	
				19 точек (C216-C234)	
			Инкр./декрем. 32 бит быстродействующий энергонезав.	9 точек (C235-C243), 1фаза, 1вход	
				3 точки (C246- C249), 1фаза, 2вх.	
				3 точки (C251- C254), 2фазы, 2вх.	
	S	Шаговые реле	Начальные	10 точек (S0-S9)	Макс. 1024 точки Используются в режиме пошагового управления
			Возвращ. в нулев. точку	10 точек (S10-S19) используются с инструкцией IST	
			Общие	492 точки (S20-S511)	
			Энергонезав.*	384 точки (S512-S825)	
			Сигнальные*	124 точки(S896-S1023)	
Регистр (16-ти битная память)	T	Текущее значение таймера		256 точек (T0-T255)	Регистры для хранения текущ. знач. таймеров
	C	Текущее значение счетчика		200 счетчиков 16-ти бит (C0-C199) 50 счетчиков 32-х бит (C200-C254)	Регистры для хранения текущ. знач. счетчиков
	D	Регистры данных	Общие	200 точек (D0-D199)	Макс. 5000 точек Используются для хранения данных. Е и F используются для индексной идентификации операндов.
			Энергонезав.*	3800 точек (D200-D999, D2000-D4999)	
			Специальные	1000 точек (D1000-D1999)	
			Индексные	8 точек: E0-E3, F0-F3	
	нет	Файловые регистры		1600 точек (K0-K1599)	Дополнит. регистры для хранения данных
Указатели	N	Номера вложенности для инструкций MC, MCR		8 точек (N0-N7).	Используются для нумерации вложенных схем исключения.
	P	Указатели для инструкций CJ, CALL		256 точек (P0-P255)	Метки для операторов перехода, подпрограмм
	I	Прерывания	Внешние	6 точек с включением по переднему фронту: I001(X0), I101(X1), I201(X2), I301(X3) , I401(X4), I501(X5)	Метки для подпрограмм обработки прерываний

			Временные	I6xx (1мс), I7xx (1мс) (xx= 10-99)	
			Быстр. счетч.	I010, I020, I030, I040, I050, I060	
			Коммуникац.	1 точка (I150)	
Константы	К	Десятичные константы	K-32768 ...K32767 (16 битные операции) K-2147483648 ...K2147483647 (32 битные операции)		
	Н	Шестнадцатеричные константы	H0000...HFFFF (16-ти битные операции) H00000000...HFFFFFFFF (32-х битные операции)		

* Область энергонезависимой памяти может быть изменена с помощью специальных параметров (см. нижеприведенную таблицу).

Установка области энергонезависимых операндов в DVP-SA/SX

М Внутренние реле	Общие		Энергонезависимые		Специальные		Энергонезависимые	
	M0 - M511		M512-M999		M1000-M1999		M2000-M4095	
	Энергозависимые и изменению не подлежат		По умолчанию энергонезависимые		Некоторые из них энергонезависимые и изменению не подлежат		По умолчанию энергонезависимые	
Начало: D1200(K512) Конец: D1201(K999)			Начало: D1202(K2000) Конец: D1203(K4095)					
Т Таймеры	100 мс		10 мс		10 мс	1 мс		100 мс
	T0-T199		T200-T239		T240-T245	T246-T249	T250-T255	
	Энергозависимые и изменению не подлежат		Энергозависимые и изменению не подлежат		Аккумулятивный тип, т.е. с памятью (энергонезависимой) текущего значения и изменению не подлежат			
С Счетчики	Инкрементный (16 бит)			Инкр./декрем. 32 бит		Инкр./декрем. 32 бит быстро- действующий		
	C0-C95		C96-C199	C200-C215	C216-C234	C235-C255		
	Энергозавис. и изменению не подлежат		По умолчанию энергонезав.	Энергозавис. и изменению не подлежат	По умолчанию энергонезав.	По умолчанию энергонезависимые		
			Начало: D1208(K96) Конец: D1209(K199)		Начало: D1210(K216) Конец: D1211(K234)	Начало: D1212(K235) Конец: D1213(K255)		
S Шаговые реле	Начальн.	Возвр. в ноль	Общие	Энергонезависимые		Сигнальные		
	S0-S9	S10-S19	S20- S511	S512-S825		S896-S1023		
	Энергозависимые и изменению не подлежат			По умолчанию энергонезависимые		Энергонезависимые и изменению не подлежат		
Начало: D1214 (K512) Конец: D1215 (K825)								
D Регистры данных	Общие		Энергонезависимые		Специальные		Энергонезависимые	
	D0-D199		D200-D999		D1000-D1999		D2000-D4999	
	Энергозависимые и изменению не подлежат		По умолчанию энергонезависимые		Некоторые из них энергонезависимые и изменению не подлежат		По умолчанию энергонезависимые	
Начало: D1216(K200) Конец: D1217(K999)			Начало: D1218(K2000) Конец: D1219(K4999)					
Файловые регистры	K0-K1599							
	Энергонезависимые и изменению не подлежат							

DVP- EN

Тип	Операнд			Диапазон адресов		Назначение	
Реле (1-но битная память)	X	Внешние входные реле		256 точек (X0 – X377)		Макс.512 точек	Входы ПЛК
	Y	Внешние выходные реле		256 точек (Y0 – Y377)			Выходы ПЛК
	M	Внутренние реле (меркеры)	Общие *	500 точек (M0 – M499)		Макс. 4096 точек	Промежуточная двоичная память. Соответствуют промежуточным реле в электросхемах
			Энергонезав.*	500 точек (M500-M999) 2096 точек (M2000-M4095)			
			Специальные	1000 точек (M1000-M1999)			
	T	Таймеры	Дискр. 100мс	200 точек (T0-T199), T192-T199 для подпрограмм, 6 точек аккумулятивн. типа (T250-T255)		Макс. 256 точек	Используются в качестве контактов (Т), которые замыкаются при достижении соотв. таймером (команда TMR) своего заданного значения
			Дискр. 10мс	40 точек (T200-T239), 6 точек аккумулятивн. типа (T240-T245)			
			Дискр. 1мс	4 точки аккумулятивн. типа (T246-T249)			
	C	Счетчики	Инкрементный (16 бит)*	100 точек (C0-C99)		Макс. 253 точек	Используются в качестве контактов (С), которые замыкаются при достижении соотв. счетчиком (команда CNT) своего заданного значения
			Инкр. (16 бит) энергонезав.*	100 точек (C100-C199)			
			Инкр./декрем. 32 бит*	20 точек (C200-C219)			
				15 точек (C220-C234)			
			Инкр./декрем. 32 бит быстродействующий энергонезав.*	10 точек (C235-C244), 1фаза, 1вх.			
				4 точки (C246- C249), 1фаза, 2вх.			
	S	Шаговые реле	Начальные	10 точек (S0-S9)		Макс. 1024 точки	Используются в режиме пошагового управления
			Возвращ. в нулев. точку	10 точек (S10-S19) используются с инструкцией IST			
			Общие	480 точек (S20-S499)			
			Энергонезав.*	400 точек (S500-S899)			
			Сигнальные*	124 точки (S900-S1023)			
Регистр (16-ти битная память)	T	Текущее значение таймера		256 точек (T0-T255)		Регистры для хранения текущ. знач. таймеров	
	C	Текущее значение счетчика		200 счетчиков 16-ти бит (C0-C199) 53 счетчика 32-х бит (C200-C254)		Регистры для хранения текущ. знач. счетчиков	
	D	Регистры данных	Общие*	200 точек (D0-D199)		Макс. 10000 точек	Используются для хранения данных. Е и F используются для индексной идентификации операндов.
			Энергонезав.*	800 точек (D200-D999) 8000тчк (D2000-D9999)			
			Специальные	1000 точек (D1000-D1999)			
			Индексные	16 точек: E0-E7, F0-F7			
нет	Файловые регистры		10000 точек (K0-K9999)		Дополнительн. регистры для хранения данных		
Указатели	N	Номера вложенности для инструкций MC, MCR		8 точек (N0-N7).		Используются для нумерации вложенных схем исключения.	
	P	Указатели для инструкций CJ, CALL		256 точек (P0-P255)		Метки для операторов перехода, подпрограмм	
	I	Прерывания	Внешние	6 точек с включением по переднему фронту (x=1) или по заднему (x=0): I00x(X0), I10x(X1), I20x(X2), I30x(X3) . I40x(X4), I50x(X5)		Метки для подпрограмм обработки прерываний	

			Временные	I6xx (1мс), I7xx (1мс) , I8xx (0.1мс) (xx= 10-99)	
			Быстр. счетч.	I010, I020, I030, I040, I050, I060	
			Импульс.вых.	4 точки (I110, I120, I130, I140)	
			Коммуникац.	2 точки (I150, I160)	
Константы	К	Десятичные константы	K-32768 ...K32767 (16 битные операции) K-2147483648 ...K2147483647 (32 битные операции)		
	Н	Шестнадцатеричные константы	H0000...HFFFF (16-ти битные операции) H00000000...HFFFFFFFF (32-х битные операции)		

* Область энергонезависимой памяти может быть изменена с помощью специальных параметров (см. нижеприведенную таблицу).

Установка области энергонезависимых операндов в DVP-EN

М Внутренние реле	Общие		Энергонезависимые		Специальные		Энергонезависимые	
	M0 – M499		M500-M999		M1000-M1999		M2000-M4095	
	Начало: D1200 (K512) Конец: D1201 (K999)				Изменению не подлежат		Начало:D1202(K2000) Конец: D1203(K4095)	
Т Таймеры	100 мс		10 мс		10 мс		1 мс	
	T0-T199		T200-T239		T240-T245		T246-T249	
	Энергозависимые и изменению не подлежат		Энергозависимые и изменению не подлежат		Аккумулятивный тип, т.е. с памятью (энергонезависимой) текущего значения и изменению не подлежат			
С Счетчики	Инкрементный (16 бит)			Инкр./декрем. 32 бит			Инкр./декрем. 32 бит быстро- действующий	
	C0-C99		C100-C199		C200-C219		C220-C234	
	По умолчанию энергозав.		По умолчанию энергонезав.		По умолчанию энергозав		По умолчанию энергонезав.	
	Начало: D1208(K100) Конец: D1209(K199)			Начало: D1210(K220) Конец: D1211(K234)			Начало: D1212(K235) Конец: D1213(K255)	
S Шаговые реле	Начальн	Возвр. в ноль	Общие	Энергонезависимые			Сигнальные	
	S0-S9	S10-S19	S20- S499	S500-S899			S900-S1023	
	По умолчанию энергозависимые			По умолчанию энергонезависимые			Энергонезависимые и изменению не подлежат	
	Начало: D1214 (K500) Конец: D1215 (K899)							
D Регистры данных	Общие		Энергонезависимые			Специальные		Энергонезависимые
	D0-D199		D200-D999			D1000-D1999		D2000-D9999
	По умолчанию энергозависимые		По умолчанию энергонезависимые			Некоторые из них энергонезависимые и изменению не подлежат		По умолчанию энергонезависимые
	Начало: D1216(K200) Конец: D1217(K999)			Начало:D1218(K2000) Конец: D1219(K9999)				
Файловые регистры	K0-K9999							
	Энергонезависимые и изменению не подлежат							

Очистка памяти в DVP-ES/EX/SS

Тип памяти	Подача питания	STOP=>RUN	RUN=>STOP	M1031 = 1	M1032 = 1	Заводская установка
Энергозав.	Очистка	Когда M1033 = 0, очистка		Очистка	Без изменений	0
		Когда M1033=1, без измен.				
Энергонезав		Без изменений		Без измен.	Очистка	0
Специальн.М, Специальн.Д, Индексные регистры	Инициализация	Без изменений		Без изменений		Начальная установка

Очистка памяти в DVP-SA/SX/EN

Тип памяти	Подача питания	STOP=>RUN	RUN=>STOP	M1031 = 1	M1032 = 1	Заводская установка
Энергозав.	Очистка	Когда M1033 = 0, очистка		Очистка	Без изменений	0
		Когда M1033=1, без измен.				
Энергонезав	Без изменений			Без измен.	Очистка	0
Специальн.М, Специальн.Д, Индексные регистры	Инициализация	Без изменений		Без изменений		Начальная установка
Файловые регистры	Без изменений					0

2.2. Форматы чисел, Константы [K] и [H]

Области числовых значений десятичных и шестнадцатеричных констант.

Константы	K	Десятичные константы	K-32768 ...K32767 (16-ти битные операции) K-2147483648 ...K2147483647 (32-х битные операции)
	H	Шестнадцатеричные константы	H0000...HFFFF (16-ти битные операции) H00000000...HFFFFFFFF (32-х битные операции)

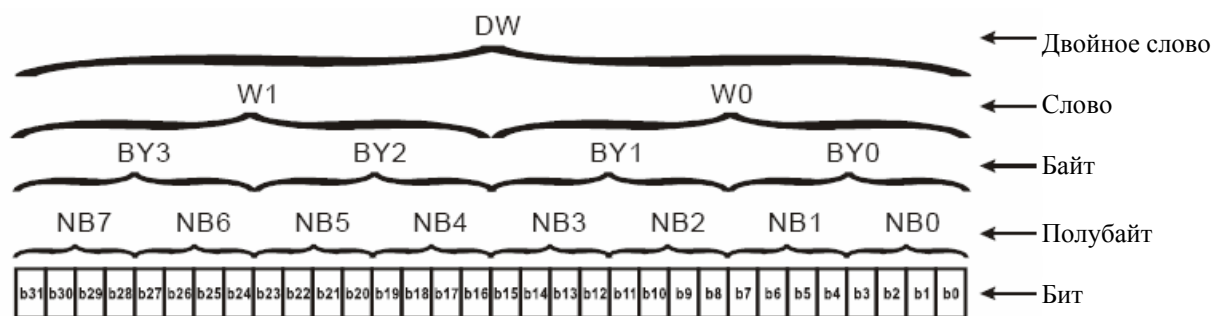
С помощью десятичных (K) и шестнадцатеричных (H) констант можно задавать числовые значения внутри программы ПЛК (например, задаваемые значения уставок времени или счета). Числовые значения кодируются внутри ПЛК в двоичном счислении.

Далее рассмотрены системы счисления используемые в DVP-PLC:

1. Двоичный формат чисел (BIN)

Используется для внутренних вычислений ПЛК и памяти и имеет следующее представление:

- Бит: Это основная единица измерения двоичной системы, может иметь два состояния 0 или 1
- Полубайт: Это единица измерения, состоящая из 4-х битов, b3 – b0. Может использоваться для представления чисел 0-9 (DEC) и 0-F (HEX)
- Байт: Это единица измерения, состоящая из 8-ми битов, b7 – b0. Может использоваться для представления чисел 00-FF (HEX)
- Слово: Это единица измерения, состоящая из 2-х байтов или 16-ти битов, b15 – b0. Может использоваться для представления чисел 0000-FFFF (HEX)
- Двойное слово: Это единица измерения, состоящая из 2-х слов или 32-х битов, b31 – b0. Может использоваться для представления чисел 00000000-FFFFFFFF (HEX)



2. Восьмеричный формат чисел (OCT)

В контроллерах DVP используется для нумерации (адресации) входов и выходов:

Входы: X0, X1, X2, X3, X4, X5, X6, X7, X10, X11, ...

Выходы: Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11, ...

3. Десятичный формат чисел (DEC)

В контроллерах DVP используется в следующих случаях:

1. для задания уставок таймеров и счетчиков, например TMR T0 K50
2. для адресации операндов S, M, T, C, D, E, F, P, I, например T30, M10

4. Двоично-десятичный формат чисел (BCD)

В BCD-формате каждая цифра десятичного числа представляется четырехбитным двоичным числом.

В контроллерах DVP BCD-формат обычно используется для чтения входных значений от DIP-переключателей или для отображения выходных значений на 7-ми сегментном индикаторе.

5. Шестнадцатеричный формат чисел (HEX)

В контроллерах DVP используется для представления операнда H в прикладных инструкциях, например MOV H1A2B D0.

Константа K: Символ "K" обычно ставится перед числом и обозначает, что число представлено в десятичном формате.

Исключение: Символ "K" может использоваться для представления однобитных операндов X, Y, M, S в виде байтов, слов и двойных слов. Например, K2Y10 или K4M100.

Константа H: Символ "H" обычно ставится перед числом и обозначает, что число представлено в шестнадцатеричном формате.

Таблица соотношений численных форматов в DVP-PLC

BIN		OCT	DEC	BCD		HEX
Для внутренних вычислений		Адресация входов/выходов X/Y	Константы K, адресация S, M, T, C, D, E, F, P, I	Для DIP-переключателей и 7-ми сегментных индикаторов		Константы H
0000	0000	0	0	0000	0000	0
0000	0001	1	1	0000	0001	1
0000	0010	2	2	0000	0010	2
0000	0011	3	3	0000	0011	3
0000	0100	4	4	0000	0100	4
0000	0101	5	5	0000	0101	5
0000	0110	6	6	0000	0110	6
0000	0111	7	7	0000	0111	7
0000	1000	10	8	0000	1000	8
0000	1001	11	9	0000	1001	9
0000	1010	12	10	0001	0000	A
0000	1011	13	11	0001	0001	B
0000	1100	14	12	0001	0010	C
0000	1101	15	13	0001	0011	D
0000	1110	16	14	0001	0100	E
0000	1111	17	15	0001	0101	F
0001	0000	20	16	0001	0110	10
0001	0001	21	17	0001	0111	11
...
0110	0011	143	99	1001	1001	63

2.3. Адресация и назначение входов [X] и выходов [Y]

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

Адресация дискретных входов/выходов выполняется в восьмеричной системе, т.е. входы и выходы не нумеруются числами с использованием цифр 8 и 9.

DVP-	14ES	14SS	20EX	24ES	32ES	60ES	Модули расширения
Входы X	X0 – X7	X0 – X7	X0 – X7	X0 – X17	X0 – X17	X0 – X43	X20(X50) – X177 *
Выходы Y	Y0 – Y5	Y0 – Y5	Y0 – Y5	Y0 – Y7	Y0 – Y17	Y0 – Y27	Y20 (Y30) – Y177 *

* Во всех моделях кроме DVP60ES нумерация входов модулей расширения начинается с X20/выходов с Y20, а в модели DVP60ES нумерация входов модулей расширения начинается с X50/выходов с Y30. Адресация входов/выходов в модулях расширения увеличивается на 8, даже если в модуле входов/выходов меньше восьми.

DVP-	12SA	10SX	Модули расширения
Входы X	X0 – X7	X0 – X3	X20 – X177
Выходы Y	Y0 – Y3	Y0 – Y1	Y20 – Y177

Для базовых модулей серии SA и SX применяются модули расширения серии SS. Адресация входов/выходов в модулях расширения увеличивается на 8, даже если в модуле входов/выходов меньше восьми.

DVP-	16ЕН	20ЕН	32ЕН	48ЕН	64ЕН	80ЕН	Модули расширения
Входы X	X0 – X7	X0 – X13	X0 – X17	X0 – X27	X0 – X37	X0 – X47	X20 – X377 *
Выходы Y	Y0 – Y7	Y0 – Y7	Y0 – Y17	Y0 – Y27	Y0 – Y37	Y0 – Y47	Y20 – Y377 *

* Указанная адресация входов/выходов модулей расширения относится к моделям DVP16ЕН и DVP20ЕН. В других моделях адресация входов/выходов модулей расширения начинается с соответствующих чисел, следующих за последними адресами входов/выходов базового модуля.

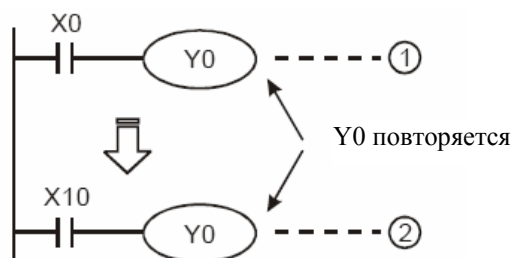
1. Назначение входных реле X

Входные реле X считывают состояния внешних физических устройств (кнопки, переключатели, контакты реле и др.) непосредственно подключенных к входным клеммам ПЛК. Каждый входной контакт X может использоваться в программе неограниченное число раз.

Так же изменять состояние входных контактов X можно с помощью программатора или WPLSoft, если M1304 = 1.

2. Назначение выходных реле Y

Выходные реле Y управляют состоянием физических выходных контактов ПЛК (релейных или транзисторных), а следовательно и устройствами нагрузки (лампы, ТЭНы, катушки реле и др.) непосредственно подключенными к выходным клеммам ПЛК. Каждый выходной контакт Y может использоваться в программе неограниченное число раз, но выходную катушку Y рекомендуется использовать в программе не более одного раза, т.к. при повторении катушки Y, состояние выхода будет определяться последним Y в скане.



Состояние выхода Y0 будет определяться только контактом X10.

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

Процесс обработки в ПЛК входных/выходных сигналов:

Входы:

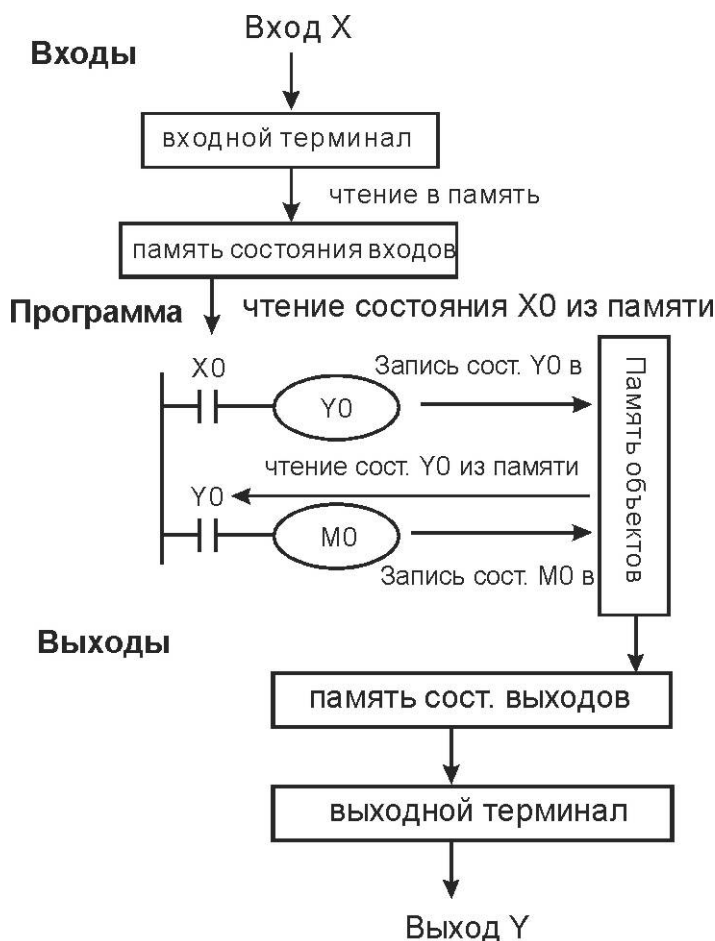
1. ПЛК будет считывать состояние внешних входных устройств, и помещать в память в начале каждого цикла сканирования.
2. Изменения состояния входа во время цикла не будут восприняты, если входной импульс очень короткий (меньше времени одного скана).
3. Время включенного или выключенного состояния входного контакта должно быть не менее 10 мс.

Программа:

4. ПЛК выполняет программу, начиная со строки 0, и сохраняет состояния всех операндов в памяти объектов.

Выходы:

5. После выполнения инструкции END состояния выходных реле Y будут переписаны в память состояния выходов и состояния выходных контактов будут изменены.



2.4. Адресация и назначение внутренних реле [M]

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

В контроллерах DVP используется три типа внутренних реле:

1. Общие, которые не сохраняют свое состояние при отключении питания;
2. Энергонезависимые, которые сохраняют свое состояние при отключении питания;
3. Специальные, которые предоставляют в распоряжение пользователя специальные функции (см. главу 2.10).

Внутренние реле программируются как выходы. Однако отсутствует возможность присоединить к ним внешние устройства. Они могут использоваться в программе неограниченное число раз.

Адресация внутренних реле выполняется в десятичном формате.

DVP-ES/EX/SS

Внутренние реле M	Общие	744 точки (M0-M511, M768-M999)	Макс. 1280 точек
	Энергонезависимые	256 точек (M512-M767)	
	Специальные	280 точек (M1000-M1279)	

DVP-SA/SX

Внутренние реле M	Общие	512 точек (M0 - M511)	Макс. 4096 точек
	Энергонезависимые	488 точек (M512-M999) 2096 точек (M2000-M4095)	
	Специальные	1000 точек (M1000-M1999)	

DVP-EN

Внутренние реле M	Общие	500 точек (M0 – M499)	Макс. 4096 точек
	Энергонезависимые	500 точек (M500-M999) 2096 точек (M2000-M4095)	
	Специальные	1000 точек (M1000-M1999)	

2.5. Адресация и назначение шаговых реле [S]

Шаговые реле (операнды состояния шага) являются основным элементом режима пошагового управления и применяются совместно с инструкциями STL/RET.

Имеется 1024 операнда состояния шагов в области от S0 до S1023 и их можно подразделить на 5 групп:

1. Начальные шаговые реле: S0 – S9, 10 адресов. В SFC используются для начальной инициализации.
2. Шаговые реле возвращения в нулевую точку: S10 – S19, 10 адресов. Используются совместно с инструкцией IST (API 60) в качестве адресов возвращения в нулевую точку. Если в программе не используется инструкция IST (API 60), эти адреса могут использоваться в качестве шаговых реле общего назначения.
3. Шаговые реле общего назначения: SA/SX: S20 – S511, 492 адреса; EN: S20 – S499, 480 адресов. Свободно выбираемая область применения. Не сохраняют свое состояние при отключении питания.
4. Энергонезависимые шаговые реле: ES/EX/SS: S20 – S127, 108 адресов; SA/SX: S512 – S895, 384 адреса; EN: S500 – S899, 400 адресов. Свободно выбираемая область применения с сохранением своего состояния при отключении питания.
5. Сигнальные шаговые реле: SA/SX: S896 – S1023, 128 адресов; EN: S900 – S1023, 124 адреса. Используются совместно с инструкцией ANS (API 46) для сигнализации тревоги. Применяются для записи предупреждений и устранения внешних неисправностей.

Шаговые реле программируются как выходы. Однако отсутствует возможность присоединить к ним внешние устройства. Они могут использоваться в программе неограниченное число раз. Шаговые реле могут применяться в качестве обычных внутренних реле, если в программе не используется режим пошагового управления.

Адресация шаговых реле выполняется в десятичном формате.

DVP-ES/EX/SS

Шаговые реле S	Начальные	10 точек (S0-S9)	Макс. 128 точек
	Возвращ. в нулев. точку	10 точек (S10-S19) используются с инстр. IST	
	Общие	108 точек (S20-S127)	

DVP-SA/SX

Шаговые реле S	Начальные	10 точек (S0-S9)	Макс. 1024 точек
	Возвращ. в нулев. точку	10 точек (S10-S19) используются с инструкцией IST	
	Общие	492 точки (S20-S511)	
	Энергонезависимые	384 точки (S512-S825)	
	Сигнальные	124 точки (S896-S1023)	

DVP-EN

Шаговые реле S	Начальные	10 точек (S0-S9)	Макс. 1024 точек
	Возвращ. в нулев. точку	10 точек (S10-S19) используются с инструкцией IST	
	Общие	480 точек (S20-S499)	
	Энергонезав.*	400 точек (S500-S899)	
	Сигнальные*	124 точки (S900-S1023)	

2.6. Адресация и назначение таймеров [T]

Для многих процессов управления необходимы реле времени. В релейной технике для этого применяются реле времени с задержкой на включение или выключение. В ПЛК для этих целей используются внутренние элементы памяти, называемые таймеры, характеристики которых могут определяться программой.

Адресация таймеров выполняется в десятичном формате.

DVP-ES/EX/SS

Таймер T	Дискретность 100мс	64 точки (T0-T63)	Макс. 128 точек
	Дискретность 10мс	63 точки (T64-T126) M1028 = 1 – дискретность 10мс M1028 = 0 - дискретность 100мс	
	Дискретность 1мс	1 точка (T127)	

DVP-SA/SX

Таймер T	Дискретность 100мс	200 точек (T0-T199), T192-T199 для подпрограмм	Макс. 256 точек
		6 точек аккумулятивного типа (T250-T255), энергонезависимые	
	Дискретность 10мс	40 точек (T200-T239)	
		6 точек аккумулятивного типа (T240-T245), энергонезависимые	
	Дискретность 1мс	4 точки аккумулятивного типа (T246-T249), энергонезависимые	

DVP-EN

Таймер T	Дискретность 100мс	200 точек (T0-T199), (T192-T199 для подпрограмм), могут быть энергонезависимыми при установке спец. параметров	Макс. 256 точек
		6 точек аккумулятивного типа (T250-T255), энергонезависимые	
	Дискретность 10мс	40 точек (T200-T239), могут быть энергонезависимыми при установке спец. параметров	
		6 точек аккумулятивного типа (T240-T245), энергонезависимые	
	Дискретность 1мс	4 точки аккумулятивного типа (T246-T249), энергонезависимые	

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

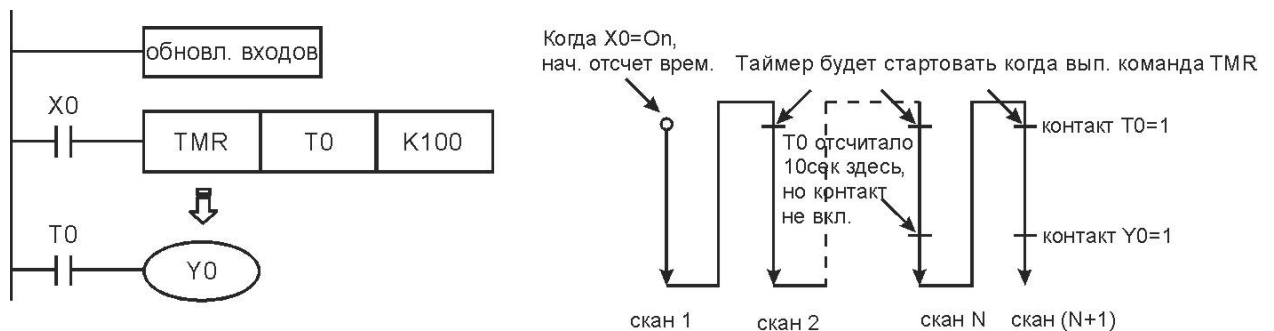
Пример: Для таймера с дискретностью 100 мс, у которого уставка времени задана как K5, действительное значение уставки будет равно $5 \times 100 = 500$ мс.

Таймер работает с задержкой на включение. Он активизируется состоянием входного контакта =1. После отсчета установленного значения времени таймер устанавливает в состояние "1" соответствующий контакт Т. Таймер возвращается в отключенное состояние и обнуляет свое текущее значение при установке своего входного контакта в "0".

Задание уставки времени может выполняться также косвенно посредством записанного ранее в регистр данных десятичного числа.

В контроллерах ES/EX/SS/SA/SX таймер начинает отчет времени после команды END в начале следующего скана.

В контроллерах EN таймер начинает отчет времени сразу с выполнением команды TMR.



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

T – заданное значение времени,

T₀ – время цикла программы,

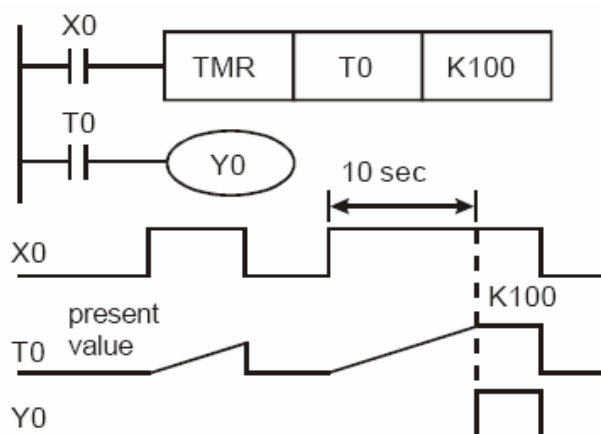
α – дискретность таймера (100 мс, 10 мс, 1 мс)

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

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

Пояснение работы трех типов таймеров:

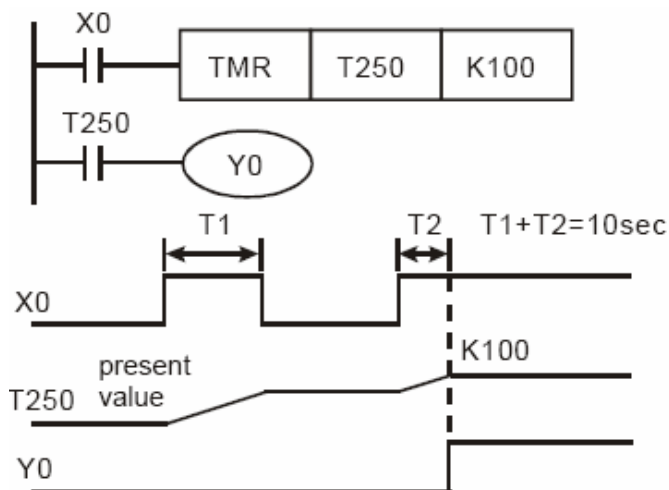
1. Таймер общего назначения.



Если вход X0 принимает состояние "1" начинается отсчет заданного времени. После отсчета запрограммированных 10 сек выход Y0 примет состояние "1". Таймер отключится и регистр T0 обнулится как только вход X0 примет состояние "0".

2. Аккумулятивный таймер (таймер с памятью).

В ПЛК серий SA/SX/EN наряду с таймерами общего назначения есть аккумулятивные таймеры, которые после отключения управляющей логической связи сохраняют накопленное значение времени. Они также являются и энергонезависимыми, т.е. текущее значение времени записывается в память, которая сохраняется при отключении питания.



3. Таймер для подпрограмм.

Если таймер используется в подпрограмме или имеет прерывание в подпрограмме, применяйте таймеры с адресами T192-T194, т.к. обычные таймеры в этих случаях не будут работать корректно.

2.7. Адресация и назначение счетчиков [C]

Для многих процессов управления необходимо считать импульсы (суммировать или вычитать). В релейной технике для этого применяются счетчики импульсов. В ПЛК для этих целей используются внутренние элементы памяти, называемые счетчики, которые могут быть нескольких видов.

Адресация счетчиков выполняется в десятичном формате.

DVP-ES/EX/SS

Счетчик C (Инкрементный 16 бит)	Общего назначения	112 точек (C0-C111)	Макс. 141 точек
	Энергонезависимый	16 точек (C112-C127)	
Высокоскоростной счетчик C (Инкр./декрем. 32 бит), энергонез.	1 фаза, 1 вход	7 точек (C235-C238, C241, C242, C244)	
	1 фаза, 2 входа	3 точки (C246, C247, C249)	
	2 фазы, 2 входа	3 точки (C251, C252, C254)	

DVP-SA/SX

Счетчик C (Инкрем. 16 бит)	Общего назначения	96 точек (C0-C95)	Макс. 250 точек
	Энергонезависимый	104 точек (C96-C199), могут быть энергозавис. при установке соотв. параметров	
Счетчик C (Инкр./декрем. 32 бит)	Общего назначения	16 точек (C200-C215)	
	Энергонезависимый	19 точек (C216-C234), могут быть энергозавис. при установке соотв. параметров	
Высокоскоростной счетчик C	1 фаза, 1 вход	9 точек (C235-C242, C244), могут быть энергозавис. при установке соотв. параметров	

(Инкр./декрем. 32 бит), энергонез.	1 фаза, 2 входа	3 точки (C246, C247, C249), могут быть энергозавис. при установке соотв. параметров	
	2 фазы, 2 входа	3 точки (C251, C252, C254), могут быть энергозавис. при установке соотв. параметров	

DVP-EN

Счетчик С (Инкрем. 16 бит)	Общего назначения	100 точек (C0-C99), могут быть энергонезавис. при установке соотв. параметров	Макс. 253 точек
	Энергонезависимый	100 точек (C100-C199), могут быть энергозавис. при установке соотв. параметров	
Счетчик С (Инкр./декрем. 32 бит)	Общего назначения	20 точек (C200-C219), могут быть энергонезавис. при установке соотв. парам.	
	Энергонезависимый	15 точек (C220-C234), могут быть энергозавис. при установке соотв. параметров	
Высокоскоростной счетчик С (Инкр./декрем. 32 бит), энергонез.	1 фаза, 1 вход (программный счетчик)	6 точек (C235-C240), могут быть энергозавис. при установке соотв. параметров	
	1 фаза, 1 вход (аппаратный счетчик)	4 точки (C241-C244), могут быть энергозавис. при установке соотв. параметров	
	1 фаза, 2 входа (аппаратный счетчик)	4 точки (C246- C249), могут быть энергозавис. при установке соотв. параметров	
	2 фазы, 2 входа (аппаратный счетчик)	4 точки (C251- C254), могут быть энергозавис. при установке соотв. параметров	

Свойства:

Элемент	16-ти битный счетчик	32-х битный счетчик	
Тип	Общий	Общий	Высокоскоростной
Направление счета	Вверх (суммирование)	Вверх/вниз	
Диапазон счета	0 ... 32 767	-2 147 483 648 ... +2 147 483 647	
Тип уставки	Константа К или регистр данных D	Константа К или регистр данных D (2 слова)	
Изменение текущего значения	Счет прекратится при достижении уставки	Счет будет продолжаться после достижения уставки	
Рабочий контакт	При достижении уставки контакт включится и зафиксируется	При текущем значении счета большем заданного контакт будет включен, при текущем значении счета меньшем заданного контакт будет выключен	
Сброс счетчика	Текущее значение счетчика будет обнулено и контакт С возвращен в исходное положение с помощью команды RST.		
Быстродействие выхода	Выход счетчика будет обновлен в конце цикла сканирование вместе с другими	Выход счетчика будет обновлен в конце цикла сканирование вместе с другими	Выход счетчика будет обновлен немедленно при достижении уставки, не зависимо от цикла сканирования

Работа и назначение счетчиков:

Когда входной сигнал счетчика изменяет свое состояние с 0 на 1, текущее значение счетчика С увеличится/уменьшится на единицу и когда оно станет равным заданному значению (уставке), рабочий контакт счетчика включится.

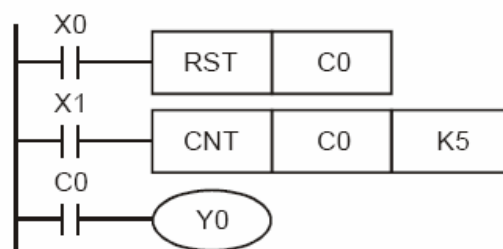
16-ти битный счетчик C0 – C199:

1. Диапазон заданных значений: K0 ... K32 767 (При K0 так же как и при K1, рабочий контакт будет замкнут после первого счета).
2. Общий счетчик будет обнулен при отключении питания ПЛК. Энергонезависимый счетчик сохранит свое текущее значение при отключении питания.

3. Если используется команда MOV, WPLSoft или программатор для изменения заданной уставки счетчика и будет записано значение больше, чем C0, при уже включенном контакте C0, то контакт C0 сохранит свое состояние и текущее значение C0 будет таким же как заданное.
4. Для задания уставки счетчика может использоваться десятичная константа K или регистр данных D (кроме специальных регистров D1000 – D1999) для косвенной уставки.
5. Если для задания уставки используется десятичная константа K, то значения могут быть только положительными, а при использовании регистра – положительными и отрицательными в диапазоне от -32 768 до +32 767

Пример:

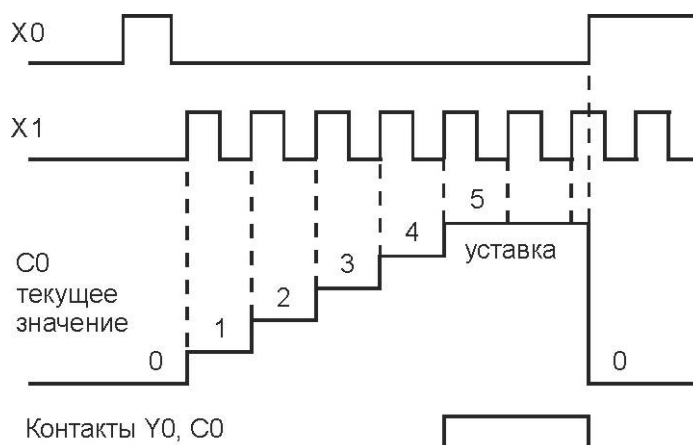
```
LD    X0
RST   C0
LD    X1
CNT   C0 K5
LD    C0
OUT   Y0
```



Когда X0 = 1, происходит сброс счетчика: текущее значение регистра C0 = 0, контакт C0 разомкнут.

При изменении X1 с 0 на 1, текущее значение регистра C0 будет увеличиваться на 1.

Когда C0 = 5, контакты C0 и Y0 замкнутся и последующие импульсы контакта X1 перестанут восприниматься.

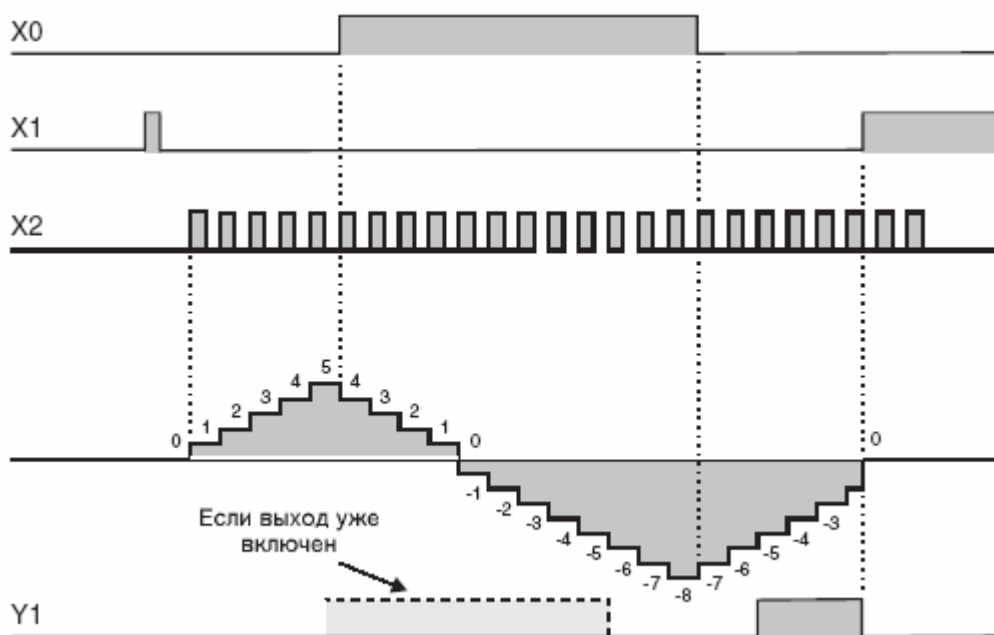
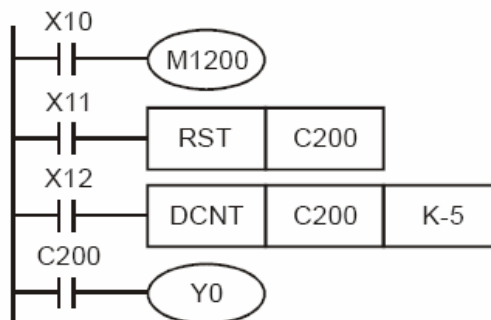
**32-х битный счетчик общего назначения C200 – C234 (нет в DVP-ES/EX/SS):**

1. Диапазон заданных значений: K-2 147 483 648 ... K2 147 483 647.
2. Для изменения направления счета (суммирование или вычитание) используются специальные реле M1200 – M1234. Так если M1200=0, счетчик C200 будет суммирующим, а если M1200=1, то вычитающим.
3. Общий счетчик будет обнулен при отключении питания ПЛК. Энергонезависимый счетчик сохранит свое текущее значение при отключении питания.
4. Для задания уставки счетчика может использоваться десятичная константа K или два регистра данных D (кроме специальных регистров D1000 – D1999) для косвенной уставки.
5. Для суммирующего счетчика при достижении максимального значения 2 147 483 647 следующим значением будет - 2 147 483 648, аналогично для

вычитающего счетчика следующим значением после -2 147 483 648 будет 2 147 483 647. Этот счетчик является "кольцевым".

Пример:

```
LD    X10
OUT   M1200
LD    X11
RST   C200
LD    X12
CNT   C200 K-5
LD    C200
OUT   Y0
```



Контакт X10 определяет направление счета.

Когда X11 = 1, происходит сброс счетчика: текущее значение регистра C200 = 0, контакт C200 разомкнут.

При изменении X12 с 0 на 1, текущее значение регистра C200 будет увеличиваться на 1, если X10 разомкнут, или уменьшаться на 1, если X10 замкнут.

Выход Y0 включится, если значение регистра C200 перейдет с -6 на -5 и выключится если значение регистра C200 перейдет с -5 на -6.

32-х битный высокоскоростной счетчик C235 – C254:

1. Диапазон заданных значений: K-2 147 483 648 ... K2 147 483 647.
2. Для изменения направления счета (суммирование или вычитание) счетчиков C235 – C244 используются специальные реле M1235 – M1244. Так если M1235=0, счетчик C235 будет суммирующим, а если M1235=1, то вычитающим.
3. Для изменения направления счета (суммирование или вычитание) счетчиков C246 – C254 используются специальные реле M1246 – M1254.

Так если M1246=0, счетчик C246 будет суммирующим, а если M1246=1, то вычитающим.

4. Для задания уставки счетчика может использоваться десятичная константа К или два регистра данных D (кроме специальных регистров D1000 – D1999) для косвенной уставки.
5. Для суммирующего счетчика при достижении максимального значения 2 147 483 647 следующим значением будет - 2 147 483 648, аналогично для вычитающего счетчика следующим значением после -2 147 483 648 будет 2 147 483 647. Этот счетчик является "кольцевым".
6. Если используется команда DMOV, WPLSoft или программатор для изменения заданной уставки счетчика и будет записано значение больше, чем С, при уже включенном контакте С, то контакт С сохранит свое состояние и счет будет продолжаться с текущего значения С.

Высокоскоростные счетчики работают по принципу прерывания и сигналы счета независимы от времени обработки программы.

Высокоскоростным счетчикам и соответствующим входам присвоены жестко заданные функции.

Назначение входов и регистров высокоскоростных счетчиков ПЛК серии ES/EX/SS

Вход	1 фаза 1 входа							1 фаза 2 входа			2 фазный вход		
	C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
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			R	R		R	R
X3				U/D		R	S			S			S

U: суммирующий счетный вход;

D: вычитающий счетный вход;

R: сброс счетчика;

A: фаза А двухфазного реверсивного счетчика;

В: фаза В двухфазного реверсивного счетчика;

S: старт счетчика

1. Максимальная частота высокоскоростных счетчиков ES/EX/SS: 20 кГц. Она является суммарной для всех используемых скоростных счетчиков.
2. Высокоскоростные однофазные счетчики в контроллерах серии ES/EX/SS на входах X0 и X1 могут иметь частоту счета до 20 кГц. Однако при совместном использовании двух этих входов частота 20 кГц является суммарной для обоих скоростных счетчиков.
3. Высокоскоростные однофазные счетчики в контроллерах серии ES/EX/SS на входах X2 и X3 могут иметь частоту счета до 10 кГц.
4. Частота двухфазного счетчика: 5 кГц.
5. Направление счета (суммирование или вычитание) определяется состояниями специальных реле M1235 - M1254.
6. Команды DHSCS и DHSCR не могут использоваться в программе более 4 раз.

Назначение входов и регистров высокоскоростных счетчиков ПЛК серии SA/SX

[illegible]

U: суммирующий счетный вход;
D: вычитающий счетный вход;
R: сброс счетчика;

A: фаза А двухфазного реверсивного счетчика;
B: фаза В двухфазного реверсивного счетчика;
S: старт счетчика

1. Максимальная частота высокоскоростных счетчиков SA/SX: 40 кГц. Она является суммарной для всех используемых скоростных счетчиков.
2. Высокоскоростные однофазные счетчики в контроллерах серии SA/SX на входах X0 и X1 могут иметь частоту счета до 20 кГц. Однако при совместном использовании двух этих входов частота 20 кГц является суммарной для обоих скоростных счетчиков.
3. Высокоскоростные однофазные счетчики в контроллерах серии SA/SX на входах X2, X3, X4, X5 могут иметь частоту счета до 10 кГц.
4. Вход X5 может иметь две функции:
M1260 = OFF: счетный U/D вход;
M1260 = ON: одновременный сброс счетчиков C235 – C239.
5. Направление счета (суммирование или вычитание) определяется состояниями специальных реле M1235 - M1254.
6. Команды DHSCS DHSCR и DHCZ не могут использоваться в программе более 6 раз.

Назначение входов и регистров высокоскоростных счетчиков ПЛК серии EN

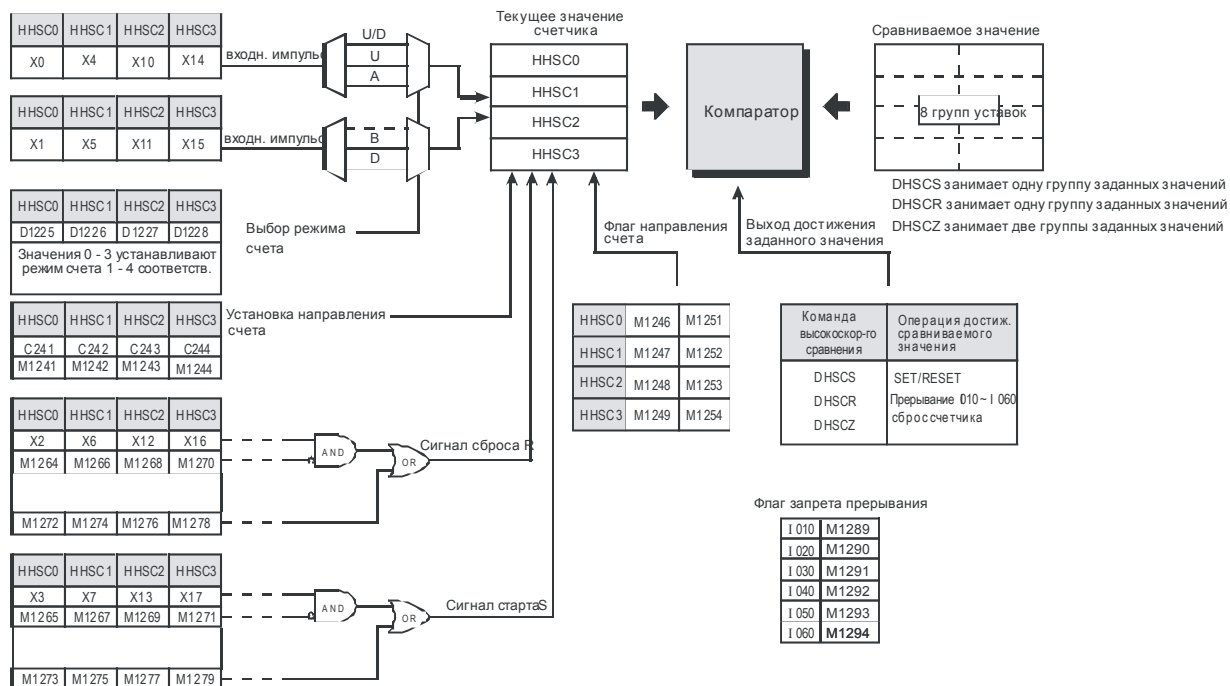
Вход	Счетчики, работающие по программным прерываниям						Аппаратные высокоскоростные счетчики											
	1 фаза 1 вход						1 фаза 1 вход				1 фаза 2 входа				2 фазный вход			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C246	C247	C248	C249	C251	C252	C253	C254
X0	U/D						U/D				U				A			
X1		U/D									D				B			
X2			U/D				R				R				R			
X3				U/D			S				S				S			
X4					U/D			U/D				U				A		
X5						U/D						D				B		
X6								R				R				R		
X7								S				S				S		
X10									U/D				U				A	
X11													D				B	
X12								R					R				R	
X13								S					S				S	
X14										U/D				U				A
X15														D				B
X16										R				R				R
X17										S				S				S

U: суммирующий счетный вход;
D: вычитающий счетный вход;
R: сброс счетчика;

A: фаза А двухфазного реверсивного счетчика;
B: фаза В двухфазного реверсивного счетчика;
S: старт счетчика

1. Частота однофазного счетчика (C235 – C240) работающего по программным прерываниям: 10 кГц. Максимальная частота может быть до 20 кГц. Аппаратные счетчики разбиты на две группы по два счетчика. Максимальная частота 200 кГц относится к каждому из используемых 2-х высокоскоростных счетчиков, а частота 30 кГц является суммарной для остальных скоростных счетчиков.
NHSC0 (C241, C246, C251) и NHSC1 (C242, C247, C252) – до 200 кГц;
NHSC2 (C243, C246, C253) и NHSC3 (C244, C249, C254) – до 30 кГц;
2. Сброс и старт аппаратных счетчиков NHSC0 – 3 также могут выполнены с помощью специальных реле M1272 – M1279.

3. Функции Сброса и Старта от внешних входов можно отключить специальными реле M1264 – M1271 и использовать данные входы в программе для других целей.
4. Направление счета (суммирование или вычитание) определяется состояниями специальных реле M1235 - M1254
5. Команды DHSCS DHSCR и DHCZ могут использоваться в программе неограниченное число раз.



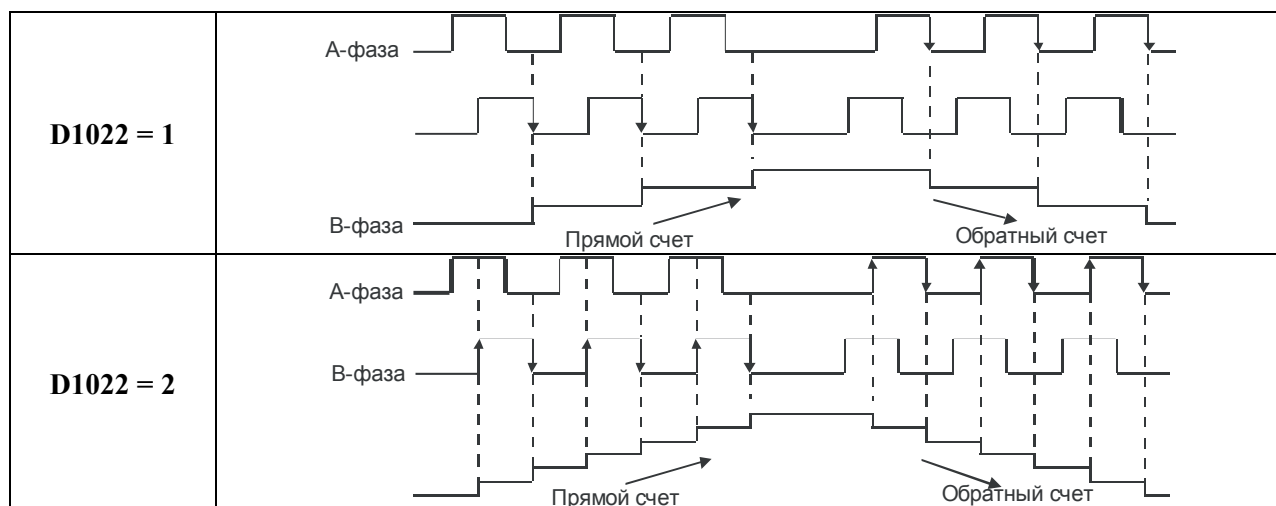
Выбор режимов работы счетчиков.

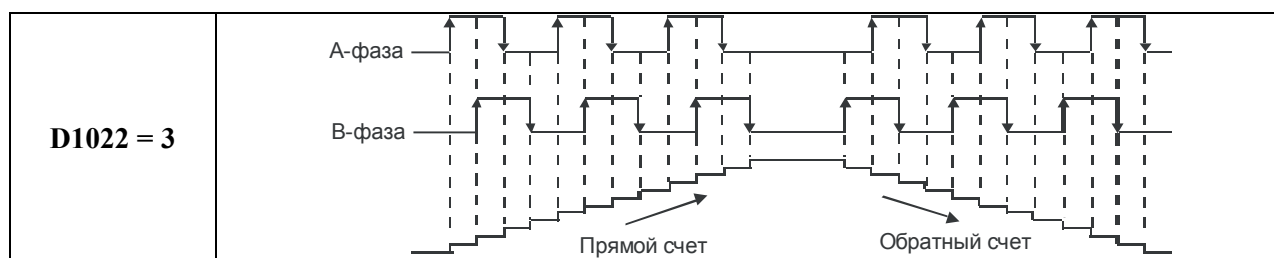
В ПЛК ES/EX/SS/SA/SX режим работы 2-фазного счетчика задается в регистре D1022:

D1022 = 1: нормальный режим счета;

D1022 = 2: двойная частота счета (заводская установка);

D1022 = 3: четверная частота счета.





В ПЛК EN режим работы аппаратных счетчиков задается в регистрах D1225 – D1228:

Тип	D1225 ... D1228 =	Прямой счет (+1)	Обратный счет (-1)
1 фаза 1 вход	1	U/D	U/D
	2	U/D	U/D
1 фаза 2 входа	1	U	D
	2	U	D
2 фазный вход	1	A	B
	2	A	B
	3	A	B
	4	A	B

Адреса и специальные регистры высокоскоростных счетчиков

Номер	Функция
M1153	Завершение выполнения режима контроля частоты
M1235 - M1244	Выбор направления счета для счетчиков C235 - C244 (0: суммирование; 1: вычитание)
M1246 – M1249 M1251 – M1254	Индикация направления счета счетчиков C246 – C249 и C251 – C254. (0: суммирование; 1: вычитание)
M1264	Запрещение внешнего старта счетчика HHSC0
M1265	Запрещение внешнего сброса счетчика HHSC0
M1266	Запрещение внешнего старта счетчика HHSC1
M1267	Запрещение внешнего сброса счетчика HHSC1
M1268	Запрещение внешнего старта счетчика HHSC2
M1269	Запрещение внешнего сброса счетчика HHSC2
M1270	Запрещение внешнего старта счетчика HHSC3
M1271	Запрещение внешнего сброса счетчика HHSC3
M1272	Сброс контроля HHSC0
M1273	Старт контроля HHSC0
M1274	Сброс контроля HHSC1
M1275	Старт контроля HHSC1
M1276	Сброс контроля HHSC2
M1277	Старт контроля HHSC2
M1278	Сброс контроля HHSC3

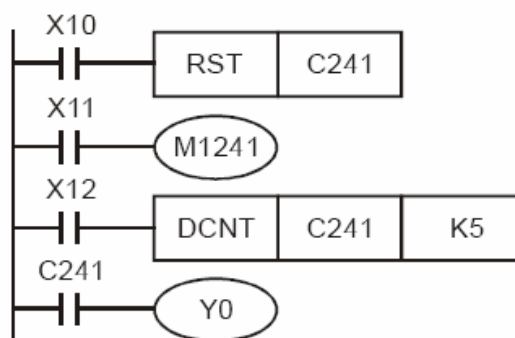
M1279	Старт контроля HHSC3
M1289	I010 флаг запрета прерывания
M1290	I020 флаг запрета прерывания
M1291	I030 флаг запрета прерывания
M1292	I040 флаг запрета прерывания
M1293	I050 флаг запрета прерывания
M1294	I060 флаг запрета прерывания
M1312	C235 старт входа
M1313	C236 старт входа
M1314	C237 старт входа
M1315	C238 старт входа
M1316	C239 старт входа
M1317	C240 старт входа
M1320	C235 сброс входа
M1321	C236 сброс входа
M1322	C237 сброс входа
M1323	C238 сброс входа
M1324	C239 сброс входа
M1325	C240 сброс входа
M1328	C235 разрешение функции старт/сброс
M1329	C236 разрешение функции старт/сброс
M1330	C237 разрешение функции старт/сброс
M1331	C238 разрешение функции старт/сброс
M1332	C239 разрешение функции старт/сброс
M1333	C240 разрешение функции старт/сброс
D1022	Удвоенная частота для двухфазного счетчика AB (DVP-ES/EX/SS/SA/SX)
D1150	Регистр для записи результата в режиме группового сравнения
D1151	Регистр для записи результата в режиме частотного управления
D1152	Изменение значения старшего слова DHSZ
D1153	Изменение значения младшего слова DHSZ
D1225	Первая группа счетчиков (HHSC0). Счетный регистры: C241, C246, C251. Выбор режима
D1226	Вторая группа счетчиков (HHSC1). Счетный регистры: C242, C247, C252. Выбор режима
D1227	Третья группа счетчиков (HHSC2). Счетный регистры: C243, C248, C253. Выбор режима
D1228	Четвертая группа счетчиков (HHSC3). Счетный регистры: C244, C249, C254. Выбор режима
D1225 - D1228	Выбор режима счета для двухфазных высокоскоростных аппаратных счетчиков HHSC0 – HHSC3 контроллеров DVP-EN. 1: нормальная частота счета; 2: двойная частота (заводская уставка); 3: тройная частота; 4: четверная частота счета.

Пример работы однофазного высокоскоростного счетчика с одним счетным входом:

```

LD    X10
RST   C241
LD    X11
OUT   M1241
LD    X12
DCNT  C241 K5
LD    C241
OUT   Y0

```

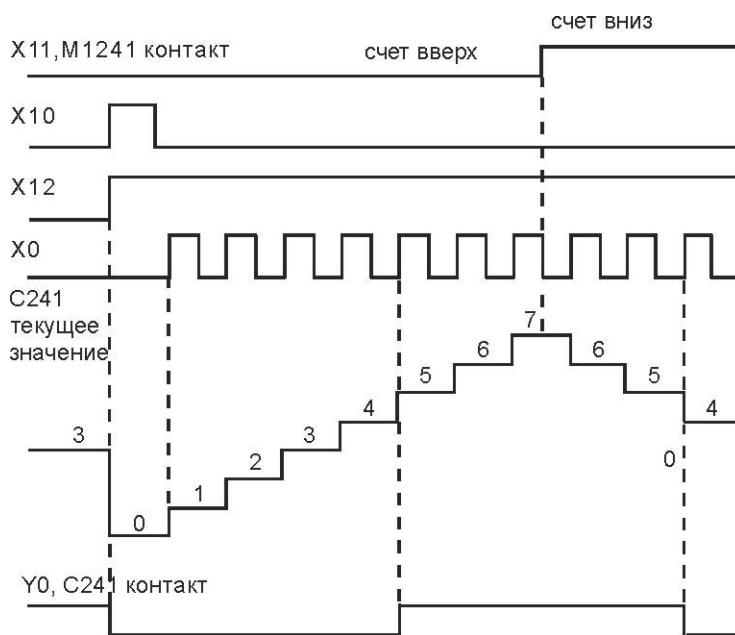


Контакт X11 устанавливает специальное реле M1241 в состояние 0 или 1, и тем самым выбирает направление счета (суммирование или вычитание).

С помощью контакта X10 осуществляется сброс счетчика: текущее значение C241 обнуляется и контакт C241 размыкается.

При включении X12 счетчик C241 активизируется и считает импульсы по входу X0.

Выход Y0 включится, если значение регистра C241 перейдет с 4 на 5 и выключится, если значение регистра C241 перейдет с 5 на 4.



В контроллерах DVP-ES/EX/SS/SA/SX счетчик C241 имеет вход внешнего сброса X1, который работает по прерыванию и не зависит от времени цикла сканирования программы.

В контроллерах DVP-EN счетчик C241 имеет вход внешнего сброса X2 и вход внешнего старта (разрешения счета) X3, которые работают по прерыванию и не зависят от времени цикла сканирования программы. Разрешение сброса и старта от внешних входов можно отключить с помощью специальных реле M1264, M1265 и осуществлять старт и сброс счетчика с помощью специальных внутренних реле M1272, M1273.

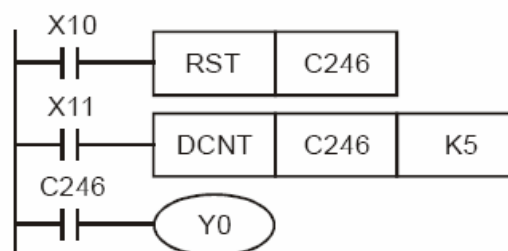
Изменить режим работы счетчика (одинарная или двойная частота счета) C241 можно в специальном регистре D1225.

Пример работы однофазного высокоскоростного счетчика с двумя счетными входами:

```

LD    X10
RST   C246
LD    X11
DCNT  C246 K5
LD    C246
OUT   Y0

```



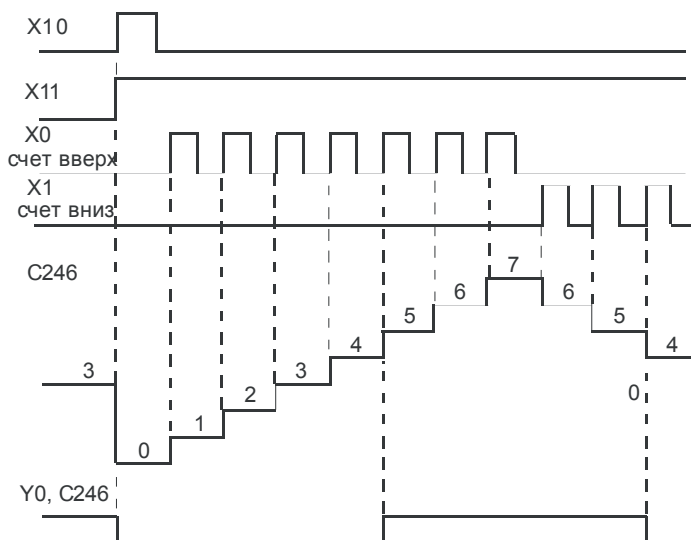
С помощью контакта X10 осуществляется сброс счетчика: текущее значение C246 обнуляется и контакт C246 размыкается.

При включении X11 счетчик C246 активизируется и считает импульсы по входу X0 – суммирует (+1), и по входу X1 – вычитает (-1).

Выход Y0 включится, если значение регистра C246 перейдет с 4 на 5 и выключится если значение регистра C246 перейдет с 5 на 4.

В контроллерах DVP-EN счетчик C246 имеет вход внешнего сброса X2 и вход внешнего старта (разрешения счета) X3, которые работают по прерыванию и не зависят от времени цикла сканирования программы. Разрешение сброса и старта от внешних входов можно отключить с помощью специальных реле M1264, M1265 и осуществлять старт и сброс счетчика с помощью специальных внутренних реле M1272, M1273.

Изменить режим работы счетчика (одинарная или двойная частота счета) C246 можно в специальном регистре D1225.

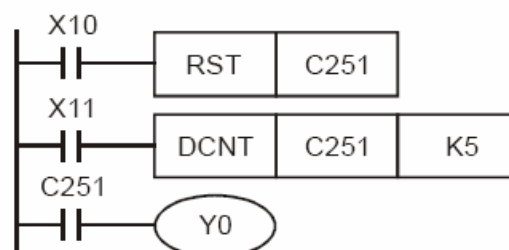


Пример работы двухфазного (AB) высокоскоростного счетчика с двумя счетными входами:

```

LD    X10
RST   C251
LD    X11
DCNT  C251 K5
LD    C251
OUT   Y0

```



С помощью контакта X10 осуществляется сброс счетчика: текущее значение C251 обнуляется и контакт C251 размыкается.

При включении X11 счетчик C251 активизируется и считает импульсы по входам X0 и X1 сдвинутые между собой на 90°, причем если импульсы на входе X0 (фаза А) опережают импульсы входа X1 (фаза В), то счет идет вверх (+1), а если фаза В опережает

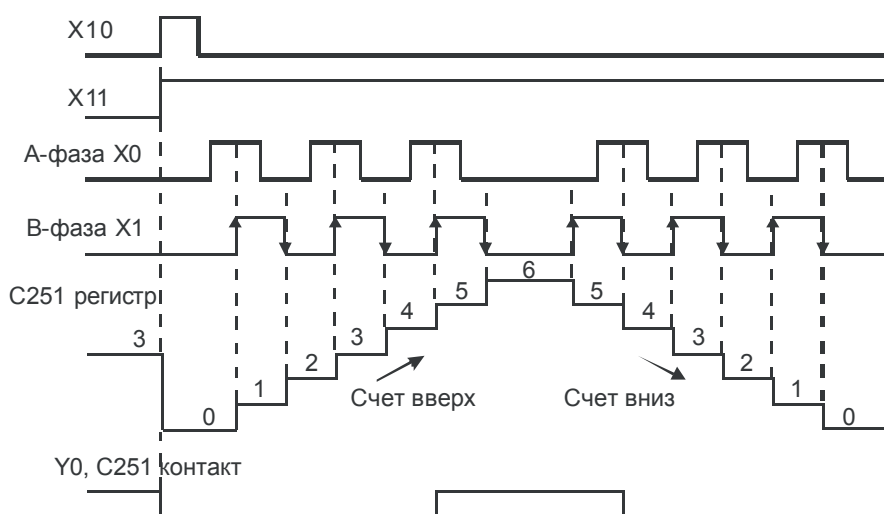
фазу А, то счет идет вниз (-1). Входы X0 и X1 жестко зарезервированы для счетных входов АВ счетчика C251 и не могут использоваться в программе для других целей.

Выход Y0 включится, если значение регистра C251 перейдет с 4 на 5 и выключится если значение регистра C251 перейдет с 5 на 4.

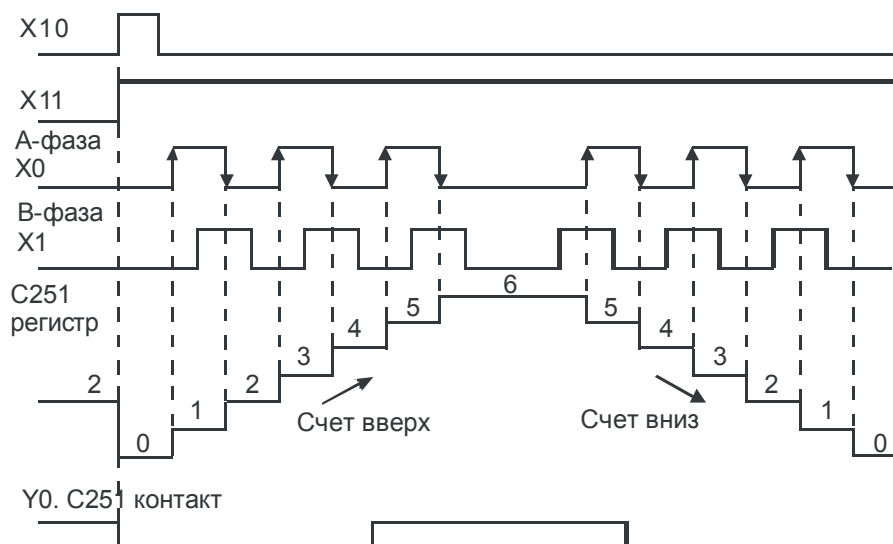
В контроллерах DVP-ЕН счетчик C251 имеет вход внешнего сброса X2 и вход внешнего старта (разрешения счета) X3, которые работают по прерыванию и не зависят от времени цикла сканирования программы. Разрешение сброса и старта от внешних входов можно отключить с помощью специальных реле M1264, M1265 и осуществлять старт и сброс счетчика с помощью специальных внутренних реле M1272, M1273.

В DVP- ES/EX/SS/SA/SX изменить режим работы счетчика (одинарная, двойная, тройная или четверная частота счета) C251 можно в специальном регистре D1022. В DVP- EN изменить режим работы счетчика (одинарная, двойная, тройная или четверная частота счета) C251 можно в специальном регистре D1225.

ES/EX/SS,SX /SA серия (двойная частота счета):



ЕН серия:(двойная частота счета)



2.8. Адресация и назначение регистров [D], [E], [F]

Регистры данных [D]

Регистры представляют память данных внутри ПЛК. В регистре можно хранить числовые значения и следующую друг за другом двоичную информацию.

Данные сохраняются в 16-ти битном регистре (D0 и др), в котором может храниться число от -32768 до +32767. Благодаря совместного включения двух 16-ти битных регистров можно образовать 32-х битный "Двойной регистр"(D0, D1 и т.д), в котором может храниться число от -2147483648 до +2147483647.

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

DVP-ES/EX/SS

Регистр данных D	Общего назначения	408 точек (D0-D407)	Макс. 744 точек
	Энергонезависимый	192 точек (D408-D599)	
	Специальный	144 точки (D1000-D1143)	
	Индексный	2 точки: E(=D1028), F(=D1029)	

DVP-SA/SX

Регистр данных D	Общего назначения	200 точек (D0-D199)	Макс. 5000 точек
	Энергонезависимый	3800 точек (D200-D999, D2000-D4999)	
	Специальный	1000 точек (D1000-D1999)	
	Индексный	8 точек: E0-E3, F0-F3	
Файловый регистр		1600 точек (K0-K1599)	

DVP-EN

Регистр данных D	Общего назначения	200 точек (D0-D199)	Макс. 10000 точек
	Энергонезависимый	800 точек (D200-D999) 8000 точек (D2000-D9999)	
	Специальный	1000 точек (D1000-D1999)	
	Индексный	16 точек: E0-E7, F0-F7	
Файловый регистр		10000 точек (K0-K9999)	

Имеются следующие типы регистров:

1. Регистр данных (общего назначения):

Регистр без сохранения данных при отключении напряжения ПЛК

2. Регистр данных (энергонезависимый):

Регистр с сохранением данных при отключении напряжения ПЛК. Данные хранятся в энергонезависимой памяти.

3. Индексный регистр:

Этот регистр служит для запоминания промежуточных результатов и для индентификации операндов. Более подробные данные см. ниже.

4. Специальный регистр:

Для определенных контрольных и проверочных функций предусмотрен ряд специальных регистров. Подробнее см. раздел 2.10.

5. Файловый регистр:

Для сохранения параметров или рецептуры удобны файловые регистры. Они есть в ПЛК серий SA/SX и EN. Эти регистры являются энергонезависимыми и для записи /

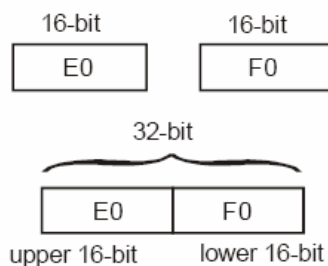
считывания в них/из них данных необходимо использовать команды MEMR (API 147) и MEMW (API 148) или программатор, или WPLSoft.

Индексные регистры [E], [F]

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

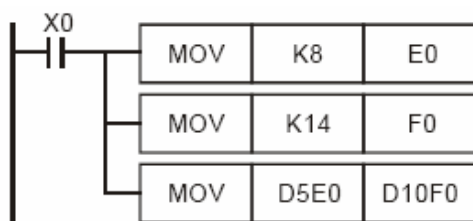
Индексный регистр является 16-ти битовым регистром.

В 32-х битовых инструкциях индексные регистры E и F применяются комбинированно. F содержит 16 младших бит, E запоминает 16 старших бит. В качестве адреса назначения указывается индексный регистр F.



Индексный регистр не может самостоятельно индизироваться.

Пример передачи данных от регистра данных D5E0 регистру данных D10F0:



Когда X0 = 1: E0 = 8, F0 = 14 и значит

адрес источника пересылки данных D5E0 = 5 + 8 = D13, а адрес пересылки D10F0 = 10 + 14 = 24:

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

Индексные регистры могут использоваться для операций передачи и сравнения данных совместно с байтовыми операндами (KnX, KnY, KnM, KnS, D, T, C) и битовыми операндами (X, Y, M, S).

В контроллерах серии EN можно индексировать так же и константы (K, H).

При индексировании констант в командном режиме WPLSoft необходимо использовать символ @. Например: MOV K10@E0 D0F0.

Файловые регистры:

Файловые регистры есть в ПЛК серий SA/SX/EN. Они отличаются от регистров общего назначения тем, что к ним не возможно прямое обращение, а только с помощью команд MEMR (API 147), MEMW (API 148) или программатора, или WPLSoft.

Область чтения и записи данных в файловый регистр задается в командах MEMR (API 147), MEMW (API 148).

Для автоматической передачи данных при включении ПЛК или переключении его из режима STOP в режим RUN область чтения и записи данных в файловый регистр может быть задана в специальных регистрах:

D1101 определяет стартовый адрес файлового регистра (в DVP-SA/SX: K0...K1599; в DVP-EN: K0...K9999).

D1102 определяет число читаемых файловых регистров (в DVP-SA/SX: K0...K1600; в DVP-EN: K0...K10000).

D1103 определяет стартовый адрес для сохранения в файловый регистр (должен быть больше 2000).

M1101 разрешает/запрещает запись/чтение файловых регистров.

Если адрес файлового регистра при чтении выходит из пользовательского диапазона, в назначенные регистры будут записаны ноли.

2.9. Указатели [N], [P], [I]

DVP-ES/EX/SS

Указатели	N	Номера вложенности для инструкций MC, MCR		8 точек (N0-N7).	Используются для нумерации вложенных схем исключения мастер-контроля.
	P	Указатели для инструкций CJ, CALL		64 точек (P0-P63)	Метки для операторов перехода, подпрограмм
	I	Прерывания	Внешние	4 точки (I001, I101, I201, I301)	Метки для подпрограмм обработки прерываний
			Временные	1 точка (I6xx = 10-99, дискр. 1 мс)	
			Коммуникац.	1 точка (I150)	

DVP-SA/SX

Указатели	N	Номера вложенности для инструкций MC, MCR		8 точек (N0-N7).	Используются для нумерации вложенных схем исключения мастер-контроля.
	P	Указатели для инструкций CJ, CALL		256 точек (P0-P255)	Метки для операторов перехода, подпрограмм
	I	Прерывания	Внешние	6 точек с включением по переднему фронту: I001(X0), I101(X1), I201(X2), I301(X3), I401(X4), I501(X5)	Метки для подпрограмм обработки прерываний
			Временные	I6xx (1мс), I7xx (1мс) (xx= 10-99)	
			Быстр. счетч.	I010, I020, I030, I040, I050, I060	
			Коммуникац.	1 точка (I150)	

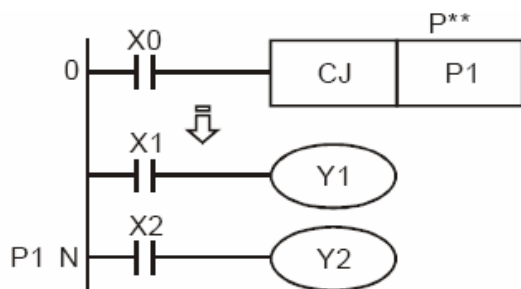
DVP-EN

Указатели	N	Номера вложенности для инструкций MC, MCR		8 точек (N0-N7).	Используются для нумерации вложенных схем исключения мастер-контроля.
	P	Указатели для инструкций CJ, CALL		256 точек (P0-P255)	Метки для операторов перехода, подпрограмм
	I	Прерывания	Внешние	6 точек с включением по переднему фронту (x=1) или по заднему (x=0): I00x(X0), I10x(X1), I20x(X2), I30x(X3), I40x(X4), I50x(X5)	Метки для подпрограмм обработки прерываний
			Временные	I6xx (1мс), I7xx (1мс), I8xx (0.1мс) (xx= 10-99)	
			Быстр. счетч.	I010, I020, I030, I040, I050, I060	
			Импульс.вых.	4 точки (I110, I120, I130, I140)	
			Коммуникац.	2 точки (I150, I160)	

Указатели (N) используются с командами мастер-контроля MC и MCR для обозначения глубины вложенности схем исключения. Мастер-контроль позволяет исключить выполнение части программы, расположенной между командами MC и MCR. Схемы исключения могут быть вложены одна в другую до 8 раз и указатели N обозначают глубину вложенности: N0 – N7. (подробнее о мастер-контроле в разделе 3.7)

Указатели (P) используются вместе с инструкциями CJ-перехода или CALL-вызова подпрограммы и являются адресами места перехода, в которых маркируется место перехода или подпрограмма в программе (подробнее в разделе 5.5).

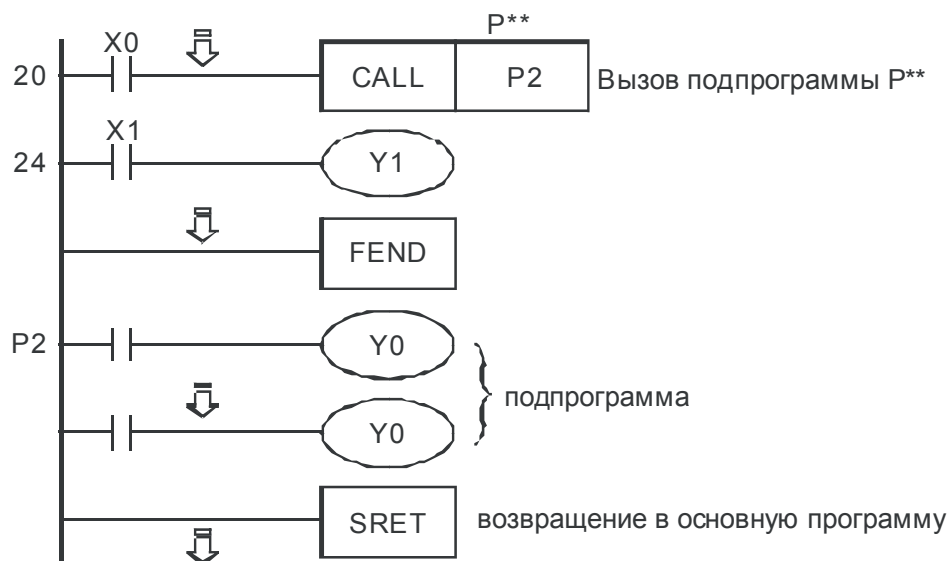
Пример выполнения команды перехода CJ:



Когда X0 = 1, после выполнения строки 0 программа сразу переходит к строке с указателем P1 и строки расположенные между ними не выполняются.

Если X0 = 0, программа выполняется нормальным образом шаг за шагом.

Пример использования подпрограмм:



Когда X0 = 1, после выполнения строки 20 программа сразу переходит к строке с указателем P2 и выполняется подпрограмма, после инструкции SRET выполнение подпрограммы завершается и продолжается выполнение основной программы со строки 24.

Указатели (I) используются вместе с инструкциями EI (API 04), DI (API 05), IRET (API 03) для прерывания выполнения основной программы и бывают следующих типов (подробнее в разделе 5.5):

1. Внешние прерывания: когда сигнал на входе X0 – X5 переключается с 0 на 1 или с 1 на 0 немедленно (независимо от цикла сканирования) происходит переход к выполнению подпрограммы обработки прерывания с соответствующим указателем I (X0 -> I0, X1 -> I1 и т.д.). Возврат в главную программу происходит после того, как будет выполнена инструкция IRET.

2. Прерывания по времени: подпрограмма обработки прерывания будет выполняться автоматически через заданные промежутки времени (10...99 мс). Возврат в главную программу происходит после того, как будет выполнена инструкция IRET.
3. Прерывания от высокоскоростных счетчиков: используются совместно с командой сравнения DHSCS (API 53) для вызова программы обработки прерывания (I010, I020, I030, I040, I050, I060) при достижении высокоскоростным счетчиком сравниваемого значения.
4. Прерывания от импульсных выходов: используются совместно с командой PLSY (API 57) для передачи вектора прерывания I130 (соотв. M1342) и I140 (соотв. M1343) одновременно с началом генерации импульсов (с первым импульсом на выходе ПЛК), а так же для передачи вектора прерывания I110 (соотв. M1340) и I120 (соотв. M1341) одновременно с завершением генерации импульсов (с последним импульсом на выходе ПЛК).
5. Коммуникационные прерывания: используются совместно с командой RS для вызова подпрограммы обработки прерывания I150, когда длина принятых данных не соответствует установленной в регистре D1168.

2.10. Специальные реле и регистры

Специальные реле и регистры могут использоваться в программе как флаги и дополнительные функции, а также для настройки, мониторинга и конфигурирования системы. Все специальные реле по умолчанию находятся в выключенном состоянии (OFF), а специальные регистры могут иметь заводские установки. Некоторые из специальных реле и регистров могут быть энергонезависимыми (* в столбце «Атрибут»), т.е. сохранять свое состояние и значение в выключенном состоянии ПЛК. Значения некоторых специальных реле и регистров можно изменять (R/W – в столбце «Атрибут»), а некоторые предназначены только для чтения (R – в столбце «Атрибут»). Состояние ON означает, что контакт замкнут, а OFF – разомкнут.

2.10.1. Специальные реле

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1000	Нормально-открытый контакт. Контакт замкнут, когда на ПЛК подано напряжение питания, и он находится в состоянии RUN.	R	+	+	+
M1001	Нормально-закрытый контакт. Контакт разомкнут, когда на ПЛК подано напряжение питания, и он находится в состоянии RUN.	R	+	+	+
M1002	Контакт замыкается при включении ПЛК во время первого цикла выполнения программы на период, равный периоду сканирования. Все остальное время контакт разомкнут.	R	+	+	+
M1003	Контакт размыкается при включении ПЛК во время первого цикла выполнения программы на период, равный периоду сканирования. Все остальное время контакт замкнут.	R	+	+	+
M1004	Замыкается при возникновении ошибки.	R	+	+	+
M1005	Пароль в карте памяти не совпадает с паролем в ПЛК	R	-	-	+
M1006	Карта памяти не инициализирована	R	-	-	+
M1007	Данные отсутствуют в области программы карты памяти.	R	-	-	+
M1008	Флаг сторожевого таймера. (ON когда вышло время WDT).	R	+	+	+
M1009	M1009=ON, когда уровень сигнала 24 В DC ниже допустимого (LV).	R	+	+	+
M1010	ES/EX/SS/SA: выбор режима PLSY импульсного выхода Y0. Когда M1010=ON импульсы идут непрерывно. EN: M1010=ON импульсы на Y0, Y1, Y2, Y3 будут идти до выполнения инструкции END.	R/W	+	+	+
M1011	Генератор импульсов с периодом 10мс (ON= 5 мс, OFF=5 мс)	R	+	+	+
M1012	Генератор импульсов с периодом 100мс (ON= 50 мс, OFF=50 мс)	R	+	+	+
M1013	Генератор импульсов с периодом 1 сек (ON= 0.5 сек, OFF=0.5 сек)	R	+	+	+
M1014	Генератор импульсов с периодом 1 мин (ON= 30 сек, OFF = 30 сек)	R	+	+	+
M1015	Пуск высокоскоростного таймера	R/W	-	+	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1016	Отображение года в часах реального времени: OFF – 2 цифры, ON – 4 цифры	R/W	-	+	+
M1017	Корректировка секунд (± 30 сек)	R/W	-	+	-
M1018	Флаг: радианы/градусы. M1018 = ON - градусы	R/W	-	+	+
M1019	Флаг запуска работы карты измерения частоты	R	-	-	+
M1020	Флаг нуля (Zero)	R	+	+	+
M1021	Флаг заема (Borrow)	R	+	+	+
M1022	Флаг переноса (Carry)	R	+	+	+
M1023	Выбор режима PLSY импульсного выхода Y1. Когда M1023=ON импульсы идут непрерывно.	R/W	+	+	-
M1024	Индикатор запроса передачи на COM1	R	+	+	+
M1025	ПЛК при получении неадекватного ответа на коммуникационный запрос от устройства (HMI, HPP или PC) установит M1025=ON и запишет код ошибки в регистр D1025	R	+	+	+
M1026	Флаг начала режима плавного разгона/замедления	R/W	-	+	+
M1027	Флаг PR выхода	R/W	-	+	+
M1028	Флаг режима работы таймеров T64...T126 M1028=ON – дискретность 10мс M1028=OFF - дискретность 100мс	R/W	+	-	-
M1029	ES/EX/SS/SA: Команда PLSY или PLSR для импульсного выхода Y0 полностью выполнена. EN: Первая импульсная группа CH0 (Y0, Y1) полностью выполнена.	R	+	+	+
M1030	ES/EX/SS/SA: Команда PLSY или PLSR для импульсного выхода Y1 полностью выполнена. EN: Вторая импульсная группа CH1 (Y2, Y3) полностью выполнена.	R	+	+	+
M1031	Очистка всей энергонезависимой памяти данных	R/W	+	+	+
M1032	Очистка всей энергонезависимой памяти данных	R/W	+	+	+
M1033	Сохранение состояния выходов в режиме STOP	R/W	+	+	+
M1034	Блокировка всех выходов Y	R/W	+	+	+
M1035	Вход X0 – X17 будет переключателем RUN/STOP (номер входа записан в D1035). В SA/SX можно использовать только X7	R/W *	-	+	+
M1039	Режим постоянного сканирования	R/W	+	+	+
M1040	Запрещение режима STL	R/W	+	+	+
M1041	Старт режима STL	R/W	+	+	+
M1042	Импульсный старт STL	R/W	+	+	+
M1043	Возвращение в нулевую точку завершено	R/W	+	+	+
M1044	Нахождение в нулевой точке	R/W	+	+	+
M1045	Запрещение сброса всех выходов	R/W	+	+	+
M1046	Режим STL выполняется	R	+	+	+
M1047	Разрешение режима STL	R/W	+	+	+
M1048	Флаг состояния аварийной точки	R	-	+	+
M1049	Управление флагом аварийной точки	R/W	-	+	+
M1050	Запрет прерывания I001	R/W	+	+	-
M1051	Запрет прерывания I101	R/W	+	+	-
M1052	Запрет прерывания I201	R/W	+	+	-

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1053	Запрет прерывания I301	R/W	+	+	-
M1054	Запрет прерывания I401	R/W	-	+	-
M1055	Запрет прерывания I501	R/W	-	+	-
M1056	Запрет прерывания I6□□	R/W	-	+	-
M1057	Запрет прерывания I7□□	R/W	-	+	-
M1059	Запрет прерываний I010 – I060	R/W	-	+	-
M1060	Системная ошибка (сообщение 1)	R	+	+	+
M1061	Системная ошибка (сообщение 2)	R	+	+	+
M1062	Системная ошибка (сообщение 3)	R	+	+	+
M1063	Системная ошибка (сообщение 4)	R	+	+	+
M1064	Ошибка оператора	R	+	+	+
M1065	Синтаксическая ошибка	R	+	+	+
M1066	Ошибка программы	R	+	+	+
M1067	Ошибка алгоритма программы	R	+	+	+
M1068	Фиксация ошибки алгоритма в регистре D1068	R	+	+	+
M1070	ES/EX/SS/SA: Команда PWM для выхода Y0. M1070=OFF: дискретность задания = 1 мс M1070=ON: дискретность задания = 100 мкс. ЕН: Команда PWM для выхода Y0. M1070=OFF: дискретность задания = 1 мс M1070=ON: дискретность задания = 100 мкс.	R/W	+	+	+
M1071	ЕН: Команда PWM для выхода Y2. M1070=OFF: дискретность задания = 1 мс M1070=ON: дискретность задания = 100 мкс.	R/W	-	-	+
M1072	Команда RUN (запуск ПЛК)	R/W	+	+	+
M1073	Используется системой.	-	-	-	-
M1074	Используется системой.	-	-	-	-
M1075	Ошибка записи во Flash-память	R	-	-	+
M1076	Ошибка календаря (часов реального времени)	R	-	+	+
M1077	Батарея разряжена	R	-	-	+
M1078	Остановка выполнения команды PLSY для Y0	R/W	+	+	-
M1079	Остановка выполнения команды PLSY для Y1	R/W	+	+	-
M1080	Индикатор запроса передачи на COM2	R	+	+	+
M1081	Флаг изменения направления преобразования инструкцией FLT	R/W	-	+	+
M1082	Флаг изменения значений в часах реального времени	R	-	+	+
M1083	Разрешение/запрещение выполнения прерываний в режиме FROM/TO	R/W	-	+	+
M1086	Установка переключателя ON/OFF от функции пароля для карты DVP-PCC01	R/W	+	+	+
M1087	Флаг работы сигнала LV	R/W	-	-	+
M1088	Флаг операции сравнения матриц. M1070=ON: матрицы одинаковые M1070=OFF: матрицы различные	R/W	-	-	+
M1089	Флаг сравнения матриц. Сравнение по первому биту.	R	-	-	+
M1090	Флаг сравнения матриц. Сравнение по последнему биту.	R	-	-	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1091	Флаг завершения сравнения матриц. Сравнение прекращается и выставляется M1091 = ON.	R	-	-	+
M1092	Флаг ошибки указателя матрицы. Когда указатель превышает зад. диапазон M1092 = ON.	R	-	-	+
M1093	Флаг увеличения указателя матрицы на 1.	R/W	-	-	+
M1094	Флаг обнуления указателя матрицы	R/W	-	-	+
M1095	Флаг переноса при операциях сдвига/вращения матрицы	R	-	-	+
M1096	Флаг дополнения при операции сдвига матрицы	R/W	-	-	+
M1097	Флаг направления при операциях сдвига/вращения матрицы	R/W	-	-	+
M1098	Флаг счетчика битов в матрице (счет единичных или нулевых битов)	R/W	-	-	+
M1099	M1091=ON, если результат счетчика битов = 0	R/W	-	-	+
M1100	Флаг величины выборки при выполнении команды SPD (вычисление скорости)	R/W	-	-	+
M1101	Разрешен ли старт файлового регистра или нет	R/W *	-	+	+
M1102	Используется системой	-	-	-	-
M1103	Используется системой	-	-	-	-
M1104	Состояние микропереключателя SW1 (функциональная карта DVP-F8ID)	R	-	-	+
M1105	Состояние микропереключателя SW2 (функциональная карта DVP-F8ID)	R	-	-	+
M1106	Состояние микропереключателя SW3 (функциональная карта DVP-F8ID)	R	-	-	+
M1107	Состояние микропереключателя SW4 (функциональная карта DVP-F8ID)	R	-	-	+
M1108	Состояние микропереключателя SW5 (функциональная карта DVP-F8ID)	R	-	-	+
M1109	Состояние микропереключателя SW6 (функциональная карта DVP-F8ID)	R	-	-	+
M1110	Состояние микропереключателя SW7 (функциональная карта DVP-F8ID)	R	-	-	+
M1111	Состояние микропереключателя SW8 (функциональная карта DVP-F8ID)	R	-	-	+
M1112	Состояние транзисторного выхода TR1 (функциональная карта DVP-F2OT)	R	-	+	+
M1113	Состояние транзисторного выхода TR2 (функциональная карта DVP-F2OT)	R	-	+	+
M1115	Старт функции разгона/замедления на импульсном выходе	R/W	+	+	-
M1116	Флаг разгона на импульсном выходе	R/W	+	+	-
M1117	Флаг достижения заданной частоты	R/W	+	+	-
M1118	Флаг замедления на импульсном выходе	R/W	+	+	-
M1119	Флаг выполнения функции	R/W	+	+	-
M1120	Фиксация коммуникационного протокола для COM2 (RS485).	R/W	+	+	+
M1121	Готовность передачи по RS-485	R	+	+	+
M1122	Требование передачи данных по RS-485	R/W	+	+	+
M1123	Прием данных по RS-485 завершен	R/W	+	+	+
M1124	Ожидание приема данных по RS-485	R/W	+	+	+
M1125	Сброс коммуникации	R/W	+	+	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1126	Выбор STX/CTX (пользователь/система)	R/W	+	+	+
M1127	Прием данных по RS-485 завершен (от команд MODRD, RDST, MODRW)	R/W	+	+	+
M1128	Индикация приема/передачи по RS-485	R/W	+	+	+
M1129	Время приема по RS-485 вышло	R/W	+	+	+
M1130	Выбор STX/CTX	R/W	+	+	+
M1131	M1131=ON, когда данные конверт. в HEX (от команд MODRD, RDST, MODRW)	R	+	+	+
M1132	M1132=ON, когда коммуникационная команда в ПЛК не релевантная.	R	+	+	+
M1133	Переключатель специального высокоскоростного (50 кГц) выхода (M1133 = ON - старт)	R/W	-	+	-
M1134	Когда M1134=ON импульсы идут непрерывно.	R/W	-	+	-
M1135	Флаг достижения заданного количества выходных импульсов	R/W	-	+	-
M1136	Фиксация коммуникационного протокола для COM3	R/W	-	-	+
M1138	Фиксация коммуникационного протокола для COM1 (RS232). Изменения в D1036 не выполняются после установки M1138.	R/W	+	+	+
M1139	Выбор режима ASCII/RTU для COM1 (RS-232): M1139 = ON – режим RTU M1139 = OFF – режим ASCII	R/W	+	+	+
M1140	Ошибка приема данных по RS-485 (от команд MODRD, RDST, MODRW)	R	+	+	+
M1141	Ошибка команд MODRD, RDST, MODRW	R	+	+	+
M1142	Ошибка приема данных по RS-485 (от команды VFD-A)	R	+	+	+
M1143	Выбор режима ASCII/RTU для COM2 (RS-485): M1143 = ON – режим RTU M1143 = OFF – режим ASCII	R/W	+	+	+
M1144	Старт выдачи импульсов на импульсном выходе Y0 с функцией разгона/замедления	R/W	-	+	-
M1145	Флаг разгона для импульсного выхода Y0 с корректировкой наклона	R	-	+	-
M1146	Флаг достижения заданной частоты для импульсного выхода Y0 с корректировкой наклона	R	-	+	-
M1147	Флаг замедления для импульсного выхода Y0 с корректировкой наклона	R	-	+	-
M1148	Флаг завершения функции разгона/замедления для импульсного выхода Y0 с корректировкой наклона	R/W	-	+	-
M1149	Флаг временной остановки функции разгона/замедления для импульсного выхода Y0 с корректировкой наклона	R/W	-	+	-
M1150	Операция группового сравнения DHSZ	R/W	-	-	+
M1151	Завершение операции группового сравнения DHSZ	R	-	-	+
M1152	Операция группового сравнения DHSZ в режиме контроля частоты	R/W	-	-	+
M1153	Завершение выполнения режима контроля частоты	R	-	-	+
M1154	Флаг начала замедления для функции разгона/замедления с заданным наклоном	R/W	-	+	-
M1161	Режим 8/16 бит (M1161 = ON – 8 бит)	R/W	+	+	+
M1167	Режим 16 бит для входа НКУ	R/W	-	+	+
M1168	Режим индикации работы функции SMOV	R/W	-	+	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1169	Выбор режима PWD	R/W	-	-	+
M1170	Старт функции пошагового выполнения программы	R/W	-	-	+
M1171	Пуск выполнения одного шага программы	R/W	-	-	+
M1172	Переключатель 2-х фазного импульсного выхода (M1172=ON – старт)	R/W	-	+	-
M1173	Когда M1173=ON импульсы идут непрерывно.	R/W	-	+	-
M1174	Флаг достижения заданного количества выходных импульсов	R/W	-	+	-
M1178	Старт VR00	R/W	-	+	+
M1179	Старт VR01	R/W	-	+	+
M1184	Запуск модема (функции MODEM)	R/W	-	-	+
M1185	Запуск инициализации модема	R/W	-	-	+
M1186	Ошибка инициализации модема	R/W	-	-	+
M1187	Инициализация модема выполнена	R/W	-	-	+
M1188	Индикация состояния соединения модема	R/W	-	-	+
M1196	Выбор формата отображения данных на дисплее DVP-SX: M1196 = ON – шестнадцатеричный; M1196 = OFF – десятичный	R/W	-	+	-
M1197	Установка десятичной точки на дисплее DVP-SX: две цифры после запятой	R/W	-	+	-
M1198	Установка десятичной точки на дисплее DVP-SX: одна цифра после запятой	R/W	-	+	-
M1199	Используется системой	-	-	-	-
M1200	C200 режим счета (ON: декрементирование)	R/W	-	+	+
M1201	C201 режим счета (ON: декрементирование)	R/W	-	+	+
M1202	C202 режим счета (ON: декрементирование)	R/W	-	+	+
M1203	C203 режим счета (ON: декрементирование)	R/W	-	+	+
M1204	C204 режим счета (ON: декрементирование)	R/W	-	+	+
M1205	C205 режим счета (ON: декрементирование)	R/W	-	+	+
M1206	C206 режим счета (ON: декрементирование)	R/W	-	+	+
M1207	C207 режим счета (ON: декрементирование)	R/W	-	+	+
M1208	C208 режим счета (ON: декрементирование)	R/W	-	+	+
M1209	C209 режим счета (ON: декрементирование)	R/W	-	+	+
M1210	C210 режим счета (ON: декрементирование)	R/W	-	+	+
M1211	C211 режим счета (ON: декрементирование)	R/W	-	+	+
M1212	C212 режим счета (ON: декрементирование)	R/W	-	+	+
M1213	C213 режим счета (ON: декрементирование)	R/W	-	+	+
M1214	C214 режим счета (ON: декрементирование)	R/W	-	+	+
M1215	C215 режим счета (ON: декрементирование)	R/W	-	+	+
M1216	C216 режим счета (ON: декрементирование)	R/W	-	+	+
M1217	C217 режим счета (ON: декрементирование)	R/W	-	+	+
M1218	C218 режим счета (ON: декрементирование)	R/W	-	+	+
M1219	C219 режим счета (ON: декрементирование)	R/W	-	+	+
M1220	C220 режим счета (ON: декрементирование)	R/W	-	+	+
M1221	C221 режим счета (ON: декрементирование)	R/W	-	+	+
M1222	C222 режим счета (ON: декрементирование)	R/W	-	+	+
M1223	C223 режим счета (ON: декрементирование)	R/W	-	+	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1224	C224 режим счета (ON: декрементирование)	R/W	-	+	+
M1225	C225 режим счета (ON: декрементирование)	R/W	-	+	+
M1226	C226 режим счета (ON: декрементирование)	R/W	-	+	+
M1227	C227 режим счета (ON: декрементирование)	R/W	-	+	+
M1228	C228 режим счета (ON: декрементирование)	R/W	-	+	+
M1229	C229 режим счета (ON: декрементирование)	R/W	-	+	+
M1230	C230 режим счета (ON: декрементирование)	R/W	-	+	+
M1231	C231 режим счета (ON: декрементирование)	R/W	-	+	+
M1232	C232 режим счета (ON: декрементирование)	R/W	-	+	+
M1233	C233 режим счета (ON: декрементирование)	R/W	-	+	+
M1234	C234 режим счета (ON: декрементирование)	R/W	-	+	+
M1235	C235 режим счета (ON: декрементирование)	R/W	+	+	+
M1236	C236 режим счета (ON: декрементирование)	R/W	+	+	+
M1237	C237 режим счета (ON: декрементирование)	R/W	+	+	+
M1238	C238 режим счета (ON: декрементирование)	R/W	+	+	+
M1239	C239 режим счета (ON: декрементирование)	R/W	-	+	+
M1240	C240 режим счета (ON: декрементирование)	R/W	-	+	+
M1241	C241 режим счета (ON: декрементирование)	R/W	+	+	+
M1242	C242 режим счета (ON: декрементирование)	R/W	+	+	+
M1243	C243 режим счета (ON: декрементирование)	R/W	-	-	+
M1244	C244 режим счета (ON: декрементирование)	R/W	+	+	+
M1245	C245 режим счета (ON: декрементирование)	R/W	-	+	-
M1246	C246 режим счета (ON: декрементирование)	R	+	+	+
M1247	C247 режим счета (ON: декрементирование)	R	+	+	+
M1248	C248 режим счета (ON: декрементирование)	R	-	-	+
M1249	C249 режим счета (ON: декрементирование)	R	+	+	+
M1251	C251 режим счета (ON: декрементирование)	R	+	+	+
M1252	C252 режим счета (ON: декрементирование)	R	+	+	+
M1253	C253 режим счета (ON: декрементирование)	R	-	-	+
M1254	C254 режим счета (ON: декрементирование)	R	+	+	+
M1256	Флаг ошибки EF (внешнее отключение)	R	-	-	+
M1258	Реверсирование импульсного сигнала на Y0 для PWM-инструкции	R/W	-	-	+
M1259	Реверсирование импульсного сигнала на Y2 для PWM-инструкции	R/W	-	-	+
M1260	Пусть на X5 будет сброс всех высокоскоростных счетчиков	R/W	-	+	-
M1261	Флаг команды сравнения DHSCR	R/W	-	-	+
M1264	Запрещение внешнего старта счетчика HHSC0	R/W	-	-	+
M1265	Запрещение внешнего сброса счетчика HHSC0	R/W	-	-	+
M1266	Запрещение внешнего старта счетчика HHSC1	R/W	-	-	+
M1267	Запрещение внешнего сброса счетчика HHSC1	R/W	-	-	+
M1268	Запрещение внешнего старта счетчика HHSC2	R/W	-	-	+
M1269	Запрещение внешнего сброса счетчика HHSC2	R/W	-	-	+
M1270	Запрещение внешнего старта счетчика HHSC3	R/W	-	-	+
M1271	Запрещение внешнего сброса счетчика HHSC3	R/W	-	-	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1272	Сброс контроля HHSC0	R/W	-	-	+
M1273	Старт контроля HHSC0	R/W	-	-	+
M1274	Сброс контроля HHSC1	R/W	-	-	+
M1275	Старт контроля HHSC1	R/W	-	-	+
M1276	Сброс контроля HHSC2	R/W	-	-	+
M1277	Старт контроля HHSC2	R/W	-	-	+
M1278	Сброс контроля HHSC3	R/W	-	-	+
M1279	Старт контроля HHSC3	R/W	-	-	+
M1280	I00□ флаг запрета прерывания	R/W	-	-	+
M1281	I10□ флаг запрета прерывания	R/W	-	-	+
M1282	I20□ флаг запрета прерывания	R/W	-	-	+
M1283	I30□ флаг запрета прерывания	R/W	-	-	+
M1284	I40□ флаг запрета прерывания	R/W	-	-	+
M1285	I50□ флаг запрета прерывания	R/W	-	-	+
M1286	I6□□ флаг запрета прерывания	R/W	-	-	+
M1287	I7□□ флаг запрета прерывания	R/W	-	-	+
M1288	I8□□ флаг запрета прерывания	R/W	-	-	+
M1289	I010 флаг запрета прерывания	R/W	-	-	+
M1290	I020 флаг запрета прерывания	R/W	-	-	+
M1291	I030 флаг запрета прерывания	R/W	-	-	+
M1292	I040 флаг запрета прерывания	R/W	-	-	+
M1293	I050 флаг запрета прерывания	R/W	-	-	+
M1294	I060 флаг запрета прерывания	R/W	-	-	+
M1295	I110 флаг запрета прерывания	R/W	-	-	+
M1296	I120 флаг запрета прерывания	R/W	-	-	+
M1297	I130 флаг запрета прерывания	R/W	-	-	+
M1298	I140 флаг запрета прерывания	R/W	-	-	+
M1299	I150 флаг запрета прерывания	R/W	-	+	+
M1300	I160 флаг запрета прерывания	R/W	-	-	+
M1301	I170 флаг запрета прерывания	R/W	-	-	+
M1302	I180 флаг запрета прерывания	R/W	-	-	+
M1303	Флаг переключения младшего/старшего байтов от XCH-инструкции	R/W	-	+	+
M1304	Вход X может быть ON/OFF	R/W	-	+	+
M1305	PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA CH0(Y0-Y1) флаг реверсивной работы	R	-	-	+
M1306	PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA CH1(Y2-Y3) флаг реверсивной работы	R	-	-	+
M1312	C235 старт входа	R/W	-	-	+
M1313	C236 старт входа	R/W	-	-	+
M1314	C237 старт входа	R/W	-	-	+
M1315	C238 старт входа	R/W	-	-	+
M1316	C239 старт входа	R/W	-	-	+
M1317	C240 старт входа	R/W	-	-	+
M1320	C235 сброс входа	R/W	-	-	+
M1321	C236 сброс входа	R/W	-	-	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1322	C237 сброс входа	R/W	-	-	+
M1323	C238 сброс входа	R/W	-	-	+
M1324	C239 сброс входа	R/W	-	-	+
M1325	C240 сброс входа	R/W	-	-	+
M1328	C235 разрешение функции старт/сброс	R/W	-	-	+
M1329	C236 разрешение функции старт/сброс	R/W	-	-	+
M1330	C237 разрешение функции старт/сброс	R/W	-	-	+
M1331	C238 разрешение функции старт/сброс	R/W	-	-	+
M1332	C239 разрешение функции старт/сброс	R/W	-	-	+
M1333	C240 разрешение функции старт/сброс	R/W	-	-	+
M1334	Стоп импульсного выхода CH0 (Y0, Y1)	R/W	-	-	+
M1335	Стоп импульсного выхода CH1 (Y2, Y3)	R/W	-	-	+
M1336	Флаг передачи импульсов CH0 (Y0, Y1)	R	-	-	+
M1337	Флаг передачи импульсов CH1 (Y2, Y3)	R	-	-	+
M1338	Старт импульсного выхода CH0 (Y0, Y1) флаг начала	R/W	-	-	+
M1339	Старт импульсного выхода CH1 (Y2, Y3) флаг начала	R/W	-	-	+
M1340	Иметь прерывание после завершения передачи по CH0 (Y0, Y1)	R/W	-	-	+
M1341	Иметь прерывание после завершения передачи по CH1 (Y2, Y3)	R/W	-	-	+
M1342	Иметь прерывание при одновременной передаче по CH0 (Y0, Y1)	R/W	-	-	+
M1343	Иметь прерывание при одновременной передаче по CH1 (Y2, Y3)	R/W	-	-	+
M1344	Старт CH0 (Y0, Y1) флаг выравнивания	R/W	-	-	+
M1345	Старт CH1 (Y2, Y3) флаг выравнивания	R/W	-	-	+
M1346	Флаг разрешения очистки вых. сигнала от команды ZRN	R/W	-	-	+
M1347	Авто-сброс флага для первой импульсной группы	R/W	-	-	+
M1348	Авто-сброс флага для второй импульсной группы	R/W	-	-	+
M1350	Флаг начала соединения ПЛК по RS-485(PLC LINK)	R/W	-	+	+
M1351	Запуск PLC LINK в автоматическом режиме	R/W	-	+	+
M1352	Запуск PLC LINK в ручном режиме	R/W	-	+	+
M1353	Запуск связи ПЛК с более, чем 32-мя внешними устройствами или более, чем 16-тью функциями чтения/записи (100 записей макс.)	R/W	-	-	+
M1354	Запуск PLC LINK с функцией синхронного чтения/записи	R/W	-	+	+
M1360	PLC LINK ID1 есть в наличии	R	-	+	+
M1361	PLC LINK ID2 есть в наличии	R	-	+	+
M1362	PLC LINK ID3 есть в наличии	R	-	+	+
M1363	PLC LINK ID4 есть в наличии	R	-	+	+
M1364	PLC LINK ID5 есть в наличии	R	-	+	+
M1365	PLC LINK ID6 есть в наличии	R	-	+	+
M1366	PLC LINK ID7 есть в наличии	R	-	+	+
M1367	PLC LINK ID8 есть в наличии	R	-	+	+
M1368	PLC LINK ID9 есть в наличии	R	-	+	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1369	PLC LINK ID10 есть в наличии	R	-	+	+
M1370	PLC LINK ID11 есть в наличии	R	-	+	+
M1371	PLC LINK ID12 есть в наличии	R	-	+	+
M1372	PLC LINK ID13 есть в наличии	R	-	+	+
M1373	PLC LINK ID14 есть в наличии	R	-	+	+
M1374	PLC LINK ID15 есть в наличии	R	-	+	+
M1375	PLC LINK ID16 есть в наличии	R	-	+	+
M1376	PLC LINK ID1 в действии	R	-	+	+
M1377	PLC LINK ID2 в действии	R	-	+	+
M1378	PLC LINK ID3 в действии	R	-	+	+
M1379	PLC LINK ID4 в действии	R	-	+	+
M1380	PLC LINK ID5 в действии	R	-	+	+
M1381	PLC LINK ID6 в действии	R	-	+	+
M1382	PLC LINK ID7 в действии	R	-	+	+
M1383	PLC LINK ID8 в действии	R	-	+	+
M1384	PLC LINK ID9 в действии	R	-	+	+
M1385	PLC LINK ID10 в действии	R	-	+	+
M1386	PLC LINK ID11 в действии	R	-	+	+
M1387	PLC LINK ID12 в действии	R	-	+	+
M1388	PLC LINK ID13 в действии	R	-	+	+
M1389	PLC LINK ID14 в действии	R	-	+	+
M1390	PLC LINK ID15 в действии	R	-	+	+
M1391	PLC LINK ID16 в действии	R	-	+	+
M1392	PLC LINK ID1 ошибка	R	-	+	+
M1393	PLC LINK ID2 ошибка	R	-	+	+
M1394	PLC LINK ID3 ошибка	R	-	+	+
M1395	PLC LINK ID4 ошибка	R	-	+	+
M1396	PLC LINK ID5 ошибка	R	-	+	+
M1397	PLC LINK ID6 ошибка	R	-	+	+
M1398	PLC LINK ID7 ошибка	R	-	+	+
M1399	PLC LINK ID8 ошибка	R	-	+	+
M1400	PLC LINK ID9 ошибка	R	-	+	+
M1401	PLC LINK ID10 ошибка	R	-	+	+
M1402	PLC LINK ID11 ошибка	R	-	+	+
M1403	PLC LINK ID12 ошибка	R	-	+	+
M1404	PLC LINK ID13 ошибка	R	-	+	+
M1405	PLC LINK ID14 ошибка	R	-	+	+
M1406	PLC LINK ID15 ошибка	R	-	+	+
M1407	PLC LINK ID16 ошибка	R	-	+	+
M1408	PLC LINK ID1 чтение завершено	R	-	+	+
M1409	PLC LINK ID2 чтение завершено	R	-	+	+
M1410	PLC LINK ID3 чтение завершено	R	-	+	+
M1411	PLC LINK ID4 чтение завершено	R	-	+	+
M1412	PLC LINK ID5 чтение завершено	R	-	+	+
M1413	PLC LINK ID6 чтение завершено	R	-	+	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1414	PLC LINK ID7 чтение завершено	R	-	+	+
M1415	PLC LINK ID8 чтение завершено	R	-	+	+
M1416	PLC LINK ID9 чтение завершено	R	-	+	+
M1417	PLC LINK ID10 чтение завершено	R	-	+	+
M1418	PLC LINK ID11 чтение завершено	R	-	+	+
M1419	PLC LINK ID12 чтение завершено	R	-	+	+
M1420	PLC LINK ID13 чтение завершено	R	-	+	+
M1421	PLC LINK ID14 чтение завершено	R	-	+	+
M1422	PLC LINK ID15 чтение завершено	R	-	+	+
M1423	PLC LINK ID16 чтение завершено	R	-	+	+
M1424	PLC LINK ID1 запись завершена	R	-	+	+
M1425	PLC LINK ID2 запись завершена	R	-	+	+
M1426	PLC LINK ID3 запись завершена	R	-	+	+
M1427	PLC LINK ID4 запись завершена	R	-	+	+
M1428	PLC LINK ID5 запись завершена	R	-	+	+
M1429	PLC LINK ID6 запись завершена	R	-	+	+
M1430	PLC LINK ID7 запись завершена	R	-	+	+
M1431	PLC LINK ID8 запись завершена	R	-	+	+
M1432	PLC LINK ID9 запись завершена	R	-	+	+
M1433	PLC LINK ID10 запись завершена	R	-	+	+
M1434	PLC LINK ID11 запись завершена	R	-	+	+
M1435	PLC LINK ID12 запись завершена	R	-	+	+
M1436	PLC LINK ID13 запись завершена	R	-	+	+
M1437	PLC LINK ID14 запись завершена	R	-	+	+
M1438	PLC LINK ID15 запись завершена	R	-	+	+
M1439	PLC LINK ID16 запись завершена	R	-	+	+
M1440	PLC LINK ID17 есть в наличии	R	-	-	+
M1441	PLC LINK ID18 есть в наличии	R	-	-	+
M1442	PLC LINK ID19 есть в наличии	R	-	-	+
M1443	PLC LINK ID20 есть в наличии	R	-	-	+
M1444	PLC LINK ID21 есть в наличии	R	-	-	+
M1445	PLC LINK ID22 есть в наличии	R	-	-	+
M1446	PLC LINK ID23 есть в наличии	R	-	-	+
M1447	PLC LINK ID24 есть в наличии	R	-	-	+
M1448	PLC LINK ID25 есть в наличии	R	-	-	+
M1449	PLC LINK ID26 есть в наличии	R	-	-	+
M1450	PLC LINK ID27 есть в наличии	R	-	-	+
M1451	PLC LINK ID28 есть в наличии	R	-	-	+
M1452	PLC LINK ID29 есть в наличии	R	-	-	+
M1453	PLC LINK ID30 есть в наличии	R	-	-	+
M1454	PLC LINK ID31 есть в наличии	R	-	-	+
M1455	PLC LINK ID32 есть в наличии	R	-	-	+
M1456	PLC LINK ID17 в действии	R	-	-	+
M1457	PLC LINK ID18 в действии	R	-	-	+
M1458	PLC LINK ID19 в действии	R	-	-	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EN
M1459	PLC LINK ID20 в действии	R	-	-	+
M1460	PLC LINK ID21 в действии	R	-	-	+
M1461	PLC LINK ID22 в действии	R	-	-	+
M1462	PLC LINK ID23 в действии	R	-	-	+
M1463	PLC LINK ID24 в действии	R	-	-	+
M1464	PLC LINK ID25 в действии	R	-	-	+
M1465	PLC LINK ID26 в действии	R	-	-	+
M1466	PLC LINK ID27 в действии	R	-	-	+
M1467	PLC LINK ID28 в действии	R	-	-	+
M1468	PLC LINK ID29 в действии	R	-	-	+
M1469	PLC LINK ID30 в действии	R	-	-	+
M1470	PLC LINK ID31 в действии	R	-	-	+
M1471	PLC LINK ID32 в действии	R	-	-	+
M1472	PLC LINK ID17 ошибка	R	-	-	+
M1473	PLC LINK ID18 ошибка	R	-	-	+
M1474	PLC LINK ID19 ошибка	R	-	-	+
M1475	PLC LINK ID20 ошибка	R	-	-	+
M1476	PLC LINK ID21 ошибка	R	-	-	+
M1477	PLC LINK ID22 ошибка	R	-	-	+
M1478	PLC LINK ID23 ошибка	R	-	-	+
M1479	PLC LINK ID24 ошибка	R	-	-	+
M1480	PLC LINK ID25 ошибка	R	-	-	+
M1481	PLC LINK ID26 ошибка	R	-	-	+
M1482	PLC LINK ID27 ошибка	R	-	-	+
M1483	PLC LINK ID28 ошибка	R	-	-	+
M1484	PLC LINK ID29 ошибка	R	-	-	+
M1485	PLC LINK ID30 ошибка	R	-	-	+
M1486	PLC LINK ID31 ошибка	R	-	-	+
M1487	PLC LINK ID32 ошибка	R	-	-	+
M1488	PLC LINK ID17 чтение завершено	R	-	-	+
M1489	PLC LINK ID18 чтение завершено	R	-	-	+
M1490	PLC LINK ID19 чтение завершено	R	-	-	+
M1491	PLC LINK ID20 чтение завершено	R	-	-	+
M1492	PLC LINK ID21 чтение завершено	R	-	-	+
M1493	PLC LINK ID22 чтение завершено	R	-	-	+
M1494	PLC LINK ID23 чтение завершено	R	-	-	+
M1495	PLC LINK ID24 чтение завершено	R	-	-	+
M1496	PLC LINK ID25 чтение завершено	R	-	-	+
M1497	PLC LINK ID26 чтение завершено	R	-	-	+
M1498	PLC LINK ID27 чтение завершено	R	-	-	+
M1499	PLC LINK ID28 чтение завершено	R	-	-	+
M1500	PLC LINK ID29 чтение завершено	R	-	-	+
M1501	PLC LINK ID30 чтение завершено	R	-	-	+
M1502	PLC LINK ID31 чтение завершено	R	-	-	+
M1503	PLC LINK ID32 чтение завершено	R	-	-	+

Номер	Функция	Атрибут	Серия ПЛК		
			ES/EX/SS	SA/SX	EH
M1504	PLC LINK ID17 запись завершена	R	-	-	+
M1505	PLC LINK ID18 запись завершена	R	-	-	+
M1506	PLC LINK ID19 запись завершена	R	-	-	+
M1507	PLC LINK ID20 запись завершена	R	-	-	+
M1508	PLC LINK ID21 запись завершена	R	-	-	+
M1509	PLC LINK ID22 запись завершена	R	-	-	+
M1510	PLC LINK ID23 запись завершена	R	-	-	+
M1511	PLC LINK ID24 запись завершена	R	-	-	+
M1512	PLC LINK ID25 запись завершена	R	-	-	+
M1513	PLC LINK ID26 запись завершена	R	-	-	+
M1514	PLC LINK ID27 запись завершена	R	-	-	+
M1515	PLC LINK ID28 запись завершена	R	-	-	+
M1516	PLC LINK ID29 запись завершена	R	-	-	+
M1517	PLC LINK ID30 запись завершена	R	-	-	+
M1518	PLC LINK ID31 запись завершена	R	-	-	+
M1519	PLC LINK ID32 запись завершена	R	-	-	+

2.10.2. Специальные регистры

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EH
D1000	Сторожевой таймер (WDT) - мсек	R/W	200	+	+	+
D1001	Модель DVP, версия ПО. D1001 = HXX27 – версия 2.7	R	#	+	+	+
D1002	Объем программы	R	#	+	+	+
D1003	Суммарная память программы	R	#	+	+	+
D1004	Количество найденных грамматических ошибок	R	0	+	+	+
D1005	Используется системой	-	-	-	-	-
D1008	Адрес шага, когда WDT = ON	R	0	+	+	+
D1009	Буфер истории ошибок LV (низкое напряжение)	R*	0	-	-	+
D1010	Текущее время сканирования (ед.=0.1 мс)	R	0	+	+	+
D1011	Минимальное время сканирования (ед.= 0.1 мс)	R	0	+	+	+
D1012	Максимальное время сканирования (ед.= 0.1 мс)	R	0	+	+	+
D1015	0...32767 (ед. 0.1 мс) текущее время высокоскоростного таймера	R/W	0	-	-	+
D1018	Число π (младший байт)	R/W	H'0F9B	-	+	+
D1019	Число π (старший байт)	R/W	H'4049	-	+	+
D1020	Входной фильтр (время задержки в мс) для входов: X0...X7 (ES/EX/SS/SA/SX), X0...X17 (EH)	R/W	10	+	+	+
D1021	Входной фильтр (время задержки в мс) для входов: X10...X17 (ES/EX/SS/SA/SX), X20...X377 (EH)	R/W	10	+	+	+
D1022	Удвоенная частота для двухфазного счетчика AB	R/W	0	+	+	-
D1024	Флаг, используемый системой	-	-	-	-	-
D1025	Код коммуникационной ошибки	R	0	+	+	+
D1028	Индексный регистр E0	R/W	0	+	+	+
D1029	Индексный регистр F0	R/W	0	+	+	+
D1030	Количество импульсов на выходе Y0 (младшее слово)	R	0	+	+	-
D1031	Количество импульсов на выходе Y0 (старшее слово)	R	0	+	+	-
D1032	Количество импульсов на выходе Y1 (младшее слово)	R	0	+	+	-
D1033	Количество импульсов на выходе Y1 (старшее слово)	R	0	+	+	-
D1034	Режим работы карты измерения частоты	R*	1	-	-	+
D1035	Номер входа X для переключателя RUN/STOP (см. M1035)	R/W*	0	-	-	+
D1036	Протокол коммуникации для COM1 (RS232)	R/W	H'86	+	+	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1037	Время сканирования для команды НКУ (мсек)	R/W *	500	-	-	+
D1038	Время задержки для ответа, когда ПЛК – ведомый в сети RS-485 (ед. = 0.1 мс)	R/W	0	+	+	-
D1039	Постоянное время сканирования (мсек)	R/W	0	+	+	+
D1040	Состояние ON номер 1 (для пошагового режима)	R	0	+	+	+
D1041	Состояние ON номер 2 (для пошагового режима)	R	0	+	+	+
D1042	Состояние ON номер 3 (для пошагового режима)	R	0	+	+	+
D1043	Состояние ON номер 4 (для пошагового режима)	R	0	+	+	+
D1044	Состояние ON номер 5 (для пошагового режима)	R	0	+	+	+
D1045	Состояние ON номер 6 (для пошагового режима)	R	0	+	+	+
D1046	Состояние ON номер 7 (для пошагового режима)	R	0	+	+	+
D1047	Состояние ON номер 8 (для пошагового режима)	R	0	+	+	+
D1049	Ошибка в пошаговом режиме	R	0	-	+	+
D1050 ... D1055	ПЛК будет автоматически конвертировать ASCII данные, сохраненные в D1070-D1085 в шестнадцатеричный формат (HEX)	R	0	+	+	+
D1056	Текущее значение на аналоговом входе (канал CH0) для DVP-EX/SX и функциональной карты DVP-EN	R	0	+	-	-
D1057	Текущее значение на аналоговом входе (канал CH1) для DVP-EX/SX и функциональной карты DVP-EN	R	0	+	-	-
D1058	Текущее значение на аналоговом входе (канал CH2) для DVP-EX	R	0	+	-	-
D1059	Текущее значение на аналоговом входе (канал CH3) для DVP-EX	R	0	+	-	-
D1061	Сообщение о системной ошибке	R*	0	-	-	+
D1065	Флаг, используемый системой	-	-	-	-	-
D1066	Флаг, используемый системой	-	-	-	-	-
D1067	Код ошибки алгоритма программы	R	0	+	+	+
D1068	Блокировка ошибки адреса	R	0	+	+	+
D1069	Номер ошибки ассоциированной с флагами D1065-D1067	R	0	+	+	+
D1070 ... D1085	Регистры для хранения данных принятых по RS-485	R	0	+	+	+
D1089 ... D1099	Регистры для хранения данных отправленных по RS-485	R	0	+	+	+
D1100	Однократное возникновение LV на Y0-Y17	R/W	0	-	-	+
D1101	Стартовый адрес файлового регистра	R/W *	0	-	+	+
D1102	Число копируемых файловых регистров	R/W *	1600	-	+	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EH
D1103	Стартовый номер для сохранения в файловый регистр (должен быть больше 2000)	R/W *	2000	-	+	+
D1104	Указатель регистра D для разгона/замедления импульсного выхода Y0	R/W	0	+	+	-
D1109	Протокол коммуникации для COM3	R/W	0	-	-	+
D1110	Усредненное значение на аналоговом входе (канал CH0) для DVP-EX/SX и функциональной карты DVP-EH	R	0	+	-	-
D1111	Усредненное значение на аналоговом входе (канал CH1) для DVP-EX/SX и функциональной карты DVP-EH	R	0	+	-	-
D1112	Усредненное значение на аналоговом входе (канал CH2) для DVP-EX	R	0	+	-	-
D1113	Усредненное значение на аналоговом входе (канал CH3) для DVP-EX	R	0	+	-	-
D1116	Значение на аналоговом выходе (канал CH0) для DVP-EX/SX и функциональной карты DVP-EH	R/W	0	+	-	-
D1117	Значение на аналоговом выходе (канал CH1) для DVP-EX/SX и функциональной карты DVP-EH	R/W	0	+	-	-
D1118	Фильтр для аналого-цифрового преобразователя DVP-EX/SX/EH (мсек)	R/W	5	+	-	-
D1119	Используется системой	-	-	-	-	-
D1120	Протокол коммуникации для COM2 (RS485) (H86 = 9600, 7, E, 1)	R/W	H'86	+	+	+
D1121	Коммуникационный адрес ПЛК	R/W *	1	+	+	+
D1122	Остаточные слова от переданных данных	R	0	+	+	+
D1123	Остаточные слова от принятых данных	R	0	+	+	+
D1124	Начало символьного определения (STX)	R/W	H'3A	+	+	+
D1125	Первое окончание символьного определения (EXT1)	R/W	H'0D	+	+	+
D1126	Второе окончание символьного определения (EXT2)	R/W	H'0A	+	+	+
D1129	Время ожидания ответа по RS-485 (мс)	R/W	0	+	+	+
D1130	Код ошибки отклика MODBUS	R	0	+	+	+
D1133	Индексный регистр для высокоскоростного импульсного выхода	R/W	0	-	+	-
D1137	Адрес оператора произошедшей ошибки	R	0	+	+	+
D1139	Число соединенных BCD модулей расширения (макс. 2 модуля)	R	0	-	-	+
D1140	Число соединенных специальных модулей расширения (макс. 8 модулей)	R	0	+	+	+
D1141	Используется системой	-	-	-	-	-
D1142	Точки ввода (X) от модулей расширения	R	0	+	+	+
D1143	Точки вывода (Y) модулей расширения	R	0	+	+	+
D1144	Указатель регистра D для крутизны наклона разгона/замедления импульсного выхода	R/W	0	-	+	-
D1145	Число соединенных KEY модулей расширения	R	0	-	-	+
D1146	Число соединенных DISP модулей расширения	R	0	-	-	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1147	Тип карты памяти: b0=0: нет флэш-карты (H0000) b0=1: есть флэш-карта b8=0: переключатель для флэш-карты - выкл (H0001) b8=1: переключатель для флэш-карты - вкл (H0101)	R	#	-	-	+
D1148	Используется системой	-	-	-	-	-
D1149	Тип функциональной карты: 0. Нет карты; 1: RS232; 2: TS01 3: RS422; 4: Потенциометр; 5: Цифровые переключатели; 6: Транзисторные выходы; 7: Высокоскор. имп. вых.; 8: 2AD; 9: 2DA; 10: карта измерения частоты	R	0	-	+	+
D1150	Регистр вычисления в режиме группового сравнения для DHSZ-инструкции	R	0	-	+	+
D1151	Регистр вычисления в режиме частотного управления для DHSZ-инструкции	R	0	-	+	+
D1152	Изменение значения старшего слова DHSZ	R	0	-	+	+
D1153	Изменение значения младшего слова DHSZ	R	0	-	+	+
D1154	Рекомендуемое значение назначенного интервала времени замедления (10...32767 мс) для импульсного выхода с заданием наклона разг./замедл.	R/W	200	-	+	-
D1155	Рекомендуемое значение назначенного интервала частоты разгона (-1...-32767 Гц) для импульсного выхода с заданием наклона разг./замедл.	R/W	-1000	-	+	-
D1156 ... D1165	Специальный регистр индикации команды RTMU (K0 – K9)	R/W	0	-	-	+
D1166	Используется системой	-	-	-	-	-
D1167	Используется системой	-	-	-	-	-
D1168	(I150) требование прерывания при приеме специфичных символов по RS-485	R/W	0	+	+	+
D1169	(I150) требование прерывания при приеме определенной длины по RS-485	R/W	0	-	-	+
D1170	Количество выполненных шагов программы	R	0	-	-	+
D1172	Частота двухфазного импульсного выхода (12 Гц – 20 кГц)	R/W	0	-	+	-
D1173	Выбор режима двухфазного импульсного выхода (K1 и K2)	R/W	0	-	+	-
D1174	Заданное значение для двухфазного импульсного выхода (младшее слово)	R/W	0	-	+	-
D1175	Заданное значение для двухфазного импульсного выхода (старшее слово)	R/W	0	-	+	-
D1176	Текущее значение на двухфазном импульсном выходе (младшее слово)	R/W	0	-	+	-

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1177	Текущее значение на двухфазном импульсном выходе (старшее слово)	R/W	0	-	+	-
D1178	Значение VR0	R	0	-	+	+
D1179	Значение VR1	R	0	-	+	+
D1182	Индексный регистр E1	R/W	0	-	+	+
D1183	Индексный регистр F1	R/W	0	-	+	+
D1184	Индексный регистр E2	R/W	0	-	+	+
D1185	Индексный регистр F2	R/W	0	-	+	+
D1186	Индексный регистр E3	R/W	0	-	+	+
D1187	Индексный регистр F3	R/W	0	-	+	+
D1188	Индексный регистр E4	R/W	0	-	-	+
D1189	Индексный регистр F4	R/W	0	-	-	+
D1190	Индексный регистр E5	R/W	0	-	-	+
D1191	Индексный регистр F5	R/W	0	-	-	+
D1192	Индексный регистр E6	R/W	0	-	-	+
D1193	Индексный регистр F6	R/W	0	-	-	+
D1194	Индексный регистр E7	R/W	0	-	-	+
D1195	Индексный регистр F7	R/W	0	-	-	+
D1196	Содержимое индикатора на DVP-SX	R/W	0	-	+	-
D1197	Используется системой	-	-	-	-	-
D1198	Используется системой	-	-	-	-	-
D1199	Используется системой	-	-	-	-	-
D1200	Начальный адрес энергонезависимых реле M0...M999	R/W *	#	-	+	+
D1201	Конечный адрес энергонезависимых реле M0...M999	R/W *	999	-	-	+
D1202	Начальный адрес энергонезависимых реле M2000...M4095	R/W *	2000	-	-	+
D1203	Конечный адрес энергонезависимых реле M2000...M4095	R/W *	4095	-	-	+
D1204	Начальный адрес энергонезависимых таймеров (100мс) T0...T199	R/W *	H'FFFF	-	-	+
D1205	Конечный адрес энергонезависимых таймеров (100мс) T0...T199	R/W *	H'FFFF	-	-	+
D1206	Начальный адрес энергонезависимых таймеров (10мс) T200...T239	R/W *	H'FFFF	-	-	+
D1207	Конечный адрес энергонезависимых таймеров (10мс) T200...T239	R/W *	H'FFFF	-	-	+
D1208	Начальный адрес энергонезависимых счетчиков (16 бит) C0...C199	R/W *	#	-	+	+
D1209	Конечный адрес энергонезависимых счетчиков (16 бит) C0...C199	R/W *	199	-	-	+
D1210	Начальный адрес энергонезависимых счетчиков (32 бит) C200...C234	R/W *	#	-	+	+
D1211	Конечный адрес энергонезависимых счетчиков (32 бит) C200...C234	R/W *	234	-	-	+
D1212	Начальный адрес энергонезависимых высокоскоростных счетчиков (32 бит) C235...C255	R/W *	235	-	-	+

Номер	Функция	Атрибут	Заводск. уставка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1213	Конечный адрес энергонезависимых высокоскоростных счетчиков (32 бит) C235...C255	R/W *	255	-	-	+
D1214	Начальный адрес энергонезависимых шаговых реле S0...S1023	R/W *	#	-	+	+
D1215	Конечный адрес энергонезависимых шаговых реле S0...S1023	R/W *	#	-	+	+
D1216	Начальный адрес энергонезависимых регистров D0...D999	R/W *	200	-	-	+
D1217	Конечный адрес энергонезависимых регистров D0...D999	R/W *	999	-	-	+
D1218	Начальный адрес энергонезависимых регистров D2000...D9999	R/W *	2000	-	-	+
D1219	Конечный адрес энергонезависимых регистров D2000...D9999	R/W *	#	-	+	+
D1220	Первая группа импульсных выходов: 00: 1 фазн. (Y0); 01: фаза A; 02: фаза B	R/W	0	-	-	+
D1221	Вторая группа импульсных выходов: 00: 1 фазн. (Y2); 01: фаза A; 02: фаза B	R/W	0	-	-	+
D1222	Задание разницы времени для CH0 между сигналом направления и импульсным выходом для DRVI, DDRVI, DRVA, DDRVA, PLSV и DPLSV.	R/W	0	-	-	+
D1223	Задание разницы времени для CH1 между сигналом направления и импульсным выходом для DRVI, DDRVI, DRVA, DDRVA, PLSV и DPLSV.	R/W	0	-	-	+
D1225	Первая группа вычисления уставки счетчиков (HHSC0). Счетный регистры: C241, C246, C251	R/W	0	-	-	+
D1226	Вторая группа вычисления уставки счетчиков (HHSC1). Счетный регистры: C242, C247, C252	R/W	0	-	-	+
D1227	Третья группа вычисления уставки счетчиков (HHSC2). Счетный регистры: C243, C248, C253	R/W	0	-	-	+
D1228	Четвертая группа вычисления уставки счетчиков (HHSC2). Счетный регистры: C244, C249, C254	R/W	0	-	-	+
D1256 ... D1295	Регистры для хранения данных принятых/переданных по RS-485 командой MODRW	R	0	+	+	+
D1296 ... D1311	ПЛК буде автоматически конвертировать ASCII данные, сохраненные в D1256-D1295 в шестнадцатеричный формат	R	0	+	+	+
D1313	Часы реального времени (RTC): отображение и коррекция секунд (00...59)	R/W	0	-	+	+
D1314	Часы реального времени (RTC): отображение и коррекция минут (00...59)	R/W	0	-	+	+
D1315	Часы реального времени (RTC): отображение и коррекция часов (00...23)	R/W	0	-	+	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1316	Часы реального времени (RTC): отображение и коррекция дня (01...31)	R/W	1	-	+	+
D1317	Часы реального времени (RTC): отображение и коррекция месяца (01...12)	R/W	1	-	+	+
D1318	Часы реального времени (RTC): отображение и коррекция дня недели (1...7)	R/W	6	-	+	+
D1319	Часы реального времени (RTC): отображение и коррекция года (00...99)	R/W	0	-	+	+
D1320	Адрес (ID) 1-го специального модуля расширения	R	0	-	-	+
D1321	Адрес (ID) 2-го специального модуля расширения	R	0	-	-	+
D1322	Адрес (ID) 3-го специального модуля расширения	R	0	-	-	+
D1323	Адрес (ID) 4-го специального модуля расширения	R	0	-	-	+
D1324	Адрес (ID) 5-го специального модуля расширения	R	0	-	-	+
D1325	Адрес (ID) 6-го специального модуля расширения	R	0	-	-	+
D1326	Адрес (ID) 7-го специального модуля расширения	R	0	-	-	+
D1327	Адрес (ID) 8-го специального модуля расширения	R	0	-	-	+
D1328	CH0 (Y0, Y1) смещение импульсов (младшее слово)	R/W	0	-	-	+
D1329	CH0 (Y0, Y1) смещение импульсов (старшее слово)	R/W	0	-	-	+
D1330	CH1 (Y2, Y3) смещение импульсов (младшее слово)	R/W	0	-	-	+
D1331	CH1 (Y2, Y3) смещение импульсов (старшее слово)	R/W	0	-	-	+
D1332	CH0 (Y0, Y1) остаток импульсов (младшее слово)	R	0	-	-	+
D1333	CH0 (Y0, Y1) остаток импульсов (старшее слово)	R	0	-	-	+
D1334	CH1 (Y2, Y3) остаток импульсов (младшее слово)	R	0	-	-	+
D1335	CH1 (Y2, Y3) остаток импульсов (старшее слово)	R	0	-	-	+
D1336	CH0 (Y0, Y1) текущее значение импульсов (младшее слово)	R	0	-	-	+
D1337	CH0 (Y0, Y1) текущее значение импульсов (старшее слово)	R	0	-	-	+
D1338	CH1 (Y2, Y3) текущее значение импульсов (младшее слово)	R	0	-	-	+
D1339	CH1 (Y2, Y3) текущее значение импульсов (старшее слово)	R	0	-	-	+
D1340	Темп разгона l	R/W *	200	-	-	+
D1341	Максимальная выходная частота (младшее слово)	R/W *	H'04D0	-	-	+
D1342	Максимальная выходная частота (старшее слово)	R/W *	3	-	-	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1343	Время разгона/замедления	R/W *	100	-	-	+
D1344	CH0 (Y0, Y1) число импульсов дополнения (младшее слово)	R/W	0	-	-	+
D1345	CH0 (Y0, Y1) число импульсов дополнения (старшее слово)	R/W	0	-	-	+
D1346	CH1 (Y2, Y3) число импульсов дополнения (младшее слово)	R/W	0	-	-	+
D1347	CH1 (Y2, Y3) число импульсов дополнения (старшее слово)	R/W	0	-	-	+
D1348	CH0 (Y10) текущее значение импульсов (младшее слово)	R	0	-	+	-
D1349	CH0 (Y10) текущее значение импульсов (старшее слово)	R	0	-	+	-
D1350	CH1 (Y11) текущее значение импульсов (младшее слово)	R	0	-	+	-
D1351	CH1 (Y11) текущее значение импульсов (старшее слово)	R	0	-	+	-
D1352	CH1 (Y2, Y3) начальная (минимальная) частота импульсов	R/W*	200	-	+	+
D1353	CH1 (Y2, Y3) время разгона/замедления	R/W*	100	-	+	+
D1355	Коммуникационный адрес чтения от PLC LINK ID1	R/W	H'1046	-	+	+
D1356	Коммуникационный адрес чтения от PLC LINK ID2	R/W	H'1046	-	+	+
D1357	Коммуникационный адрес чтения от PLC LINK ID3	R/W	H'1046	-	+	+
D1358	Коммуникационный адрес чтения от PLC LINK ID4	R/W	H'1046	-	+	+
D1359	Коммуникационный адрес чтения от PLC LINK ID5	R/W	H'1046	-	+	+
D1360	Коммуникационный адрес чтения от PLC LINK ID6	R/W	H'1046	-	+	+
D1361	Коммуникационный адрес чтения от PLC LINK ID7	R/W	H'1046	-	+	+
D1362	Коммуникационный адрес чтения от PLC LINK ID8	R/W	H'1046	-	+	+
D1363	Коммуникационный адрес чтения от PLC LINK ID9	R/W	H'1046	-	+	+
D1364	Коммуникационный адрес чтения от PLC LINK ID10	R/W	H'1046	-	+	+
D1365	Коммуникационный адрес чтения от PLC LINK ID11	R/W	H'1046	-	+	+
D1366	Коммуникационный адрес чтения от PLC LINK ID12	R/W	H'1046	-	+	+
D1367	Коммуникационный адрес чтения от PLC LINK ID13	R/W	H'1046	-	+	+
D1368	Коммуникационный адрес чтения от PLC LINK ID14	R/W	H'1046	-	+	+
D1369	Коммуникационный адрес чтения от PLC LINK ID15	R/W	H'1046	-	+	+
D1370	Коммуникационный адрес чтения от PLC LINK ID16	R/W	H'1046	-	+	+
D1375	Координата X первого KEY модуля	R	0	-	-	+
D1376	Координата Y первого KEY модуля	R	0	-	-	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1377	Число кнопок первого KEY модуля	R	0	-	-	+
D1378	Координата X второго KEY модуля	R	0	-	-	+
D1379	Координата Y второго KEY модуля	R	0	-	-	+
D1380	Число кнопок второго KEY модуля	R	0	-	-	+
D1381	Первый BCD модуль (старший байт)	R	0	-	-	+
D1382	Первый BCD модуль (младший байт)	R	0	-	-	+
D1383	Второй BCD модуль (старший байт)	R	0	-	-	+
D1384	Второй BCD модуль (младший байт)	R	0	-	-	+
D1385	Первый DISP модуль (старший байт)	R/W	0	-	-	+
D1386	Первый DISP модуль (младший байт)	R/W	0	-	-	+
D1387	Первый DISP модуль – десятичная точка	R/W	0	-	-	+
D1388	Второй DISP модуль (старший байт)	R/W	0	-	-	+
D1389	Второй DISP модуль (младший байт)	R/W	0	-	-	+
D1390	Второй DISP модуль – десятичная точка	R/W	0	-	-	+
D1391	Третий DISP модуль (старший байт)	R/W	0	-	-	+
D1392	Третий DISP модуль (младший байт)	R/W	0	-	-	+
D1393	Третий DISP модуль – десятичная точка	R/W	0	-	-	+
D1399	PLC LINK указание ID номера начальной SLAVE-станции	R/W	1	-	+	+
D1415	Коммуникационный адрес записи в PLC LINK ID1	R/W	H'10C8	-	+	+
D1416	Коммуникационный адрес записи в PLC LINK ID2	R/W	H'10C8	-	+	+
D1417	Коммуникационный адрес записи в PLC LINK ID3	R/W	H'10C8	-	+	+
D1418	Коммуникационный адрес записи в PLC LINK ID4	R/W	H'10C8	-	+	+
D1419	Коммуникационный адрес записи в PLC LINK ID5	R/W	H'10C8	-	+	+
D1420	Коммуникационный адрес записи в PLC LINK ID6	R/W	H'10C8	-	+	+
D1421	Коммуникационный адрес записи в PLC LINK ID7	R/W	H'10C8	-	+	+
D1422	Коммуникационный адрес записи в PLC LINK ID8	R/W	H'10C8	-	+	+
D1423	Коммуникационный адрес записи в PLC LINK ID9	R/W	H'10C8	-	+	+
D1424	Коммуникационный адрес записи в PLC LINK ID10	R/W	H'10C8	-	+	+
D1425	Коммуникационный адрес записи в PLC LINK ID11	R/W	H'10C8	-	+	+
D1426	Коммуникационный адрес записи в PLC LINK ID12	R/W	H'10C8	-	+	+
D1427	Коммуникационный адрес записи в PLC LINK ID13	R/W	H'10C8	-	+	+
D1428	Коммуникационный адрес записи в PLC LINK ID14	R/W	H'10C8	-	+	+
D1429	Коммуникационный адрес записи в PLC LINK ID15	R/W	H'10C8	-	+	+
D1430	Коммуникационный адрес записи в PLC LINK ID16	R/W	H'10C8	-	+	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1431	PLC LINK время	R/W	0	-	+	+
D1432	PLC LINK счетчик	R/W	0	-	+	+
D1433	PLC LINK единицы	R/W	0	-	+	+
D1434	Число читаемых регистров от PLC LINK ID1	R/W	16	-	+	+
D1435	Число читаемых регистров от PLC LINK ID2	R/W	16	-	+	+
D1436	Число читаемых регистров от PLC LINK ID3	R/W	16	-	+	+
D1437	Число читаемых регистров от PLC LINK ID4	R/W	16	-	+	+
D1438	Число читаемых регистров от PLC LINK ID5	R/W	16	-	+	+
D1439	Число читаемых регистров от PLC LINK ID6	R/W	16	-	+	+
D1440	Число читаемых регистров от PLC LINK ID7	R/W	16	-	+	+
D1441	Число читаемых регистров от PLC LINK ID8	R/W	16	-	+	+
D1442	Число читаемых регистров от PLC LINK ID9	R/W	16	-	+	+
D1443	Число читаемых регистров от PLC LINK ID10	R/W	16	-	+	+
D1444	Число читаемых регистров от PLC LINK ID11	R/W	16	-	+	+
D1445	Число читаемых регистров от PLC LINK ID12	R/W	16	-	+	+
D1446	Число читаемых регистров от PLC LINK ID13	R/W	16	-	+	+
D1447	Число читаемых регистров от PLC LINK ID14	R/W	16	-	+	+
D1448	Число читаемых регистров от PLC LINK ID15	R/W	16	-	+	+
D1449	Число читаемых регистров от PLC LINK ID16	R/W	16	-	+	+
D1450	Число записываемых регистров в PLC LINK ID1	R/W	16	-	+	+
D1451	Число записываемых регистров в PLC LINK ID2	R/W	16	-	+	+
D1452	Число записываемых регистров в PLC LINK ID3	R/W	16	-	+	+
D1453	Число записываемых регистров в PLC LINK ID4	R/W	16	-	+	+
D1454	Число записываемых регистров в PLC LINK ID5	R/W	16	-	+	+
D1455	Число записываемых регистров в PLC LINK ID6	R/W	16	-	+	+
D1456	Число записываемых регистров в PLC LINK ID7	R/W	16	-	+	+
D1457	Число записываемых регистров в PLC LINK ID8	R/W	16	-	+	+
D1458	Число записываемых регистров в PLC LINK ID9	R/W	16	-	+	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1459	Число записываемых регистров в PLC LINK ID10	R/W	16	-	+	+
D1460	Число записываемых регистров в PLC LINK ID11	R/W	16	-	+	+
D1461	Число записываемых регистров в PLC LINK ID12	R/W	16	-	+	+
D1462	Число записываемых регистров в PLC LINK ID13	R/W	16	-	+	+
D1463	Число записываемых регистров в PLC LINK ID14	R/W	16	-	+	+
D1464	Число записываемых регистров в PLC LINK ID15	R/W	16	-	+	+
D1465	Число записываемых регистров в PLC LINK ID16	R/W	16	-	+	+
D1466	СН0 число импульсов на один оборот двигателя (младшее слово)	R*	2000	-	-	+
D1467	СН0 число импульсов на один оборот двигателя (старшее слово)	R*	0	-	-	+
D1468	СН1 число импульсов на один оборот двигателя (младшее слово)	R*	2000	-	-	+
D1469	СН1 число импульсов на один оборот двигателя (старшее слово)	R*	0	-	-	+
D1470	СН0 линейное перемещение на один оборот двигателя (младшее слово)	R*	1000	-	-	+
D1471	СН0 линейное перемещение на один оборот двигателя (старшее слово)	R*	0	-	-	+
D1472	СН1 линейное перемещение на один оборот двигателя (младшее слово)	R*	1000	-	-	+
D1473	СН1 линейное перемещение на один оборот двигателя (старшее слово)	R*	0	-	-	+
D1474	СН0 механическая единица движения (младшее слово)	R*	0	-	-	+
D1475	СН0 механическая единица движения (старшее слово)	R*	0	-	-	+
D1476	СН1 механическая единица движения (младшее слово)	R*	0	-	-	+
D1477	СН1 механическая единица движения (старшее слово)	R*	0	-	-	+
D1480 ... D1495	Регистры, из которых считываются данные для PLC LINK ID1 с адресом D1355. Диапазон D100-D115.	R	0	-	+	+
D1496 ... D1511	Регистры, в которые записываются данные от PLC LINK ID1 с адресом D1415. Диапазон D200-D215.	R/W	0	-	+	+
D1512 ... D1527	Регистры, из которых считываются данные для PLC LINK ID2 с адресом D1356. Диапазон D100-D115.	R	0	-	+	+
D1528 ... D1543	Регистры, в которые записываются данные от PLC LINK ID2 с адресом D1416. Диапазон D200-D215.	R/W	0	-	+	+
D1544 ... D1559	Регистры, из которых считываются данные для PLC LINK ID3 с адресом D1357. Диапазон D100-D115.	R	0	-	+	+

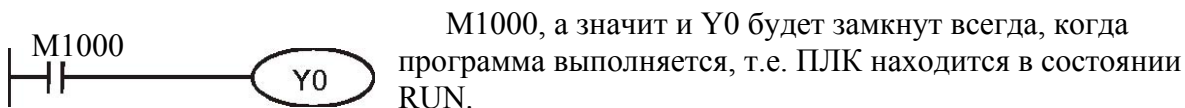
Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EN
D1560 ... D1575	Регистры, в которые записываются данные от PLC LINK ID3 с адресом D1417. Диапазон D200-D215.	R/W	0	-	+	+
D1576 ... D1591	Регистры, из которых считываются данные для PLC LINK ID4 с адресом D1358. Диапазон D100-D115.	R	0	-	+	+
D1592 ... D1607	Регистры, в которые записываются данные от PLC LINK ID4 с адресом D1418. Диапазон D200-D215.	R/W	0	-	+	+
D1608 ... D1623	Регистры, из которых считываются данные для PLC LINK ID5 с адресом D1359. Диапазон D100-D115.	R	0	-	+	+
D1624 ... D1639	Регистры, в которые записываются данные от PLC LINK ID5 с адресом D1419. Диапазон D200-D215.	R/W	0	-	+	+
D1640 ... D1655	Регистры, из которых считываются данные для PLC LINK ID6 с адресом D1360. Диапазон D100-D115.	R	0	-	+	+
D1656 ... D1671	Регистры, в которые записываются данные от PLC LINK ID6 с адресом D1420. Диапазон D200-D215.	R/W	0	-	+	+
D1672 ... D1687	Регистры, из которых считываются данные для PLC LINK ID7 с адресом D1361. Диапазон D100-D115.	R	0	-	+	+
D1688 ... D1703	Регистры, в которые записываются данные от PLC LINK ID7 с адресом D1421. Диапазон D200-D215.	R/W	0	-	+	+
D1704 ... D1719	Регистры, из которых считываются данные для PLC LINK ID8 с адресом D1362. Диапазон D100-D115.	R	0	-	+	+
D1720 ... D1735	Регистры, в которые записываются данные от PLC LINK ID8 с адресом D1422. Диапазон D200-D215.	R/W	0	-	+	+
D1736 ... D1751	Регистры, из которых считываются данные для PLC LINK ID9 с адресом D1363. Диапазон D100-D115.	R	0	-	+	+
D1752 ... D1767	Регистры, в которые записываются данные от PLC LINK ID9 с адресом D1423. Диапазон D200-D215.	R/W	0	-	+	+
D1768 ... D1783	Регистры, из которых считываются данные для PLC LINK ID10 с адресом D1364. Диапазон D100-D115.	R	0	-	+	+
D1784 ... D1799	Регистры, в которые записываются данные от PLC LINK ID10 с адресом D1424. Диапазон D200-D215.	R/W	0	-	+	+

Номер	Функция	Атрибут	Заводск. установка	Серия ПЛК		
				ES/EX/SS	SA/SX	EH
D1800 ... D1815	Регистры, из которых считываются данные для PLC LINK ID11 с адресом D1365. Диапазон D100-D115.	R	0	-	+	+
D1816 ... D1831	Регистры, в которые записываются данные от PLC LINK ID11 с адресом D1425. Диапазон D200-D215.	R/W	0	-	+	+
D1832 ... D1847	Регистры, из которых считываются данные для PLC LINK ID12 с адресом D1366. Диапазон D100-D115.	R	0	-	+	+
D1848 ... D1863	Регистры, в которые записываются данные от PLC LINK ID12 с адресом D1426. Диапазон D200-D215.	R/W	0	-	+	+
D1864 ... D1879	Регистры, из которых считываются данные для PLC LINK ID13 с адресом D1367. Диапазон D100-D115.	R	0	-	+	+
D1880 ... D1895	Регистры, в которые записываются данные от PLC LINK ID13 с адресом D1427. Диапазон D200-D215.	R/W	0	-	+	+
D1896 ... D1911	Регистры, из которых считываются данные для PLC LINK ID14 с адресом D1368. Диапазон D100-D115.	R	0	-	+	+
D1912 ... D1927	Регистры, в которые записываются данные от PLC LINK ID14 с адресом D1428. Диапазон D200-D215.	R/W	0	-	+	+
D1928 ... D1943	Регистры, из которых считываются данные для PLC LINK ID15 с адресом D1369. Диапазон D100-D115.	R	0	-	+	+
D1944 ... D1959	Регистры, в которые записываются данные от PLC LINK ID15 с адресом D1429. Диапазон D200-D215.	R/W	0	-	+	+
D1960 ... D1975	Регистры, из которых считываются данные для PLC LINK ID16 с адресом D1370. Диапазон D100-D115.	R	0	-	+	+
D1976 ... D1991	Регистры, в которые записываются данные от PLC LINK ID16 с адресом D1430. Диапазон D200-D215.	R/W	0	-	+	+

2.11. Описание специальных реле и регистров

2.11.1. Флаги состояния ПЛК (M1000 – M1003)

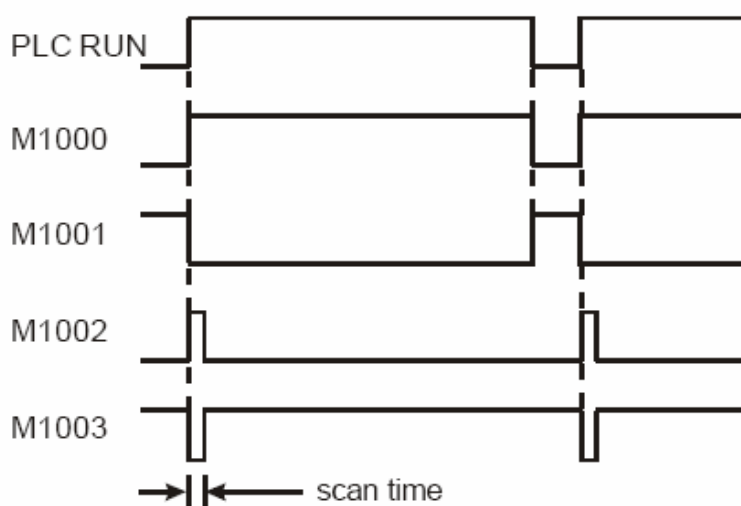
M1000: Отображает состояние ПЛК – контакт всегда замкнут в режиме RUN (работа) и разомкнут в состоянии STOP (стоп).



M1001: Отображает состояние ПЛК – контакт всегда разомкнут в режиме RUN (работа) и замкнут в состоянии STOP (стоп).

M1002 замкнут в течение первого цикла сканирования, когда ПЛК стартует и разомкнут в течении остального времени. Реле M1002 удобно использовать в программе в качестве импульса для функции начальной инициализации.

M1003 разомкнут в течение первого цикла сканирования, когда ПЛК стартует и замкнут в течении остального времени.



2.11.2. Сторожевой таймер (D1000)

Используется для контроля времени цикла. Если время сканирования превысит время заданное в сторожевом таймере, загорится светодиод "ERROR" и все выходы будут отключены. Настраивается в единицах 1 мс. Стандартное значение 200 мс. Для изменения времени можно использовать команду MOV:



Максимальное значение сторожевого таймера: 32767 мс. Внимательно устанавливайте значение таймера, т.к. большое значение может привести к большой задержке обнаружения аварийной ситуации. Время сканирования может быть очень длинным за счет использования в программе сложных вычислений или большого количества специальных модулей расширения. Для корректной установки времени сторожевого таймера посмотрите реальное и минимальное время сканирования в регистрах D1010, D1011.

Вы так же можете использовать в программе для контроля времени цикла команду WDT (API 07), которая может разделять цикл программы на отрезки со сбросом времени сторожевого таймера в конце каждого отрезка программы.

2.11.3. Объем памяти программы (D1002)

В различных сериях ПЛК это значение будет разным:

1. ES, EX, SS серия: 3792 шагов
2. SX, SA серия: 7920 шагов
3. EN серия: 15872 шагов

2.11.4. Проверка грамматических ошибок в программе (M1004, D1004, D1137)

При обнаружении в программе ошибки светодиод "ERROR" будет мигать и специальное реле M1004 установится в состояние 1.

Проверка грамматических ошибок происходит при включении ПЛК, а так же при загрузке программы в ПЛК и при использовании функции On-line программирования в DVP-EN и WPLSoft.

Каждая возникающая ошибка фиксируется в специальном регистре (записывается номер шага в D1137 и код ошибки в D1004). Эта информация может быть считана с помощью ПК или программатора. В разделе 2.12 приведен список с описаниями кодов ошибок, которые могут быть считаны из ЦПУ.

2.11.5. Память резервирования данных (M1005 ~ M1007)

Когда внешняя карта памяти вставлена в DVP-EN и переключатель ON/OFF включен, процессор производит её инициализацию, и если не возникло проблем, то данные из карты памяти будут скопированы в ПЛК. Если при инициализации возникли проблемы, они будут отражены в следующих специальных реле:

- M1005=1, если пароль в карте памяти не совпадает с паролем в ПЛК;
- M1006=1, если карта памяти не прошла инициализацию;
- M1007=1, если данные отсутствуют в области программы карты памяти.

2.11.6. Флаги сторожевого таймера (M1008, D1008)

Если время сканирования превысит заданное время, загорится светодиод "ERROR" и контакт M1008 замкнется. В регистре D1008 будет записан номер шага программы на котором была вызвана задержка выполнения программы.

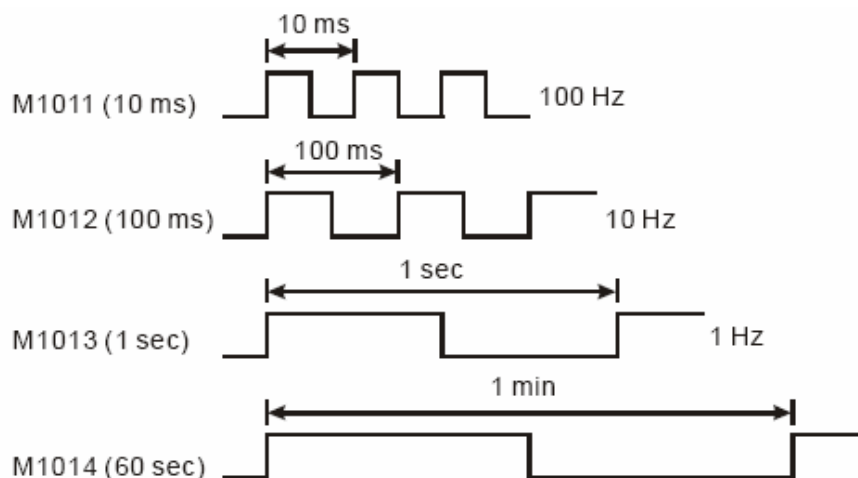
2.11.7. Монитор времени сканирования (D1010~D1012)

В этих регистрах отображается текущее, минимальное и максимальное время выполнения программы:

- D1010: текущее время сканирования.
- D1011: минимальное время сканирования.
- D1012: максимальное время сканирования.

2.11.8. Внутренние тактовые импульсы (M1011~M1014)

ПЛК может автоматически выдавать в программе импульсы с четырьмя различными периодами:



Внутренние контакты (M1011~M1014) при использовании их в программе будут замыкаться с определенной частотой на протяжении всего времени работы ПЛК (даже в состоянии СТОП). При переводе ПЛК в состояние RUN тактовые импульсы с началом выполнения программы не синхронизируются.

2.11.9. Высокоскоростной таймер (M1015, D1015)

Используется для подсчета времени с дискретностью 0.1 мс.

Работает только в состоянии RUN.

Когда M1015 = 1, начнется работа высокоскоростного таймера сразу после выполнения инструкции END в текущем скане. Текущее время записывается в специальном регистре D1015. Диапазон D1015: 0 ... 32767. Счет идет по кругу, т.е. при достижении текущего значения 32767 счет снова начнется с нуля.

Когда M1015 = 1, счет времени немедленно прекратится.

Для работы с высокоскоростным таймером в DVP-EN может использоваться инструкция HST (API 196).



2.11.10. Часы реального времени (M1016, M1017, M1076, D1313~D1319)

M1016	Отображение года	Когда M1016=OFF, будут отображаться две младших цифры. Когда M1016=ON, будут отображаться две младших цифры + 2000
M1017	Корректировка секунд (± 30 сек)	При переключении с OFF на ON секунды будут обнулены и минуты не изменятся, если секунды находились в диапазоне от 0 до 29 и будет добавлена 1 минута, если секунды находились в диапазоне 30 ... 59 сек.
M1076	Ошибка календаря (часов реального времени)	Контакт включится если будет превышен установленный диапазон или батарея будет иметь низкий заряд
D1313	Часы реального времени (RTC): отображение и коррекция секунд	(00...59)
D1314	Отображение и коррекция минут	(00...59)
D1315	Отображение и коррекция часов	(00...23)
D1316	Отображение и коррекция дня месяца	(01...31)
D1317	Отображение и коррекция месяца	(01...12)
D1318	Отображение и коррекция дня недели	(1...7)
D1319	Отображение и коррекция года	(00...99)

При сбросе часов реального времени произойдет сброс на 1 января 2000 г. 00:00 суббота.

Корректировка часов реального времени в DVP-SA/SX/EN может быть выполнена с помощью инструкции TWR (API 167), а также WPLSoft или панели DU01.

2.11.11. Число π (D1018, D1019)

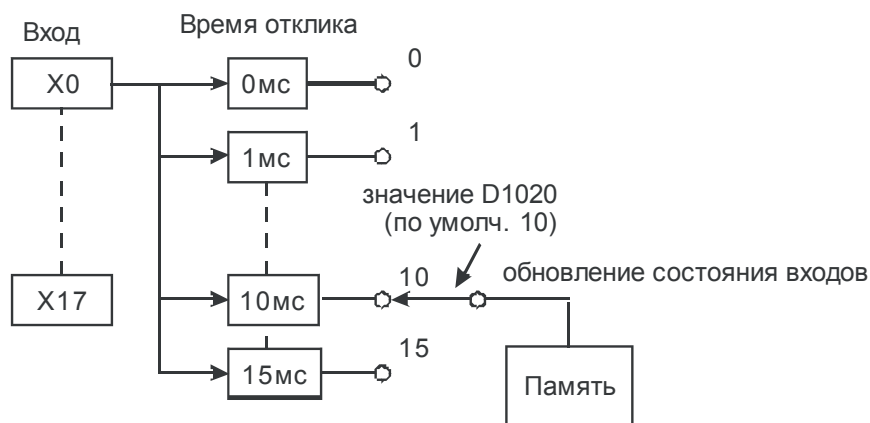
Число π записано в двух регистрах (32 бита) в формате числа с плавающей запятой.

Значение в шестнадцатеричном виде = 40490FBD.

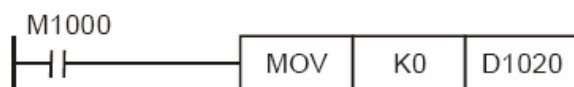
2.11.12. Корректировка времени реакции дискретных входов (M1019, D1020, D1021)

Входной цифровой фильтр терминалов X0 – X17 позволяет устанавливать в регистрах D1020, D1021 время отклика (в мс) на входные импульсы.

При включении ПЛК содержимое регистров D1020, D1021 будет автоматически установлено на значение 10 мс.



Если D1020 = 0, время отклика будет зависеть только от последовательного RC-фильтра и будет максимально быстрым (50 мкс).



Не необходимости корректировать время отклика дискретных входов, когда в программе используются высокоскоростные счетчики, внешние прерывания или импульсные ловушки.

Для корректировки времени отклика дискретных входов так же можно использовать команду REFF (API 51).

2.11.13. Флаги завершения выполнения (M1029, M1030)

1. M1029 = 1 при завершении выполнения команд MTR (API 52), HKY (API 71), DSW (API 72), SEGL (API 74), PR (API 77).
2. Команды PLSY (API 57), PLSR (API 59):
 - Для DVP-ES/EX/SS/SA/SX, M1029 будет = 1, когда на выходе Y0 завершится генерация импульсов; M1030 будет = 1, когда на выходе Y1 завершится генерация импульсов. Когда условие выполнения команд будет = 0, M1029, M1030 будут = 0.
 - Для DVP-EN, M1029 будет = 1, когда на выходах Y0, Y1 завершится генерация импульсов; M1030 будет = 1, когда на выходах Y2, Y3 завершится генерация импульсов. Когда условие выполнения команд будет = 0, M1029, M1030 будут = 0.
3. M1029 = 1 при достижении последней уставки многоуставочного счетчика, заданного командой INCD (API 63).
4. M1029 = 1 при завершении выполнения команд RAMP (API 67), SORT (API 69). Когда условие выполнения команд будет = 0, M1029 будет = 0.
5. Команды DABSR (API 155), ZRN (API 156), DRV1 (API 158), DRVA (API 158):
 - M1029 будет = 1, когда на выходах Y0, Y1 завершится генерация импульсов; M1030 будет = 1, когда на выходах Y2, Y3 завершится генерация импульсов.
 - Во время выполнения команд, M1029, M1030 будут = 0, а после завершения выполнения M1029, M1030 будут = 1.

2.11.14. Код ошибки коммуникации (D1025)

D1025 = 1: недопустимая команда;

D1025 = 2: недействительный коммуникационный адрес устройства;

D1025 = 3: запрашиваемые данные превышают допустимый диапазон;

D1025 = 7: ошибка контрольной суммы.

2.11.15. Команда очистки памяти данных (M1031, M1032)

M1031 очищает всю оперативную память данных:

- состояние контактов Y, реле общего назначения M и S;
- регистры и контакты таймеров общего назначения T;
- регистры и контакты счетчиков общего назначения C;
- регистры данных общего назначения D.

M1032 очищает всю энергонезависимую память данных:

- состояние контактов энергонезависимых реле M и S;
- регистры и контакты аккумулятивных таймеров T;
- регистры и контакты энергонезависимых счетчиков C;
- энергонезависимые регистры данных D.

2.11.16. Сохранение состояния выходов в режиме СТОП (M1033)

Если M1033 = 1, состояние выходов будет сохранено в режиме СТОП, например, если нельзя отключать выходы при перезаписи программы.

2.11.17. Запрещение включения выходов Y (M1034)

Когда M1034 = 1, все выходы Y будут отключены независимо от условий выполнения программы.

2.11.18. Переключатель режимов РАБОТА/СТОП (M1035, D1035)

В DVP-EN, когда M1035 = 1, одному из входов X0 – X17 будет назначена функция переключателя режимов РАБОТА/СТОП. Номер входа будет зависеть от содержимого регистра D1035 (0 - 17).

В DVP-SA/SX, когда M1035 = 1, входу X7 будет назначена функция переключателя режимов РАБОТА/СТОП.

2.11.19. Настройка коммуникационных портов (M1120, M1136, M1138, M1139, M1143, D1036, D1109, D1120)

Порт COM1(RS-232): может работать только в режиме SLAVE с кодами ASCII/RTU с корректировкой скорости передачи (до 115 200 бит/с).

Порт COM2(RS-232/RS-485/RS-422): может работать как в режиме MASTER так и SLAVE с кодами ASCII/RTU с корректировкой скорости передачи (до 115 200 бит/с).

Порт COM3(RS-232/RS485): может работать только в режиме SLAVE с кодами ASCII с протоколом 7, E, 1 и корректировкой скорости передачи (до 38 400 бит/с).

Установка протоколов коммуникации:

COM1	D1036: установка протокола коммуникации M1138: фиксация протокола коммуникации M1139: выбор ASCII/RTU-режима
COM2	D1120: установка протокола коммуникации M1120: фиксация протокола коммуникации M1143: выбор ASCII/RTU-режима
COM3	D1109: установка протокола коммуникации M1136: фиксация протокола коммуникации

D1036: протокол коммуникации для COM1 RS-232 (Slave). Биты b8~b15 не поддерживаются.

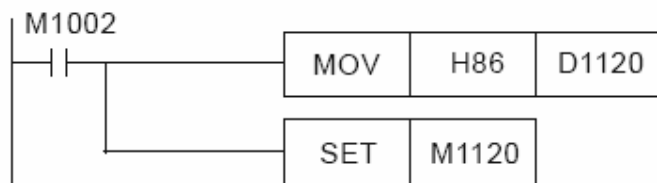
D1109: протокол коммуникации для COM3 RS-232/RS-485 (Slave). Биты b0~b3 и b8~b15 не поддерживаются

D1120: протокол коммуникации для COM2 RS-232/RS-485/RS-422 (Master и Slave). Описание битов см. в таблице:

	Содержимое	0	1
b0	Длина данных	b0=0: 7	b0=1: 8
b1 b2	Бит паритета	b2, b1=00 : (None) b2, b1=01 : (Odd) b2, b1=11 : (Even)	
b3	Стоповый бит	b3=0: 1 bit	b3=1: 2 bit
b4 b5 b6 b7	b7~b4=0001 (H1) : b7~b4=0010 (H2) : b7~b4=0011 (H3) : b7~b4=0100 (H4) : b7~b4=0101 (H5) : b7~b4=0110 (H6) : b7~b4=0111 (H7) : b7~b4=1000 (H8) : b7~b4=1001 (H9) : b7~b4=1010 (HA) : b7~b4=1011 (HB) : b7~b4=1100 (HC) :	110 150 300 600 1200 2400 4800 9600 19200 38400 57600 115200	bps bps bps bps bps bps bps bps bps bps bps bps
b8	Стартовый символ	b8=0: нет	b8=1: D1124
b9	Первый стоповый символ	b9=0: нет	b9=1: D1125
b10	Второй стоповый символ	b10=0: нет	b10=1: D1126
b15~b11	Нет функции		

Пример 1: Установка протокола коммуникации для COM2

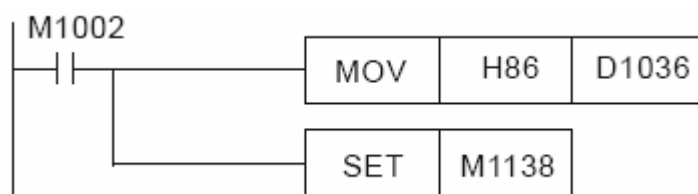
При включении ПЛК для COM2 будет установлен следующий коммуникационный протокол: ASCII, 9600bps, 7 Data bits, Even parity, 1 Stop bits (9600, 7, E, 1) и зафиксирован с помощью реле M1120.

**Примечания:**

1. Когда порт COM2 используется в режиме SLAVE, убедитесь, что бы в программе не было коммуникационных инструкций.
2. Коммуникационный протокол не будет изменяться, когда ПЛК переключается с RUN на STOP.
3. После каждого выключения и включения ПЛК, коммуникационный протокол будет возвращаться к заводской установке.

Пример 2: Установка протокола коммуникации для COM1

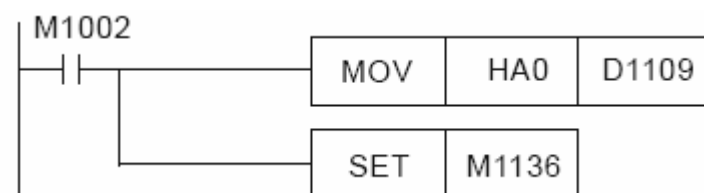
При включении ПЛК для COM1 будет установлен следующий коммуникационный протокол: ASCII, 9600bps, 7 Data bits, Even parity, 1 Stop bits (9600, 7, E, 1) и зафиксирован с помощью реле M1138.

**Примечания:**

1. Коммуникационный протокол не будет изменяться, когда ПЛК переключается с RUN на STOP.
2. После каждого выключения и включения ПЛК, коммуникационный протокол будет возвращаться к заводской установке.

Пример 3: Установка протокола коммуникации для COM3

При включении ПЛК для COM3 будет установлен следующий коммуникационный протокол: ASCII, 9600bps, 7 Data bits, Even parity, 1 Stop bits (9600, 7, E, 1) и зафиксирован с помощью реле M1136.



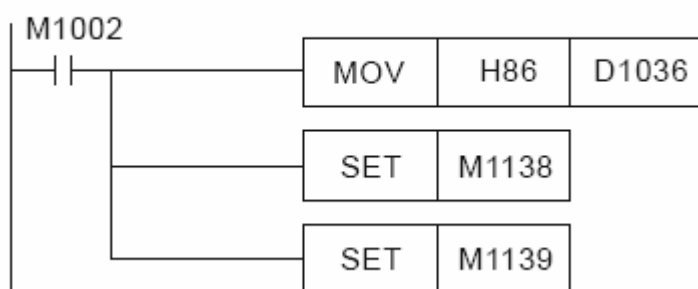
Примечания:

1. Коммуникационный протокол не будет изменяться, когда ПЛК переключается с RUN на STOP.
2. После каждого выключения и включения ПЛК, коммуникационный протокол будет возвращаться к заводской установке.

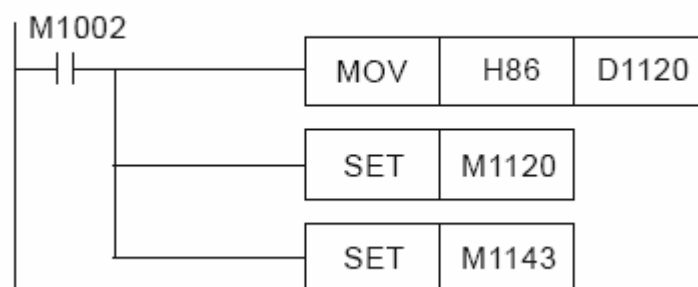
Пример 4: Установка режима RTU для COM1 и COM2

При включении ПЛК для портов COM1 и COM2 с помощью M1139 и M1143 будет установлен режим RTU.

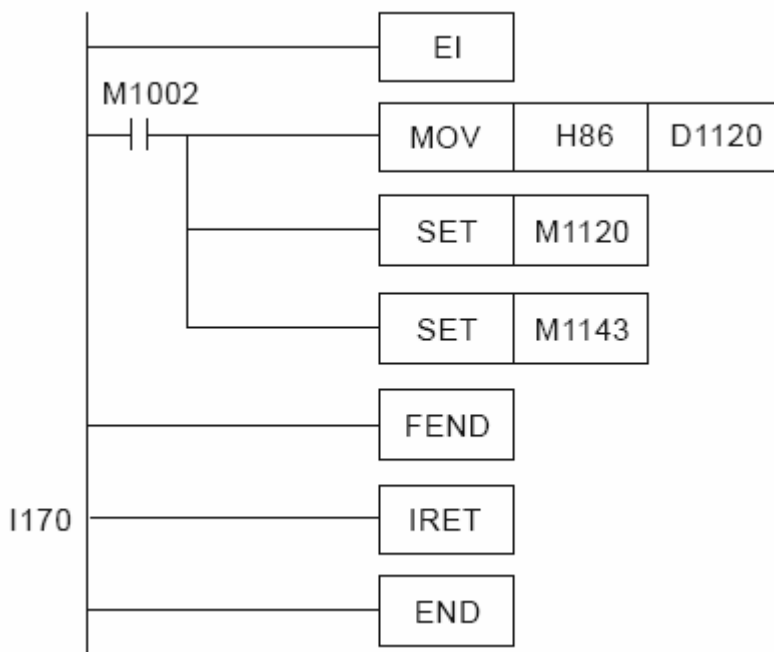
COM1 :



COM2 :



Пример 5: Коммуникационное прерывание I170 после завершения приема данных в режиме SLAVE по COM2.



Примечание:

1. Не обновляйте программу в онлайн когда используется коммуникационное прерывание.
2. Время скана будет увеличено.

2.11.20. Задержка коммуникационного отклика (D1038)

Время задержки для ответной посылки данных, когда ПЛК – Slave в сети RS-485 (ед. = 0.1 мс). Диапазон: 0...10000

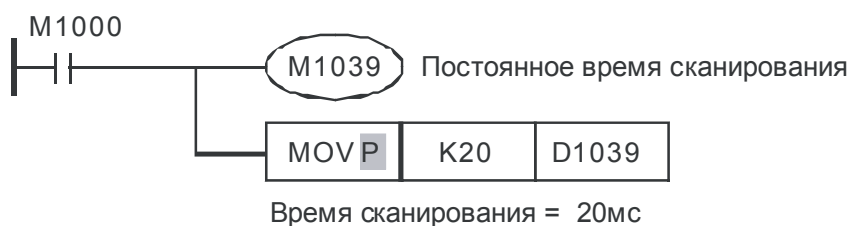
2.11.21. Постоянная величина времени сканирования (M1039, D1039)

Когда M1039 = 1, время выполнения цикла программы будет определяться значением регистра D1039.

Если реальное время выполнения цикла программы будет меньше значения D1039, то время сканирования будет равно значению D1039.

Если реальное время выполнения цикла программы будет больше значения D1039, то время сканирования будет равно реальному времени сканирования.

Выбирайте постоянное время сканирования больше чем среднее время скана программы.



Команды, зависящие от времени сканирования (RAMP(API 67), HKY(API 71), SEGL(API 74), ARWS(API 75) и PR(API 77)) должны использоваться с функцией постоянного времени сканирования или вставки постоянного времени прерывания.

Для команды HKY(API 71) время сканирования должно быть 20 мс и выше при использовании в работе от 16-ти клавиш.

Отображение времени сканирования в регистрах D1010~D1012 включает в себя постоянное время сканирования.

2.11.22. Функции аналоговых входов и выходов (D1056~D1059, D1110~D1113, D1116~D1118)

D1056	Текущее значение на аналоговом входе (канал CH0) для DVP-EX/SX и функциональной карты DVP-EN
D1057	Текущее значение на аналоговом входе (канал CH1) для DVP-EX/SX и функциональной карты DVP-EN
D1058	Текущее значение на аналоговом входе (канал CH2) для DVP-EX и функциональной карты DVP-EN
D1059	Текущее значение на аналоговом входе (канал CH3) для DVP-EX и функциональной карты DVP-EN
D1110	Усредненное значение на аналоговом входе (канал CH0) для DVP-EX/SX и функциональной карты DVP-EN
D1111	Усредненное значение на аналоговом входе (канал CH1) для DVP-EX/SX и функциональной карты DVP-EN
D1112	Усредненное значение на аналоговом входе (канал CH2) для DVP-EX и функциональной карты DVP-EN
D1113	Усредненное значение на аналоговом входе (канал CH3) для DVP-EX и функциональной карты DVP-EN
D1116	Значение на аналоговом выходе (канал CH0) для DVP-EX/SX и функциональной карты DVP-EN
D1117	Значение на аналоговом выходе (канал CH1) для DVP-EX/SX и функциональной карты DVP-EN
D1118	Фильтр для аналого-цифрового преобразователя DVP-EX/SX (мсек.)

АЦП в DVP-EX имеет разрядность 10 бит ($\pm 10\text{В}$ или $\pm 20\text{мА}$).

ЦАП в DVP-EX имеет разрядность 8 бит ($0 \dots 10\text{В}$ или $0 \dots 20\text{мА}$).

АЦП в DVP-SX и карта аналогового ввода для DVP-EN имеет разрядность 12 бит ($\pm 10\text{В}$ или $\pm 20\text{мА}$).

ЦАП в DVP-SX имеет разрядность 12 бит ($(\pm 10\text{В}$ или $\pm 20\text{мА})$).

ЦАП карты аналогового ввода для DVP-EN имеет разрядность 12 бит ($0 \dots 10\text{В}$ или $0 \dots 20\text{мА}$).

Минимальное значение цифрового фильтра АЦП может быть 5 мс.

2.11.23. Флаги ошибок алгоритма программы (M1067~M1068, D1067~D1068)

Флаги ошибок алгоритма.

Устройство	Описание	STOP → RUN	RUN → STOP
M1067	Флаг ошибки алгоритма программы	Сброс	Сохраняется
M1068	Флаг фиксации ошибки алгоритма программы	Сохраняется	Сохраняется
D1067	Код ошибки алгоритма программы	Сброс	Сохраняется
D1068	Шаг ошибки алгоритма программы	Сохраняется	Сохраняется

Коды ошибок алгоритма

Код ошибки D1067	Описание
0E18	Ошибка преобразования BCD
0E19	Деление на ноль
0E1A	Значение выходит за границы диапазона (включая E/F).
0E1B	Значение квадратного корня отрицательное
0E1C	Ошибка коммуникации FROM/TO

2.11.24. Низкое напряжение (M1087, D1100)

Когда ПЛК обнаружит низкое напряжение (сигнал LV), и если M1087=1, содержимое регистра D1100 будет сохранено на выходах Y0~Y17.

bit0 (LSB) в D1100 будет передан Y0, bit1 - Y1, bit8 - Y10, и т.д.

2.11.25. Файловые регистры (M1101, D1101 – D1103)

Для автоматической передачи данных при включении ПЛК или переключении его из режима STOP в режим RUN область чтения и записи данных в файловый регистр может быть задана в специальных регистрах:

D1101 определяет стартовый адрес файлового регистра (в DVP-SA/SX: K0...K1599; в DVP-EN: K0...K9999).

D1102 определяет число читаемых файловых регистров (в DVP-SA/SX: K0...K1600; в DVP-EN: K0...K10000).

D1103 определяет стартовый адрес для сохранения в файловый регистр (должен быть больше 2000).

M1101 разрешает/запрещает запись/чтение файловых регистров.

Область чтения и записи данных в файловый регистр задается в командах MEMR (API 147), MEMW (API 148).

2.11.26. Функциональная карта микропереключателей (M1104 – M1111)

Если контроллер DVP-EN используется с функциональной картой DVP-F8ID, имеющей 8 микропереключателей, то их состояние фиксируется в специальных реле M1104 – M1111.

Подробнее см. описание инструкции SWRD (API 109).

2.11.27. Функциональная карта расширения транзисторных выходов (M1112, M1113)

Если контроллер DVP-EN используется с функциональной картой DVP-F2OT, имеющей 2 дискретных транзисторных выхода, то их состояние задается в специальных реле M1112, M1113.

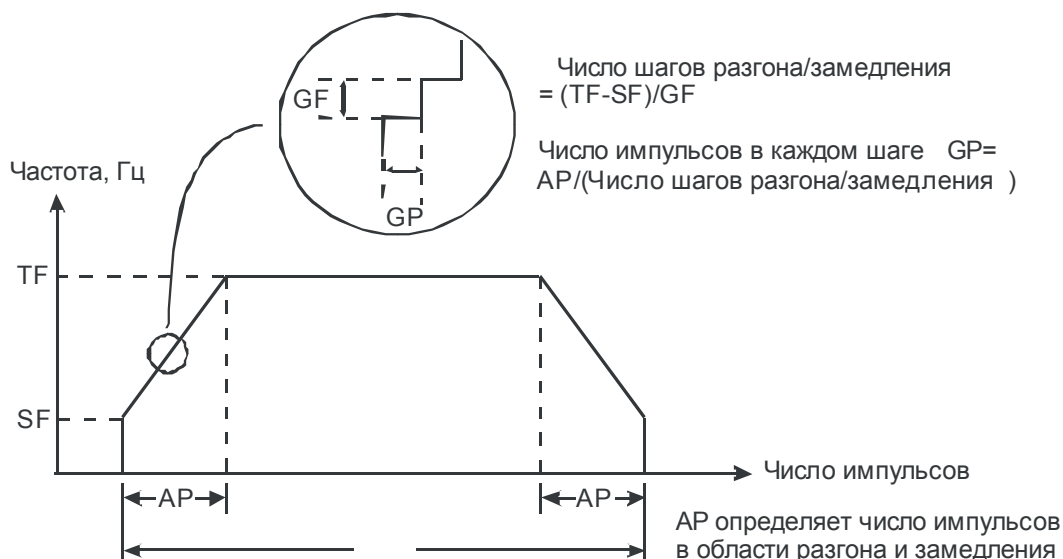
2.11.28. Импульсный выход с ускорением/замедлением (M1115 – M1119, D1104)

M1115	Старт выдачи импульсов с функцией разгона/замедления
M1116	Флаг разгона
M1117	Флаг достижения заданной частоты
M1118	Флаг замедления
M1119	Флаг выполнения функции
D1104	Указатель регистра D для задания параметров функции разгона/ замедления импульсного выхода Y0

Параметры функции разгона/замедления (диапазон частот: 25 Гц ... 10 кГц)

Индекс	Параметр	
+0	Начальная частота (SF)	
+1	Частота шага ускорения / замедления (GF)	
+2	Заданная частота (TF)	
+3	Полное число импульсов (младшие 16 бит)	(TP)
+4	Полное число импульсов (старшие 16 бит)	
+5	Число импульсов ускорения (младшие 16 бит)	(AP)
+6	Число импульсов замедления (старшие 16 бит)	

Функция импульсного выхода с разгоном/замедлением не требует использования специальной команды. Нужно только заполнить параметры функции и включить контакт M1115 и на выходе Y0 начнется генерация импульсов.



Примечание!

Функция будет выполняться только при соблюдении следующих условий:

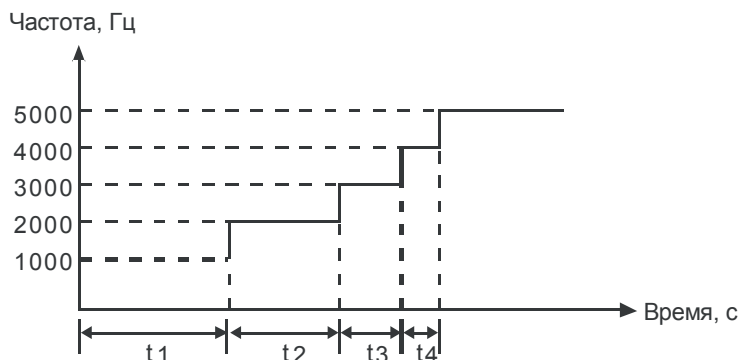
- начальная частота должна быть меньше заданной;
- частота шага должна быть меньше разности между заданной и начальной частотами;
- полное число импульсов должно быть больше числа импульсов разгона / замедления $\times 2$;
- начальная и заданная частоты должны находиться в диапазоне от 25 Гц до 10 кГц;
- число импульсов разгона/замедления должно быть больше чем число шагов разгона/замедления.

При выключении M1115, реле M1119 будет = 0, а состояния реле M1116 – M1118 останутся неизменными. При изменении состояния ПЛК: РАБОТА -> СТОП или СТОП -> РАБОТА все реле M1115 – M1119 будут установлены в ноль. D1104 будет = 0 при выключении ПЛК.

Если функция "импульсный выход с разгоном / замедлением" и команда PLSY Y0 используются одновременно, будет выполняться только одна из инструкций, запущенная первой.

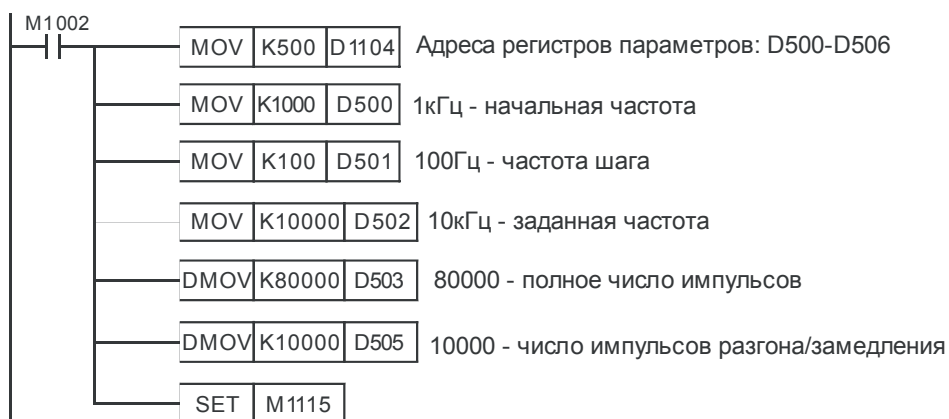
Пример расчета времени разгона:

Допустим начальная частота = 1 кГц, заданная частота = 5 кГц, частота шага = 1 кГц, полное число импульсов = 100, импульсов разгона/замедления = 40.

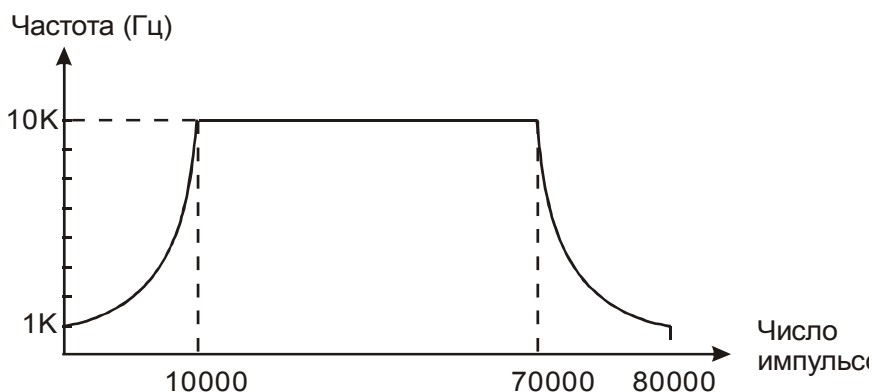


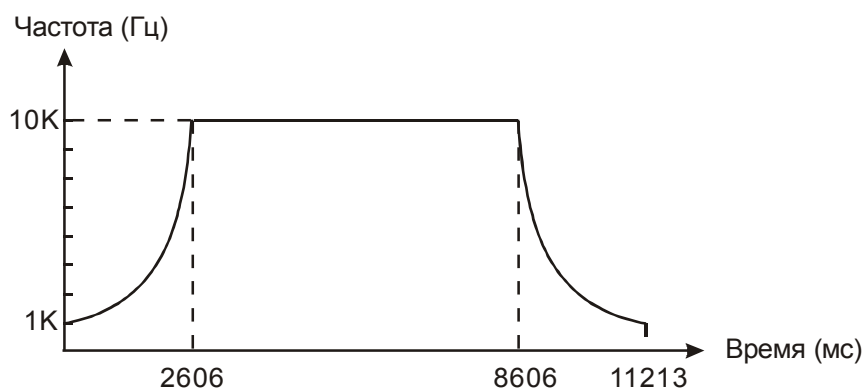
Число шагов разгона/замедления = $(5K - 1K) / 1K = 4$ и число импульсов в шаге $40 / 4 = 10$. Время каждого шага: $t1 = (1 / 1K) * 10 = 10\text{мс}$, $t2 = (1 / 2K) * 10 = 5\text{мс}$, $t3 = (1 / 3K) * 10 = 3.33\text{мс}$ и $t4 = (1 / 4K) * 10 = 2.5\text{мс}$.

Пример программы управления шаговым двигателем:



Диаграмма, иллюстрирующая выполнение программы:





2.11.29. Специальный высокоскоростной импульсный выход (M1133 – M1135, D1133)

Функция есть только в контроллерах серий SA/SX. Максимальная частота импульсов: 50 кГц.

M1133	Переключатель специального высокоскоростного (50 кГц) выхода (M1133 = ON - старт)
M1134	Когда M1134=ON импульсы идут непрерывно.
M1135	Флаг достижения заданного количества выходных импульсов
D1133	Указатель регистра D для задания параметров функции специального высокоскоростного импульсного выхода

Параметры функции специального высокоскоростного выхода

Индекс	Параметр
+0	Частота импульсов (младшие 16 бит)
+1	Частота импульсов (старшие 16 бит)
+2	Число импульсов (младшие 16 бит)
+3	Число импульсов (старшие 16 бит)
+4	Индикация числа переданных импульсов (младшие 16 бит)
+5	Индикация числа переданных импульсов (старшие 16 бит)

Выходная частота и заданное число импульсов могут быть изменены во время выполнения функции, когда M1133 = 1, а M1135 = 0.

Индикация числа переданных импульсов будет обновляться с началом каждого скана.

Функция "специальный высокоскоростной импульсный выход" может быть использована с выходом Y1.

Если функция "специальный высокоскоростной импульсный выход" и команда PLSY Y1 используются одновременно, будет выполняться только одна из инструкций, запущенная первой. С другими выходами PLSY Y0, Y2 – Y7, функция может работать одновременно.

Преимуществом функции "специальный высокоскоростной импульсный выход" перед командой PLSY является большая максимальная выходная частота (50 кГц).

2.11.30. Детектирование подключения специальных модулей расширения (D1139, D1140, D1142, D1143, D1145, D1146)

D1139	Число соединенных BCD-модулей расширения (макс. 2 модуля)
D1140	Число соединенных специальных модулей расширения AD, DA, XA, PT, TC, RT, HC, PU (макс. 8 модулей)
D1142	Точки ввода (X) от модулей расширения

D1143	Точки вывода (Y) модулей расширения
D1145	Число соединенных KEY-модулей расширения (макс. 2 модуля)
D1146	Число соединенных DISP-модулей расширения (макс. 3 модуля)

2.11.31. BCD-модуль (D1139, D1381 – D1384)

BCD-модули используются в контроллерах серии EN для цифрового ввода с помощью внешних микропереключателей в двоично-десятичном коде.

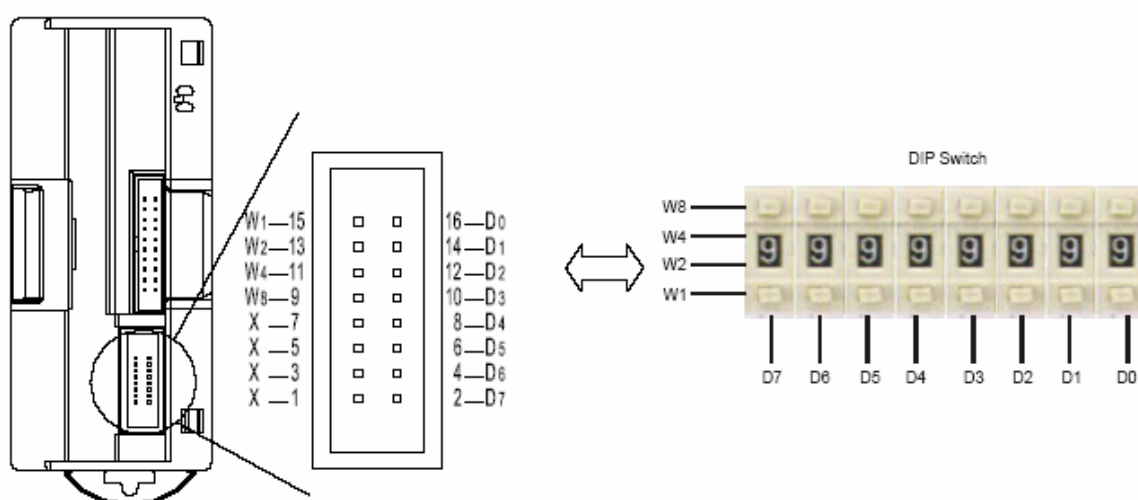
D1139	Число соединенных BCD модулей расширения (макс. 2 модуля)
D1381	Первый BCD-модуль (младший байт)
D1382	Первый BCD-модуль (старший байт)
D1383	Второй BCD-модуль (младший байт)
D1384	Второй BCD-модуль (старший байт)

ПЛК будет считывать состояние DIP-переключателей в каждом скане.

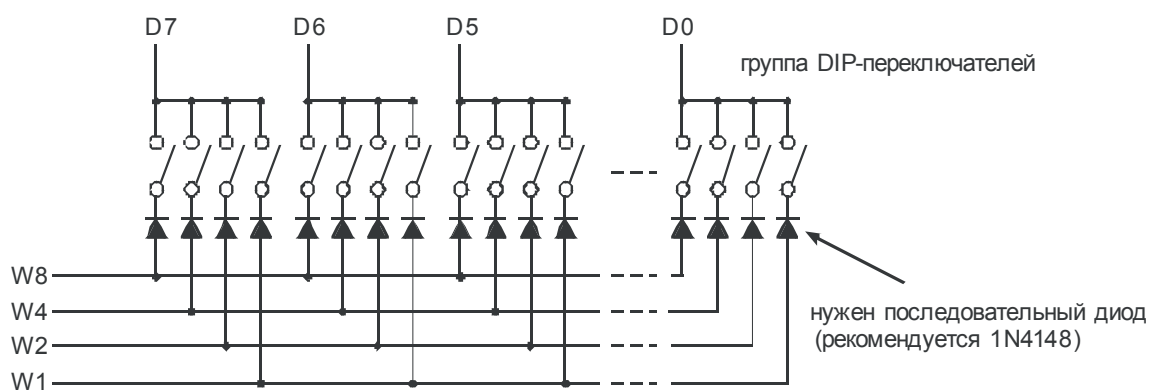
Специальные регистры будут обновлены при переводе ПЛК в состояние РАБОТА.

Максимальное количество BCD/ KEY-модулей подключенных к одному базовому модулю два: или два BCD-модуля, или два KEY-модуля, или один BCD- и один KEY-модуль.

Внешнее соединение BCD-модуля:



Пример соединения BCD-модуля:



2.11.32. KEY-модуль (D1145, D1375 – D1380)

KEY-модули используются в контроллерах серии EN для цифрового ввода с помощью внешней матричной клавиатуры.

D1145	Число соединенных KEY-модулей расширения (макс. 2 модуля)
D1375	Координата X первого KEY-модуля (1 – 8)
D1376	Координата Y первого KEY-модуля (1 – 8)
D1377	Число клавиш первого KEY-модуля (1 – 64)
D1378	Координата X второго KEY-модуля (1 – 8)
D1379	Координата Y второго KEY-модуля (1 – 8)
D1380	Число клавиш второго KEY-модуля (1 – 64)

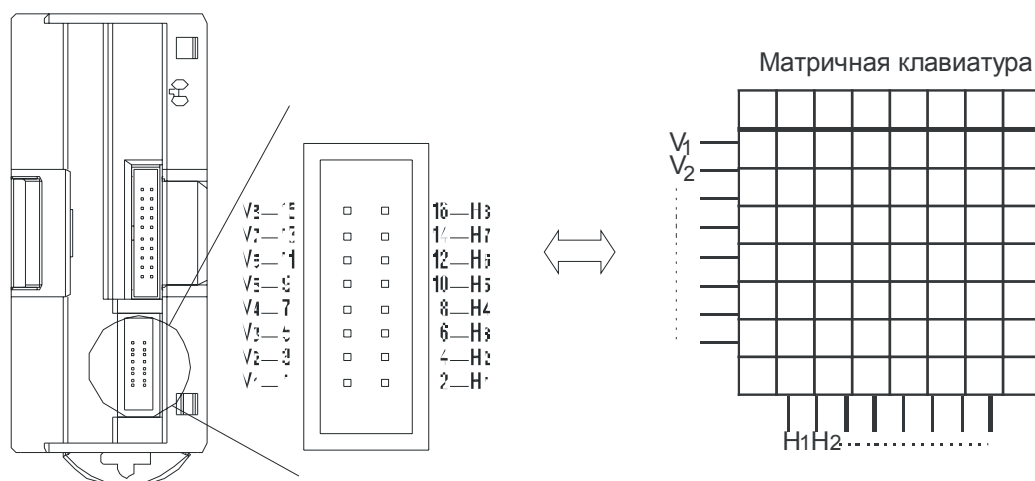
При одновременном нажатии нескольких клавиш считана будет только одна, нажатая первой.

ПЛК будет считывать состояние KEY-модуля в каждом скане.

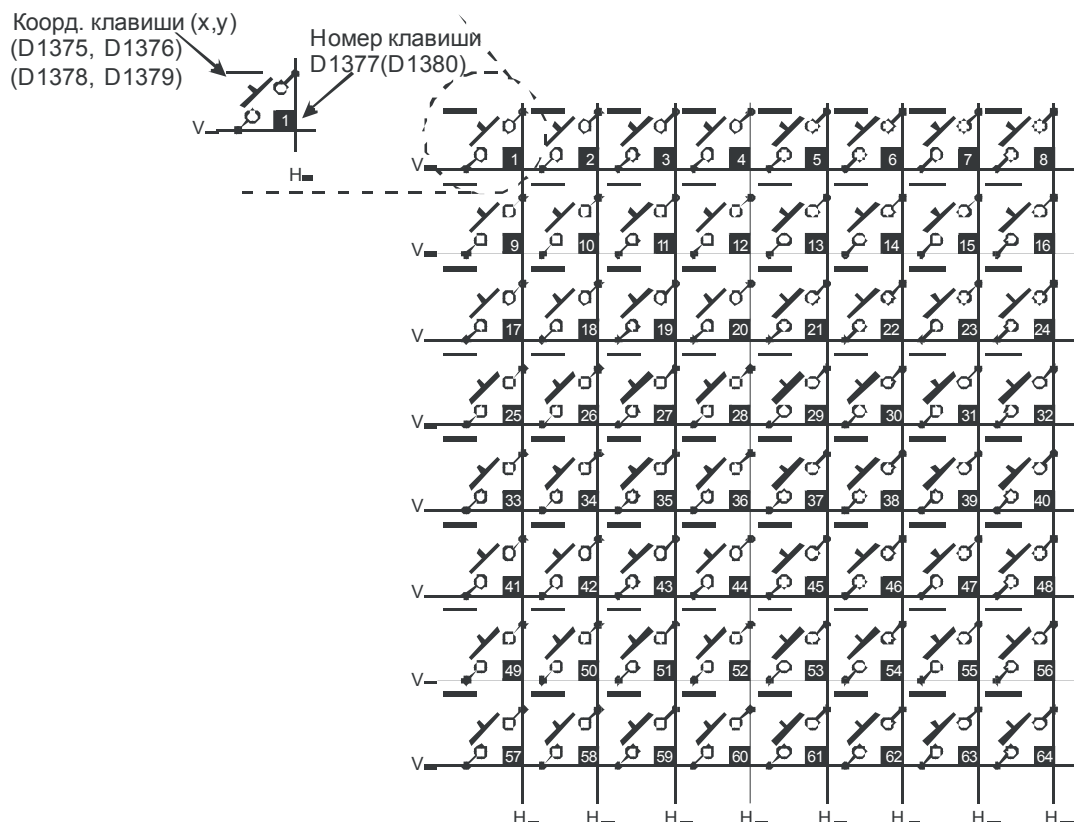
Максимальное количество VCD/ KEY-модулей подключенных к одному базовому модулю два: или два VCD-модуля, или два KEY-модуля, или один VCD- и один KEY-модуль.

Рассчитать номер клавиши по её координатам можно следующим образом: $H+(V-1) \times 8$

Внешнее соединение KEY-модуля:



Пример соединения KEY -модуля:



2.11.33. DISP-модуль (D1146, D1385 – D1393)

DISP-модули используются в контроллерах серии EN для вывода информации на внешний 7-ми сегментный индикатор.

D1146	Число соединенных DISP-модулей расширения (макс. 3 модуля)
D1385	Первый DISP-модуль (младший байт)
D1386	Первый DISP-модуль (старший байт)
D1387	Первый DISP-модуль – десятичная точка
D1388	Второй DISP-модуль (младший байт)
D1389	Второй DISP-модуль (старший байт)
D1390	Второй DISP-модуль – десятичная точка
D1391	Третий DISP-модуль (младший байт)
D1392	Третий DISP-модуль (старший байт)
D1393	Третий DISP-модуль – десятичная точка

Должен использоваться 7-ми сегментный индикатор с общим катодом.

Максимальное количество DISP -модулей подключенных к одному базовому модулю - три, а к каждому DISP-модулю можно подключить по восемь 7-ми сегментных индикаторов.

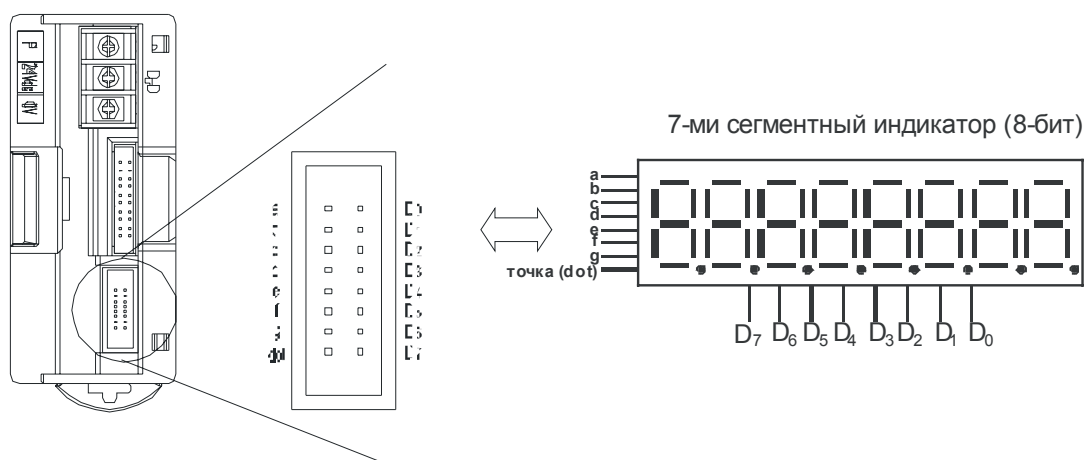
Каждый 7-ми сегментный индикатор использует 4 бита.

Каждый 7-ми сегментный индикатор имеет десятичную точку.

Отображение нуля: эта функция используется, чтобы решить – надо ли отображать ноль. Например: если значения DISP8-DISP1 = 0 1 2 3 4 5 6 7, то ноль в DISP8 отображаться не будет.

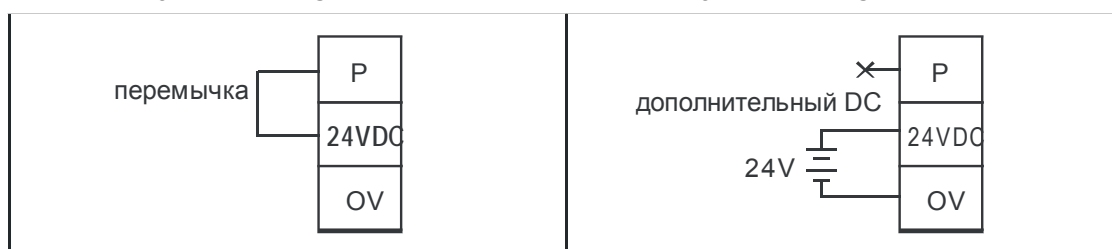
DISP-модуль 1	D1385				D1386				D1387	
бит	b12–b15	b8–b11	b4 – b7	b0 – b3	b12–b15	b8–b11	b4 – b7	b0 – b3	b15–b8	b7 – b0
номер индикатора	4	3	2	1	8	7	6	5	отобр. нолей	точка
значение при STOP->RUN	F	F	F	F	F	F	F	F	0	0

Внешнее соединение DISP-модуля:

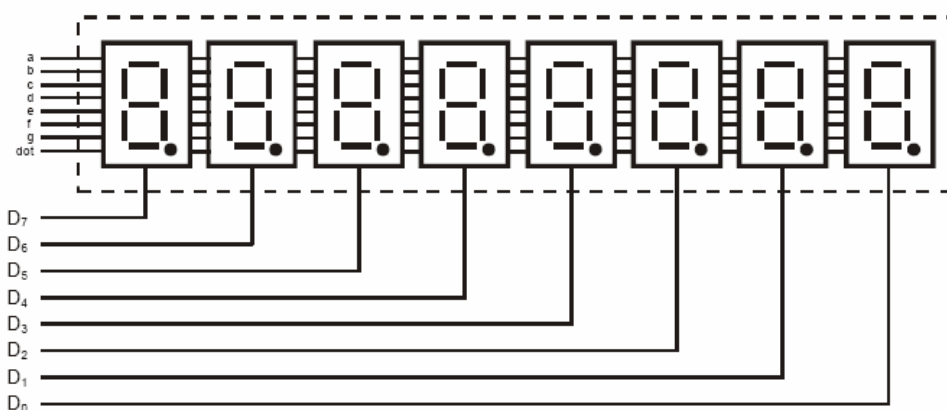


Пример соединения DISP-модуля:

1. Использование внутреннего источника питания +24VDC. 2. Использование внешнего источника питания +24VDC.



7-ми сегментный индикатор с общим катодом:



2.11.34. Описание функции многосекционного импульсного выхода с коррективкой разгона/замедления (M1144 – M1149, M1154, D1032, D1033, D1144, D1154, D1155)

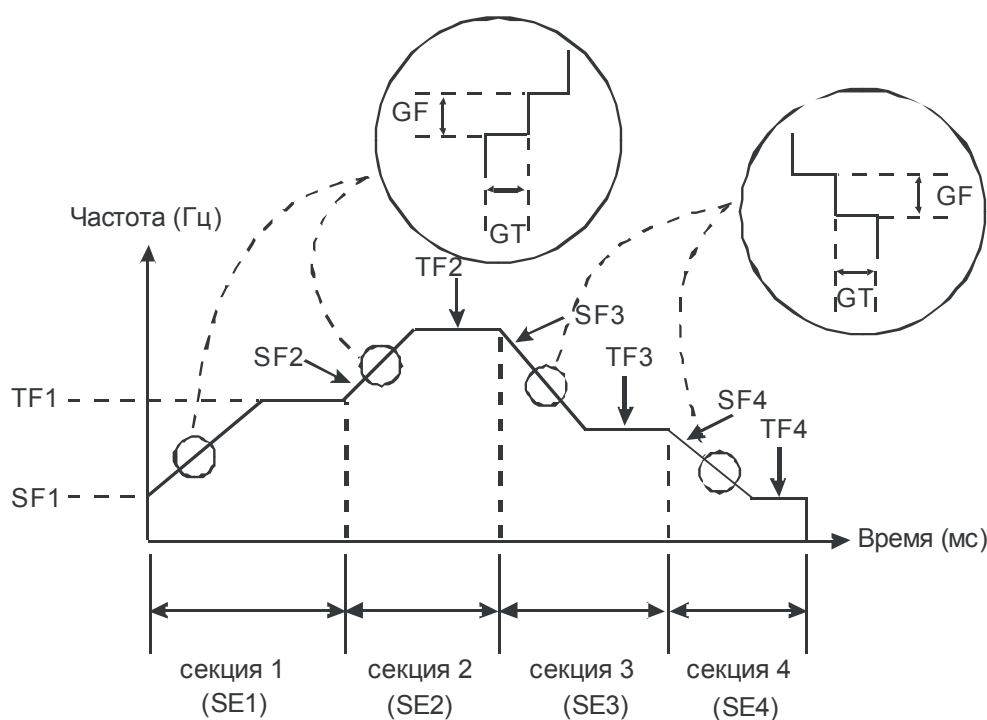
Используется в контроллерах серии SA/SX

M1144	Старт выдачи импульсов с функцией разгона/замедления
M1145	Флаг разгона
M1146	Флаг достижения заданной частоты
M1147	Флаг замедления
M1148	Флаг выполнения функции
M1149	Флаг временной остановки функции разгона/замедления
M1154	Флаг начала назначенного времени интервала шага замедления и частоты
D1032	Количество импульсов на выходе Y1 (младшее слово)
D1033	Количество импульсов на выходе Y1 (старшее слово)
D1144	Указатель регистра D для задания параметров функции разгона/ замедления импульсного выхода Y1
D1154	Рекомендуемое значение назначенного времени интервала шага при замедлении (10...32767 мс) для импульсного выхода с заданием наклона разг./замедл.
D1155	Рекомендуемое значение назначенного частоты интервала шага (-1...-32767 Гц) для импульсного выхода с заданием наклона разг./замедл.

Параметры функции разгона/замедления (D1154)

Индекс	Параметр
+0	Полное число сегментов n (макс. 10)
+1	Номер текущего выполняемого сегмента (только чтение)
+2	Начальная частота первого сегмента (SF1)
+3	Время шага ускорения / замедления первого сегмента (GT1)
+4	Частота шага ускорения / замедления первого сегмента (GF1)
+5	Заданная частота первого сегмента (TF1)
+6	Полное число импульсов первого сегмента (младшие 16 бит)
+7	Полное число импульсов первого сегмента (старшие 16 бит)
+8	Начальная частота второго сегмента (SF2)
+9	Время шага ускорения / замедления второго сегмента (GT2)
+10	Частота шага ускорения / замедления второго сегмента (GF2)
+11	Заданная частота второго сегмента (TF2)
+12	Полное число импульсов второго сегмента (младшие 16 бит)
+13	Полное число импульсов второго сегмента (старшие 16 бит)
...	...
+6n+2	Начальная частота n-го сегмента (SFn)
+6n+3	Время шага ускорения / замедления n-го сегмента (GTn)
+6n+4	Частота шага ускорения / замедления n-го сегмента (GFn)
+6n+5	Заданная частота n-го сегмента (TFn)
+6n+6	Полное число импульсов n-го сегмента (младшие 16 бит)
+6n+7	Полное число импульсов n-го сегмента (старшие 16 бит)

Функция импульсного выхода с разгоном/замедлением может быть использована только с выходом Y1. Нужно заполнить параметры функции и включить контакт M1144 и на выходе Y1 начнется генерация импульсов как показано ниже:



Правила и ограничения использования функции:

- Минимальные значения начальной и заданной частот не должны быть меньше 200 Гц;
- Максимальные значения начальной и заданной частот не должны быть больше 32700 Гц;
- Время интервала шага должно быть в диапазоне 1 ... 32767 мс;
- Частота интервала шага при разгоне должна быть в диапазоне 1 ... 32760 Гц, а при торможении -1 ... -32760 Гц;
- Заданное число импульсов в сегменте должен быть больше чем $((GF*GT/1000)*((TF-SF)/GF))$, см. пример 1 для пояснения;
- Если выход Y1 используется командой высокоскоростного выхода, то функция "импульсный выход с разгоном/замедлением" не может быть выполнена.
- Если функция до конца не выполнена, реле M1148 = 0, и состояния реле M1144 изменить на выключенное, то будет выполняться функция замедления. Если M1354=0, замедление будет происходить со скоростью 200 Гц за 200 мс до частоты 200 Гц и M1147 будет = 1. Если M1354=1, замедление будет происходить с интервалами скорости и времени, заданными в параметрах D1154, D1155.
- Если функция выполнена до конца, реле M1148 = 1, и состояния реле M1144 изменить на выключенное, то функция замедления не будет выполняться и M1148 сбросится на 0.
- Максимальное количество сегментов функции равно 10.
- Когда заданная частота выполняемого сегмента меньше чем начальная частота следующего сегмента, будет выполняться ускорение и заданная частота следующего сегмента должна быть больше, чем начальная частота. Когда заданная частота выполняемого сегмента больше чем начальная частота

следующего сегмента, будет выполняться замедление и следовательно заданная частота следующего сегмента должна быть меньше, чем начальная частота. Если эти моменты не будут учтены, функция может работать не корректно.

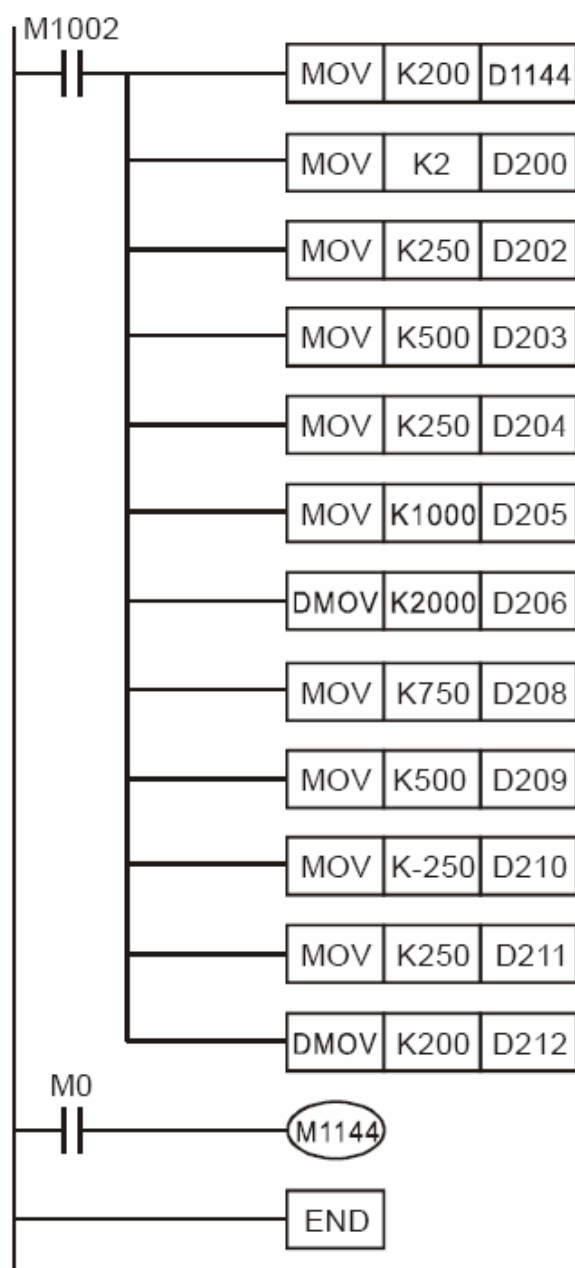
- При изменении состояния ПЛК: СТОП -> РАБОТА все реле M1144 – M1149 будут установлены в ноль. При изменении состояния ПЛК: РАБОТА -> СТОП реле M1144 будет установлено в ноль, а M1145– M1149 не изменят свое состояние. D1104 будет = 0 при выключении ПЛК, но не изменит свое значение в других случаях.
- Используйте параметры в диапазоне D0 – D999 и D2000 – D4999.

Пример1 расчета времени и числа импульсов разгона/замедления в каждом сегменте и заданной частоты:

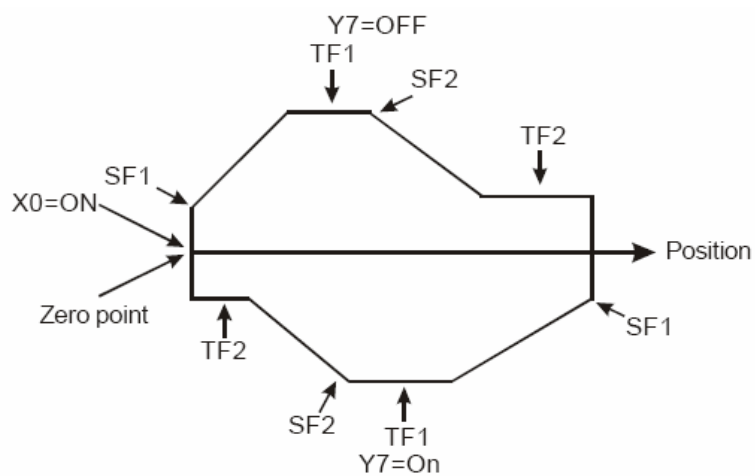
Допустим начальная частота в сегменте = 200 Гц, заданная частота = 500 Гц, частота шага = 100 Гц, полное число импульсов в сегменте = 1000.

- Число выходных импульсов при старте разгона/замедления = $200 \times 100 / 1000 = 20$
- Число выходных импульсов в первом интервале разгона = $300 \times 100 / 1000 = 30$
- Число выходных импульсов во втором интервале разгона = $400 \times 100 / 1000 = 40$
- Число выходных импульсов при заданной частоте = $1000 - (40 + 20 + 30) = 910$
(примечание: рекомендуется устанавливать это число > 10)
- Время работы при заданной частоте = $1 / 500 \times 910 = 1820$ мс
- Полное время сегмента = $1820 + 3 \times 100 = 2120$ мс.

Пример2: простая программа импульсного выхода с сегментом разгона и сегментом замедления.



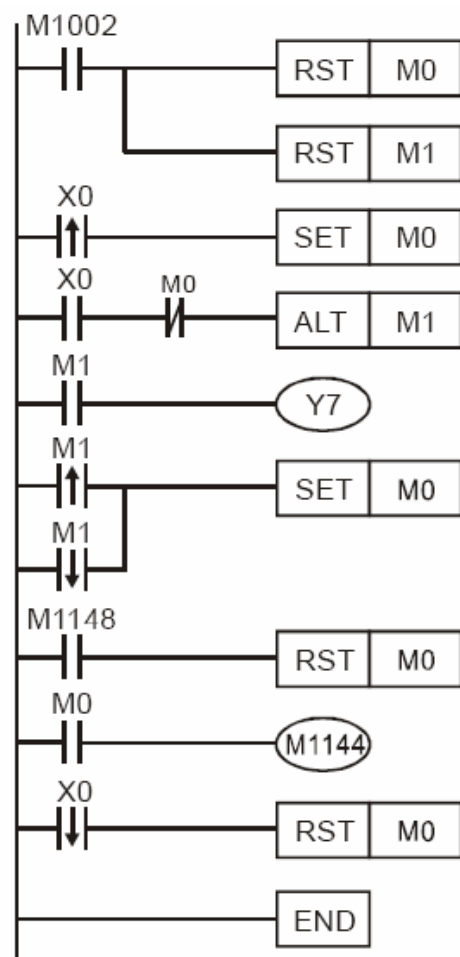
Пример3: программа импульсного выхода изменением направления.



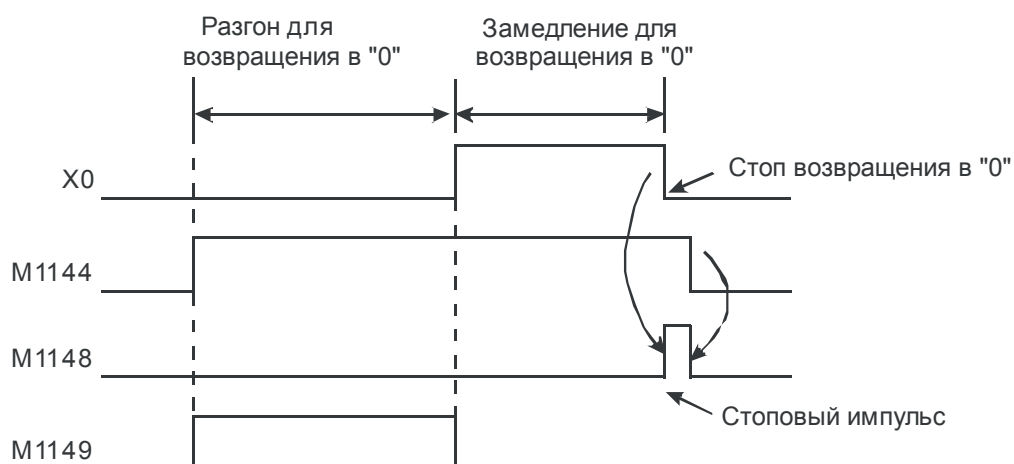
Пояснения:

- Разгон/замедление установлены, как в примере 2
- График показывает пример позиционного перемещения. Когда X0=1, начнется движение, и когда X0=0, произойдет останов. Y7 выбирает направление перемещения.

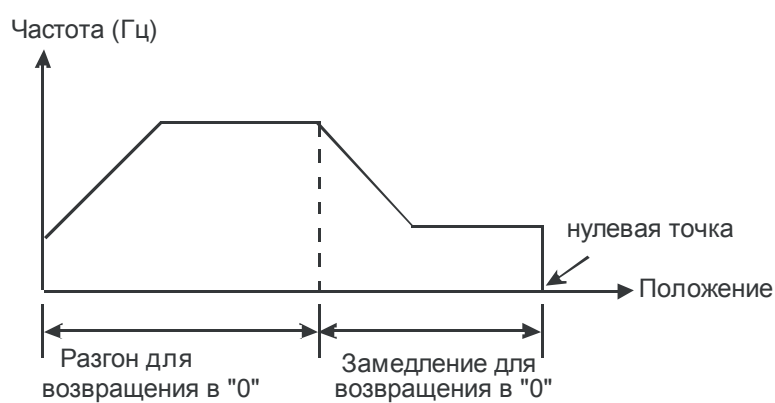
Программа:



Пример4: применение сегментов разгона и замедления в программе возвращения в нулевую точку.



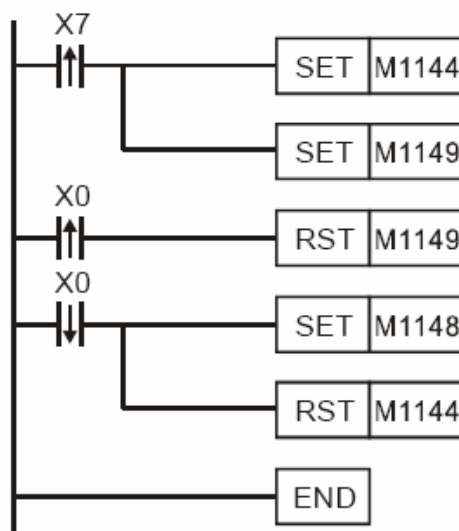
Зависимость частоты от положения показана ниже:



Установленные параметры (по адресу регистра D):

Индекс	Параметр
+0	2
+2	250 Гц
+3	100 мс
+4	500 Гц
+5	10000 Гц
+6	10 импульсов
+7	10 импульсов
+8	9750 Гц
+9	50 мс
+10	-500 Гц
+11	250 Гц
+12	30000 импульсов
+13	30000 импульсов

Программа:



Пояснения:

- Передний фронт на X7, установит M1144=1 и даст старт разгону, M1144=1 не будет считать число импульсов. Будут переданы 10 импульсов и передний фронт X0, запустит замедление первого сегмента. Задним фронтом X0 можно вручную завершить выполнение функции.

2.11.35. Функция пошагового выполнения программы (M1170, M1171, D1170)

M1170	Старт функции пошагового выполнения программы
M1171	Пуск выполнения одного шага программы
D1170	Номер текущего шага программы

Пояснение функции:

- Эта функция действительна только в режиме РАБОТА
- Принцип действия: реле M1170 запускает режим пошагового выполнения программы, т.е. каждый следующий шаг программы выполняется только при включении M1171, а когда шаг будет выполнен, M1171 будет=0 и программа будет ожидать следующего включения M1171. В регистре D1170 будет отображаться номер текущего шага программы.
- Примечание: Команды импульсных входов/выходов, высокоскоростного сравнения и команда НКУ не будут работать в режиме пошагового выполнения программы.

2.11.36. Функция двухфазного импульсного выхода (M1172 – M1174, D1172 – D1177)

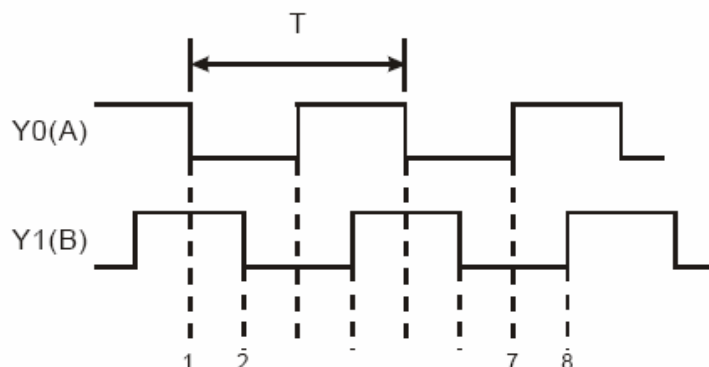
Используется в контроллерах серии SA/SX

M1172	Старт 2-х фазного импульсного выхода (M1172=ON – старт)
M1173	Когда M1173=ON импульсы идут непрерывно.
M1174	Флаг достижения заданного количества выходных импульсов
D1172	Частота двухфазного импульсного выхода (12 Гц – 20 кГц)
D1173	Выбор режима двухфазного импульсного выхода (K1 и K2)
D1174	Заданное значение для двухфазного импульсного выхода (младшее слово)

D1175	Заданное значение для двухфазного импульсного выхода (старшее слово)
D1176	Текущее значение на двухфазном импульсном выходе (младшее слово)
D1177	Текущее значение на двухфазном импульсном выходе (старшее слово)

Пояснение функции:

- Выходная частота = $1/T$ (см. рис.). На рис. показано 8 выходных импульсов.
- В режиме K1 фаза А опережает фазу В, а в режиме K2 фаза В опережает фазу А.
- Когда количество импульсов достигнет заданного значения, M1174 будет = 1, а M1172 сбросится на ноль.



Выходная частота, заданное число импульсов и режим могут быть изменены при M1172=1 и M1174=0, но эти изменения не будут действительны для текущего пакета импульсов за исключением изменения режима, который сбросит текущее значение импульсов в ноль. Текущее значение импульсов будет обновляться в каждом скане и будет обнулено при переключении M1172 из 0 в 1.

Примечание: эта функция может быть использована только в режиме РАБОТА и может выполняться одновременно с командой PLSY.

2.11.37. Значение потенциометра VR (M1178 – M1179, D1178 - D1179)

Используются в контроллерах SA/EH для работы со встроенными потенциометрами.

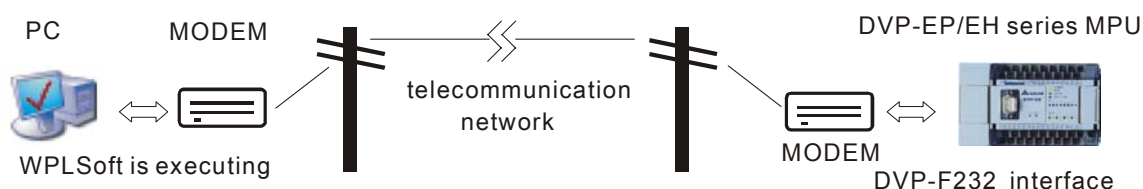
M1178	Старт VR0
M1179	Старт VR1
D1178	Значение VR0
D1179	Значение VR1

Эта функция может быть использована только в режиме РАБОТА. Когда M1178=1, аналоговый сигнал от встроенного переменного резистора VR1 будет оцифровываться (1 - 255) и сохраняться в D1178. Аналогично будет работать и второй потенциометр.

Для детализации см. описание инструкции VRRD (API 85).

2.11.38. Функции модемного соединения (M1184~M1188)

1. Система соединения:



2. Специальные реле:

Реле	Функция	Пояснение
M1184	Запуск модема (функции MODEM)	Когда M1184=1, нижеследующее действие возможно.
M1185	Запуск инициализации модема	Этот флаг будет сброшен после завершения инициализации
M1186	Ошибка инициализации модема	Когда M1185=1, M1186=0.
M1187	Инициализация модема выполнена	Когда M1185=1, M1187=0.
M1188	Индикация состояния соединения модема	Есть соединение

Специальные реле всегда действительны независимо от состояния ПЛК: RUN или STOP.

3. Последовательность операций:

- a) Установите M1184=1 со стороны ПЛК (запуск MODEMa)
 - b) Установите M1185=1 (запуск инициализации MODEMa)
 - c) Проверьте результат инициализации MODEMa: M1186=1 при успешной инициализации. M1187=1 при ошибке инициализации.
 - d) После успешного завершения инициализации, WPLSoft на удаленном ПК может быть готов к связи. WPL метод соединения: установка -> связь с модемом (должен быть установлен драйвер для модема) -> вызовите диалог набора номера и заполните соответствующей информацией.
1. Внимание:
- a) Используйте RS-232 карту для связи с MODEMом со стороны ПЛК. Если нет, вышеприведенные специальные реле M будут не работоспособны.
 - b) Вы должны установить M1185=1 для инициализации MODEMa после запуска MODEMa (M1184=1). Если нет, MODEM не сможет быть автоматически запущен со стороны ПЛК.
 - c) MODEM войдет в режим автодозвона после инициализации.
 - d) MODEM войдет в режим готовности со стороны ПЛК после разрыва связи со стороны ПК. Если пользователь выключит MODEM из сети, будет нужна новая инициализация после его следующего включения.
 - e) Скорость связи модема с ПЛК фиксированная 9600bps и не может быть изменена.
 - f) Исходный формат MODEMa со стороны ПЛК: ATZ и ATS0=1.

Если ПЛК не может определить модем при инициализации, попробуйте использовать Hyper Terminal на ПК для исходного формата ATZ и ATS0=1.

2.11.39. Установка диапазона энергонезависимых реле и регистров (D1200 – D1219)

См. главу 2.1

2.11.40. Разрешение управления входами X от WPLSoft или программатора (M1304)

В контроллерах SA/SX/EH, когда M1304=1, становится возможным включение/выключение входов X базового модуля с помощью WPLSoft в режиме отладки программы или с программатора HPP.

2.11.41. Номера специальных модулей расширения (D1320 – D1327)

В контроллерах EH в регистрах D1320 – D1327 отображаются идентификационные коды (ID) всех подключенных в данный момент специальных модулей расширения.

Модуль расширения	Идентификационный код (ID)	Модуль расширения	Идентификационный код (ID)
DVPEH04AD	H'0400	DVPEH01PU	H'0110
DVP04DA-H	H'0401	DVPEH01HC	H'0120
DVPEH04PT	H'0402	DVPEH02HC	H'0220
DVPEH04TC	H'0403	DVPEH01DT	H'0130
DVPEH06XA	H'0604	DVPEH02DT	H'0230
DVPEH06RT	H'0405		

2.11.42. Параметры режима PLC Link (M1350 – M1354, M1360 – M1519, D1355 – D1370, D1415 – D1465, D1480 – D1491)

Режим PLC Link используется для связи и обмена данными между базовыми модулями ПЛК серии EH/SA/SX.

1. M1353=0 (для DVP-EH/SA/SX): один мастер и от 1-го до 16-ти ведомых.

Ведущий ПЛК (Master PLC) : M1353=0															
Ведомый 1		Ведомый 2		Ведомый 3		Ведомый 4		Ведомый 5		Ведомый 6		Ведомый 7		Ведомый 8	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D1480	D1496	D1512	D1528	D1544	D1560	D1576	D1592	D1608	D1624	D1640	D1656	D1672	D1688	D1704	D1720
...
D1495	D1511	D1527	D1543	D1559	D1575	D1591	D1607	D1623	D1639	D1655	D1671	D1687	D1703	D1719	D1735
Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Коммуникационный адрес для чтения данных от ПЛК															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
Обнаружение наличия ведомого ПЛК в сети															
M1360		M1361		M1362		M1363		M1364		M1365		M1366		M1367	
Флаг - идет передача данных															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	

[illegible]

Заводская установка коммуникационного адреса для чтения данных – H1064 (D100).

Заводская установка коммуникационного адреса для записи данных – H10C8 (D200).

Ведущий ПЛК (Master PLC) : M1353=0															
Ведомый 9		Ведомый 10		Ведомый 11		Ведомый 12		Ведомый 13		Ведомый 14		Ведомый 15		Ведомый 16	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D1736	D1752	D1768	D1784	D1800	D1816	D1832	D1848	D1864	D1880	D1896	D1912	D1928	D1944	D1960	D1976
...
D1751	D1767	D1783	D1799	D1815	D1831	D1847	D1863	D1879	D1895	D1911	D1927	D1943	D1959	D1975	D1991
Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
Коммуникационный адрес для чтения данных от ПЛК															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
Обнаружение наличия ведомого ПЛК в сети															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Флаг - идет передача данных															
M1384		M1385		M1386		M1387		M1388		M1389		M1390		M1391	
Флаг ошибки чтения / записи данных															
M1400		M1401		M1402		M1403		M1404		M1405		M1406		M1407	

[illegible]

Заводская установка коммуникационного адреса для чтения данных – H1064 (D100).

Заводская установка коммуникационного адреса для записи данных – H10C8 (D200).

1. M1353=1(только DVP-ЕН): один мастер и от 1-го до 32-х ведомых.

Ведущий ПЛК (Master PLC): M1353=1															
Ведомый 1		Ведомый 2		Ведомый 3		Ведомый 4		Ведомый 5		Ведомый 6		Ведомый 7		Ведомый 8	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D1480	D1496	D1481	D1497	D1482	D1498	D1483	D1499	D1484	D1500	D1485	D1501	D1486	D1502	D1487	D1503
Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Коммуникационный адрес для чтения данных от ПЛК															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
Обнаружение наличия ведомого ПЛК в сети															
M1360		M1361		M1362		M1363		M1364		M1365		M1366		M1367	
Флаг - идет передача данных															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	
Флаг ошибки чтения / записи данных															
M1392		M1393		M1394		M1395		M1396		M1397		M1398		M1399	
Флаг – чтение завершено															
M1408		M1409		M1410		M1411		M1412		M1413		M1414		M1415	
Флаг – запись завершена															

[illegible]

Заводская установка коммуникационного адреса для чтения данных – H1064 (D100).

Заводская установка коммуникационного адреса для записи данных – H10C8 (D200).

Ведущий ПЛК (Master PLC) : M1353=1															
Ведомый 9		Ведомый 10		Ведомый 11		Ведомый 12		Ведомый 13		Ведомый 14		Ведомый 15		Ведомый 16	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D1488	D1504	D1489	D1505	D1490	D1506	D1491	D1507	D1492	D1508	D1493	D1509	D1494	D1510	D1495	D1511
Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
Коммуникационный адрес для чтения данных от ПЛК															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
Обнаружение наличия ведомого ПЛК в сети															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Флаг - идет передача данных															
M1384		M1385		M1386		M1387		M1388		M1389		M1390		M1391	
Флаг ошибки чтения / записи данных															
M1400		M1401		M1402		M1403		M1404		M1405		M1406		M1407	
Флаг – чтение завершено															
M1416		M1417		M1418		M1419		M1420		M1421		M1422		M1423	
Флаг – запись завершена															
M1432		M1433		M1434		M1435		M1436		M1437		M1438		M1439	
↓		↓		↓		↓		↓		↓		↓		↓	
Ведомый 1		Ведомый 2		Ведомый 3		Ведомый 4		Ведомый 5		Ведомый 6		Ведомый 7		Ведомый 8	

Заводская установка коммуникационного адреса для записи данных – H10C8 (D200).

[illegible]

Заводская установка коммуникационного адреса для чтения данных – H1064 (D100).

Заводская установка коммуникационного адреса для записи данных – H10C8 (D200).

Ведущий ПЛК (Master PLC) : M1353=1															
Ведомый 25		Ведомый 26		Ведомый 27		Ведомый 28		Ведомый 29		Ведомый 30		Ведомый 31		Ведомый 32	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D1584	D1600	D1585	D1601	D1586	D1602	D1587	D1603	D1588	D1604	D1589	D1605	D1590	D1606	D1591	D1607
Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в	Число рег-в
D1552	D1568	D1553	D1569	D1554	D1570	D1555	D1571	D1556	D1572	D1557	D1573	D1558	D1574	D1559	D1575
Коммуникационный адрес для чтения данных от ПЛК															
D1520	D1536	D1521	D1537	D1522	D1538	D1523	D1539	D1524	D1540	D1525	D1541	D1526	D1542	D1527	D1543
Обнаружение наличия ведомого ПЛК в сети															
M1448		M1449		M1450		M1451		M1452		M1453		M1454		M1455	
Флаг - идет передача данных															
M1464		M1465		M1466		M1467		M1468		M1469		M1470		M1471	
Флаг ошибки чтения / записи данных															
M1480		M1481		M1482		M1483		M1484		M1485		M1486		M1487	
Флаг – чтение завершено															
M1496		M1497		M1498		M1499		M1500		M1501		M1502		M1503	
Флаг – запись завершена															
M1512		M1513		M1514		M1515		M1516		M1517		M1518		M1519	
↓		↓		↓		↓		↓		↓		↓		↓	
Ведомый 1		Ведомый 2		Ведомый 3		Ведомый 4		Ведомый 5		Ведомый 6		Ведомый 7		Ведомый 8	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200
...
D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215

Заводская установка коммуникационного адреса для чтения данных – H1064 (D100).

Заводская установка коммуникационного адреса для записи данных – H10C8 (D200).

Пояснения:

- Базовый коммуникационный протокол для режима PLC Link – MODBUS
- Все ПЛК в сети должны иметь одинаковый коммуникационный формат (D1120) и режим ASCII или RTU.
- Один ведущий ПЛК (серия SA/SX) может иметь до 16-х ведомых ПЛК, а один ведущий ПЛК (серия EH) может иметь до 32-х ведомых. Для работы с более, чем 16-ю ведомыми ПЛК в ведущем ПЛК должно быть включено специальное реле M1353=1.
- Адреса (ID) в сети не должны повторяться, каждый ведомый ПЛК должен иметь уникальный ID (1 - 32).
- Для связи одного ведущего ПЛК с одним ведомым можно использовать интерфейсы: RS-232, RS-485, RS-422.
- Для связи одного ведущего ПЛК с несколькими ведомыми можно использовать только интерфейс RS-485.

Работа:

- Установите протоколы используемых коммуникационных портов: COM1 (RS-232): D1036; COM2 (RS-232/RS-485/RS-422): D1120; COM3 (RS-232/RS-485): D1109. Скорость обмена и формат передачи данных у ведомых ПЛК должны быть как у ведущего Master-PLC.
- Установите адреса (ID) ведущего и ведомых ПЛК в регистре D1121. Адреса не должны повторяться.
- Установите число регистров для чтения/записи данных от ведомых ПЛК (макс. 16 – когда M1353=0; макс.100 - когда M1353=1) в D1480~D1495/D1576~D1591 для сохранения после чтения и D1496~D1511/D1592~1607 для сохранения после записи.
- Установите коммуникационные адреса регистров в ведомых ПЛК: заводская установка для адресов чтения - H1064 (D100), для адресов записи - H10C8 (D200).
- Разрешение функции одновременного чтения/записи (M1354=1)
- Установка автоматического режима связи PLC Link (M1351=1).
- Установка ручного режима связи PLC Link (M1352=1).
- Старт режима PLC LINK (M1350=1).

Пояснение работы режима PLC LINK (ведущий ПЛК):

- Когда M1350=1, ведущий ПЛК начинает детектирование подключенных ведомых ПЛК, и записывает их количество в D1433.
- Адреса обнаруженных ведомых ПЛК можно увидеть в M1360-M1375 и M1440-M1455.
- Если не обнаружено ни одного подключенного ведомого ПЛК, M1350 будет=0 и связь будет остановлена.

- M1353 и M1354 должны быть установлены до PLC EASY LINK.
- Чтение/запись от ведущего и ведомого ПЛК: после завершения детектирования ведомых ПЛК начнется обмен данными между ведущим и ведомыми ПЛК. Если ведомый ПЛК появится в сети после детектирования, связь с ним не будет установлена, необходимо новое детектирование.
- Ведущий ПЛК будет читать состояния 16-ти регистров (начиная с D100) и записывать данные в регистры, начиная с D200 каждого ведомого ПЛК.
- Ведущий ПЛК будет читать / писать данные в ведомые ПЛК по порядку, т.е. после ID1 будет чтение/запись в ID2 и т.д.
- Если число читаемых/записываемых регистров больше 16-ти, установите D1129>500мс для предотвращения коммуникационного тайм-аута.

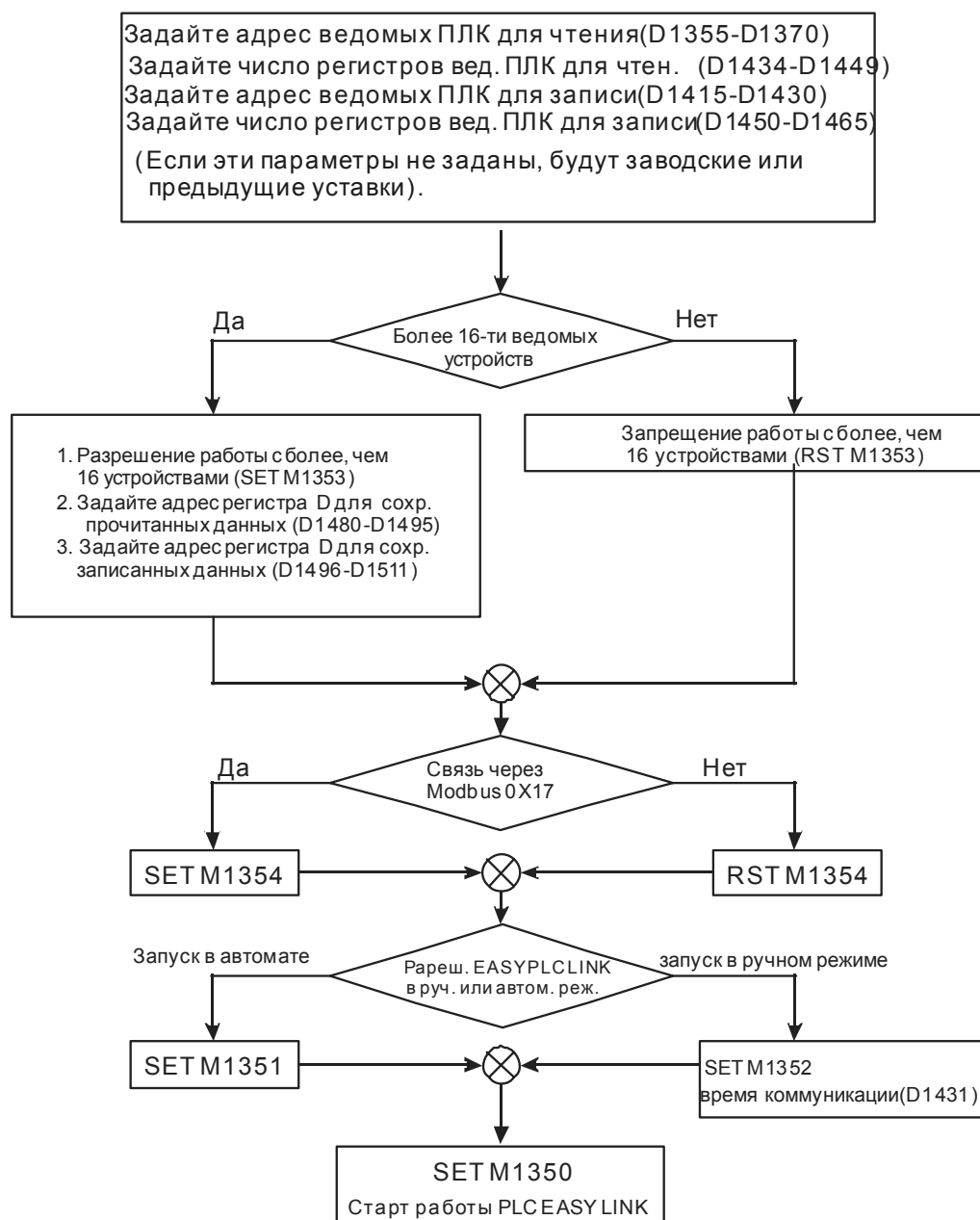
Автоматический и ручной режимы связи:

- В автоматическом режиме (M1351=0) ведущий будет читать/писать данные до тех пор, пока M1350 не будет выключен.
- В ручном режиме (M1351=1) ведущий будет читать/писать данные в течении времени заданном в D1432, т.е. когда после старта режима D1431 будет = D1432, связь прекратится и для её возобновления надо будет переключить M1351.

Примечания:

- Автоматический режим (M1351) и ручной режим (M1352) не могут быть включены одновременно.
- Перед тем как переключить режим ручной/автоматический, сначала надо установить M1350=0.
- Время ожидания ответа может быть установлено в D1129 в диапазоне 300 ... 3000 мс.
- Функция PLC LINK возможна только при скорости обмена больше 1200 бит/с. Когда скорость обмена меньше 9600 бит/с, время ожидания ответа надо установить не меньше 1 сек.
- Связь не работает, когда число читаемых/записываемых регистров = 0.

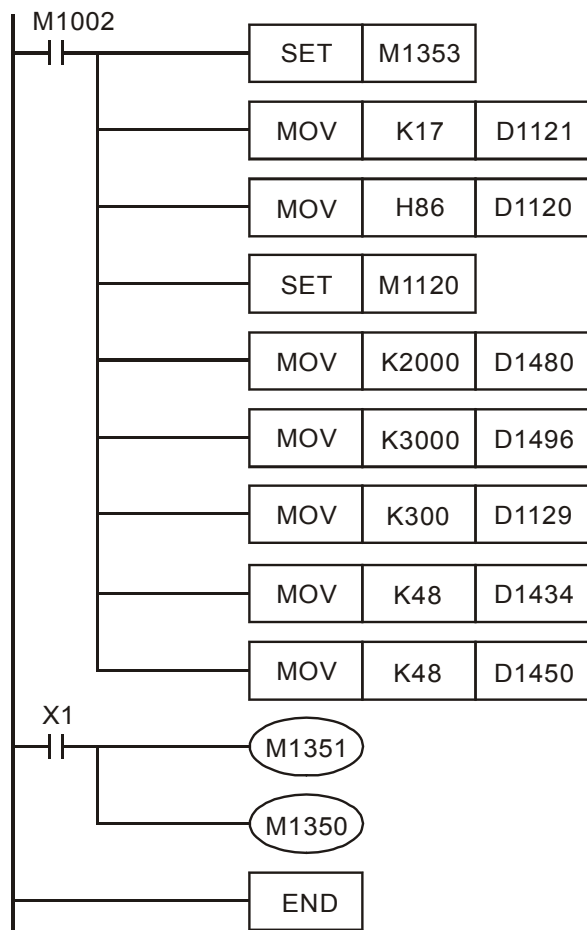
Алгоритм настройки режима PLC EASY Link.

**Пример 1: Режим PLC EASY LINK с M1353=1**

1. M1353 должно быть установлено до старта режима PLC EASY LINK.
2. D1480~D1495 - стартовые адреса регистров D для чтения в ведущем ПЛК и D1496~D1511 - стартовые адреса регистров D для записи в ведущем ПЛК. Только регистры D могут использоваться для этих функций и каждый D указывает на один ведомый ПЛК, так, например, D1480 указывает ID1, D1481 указывает ID2 и т.д.
3. Вы можете установить D1480~D1495 и D1496~D1511 до разрешения PLC EASY LINK.
4. Диапазон D1480~D1495 и D1496~D1511: от 0 до 9900. При выходе из диапазона будет установлено 9900.
5. D1434~D1449 - количество читаемых регистров (D1434 - количество читаемых регистров для ID1) и D1450~D1465 - количество записываемых регистров (D1450 -

количество записываемых регистров для ID1). Когда выполняется PLC EASY LINK при M1353=0, количество регистров может быть от 0 до 16. Когда M1353=1, количество регистров может быть от 0 до 48.

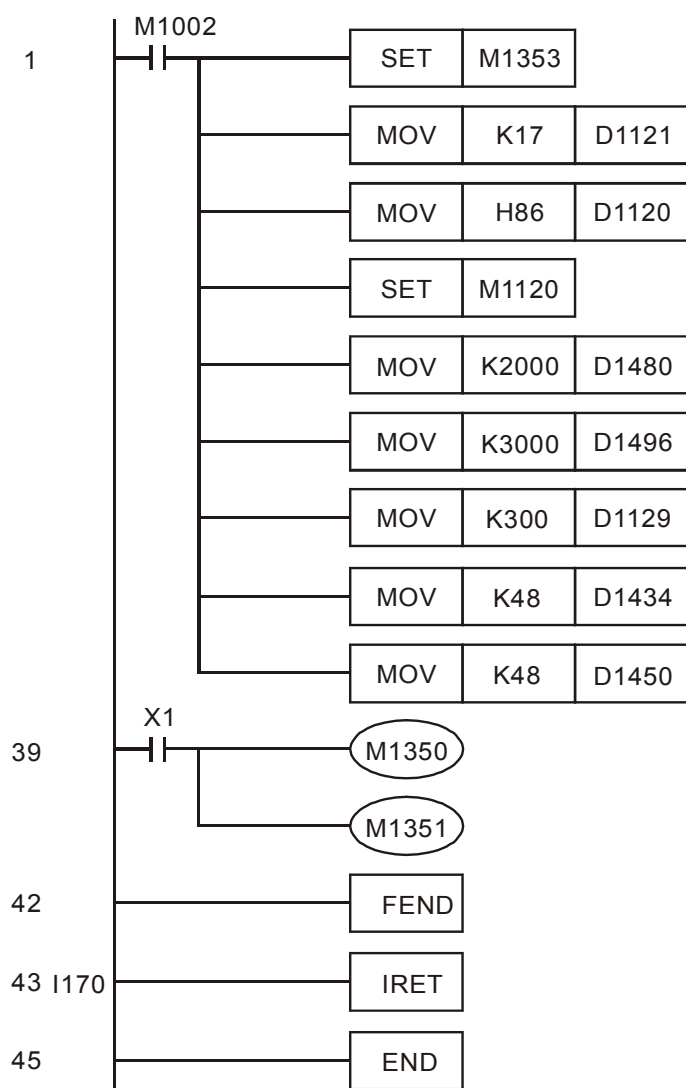
6. Вы можете установить D1434~D1449 и D1450~D1465 после разрешения PLC EASY LINK, но изменения вступят в силу только при следующем детектировании.



Пример 2: Режим PLC EASY LINK с M1353=1 и прерыванием I170

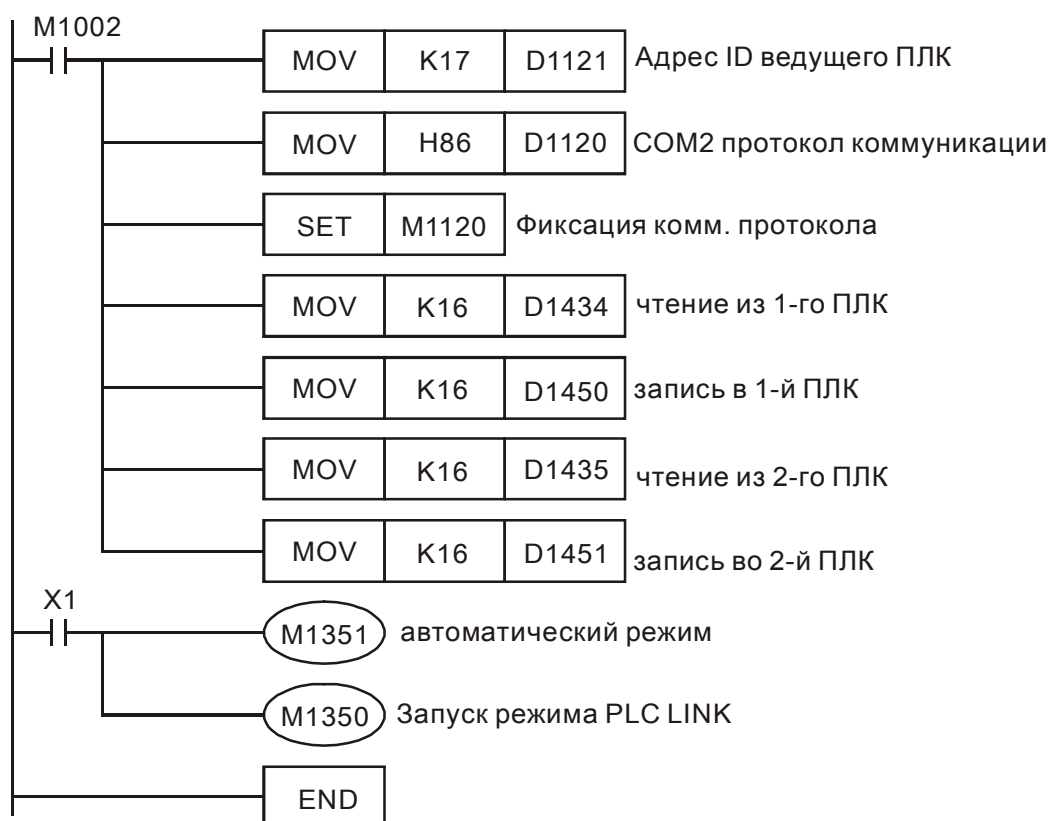
1. Сразу после завершения приема данных начнется обработка прерывания I170.
2. Не рекомендуется использовать функцию прерывания I170 при низких скоростях отклика ведомого ПЛК.
3. Дополнительный специальный регистр (D1399) для PLC EASY LINK определяет начальный адрес ID ведомых ПЛК.

Например, если D1399 = k20, ведущий ПЛК будет детектировать ID от 20 до 35.

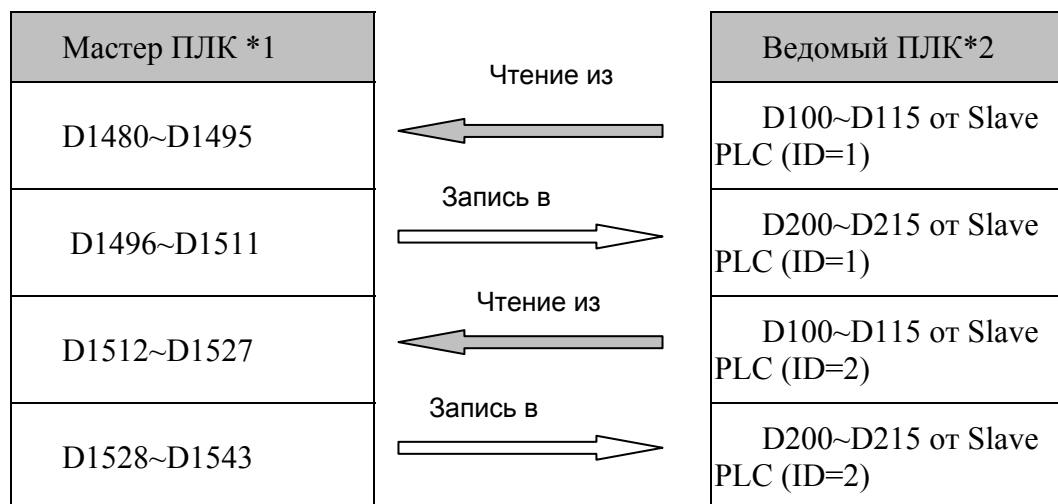


Пример 3: 3 ПЛК соединены по RS-485, и используется режим PLC LINK для обмена 16 регистрами данных между 1 ведущим и 2 ведомыми ПЛК. (M1353=0, 16 PLC LINK)

Ведущий PLC (ID=17)



Когда X1=ВКЛ, будет происходить автоматический обмен данными между ведущим ПЛК и двумя ведомыми. Данные D100~D115 во 2 –м ведомом ПЛК будут читаться в D1480~D1495 и D1512~D1527 ведущего ПЛК. Так же данные D1496~D1511 и D1528~D1543 ведущего ПЛК будут записываться в D200~D215 ведомого ПЛК 1.



До активизации режима PLC LINK (M1350=0), допускается изменение данных, как показано ниже:

Мастер ПЛК	Заданное значение	Ведомый ПЛК	Заданное значение
D1480~D1495	K0 для всех	D100~D115 из Slave PLC (ID=1)	K5000 для всех
D1496~D1511	K1000 для всех	D200~D215 из Slave PLC (ID=1)	K0 для всех
D1512~D1527	K0 для всех	D100~D115 из Slave PLC (ID=2)	K6000 для всех
D1528~D1543	K2000 для всех	D200~D215 из Slave PLC (ID=2)	K0 для всех

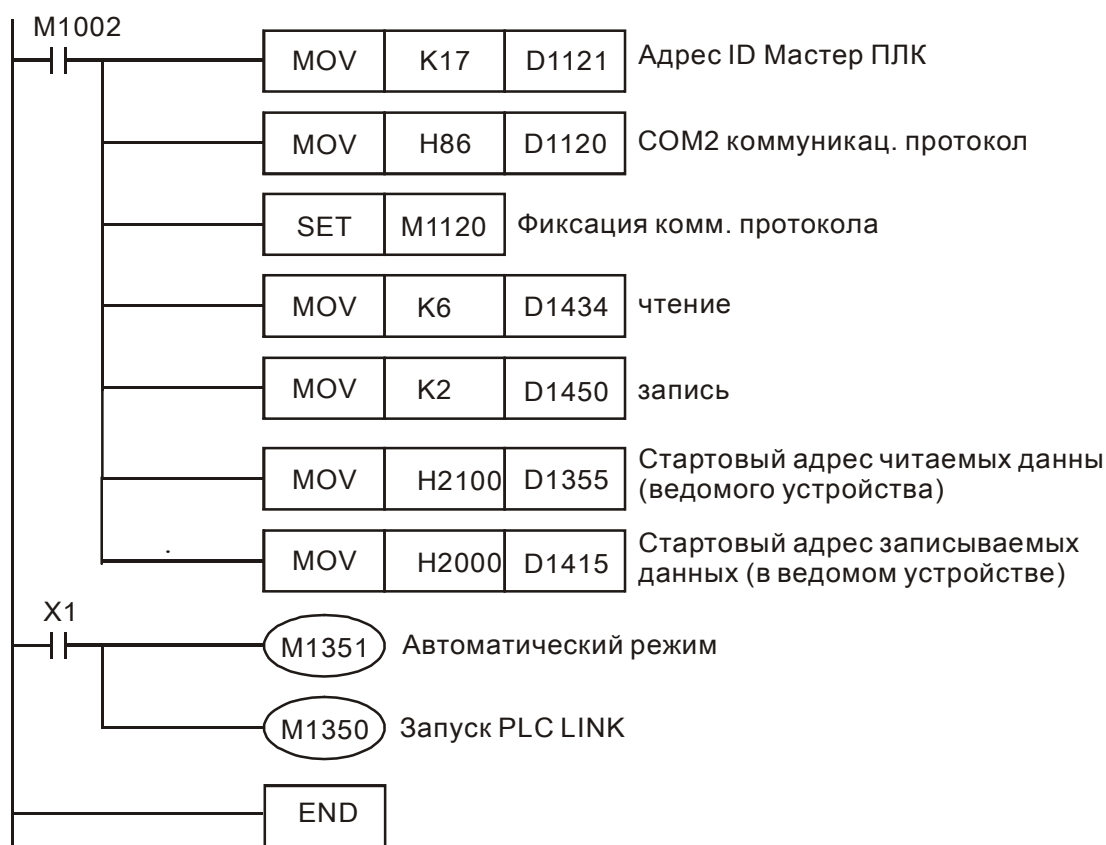
После активизации режима PLC LINK (M1350=1), допускается изменение данных, как показано ниже:

Мастер ПЛК	Заданное значение	Ведомый ПЛК	Заданное значение
D1480~D1495	K5000 для всех	D100~D115 из Slave PLC (ID=1)	K5000 для всех
D1496~D1511	K1000 для всех	D200~D215 из Slave PLC (ID=1)	K1000 для всех
D1512~D1527	K6000 для всех	D100~D115 из Slave PLC (ID=2)	K6000 для всех
D1528~D1543	K2000 для всех	D200~D215 из Slave PLC (ID=2)	K2000 для всех

- ◆ В качестве Мастер ПЛК могут использоваться модели SA/SX/SC/EH. В качестве ведомых могут использоваться любые модели ПЛК.

Пример 4: когда ПЛК соединен с преобразователем частоты Delta серии VFD-M, и используется режим PLC LINK для задания/чтения частоты вращения и управления приводом.

Адрес ведущего ПЛК: ID=17



- ◆ Регистры ПЛК D1480~D1485 передают параметры H2100~H2105 преобразователя частоты. Когда X1=ВКЛ стартует режим обмена данными PLC LINK, и данные из параметров H2100~H2105 будут копироваться в D1480~D1485.
- ◆ Регистры ПЛК D1496~D1497 передают параметры H2000~H2001 преобразователя частоты. Когда X1=ВКЛ, стартует режим обмена данными PLC LINK, и данные из регистров D1496~D1497 будут копироваться в параметры H2000~H2001.
- ◆ D1496 определяет команду управления VFD (например D1496=H12: команда ПУСК ВПЕРЕД или D1496=H1: команда СТОП)
- ◆ D1497 определяет выходную частоту VFD (например D1497=K5000, выходная частота = 50.00 Гц.)
- ◆ Используя протокол MODBUS, в качестве ведомых устройств можно использовать температурные контроллеры DTA и DTB, преобразователи частоты VFD-B,M,S,F,V,L,E и сервоприводы ASD-A. Число ведомых устройств – до 16.

Внимание:

1. Заданный диапазон для D1399: от 1 до 230.
2. Изменения D1399 действительны до разрешения запуска режима PLC EASY LINK.

Коммуникационные адреса операндов ПЛК:

Операнд	Диапазон		Тип	Адрес	ПЛК		
					ES/EX/SS	SA/SX/SC	EH
S	000~255		бит	0000~00FF	0~127	0~1024	0~1024
S	246~511		бит	0100~01FF	-		
S	512~767		бит	0200~02FF			
S	768~1023		бит	0300~03FF			
X	000~377 (Octal)		бит	0400~04FF	0~177	0~177	0~377
Y	000~377 (Octal)		бит	0500~05FF	0~177	0~177	0~377
T	000~255		бит/слово	0600~06FF	0~127	0~255	0~255
M	000~255		бит	0800~08FF	0~1279	0~4095	0~4095
M	256~511		бит	0900~09FF			
M	512~767		бит	0A00~0AFF			
M	768~1023		бит	0B00~0BFF			
M	1024~1279		бит	0C00~0CFF			
M	1280~1535		бит	0D00~0DFF	-		
M	1536~1791		бит	B000~B0FF			
M	1792~2047		бит	B100~B1FF			
M	2048~2303		бит	B200~B2FF			
M	2304~2559		бит	B300~B3FF			
M	2560~2815		бит	B400~B4FF			
M	2816~3071		бит	B500~B5FF			
M	3072~3327		бит	B600~B6FF			
M	3328~3583		бит	B700~B7FF			
M	3584~3839		бит	B800~B8FF			
M	3840~4095		бит	B900~B9FF			
C	0~199	16-бит	бит/слово	0E00~0EC7	0~127	0~199	0~199
	200~255	32-бит	бит/2слово	0EC8~0EFF	232~255	200~255	200~255
D	000~256		слово	1000~10FF	0~1311	0~4999	0~9999
D	256~511		слово	1100~11FF			
D	512~767		слово	1200~12FF			
D	768~1023		слово	1300~13FF			
D	1024~1279		слово	1400~14FF			
D	1280~1535		слово	1500~15FF			
D	1536~1791		слово	1600~16FF	-	0~4999	0~9999
D	1792~2047		слово	1700~17FF			
D	2048~2303		слово	1800~18FF			
D	2304~2559		слово	1900~19FF			
D	2560~2815		слово	1A00~1AFF			
D	2816~3071		слово	1B00~1BFF			
D	3072~3327		слово	1C00~1CFF			
D	3328~3583		слово	1D00~1DFF			
D	3584~3839		слово	1E00~1EFF			
D	3840~4095		слово	1F00~1FFF			
D	4096~4351		слово	9000~90FF			
D	4352~4607		слово	9100~91FF			
D	4608~4863		слово	9200~92FF			
D	4864~5119		слово	9300~93FF			
D	5120~5375		слово	9400~94FF			
D	5376~5631		слово	9500~95FF			
D	5632~5887		слово	9600~96FF			
D	5888~6143		слово	9700~97FF			
D	6144~6399		слово	9800~98FF			
D	6400~6655		слово	9900~99FF			
D	6656~6911		слово	9A00~9AFF			
D	6912~7167		слово	9B00~9BFF			
D	7168~7423		слово	9C00~9CFF			
D	7424~7679		слово	9D00~9DFF			

D	7680~7935	слово	9E00~9EFF			
D	7936~8191	слово	9F00~9FFF			
D	8192~8447	слово	A000~A0FF			
D	8448~8703	слово	A100~A1FF			
D	8704~8959	слово	A200~A2FF			
D	8960~9215	слово	A300~A3FF			
D	9216~9471	слово	A400~A4FF			
D	9472~9727	слово	A500~A5FF			
D	9728~9983	слово	A600~A6FF			
D	9984~9999	слово	A700~A70F			

2.12. Коды ошибок

Если светодиод "ERROR" начинает мигать или специальное реле M1004 = 1 после загрузки программы в ПЛК, это означает, что в программе есть ошибка: грамматическая или неправильный операнд. Каждая возникающая ошибка фиксируется в специальном регистре (записывается номер шага и код ошибки). Эта информация может быть считана с помощью ПК или программатора. В таблице приведен список с описаниями кодов ошибок, которые могут быть считаны из ЦПУ.

Код ошибки: D1004

Шаг ошибки: D1137

Код ошибки	Описание
0001	Операнд битового устройства S выходит за границы диапазона.
0002	Метка P выходит за границы диапазона или дублируется.
0003	Операнд KnSm выходит за границы диапазона.
0102	Флаг прерывания I выходит за границы диапазона или дублируется.
0202	Инструкция MC выходит за границы диапазона.
0302	Инструкция MCR выходит за границы диапазона.
0401	Операнд битового устройства X выходит за границы диапазона.
0403	Операнд битового устройства KnXm выходит за границы диапазона.
0501	Операнд битового устройства Y выходит за границы диапазона.
0503	Операнд битового устройства KnYm выходит за границы диапазона.
0601	Операнд битового устройства T выходит за границы диапазона.
0604	Операнд регистра T выходит за границы диапазона.
0801	Операнд битового устройства M выходит за границы диапазона.
0803	Операнд битового устройства KnMm выходит за границы диапазона.
0D01	Неправильный операнд в инструкции DECO
0D02	Неправильный операнд в инструкции ENCO
0D03	Неправильный операнд в инструкции DHSCS
0D04	Неправильный операнд в инструкции DHSCR
0D05	Неправильный операнд в инструкции PLSY
0D06	Неправильный операнд в инструкции PWM
0D07	Неправильный операнд в инструкциях FROM/TO
0D08	Неправильный операнд в инструкции PID
0E01	Операнд битового устройства C выходит за границы диапазона.
0E04	Операнд регистра C выходит за границы диапазона.

0E05	Неправильный операнд C в инструкции DCNT
0E18	Ошибка преобразования BCD
0E19	DIVISION (делитель =0)
0F04	Операнд регистра D выходит за границы диапазона.
0F05	Неправильный операнд D в инструкции DCNT
0F06	Неправильный операнд в инструкции SFTR
0F07	Неправильный операнд в инструкции SFTL
0F08	Неправильный операнд в инструкции REF
1000	Неправильный операнд в инструкции ZRST
C400	Нераспознанная инструкция
C401	Ошибка цикла
C402	Инструкция LD/LDI непрерывно используется более 9 раз
C403	Инструкция MPS непрерывно используется более 9 раз
C404	FOR-NEXT превышает 6 уровней вложения
C405	Инструкция STL/RET находится в цикле FOR/NEXT. Инструкция SRET/IRET находится в цикле FOR/NEXT. Инструкция MC/MCR находится в цикле FOR/NEXT. Инструкция END/FEND находится в цикле FOR/NEXT
C407	Инструкция STL непрерывно используется более 9 раз
C408	Использование MC/MCR в STL или I/P в STL
C409	Использование STL/RET в подпрограмме
C40A	Использование MC/MCR в подпрограмме
C40B	MC/MCR начинается не с N0 или прерывается
C40C	MC/MCR соответствует разным значениям N
C40D	Некорректное использование P/I
C40E	IRET выполняется не прошлой команды FEND. SRET выполняется не прошлой команды FEND.
C41C	Количество точек ввода/вывода модулей расширения превышает максимальное число.
C4EE	В программе нет инструкции END.

Фиксация возникновения ошибок.

Устройство	Описание	STOP → RUN	RUN → STOP
M1067	Флаг ошибки выполнения программы	Сброс	Сохраняется
M1068	Флаг фиксации ошибки выполнения программы	Сохраняется	Сохраняется
D1067	Код ошибки выполнения программы	Сброс	Сохраняется
D1068	Шаг ошибки выполнения программы	Сохраняется	Сохраняется

Код ошибки D1067	Описание
0E18	Ошибка преобразования BCD
0E19	DIVISION (делитель =0)
0E1A	Операнд битового устройства выходит за границы диапазона (включая E/F).
0E1B	Значение квадратного корня отрицательное

3. БАЗОВЫЕ КОМАНДЫ

3.1. Перечень базовых команд

Основные логические команды:

Инструкция	Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
			ES/EX/SS/ SA/SX	EH		
LD	Нормально-открытый контакт	X, Y, M, S, T, C	5.6	0.24 (0.56)	1 - 3	127
LDI	Нормально-закрытый контакт	X, Y, M, S, T, C	5.68	0.24 (0.56)	1 - 3	127
AND	Последовательный нормально-открытый контакт (логическое И)	X, Y, M, S, T, C	4.8	0.24 (0.56)	1 - 3	128
ANI	Последовательный нормально-закрытый контакт (И-НЕ)	X, Y, M, S, T, C	4.88	0.24 (0.56)	1 - 3	128
OR	Параллельный нормально-открытый контакт (логическое ИЛИ)	X, Y, M, S, T, C	4.8	0.24 (0.56)	1 - 3	129
ORI	Параллельный нормально-закрытый контакт (ИЛИ-НЕ)	X, Y, M, S, T, C	4.88	0.24 (0.56)	1 - 3	129
ANB	«И» блок: последовательное включение параллельных связей	нет	4.4	0.24	1 - 3	130
ORB	«ИЛИ» блок: параллельное включение последовательных связей	нет	4.4	0.24	1 - 3	131
MPS	Смещение вниз по стеку	нет	4.64	0.24	1 - 3	131
MRD	Считать значение стека	нет	4	0.24	1	131
MPP	Выход из стека	нет	4.4	0.24	1	132

Выходные команды:

Инструкция	Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
			ES/EX/SS/ SA/SX	EH		
OUT	ВЫХОД: присвоение выходу результата логического выражения	Y, M, S	6.4	0.24 (0.56)	1 - 3	133
SET	Включение операнда (установка лог. 1)	Y, M, S	5.04	0.24 (0.56)	1 - 3	133
RST	Сброс состояния операнда	Y, M, S, T, C, D, E, F	7.6	0.24 (0.56)	3	134

Таймеры и счетчики:

Инструкция		Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
API	Код			ES/EX/SS/ SA/SX	EH		
96	TMR	Таймер (16 бит)	T-K или T-D	9.6	0.25	4	135
97	CNT	Счетчик (16 бит)	C-K или C-D	12.8	0.30	4	135
97	DCNT	Счетчик (32 бит)	C-K или C-D	14.32	0.50	6	136

Команды магистрального управления (мастер-контроля):

Инструкция	Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
			ES/EX/SS/ SA/SX	EH		
MC	Включение условий мастер-контроля	N0 – N7	5.6	0.20	3	137
MCR	Отключение условий мастер-контроля	N0 – N7	5.76	0.12	3	137

Входные команды с обнаружением переднего и заднего фронта:

Инструкция	Код	Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
				ES/EX/SS/ SA/SX	EH		
90	LDP	Начало логического выражения с опросом по переднему фронту (импульс)	X, Y, M, S, T, C	8.16	0.56 (0.88)	3	138
91	LDF	Начало логического выражения с опросом по заднему фронту (импульс)	X, Y, M, S, T, C	8.32	0.56 (0.88)	3	139
92	ANDP	«И» с опросом по переднему фронту (импульс)	X, Y, M, S, T, C	7.68	0.56 (0.88)	3	139
93	ANDF	«И» с опросом по заднему фронту (импульс)	X, Y, M, S, T, C	7.76	0.56 (0.88)	3	140
94	ORP	«ИЛИ» с опросом по переднему фронту (импульс)	X, Y, M, S, T, C	7.68	0.56 (0.88)	3	140
95	ORF	«ИЛИ» » с опросом по заднему фронту (импульс)	X, Y, M, S, T, C	7.76	0.56 (0.88)	3	140

Выходные команды с выдачей импульса по переднему и заднему фронту:

Инструкция	Код	Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
				ES/EX/SS/ SA/SX	EH		
89	PLS	Создание импульса по переднему фронту	X, Y, M, S, T, C	8.16	0.56 (0.88)	3	141
99	PLF	Создание импульса по заднему фронту	X, Y, M, S, T, C	8.32	0.56 (0.88)	3	142

Конец программы:

Инструкция	Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
			ES/EX/SS/ SA/SX	EH		
END	Конец программы	нет	7.44	0.24	1	142

Другие команды:

Инструкция	Код	Функция	Операнд	Время выполнения, мкс		Число шагов	Стр.
				ES/EX/SS/ SA/SX	EH		
	NOP	Пустая строка	нет	3.52	0.16	1	143
98	INV	Инверсия: замена результата логических связей на противоположный	нет	3.92	0.24	1	143
	P	Адресация точки	P0 – P255	-	-	1	144
	I	Адресация точки прерывания	Ixxx	-	-	1	144

Команды пошагового управления:

Инструкция	Функция	Операнд	Число шагов	Стр.
STL	Выполнение шага	S	1	146
RET	Выход из режима шагового выполнения	нет	1	146

Примечание: Значение времени выполнения команд в EH, показанное в скобках () относится к выполнению этих команд со специальными операндами M1536 – M4095.

3.2. Описание базовых команд

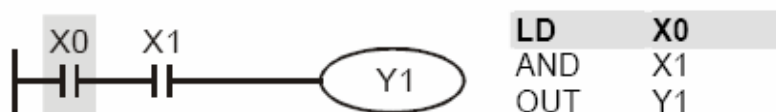
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
LD	Нормально-открытый контакт	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда LD используется в качестве нормально-открытого контакта для программирования начала логических цепочек. В контактных схемах команда всегда расположена слева и соединяется непосредственно с шиной питания.

Применение:



Команда "нормально-открытый контакт X0" открывает последовательную логическую связь. Если на входах X0 и X1 одновременно будет сигнал "1", тогда и выход Y1 установится в состояние "1".

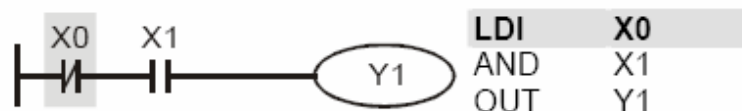
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
LDI	Нормально-закрытый контакт	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда LDI используется в качестве нормально-закрытого контакта для программирования начала логических цепочек. В контактных схемах команда всегда расположена слева и соединяется непосредственно с шиной питания.

Применение:



Команда "нормально-закрытый контакт X0" открывает последовательную логическую связь. Если на входе X0 будет "0", а на X1 будет сигнал "1", тогда выход Y1 установится в состояние "1".

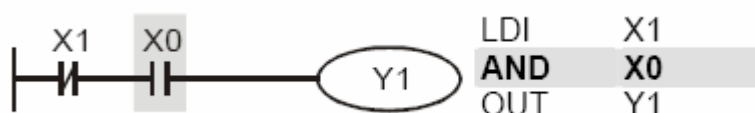
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
AND	Последовательный нормально-открытый контакт (логическое И)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда AND используется в качестве последовательного нормально-открытого контакта для программирования операции логического умножения (И). Команда представляет логическую операцию и поэтому не может программироваться в начале цепи. В начале логического выражения программируются инструкции LD или LDI.

Применение:



Команда "последовательный нормально-открытый контакт X0" создает последовательную логическую связь с контактом X1 и служит для выполнения операции логического умножения. Если на входе X1 будет "0" и на X0 будет сигнал "1", тогда выход Y1 установится в состояние "1".

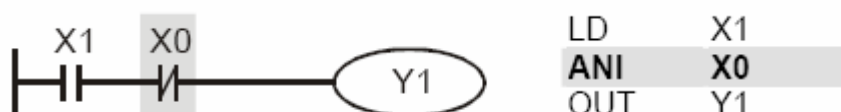
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ANI	Последовательный нормально-закрытый контакт (И-НЕ)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда ANI используется в качестве последовательного нормально-закрытого контакта для программирования операции И-НЕ. Команда представляет логическую операцию и поэтому не может программироваться в начале цепи. В начале логического выражения программируются инструкции LD или LDI.

Применение:



Команда "последовательный нормально-закрытый контакт X0" создает последовательную логическую связь с контактом X1 и служит для выполнения логической операции И-НЕ. Если на входе X1 будет "1" и на X0 не будет сигнала "1", тогда выход Y1 установится в состояние "1".

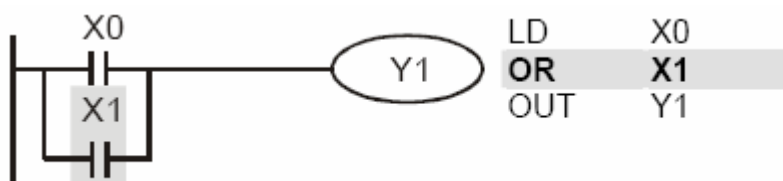
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
OR	Параллельный нормально-открытый контакт (логическое ИЛИ)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда OR используется в качестве параллельного нормально-открытого контакта для программирования операции логического сложения (ИЛИ). Команда представляет логическую операцию и поэтому не может программироваться в начале цепи. В начале логического выражения программируются инструкции LD или LDI.

Применение:



Команда "параллельный нормально-открытый контакт X1" создает параллельную логическую связь с контактом X0 и служит для выполнения операции логического сложения. Если хотя бы на одном из входов X0 или X1 будет "1", тогда и на выходе Y1 будет состояние "1".

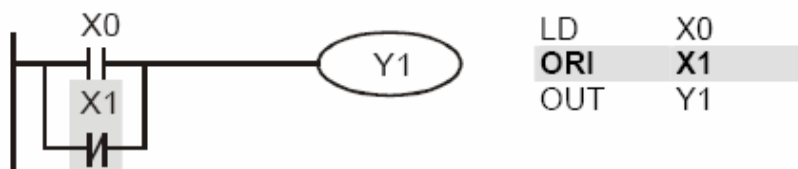
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ORI	Параллельный нормально-закрытый контакт (ИЛИ-НЕ)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда ORI используется в качестве параллельного нормально-закрытого контакта для программирования логической операции ИЛИ-НЕ. Команда представляет логическую операцию и поэтому не может программироваться в начале цепи. В начале логического выражения программируются инструкции LD или LDI.

Применение:



Команда "параллельный нормально-закрытый контакт X1" создает параллельную логическую связь с контактом X0 и служит для выполнения логической операции ИЛИ-НЕ. Если на входе X0 будет "1" или на входе X1 будет "0" (одно или оба условия одновременно), тогда на выходе Y1 будет состояние "1".

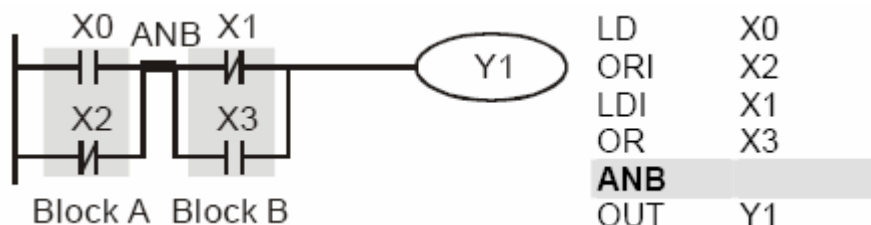
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ANB	«И» блок: последовательное включение параллельных блоков	+	+	+

Операнд	нет
---------	-----

Описание:

- Команда ANB используется для последовательного соединения цепочек из двух параллельных контактов. 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.
- 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 списка инструкций (IL), при конвертировании в контактную схему появляется автоматически и изображается как перемычка.
- 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.

Применение:



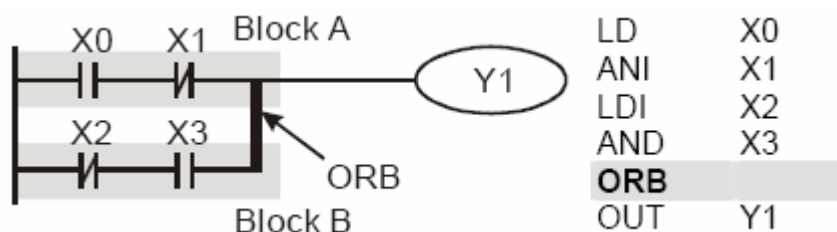
Команда ANB создает последовательную логическую связь между двумя логическими блоками (Block A и Block B).

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ORB	«ИЛИ» блок: параллельное включение последовательных блоков	+	+	+

Операнд	нет
---------	-----

Описание:

- Команда ORB используется для параллельного соединения цепочек из двух последовательных контактов. Если несколько последовательных блоков включаются параллельно, то нужно после программирования каждого отдельного блока вводить ORB-инструкцию.
- Начало разветвления программируется с помощью инструкций LD и 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 списка инструкций (IL), при конвертировании в контактную схему появляется автоматически и изображается как перемычка.
- 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.

Применение:

Команда ORB создает параллельную логическую связь между двумя логическими блоками (Block A и Block B).

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
MPS	Смещение вниз по стеку	+	+	+

Операнд	нет
---------	-----

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
MRD	Считать значение стека	+	+	+

Операнд	нет
---------	-----

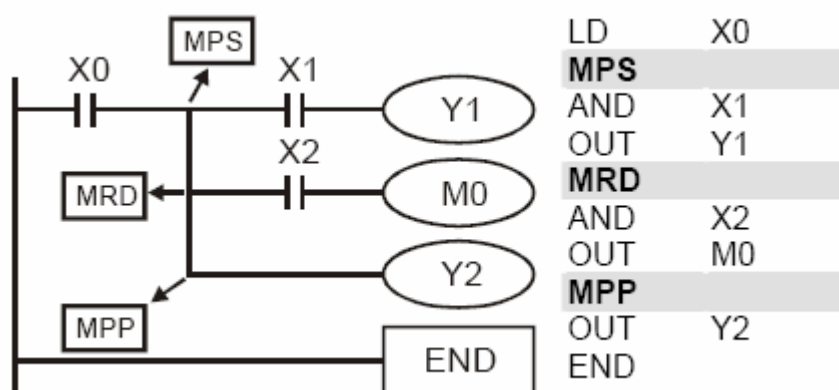
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
MPP	Выход из стека	+	+	+

Операнд	нет
---------	-----

Описание:

- 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)
- 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 списка инструкций (IL) появляются автоматически, после того как программа конвертируется в контактную схему.

Применение:



1) MPS

Промежуточный результат (здесь X0) на 1-ом уровне логических связей занесен на 1-ое место в стековую память промежуточных связей. Выполняется логическое умножение X1 с X0 и устанавливается выход Y1.

2) MRD

Перед выполнением следующей инструкции опрашивается промежуточный результат на 1-ом месте памяти логических связей. Выполняется логическое умножение X2 с X0 и устанавливается выход M0.

3) MPP

Перед выполнением следующей инструкции опрашивается промежуточный результат на 1-ом месте памяти логических связей. Устанавливается выход M0. Операция на 1-ом уровне промежуточных результатов завершена, и память логических связей стирается.

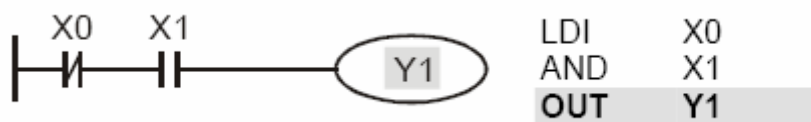
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
OUT	Выход	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	-	+	+	+	-	-	-

Описание:

- Команда OUT служит для присвоения состояния сигнала (включения или отключения выхода) в зависимости от результата логических связей (результата обработки центральным процессором логического выражения).
- С помощью инструкции OUT можно завершить программирование строки (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.
- При программировании двойной записи одинаковых выходов (их адресов) могут возникнуть проблемы при отработке программы. Избегайте двойной записи выходов, так как может привести к помехам при отработке программы.

Применение:



При условии: X0=0 и X1=1 – команда OUT Y1 установит выход контроллера Y1 в состояние "1".

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
SET	Включение выхода с фиксацией	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	-	+	+	+	-	-	-

Описание:

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

Применение:

Выход Y1 включится при выполнении условий X0, Y0 и больше от этих условий зависеть не будет. Выключить выход Y1 можно будет только командой RST Y1 или снятием питания с ПЛК.

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
RST	Сброс состояния операнда	+	+	+

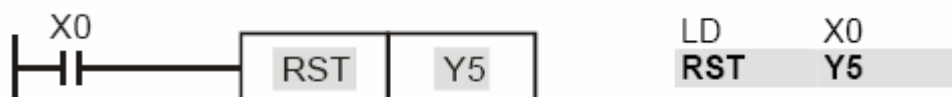
Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	-	+	+	+	+	+	+

Описание:

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

- С помощью RST-инструкции могут отключаться соответствующие операнды. Это означает:

- выходы Y, контакты M и операнды состояния шагов S выключаются (состояние сигнала "0").
- действительное значение таймера и счетчика, а также содержание регистров D, E и F 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.

Применение:

Выход Y5 выключится при выполнении условия X0 и останется выключенным даже когда условие X0 выполняться не будет.

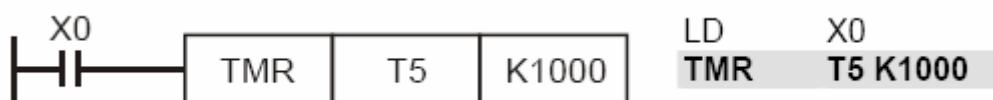
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
TMR	Таймер (16 бит)	+	+	+

Операнд	T – K	T0 – T255, K0 - K32767
	T – D	T0 – T255, D0 – D9999

Описание:

- Команда TMR служит для присвоения состояния сигнала (включения или отключения выхода) в зависимости от результата логических связей через заданный в инструкции промежуток времени.
- С помощью инструкции TMR можно завершить программирование строки (логического выражения).
- Результат логических связей, представленный посредством инструкции TMR, может применяться в следующих шагах программы как состояние входного сигнала, т.е. может многократно опрашиваться во многих логических выражениях.
- Результат логических связей, представленный TMR инструкцией, активен (включен) до тех пор, пока действуют условия его включения.
- См. так же главу 2.6 по использованию таймера.

Применение:



При условии X0=1 инструкция TMR T5 будет вести отчет времени, пока значение в регистре T5 не достигнет значения K1000 (100 сек). При X0=0 выполнение инструкции TMR прекратится и T5 сбросится на "0".

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
CNT	Счетчик (16 бит)	+	+	+

Операнд	C – K	C0 – C199, K0 - K32767
	C – D	C0 – C199, D0 – D9999

Описание:

- Команда CNT служит для суммирования количества замыканий входного контакта (макс. 32767 импульсов) и присвоения состояния сигнала (включения или отключения выхода) когда текущее значение счетчика достигнет заданного значения.
- С помощью инструкции CNT можно завершить программирование строки (логического выражения).

- Результат логических связей, представленный посредством инструкции CNT, может применяться в следующих шагах программы как состояние входного сигнала, т.е. может многократно опрашиваться во многих логических выражениях.
- Для сброса текущего значения счетчика можно использовать команду RST.
- См. так же главу 2.7 по использованию счетчика.

Применение:



При изменении состояния X0 с "0" на "1" значение регистра C20 будет увеличено на 1, и так пока значение в регистре C20 не достигнет значения K100 (100 импульсов). После этого счет прекратится. Для сброса значения регистра C20 можно использовать команду RST C20.

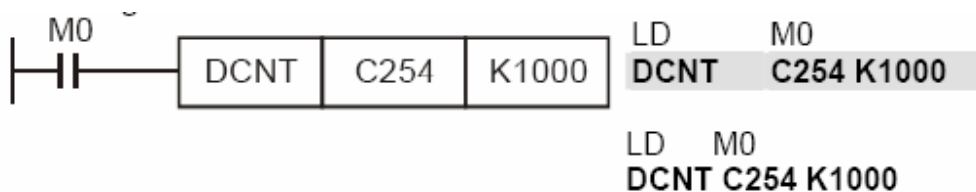
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
DCNT	Счетчик (32 бит)	+	+	+

Операнд	C – K	C200 – C254, K -2147483648 - K2147483647
	C – D	C200 – C254, D0 – D9999

Описание:

- Команда DCNT (при использовании с операндами C200 – C234) служит для суммирования/вычитания количества замыканий входного контакта и присвоения состояния сигнала (включения или отключения выхода) когда текущее значение счетчика достигнет заданного значения.
- Команда DCNT (при использовании с операндами C235 – C254) служит для высокоскоростного суммирования/вычитания количества импульсов на жестко привязанных к этим операндам входах (X0 – X17) при выполнении входного условия инструкции и присвоения состояния сигнала (включения или отключения выхода) когда текущее значение счетчика достигнет заданного значения.
- Результат логических связей, представленный посредством инструкции DCNT, может применяться в следующих шагах программы как состояние входного сигнала, т.е. может многократно опрашиваться во многих логических выражениях.
- Для сброса текущего значения счетчика можно использовать команду RST.
- Подробное описание счетчиков в главе 2.7.

Применение:



При условии M0 = 1 значение регистра C254 будет увеличено или уменьшено на 1 при каждом изменении состояния соответствующих входах (X0, X1), и так пока значение в

регистре C254 не достигнет значения K1000 (1000 импульсов). Для сброса значения регистра C20 можно использовать команду RST C254.

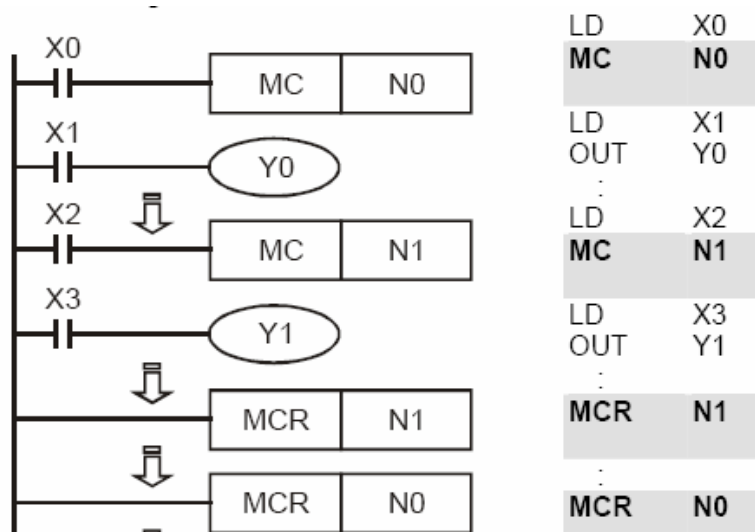
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
MC / MCR	Включение / выключение условий мастер-контроля	+	+	+

Операнд	N0 – N7
---------	---------

Описание:

- 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).
- 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.
- 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:
 - все операнды, которые программировались по OUT инструкции, отключатся
 - операнды, которые программируются по SET и RST инструкциям, сохраняют свое состояние
 - счетчики сохраняют свое текущее значение, а их контакты будут сброшены
 - общие таймеры (их текущие значения и контакты) будут сброшены
 - аккумулятивные таймеры сохраняют свое текущее значение, а их контакты будут сброшены
 - все прикладные инструкции выполняться не будут.
- Внутри программы ПК могут быть созданы до 8 уровней вложенности. 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.

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

Применение:

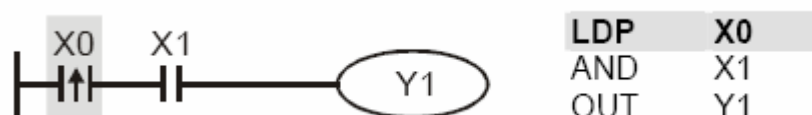
Область программы между инструкциями MC N0 и MCR N0 будет выполняться только если X0=1. Область программы между инструкциями MC N1 и MCR N1 будет выполняться только если X0=1 и X2=1

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
LDP	Начало логического выражения с опросом по переднему фронту (импульс)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

- Команда LDP используется для программирования импульсного начала логической связи.
- Инструкция LDP должна программироваться в начале цепи.
- LDP-инструкция используется также вместе с инструкциями ANB и ORB для запуска разветвлений.
- LDP-инструкция после положительного фронта сохраняется на время цикла программы (скана).

Применение:

Команда "LDP X0" открывает последовательную логическую связь. Если вход X0 изменит свое состояние с "0" на "1" (при этом X1=1), тогда выход Y1 будет в состоянии "1" в течении одного скана.

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
LDF	Начало логического выражения с опросом по заднему фронту (импульс)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

- Команда LDF используется для программирования импульсного начала логической связи.
- Инструкция LDF должна программироваться в начале цепи.
- LDF-инструкция используется также вместе с инструкциями ANB и ORB для запуска разветвлений.
- LDF-инструкция после отрицательного фронта сохраняется на время цикла программы (скана).

Применение:

```

LDF  X0
AND  X1
OUT  Y1

```

Команда "LDF X0" открывает последовательную логическую связь. Если вход X0 изменит свое состояние с "1" на "0" (при этом X1=1), тогда выход Y1 будет в состоянии "1" в течении одного скана.

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ANDP	«И» с опросом по переднему фронту (импульс)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда ANDP используется для программирования последовательного соединения импульсного контакта с опросом по переднему фронту..

Применение:

```

LD    X0
ANDP  X1
OUT   Y1

```

Команда "ANDP X1" создает последовательную логическую связь. Если вход X1 изменит свое состояние с "0" на "1" (при этом X0=1), тогда выход Y1 будет в состоянии "1" в течении одного скана.

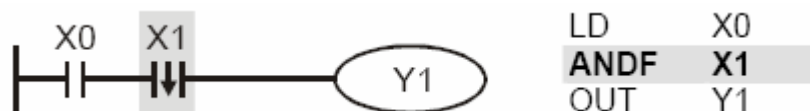
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ANDF	«И» с опросом по заднему фронту (импульс)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда ANDF используется для программирования последовательного соединения импульсного контакта с опросом по заднему фронту.

Применение:



Команда "ANDF X1" создает последовательную логическую связь. Если вход X1 изменит свое состояние с "1" на "0" (при этом X0=1), тогда выход Y1 будет в состоянии "1" в течении одного скана.

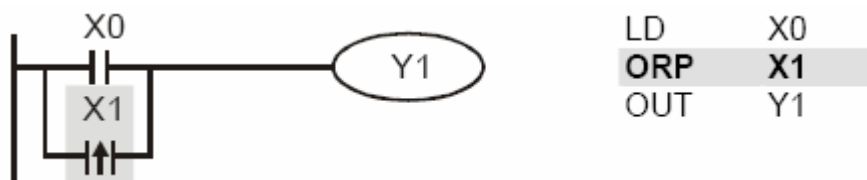
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ORP	«ИЛИ» с опросом по переднему фронту (импульс)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда ORP используется для программирования параллельного соединения импульсного контакта с опросом по переднему фронту..

Применение:



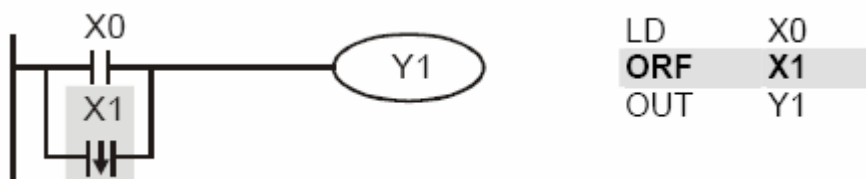
Команда "ORP X1" создает параллельную логическую связь. Выход Y1 будет в состоянии "1" в течении одного скана если вход X1 изменит свое состояние с "0" на "1" или X0=1.

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
ORF	«ИЛИ» с опросом по заднему фронту (импульс)	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	+	+	+	+	+	+	-

Описание:

Команда ORF используется для программирования параллельного соединения импульсного контакта с опросом по заднему фронту.

Применение:

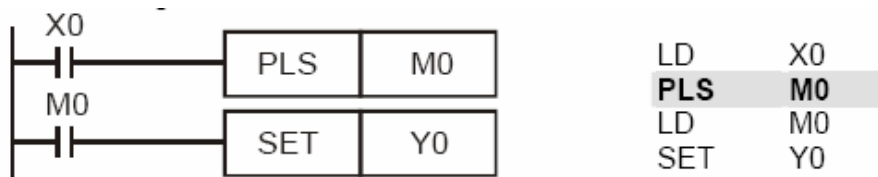
Команда "ORF X1" создает параллельную логическую связь. Выход Y1 будет в состоянии "1" в течении одного скана если вход X1 изменит свое состояние с "1" на "0" или X0=1.

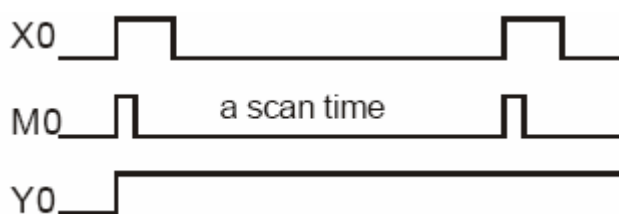
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
PLS	Создание импульса по переднему фронту	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	-	+	+	-	-	-	-

Описание:

- Команда PLS служит для генерации одного импульса – опознание переднего фронта сигнала - независимо от продолжительности входного сигнала для включения соответствующего операнда. Операнд остается во включенном состоянии на протяжении одного цикла программы (скана).
- PLS-инструкция может использоваться совместно с реле M и цифровым выходом Y. Инструкции генерируют одинаковые импульсы независимо от продолжительности входного сигнала.
- После исполнения PLS, сигнал соответствующего операнда (Y или M) удерживается в "1" на протяжении одного скана.
- PLS-инструкция генерирует одиночный импульс по возрастающему фронту входного сигнала.

Применение:



При изменении входного сигнала на входе X0 с "0" на "1" (возрастающий фронт) реле M0 благодаря PLS-инструкции получает импульс (включается на время одного скана). С помощью этого импульса по контакту реле M0 включается выход Y0.

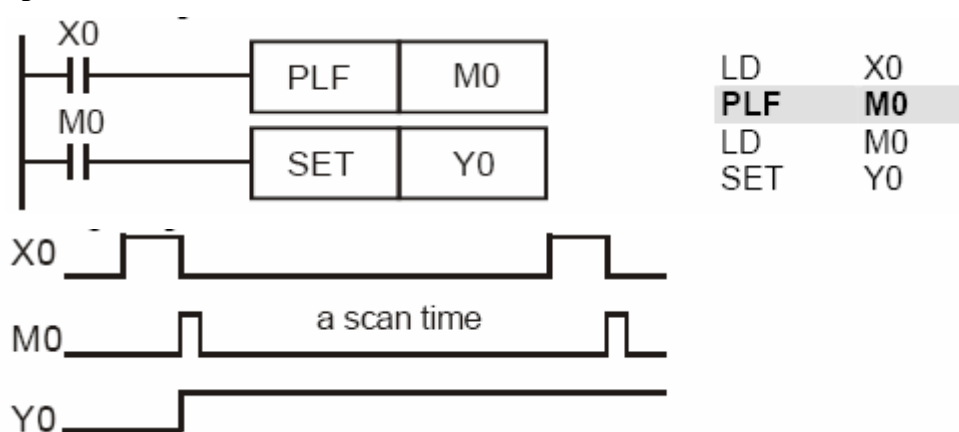
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
PLF	Создание импульса по заднему фронту	+	+	+

Операнд	X0 – X377	Y0 – Y377	M0-M4095	S0-S1023	T0 – T255	C0 – C255	D0-D9999
	-	+	+	-	-	-	-

Описание:

- Команда PLF служит для генерации одного импульса – опознание заднего фронта сигнала - независимо от продолжительности входного сигнала для включения соответствующего операнда. Операнд остается во включенном состоянии на протяжении одного цикла программы (скана).
- PLF-инструкция может использоваться совместно с реле M и цифровым выходом Y. Инструкции генерируют одинаковые импульсы независимо от продолжительности входного сигнала.
- После исполнения PLF, сигнал соответствующего операнда (Y или M) удерживается в "1" на протяжении одного скана.
- PLF-инструкция генерирует одиночный импульс по заднему фронту входного сигнала.

Применение:



При изменении входного сигнала на входе X0 с "1" на "0" (спадающий фронт) реле M0 благодаря PLS-инструкции получает импульс (включается на время одного скана). С помощью этого импульса по контакту реле M0 включается выход Y0.

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
END	Конец программы	+	+	+

Операнд	нет
---------	-----

Описание:

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

- Каждая программа ПЛК 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 установка 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 сторожевого таймера (Watch-Dog-Timers)

Применение:

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
NOP	Пустая строка в программе	+	+	+

Операнд	нет
---------	-----

Описание:

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

- 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.

Применение:

```

LD      X0
NOP
OUT     Y1
  
```

В контактных схемах инструкция NOP не отображается.

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
INV	Инверсия - замена результата логических связей на противоположный	+	+	+

Операнд	нет
---------	-----

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.
- 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.

Применение:

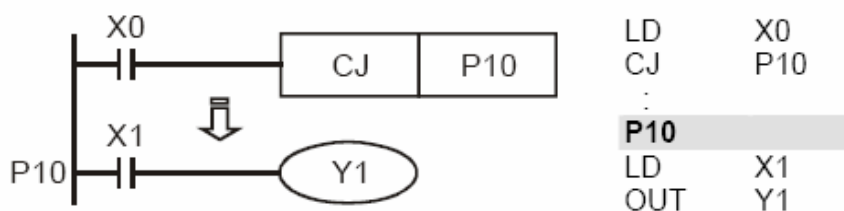
Если $X0 = 0$, выход $Y1 = 1$. Если $X0 = 1$, выход $Y1 = 0$.

Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
P	Адресация точки	+	+	+

Операнд	P0 – P255
---------	-----------

Описание:

- P-инструкция служит для указания точки перехода для команд CJ, CALL.
- Номер точки в программе не должен повторяться.

Применение:

Точка P10 указывает адрес перехода программы при выполнении инструкции CJ P10.

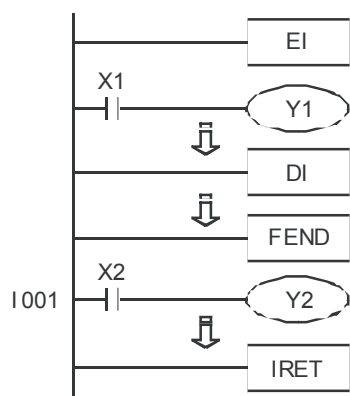
Инструкция	Функция	ПЛК		
		ES/EX/SS/	SA/SX	EH
I	Адресация точки прерывания	+	+	+

Операнд	I00□, I10□, I20□, I30□, I40□, I50□, I6□□, I7□□, I8□□ I010, I020, I030, I040, I050, I060, I110, I120, I130, I140
---------	--

Описание:

- I-инструкция служит для указания точки перехода к подпрограмме обработки прерывания для команд IRET (API 03), EI (API 04), DI (API 05)
- Число точек прерывания I в DVP-ES/EX/SS - четыре: (I001, X0), (I101, X1), (I201, X2) и (I301, X3).
- Число точек внешнего прерывания I в DVP-SA/SX - шесть: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4) и (I50□, X5). (□=1: прерывание по переднему фронту; □=0: прерывание по заднему фронту).
- Число точек временного прерывания в DVP-SA/SX - две: I6□□, I7□□. (□□=10~99мс)
- Число точек прерывания высокоскоростного счета в DVP-SA/SX - шесть: I010 (используется с C235, C241, C244, C246, C247, C249, C251, C252, C254), I020 (используется с C236), I030 (используется с C237, C242), I040(используется с C238), I050(используется с C239), I060 (используется с C240). (используется с командами API 53 DHSCS)
- Число точек внешнего прерывания I в DVP-EH - шесть: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4) и (I50□, X5). (□=1: прерывание по переднему фронту; □=0: прерывание по заднему фронту).
- Число точек временного прерывания в DVP-EH - три: I6□□, I7□□, I8□□. (□□=10~99мс)
- Число точек прерывания высокоскоростного счета в DVP-EH - шесть: I010, I020, I030, I040(используется с C238), I050, I060. (используется с командами API 53 DHSCS).
- Число точек прерывания импульсных выходов в DVP-EH - четыре: I110, I120 (включаются после последнего импульса), I130, I140(включаются с первым импульсом).

Применение:



EI	Разрешение прерывания
LD	X1 Норм. откр. контакт X1
OUT	Y1 Выход Y1
:	
DI	Запрещение прерывания
:	
FEND	Конец основной программы
I001	Точка подпр. обработки прерывания
LD	X2 Норм.откр.контакт X2
OUT	Y2 Выход Y2
:	
IRET	Конец подпрограммы обраб. прерывания

4. КОМАНДЫ ПОШАГОВОГО УПРАВЛЕНИЯ

Команды пошагового управления STL и RET являются элементарными инструкциями ПЛК для единого программирования процесса управления. STL-инструкция применяется совместно с шаговыми реле 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 реле S. В зависимости от серии DVP имеется более 1000 шаговых операндов в области от S0 до S1023, при этом операндам S0...S9 присвоены жесткие функции.

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.

4.1. Инструкции [STL], [RET]

Инструкция	Функция	Операнд	ПЛК		
			ES/EX/SS/	SA/SX	EH
STL	Выполнение шага	S0 – S1023	+	+	+

Описание

- STL-инструкция применяется совместно с операндом шагов S..
- В программе без шагового управления шаговые операнды S могут применяться также как обычные внутренние реле.
- 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.
- 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.

Инструкция	Функция	Операнд	ПЛК		
			ES/EX/SS/	SA/SX	EH
RET	Конец области пошагового управления	нет	+	+	+

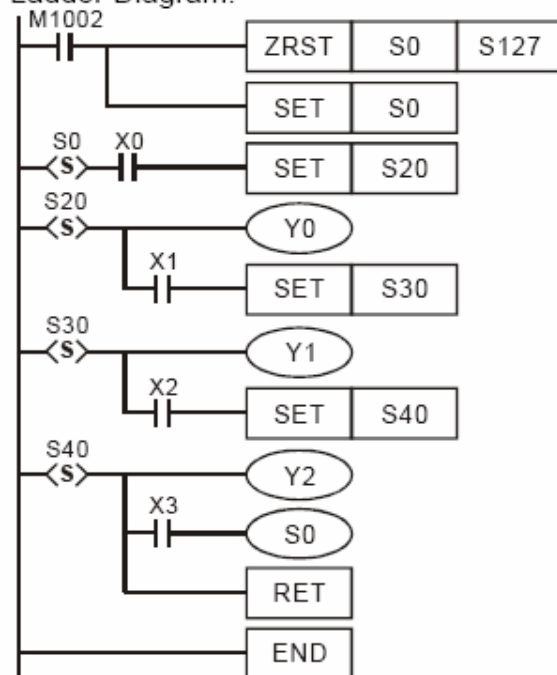
С помощью RET-инструкции заканчивается вся область STL-программы (пошагового управления).

Последний шаг управления должен заканчиваться RET-инструкцией

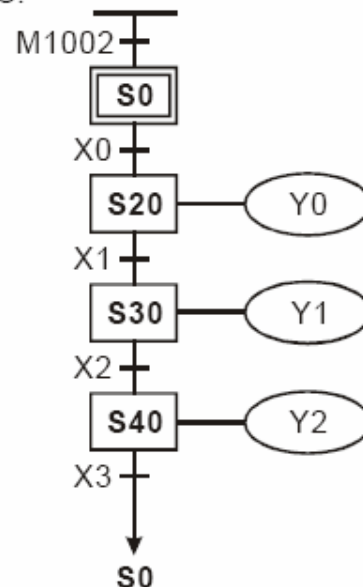
Активизированный последним операнд S должен отключаться с помощью RST-инструкции, или он должен выполнять дальнейшее включение (разрешение) обратно к началу шаговой цепи.

Пример применения:

Ladder Diagram:



SFC:

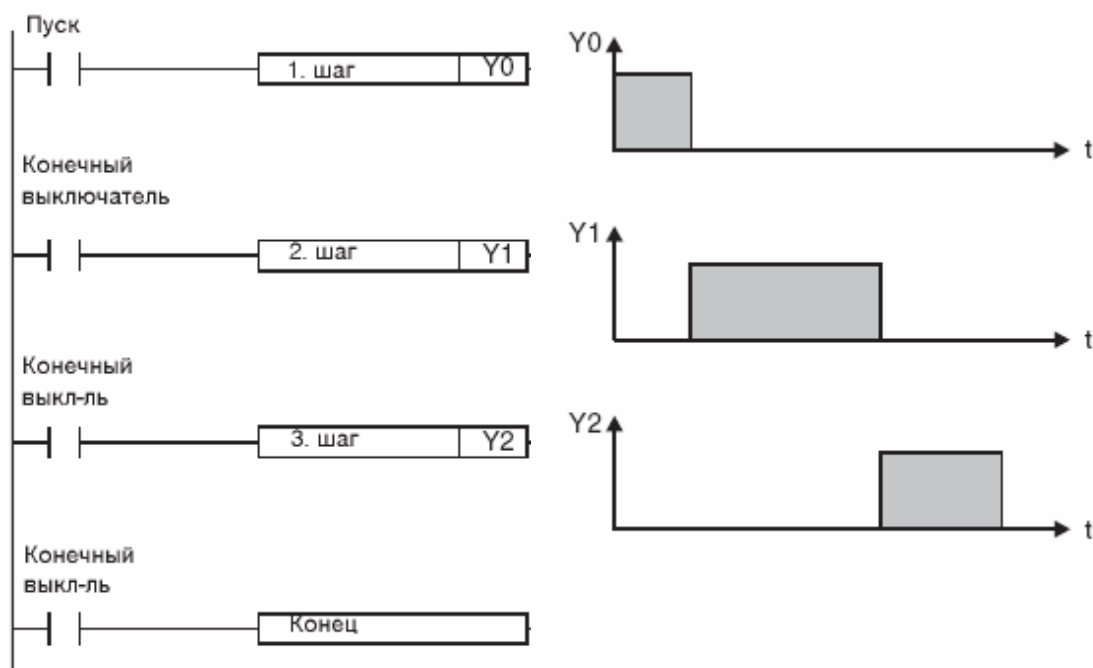


4.2. Последовательные функциональные диаграммы (SFC)

Традиционный метод проектирования посредством контактной схемы состоит в том, что выход включается 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.

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.

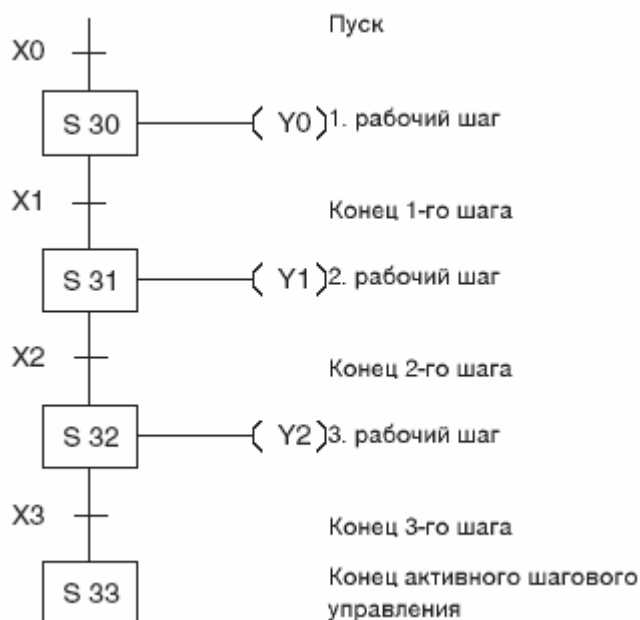
На основании краткой последовательности процесса в качестве примера описано шаговое управление с четырьмя рабочими шагами. Четвертый шаг заканчивает шаговое управление.












По рисунку видно, что 2-ой рабочий шаг включается, как только заканчивается 1-ый шаг 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).

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



Символы SFC-диаграмм:

	Символ используется для перехода в режим релейно-контактных схем (LAD)
	Символ используется для программирования инициализирующих шаговых операндов S0...S9.
	Символ используется для программирования шаговых операндов общего назначения S10...S1023.
	Символ используется для программирования команды перехода в заданную точку диаграммы: переход вверх или вниз основной программы, возвращение на начальный шаг и т.д.
	Символ используется для программирования условных переходов между отдельными шагами в программе.
	Символ используется для программирования селективного разветвления в программе, когда в зависимости от соответственно примененных входных условий производится выбор, какой процесс состояний должен активизироваться в программе.
	Символ используется для программирования сборки (окончания) селективного разветвления.
	Символ используется для программирования параллельного разветвления в программе, при котором два или несколько процессов состояний обрабатываются одновременно.
	Символ используется для программирования сборки (окончания) параллельного разветвления.

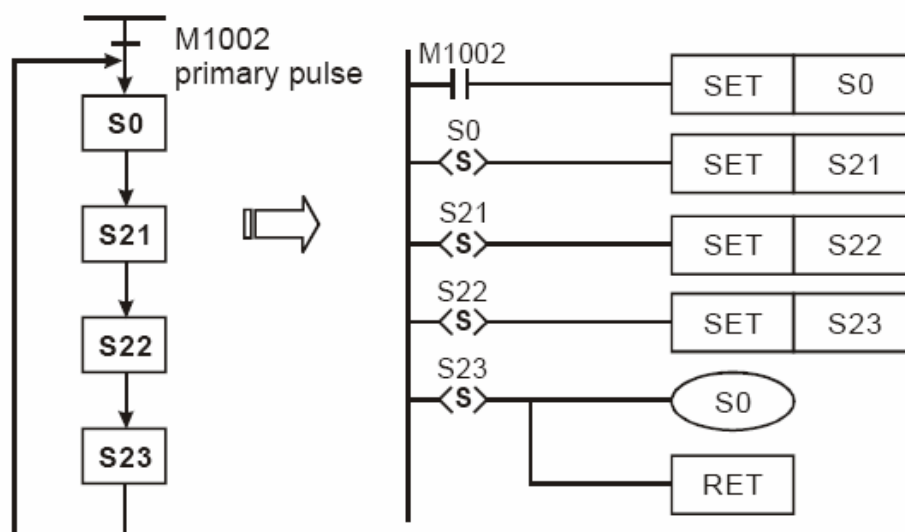
4.3. Описание инструкций пошагового управления

Каждое состояние шага требует инициализации. Для этого имеются, например, инициализирующие операнды S0...39. 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.).

Последний шаг управления должен заканчиваться RET-инструкцией. Без нее при компиляции будет выдана ошибка с указанием последнего шага.

Специальное реле M1002 задействует при включении ПЛК определенное системное состояние (импульс при включении). 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 (см. рис.)

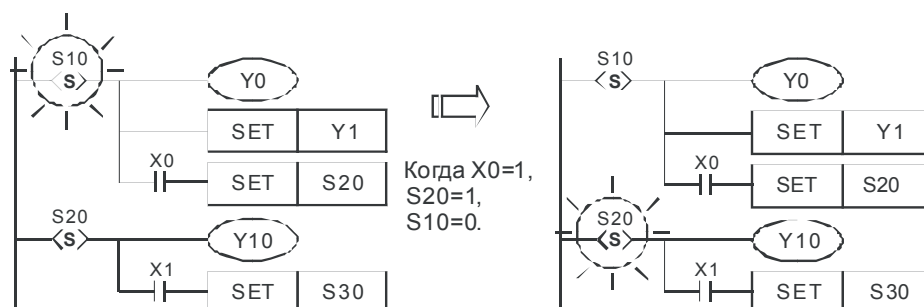


1. Работа режима пошагового управления:

Каждый шаг представляет собой набор определенных управляющих процедур. Одновременно могут выполняться процедуры только одного шага. Каждый шаг должен выполнять следующие задачи:

- Установка состояния выхода;
- Проверка входного условия;
- Определение номера следующего управляющего шага.

Пример:



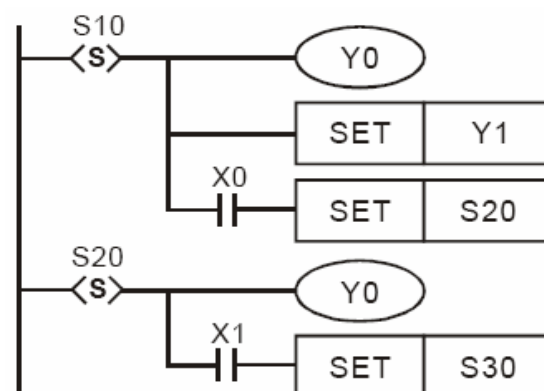
Шаг1 S10=1: Y0=1, Y1=1, и когда X0 буде =1: S10 = 0, Y0=0, Y1=1 (при использовании команды SET состояние выхода сохраняется) => Шаг2 S20 = 1: Y10 = 1 и т.д.

2. Задержка по времени: когда состояние шага $S_n = 1$, его схема будет активна, и его схема будет неактивна, когда состояние шага $S_n = 0$ (задержка между этими акциями – время одного скана).

3. Повторение использования выходов:

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

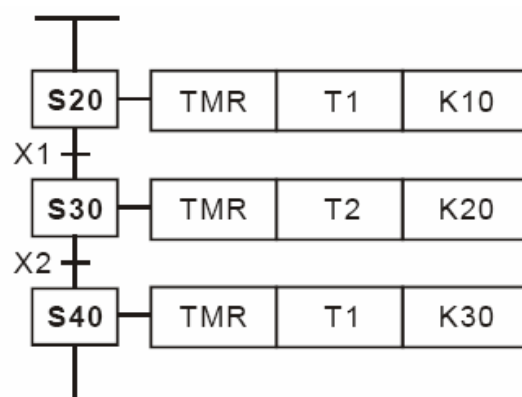
На приведенном участке программы одинаковый выход (Y0) опрашивается через разные STL-инструкции или операнды состояния шага (S10 или S20). Y0 включается, если активен S10 или S20. Y0 выключается, если не активны S10 и S20. В этом случае



двойная запись не создает проблем, так как шаги 10 и 20 не могут быть активными одновременно.

4. Многократная запись таймера

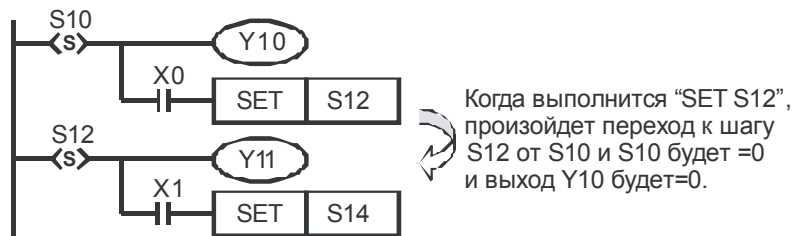
В одной программе таймер благодаря применению операнда шагового состояния (статуса) может записываться многократно. Однако одинаковый таймер не может применяться в двух один за другим следующих шагах.



5. Переход к следующему шагу

Переход к следующему шагу может выполняться командами SET Sn и OUT Sn. Причины использования той или иной из этих команд могут быть различными.

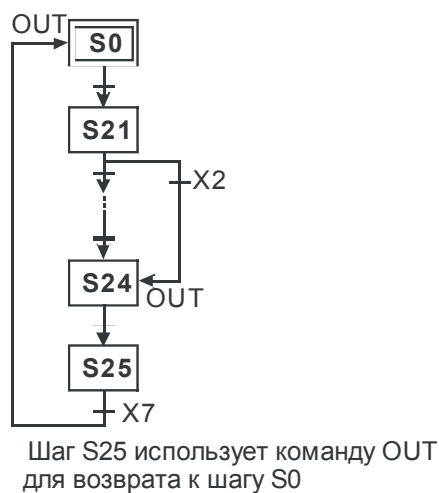
Команда SET Sn: используется для перехода к следующему шагу в одном процессе и после выполнения происходит ее самообнуление.



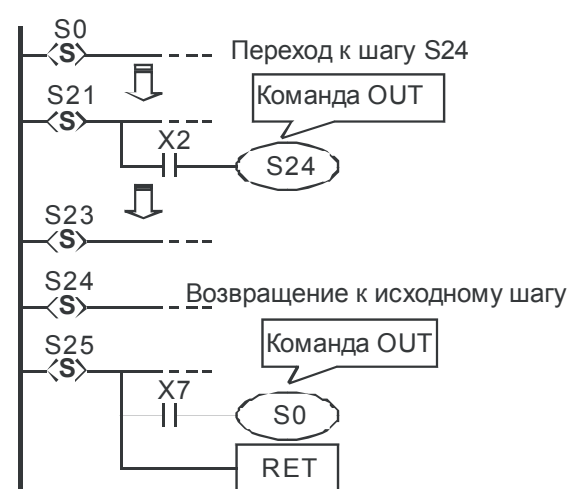
Команда OUT Sn: используется для переходного разветвления в одном процессе, для возвращения в начальную точку процесса, для перехода к другому процессу, и после выполнения происходит ее самообнуление.

Пример скачкообразного перехода (JUMP) к заданному шагу в одном процессе, и возвращения в начальную точку процесса:

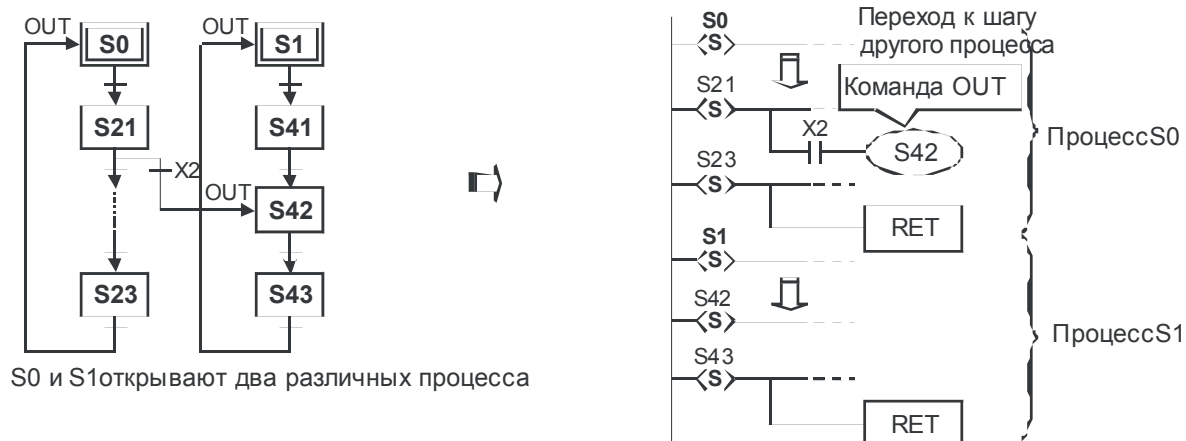
SFC:



LAD:

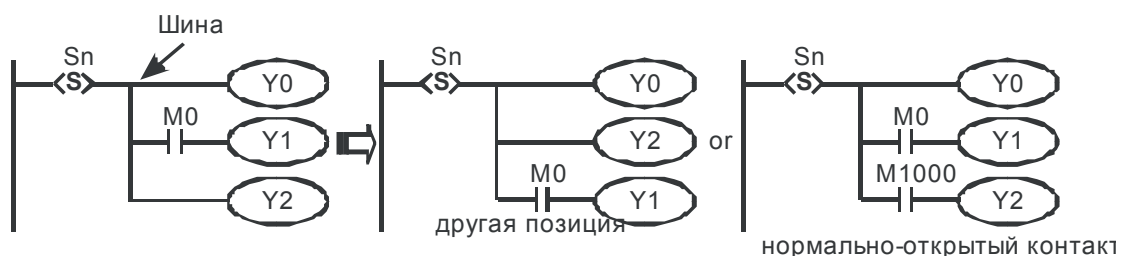


Пример скачкообразного перехода к заданному шагу другого процесса:



6. Корректное соединение выходов

Выход третьей строки не должен быть соединен напрямую (как показано в левой диаграмме). Корректное соединение в средней и правой диаграмме.



7. Допустимые инструкции в режиме пошагового управления

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

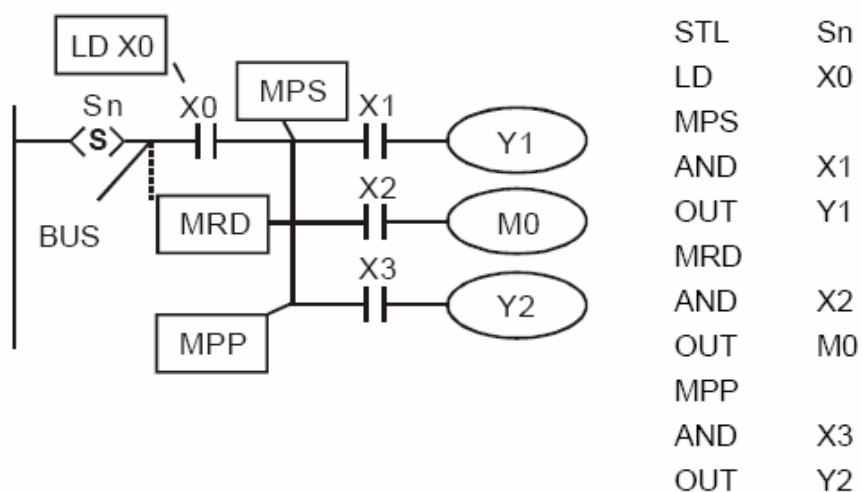
Состояние		Инструкции		
		LD/LDI/LDP/LDF/AND/ ANI/ANDP/ANDF/OR/ ORI/ORP/ORF/INV/ OUT/SET/RST	ANB/ORB MPS/MRD/MPP	MC/MCR
Инициализирующее состояние		да	да	нет
Разветвление программы	Общие выходы	да	да	нет
	Выходы перехода к следующему шагу	да	нет	нет

Инструкции мастер-контроля MC/MCR не могут быть использованы в режиме пошагового управления.

STL-инструкции нельзя использовать в общих подпрограммах и подпрограммах обработки прерывания.

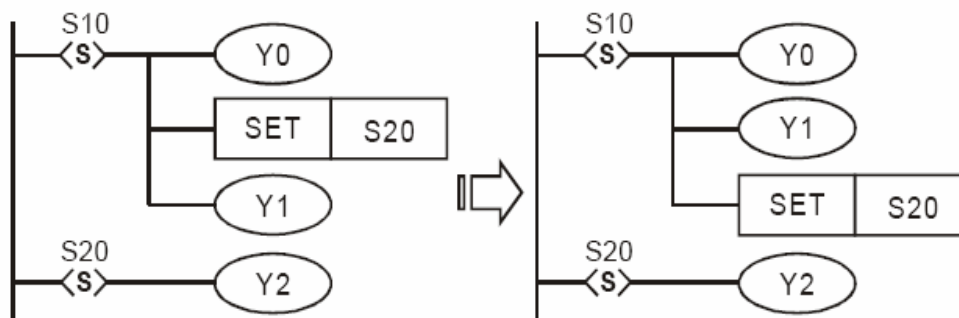
Команда CJ может быть использована в режиме пошагового управления, однако она может вызвать трудности работы режима и лучше ее не применять.

Команды MPS/MRD/MPP нельзя использовать сразу за STL Sn, а только после команд LD или LDI:

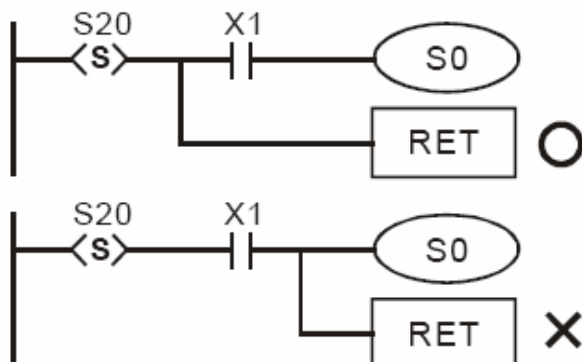


8. Другие замечания

Команды перехода к следующему шагу рекомендуется ставить в последней строке шага:



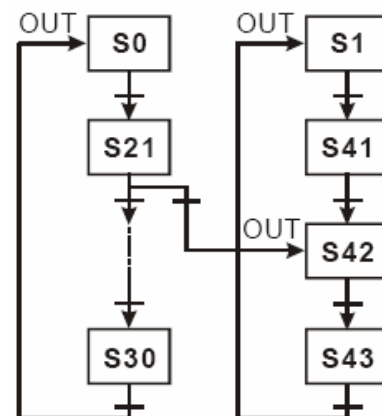
Команду RET нельзя ставить после условия:



4.4. Особенности программирования пошагового управления

4. Каждый процесс должен начинаться с инициализирующего реле (S0 –S9), а заканчиваться командой RET.
5. В программе без шагового управления шаговые операнды S могут применяться также как обычные внутренние реле.
6. Операнд состояния шага S может программироваться только один раз в программе с помощью STL-инструкции и не должен повторяться.
7. Пошаговое управление может обрабатывать различные, друг от друга независимые процессы состояний и разветвления. Нужно различать процессы:
 - Простой (линейный) процесс
 - Один процесс с селективными и/или параллельными (одновременными) разветвлениями
 - Комбинационные процессы: параллельное выполнение нескольких процессов (макс.10: S0-S9).
8. Разветвления и переходы между различными процессами: имеется возможность перескочить через часть области (схемы) последовательности состояния или многократно выполнить петлю программы.

Шаг S21 называется точкой переходного разветвления.



9. Ограничения применения разветвлений:
 - Может программироваться максимум 8 разветвлений, выходящих из одного шагового операнда.
 - Общее количество всех разветвлений селективных или параллельных не должно превышать 16.
 - Из одной точки в одном процессе можно совершить переход только в одну точку другого процесса.
10. Сброс шаговых реле и блокировка выходов:
 - Для сброса шаговых реле можно использовать команду ZRST.
 - Для блокировки выходов Y можно использовать специальное реле M1034=1.

Сохранение состояния шаговых реле:

При сбое электропитания ПЛК текущее состояние шаговых реле будет сохранено и при восстановлении питания выполнение программы будет продолжено с того же места.

11. Специальные реле и регистры для режима пошагового управления:

M1040	Запрещение переходов в режиме STL
M1041	Старт шагового перехода. Флаг для инструкции IST
M1042	Импульсный старт STL. Флаг для инструкции IST
M1043	Возвращение в нулевую точку завершено. Флаг для инструкции IST
M1044	Нахождение в нулевой точке. Флаг для инструкции IST
M1045	Запрещение сброса всех выходов. Флаг для инструкции IST
M1046	Режим STL выполняется
M1047	Разрешение отображение состояния шагов в регистрах D1040 – D1047
D1040	Номер 1-го активированного состояния шага (для последовательного режима)
D1041	Номер 2-го активированного состояния шага (для последовательного режима)
D1042	Номер 3-го активированного состояния шага (для последовательного режима)
D1043	Номер 4-го активированного состояния шага (для последовательного режима)
D1044	Номер 5-го активированного состояния шага (для последовательного режима)
D1045	Номер 6-го активированного состояния шага (для последовательного режима)
D1046	Номер 7-го активированного состояния шага (для последовательного режима)
D1047	Номер 8-го активированного состояния шага (для последовательного режима)

4.5. Типы процессов пошагового управления

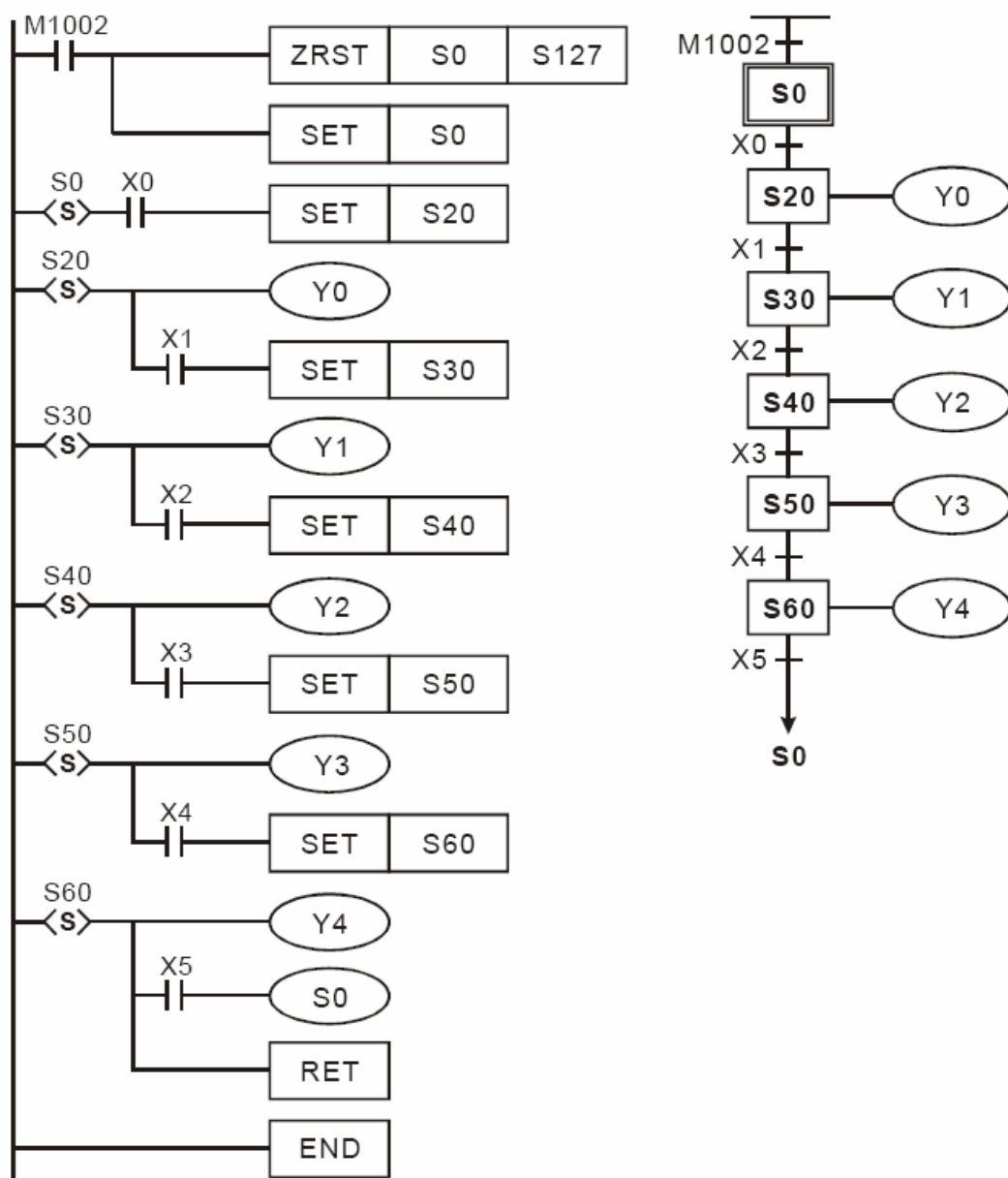
А. ПРОСТОЙ ОДИНОЧНЫЙ ПРОЦЕСС

Процесс начинается с реле инициализации S0 – S9, далее следуют общие шаговые реле S10 – S1023. S10 – S19 могут использоваться в качестве точек сброса с командой IST.

А-1. Простой (линейный) процесс без разветвлений

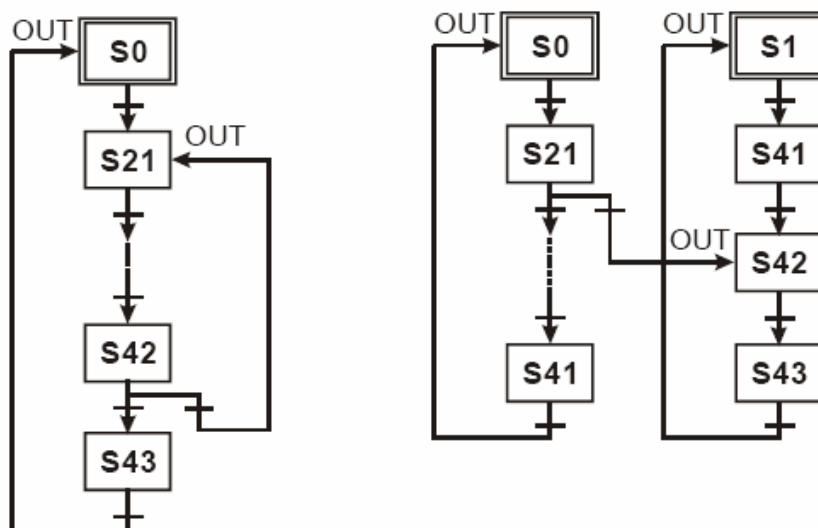
При простом процессе шаговые состояния обрабатываются последовательно (один за другим). Последовательность обработки определяется только положением шагового состояния в простом процессе и благодаря независимости от адреса шагового состояния.

Пример простого процесса без разветвлений:



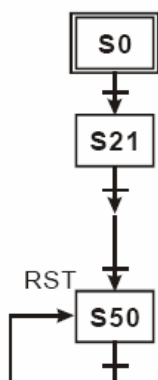
A-2. Процесс с переходными разветвлениями (JUMP)

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



А-3. Процесс со сбросом

S50 сбросит себя и завершит процесс, когда выполнится условие.



В. СЛОЖНЫЙ ОДИНОЧНЫЙ ПРОЦЕСС

Включает в себя селективные, параллельные и комбинированные разветвления.

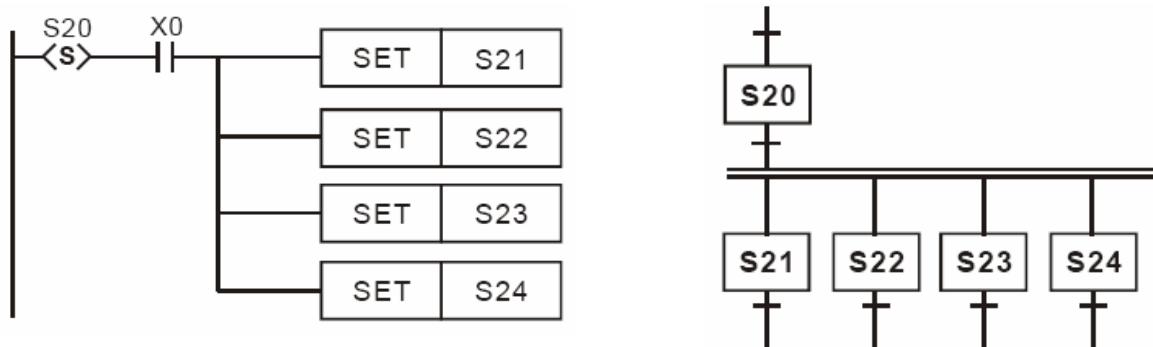
В-1. Параллельное разветвление

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

Количество всех разветвлений не должно превышать 16.

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.



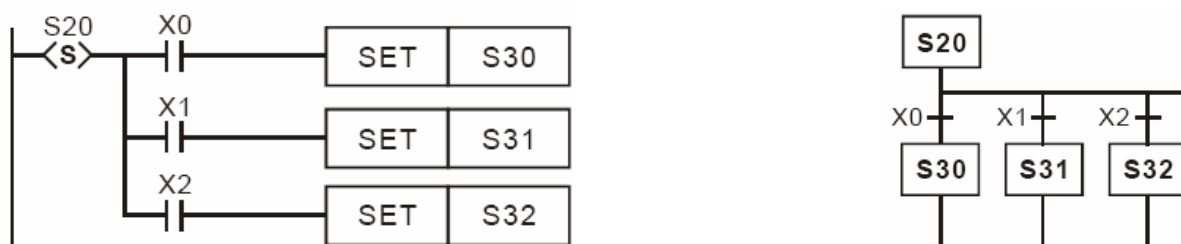
В-2. Селективное разветвление

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

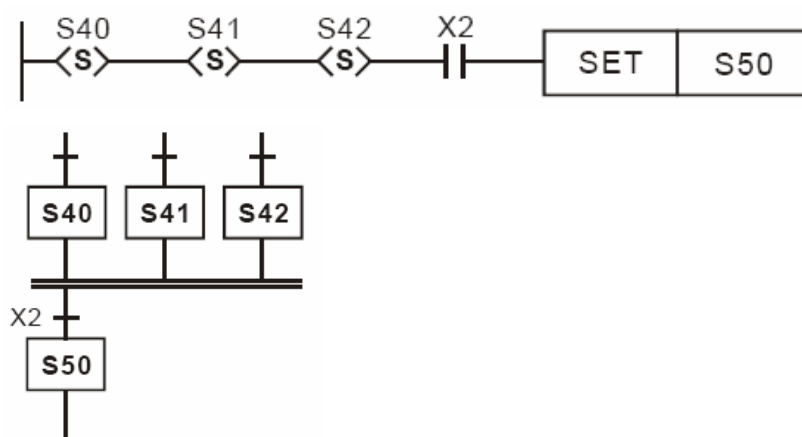
Из одного шагового состояния разветвление может создавать несколько (максимум 8) процессов состояний.

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

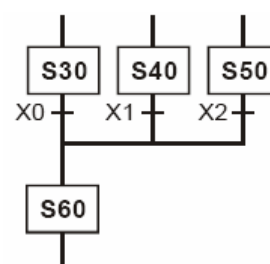
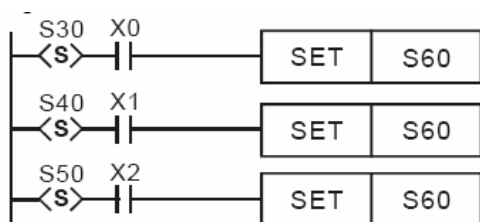
Может программироваться максимум 8 разветвлений, выходящих из одного шагового операнда. Общее количество всех селективных разветвлений не должно превышать 16.



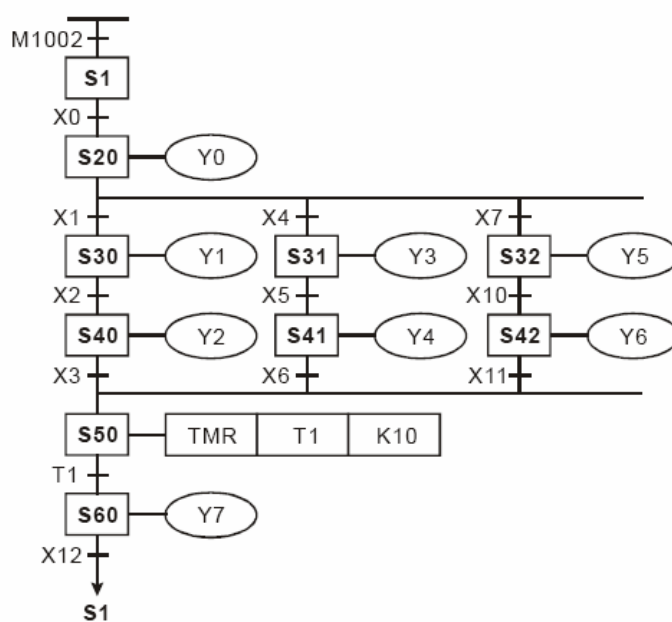
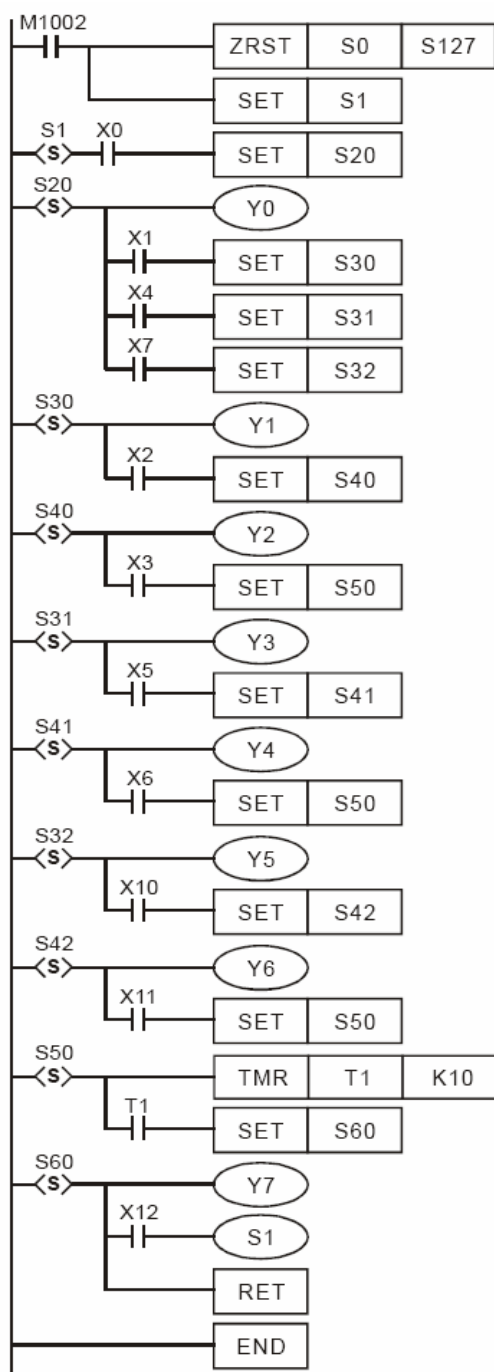
В-3. Сборка параллельного разветвления



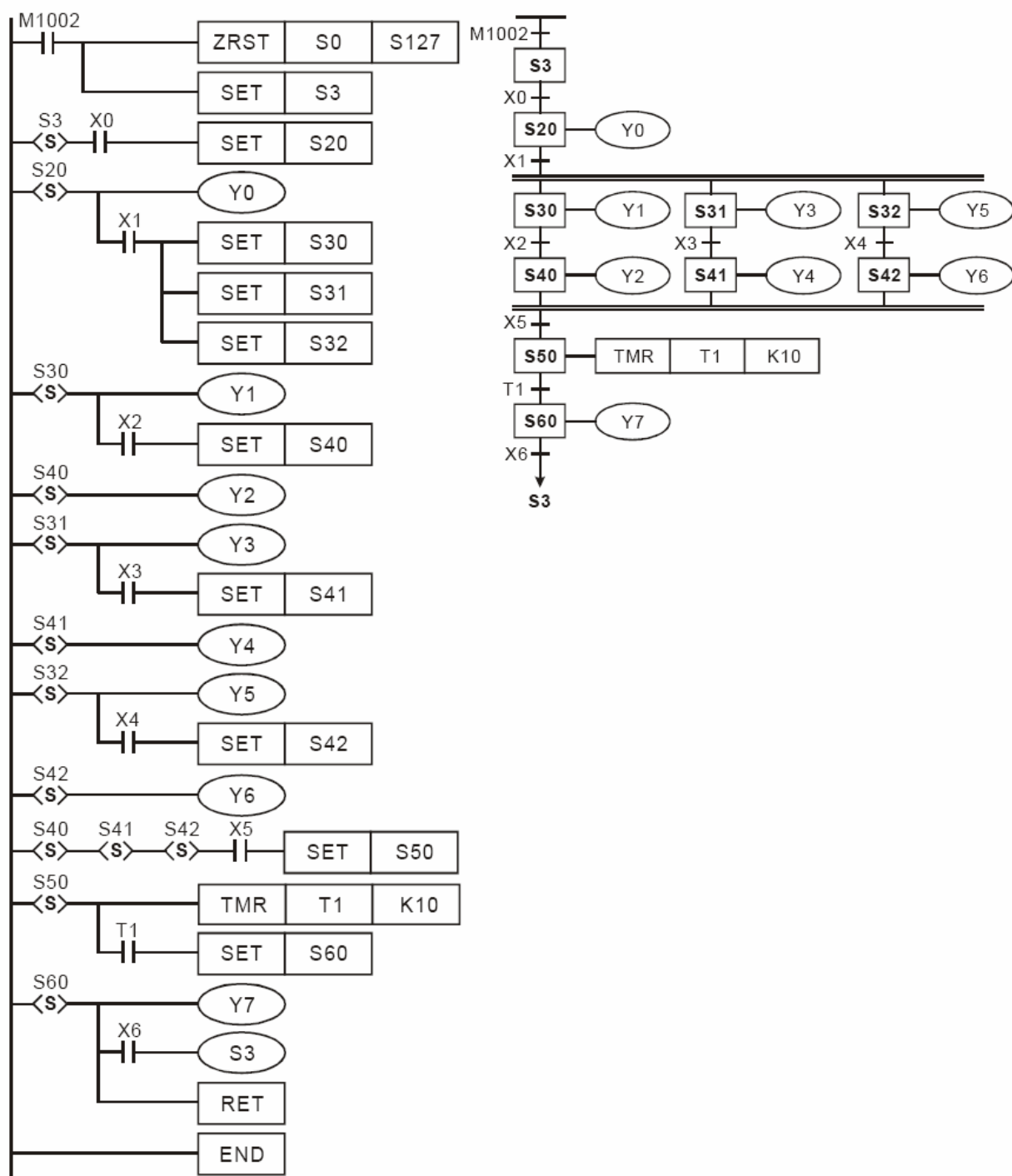
В-4. Сборка селективного разветвления



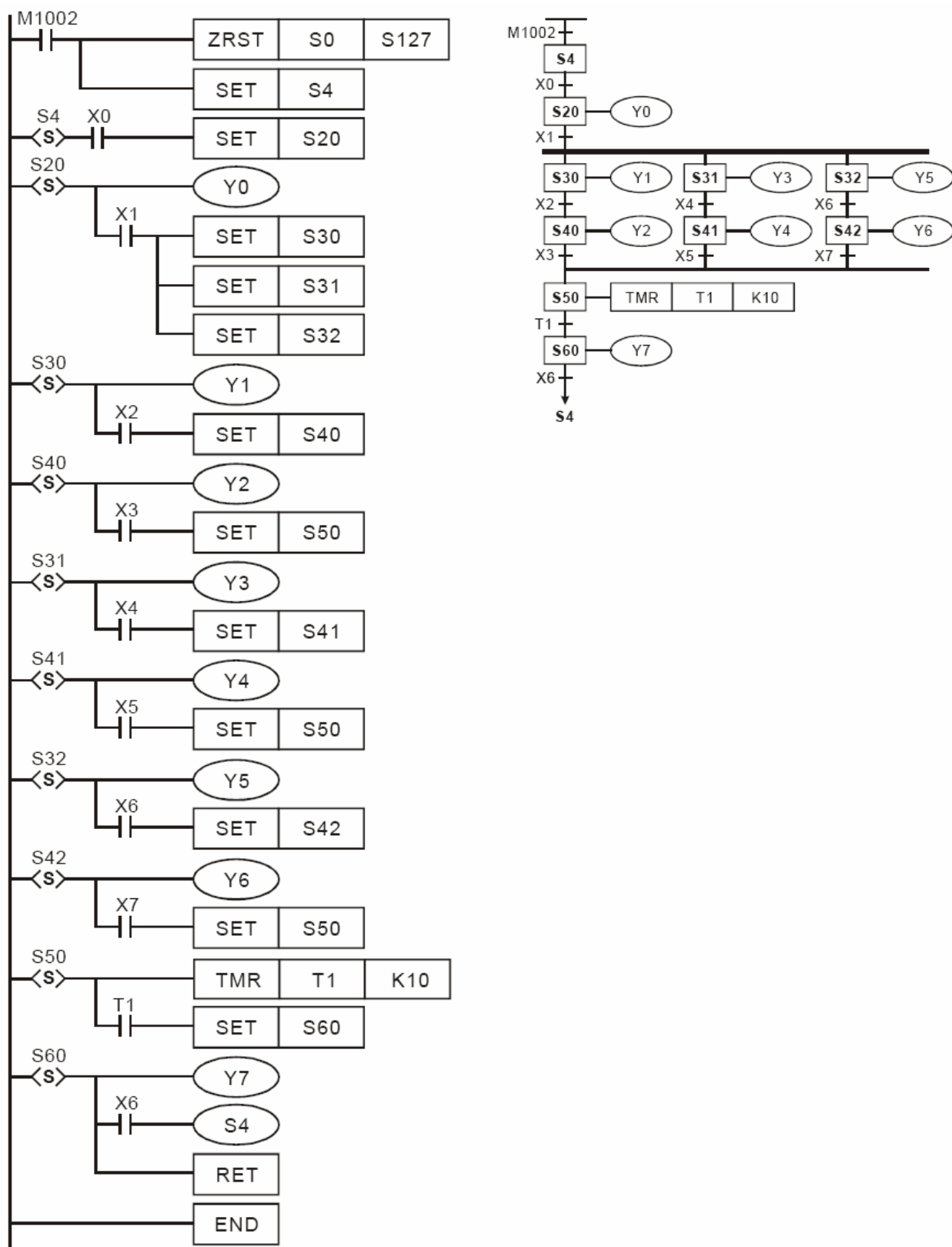
Пример процесса с селективным разветвлением и сборкой:



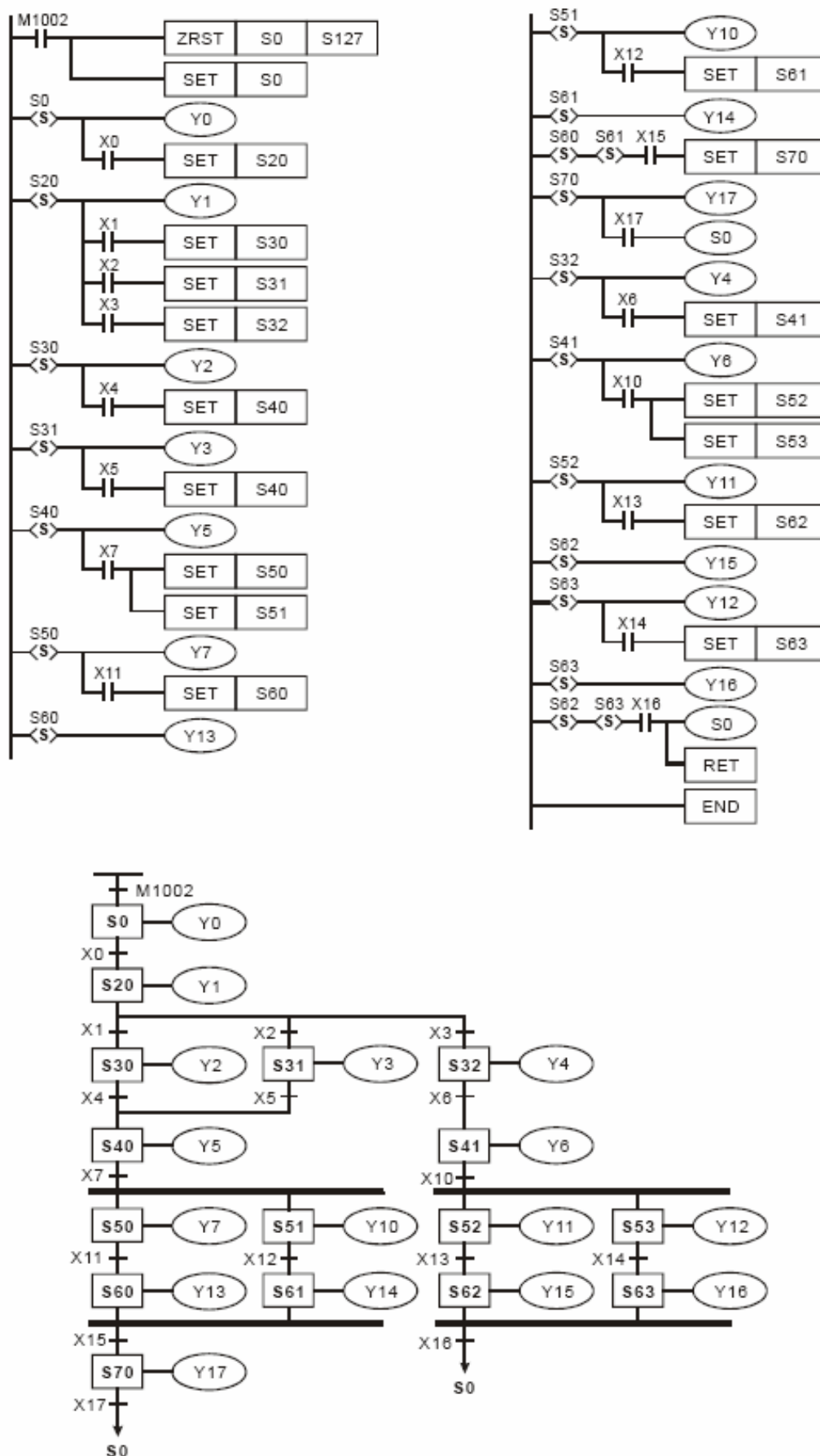
Пример процесса с параллельным разветвлением и сборкой:



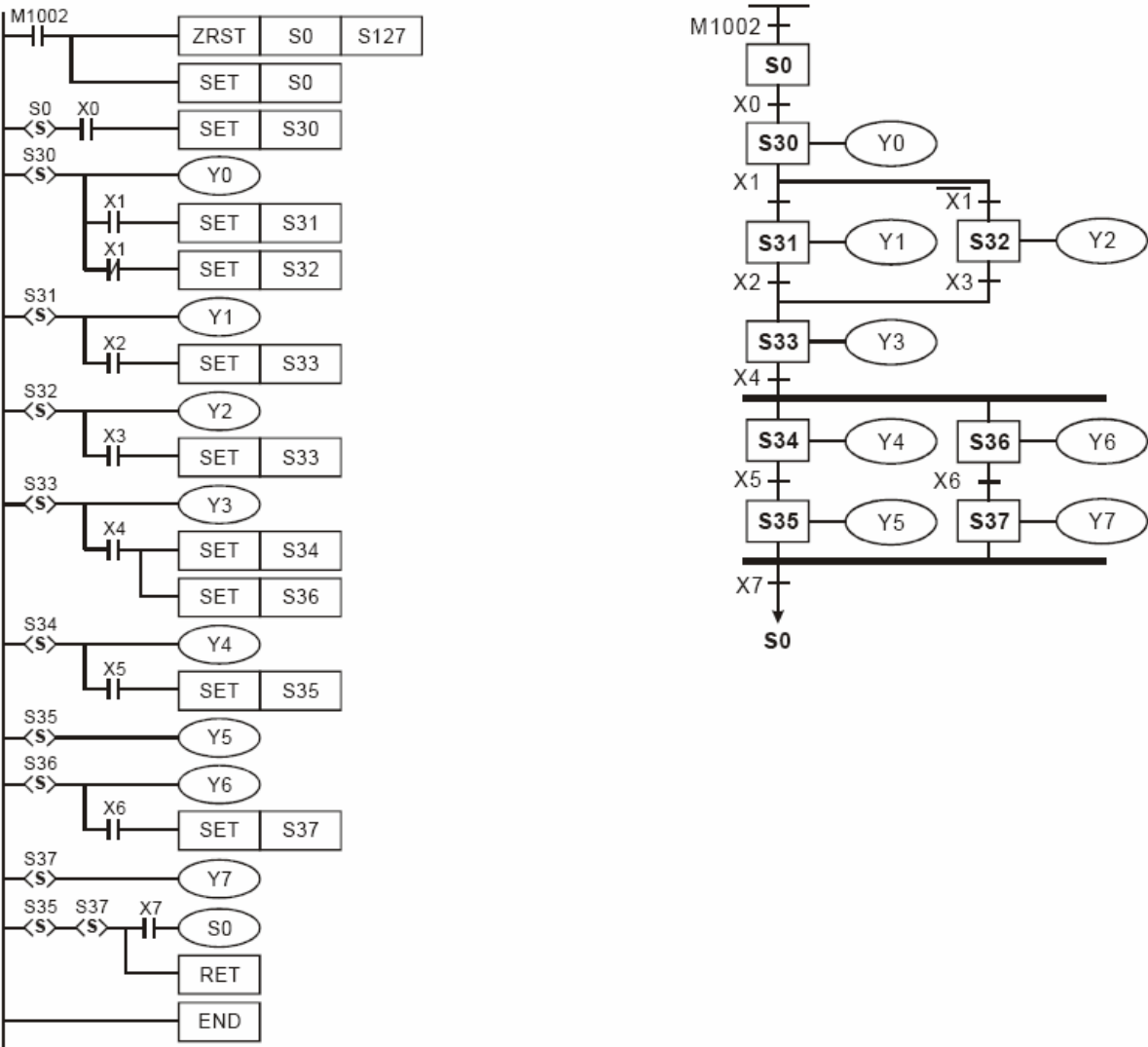
Пример процесса с параллельным разветвлением и селективной сборкой:



Пример 1 процесса с комбинацией параллельного разветвления и сборки и селективного разветвления и сборки:



Пример 2 процесса с комбинацией параллельного разветвления и сборки и селективного разветвления и сборки:



4.6. Команда IST

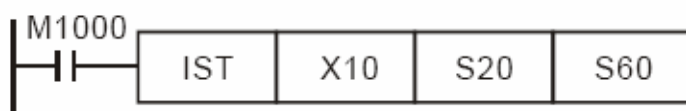
API	IST	<div><div>S</div><div>D1</div><div>D2</div></div>	Ручное/автоматическое управление	ПЛК		
				ES/EX/SS	SA/SX	EH
60				+	+	+

Опе-ранд	Биты				Слова										16-ти битная инструкция (7 шагов). IST Непрерывное выполнение. 32-х битная инструкция ---	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S	*	*	*													
D1				*												
D2				*												
<div>Примечания: Операнд S может занимать 8 устройств. Используемый диапазон операндов D1 и D2: S20 – S899 и D2 > D1. Команда IST в программе может использоваться только один раз.</div>																Флаги: M1040 – M1047

Описание:

- 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).
- В (S) определяется область управляющих входов. В качестве управляющих входов могут применяться операнды X, Y или M. Они указывают стартовые адреса областей операндов. В (D1) и (D2) определяется область операндов шаговых состояний для шаговых цепей автоматического режима работы. При этом должно выполняться условие: $(D1) < (D2)$.

Пример применения:

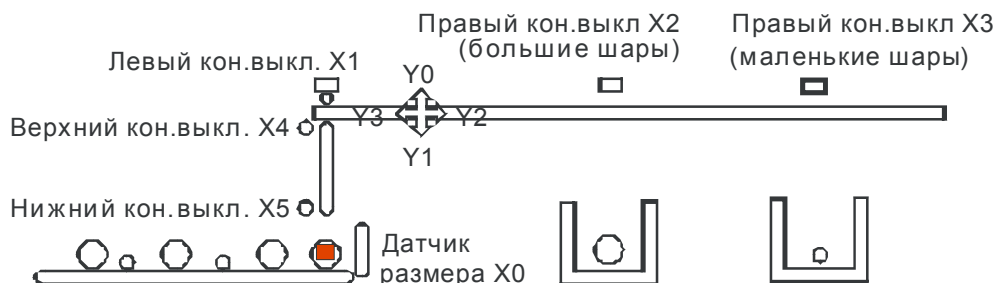


- Операнд (S) определяет управляющие входы, которые служат для непосредственного управления процессом работы. Ими выбирается нужный режим работы:
 - X10: ручной (наладочный) режим работы
 - X11: возврат в исходную позицию
 - X12: шаговый режим работы
 - X13: выполнение одного цикла
 - X14: автоматический режим работы
 - X15: кнопка возврата в исходную позицию
 - X16: кнопка запуска автоматического режима работы
 - X17: кнопка останова автоматического режима работы
- Следующие специальные реле благодаря IST-инструкции влияют или управляют обработкой шаговых цепей:
 - M1040: запрещение переходов
 - M1041: старт последующего перехода
 - M1042: импульс запуска
 - M1047: разрешение отображения состояния шагов в регистрах D1040 – D1047
- Операнды шаговых состояний S0...S2 предназначены (резервируются) для инициализации шаговых цепей:
 - S0: наладочный режим работы,
 - S1: перемещение в нулевую точку
 - S2: автоматический режим работы
- 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).

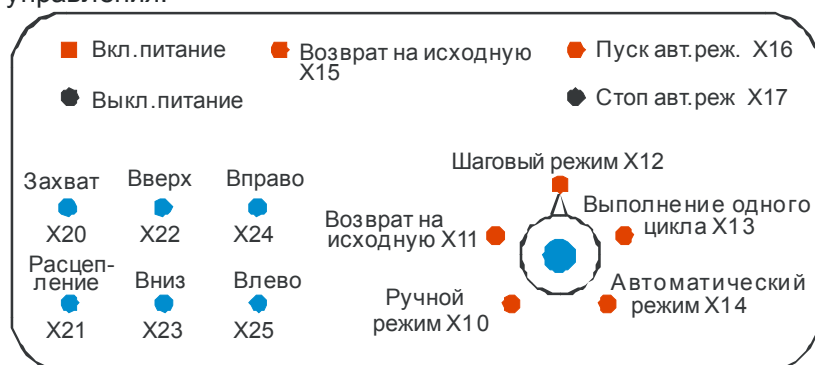
Пример управления роботом-манипулятором (с помощью IST-инструкции):

Задача: Надо произвести сортировку больших и маленьких шаров и поместить их в соответствующие коробки.

Движения робота-манипулятора: опустить манипулятор вниз, взять шар, поднять манипулятор с шаром, переместить манипулятор с шаром вправо, опустить манипулятор с шаром вниз, положить шар в коробку, поднять манипулятор, переместить манипулятор влево на исходную позицию.



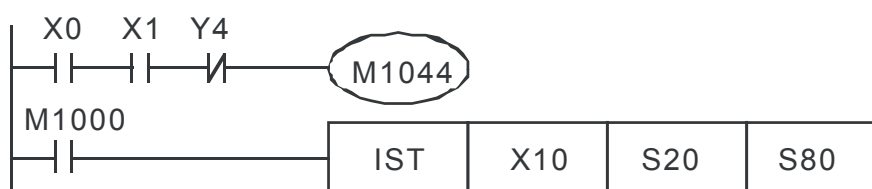
Пульт управления:

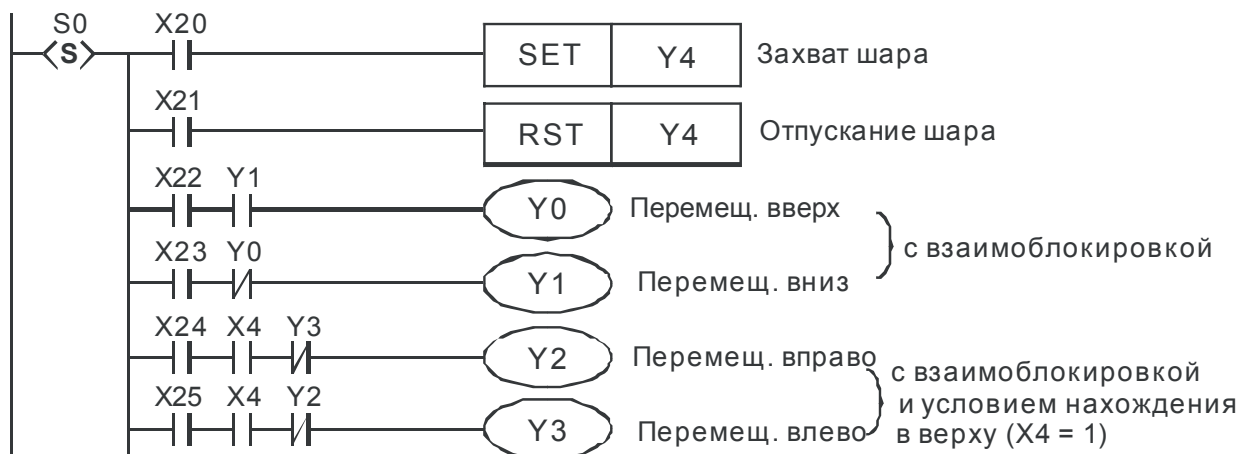


Описание датчиков и исполнительных устройств:

- Конечные выключатели: слева – X1; справа – X2 (для больших шаров) и X3 (для маленьких шаров); снизу – X5; сверху – X4
- X0 – датчик определения размера шара
- X10 – X14: переключатели выбора режима работы
- X15 – X17: кнопки пуска/стопа выполнения выбранного режима
- X20 – X25: кнопки управления манипулятором в ручном режиме
- Команды перемещения манипулятора: Y0 – вверх; Y1 – вниз; Y2 – вправо; Y3 – влево

Инициализация:

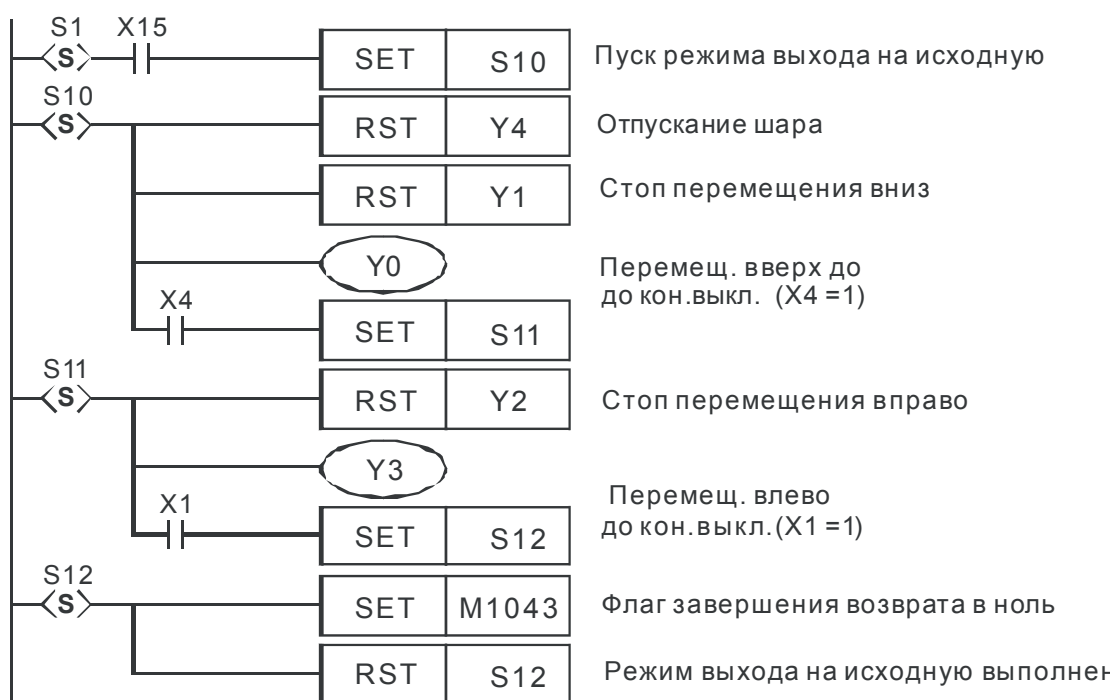


Ручной режим работы:**Режим выхода на исходную позицию:**

SFC-диаграмма:

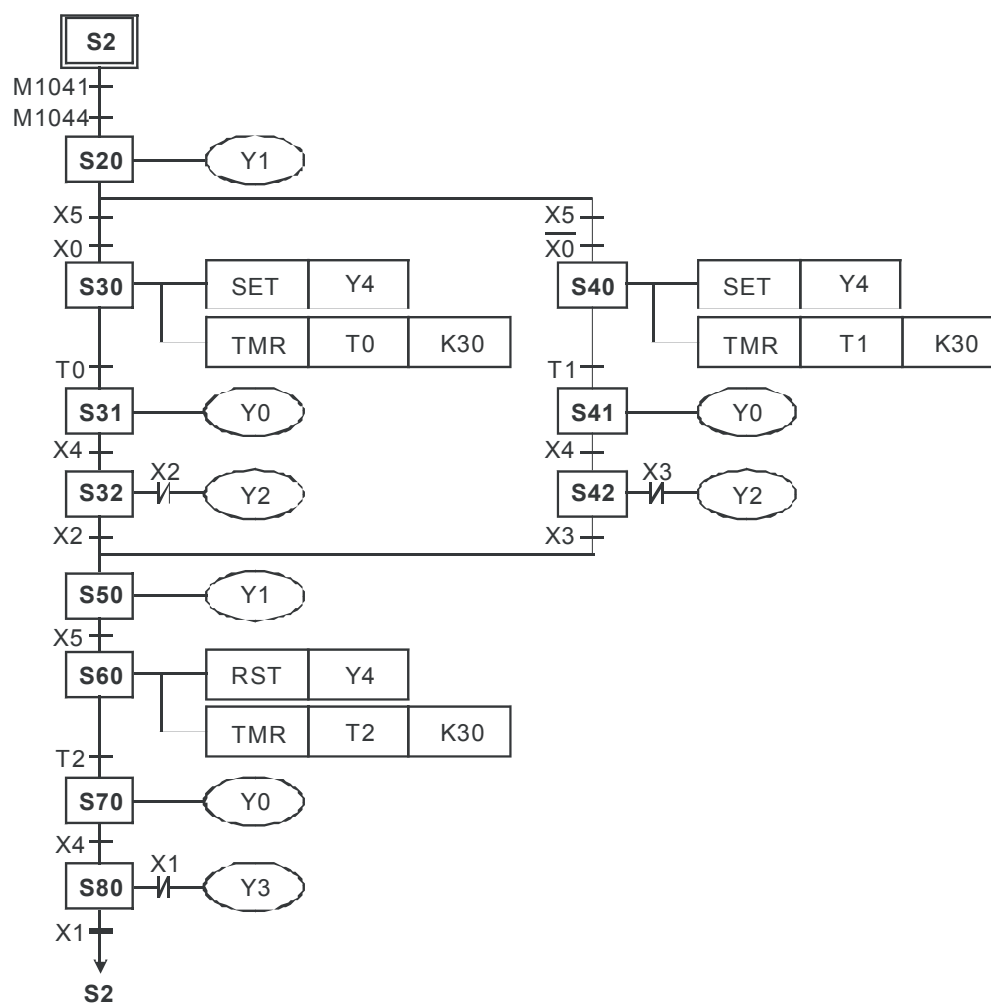


Релейно-контактная схема:

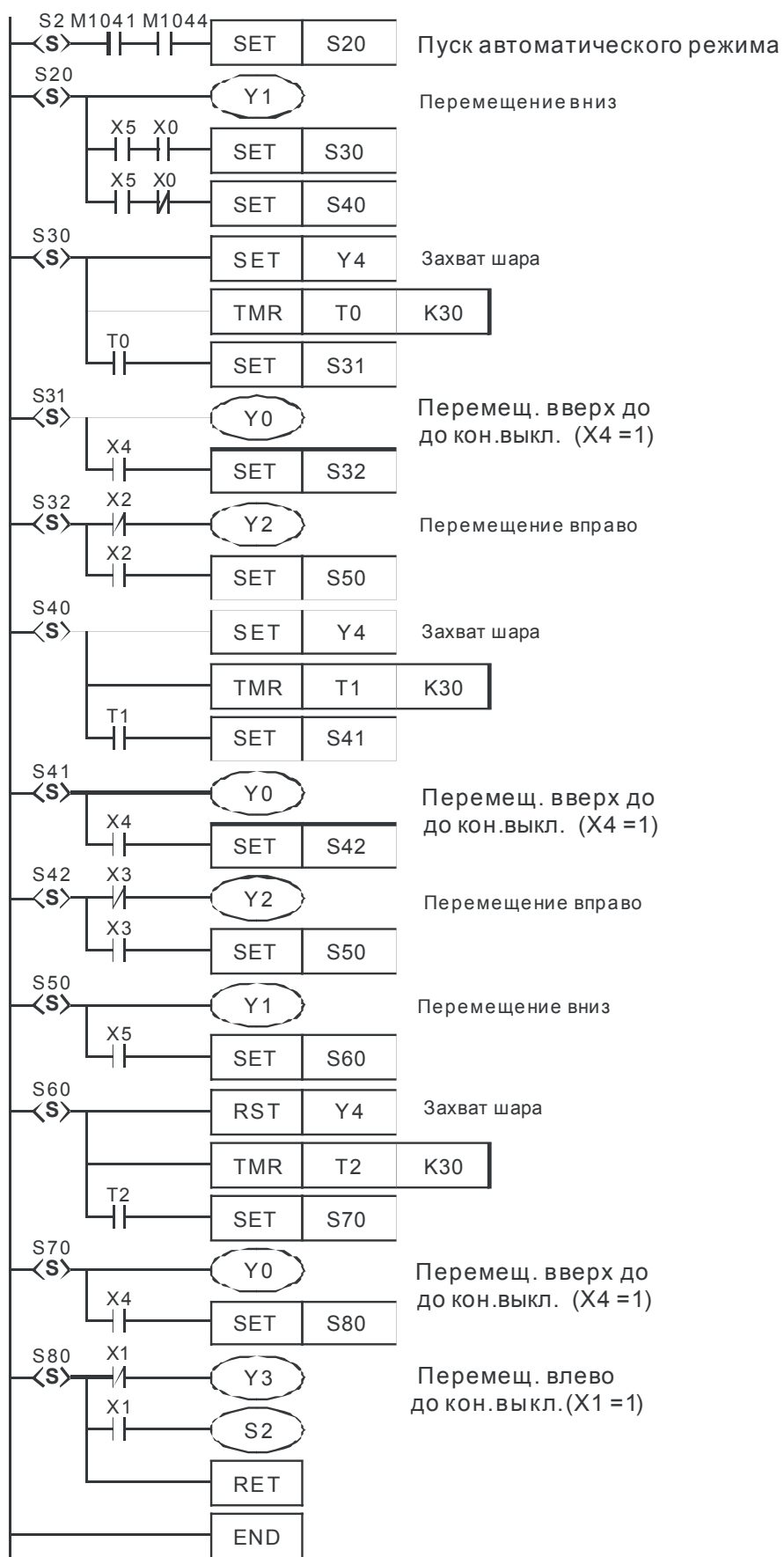


Автоматический режим (пошаговый/ один цикл/ непрерывный)

SFC-диаграмма:



Релейно-контактная схема:



5. ПРИКЛАДНЫЕ ИНСТРУКЦИИ

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

5.1. Перечень прикладных инструкций

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Работа с циклами	00	CJ	-	+	Переход к заданной строке	3	-	+	+	+
	01	CALL	-	+	Переход к подпрограмме	3	-	+	+	+
	02	SRET	-	-	Конец подпрограммы	1	-	+	+	+
	03	IRET	-	-	Конец обработки прерывания	1	-	+	+	+
	04	EI	-	-	Разрешение прерывания	1	-	+	+	+
	05	DI	-	-	Запрещение прерывания	1	-	+	+	+
	06	FEND	-	-	Конец главной программы	1	-	+	+	+
	07	WDT	-	+	Сброс сторожевого таймера	1	-	+	+	+
	08	FOR	-	-	Начало цикла	3	-	+	+	+
	09	NEXT	-	-	Конец цикла	1	-	+	+	+
Пересылка и сравнение	10	CMP	DCMP	+	Сравнение числовых данных	7	13	+	+	+
	11	ZCP	DZCP	+	Зонное сравнение числовых данных	9	17	+	+	+
	12	MOV	DMOV	+	Пересылка данных	5	9	+	+	+
	13	SMOV	-	+	Пересылка данных со смещением	11	-	-	+	+
	14	CML	DCML	+	Пересылка данных с их инвертированием	5	9	+	+	+
	15	BMOV	-	+	Пересылка блока данных	7	-	+	+	+
	16	FMOV	DFMOV	+	Пересылка в несколько адресов	7	13	+	+	+
	17	XCH	DXCH	+	Обмен данными	5	9	+	+	+
	18	BCD	DBCD	+	Преобразование числа из двоичного вида в двоично-десятичный	5	9	+	+	+
	19	BIN	DBIN	+	Преобразование из двоично-десятичного вида в двоичный	5	9	+	+	+
Арифметические инструкции	20	ADD	DADD	+	Сложение двух чисел	7	13	+	+	+
	21	SUB	DSUB	+	Вычитание двух чисел	7	13	+	+	+
	22	MUL	DMUL	+	Умножение двух чисел	7	13	+	+	+
	23	DIV	DDIV	+	Деление двух чисел	7	13	+	+	+
	24	INC	DINC	+	Инкрементирование (увеличение на 1)	3	5	+	+	+
	25	DEC	DDEC	+	Декрементирование (уменьшение на 1)	3	5	+	+	+
	26	WAND	DAND	+	Логическое умножение данных (И)	7	13	+	+	+
	27	WOR	DOR	+	Логическое сложение данных (ИЛИ)	7	13	+	+	+
	28	WXOR	DXOR	+	Исключающее «ИЛИ»	7	13	+	+	+
	29	NEG	DNEG	+	Отрицание	3	5	+	+	+

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Инструкции сдвигов	30	ROR	DROR	+	Кольцевой сдвиг вправо	5	9	+	+	+
	31	ROL	DROL	+	Кольцевой сдвиг влево	5	9	+	+	+
	32	RCR	DRCR	+	Кольцевой сдвиг вправо с установкой флага переноса в M1022	5	9	+	+	+
	33	RCL	DRCL	+	Кольцевой сдвиг влево с установкой флага переноса в M1022	5	9	+	+	+
	34	SFTR	-	+	Сдвиг значений битовых устройств вправо	9	-	+	+	+
	35	SFTL	-	+	Сдвиг значений битовых устройств влево	9	-	+	+	+
	36	WSFR	-	+	Сдвиг значений регистров вправо	9	-	-	+	+
	37	WSFL	-	+	Сдвиг значений регистров влево	9	-	-	+	+
	38	SFWR	-	+	Запись данных в стек	7	-	-	+	+
	39	SFRD	-	+	Чтение данных из стека	7	-	-	+	+
Операции с данными	40	ZRST	-	+	Групповой сброс операндов в заданном диапазоне	5	-	+	+	+
	41	DECO	-	+	Дешифратор 8 → 256 бит	7	-	+	+	+
	42	ENCO	-	+	Шифратор 256 → 8 бит	7	-	+	+	+
	43	SUM	DSUM	+	Сумма единичных битов в регистре	5	9	+	+	+
	44	BON	DBON	+	Опрос состояния бита регистра с установкой выхода	7	13	+	+	+
	45	MEAN	DMEAN	+	Среднее арифметическое	7	13	+	+	+
	46	ANS	-	-	Сигнализация тревоги с задержкой на включение	7	-	-	+	+
	47	ANR	-	+	Сброс тревожной сигнализации	1	-	-	+	+
	48	SQR	DSQR	+	Вычисление квадратного корня	5	9	+	+	+
	49	FLT	DFLT	+	Преобразование целого числа в число с плавающей точкой	5	9	+	+	+
Высокоскоростные инструкции	50	REF	-	+	Обновление состояния входов/выходов	5	-	+	+	+
	51	REFF	-	+	Изменение времени задержки входного фильтра	3	-	-	+	+
	52	MTR	-	-	Матричный ввод	9	-	-	+	+
	53	-	DHSCS	-	Установка состояния выхода при высокоскоростном счете	-	13	+	+	+
	54	-	DHSCR	-	Сброс состояния выхода при высокоскоростном счете	-	13	+	+	+
	55	-	DHSZ	-	Операция зонного сравнения при высокоскоростном счете	-	17	-	+	+
	56	SPD	-	-	Вычисление скорости	7	-	+	+	+
	57	PLSY	DPLSY	-	Выдача определенного числа импульсов	7	-	+	+	+
	58	PWM	-	-	Выдача импульсов с модуляцией ширины импульса (ШИМ)	7	-	+	+	+
	59	PLSR	DPLSR	-	Импульсный выход с ускорением/ замедлением	9	17	+	+	+

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Инструкции пользователя	60	IST	-	-	Ручное/автоматическое управление	7	-	+	+	+
	61	SER	DSER	+	Поиск данных стека	9	17	-	+	+
	62	ABSD	DABSD	-	Абсолютный многоуставочный счетчик	9	17	-	+	+
	63	INCD	-	-	Инкрементный многоуставочный счетчик	9	-	-	+	+
	64	TTMR	-	-	Обучающийся таймер	5	-	-	+	+
	65	STMR	-	-	Специальный таймер	7	-	-	+	+
	66	ALT	-	+	Импульсное реле (Т-триггер)	3	-	+	+	+
	67	RAMP	-	-	Линейное изменение сигнала (RAMP)	9	-	-	+	+
	69	SORT	-	-	Сортировка данных	11	-	-	+	+
Инструкции ввода/вывода	70	TKY	DTKY	-	Ввод с 10-ти кнопочной клавиатуры	7	13	-	+	+
	71	HKY	DHKY	-	Ввод с 16-ти кнопочной клавиатуры	9	17	-	+	+
	72	DSW	-	-	Ввод с цифрового переключателя	9	-	-	+	+
	73	SEGD	-	+	Дешифратор для 7-ми сегментного индикатора	5	-	+	+	+
	74	SEGL	-	-	Вывод на 7-ми сегментный индикатор	7	-	+	+	+
	75	ARWS	-	-	Ввод со стрелочной клавиатуры (←↑→↓)	9	-	-	+	+
	76	ASC	-	-	ASCII-конвертирование	11	-	-	+	+
	77	PR	-	-	Выдача ASCII-знаков по выходам	5	-	-	+	+
	78	FROM	DFROM	+	Чтение данных из модулей аналогового ввода/вывода (из CR регистров)	9	17	+	+	+
	79	TO	DTO	+	Запись данных в модули аналогового ввода/вывода (в CR регистры)	9	17	+	+	+
Инструкции последовательной коммуникации	80	RS	-	-	Последовательная передача и прием данных по RS-485	9	-	+	+	+
	81	PRUN	DPRUN	+	Пересылка данных 8-миричном формате	5	9	-	+	+
	82	ASCI	-	+	Преобразование ASCII в HEX	7	-	+	+	+
	83	HEX	-	+	Преобразование HEX в ASCII	7	-	+	+	+
	84	CCD	-	+	Расчет контрольной суммы	7	-	-	+	+
	85	VRRD	-	+	Чтение значения, заданного с потенциометра (встроенного в DVP-SA/EH или DVP-F6VR)	5	-	-	+	+
	86	VRSC	-	+	Масштаб значения потенциометра	5	-	-	+	+
	87	ABS	DABS	+	Абсолютное значение (модуль)	3	5	+	+	+
	88	PID	DPID	-	ПИД-регулятор	9	17	+	+	+

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Базовые инструкции	89	PLS	-	-	Создание импульса по переднему фронту	3	-	+	+	+
	90	LDP	-	-	Начало логического выражения с опросом по переднему фронту (импульс)	3	-	+	+	+
	91	LDF	-	-	Начало логического выражения с опросом по заднему фронту (импульс)	3	-	+	+	+
	92	ANDP	-	-	«И» с опросом по переднему фронту (импульс)	3	-	+	+	+
	93	ANDF	-	-	«И» с опросом по заднему фронту (импульс)	3	-	+	+	+
	94	ORP	-	-	«ИЛИ» с опросом по переднему фронту (импульс)	3	-	+	+	+
	95	ORF	-	-	«ИЛИ» с опросом по заднему фронту (импульс)	3	-	+	+	+
	96	TMR	-	-	Таймер (16 бит)	4	-	+	+	+
	97	CNT	DCNT	-	Счетчик (16 бит)	4	6	+	+	+
	98	INV	-	-	Инверсия	1	-	+	+	+
	99	PLF	-	-	Создание импульса по заднему фронту	3	-	+	+	+
Инструкции MODBUS	100	MODRD	-	-	Чтение данных MODBUS через RS-485	7	-	+	+	+
	101	MODWR	-	-	Запись данных MODBUS через RS-485	7	-	+	+	+
	102	FWD	-	-	Команда «ПУСК» вперед для привода VFD-A	7	-	+	+	+
	103	REV	-	-	Команда «ПУСК» реверсивно для привода VFD-A	7	-	+	+	+
	104	STOP	-	-	Команда «СТОП» для привода VFD-A	7	-	+	+	+
	105	RDST	-	-	Чтение текущего состояния привода VFD-A	5	-	+	+	+
	106	RSTEF	-	-	Команда «СБРОС» для привода VFD-A	5	-	+	+	+
	107	LRC	-	+	Расчет контрольной суммы LRC	7	-	+	+	+
	108	CRC	-	+	Расчет контрольной суммы CRC	7	-	+	+	+
	109	SWRD	-	+	Чтение значения с карты DVP-F8ID (8 DIP переключателей)	3	-	-	+	+

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Инструкции чисел с плавающей запятой	110	-	DECMF	+	Сравнение двух чисел с плавающей запятой	-	13	+	+	+
	111	-	DEZCP	+	Зонное сравнение двух чисел с плавающей запятой	-	17	+	+	+
	116	-	DRAD	+	Перевод градусов в радианы	-	9	-	+	+
	117	-	DDEG	+	Перевод радианов в градусы	-	9	-	+	+
	118	-	DEBCD	+	Перевод двоичного числа с плавающей запятой в десятичное с плавающей запятой	-	9	+	+	+
	119	-	DEBIN	+	Перевод десятичного числа с плавающей запятой в двоичное с плавающей запятой	-	9	+	+	+
	120	-	DEADD	+	Сложение чисел с плавающей запятой	-	13	+	+	+
	121	-	DESUB	+	Вычитание чисел с плавающей запятой	-	13	+	+	+
	122	-	DEMUL	+	Умножение чисел с плавающей запятой	-	13	+	+	+
	123	-	DEDIV	+	Деление чисел с плавающей запятой	-	13	+	+	+
	124	-	DEXP	+	Вычисление операции с экспонентой в формате с плавающей запятой	-	9	+	+	+
	125	-	DLN	+	Вычисление логарифма натурального в формате с плавающей запятой	-	9	+	+	+
	126	-	DLOG	+	Вычисление логарифма в формате с плавающей запятой	-	13	+	+	+
	127	-	DESQR	+	Вычисление корня квадратного в формате с плавающей запятой	-	9	+	+	+
	128	-	DPOW	+	Возведение числа в степень в формате с плавающей запятой	-	13	+	+	+
	129	INT	DDINT	+	Преобразование числа с плавающей запятой в целое	5	9	+	+	+
	130	-	DSIN	+	Вычисление синуса	-	9	+	+	+
	131	-	DCOS	+	Вычисление косинуса	-	9	+	+	+
	132	-	DTAN	+	Вычисление тангенса	-	9	+	+	+
	133	-	DASIN	+	Вычисление арксинуса	-	9	-	+	+
	134	-	DACOS	+	Вычисление арккосинуса	-	9	-	+	+
	135	-	DATAN	+	Вычисление арктангенса	-	9	-	+	+
	136	-	DSINH	+	Вычисление гиперболического синуса	-	9	-	+	+
	137	-	DCOSH	+	Вычисление гиперболического косинуса	-	9	-	+	+
	138	-	DTANH	+	Вычисление гиперболического тангенса	-	9	-	+	+

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Дополнительные инструкции	143	DELAY	-	+	Задержка выполнения			-	+	+
	144	GPWM	-	-	Общая команда генерации импульсов ШИМ	-	7	-	+	+
	145	FTC	-	-	Температурный контроллер (FTC)	-	9	-	+	+
	147	SWAP	DSWAP	+	Перестановка младшего и старшего байтов в регистре	3	5	+	+	+
	148	SWAP	DSWAP	+	Чтение данных из файловых регистров	7	13	-	+	+
	149	SWAP	DSWAP	+	Запись данных в файловые регистры	7	13	-	+	+
	150	MODRW	-	-	Чтение/запись данных MODBUS через RS-485	11	-	+	+	+
	151	PWD	-	-	Импульсная ловушка	5	-	-	-	+
	152	RTMU	-	-	Начало подпрограммы обработки временного прерывания	5	-	-	-	+
	153	RTMD	-	-	Конец подпрограммы обработки временного прерывания	3	-	-	-	+
Инструкции позиционирования	154	RAND	-	+	Генератор случайных чисел	9	-	-	+	+
	155	ABSR	DABSR	-	Чтение абсолютного текущего положения	7	13	-	-	+
	156	ZRN	DZRN	-	Выход в исходную позицию	9	17	-	-	+
	157	PLSV	DPLSV	-	Импульсный выход с заданием частоты и направления вращения серводвигателя	7	13	-	-	+
	158	DRVI	DDRVI	-	Команда перемещения в заданное положение в относительных координатах	9	17	-	-	+
	159	DRVA	DDRVA	-	Команда перемещения в заданное положение в абсолютных координатах	9	17	-	-	+
Инструкции реального времени	160	TCMP	-	+	Сравнение времени	11	-	-	+	+
	161	TZCP	-	+	Сравнение времени в заданном диапазоне	9	-	-	+	+
	162	TADD	-	+	Сложение времени	7	-	-	+	+
	163	TSUB	-	+	Вычитание времени	7	-	-	+	+
	166	TRD	-	+	Чтение текущего значения часов реального времени	3	-	-	+	+
	167	TWR	-	+	Изменение значения часов реального времени	3	-	-	+	+
	169	HOURL	DHOUR	-	Часовой счетчик времени наработки	7	13	-	+	+
Код Грея	170	GRY	DGRY	+	Преобразование целого числа в код Грея	5	9	-	+	+
	171	GBIN	DGBIN	+	Преобразование кода Грея в целое число	5	9	-	+	+

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Операции с матрицами	180	MAND	-	+	Логическое умножение матриц (И)	9	-	-	+	+
	181	MOR	-	+	Логическое сложение матриц (ИЛИ)	9	-	-	+	+
	182	MXOR	-	+	Исключающее «ИЛИ» для матриц	9	-	-	+	+
	183	MXNR	-	+	Исключающее «НЕ-ИЛИ» для матриц	9	-	-	+	+
	184	MINV	-	+	Инверсия матрицы	7	-	-	+	+
	185	MCMP	-	+	Сравнение матриц	9	-	-	+	+
	186	MBRD	-	+	Чтение битов в матрице	7	-	-	+	+
	187	MBWR	-	+	Запись битов в матрицу	7	-	-	+	+
	188	MBS	-	+	Сдвиг битов в матрице	7	-	-	+	+
	189	MBR	-	+	Кольцевой сдвиг битов в матрице	7	-	-	+	+
	190	MBC	-	+	Счетчик битов	7	-	-	+	+
	196	HST	DHST	+	Высокоскоростной таймер	3	3	-	+	+
Логические операции контактного типа	215	LD&	DLD&	-	Контакт замкнут, если S1 & S2 ≠ 0	5	9	-	+	+
	216	LD	DLD	-	Контакт замкнут, если S1 S2 ≠ 0	5	9	-	+	+
	217	LD^	DLD^	-	Контакт замкнут, если S1 ^ S2 ≠ 0	5	9	-	+	+
	218	AND&	DAND&	-	Последовательный контакт замкнут, если S1 & S2 ≠ 0	5	9	-	+	+
	219	AND	DAND	-	Последовательный контакт замкнут, если S1 S2 ≠ 0	5	9	-	+	+
	220	AND^	DAND^	-	Последовательный контакт замкнут, если S1 ^ S2 ≠ 0	5	9	-	+	+
	221	OR&	DOR&	-	Параллельный контакт замкнут, если S1 & S2 ≠ 0	5	9	-	+	+
	222	OR	DOR	-	Параллельный контакт замкнут, если S1 S2 ≠ 0	5	9	-	+	+
	223	OR^	DOR^	-	Параллельный контакт замкнут, если S1 ^ S2 ≠ 0	5	9	-	+	+

Тип	API	Инструкция		P*	Функция	Число шагов		ПЛК		
		16 бит	32 бит			16 бит	32 бит	ES/EX/SS	SA/SX	EH
Операции сравнения контактного типа	224	LD=	DLD=	-	Контакт замкнут, если S1 = S2	5	9	+	+	+
	225	LD>	DLD>	-	Контакт замкнут, если S1 > S2	5	9	+	+	+
	226	LD<	DLD<	-	Контакт замкнут, если S1 < S2	5	9	+	+	+
	228	LD<>	DLD<>	-	Контакт замкнут, если S1 ≠ S2	5	9	+	+	+
	229	LD≤	DLD≤	-	Контакт замкнут, если S1 ≤ S2	5	9	+	+	+
	230	LD≥	DLD≥	-	Контакт замкнут, если S1 ≥ S2	5	9	+	+	+
	232	AND=	DAND=	-	Последовательный контакт замкнут, если S1 = S2	5	9	+	+	+
	233	AND>	DAND>	-	Последовательный контакт замкнут, если S1 > S2	5	9	+	+	+
	234	AND<	DAND<	-	Последовательный контакт замкнут, если S1 < S2	5	9	+	+	+
	236	AND<>	DAND<>	-	Последовательный контакт замкнут, если S1 ≠ S2	5	9	+	+	+
	237	AND≤	DAND≤	-	Последовательный контакт замкнут, если S1 ≤ S2	5	9	+	+	+
	238	AND≥	DAND≥	-	Последовательный контакт замкнут, если S1 ≥ S2	5	9	+	+	+
	240	OR=	DOR=	-	Параллельный контакт замкнут, если S1 = S2	5	9	+	+	+
	241	OR>	DOR>	-	Параллельный контакт замкнут, если S1 > S2	5	9	+	+	+
	242	OR<	DOR<	-	Параллельный контакт замкнут, если S1 < S2	5	9	+	+	+
	244	OR<>	DOR<>	-	Параллельный контакт замкнут, если S1 ≠ S2	5	9	+	+	+
	245	OR≤	DOR≤	-	Параллельный контакт замкнут, если S1 ≤ S2	5	9	+	+	+
	246	OR≥	DOR≥	-	Параллельный контакт замкнут, если S1 ≥ S2	5	9	+	+	+

* P – возможно импульсное выполнение команды (только для SA/SX/EH). К мнемонике инструкции добавляется символ "P": например, MOV**P**

5.2. Структура прикладных инструкций

5.2.1. Структура таблиц описания прикладных инструкций.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
API		DECO	☺	S D n	Дешифратор 8 → 256 бит	DVP-
41			P			ES/EX/SS SA/SX EH
						+ + +

(15)	Оп- еранд	Биты				Слова						16-ти битная инструкция (7 шагов).				(8)
(14)	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DECO - Непрерывное выполнение.
(13)	S	*	*	*	*	*					*	*	*	*	*	DECOP – Импульс. выполн.
	D		*	*	*						*	*	*	*	*	
	n				*	*							*			
(12)	Примечания: Когда D битовый операнд, n = 1 – 8 Когда D словный операнд, n = 1 – 4 Диапазон операндов см. в технических характеристиках для каждой серии DVP															32-х битная инструкция --- Флаги: нет (10)

(1) Номер прикладной инструкции

(2) Верхняя ячейка показывает возможность 16-ти разрядной обработки инструкции. Если стоит "-", то 16-ти разрядная обработка инструкции невозможна.

Нижняя ячейка показывает возможность 32-х разрядной обработки инструкции. Если стоит символ "D", то 32-х разрядная обработка инструкции возможна, если к мнемонике добавить символ "D", например, **DMOV**.

(3) Мнемоника (имя) инструкции

(4) Символ "☺" в верхней ячейке говорит о том, что инструкцию рекомендуется использовать в импульсном выполнении.

Символ "P" в нижней ячейке говорит о том, что инструкция может использоваться в импульсном выполнении, если к мнемонике инструкции добавляется символ "P": например, **MOV_P**.

(5) Формат операндов прикладной инструкции

(6) Краткое описание инструкции

(7) Модели DVP в которых может использоваться инструкция

(8) Здесь указывается количество шагов программы, которые требуются для выполнения 16-ти битной инструкции, а также мнемоники непрерывного и импульсного выполнения инструкции

(9) Здесь указывается количество шагов программы, которые требуются для выполнения 32-х битной инструкции, а также мнемоники непрерывного и импульсного выполнения инструкции

(10) Здесь указываются связанные с инструкцией флаги

(11) Символ * показывает, что операнд может использовать индексный регистр

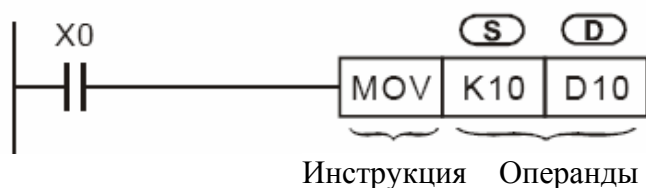
(12) Примечание

- (13) Символ * показывает, какие операнды могут использоваться в инструкции
- (14) Название операнда
- (15) Тип операнда

5.2.2. Ввод прикладных инструкций.

Прикладная инструкция имеет наименование (мнемонику) и относящийся к ней API-номер (API - номер функции), которое применяется при программировании на языке списка инструкций (IL). В зависимости от системы программирования можно применять или имя инструкции или ее номер (например, при вводе с программатора).

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



Инструкция передает значение операнда S операнду D.

	Источники данных: если их больше одного, то обозначаются S1, S2, S3, ...
	Конечные данные: если их больше одного, то обозначаются D1, D2, D3, ...
Если операнд представлен константой, то обозначения могут быть следующими: m, m1, m2, n, n1, n2	

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.

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.

5.2.3. 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 в обзорной таблице, 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).



Пример программирования 16-ти битной инструкции:

Когда X0=1, в регистр D10 будет записано число K10.



Пример программирования 32-х битной инструкции:

Когда X1=1, данные из регистров D10, D11 будут записаны в регистры D20, D21.

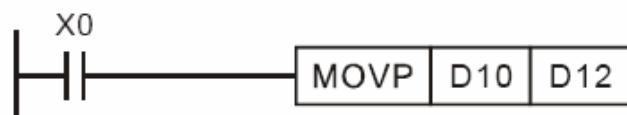
5.2.4. Непрерывное и импульсное выполнение инструкции

Имеется две возможности исполнения инструкций:

- Исполняемый сигнал может быть статическим. При включении сигнала словная инструкция выполняется в постоянно в каждом скане.
- 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) или с помощью командного параметра "P", который позволяет опознавать изменение управляющего сигнала при возрастающем фронте. Функция параметра "P" соответствует инструкции "PLS" базового набора команд.

Некоторые инструкции рекомендуется выполнять в импульсном режиме (знак ☺), например: INC, DEC, MOV

Выполнение инструкции по импульсному сигналу:



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.

Выполнение инструкции по статическому сигналу:



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.

5.2.5. Типы операндов

Битовый операнд (X, Y, M, S) может принимать два состояния сигнала ("0" и "1"). Его состояние сигнала может определяться тем самым с помощью бита (0 и 1).

Пословный операнд (T, C, D, E, F) может принимать информационное состояние, которое состоит из нескольких бит (числовое значение данных). При этом 8 бит составляют байта и 2 байта - слово данных (16 бит).

Возможна пословная обработка битовых операндов: KnX, KnY, KnM, KnS (см. главу 5.3)

5.2.6. Флаги

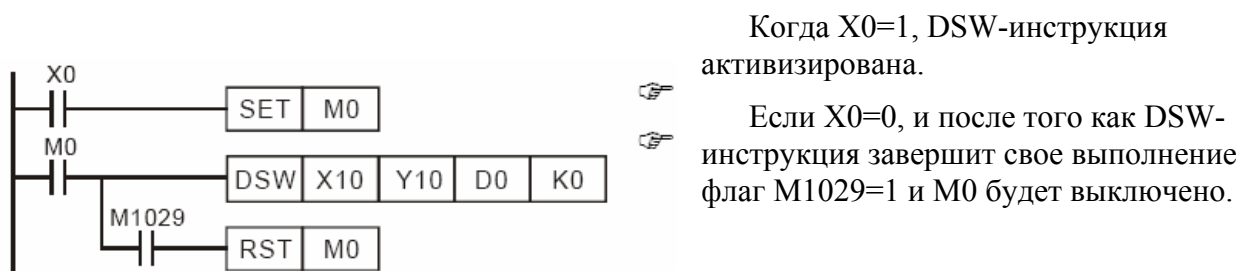
При обработке некоторых прикладных инструкций автоматически из ПЛК включаются или отключаются различные флаги (специальные реле). Используемый флаг показывает определенное состояние программы (например, превышение допустимой числовой области данных при выполнении словной инструкции). Этот флаг каждый 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.

Например: M1020 - флаг нуля, который включается если результат сложения или вычитания равен нулю;

M1021 - флаг заимствования (Borrow), если результат вычитания меньше самого малого значения ;

M1022 флаг переноса (Carry), включается при передаче значения числа, при суммировании или при передаче данных, при выполнении инструкции сдвига;

M1029 - флаг завершения выполнения инструкции



Обзор всех флагов и их значение находятся в главе 2.10.

При ошибочном программировании словных инструкций или адресов операндов запоминаются сообщения об ошибках в регистре данных ошибок и выставляются флаги (M1067, M1068). Глава описания ошибок 2.12. содержит детальный обзор всех регистров данных ошибок и сообщений об ошибках.

5.2.7. Ограничение числа использования инструкции в программе

Следующие инструкции в некоторых моделях DVP можно использовать ограниченное число раз в одной программе:

1. Не более одного раза:

PWM (API 58) – в моделях ES/EX/SS

SEGL (API 74) – в моделях ES/EX/SS

IST (API 68) – во всех моделях

PID (API 88) – в моделях ES/EX/SS/SA/SX

2. Не более двух раз:

PLSY (API 57) – в моделях ES/EX/SS

PLSR (API 59) – в моделях ES/EX/SS

SEGL (API 74) – в модели EH

PR (API 77) – в моделях SA/SX/EH

3. Не более четырех раз:

HOURL (API 169) – в моделях SA/SX

4. Не более восьми раз:

TTMR (API 64) – в моделях SA/SX

5. Не более четырех раз при одновременном использовании:

DHSCS (API 53) и DHSCR (API 54) – в моделях ES/EX/SS

6. Не более шести раз при одновременном использовании:

DHSCS (API 53), DHSCR (API 54) и DHSZ (API 55) – в моделях SA/SX

5.2.8. Ограничение одновременного выполнения инструкции в программе

Следующие инструкции в DVP можно использовать неограниченное число раз в одной программе, однако ограничено число их одновременного выполнения:

1. Не более одной инструкции одновременно:

API 52 (MTR), API 56 (SPD), API 62 (ABSD), API 63 (INCD), API 69 (SORT), API 70 (TKY), API 71 (HKY), API 72 (DSW) (EP models), API 74 (SEGL) (в моделях SA/SX), API 75 (ARWS), API 80 (RS), API 100 (MODRD), API 101 (MODWR), API 102 (FWD), API 103 (REV), API 104 (STOP), API 105 (RDST), API 06 (RSTEF), API 150 (MODRW), API 151 (PWD)

2. Не более двух инструкций одновременно:

API 57 (PLSY), API 58 (PWM), API 59 (PLSR), API 72 (DSW) (в модели EH)

3. Не более четырех инструкций одновременно:

API 169 (HOURL) (в модели EH)

4. Не более восьми инструкций одновременно:

API 64 (TTMR) (в модели EH)

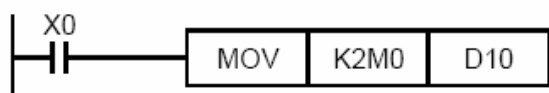
5. В контроллерах EH не ограничено число использования инструкций высокоскоростного счета: DHSCS (API 53), DHSCR (API 54) и DHSZ (API 55), однако ограничено число их одновременного выполнения. Инструкции DHSCS и DHSCR будут использовать по одной единице памяти, а инструкция DHSZ – две единицы памяти. Надо учесть, что при выполнении одновременно не должно использоваться более восьми

единиц памяти. При превышении данного значения инструкции высокоскоростного счета расположенные в программе ниже (тех, что используют 8 ед. памяти) будут проигнорированы.

5.3. Обработка прикладных инструкций

5.3.1. Пословная обработка битовых операндов

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



Когда X=1, содержимое битов M0 – M7 будет записано в регистр D10 (разряды 0 - 7), а разряды 8 – 15 будут =0

Количество адресов битовых операндов, которые должны сработать от пословной инструкции, определяются указанием константы K. 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 K1 до K4.

Длина блока	Количество адресов	Диапазон значений
K1	4	0 ... 15
K2	8	0 ... 255
K3	12	0 ... 4095
K4	16	-32768 ... 32767

Для 32-х битовых инструкций длина блоков лежит в области от K1 до K8.

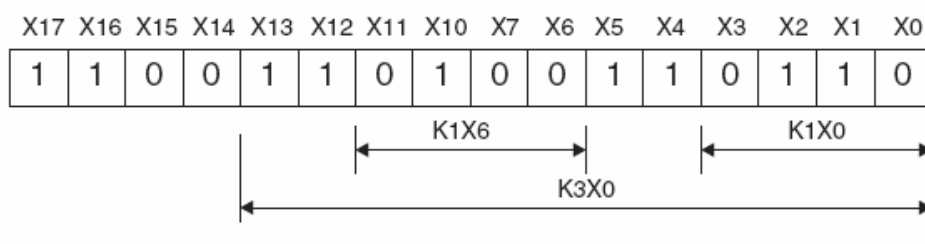
Длина блока	Количество адресов	Диапазон значений
K1	4	0 ... 15
K2	8	0 ... 255
K3	12	0 ... 4 095
K4	16	0 ... 65 535
K5	20	0 ... 1 048 575
K6	24	0 ... 167 772 165
K7	28	0 ... 268 435 455
K8	32	-2 147 483 648 ... 2 147 483 647

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

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

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

Пример 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

5.3.2. Обработка чисел с плавающей запятой

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

Инструкции для работы с числами с плавающей запятой:

API 49 (FLT),	API 110 (D ECMP),	API 111 (D EZCP),	API 116 (D RAD),
API 117 (D DEG),	API 118 (D EBCD),	API 119 (D EBIN),	API 120 (D EADD),
API 121 (D ESUB),	API 122 (D EMUL),	API 123 (D EDIV),	API 124 (D EXP),
API 125 (D LN),	API 126 (D LOG),	API 127 (D ESQR),	API 128 (D POW)
API 129 (INT)	API 130 (D SIN)	API 131 (D COS)	API 132 (D TAN)
API 133 (D ASIN)	API 134 (D ACOS)	API 135 (D ATAN)	API 136 (D SINH)
API 137 (D COSH)	API 138 (D TANH)		

- Двоичный формат чисел с плавающей запятой

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.



Формат: $\pm \text{Мантисса} \times 2^{\text{Экспонента}}$ $= (-1)^S \times 2^{E-B} \times 1.M; B = 127$

Диапазон чисел с плавающей запятой:

$$\pm 2^{-128} \dots \pm 2^{128}, \text{ или } \pm 1.1755 \times 10^{-38} \dots \pm 3.4028 \times 10^{38}$$

Пример представления десятичного числа 23 в формате с плавающей запятой:

Шаг 1: преобразование в двоичный формат: $23 = 10111$

Шаг 2: нормализация: $10111 = 1.0111 \times 2^4$, 0111 – мантисса, 4 для экспоненты

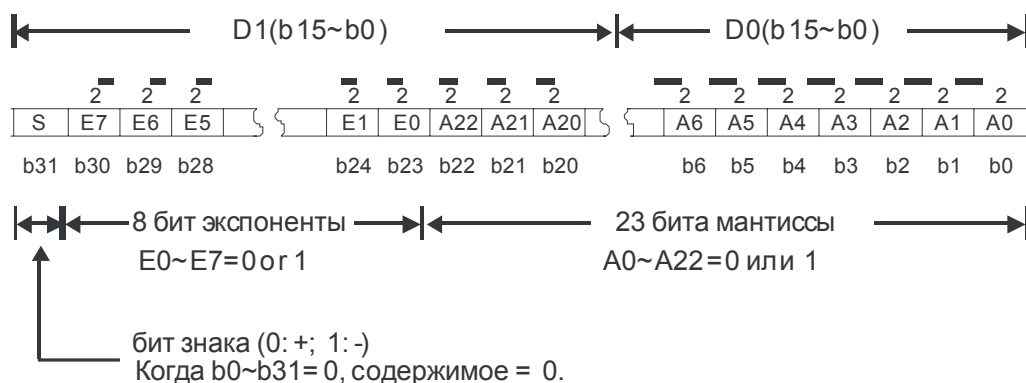
Шаг 3: получение экспоненты: $E - B = 4 \Rightarrow E - 127 = 4 \Rightarrow E = 131 = 10000011_2$

$$0 \ 10000011 \ 01110000000000000000000_2 = 41B80000_{16}$$

Пример представления десятичного числа -23 в формате с плавающей запятой:

$$1 \ 10000011 \ 01110000000000000000000_2 = C1B80000_{16}$$

Число с плавающей запятой располагается в двух регистрах (например, D0, D1):



- Десятичный формат чисел с плавающей запятой

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

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

Область значений:

- Мантиссы: $\pm 1000 \dots \pm 9999$, или 0
- Экспонента: $-41 \dots +35$

Например, скорость света:

- как десятичное число: 299792458 м/с
- в десятичном формате с плавающей запятой: 2998×10^5 м/с

Здесь 2998 является мантиссой и 5 - экспонентой. В регистре данных число сохраняется, например, в форме $D0 \times 10^{D121}$.

Инструкции для работы в десятичном формате с плавающей запятой:

DEBCD – преобразование из двоичного формата с плавающей запятой в десятичный формат с плавающей запятой;

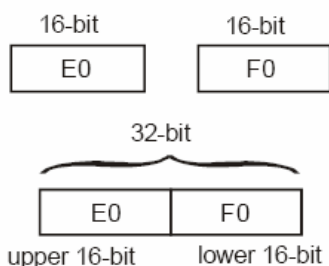
DEBIN – преобразование из десятичного формата с плавающей запятой в двоичный формат с плавающей запятой

При выполнении с этими инструкциями работают флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)

5.4. Индексные регистры E, F

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

В контроллерах ES/EX/SS есть два индексных регистра: E и F. В контроллерах SA/SX есть 8 индексных регистров: E0 – E3 и F0 – F3. . В контроллерах EN есть 16 индексных регистров: E0 – E7 и F0 – F7.



Индексный регистр является 16-ти битовым регистром.

В 32-х битовых инструкциях индексные регистры E и F применяются комбинированно. F содержит 16 младших бит, E запоминает 16 старших бит (E0, F0), (E1, F1), (E2, F2)...(E7, F7). В качестве адреса назначения указывается индексный регистр F. Индексный регистр не может самостоятельно индизироваться.

Пример использования индексного регистра:



E0=8 F0=14
 20+8=28 10+14=24
 K28 → D24

Индексные регистры могут использоваться для операций передачи и сравнения данных совместно с словными операндами (KnX, KnY, KnM, KnS, D, T, C) и битовыми операндами (X, Y, M, S).

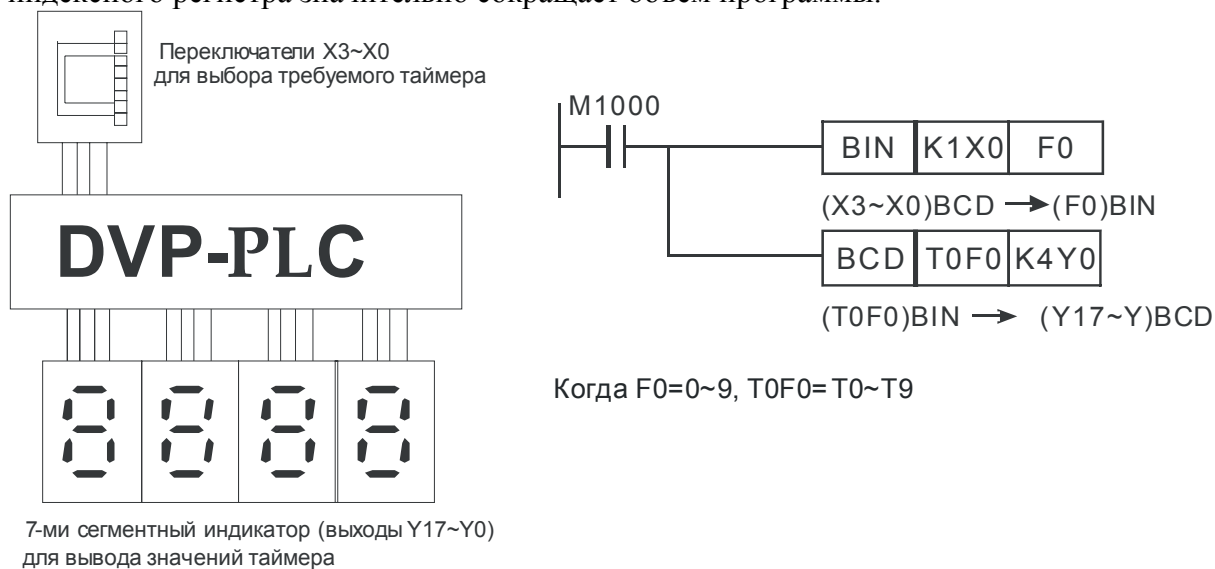
В контроллерах серии EN можно индексировать так же и константы (K, H).

В структурной таблице символ * показывает, что операнд может использовать индексный регистр.


При индексировании констант в командном режиме WPLSoft необходимо использовать символ @. Например: MOV K10@E0 D0F0.

Пример программы с применением индексного регистра:

С помощью переключателей X0 – X3 можно выбирать текущее значение одного из таймеров (T0 – T9) для вывода его на 7-ми сегментный индикатор. Использование индексного регистра значительно сокращает объем программы.



6. ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 00-49

API		CJ			Условный переход	DVP-		
00			P				ES/EX/SS	SA/SX
						+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (3 шага). CJ - Непрерывное выполнение. CJP – Импл. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
<u>Примечания:</u> В качестве операнда могут использоваться указатели P, которые могут индексироваться (E, F)																<u>32-х битная инструкция</u> --- Флаги: нет
Диапазон операндов:																
ES серия: S = P0...P63																
SA/EH серия: S = P0...P255																
Серия ES не поддерживает импульсное выполнение инструкции																

Функция

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

Описание

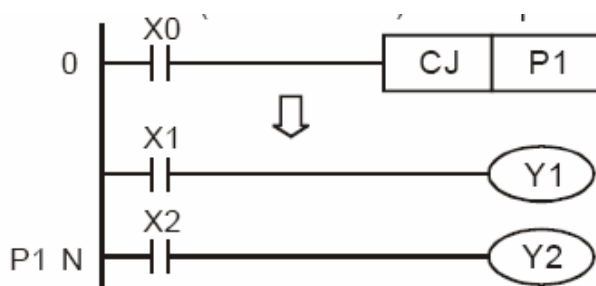
- Цель (конец) перехода определяется установкой указателя (точки перехода) в программе.
- Указание адреса конца перехода (Адреса точки) определяет, к какому указателю должен выполняться переход. CJ-инструкции могут использовать один адрес перехода многократно в программе, однако использование одинаковых указателей инструкциями CJ и CALL недопустимо. Это вызовет ошибку в программе.
- Обратный переход (вверх программы) также может выполняться внутри программы. Если входной сигнал для CJ-инструкции держится больше 200 мс, то появляется ошибка времени работы (Watchdog Timer).
- Имеется возможность дублирования записи выхода. При дублировании записи выходов следите за тем, чтобы оба выхода никогда не были активными в одно и тоже время. Это может привести к ошибочной отработке программы.
- Изменение состояния устройств внутри перехода CJ:
 1. Y, M, S сохраняют свое состояние, которое было до перехода CJ;
 2. Выполнение таймеров (100мс, 10 мс) прекратится (накопленное время сохраняется);
 3. Таймеры T192-199 и их рабочие контакты будут продолжать работать
 4. Высокоскоростные счетчики и их рабочие контакты будут продолжать работать
 5. Обычные счетчики прекратят работу (накопленное значение сохраняется)
 6. Если внутри подпрограммы перехода программируется инструкция сброса (отключения) для аккумулятивного таймера, то процесс сброса (стирание накопленного значения) имеет место тогда, когда перепрыгивается цепь схемы катушки счетчика

7. Прикладные инструкции (кроме API 53 DHSCS, API 54 DHSCR, API 55 DHSZ, API 56 SPD, API 57 PLSY, API 58 PWM, API 59 PLSR, API 157 PLSV, API 158 DRVI, API 159 DRVA) внутри перехода выполняться не будут.

Выполнение маркировки точки в программе

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

Пример применения



Если включается X0, то выполняется переход к точке P1 и строки программы между 0 и N выполняться не будут.

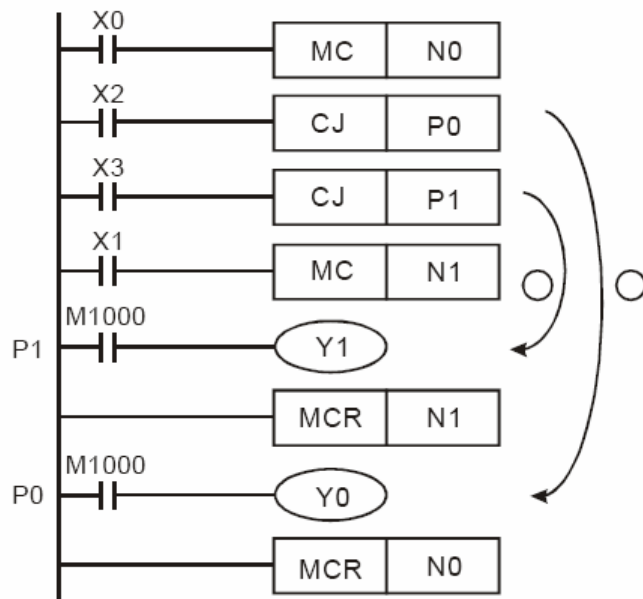
Если X0 выключено, то выполняется полностью.

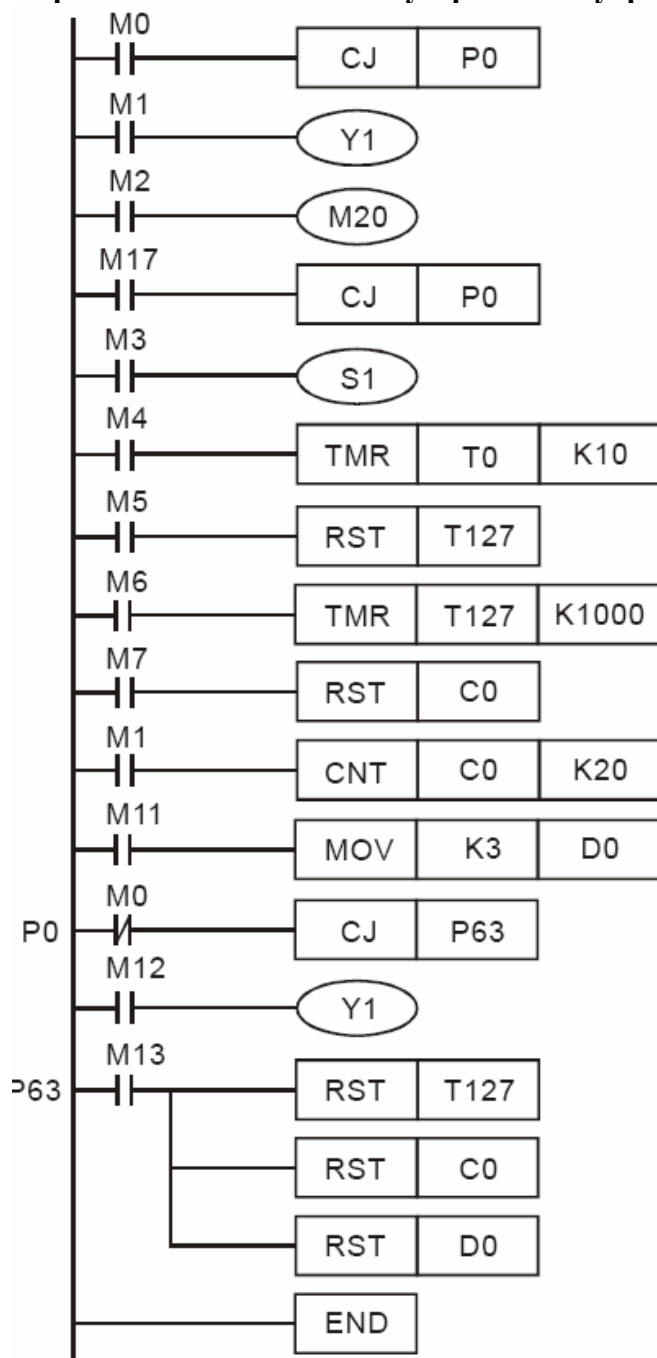
Переходы в области главного управления (Master-Control)

Процесс программирования при использовании CJ-инструкции вместе с MC- и MCR-инструкциями допускает 5 вариантов:

1. Перепрыгивание через область MC-MCR
2. Переход во внутреннюю область MC-MCR из внешней (P1 на рис.)
3. Переход внутри области MC-MCR
4. Переход из области MC-MCR
5. Переход из одной области MC-MCR в другую (только в моделях SA/SX/EH и ES версии 4.7 и выше)

На рис. переходы отмеченные "О" нельзя использовать в ES версии ниже 4.7




Пример изменения состояний устройств внутри пропущенной части программы:

Операнды	Состояние контактов перед переходом	Состояние контактов в течение перехода	Состояние катушек в течение перехода
Y, M, S	M1, M2, M3 выкл.	M1, M2, M3 выкл. → вкл.	Y1 (*), M20, S1 выкл.
	M1, M2, M3 вкл.	M1, M2, M3 вкл. → выкл.	Y1 (*), M20, S1 вкл.
Таймер (10мс, 100мс) (ES/SA/EH)	M4 выкл.	M4 выкл. → вкл.	Таймеры не активны
	M4 вкл.	M4 вкл. → выкл.	Счет остановлен. Накопленное время сохраняется и будет продолжено когда M0=0
Аккумулят. таймер (1мс, 10мс, 100мс)	M6 выкл.	M6 выкл. → вкл.	Таймер (T240) не активен

(SA/EH)	M6 вкл.	M6 вкл.→выкл.	Счет остановлен. Накопленное время сохраняется и будет продолжено когда M0=0
C0~C234	M7, X10 выкл.	M10 вкл./выкл.	Счетчик не активен
	M7 выкл., X10 вкл./выкл.	M10 вкл./выкл.	Счет остановлен. Накопленное значение сохраняется и будет продолжено когда M0=0
Прикладные инструкции	M11 выкл.	M11 выкл.→вкл.	не будут выполняться
	M11 вкл.	M11 вкл.→выкл.	не будут выполняться все кроме API53~59, API 157~159

* Выход Y1 дублируется. Когда M0 выключен, Y1 управляется от M1. Когда M0 включен, Y1 управляется от M12.

API		CALL	P		Вызов подпрограммы	DVP-		
						ES/EX/SS	SA/SX	EH
01						+	+	+

Оп- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (3 шага). CALL - Непрерывное выполнение. CALLP – Импл. выполн.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
<u>Примечания:</u> В качестве операнда могут использоваться указатели P, которые могут индексироваться (E, F)																<u>32-х битная инструкция</u> --- Флаги: нет
Диапазон операндов:																
ES серия: S = P0...P63																
SA/EH серия: S = P0...P255																
Серия ES не поддерживает импульсное выполнение инструкции																

Функция

С помощью CALL-инструкции вызывается подпрограмма

Описание

- Подпрограмма маркируется с помощью точек P 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 в главную программу к инструкции, следующей за CALL-инструкцией.
- Активированные в подпрограмме операнды остаются активированными после отработки подпрограммы до новой обработки подпрограммы.
- Те же точки могут использоваться с любым числом CALL-инструкций.

- Внутри подпрограммы могут вызываться другие подпрограммы. Возможно максимум 5 уровней вложенности включая начальную подпрограмму.

API		SRET			Конец подпрограммы	DVP-		
02						ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (1 шаг). SRET - Непрерывное выполнение. <u>32-х битная инструкция</u> --- Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
<u>Примечания:</u> Нет операндов Контакт для условия выполнения инструкции не требуется															

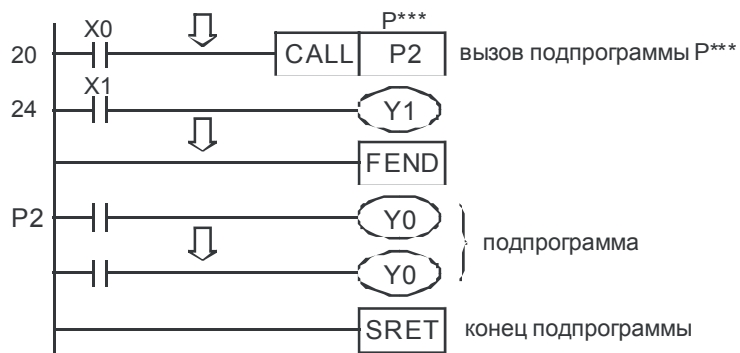
Функция

С помощью SRET-инструкции определяется конец подпрограммы.

Описание

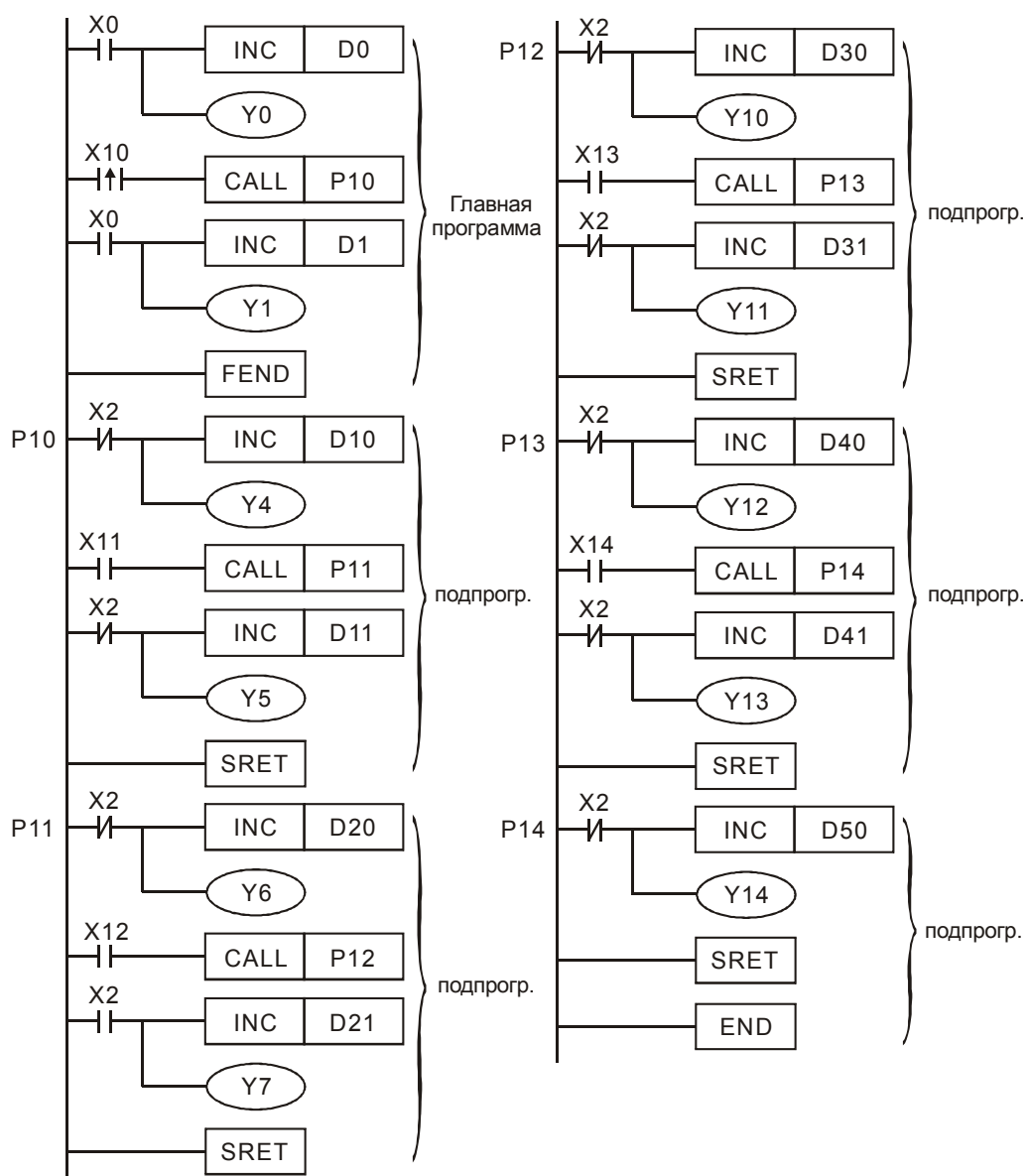
- В конце подпрограммы должна стоять SRET-инструкция.
- После обработки SRET-инструкции осуществляется переход обратно к инструкции следующей за CALL-инструкцией.
- SRET-инструкция может программироваться только вместе с CALL-инструкцией.

Пример программирования с применением CALL- и SRET- инструкций



Когда X0=1, выполнится инструкция CALL и программа перейдет к точке P2. После выполнения SRET-инструкции осуществится переход обратно в главную программу к строке 24.

Пример разветвленной вложенности подпрограмм.



X10 вызывает подпрограмму с P10.

Если во время выполнения подпрограммы P10 будет замкнут контакт X11, то будет вызвана подпрограмма P11.

Если во время выполнения подпрограммы P11 будет замкнут контакт X12, то будет вызвана подпрограмма P12.

Если во время выполнения подпрограммы P12 будет замкнут контакт X13, то будет вызвана подпрограмма P13.

Если во время выполнения подпрограммы P13 будет замкнут контакт X14, то будет вызвана подпрограмма P14. После выполнения SRET-инструкции осуществится переход в предыдущую подпрограмму и так далее по цепочке.

После выполнения SRET-инструкции в подпрограмме P10 осуществится переход в главную программу.

API		IRET		Конец подпрограммы обработки прерывания										DVP-		
03															ES/EX/SS	SA/SX
														+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (1 шаг). IRET - Непрерывное выполнение. <u>32-х битная инструкция</u> --- Флаги: нет		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			
<u>Примечания:</u> Нет операндов Контакт для условия выполнения инструкции не требуется																	

Функция

IRET-инструкция завершает процесс обработки прерывания.

API		EI		Разрешение обработки прерывания	DVP-		
04					ES/EX/SS	SA/SX	EH
					+	+	+

Опе-ранд	Биты				Слова										16-ти битная инструкция (1 шаг). EI - Непрерывное выполнение. <u>32-х битная инструкция</u> --- Флаги: M1050 – M1059, M1280 – M1294		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			
<u>Примечания:</u> Нет операндов Контакт для условия выполнения инструкции не требуется Ширина импульса вызова прерывания должна быть не менее 200 мкс.																	

API		DI			Запрещение обработки прерывания										DVP-		
05																ES/EX/SS	SA/SX
															+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (1 шаг). DI - Непрерывное выполнение.					
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E				F		
<u>Примечания:</u> Нет операндов																		32-х битная инструкция ---		
Контакт для условия выполнения инструкции не требуется																				
																		Флаги: нет		

Принцип функционирования

Вызов, окончание, разрешение и запрещение обработки прерывания

Вызов подпрограммы обработки прерывания

- Допускается прерывание от внешних входных сигналов, прерывание по времени и прерывание по счету.
- При вызове подпрограммы обработки прерывания оставляется главная программа ПЛК и выполняется переход к подпрограмме прерывания. После окончания подпрограммы прерывания выполняется возврат к главной программе ПЛК.
- Начало программы прерывания определяется установкой маркировки (точки прерывания).
- Конец программы прерывания определяется IRET-инструкцией.
- Сигналы прерывания должны иметь ширину импульса минимум в 200 мкс.
- Программа прерывания должна программироваться в конце программы ПЛК за последней FEND-инструкцией и перед END-инструкцией.

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 X, 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 X, 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.

Обработка 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.

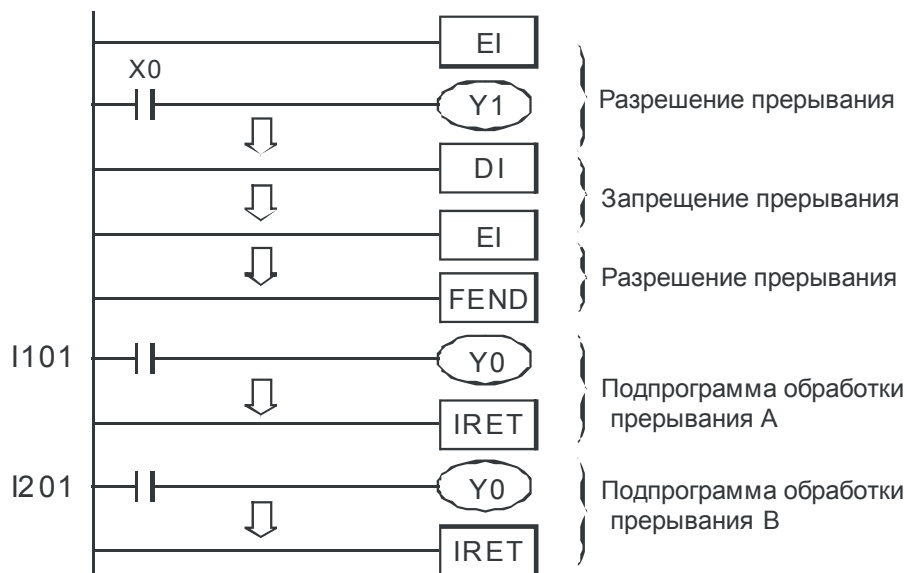
Запрещение обработки прерывания

- Обработка любого прерывания может временно или постоянно выключаться посредством включения соответствующего специального реле M1050 – M1059 (в ES/SA сериях) и M1280 – M1294 (в EN серии).

Пример программирования при использовании инструкций EI, DI и IRET

- Если вход X1 устанавливает сигнал прерывания во время выполнения шага программы внутри области от EI-инструкции до DI-инструкции, то имеет место переход к подпрограмме обработки прерывания I101. Программа прерывания выполняется и происходит возврат в главную программу ПЛК.

- Программа прерывания I101 не выполняется, если активизировано специальное реле M1051.
- Если вход X2 устанавливает сигнал прерывания во время выполнения шага программы внутри области от EI-инструкции до DI-инструкции, то имеет место переход к программе прерывания I201. Программа прерывания выполняется и происходит возврат в главную программу ПЛК.
- Если появляются одновременно сигналы X1 и X2, то вначале обрабатывается программа прерывания A (I101), а затем программа прерывания B (I201).



Адресация точек прерывания должна выполняться следующим образом:

Точки прерывания в ES серии:

1. Внешнее прерывание: (I001, X0), (I101, X1), (I201, X2), (I301, X3) 4 точки.
2. Прерывание по времени: I6□□, 1 точка (□ □ 10~99 мс) (только начиная с версии V5.7)
3. Коммуникационное прерывание для приема специальных символов: (I150) (только начиная с версии V5.7)

Точки прерывания в SA/SX серии:

1. Внешнее прерывание: (I001, X0), (I101, X1), (I201, X2), (I301, X3), (I401, X4), (I501, X5) 6 точек.
2. Прерывание по времени: I6□□, I7□□, 2 точки (□ □ 10~99 мс)
3. Прерывания по счету: I010, I020, I030, I040 4 точки. (Программа прерывания обрабатывается после достижения предварительно заданного значения счета в инструкции DHSCS).
4. Коммуникационное прерывание для приема специальных символов: (I150)
5. Приоритет обработки прерывания: прерывания по счету, внешние прерывания, прерывания по времени и коммуникационные прерывания.

Точки прерывания в EH серии:

1. Внешнее прерывание: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4), (I50□, X5) 6 точек. (□=0 – прерывание по заднему фронту, □=1 – прерывание по переднему фронту импульса.)

2. Прерывание по времени: I6□□, I7□□, 2 точки (□ □ 10~99 мс), I8□□, 1 точка (□ □ 10~99 ед. измерения = 0.1мс)
3. Прерывания по счету: I010, I020, I030, I040 4 точки. (Программа прерывания обрабатывается после достижения предварительно заданного значения счета в инструкции DHSCS).
4. Прерывания по началу и концу выдачи импульсов PLSY-инструкцией: I130, I140
5. Коммуникационное прерывание для приема специальных символов: (I150)
6. Приоритет обработки прерывания: внешние прерывания, прерывания по времени, прерывания по счету, коммуникационные прерывания и импульсные прерывания.

Указание: Адрес прерывания может использоваться только один раз

Флаги запрета обработки прерывания:

DVP-ES/SS/EX

M1050	Запрет внешнего прерывания I001
M1051	Запрет внешнего прерывания I101
M1052	Запрет внешнего прерывания I201
M1053	Запрет внешнего прерывания I301

DVP-SA/SX

M1050	Запрет внешнего прерывания I001
M1051	Запрет внешнего прерывания I101
M1052	Запрет внешнего прерывания I201
M1053	Запрет внешнего прерывания I301
M1054	Запрет внешнего прерывания I401
M1055	Запрет внешнего прерывания I501
M1056	Запрет прерывания по времени I6□□
M1057	Запрет прерывания по времени I7□□
M1059	Запрет прерываний по счету I010 – I060

DVP-EH

M1280	Запрет внешнего прерывания I00□
M1281	Запрет внешнего прерывания I10□
M1282	Запрет внешнего прерывания I20□
M1283	Запрет внешнего прерывания I30□
M1284	Запрет внешнего прерывания I40□
M1285	Запрет внешнего прерывания I50□
M1286	Запрет прерывания по времени I6□□
M1287	Запрет прерывания по времени I7□□
M1288	Запрет прерывания по времени I8□□
M1289	Запрет прерываний по счету I010
M1290	Запрет прерываний по счету I020
M1291	Запрет прерываний по счету I030
M1292	Запрет прерываний по счету I040
M1293	Запрет прерываний по счету I050
M1294	Запрет прерываний по счету I060
M1295	Запрет импульсного прерываний I110
M1296	Запрет импульсного прерываний I120
M1297	Запрет импульсного прерываний I130
M1298	Запрет импульсного прерываний I140

M1299	Запрет импульсного прерываний I150
M1340	Иметь прерывание I110 после завершения передачи по CH0 (Y0, Y1)
M1341	Иметь прерывание I120 после завершения передачи по CH1 (Y2, Y3)
M1342	Иметь прерывание I130 при одновременной передаче по CH0 (Y0, Y1)
M1343	Иметь прерывание I140 при одновременной передаче по CH1 (Y2, Y3)

API		FEND		Конец главной программы	DVP-		
					ES/EX/SS	SA/SX	EH
06					+	+	+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (1 шаг). FEND - Непрерывное выполнение. <u>32-х битная инструкция</u> --- Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
<u>Примечания:</u> Нет операндов Контакт для условия выполнения инструкции не требуется																

Функция

Окончание области главной программы внутри программы ПЛК

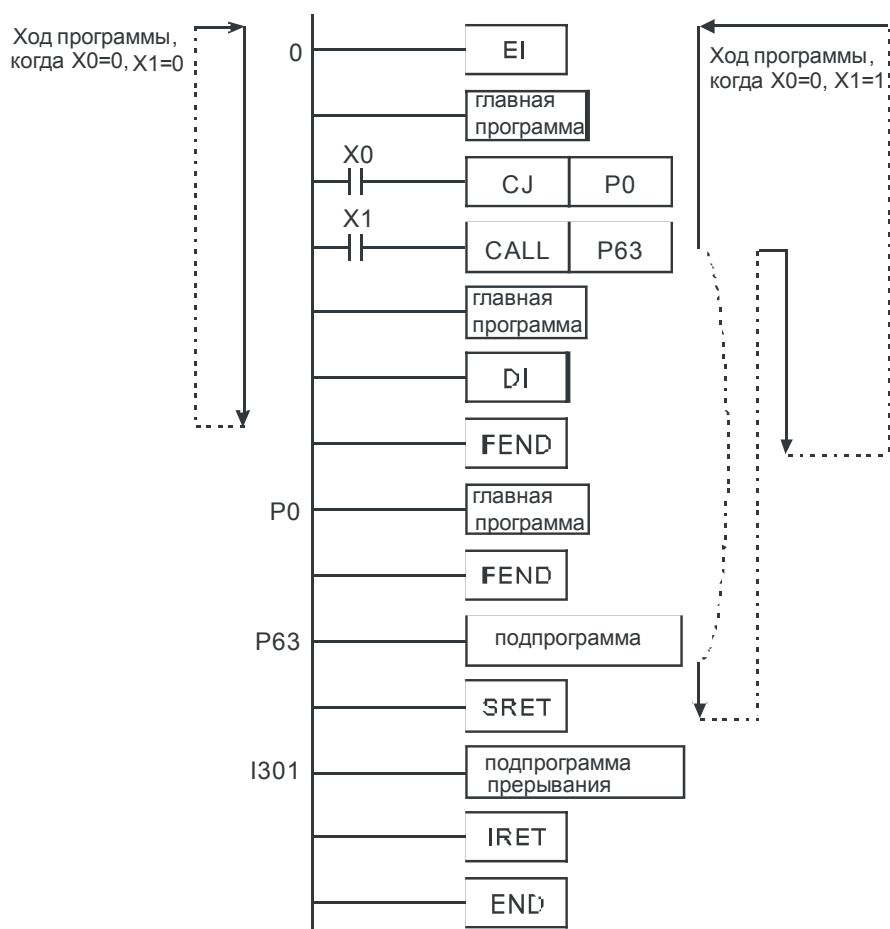
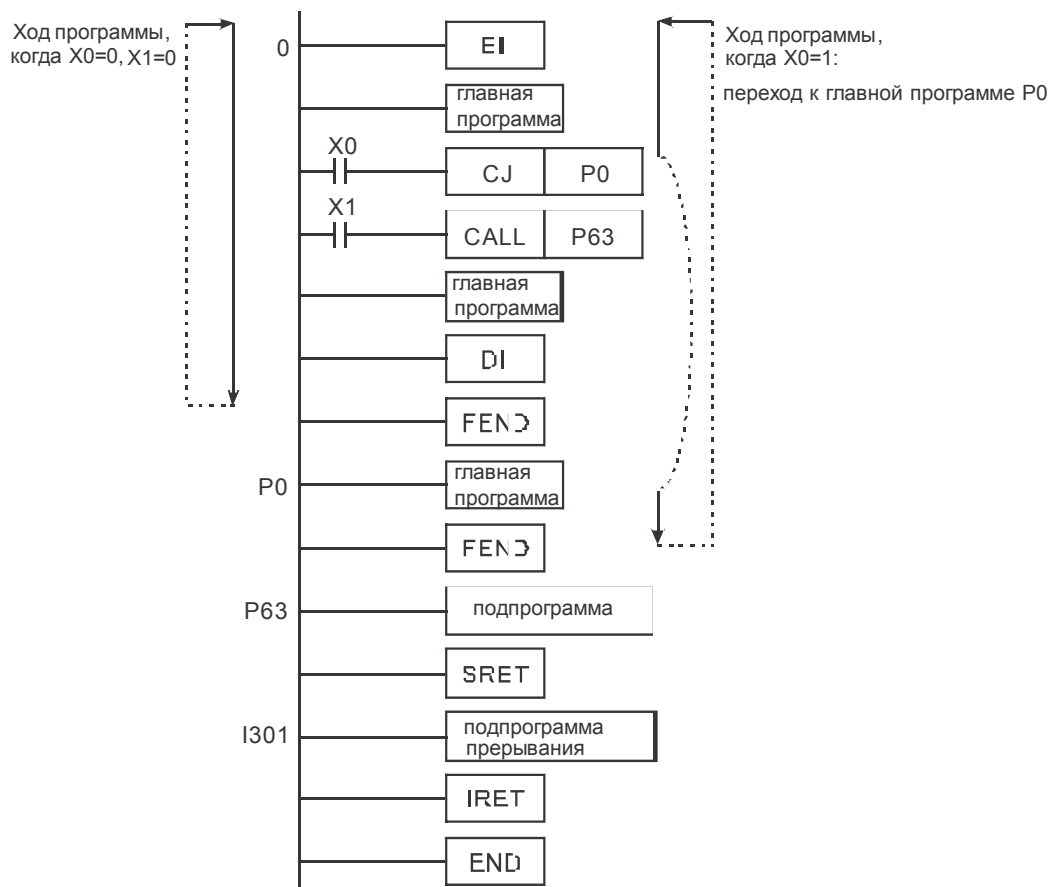
Описание

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

Указание: 1) Программируйте подпрограмму прерывания между FEND-инструкцией и END-инструкцией.

2) Не путайте FEND-инструкцию с END-инструкцией. С помощью END-инструкции завершается вся программа ПЛК.

Пример программирования при использовании FEND-инструкции



API	WDT	P	Сброс сторожевого таймера	DVP-		
				ES/EX/SS	SA/SX	EH
07				+	+	+

Оп- еранд	Биты				Слова										16-ти битная инструкция (1 шаг). WDTP - Непрерывное выполнение. WDTP - Импульсное выполнение. <u>32-х битная инструкция</u> --- Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
<u>Примечания:</u> Нет операндов Импульсное выполнение инструкции WDTP в серии ES/EX/SS не поддерживается.															

Функция

Сторожевой таймер используется для контроля времени выполнения программы и его максимальное значение - 200 мс. Если время выполнения цикла программы превысит это значение, светодиод "ERROR" начнет мигать и ПЛК автоматически перейдет в режим СТОП.

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

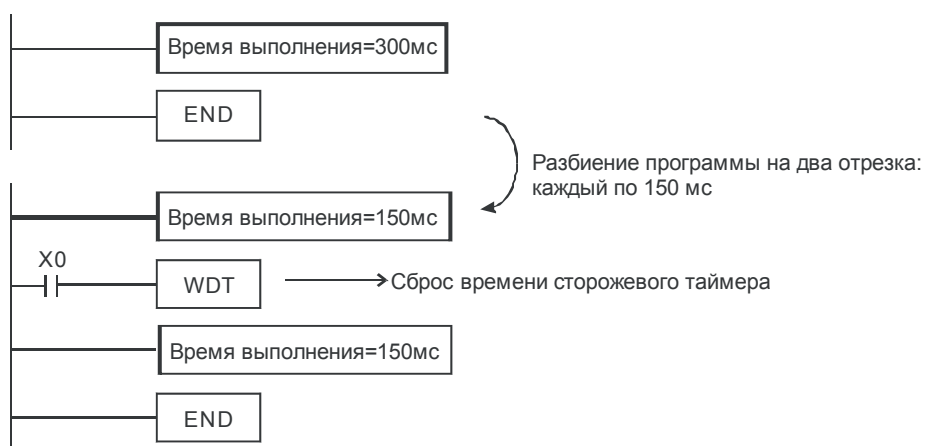
Описание

- 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)
- WDT-инструкция может применяться также внутри FOR-NEXT-цикла

Указание 1) Время контроля цикла обновляется при каждом выполнении инструкций END, FEND или WDT.

Если время цикла программы постоянно превышает значение 200 мс, можно изменить значение максимально допустимого времени цикла в специальном регистре D1000.

Пример программирования при использовании WDT-инструкции



API		FOR		S	Начало цикла FOR-NEXT	DVP-		
08						ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (1 шаг). FOR - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S					*	*	*	*	*	*	*	*	*	*	*	
<u>Примечания:</u> Контакт для условия выполнения инструкции не требуется																32-х битная инструкция --- Флаги: нет

S – число повторений цикла

API		NEXT		-	Конец цикла FOR-NEXT	DVP-		
09						ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (1 шаг). NEXT - Непрерывное выполнение. <u>32-х битная инструкция</u> --- Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
<u>Примечания:</u> Нет операндов Контакт для условия выполнения инструкции не требуется																

Функция

Инструкции FOR/NEXT используются для программирования циклических повторений частей 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 отработается только один раз.

- Можно программировать до пяти FOR-NEXT-уровней вложенности.

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:

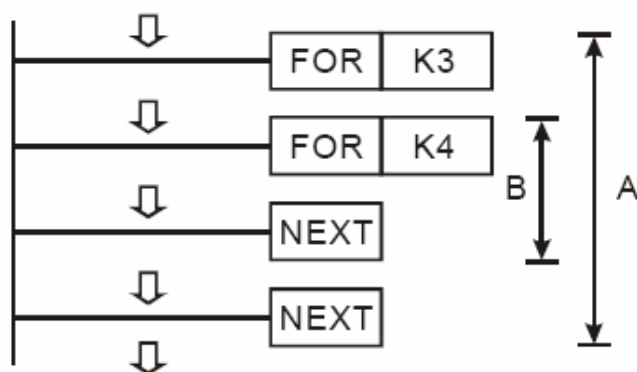
- NEXT-инструкция запрограммирована перед FOR-инструкцией.
- NEXT-инструкция запрограммирована после FEND-инструкции или END-инструкции.
- Количество NEXT-инструкции не соответствует количеству FOR-инструкций.

Большое число повторений "n" может значительно увеличить время выполнения программы и сторожевой таймер может вызвать ошибку. Используйте WDT-инструкцию.

Пример программирования при использовании FOR- и NEXT-инструкций

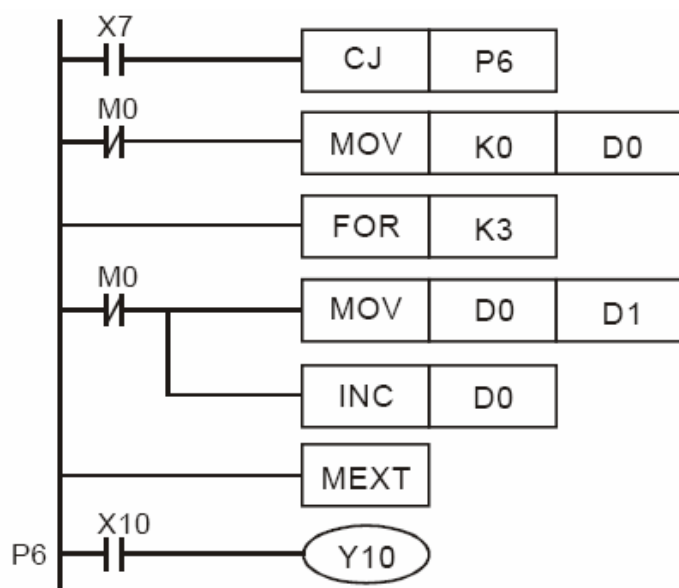
В примере запрограммированы два входящие друг в друга FOR- и NEXT-цикла.

- Отрезок программы А обрабатывается три раза (здесь K3 константа 3).
- При каждом исполнении отрезка А отрезок программы В обрабатывается четыре раза (здесь K4 константа 4).
- Поэтому отрезок В обрабатывается $3 \times 4 = 12$ раз.



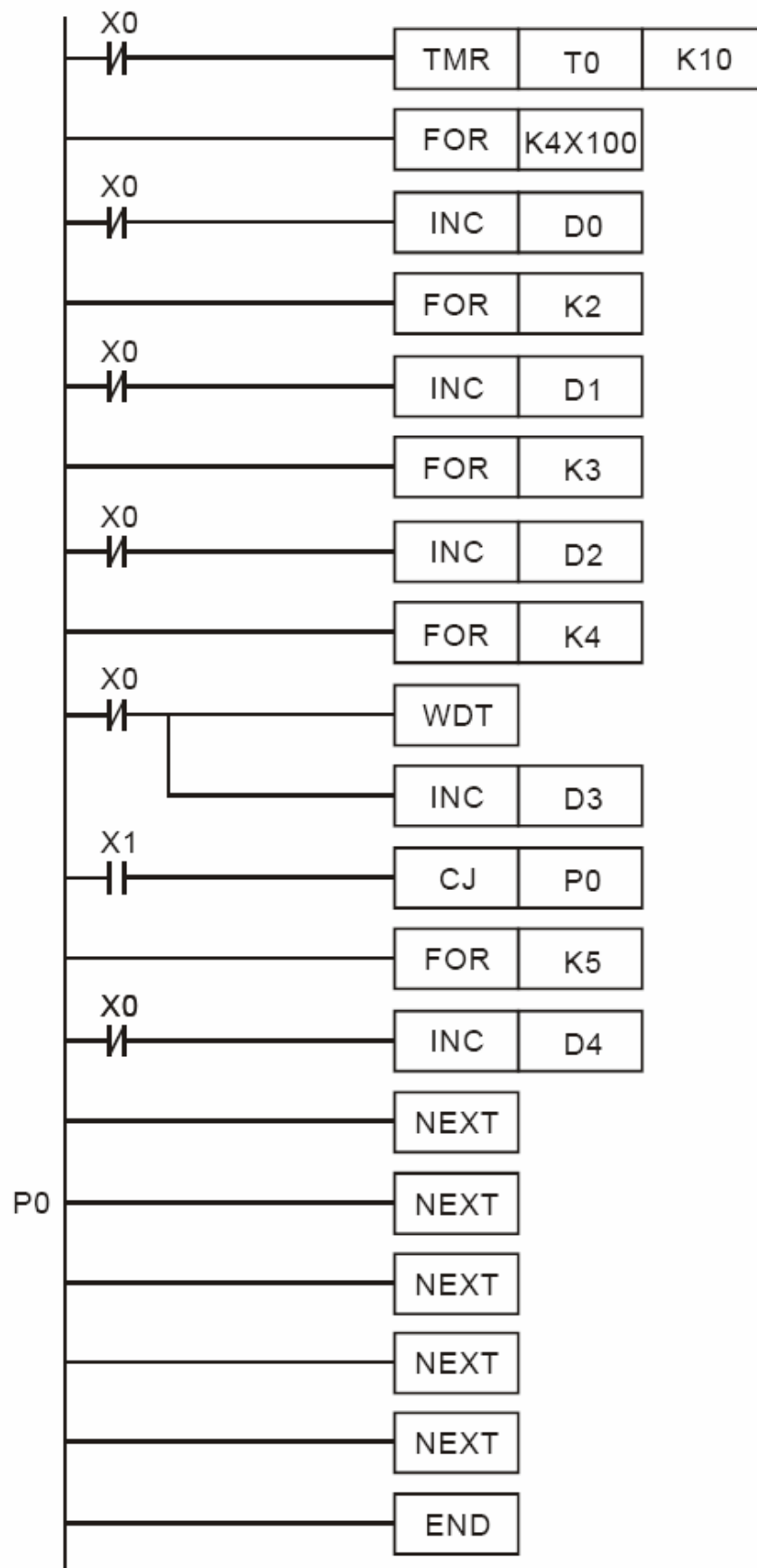
Пример 2 программирования при использовании FOR- и NEXT-инструкций

Если вход X7 включен, то FOR-NEXT-цикл (отрезок программы) пропускается (не обрабатывается) с помощью CJ-инструкции.



Пример 3 программирования при использовании FOR- и NEXT-инструкций

Если вход X1 включен, то вложенный FOR-NEXT-цикл пропускается с помощью CJ-инструкции.



API		CMP		S₁ S₂ D	Сравнение числовых данных	DVP-		
10	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (7 шагов). CMP - Непрерывное выполнение. CMPPP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												
<div><div><div><div><div><div>Примечания:</div></div></div><div><div>Если операнды S₁, S₂ используются с индексом F, возможно только 16-ти битное выполнение команды.</div></div></div><div><div>Операнд D занимает 3 адреса</div><div>Импульсное выполнение инструкции CMPPP в серии ES/EX/SS не поддерживается.</div></div></div></div> <div><div>32-х битная инструкция (13 шагов) DCMP - Непрерывное выполнение. DCMPPP – Имп. выполн.</div><div>Флаги: нет</div></div>																

Функция

Сравнение двух числовых значений данных (больше, меньше, равно)

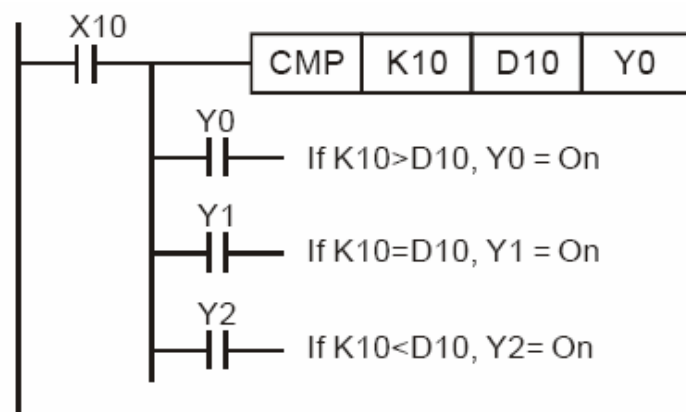
Описание

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

$(S1) > (S2) \rightarrow (D)$
 $(S1) = (S2) \rightarrow (D+1)$
 $(S1) < (S2) \rightarrow (D+2)$

- Данные в S₁ и S₂ обрабатываются как двоичные данные.

Пример программирования с использованием CMP-инструкции

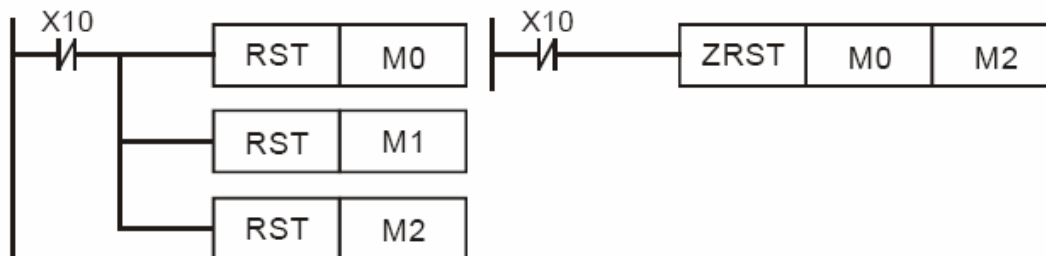


В адресе результата (D) в этом примере указан выход Y0. Соответствующие результаты сравнения автоматически присваиваются приращиваемым на 1 последующим адресам выходов Y0, Y1, Y2 и имеют следующие значения:

- 1) Y0: включен, если K10 > значения регистра D10
- 2) Y1: включен, если K10 = значению регистра D10
- 3) Y2: включен, если K10 < значения регистра D10

Y0, Y1, Y2 не изменяются, если входное условие X10 выключено.

Для сброса результатов сравнения используйте команды RST, ZRST:



API		ZCP		S₁ S₂ S D	Зонное сравнение числовых данных	DVP-		
11	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (9 шагов). ZCP - Непрерывное выполнение. ZCPP – Импл. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
S					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												
<div>Примечания: Если операнды S₁, S₂, S используются с индексом F, возможно только 16-ти битное выполнение команды.</div> <div>Операнд S₁ должен быть меньше чем операнд S₂</div> <div>Операнд D занимает 3 адреса</div> <div>Импульсное выполнение инструкции ZCPP в серии ES/EX/SS не поддерживается.</div>																<div>32-х битная инструкция (17 шагов). DZCP - Непрерывное выполнение. DZCPP – Импл. выполн.</div> <div>Флаги: нет</div>

Функция

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

Описание

- Данные в источнике (S) сравниваются с данными обоих источников (S1) и (S2).
 - Результат сравнения (больше, меньше, равно) отображается (индицируется) благодаря задействованию реле 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) → (D)

(S1) < (S) < (S2) → (D+1)

(S2) < (S) → (D+2)

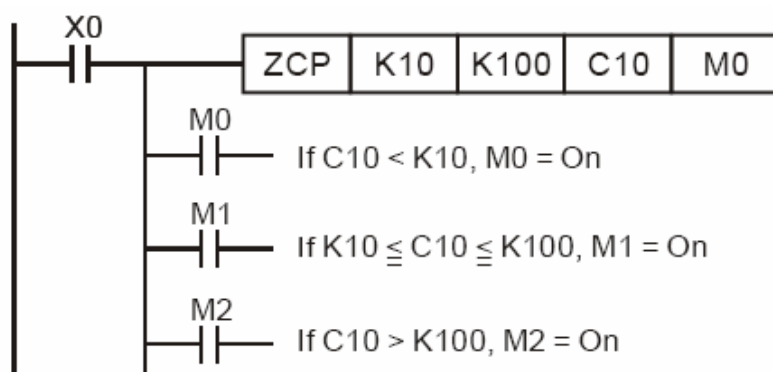
- Данные в (S1) не должны быть больше, чем данные в (S2).

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

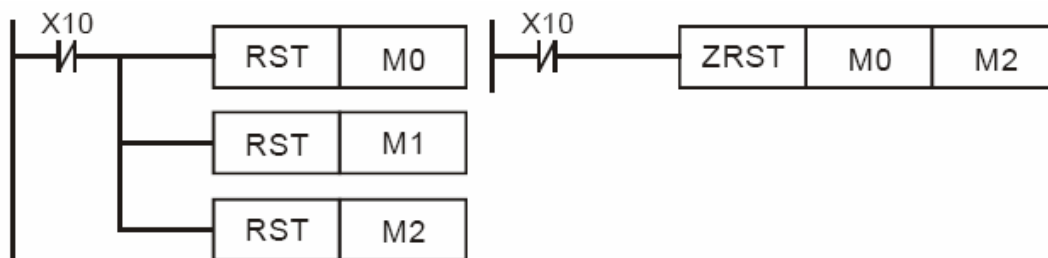
Пример программирования с использованием ZCP-инструкции

В адресе цели (D) в этом примере указано реле M0. Соответствующие результаты сравнения автоматически присваиваются приращиваемым на 1 последующим адресам реле M0, M1, M2 и имеют следующие значения:

- 1) M0: включен, если $K10 >$ накопленного в счетчике C10 значения
 - 2) Если накопленное значение счетчика C10 находится в области от 10 до 100, то реле M1 включено.
 - 3) M2: включен, если накопленное в счетчике C10 значение $> K100$
- M0, M1, M2 не изменяются, если входное условие X0 выключено.



Для сброса результатов сравнения используйте команды RST, ZRST:



API		MOV		<div>S</div> <div>D</div>	Передача данных	DVP-		
12	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты					Слова										16-ти битная инструкция (5 шагов). MOV - Непрерывное выполнение. MOVP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
	S					*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	*

Примечания: Если операнды S и D используются с индексом F, возможно только 16-ти битное выполнение команды.
Импульсное выполнение инструкции MOVP в серии ES/EX/SS не поддерживается.

32-х битная инструкция
(9 шагов).
DMOV - Непрерывное
выполнение.
DMOVP – Имп. выполн.

Флаги: нет

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). Содержимое источника (S) при этом не изменяется.
- Данные в источнике данных (S) при выполнении MOV-инструкции автоматически интерпретируются как двоичные значения.

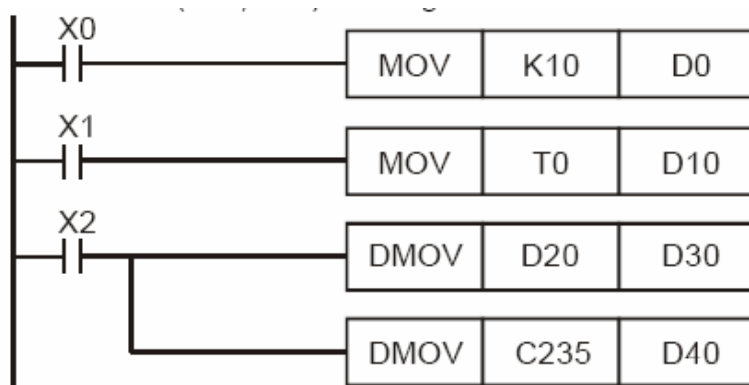
Инструкции выполняются в каждом цикле программы. Этого можно избежать благодаря использованию вставленной впереди импульсной функции (PLS- или PLF-инструкции или же параметра "P").

Пример: Применение MOV-инструкции

Если входное условие X0 включено, то значение регистра D0 будет равно 10. Если X0 выключен, значение D0 не изменится.

Если входное условие X1 включено, то регистру D10 будет передаваться текущее значение таймера T0. Если X1 выключен, значение D10 не изменится.

Если входное условие X2 включено, то регистрам (D30, D31) будет передаваться значение регистров (D20, D21) и регистрам (D40, D41) будет передаваться текущее значение счетчика C235.



API		SMOV		<div><div>S</div><div>m₁</div><div>m₂</div><div>D</div><div>n</div></div>					Передача данных со смещением		DVP-		
13			P	ES/EX/SS	SA/SX	EH							
											+	+	+

Оп-ранд	Биты				Слова										16-ти битная инструкция (11 шагов). SMOV - Непрерывное выполнение. SMOVP – Импл. выполн.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S							*	*	*	*	*	*	*	*	*	
m ₁					*	*										
m ₂					*	*										
D								*	*	*	*	*	*	*	*	
n					*	*										
<div><div>Примечания:</div><div>диапазон операндов:</div><div>m₁ = 1 – 4 ; m₂ = 1 - m₁; n = m₂ – 4</div></div>																32-х битная инструкция ---- Флаги: M1168 (выбор режима операции SMOV) M1168=0: BCD режим M1168=1: BIN режим

Функция

Передача данных и изменение их значимости

Описание

S – источник передаваемых данных

n, m₁, m₂ определяют вид (параметры) сдвига BCD-мест.

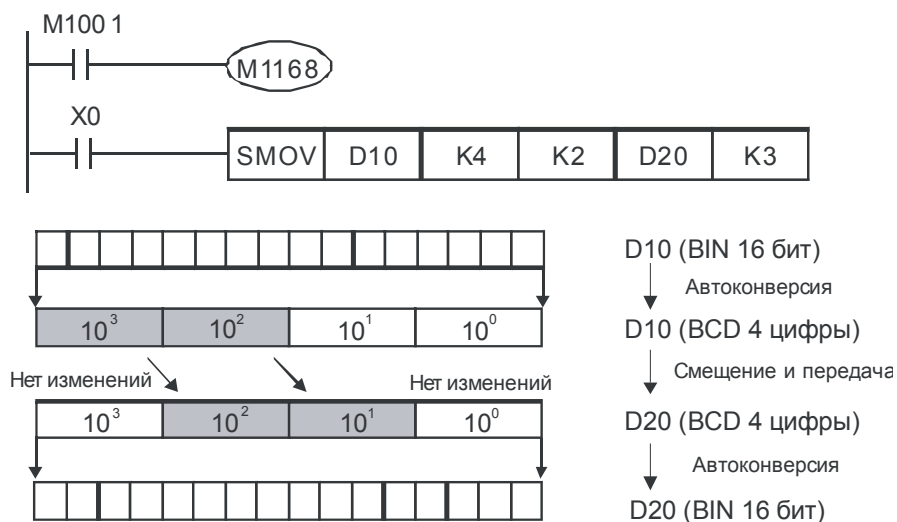
m₁ = 1. первое место (в слове), которое должно быть смещено

m₂ = число мест, которые должны быть смещены

n = 1. первый адрес в цели (место BCD, куда смещать)

D – приемник данных

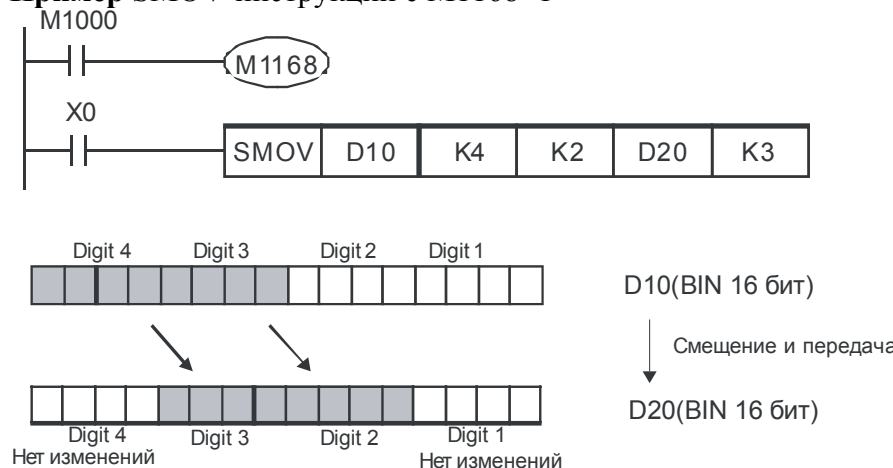
Пример SMOV-инструкции с M1168=0



Если входное условие X0 включено, то часть содержимого регистра D10=H1234 будет передано части содержимого регистра D20:

До выполнения: D10=H1234, D20=H5678

После выполнения: D10=H1234, D20=H5128

Пример SMOV-инструкции с M1168=1

Когда M1168=1, данные регистров D10, D20 не конвертируются в BCD-формат, а передаются как 4-х разрядное BIN-число.

Пример программирования для двоичных данных

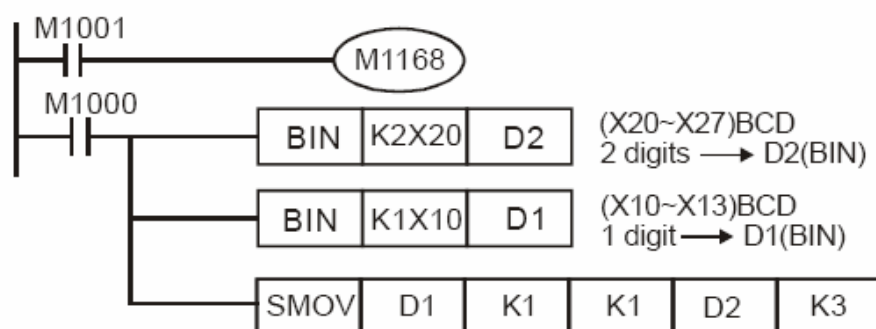
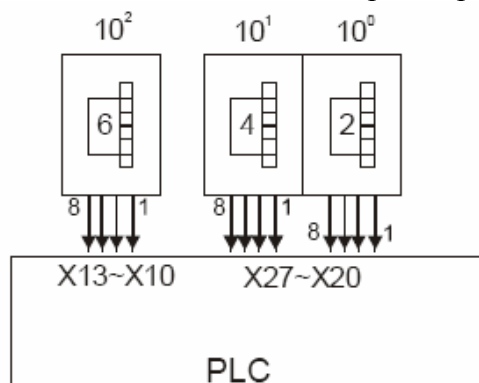
Входные данные поставляются от трех BCD-переключателей, распределенных по соответствующим входам ПЛК.

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 X10...X13 (K1X10).

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.



API		CML		S D	Передача данных с их инвертированием	DVP-		
14	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (5 шагов).
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CML - Непрерывное выполнение. CMLP – Имп. выполн.
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	
<div><div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div></div></div>																

Функция

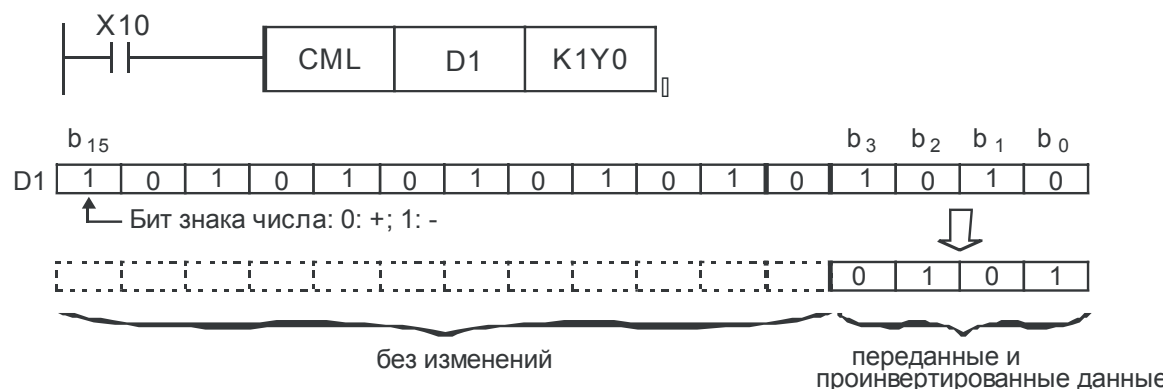
Образование 1-го дополнения двоичного числа

Описание

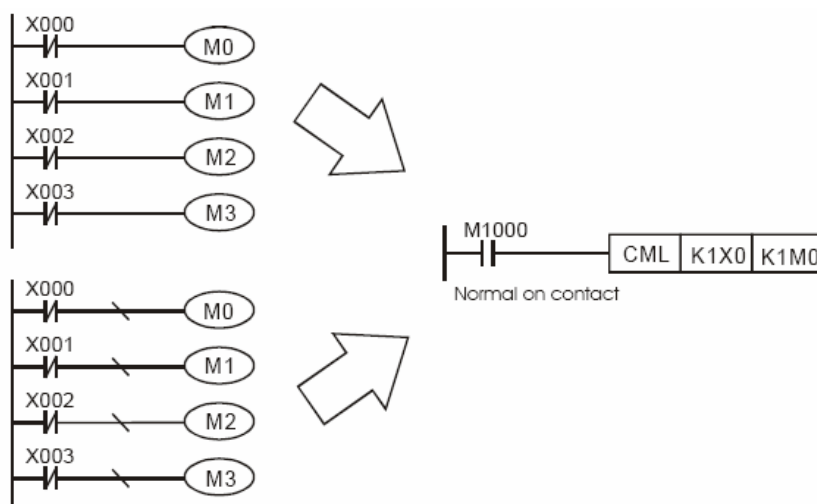
Двоичное значение числа в (S) преобразовывается в свое 1-ое дополнение и записывается в данные цели (D).

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

Пример CML-инструкции



Ниже показан пример передачи обратного состояния входов X внутренним реле M (программы слева идентичны инструкции справа):



API	BMOV	P	S D n	Передача блока данных	DVP-		
					ES/EX/SS	SA/SX	EH
15					+	+	+

Оп- еранд	Биты				Слова										16-ти битная инструкция (7 шагов). BMOV - Непрерывное выполнение. BMOVP – Импл. выполн.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*											
<u>Примечания:</u> n = 1 - 512 Импульсное выполнение инструкции BMOVP в серии ES/EX/SS не поддерживается.																32-х битная инструкция --- Флаги: нет	

Функция

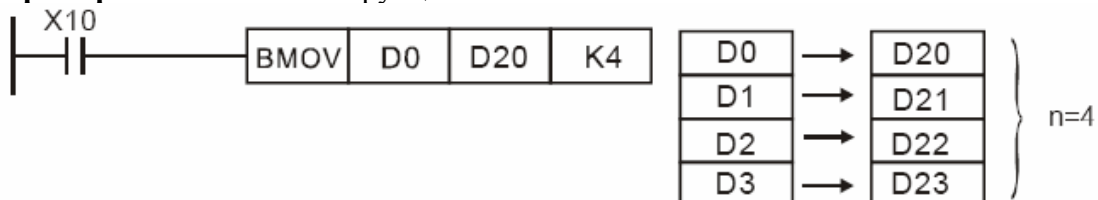
Копирование пакета данных

Описание

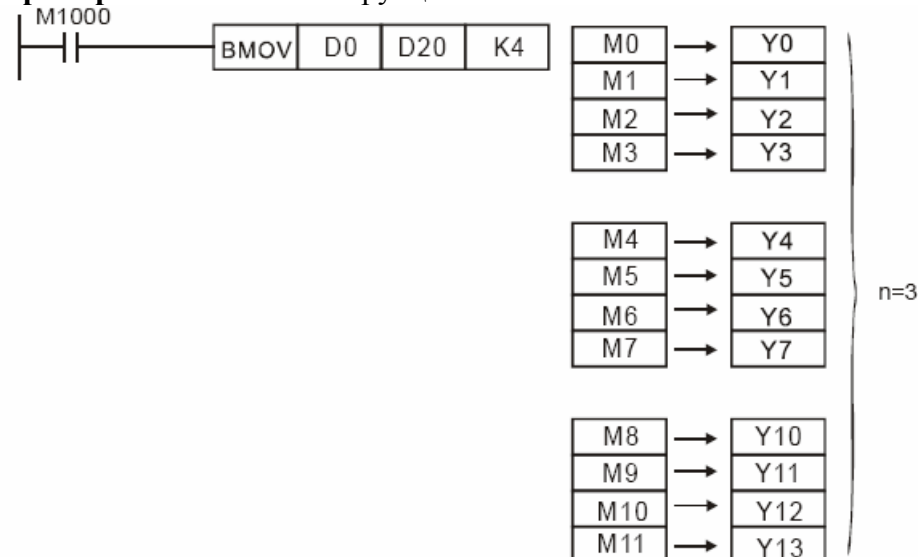
- Переносится (передается) предварительно заданное количество словных операндов.
- Для передачи предварительно задаются стартовый адрес (S), адрес цели (D) и число переносимых слов (n).

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

Контроллеры ES/EX/SS не поддерживают передачу блока данных между операндами KnX, KnY, KnM, KnS

Пример 1 BMOV-инструкции

Когда X10 включено, содержимое регистров D0 – D3 будет копироваться в регистры D20 – D23.

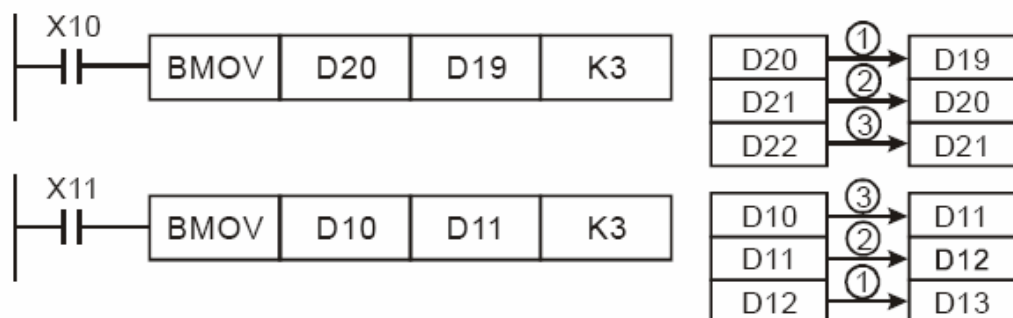
Пример 2 BMOV-инструкции

Если адреса источников S и приемников D данных пересекаются, то порядок копирования будет следующий:

Если $S > D$: порядок копирования $1 \rightarrow 2 \rightarrow 3$

Если $S < D$: порядок копирования $3 \rightarrow 2 \rightarrow 1$

Второго случая, когда $S < D$ в контроллерах ES/EX/SS надо стараться избегать, т.к. в регистры D11 – D13 будет записано значение D10 (см. пример):



API		FMOV		S	D	n	Передача данных в несколько адресов	DVP-		
16	D		P					ES/EX/SS	SA/SX	EH
								+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (7 шагов). FMOV - Непрерывное выполнение. FMOVP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*			
n					*	*										
<p><u>Примечания:</u> Если операнды S и D используются с индексом F, возможно только 16-ти битное выполнение команды. n = 1 – 512 (16-ти битный режим), n = 1 – 256 (32-х битный режим) Импульсное выполнение инструкции FMOVP в серии ES/EX/SS не поддерживается.</p>																<p><u>32-х битная инструкция</u> (13 шагов). DFMOV - Непрерывное выполнение. DFMOVP – Имп. выполн.</p> <p>Флаги: нет</p>

Функция

Перенос значения данных в несколько операндов

Описание

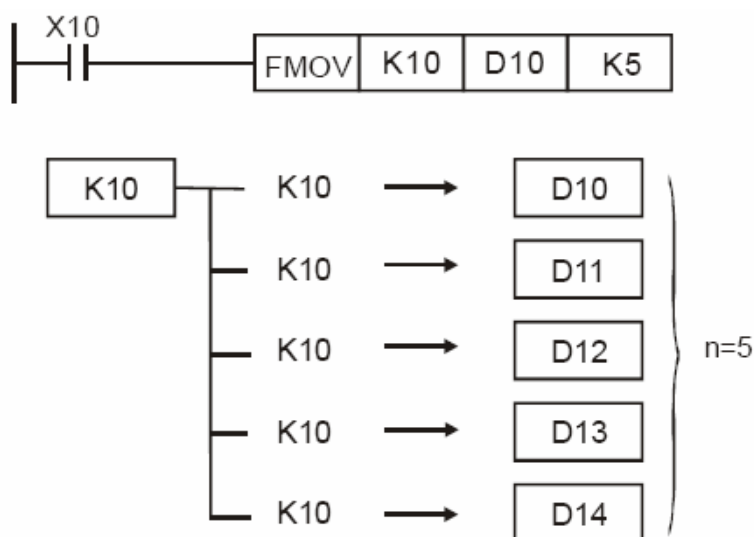
- Содержание данных из (S) переносится (передается) в несколько операндов одинакового типа.
- Первый операнд цели определяется в (D).
- Исходя от операнда цели (D) значение данных передается из (S) в "n" операндов.

Если "n" больше числа используемых операндов, то выполняется передача только до последнего имеющегося операнда.

Контроллеры ES/EX/SS не поддерживают передачу данных между операндами KnX, KnY, KnM, KnS

Пример FMOV-инструкции

Передача(перенос)значения "10" в регистры данных D10...D14



API		XCH	☺	D₁	D₂	Обмен данными	DVP-		
17	D		P				ES/EX/SS	SA/SX	EH
							+	+	+

Опе- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (5 шагов). XCH - Непрерывное выполнение. XCHP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D ₁								*	*	*	*	*	*	*	*	
D ₂								*	*	*	*	*	*	*	*	
<div><div><p><u>Примечания:</u> Если операнды D₁ и D₂ используются с индексом F, возможно только 16-ти битное выполнение команды.</p><p>Импульсное выполнение инструкции XCHP в серии ES/EX/SS не поддерживается.</p></div><div><p><u>32-х битная инструкция</u> (9 шагов). DXCH - Непрерывное выполнение. DXCHP – Имп. выполн.</p><p>Флаги: нет</p></div></div>																

Функция

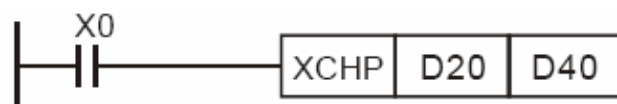
Обмен данными между двумя операндами

Описание

Обмениваются данные (D1) и (D2).

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

ПРИМЕР XCH-инструкции



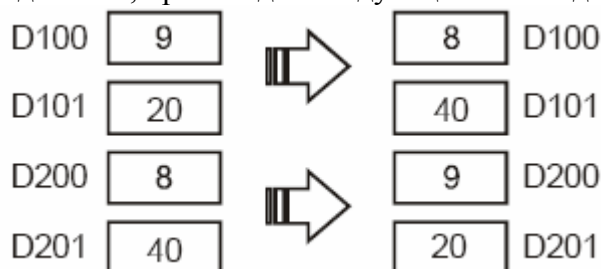
Когда X0=1, происходит следующий обмен данными:



Пример DXCH-инструкции



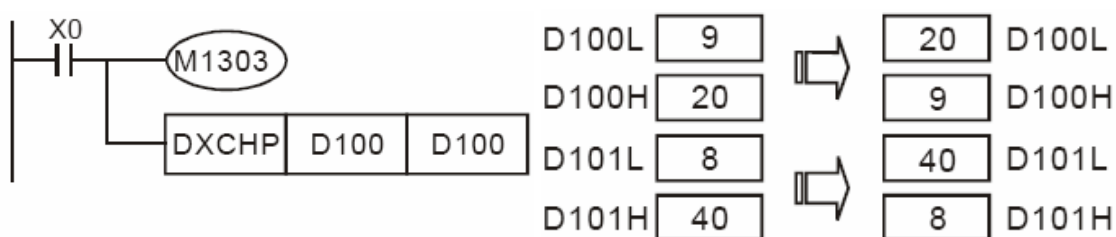
Когда $X_0=1$, происходит следующий обмен данными:



Принцип функционирования со специальным реле M1303:

После установки специального реле M1303=1 в (D1) и (D2) обмениваются старший и младший байты. После ХСН-инструкция специальное реле M1303 должно снова отключиться.

При применении ХСН-инструкции со специальным реле М1303 (D1) и (D2) должны иметь одинаковые типы регистров данных, в противном случае может появиться ошибка.



API		BCD		<div><div>S</div><div>D</div></div>	BCD-конвертирование	DVP-		
18	D					P	ES/EX/SS	SA/SX
						+	+	+

Опе- ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S							*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*
<p><u>Примечания:</u> Если операнды S и D используются с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Импульсное выполнение инструкции BCDP в серии ES/EX/SS не поддерживается.</p>															

16-ти битная инструкция
(5 шагов).
BCD - Непрерывное
выполнение.
BCDP – Имп. выполн.

32-х битная инструкция
(9 шагов).
DBCD - Непрерывное
выполнение.
DBCDP – Имп. выполн.

Флаги: M1067 (ошибка
операции);
M1068 (ошибка
операции);
D1068 (код ошибки)

Функция

Конвертирование двоичных данных (BIN) в BCD-формат

Описание

Основные математические операции (+, -, x, ÷, INC, DEC) в ПЛК выполняются в двоичном (BIN) формате. Благодаря применению BCD-инструкции могут выдаваться данные также в BCD-формате (например, для управления 7-ми сегментным устройством отображения).

- Двоичные данные источника (S) конвертируются в BCD-данные и передаются по адресу цели (D).

- Результат BCD-конвертирования должен находиться внутри допустимой области:

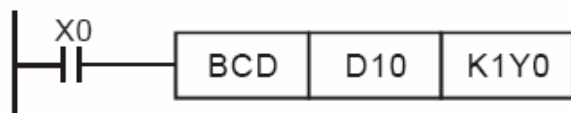
16-ти-битовая-инструкция: от 0 до +9 999

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 (M1067, M1068 = 1) с кодом в D1068 = 0E18

Пример: BCD-инструкция может применяться, например, чтобы прочесть двоичное число из ПЛК и изобразить его на 7-ми сегментном индикаторе.



Когда X0=1: двоичные данные из регистра данных D10 конвертируются в BCD-формат и затем выдаются по выходам Y0...Y3.

API		BIN		S D	BIN-конвертирование	DVP-		
19	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (5 шагов). BIN - Непрерывное выполнение. BINP – Импл. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	
<p><u>Примечания:</u> Если операнды S и D используются с индексом F, возможно только 16-ти битное выполнение команды. Импульсное выполнение инструкции BCDP в серии ES/EX/SS не поддерживается.</p>																<p><u>32-х битная инструкция</u> (9 шагов). DBIN - Непрерывное выполнение. DBINP – Импл. выполн.</p> <p>Флаги: M1067 (ошибка операции); M1068 (ошибка операции); D1068 (код ошибки)</p>

Функция

Конвертирование данных из BCD-формата в двоичный формат (BIN)

Описание

Основные математические операции (+, -, x, ÷, INC, DEC) в ПЛК выполняются в двоичном (BIN) формате. 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:

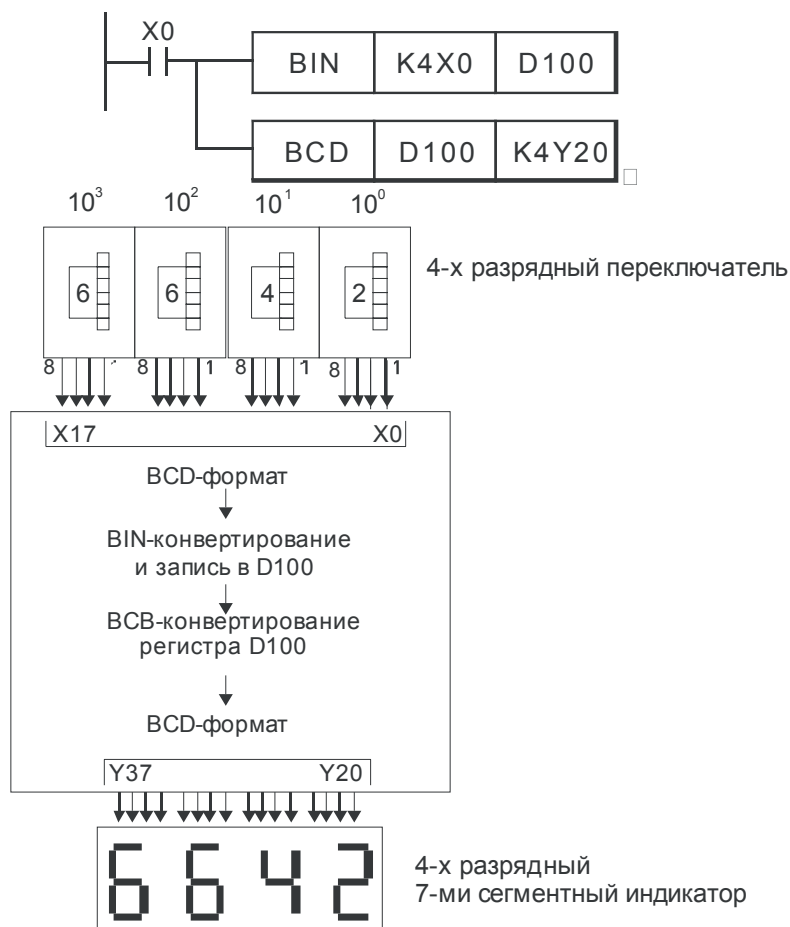
Если данные в (S) не в BCD-формате, появляется ошибка. Ошибка отображается благодаря включившемуся специальному реле M1067, M1068 = 1 с кодом в D1068 = 0E18

ПРИМЕР BIN-инструкции



Когда X0=1: двоично-десятичные данные входов X0...X3 конвертируются в BIN-формат и затем записываются в регистр D10.

Пример программирования конвертирования BCD-данных в от цифрового переключателя в BIN-формат и вывода данных в BCD-формате на 7-ми сегментном индикаторе:



BCD-данные на входах X0...X17 конвертируются в формат двоичных данных по адресу цели D100 и затем выдаются на выходы Y20...Y37. В этом примере: 6642 (десятичное число).

API			ADD			S ₁ S ₂ D			Сложение числовых данных							DVP-				
																ES/EX/SS		SA/SX		EH
20		D			P											+		+		+

Оп- ранд	Биты				Слова												<u>16-ти битная инструкция</u> (7 шагов). ADD - Непрерывное выполнение. ADDP – Имп. выполн.				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F						
	S ₁				*	*	*	*	*	*	*	*	*	*	*						
	S ₂				*	*	*	*	*	*	*	*	*	*	*						
	D							*	*	*	*	*	*	*	*						
<p><u>Примечания:</u> Если операнды S₁, S₂, D используются с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Импульсное выполнение инструкции ADDP в серии ES/EX/SS не поддерживается.</p>																			<u>32-х битная инструкция</u> (13 шагов) DADD - Непрерывное выполнение. DADDP – Имп. выполн.		
																			Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)		

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.

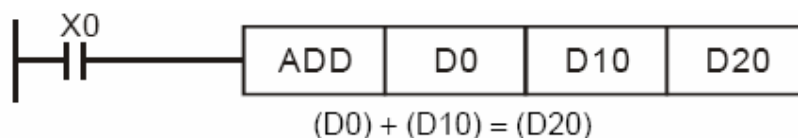
- При определенных результатах счета после исполнения инструкции включается специальное реле (флаг).

1) ФЛАГ НУЛЯ M1020: если результатом сложения является 0, включается флаг нуля

2) ФЛАГ ЗАИМСТВОВАНИЯ M1021: 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.

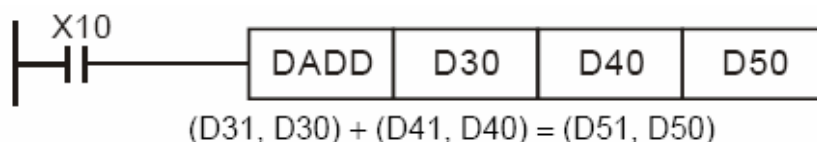
3) ФЛАГ ПЕРЕНОСА M1022: если результатом сложения явилось число выше +32 767 (16-ти битовая операция) ил и же +2 147 483 647 (32-х битовая операция), включается флаг переноса.

Пример: Применение ADD-инструкции (16 бит)



Если включен X0, то суммируются значения данных в регистрах D0 и D10. Результат суммирования запоминается в регистре данных D20.

Пример: Применение DADD-инструкции (32 бит)



Если включен X10, то суммируются значения данных в регистрах (D31, D30) и (D41, D40). Результат суммирования запоминается в регистрах данных (D51, D50).

API		SUB		$\textcircled{S_1}$ $\textcircled{S_2}$ \textcircled{D}	Вычитание числовых данных	DVP-		
						ES/EX/SS	SA/SX	EH
21	D		P			+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). SUB - Непрерывное выполнение. SUBP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S ₁					*	*	*	*	*	*	*	*	*	*	*
S ₂					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*
<div><div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div>															

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). Операция выполняется в BIN-формате.

$$(S1)-(S2) = (D)$$

- 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.

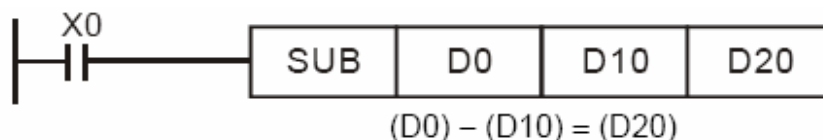
- При определенных результатах счета после исполнения инструкции включается специальное реле (флаг).

1) ФЛАГ НУЛЯ M1020: если результатом операции является 0, включается флаг нуля

2) ФЛАГ ЗАИМСТВОВАНИЯ M1021: если результатом операции явилось число меньше -32 767 (16-ти битовая операция) или же -2 147 483 648 (32-х битовая операция), включается флаг заимствования.

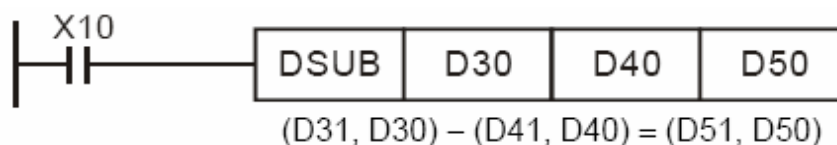
3) ФЛАГ ПЕРЕНОСА M1022: если результатом операции явилось число выше +32 767 (16-ти битовая операция) ил и же +2 147 483 647 (32-х битовая операция), включается флаг переноса.

ПРИМЕР: Применение SUB-инструкции (16 бит)



Если включен X0, то находится разность значений данных в регистрах D0 и D10. Результат вычитания запоминается в регистре данных D20.

ПРИМЕР: Применение DSUB-инструкции (32 бит)



Если включен X10, то находится разность значений данных в регистрах (D31, D30) и (D41, D40). Результат вычитания запоминается в регистрах данных (D51, D50).

API		MUL		S ₁ S ₂ D	Умножение числовых данных	DVP-		
						ES/EX/SS	SA/SX	EH
22	D		P			+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). MUL - Непрерывное выполнение. MULP – Имп. выполн.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*		
<p><u>Примечания:</u> Если операнды S₁, S₂, D используются с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Если операнд D используются с индексом E, возможно только 16-ти битное выполнение команды.</p> <p>В 16-ти битном режиме операнд D занимает 2 адреса.</p> <p>В 32-х битном режиме операнд D занимает 4 адреса.</p> <p>Импульсное выполнение инструкции MULP в серии ES/EX/SS не поддерживается.</p>																32-х битная инструкция (13 шагов) DMUL - Непрерывное выполнение. DMULP – Имп. выполн. Флаги: нет

Функция

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

Описание

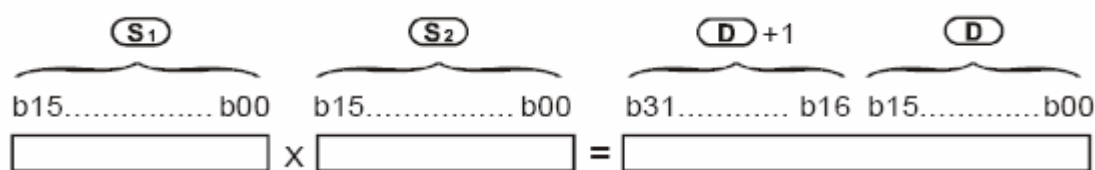
• Данные в S1 и S2 перемножаются между собой. Результат умножения запоминается по адресу операнда указанного в D и в следующем за ним адресе операнда. Операция выполняется в BIN-формате.

$$(S1) \times (S2) = (D)$$

• В старшем бите запоминается знак результата перемножаемых чисел:

0: знак положительного числа 1: знак отрицательного числа

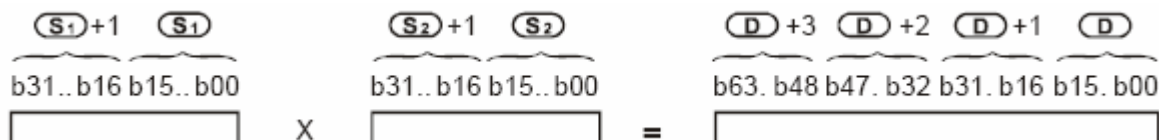
• При выполнении 16-ти битной операции результат заносится в 32-х битное число в (D) и (D+1). Результат 16-ти битного умножения оказывается 32-х битным числом. Это число запоминается как 32-х битное значение. Младшие 16 бит записываются по адресу операнда, заданному в (D). Старшие 16 бит записываются по следующему за ним адресу операнда.



В бите b31 запоминается знак результата перемножаемых чисел

Если D - битовый операнд и размер его K1-K4 для 16-ти битной операции, то флаги ошибки M1067, M1068 включаться в регистре D1067 будет записан код "0E19"

• При выполнении 32-х битной операции результат заносится в 64-х битное число в (D) и в три следующих за ним операнда. Результат 32-х битного умножения запоминается 64-х значением данных. Младшие 16 бит запоминаются по адресу операнда, заданному в (D). Старшие биты записываются по следующим за ним адресам операнда. При 32-х битном операнде для (D) нельзя применять индексы E(F).

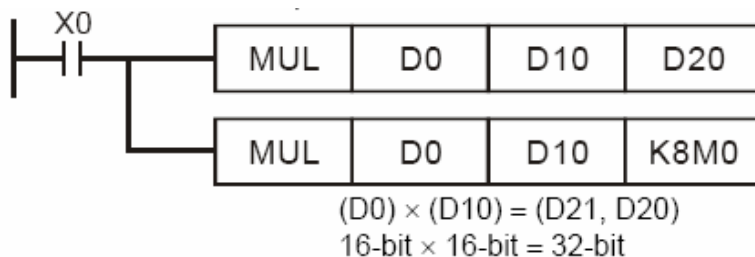


В бите b63 запоминается знак результата перемножаемых чисел

Если D - битовый операнд и размер его K1-K8 для 32-ти битной операции, в назначенных операндах будут сохраняться только младшие 32 бита.

Пример: Применение MUL-инструкции

Если включен X0, то находится произведение значений данных в регистрах D0 и D10. Результат умножения запоминается в регистре данных D20 и битовых операндах M0 – M31.



API		DIV		P	S₁ S₂ D	Деление числовых данных	DVP-		
							ES/EX/SS	SA/SX	EH
23	D						+	+	+

Оп- еранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S ₁					*	*	*	*	*	*	*	*	*	*	*
S ₂					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	

16-ти битная инструкция
(7 шагов).
DIV - Непрерывное выполнение.
DIVP – Имп. выполн.

32-х битная инструкция
(13 шагов)
DDIV - Непрерывное выполнение.
DDIVP – Имп. выполн.

Флаги: нет

Примечания: Если операнды S₁, S₂, D используются с индексом F, возможно только 16-ти битное выполнение команды.
Если операнд D используется с индексом E, возможно только 16-ти битное выполнение команды.
В 16-ти битном режиме операнд D занимает 2 адреса.
В 32-х битном режиме операнд D занимает 4 адреса.
Импульсное выполнение инструкции DIVP в серии ES/EX/SS не поддерживается.

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. Операция выполняется в BIN-формате.

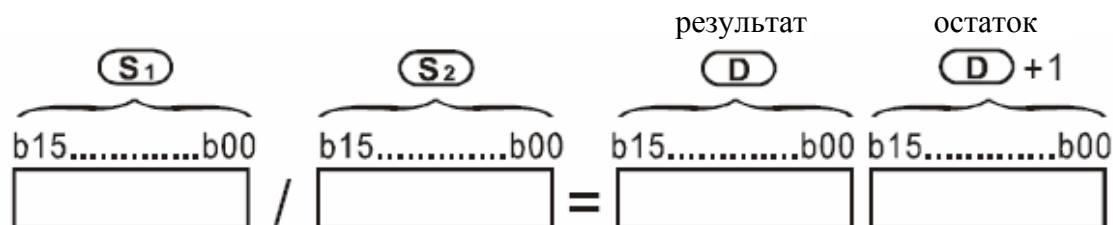
В старшем бите запоминается знак числа результата деления.

0: знак положительного числа; 1: знак отрицательного числа

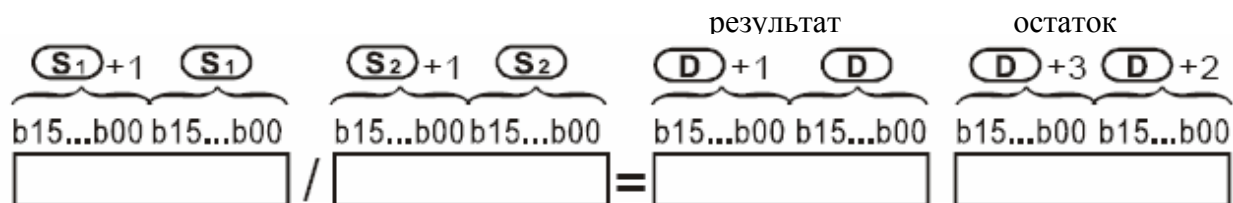
- Знак результата деления зависит от знаков чисел делимого и делителя. При 32-х битном операнде для (D) нельзя применять E(F).

- При работе программы появляется ошибка, если значение делителя равно 0. Флаги ошибки M1067, M1068 включаются в регистре D1067 будет записан код "0E19".

- Результат 16-ти битного деления запоминается по адресу операндов заданном в D. Остаток от деления запоминается в следующем за ним адресе операнда.

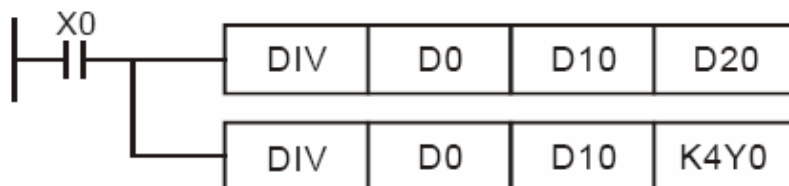


- При делении 32-х битных данных для делимого, делителя, результата и остатка от деления имеется соответственно по два следующих друг за другом регистра данных. В DDIV-инструкции должны указываться соответственно регистры данных с нижними адресами операндов.



Пример: Применение DIV-инструкции

Если включен X0, то выполняется деление значений данных в регистрах D0 и D10. Результат деления запоминается в регистре данных D20 и битовых операндах Y0 – Y15, а остаток деления запоминается в D21.



API		INC	☺		Инкрементирование числовых данных	DVP-		
24	D		P	D		ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (3 шага). INC - Непрерывное выполнение. INCP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
D								*	*	*	*	*	*	*	*
<p><u>Примечания:</u> Если операнд D используется с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Импульсное выполнение инструкции INCP в серии ES/EX/SS не поддерживается.</p> <p><u>32-х битная инструкция</u> (5 шагов) DINC - Непрерывное выполнение. DINCP – Имп. выполн.</p> <p>Флаги: нет</p>															

Функция

К числовому значению данных прибавляется число 1.

Инструкция выполняется в каждом цикле программы. Этого можно избежать благодаря введению впереди функций импульса (PLS- или PLF-инструкции) или применив командный параметр P (для DVP-SA/SX/EH).

К значению числа, имеющемуся в D, прибавляется число 1, как только выполнится входное условие.

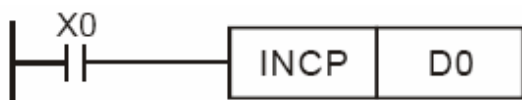
• 16-ТИ БИТОВЫЙ ОПЕРАНД (INC-ИНСТРУКЦИЯ)

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



Значение данных в регистре D0 при наличии входного сигнала X0 увеличится на число 1.

Инструкция активизируется благодаря подключенной впереди функции импульса. Это важно, чтобы процесс суммирования не выполнялся в каждом цикле программы.

API		DEC	☺		Декрементирование числовых данных	DVP-		
25	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (3 шага). DEC - Непрерывное выполнение. DECP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
D								*	*	*	*	*	*	*	*
<p><u>Примечания:</u> Если операнд D используется с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Импульсное выполнение инструкции DECP в серии ES/EX/SS не поддерживается.</p> <p><u>32-х битная инструкция</u> (5 шагов) DDEC - Непрерывное выполнение. DDECP – Имп. выполн.</p> <p>Флаги: нет</p>															

Функция

Из числового значения данных вычитается число 1.

Инструкция выполняется в каждом цикле программы. Этого можно избежать благодаря введению впереди функций импульса (PLS- или PLF-инструкции) или применив командный параметр P (для DVP-SA/SX/EH).

Из 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



Значение данных в регистре D0 при наличии 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.

API	W	AND		S₁ S₂ D	Логическое умножение числовых данных (И)	DVP-		
26	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). WAND - Непрерывное выполнение. WANDP – Импл. выполн.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	
<p><u>Примечания:</u> Если операнды S₁, S₂ , D используются с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Импульсное выполнение инструкции WANDP в серии ES/EX/SS не поддерживается.</p>																<p><u>32-х битная инструкция</u> (13 шагов) DAND - Непрерывное выполнение. DANDP – Импл. выполн.</p> <p>Флаги: нет</p>

Функция

Операция "логическое И" для числовых данных.

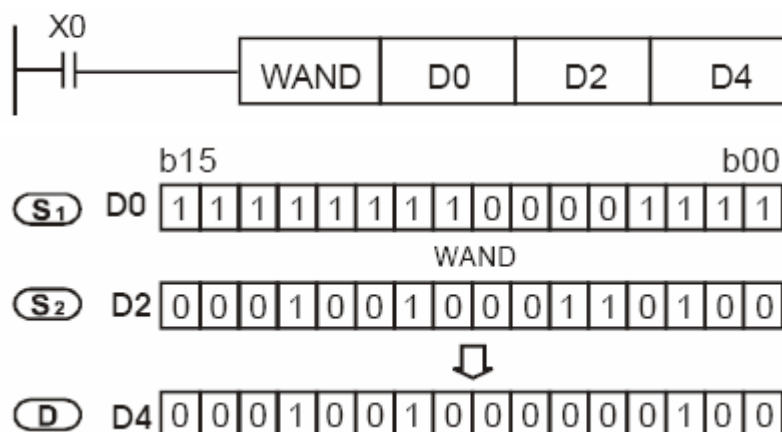
Описание

- Операция "логическое И" для числовых данных выполняется по отдельным битам.
- Данные в (S1) и (S2) побитно логически связываются друг с другом. Результат связи сохраняется в (D).
- Таблица истинности логического умножения

(S1)	(S2)	(D)
1	1	1
1	0	0
0	1	0
0	0	0

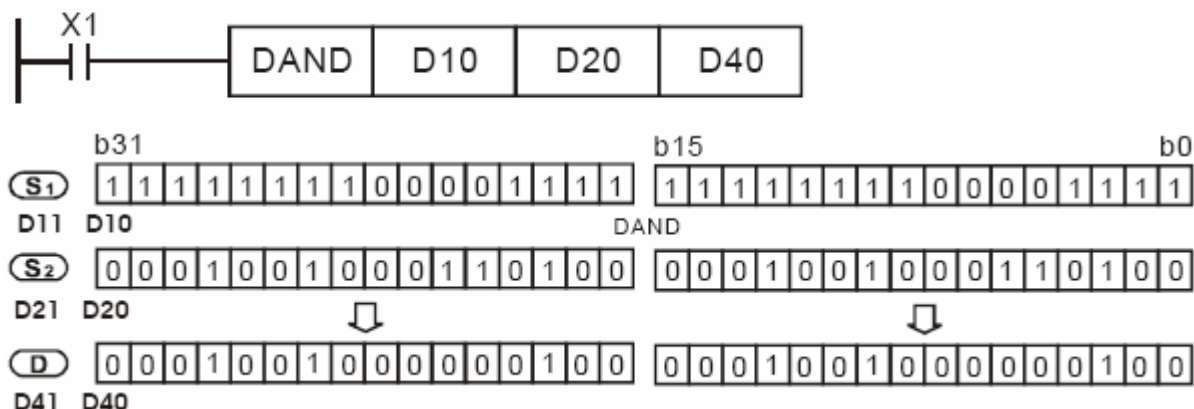
Пример: Программирование WAND-инструкции (16 бит)

Когда X0=1, производится логическое умножение данных в регистрах D0 и D2. Результат операции сохраняется в регистре D4.



Пример: Программирование DAND-инструкции (32 бит)

Когда X1=1, производится логическое умножение данных в регистрах (D11, D10) и (D21, D20). Результат операции сохраняется в регистрах (D41, D40).



API	W	OR		<div>S₁</div> <div>S₂</div> <div>D</div>			Логическое сложение числовых данных (ИЛИ)		DVP-		
27	D		P			ES/EX/SS	SA/SX	EH			
									+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). WOR - Непрерывное выполнение. WORP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S ₁					*	*	*	*	*	*	*	*	*	*	*
S ₂					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

Примечания: Если операнды S₁, S₂, D используются с индексом F, возможно только 16-ти битное выполнение команды.

Импульсное выполнение инструкции WORP в серии ES/EX/SS не поддерживается.

32-х битная инструкция
(13 шагов)
DOR - Непрерывное
выполнение.
DORP – Имп. выполн.

Флаги: нет

Функция

Операция "логическое ИЛИ" для числовых данных.

Описание

- Операция "логическое ИЛИ" для числовых данных выполняется по отдельным битам.
- Данные в (S1) и (S2) побитно логически связываются друг с другом. Результат связи сохраняется в (D).
- Таблица истинности логического сложения

(S1)	(S2)	(D)
1	1	1
1	0	1
0	1	1

Функция

Логическая операция "ИСКЛЮЧАЮЩЕЕ ИЛИ" для числовых данных.

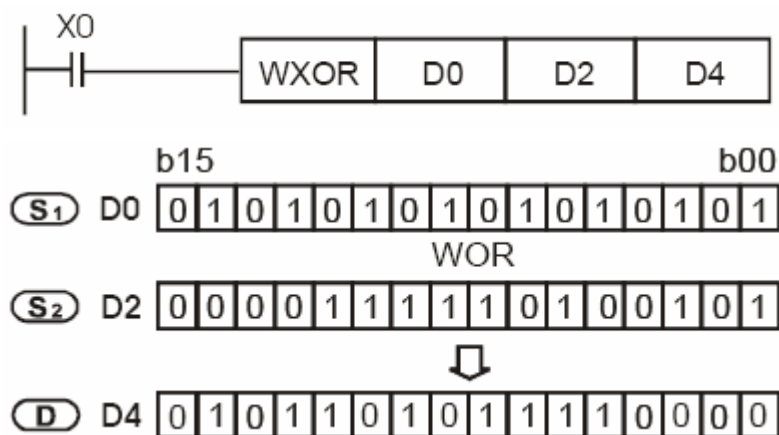
Описание

- Операция "исключающее ИЛИ" для числовых данных выполняется по отдельным битам.
- Данные в (S1) и (S2) побитно логически связываются друг с другом. Результат связи сохраняется в (D).
- Таблица истинности логического сложения

(S1)	(S2)	(D)
1	1	0
1	0	1
0	1	1
0	0	0

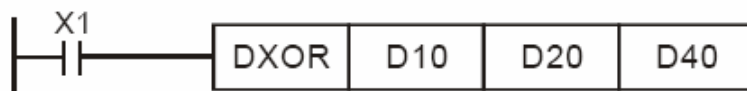
Пример: Программирование WXOR-инструкции (16 бит)

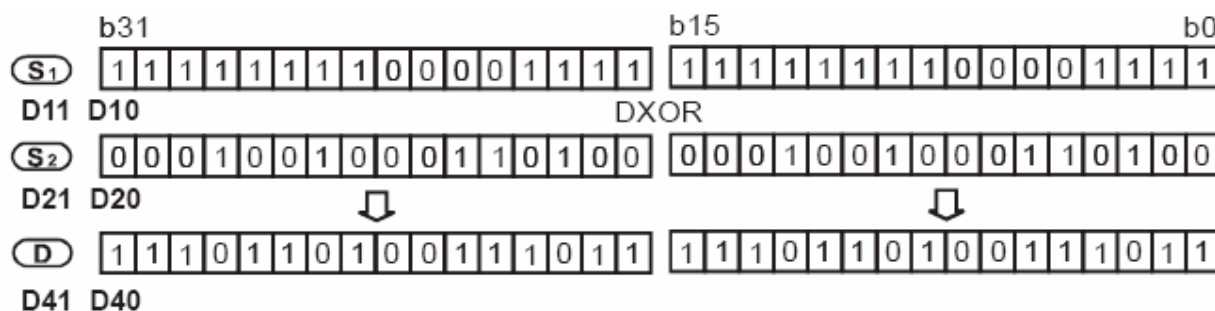
Когда X0=1, производится логическая операция "Исключающее ИЛИ" с данными в регистрах D0 и D2. Результат операции сохраняется в регистре D4.



Пример: Программирование DXOR-инструкции (32 бит)

Когда X1=1, производится логическая операция "Исключающее ИЛИ" с данными в регистрах (D11, D10) и (D21, D20). Результат операции сохраняется в регистрах (D41, D40).





API	W	NEG	☺	ⓓ	Логическое отрицание	DVP-		
29	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (3 шага). WNEG - Непрерывное выполнение. WNEGP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	

Примечания: Если операнд, D используется с индексом F, возможно только 16-ти битное выполнение команды.

Импульсное выполнение инструкции WNEGP в серии ES/EX/SS не поддерживается.

32-х битная инструкция
(5 шагов)
DNEG - Непрерывное
выполнение.
DNEGP – Имп. выполн.

Флаги: нет

Функция

Операция логического отрицания (инверсия всех битов в BIN-формате и сложение с 1) для числовых данных.

Описание

- С помощью инструкции NEG образуется дополнение до 2 значения данных, записанных в (D), и оно сохраняется в том же операнде (D).

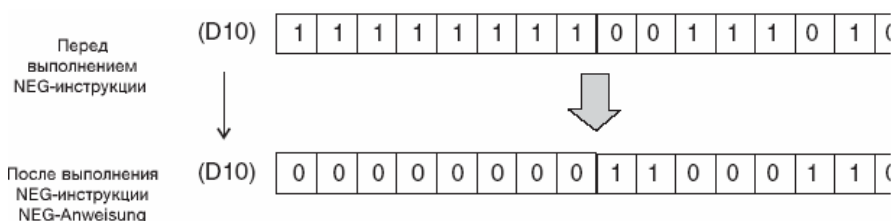
Если не программируется никакого опознавания фронта, то образование дополнения повторяется в каждом цикле программы.

Пример: Программирование NEG-инструкции (16 бит)

Когда X0 включится, произойдет инвертирование всех битов в регистре D0 (0→1, 1→0) и к результату прибавится число 1.

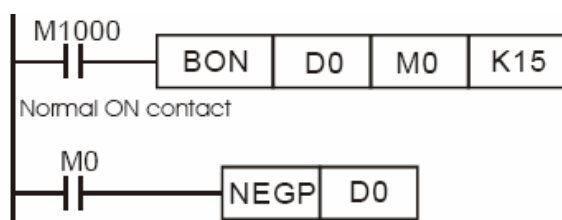


Двоичное: $\overline{D10} + 1 \rightarrow D10$



Пример: Получение абсолютного значения числа с помощью XOR-инструкции:

Если 15 бит в регистре D0 будут равны 1, т.е. значение D0 – отрицательное, включится реле M0 с помощью инструкции NEGP сделает 15 бит = 0, и тем самым число D0 будет положительным.



(D0=2)

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

(D0=1)

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D0=0)

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(D0=-1)

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$\overline{(D0)}+1=1$

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D0=-2)

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0

$\overline{(D0)}+1=2$

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

(D0=-3)

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1

$\overline{(D0)}+1=3$

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

(D0=-4)

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0

$\overline{(D0)}+1=4$

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

(D0=-5)

1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1

$\overline{(D0)}+1=5$

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

⋮

⋮

(D0=-32,765)

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

$\overline{(D0)}+1=32,765$

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1

(D0=-32,766)

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

$\overline{(D0)}+1=32,766$

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0

(D0=-32,767)

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

$\overline{(D0)}+1=32,767$

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

(D0=-32,768)

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

$\overline{(D0)}+1=32,768$

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

API		ROR			Кольцевой сдвиг вправо	DVP-		
						ES/EX/SS	SA/SX	EH
30	D		P			+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (5 шагов). ROR - Непрерывное выполнение. RORP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
D								*	*	*	*	*	*	*	*
<p><u>Примечания:</u> Если операнд, D используется с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Если операнд, D это KnY, KnM, KnS, возможно только Kn = K4 (16 бит) или Kn = K8 (32 бит).</p> <p>Необходимое условие: $1 \leq n \leq 16$ (16 бит), $1 \leq n \leq 32$ (32 бит)</p> <p>Импульсное выполнение инструкции RORP в серии ES/EX/SS не поддерживается.</p> <p><u>32-х битная инструкция</u> (9 шагов) DROR - Непрерывное выполнение. DRORP – Имп. выполн.</p> <p>Флаги: M1022 (флаг переноса)</p>															

Функция

Ротация бит по (n) мест вправо

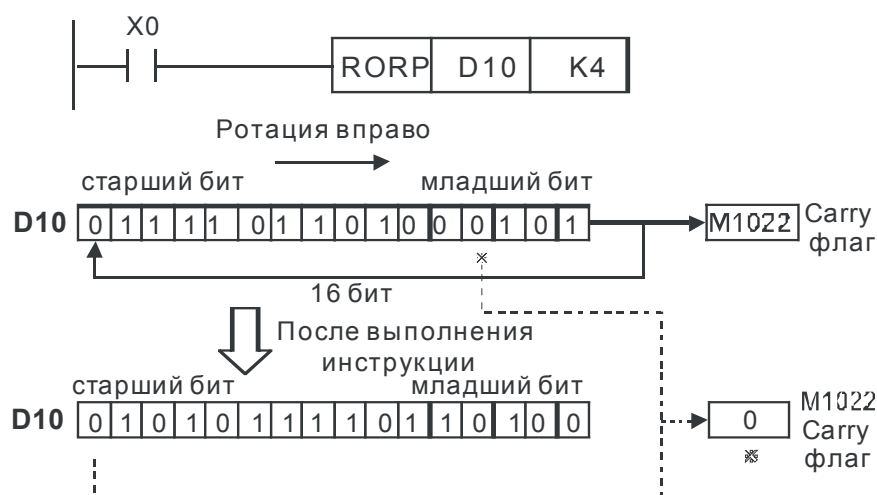
Описание

- Битовое отображение в (D) сдвигается вправо на n мест при каждом исполнении ROR
- Состояние последнего сдвигаемого бита копируются в M1022 (флаг переноса - Carry)

Если не программируется никакого опознания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

Пример: Программирование ROR-инструкции

Битовые данные в регистре данных D10 каждый раз сдвигаются вправо на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ. Значение последнего сдвигаемого бита запоминается во флаге переноса (M1022).



API		ROL		 	Кольцевой сдвиг влево	DVP-		
						ES/EX/SS	SA/SX	EH
31	D		P			+	+	+

Оп- ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
D								*	*	*	*	*	*	*	*
<p><u>Примечания:</u> Если операнд, D используется с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Если операнд, D это KnY, KnM, KnS, возможно только Kn = K4 (16 бит) или Kn = K8 (32 бит).</p> <p>Необходимое условие: $1 \leq n \leq 16$ (16 бит), $1 \leq n \leq 32$ (32 бит)</p> <p>Импульсное выполнение инструкции ROLP в серии ES/EX/SS не поддерживается.</p>															

<u>16-ти битная инструкция</u> (5 шагов). ROL - Непрерывное выполнение. ROLP – Имп. выполн.
<u>32-х битная инструкция</u> (9 шагов) DROL - Непрерывное выполнение. DROLP – Имп. выполн.
Флаги: M1022 (флаг переноса)

Функция

Ротация бит по (n) мест влево

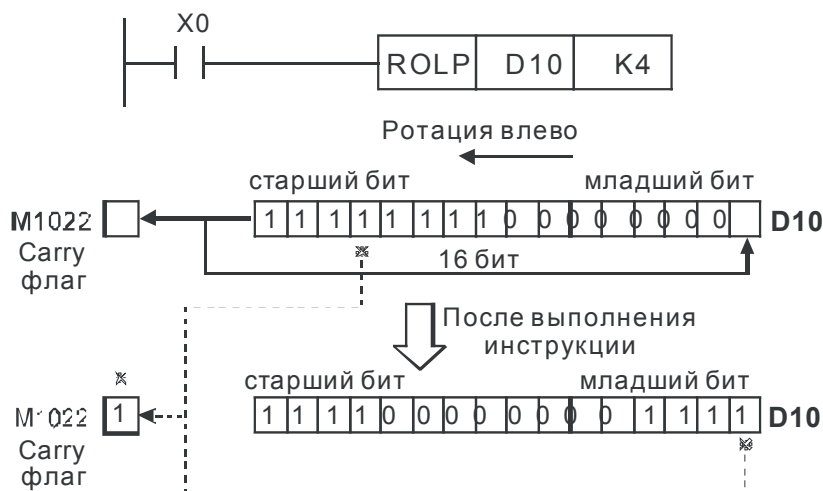
Описание

- Битовое отображение в (D) сдвигается влево на n мест при каждом исполнении ROL
- Состояние последнего сдвигаемого бита копируются в M1022 (флаг переноса - Carry)

Если не программируется никакого опознания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

Пример: Программирование ROL-инструкции

Битовые данные в регистре данных D10 каждый раз сдвигаются влево на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ. Значение последнего сдвигаемого бита запоминается во флаге переноса (M1022).



API		R	☺	D n		Кольцевой сдвиг вправо с использованием флага переноса	DVP-		
32	D		P				ES/EX/SS	SA/SX	EH
							+	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (5 шагов).
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RCR - Непрерывное выполнение. RCRP – Имп. выполн.
D								*	*	*	*	*	*	*	*	
<p><u>Примечания:</u> Если операнд, D используется с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Если операнд, D это KnY, KnM, KnS, возможно только Kn = K4 (16 бит) или Kn = K8 (32 бит).</p> <p>Необходимое условие: $1 \leq n \leq 16$ (16 бит), $1 \leq n \leq 32$ (32 бит)</p> <p>Импульсное выполнение инструкции RCRP в серии ES/EX/SS не поддерживается.</p> <p><u>32-х битная инструкция</u> (9 шагов) DRCR - Непрерывное выполнение. DRCRP – Имп. выполн.</p> <p>Флаги: M1022 (флаг переноса)</p>																

Функция

Ротация бит по (n) мест вправо с использованием флага переноса (Carry) M1022

Описание

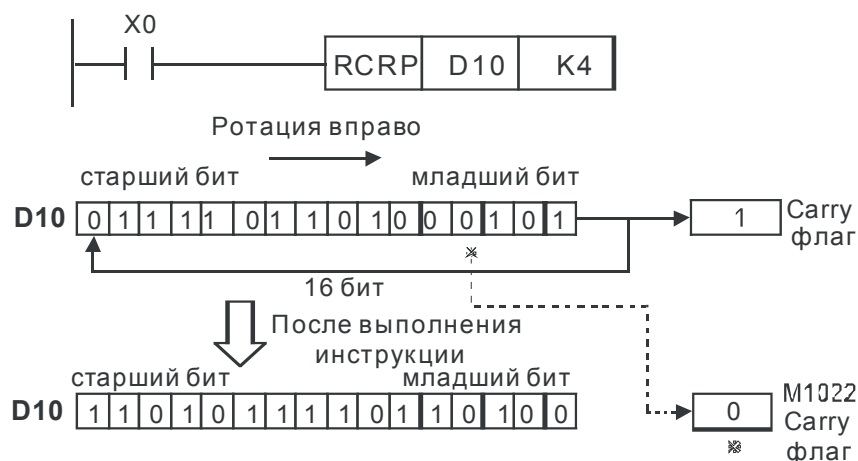
- Битовое отображение в (D) сдвигается вправо на n мест при каждом исполнении ROR
- Флаг переноса вводится в петлю смещения.
- Если в (D) должен быть использован операнд, составленный из битов, то для их составления используются только константы K4 (16-ти битный операнд) и Kn = K8 (32-х битный операнд).

Если не программируется никакого опознавания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

Состояние бита-переноса при включении инструкции воспринимается вместе со сдвигаемым битовым отображением.

Пример: Программирование RCR-инструкции

Битовые данные в регистре данных D10 каждый раз сдвигаются вправо на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ. Значение последнего сдвигаемого бита запоминается во флаге передачи (M1022).



API		RCL			Кольцевой сдвиг влево с использованием флага переноса	DVP-		
						ES/EX/SS	SA/SX	EH
33	D		P			+	+	+

Опе- ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
D								*	*	*	*	*	*	*	*
<p><u>Примечания:</u> Если операнд, D используется с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Если операнд, D это KnY, KnM, KnS, возможно только Kn = K4 (16 бит) или Kn = K8 (32 бит).</p> <p>Необходимое условие: $1 \leq n \leq 16$ (16 бит), $1 \leq n \leq 32$ (32 бит)</p> <p>Импульсное выполнение инструкции RCLP в серии ES/EX/SS не поддерживается.</p>															

<p><u>16-ти битная инструкция</u> (5 шагов). RCL - Непрерывное выполнение. RCLP – Имп. выполн.</p>
<p><u>32-х битная инструкция</u> (9 шагов) DRLR - Непрерывное выполнение. DRCLP – Имп. выполн.</p>
<p>Флаги: M1022 (флаг переноса)</p>

Функция

Ротация бит по (n) мест влево с использованием флага переноса (Carry) M1022

Описание

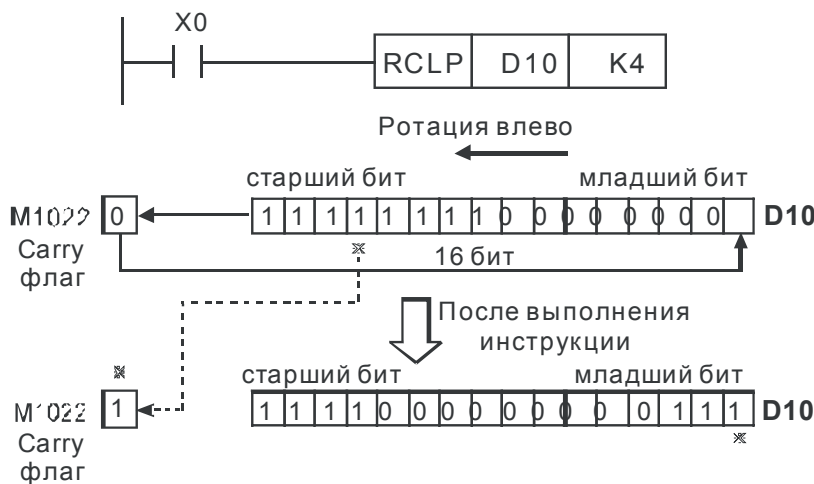
- Битовое отображение в (D) сдвигается влево на n мест при каждом исполнении ROR
- Флаг переноса вводится в петлю смещения.
- Если в (D) должен быть использован операнд, составленный из битов, то для их составления используются только константы K4 (16-ти битный операнд) и Kn = K8 (32-х битный операнд).

Если не программируется никакого опознавания фронта, то сдвиг битового отображения повторяется в каждом цикле программы.

Состояние бита-переноса при включении инструкции воспринимается вместе со сдвигаемым битовым отображением.

Пример: Программирование RCL-инструкции

Битовые данные в регистре данных D10 каждый раз сдвигаются влево на 4 бита (K4), когда вход X0 переходит из состояния ОТКЛ. в состояние ВКЛ. Значение последнего сдвигаемого бита запоминается во флаге передачи (M1022).



API	SFTR			Сдвиг значений битовых устройств вправо	DVP-		
					ES/EX/SS	SA/SX	EH
34		P			+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (9 шагов). SFTR - Непрерывное выполнение. SFTRP – Имп. выполн. <u>32-х битная инструкция</u> ---. Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												
D		*	*	*												
n ₁					*	*										
n ₂					*	*										
<u>Примечания:</u> Необходимое условие: $1 \leq n_1 \leq 1024$, $1 \leq n_2 \leq n_1$ в серии ES/EX/SS: $1 \leq n_2 \leq n_1 \leq 512$ Импульсное выполнение инструкции SFTRP в серии ES/EX/SS не поддерживается.																

Функция

Сдвиг значений бит вправо

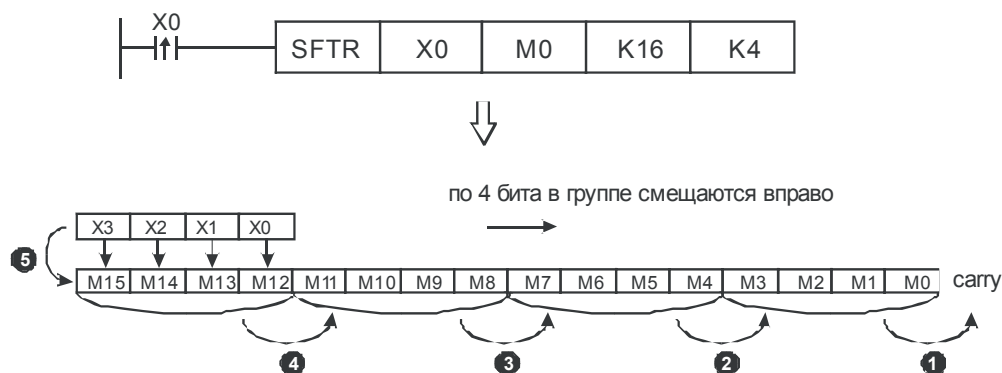
Описание:

- Эта инструкция касается 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.

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

При срабатывании X0 двоичные сигналы, находящиеся на входах X0...X3 побитно считываются в указанную область внутренних реле и соответственно сдвигаются вправо.



В течении одного скана выполняются следующие 5 шагов:

- ❶ M3~M0 → carry
- ❷ M7~M4 → M3~M0
- ❸ M11~M8 → M7~M4
- ❹ M15~M12 → M11~M8
- ❺ X3~X0 → M15~M12

API	SFTL		   	Сдвиг значений битовых устройств влево	DVP-		
					ES/EX/SS	SA/SX	EH
35		P			+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (9 шагов). SFTL - Непрерывное выполнение. SFTRL – Имп. выполн. 32-х битная инструкция ---. Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												
D		*	*	*												
n ₁					*	*										
n ₂					*	*										
<u>Примечания:</u> Необходимое условие: $1 \leq n_1 \leq 1024$, $1 \leq n_2 \leq n_1$ в серии ES/EX/SS: $1 \leq n_2 \leq n_1 \leq 512$ Импульсное выполнение инструкции SFTLP в серии ES/EX/SS не поддерживается.																

Функция

Сдвиг значений бит влево

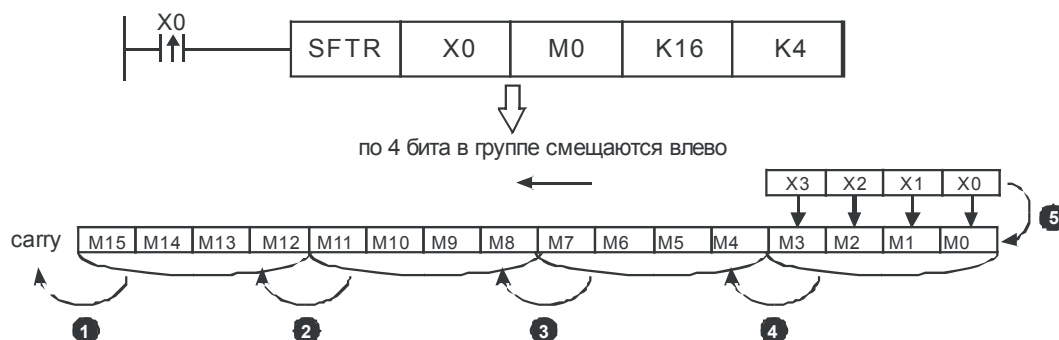
Описание:

- Эта инструкция касается 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.

Инструкция выполняется в каждом цикле программы. Этого можно избежать введением впереди функции импульса (PLS- или PLF- инструкции) или применением командного параметра P.

Пример: Применение SFTL-инструкции

При срабатывании X0 двоичные сигналы, находящиеся на входах X0...X3 побитно считываются в указанную область внутренних реле и соответственно сдвигаются влево.



В течение одного скана выполняются следующие 5 шагов:

- M15~M12 → carry
- M11~M8 → M15~M12
- M7~M4 → M11~M8
- M3~M0 → M7~M4
- X3~X0 → M3~M0 complete

API		WSFR	☺	S D n1 n2				Пословный сдвиг данных вправо		DVP-		
36			P							ES/EX/SS	SA/SX	EH
										-	+	+

Оп- еранд	Биты				Слова										<u>16-ти битная инструкция</u> (9 шагов). WSFR - Непрерывное выполнение. WSFRP – Имп. выполн. <u>32-х битная инструкция</u> ---. Флаги: нет		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
	S						*	*	*	*	*	*	*				
	D							*	*	*	*	*	*				
	n1				*	*											
n2					*	*											
Примечания: При применении объединенных битовых операндов необходимо следить затем, чтобы (S) и (D) располагали одинаковым числом бит. Необходимое условие: 1≤n1≤512, 1≤n2≤ n1																	

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.

Инструкция выполняется в каждом цикле программы. Этого можно избежать введением впереди функции импульса (PLS- или PLF- инструкции) или применением командного параметра P.

Пример: Применение WSFR-инструкции для регистров:

При срабатывании X0 данные, находящиеся в регистрах D10 ... D13 пословно считываются в указанную область адресов регистров и соответственно сдвигаются вправо.



В течение одного скана выполняются следующие 5 шагов:

- ① D23~D20 → carry
- ② D27~D24 → D23~D20
- ③ D31~D28 → D27~D24
- ④ D35~D32 → D31~D28
- ⑤ D13~D10 → D35~D32

Пример: Применение WSFR-инструкции для объединенных битовых операндов:

При срабатывании X0, данные, X20 ... X27 параллельно сдвигаются на два числа вправо.



В течение одного скана выполняются следующие 3 шага:

- ❶ Y17~Y10 → carry
- ❷ Y27~Y20 → Y17~Y10
- ❸ X27~X20 → Y27~Y20

API	WSFL	☺ P	S D n1 n2	Пословный сдвиг данных влево	DVP-		
					ES/EX/SS	SA/SX	EH
37					-	+	+

Оп- код	Биты				Слова											16-ти битная инструкция (9 шагов). WSFL - Непрерывное выполнение. WSFLP – Импульс. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n ₁					*	*										
n ₂					*	*										
Примечания: При применении объединенных битовых операндов необходимо следить затем, чтобы (S) и (D) располагали одинаковым числом бит. Необходимое условие: 1≤n ₁ ≤512, 1≤n ₂ ≤ n ₁																32-х битная инструкция ---. Флаги: нет

Функция

Данные пословно сдвигаются влево

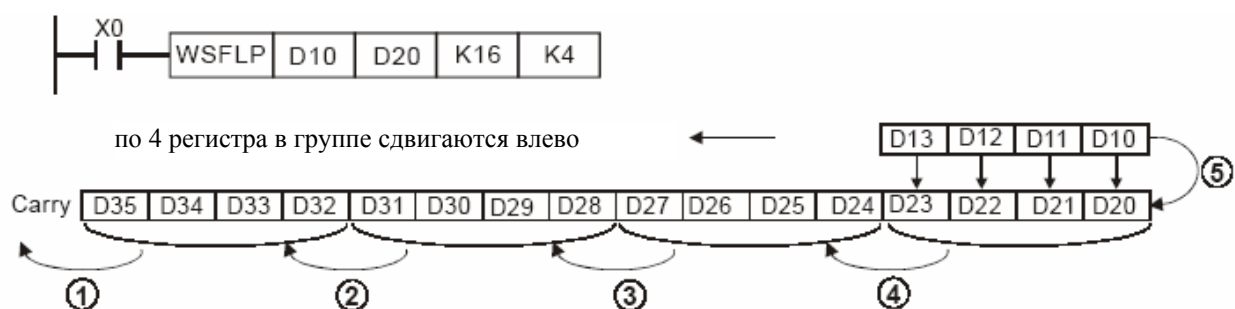
Описание

- 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.

Инструкция выполняется в каждом цикле программы. Этого можно избежать введением впереди функции импульса (PLS- или PLF- инструкции) или применением командного параметра P.

Пример: Применение WSFL-инструкции:

При срабатывании X0 данные, находящиеся в регистрах D10 ... D13 пословно считываются в указанную область адресов регистров и соответственно сдвигаются вправо.



В течение одного скана выполняются следующие 5 шагов:

- ① D35~D32 → carry
- ② D31~D28 → D35~D32
- ③ D27~D24 → D31~D28
- ④ D23~D20 → D27~D24
- ⑤ D13~D10 → D23~D20

API		SFWR	☺	<div>S</div> <div>D</div> <div>n</div>			Запись данных в стек	DVP-		
38			P	ES/EX/SS	SA/SX	EH				
				-	+	+				

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). SFWR - Непрерывное выполнение. SFWRP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*		
n					*	*									

Примечания: Необходимое условие: $2 \leq n \leq 512$

32-х битная инструкция
---.

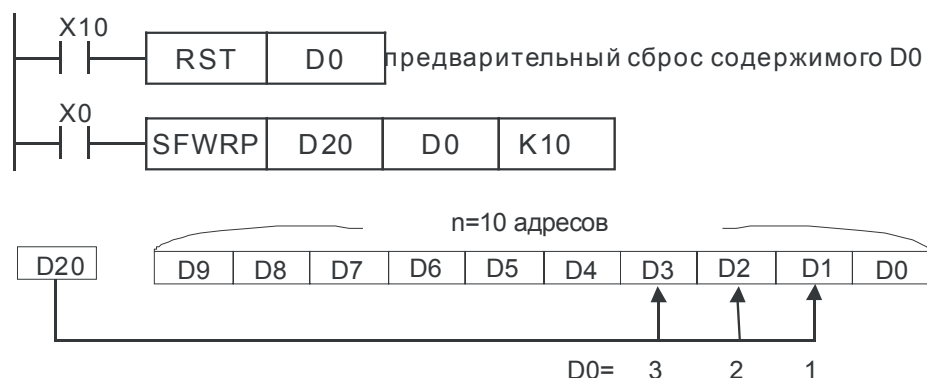
Флаги: M1022 (флаг переноса)

Функция

Запись данных стековую память типа FIFO (First-in/First-OUT)

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 (M1022)).
- При каждом исполнении инструкции выполняется приращение указателя (D).
- Инструкция применяется совместно с инструкцией SFRD; параметр (n) должен быть одинаковым в обеих инструкциях.

Пример: Применение SFWR-инструкции

API		SFRD					Чтение данных из стека	DVP-		
39			P					ES/EX/SS	SA/SX	EH
								-	+	+

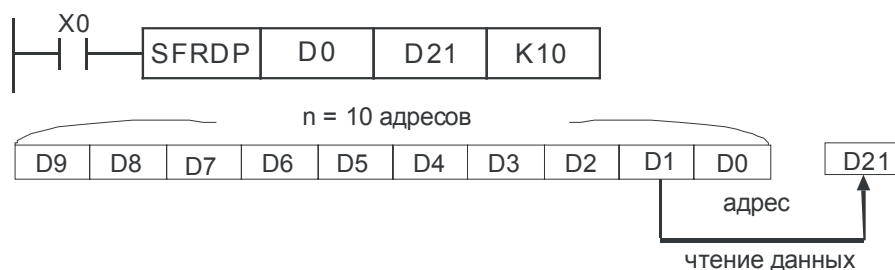
Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). SFRD - Непрерывное выполнение. SFRDP – Импл. выполн.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S								*	*	*	*	*	*			
D								*	*	*	*	*	*	*	*	
n					*	*										
<u>Примечания:</u> Необходимое условие: $2 \leq n \leq 512$																32-х битная инструкция ---. Флаги: M1022 (флаг переноса)

Функция

Чтение данных из стековой памяти типа FIFO (First-in/First-OUT)

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 M1020.
- Инструкция SFRD работает совместно с инструкцией SFWR. Параметр (n) должен быть одинаковым в обеих инструкциях.

Пример: Применение SFRD-инструкции

API		ZRST	☺			Групповой сброс операндов	DVP-		
40			P	(D1)	(D2)		ES/EX/SS	SA/SX	EH
							+	+	+

Опе- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (5 шагов). ZRST - Непрерывное выполнение. ZRSTP – Имп. выполн.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
	D ₁		*	*	*							*	*	*		*
D ₂		*	*	*							*	*	*	*	*	
<u>Примечания:</u> Операнды D ₁ и D ₂ должны быть однотипные. Необходимое условие: D ₁ ≤ D ₂ Импульсное выполнение инструкции ZRSTP в серии ES/EX/SS не поддерживается.																<u>32-х битная инструкция</u> --- Флаги: нет

Функция

Значения нескольких следующих друг за другом операндов (область операндов) могут быть сброшены только одной ZRST-инструкцией, т.е битовые устройства отключены, а регистры установлены на действительное значение "0".

Описание

- В (D1) и (D2) определяется область операндов, которые могут быть сброшены.
- В (D1) и (D2) нужно указать одинаковые типы операндов.

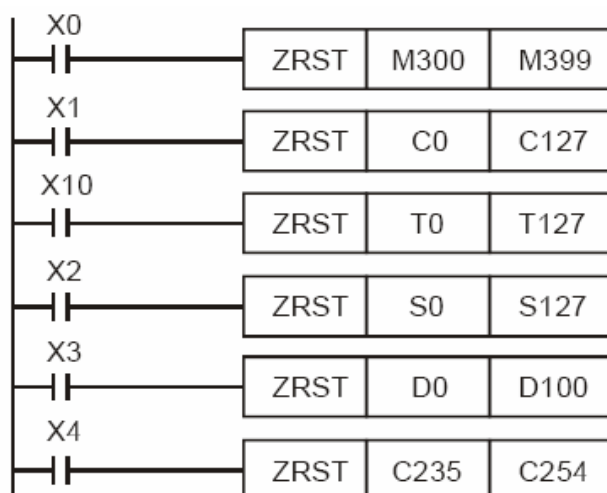
(D1): адрес первого операнда (D2): адрес последнего операнда

Должно соблюдаться: (D1) < (D2)

Если (D1) > (D2), отключается только операнд, указанный в (D1).

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.

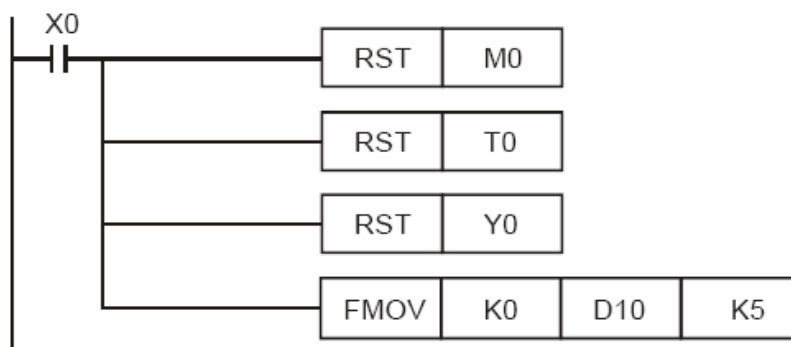
Пример: Применение ZRST-инструкции



При выполнении соответствующих входных условий битовые операнды M300...M399, S0...S127 отключаются до состояния сигнала "0". Словные операнды C0...C127,

C235...254, T0...T127, D0...D10 отключаются до состояния действительного значения "0". Выключаются соответствующие катушки и контакты.

Примечание: Отдельные операнды могут отключаться с помощью RST-инструкции, а групповой сброс словных операндов можно выполнить инструкцией FMOV:



API		DECO	☺	S	D	n	Дешифратор 8 → 256 бит	DVP-		
41			P					ES/EX/SS	SA/SX	EH
								+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). DECO - Непрерывное выполнение. DECOP – Импл. выполн.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
	S	*	*	*	*	*					*	*	*	*			*
	D		*	*	*						*	*	*	*			*
	n					*	*										

Примечания: Когда D битовый операнд: n = 1...8
Когда D словный операнд: n = 1...4
Импульсное выполнение инструкции DECOP в серии ES/EX/SS не поддерживается.

32-х битная инструкция

Флаги: нет

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 дешифрации.

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

2^n : Количество операндов цели

(D): Стартовый адрес операнда цели

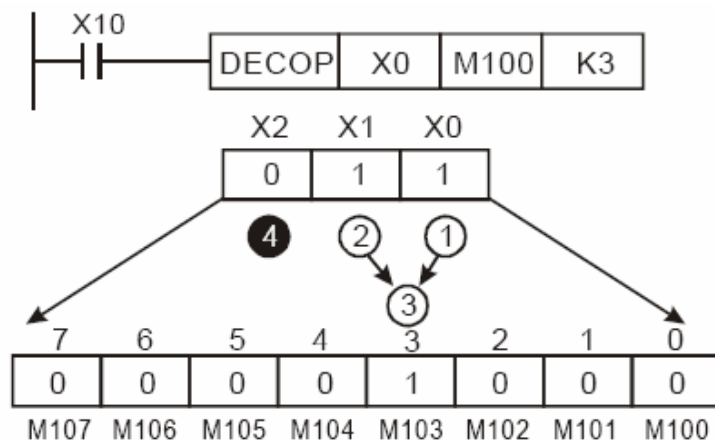
Внимание! Инструкция не выполняется, если n = 0.

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 отработки программы появляется, если n находится не в области от 0 до 8.
- Ошибка при отработки программы появляется, если все биты выходных операндов имеют значение "0".

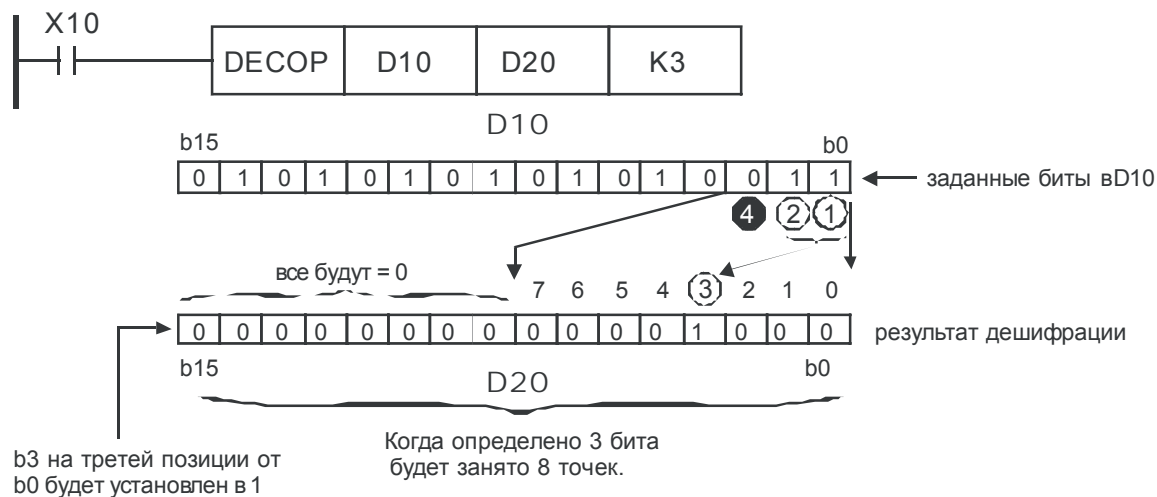
Пример: Применение DECO-инструкции с указанием битовых операндов в D



Если $n = 3$, обрабатываются входные операнды X_0 , X_1 и X_2 . Потому что $2^n = 2^3 = 8$ представлены в качестве адресов цели реле $M100...M107$.





Значения входных операндов равны $1+2 = 3$. Соответственно третий бит адреса цели, т.е. реле $M103$, включается. Если обрабатывается значение входного операнда "0", то включается реле $M100$.

Пример: Применение DECO-инструкции с указанием словных операндов в D



Младшие 3 бита из регистра данных $D10$ декодируются. Результат декодирования $1+2 = 3$ передается в регистр данных $D10$. В этом регистре данных включается 3-й бит.

Если значение для $n < 3$, то все ненужные биты более высокого номера в адресах цели устанавливаются на ноль.

API	ENCO		  	Шифратор 256 → 8 бит	DVP-		
					ES/EX/SS	SA/SX	EH
42		P			+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов). ENCO - Непрерывное выполнение. ENCOP – Импл. выполн.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S	*	*	*	*	*	*					*	*	*	*	*	
D		*	*	*							*	*	*	*	*	
n					*	*										
<div>Примечания: Когда D битовый операнд: n = 1...8</div> <div>Когда D словный операнд: n = 1...4</div> <div>Импульсное выполнение инструкции ENCOP в серии ES/EX/SS не поддерживается.</div>																32-х битная инструкция --- Флаги: нет

Функция

Кодирование данных

Описание

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

2^n : Количество операндов, данные которых должны кодироваться,

n: Число операндов цели

При указании битового операнда в S должно соблюдаться: ($1 \leq n \leq 8$) При указании словного операнда в S+ должно соблюдаться: ($1 \leq n \leq 4$)

(S): Стартовый адрес операндов, данные которых должны кодироваться.

(D): Операнд цели

Если несколько операндов, указанных в (S), имеют значение 1, то обрабатывается только старший бит.

Внимание! 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".

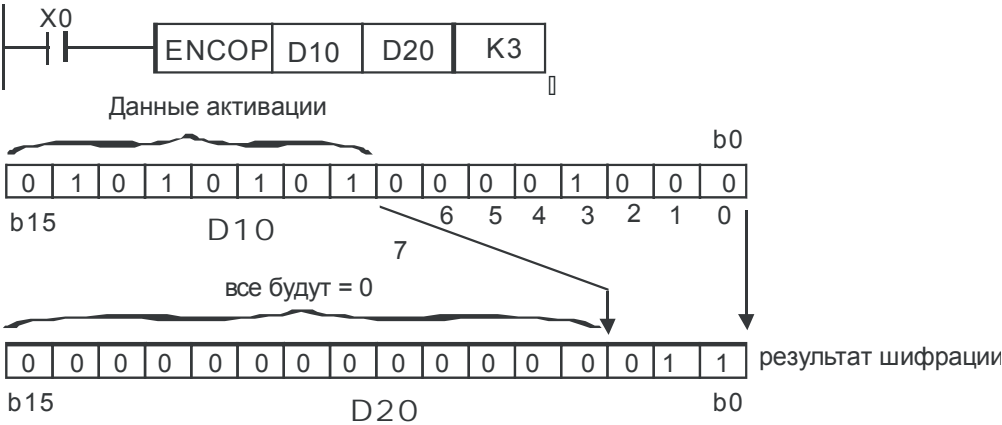
Пример: программирование ENCO-инструкции с указанием битовых операндов в (S)

Если $2^n = 23 = 8$, то в качестве адресов выходов имеются реле M0...M7. Поскольку у операндов выхода 3-ий операнд, т.е. реле M3 включается, в регистр данных D0 записывается значение 3.



Пример: программирование ENCO-инструкции с указанием словных операндов в (S)

В регистре данных D10 включается 3-ий бит. Тем самым значение числа 3 кодируется и сохраняется в регистре данных D10.



API		SUM		<div>S</div> <div>D</div>	Сумма единичных битов	DVP-		
	D		P			ES/EX/SS	SA/SX	EH
43						+	+	+

Опе-ранд	Биты				Слова										16-ти битная инструкция (5 шагов). SUM - Непрерывное выполнение. SUMP – Имп. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*
<div>Примечания: Если операнды S и D используются с индексом F, возможно только 16-ти битное выполнение команды. Импульсное выполнение инструкции SUMP в серии ES/EX/SS не поддерживается.</div> <div>32-х битная инструкция (9 шагов). DSUM - Непрерывное выполнение. DSUMP – Имп. выполн.</div> <div>Флаги: M1020 (флаг ноля)</div>															

Функция

Определение количества активных битов в слове данных.

Описание

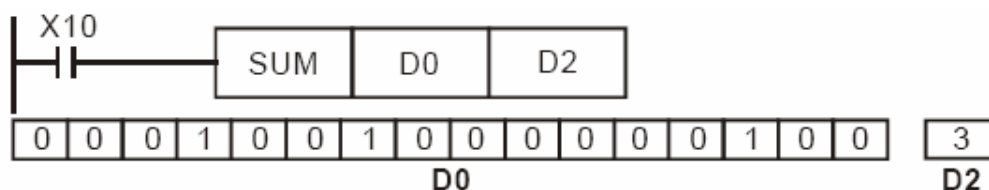
- Определяется количество включенных битов в (S).
- Определенное значение заносится в (D).

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.

Если все биты = 0, выставляется флаг ноля M1020 = 1

Пример: SUM-инструкция

Когда X10 включен, все биты регистра D0, которые равны 1 будут просуммированы и результат занесен в D2.



API		BON		S	D	n	Проверка состояния битов	DVP-		
44	D		P					ES/EX/SS	SA/SX	EH
								+	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (7 шагов).
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BON - Непрерывное выполнение. BONP – Имп. выполн.
S					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												
n					*	*					*	*	*	*	*	
<p><u>Примечания:</u> Если операнд S используется с индексом F, возможно только 16-ти битное выполнение команды.</p> <p>Необходимое условие: n = 0...15 (16 бит), n = 0...31 (32 бит)</p> <p>Импульсное выполнение инструкции BONP в серии ES/EX/SS не поддерживается.</p>																<p><u>32-х битная инструкция</u> (13 шагов).</p> <p>DBON - Непрерывное выполнение. DBONP – Имп. выполн</p> <p>Флаги: нет</p>

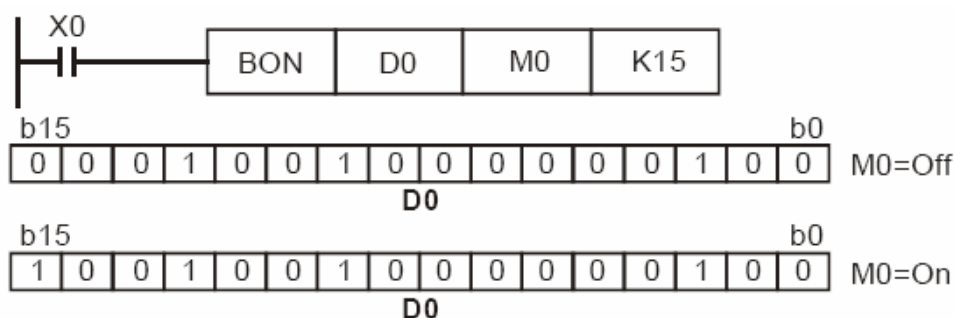
Функция

Проверяется отдельный бит внутри слова данных.

Описание

- Если бит по номеру (n) включен внутри (S), то включается соответствующий бит в (D).

ПРИМЕР: BON-инструкция



API		MEAN		S D n	Среднее арифметическое	DVP-		
						ES/EX/SS	SA/SX	EH
45	D		P			+	+	+

Оп- ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S							*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*	*
n					*	*									

16-ти битная инструкция
(7 шагов).
MEAN - Непрерывное выполнение.
MEANP – Имп. выполн.

32-х битная инструкция
(13 шагов).
DMEAN - Непрерывное выполнение.
DMEANP – Имп. выполн

Флаги: нет

Примечания: Если операнд S используется с индексом F, возможно только 16-ти битное выполнение команды.
Необходимое условие: n = 1...64
Импульсное выполнение инструкции MEANP в серии ES/EX/SS не поддерживается.

Функция

Вычисляется среднее арифметическое значение по нескольким словам данных.

Описание

Начиная с (S) суммируются (n) слов данных и делятся на (n). Целое число результата заносится в (D).

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



$$(D0 + D1 + D2) / D3 \rightarrow D10$$

D0 K100

D1 K113

→ K112 D10

D2 K125

остаток = 3, будет игнорирован

API		ANS		S m D	Сигнализация тревоги с задержкой на включение	DVP-		
46						ES/EX/SS	SA/SX	EH
						-	+	+

Оп-ранд	Биты				Слова										16-ти битная инструкция (7 шагов). ANS - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S											*				
m					*										
D				*											
<div><div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><</div></div></div></div></div>															

Функция

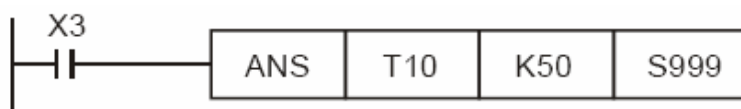
Запуск интервалов времени и включение сигнального бита

Описание

- Запуск интервалов времени и включение сигнального бита.
- При выполнении этой инструкции запускается время (m) x 100 мс.
- После отсчета времени включается сигнальный бит (D).
- В (S) предварительно задается таймер, который определяет интервал времени.

Примененный таймер не должен больше использоваться в последующей программе.

Программирование ANS-инструкции:



Если X3 включится, то запустится отсчет времени в таймере T10, и через 5 сек включится аварийное реле S999. Если X3 выключится, то T10 будет сброшено, а S999 останется включенным. S999 может быть сброшено с помощью инструкции ANR.

API		ANR	☺		Сброс сигнализации	DVP-		
47			P	-		ES/EX/SS	SA/SX	EH
						-	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (1 шаг). ANR - Непрерывное выполнение. ANRP - Импл. вып.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
<u>Примечания:</u> нет операндов																32-х битная инструкция ---

Функция

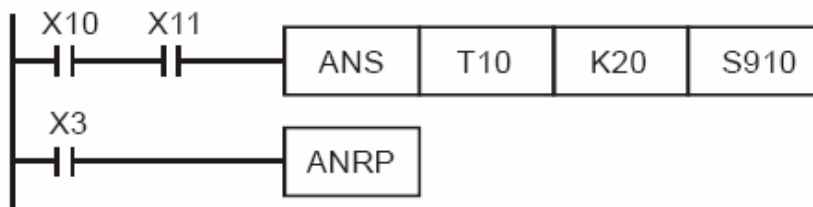
Сброс сигнального бита

Описание

Если инструкция активна, то отключается активный бит тревожной сигнализации с самого младшего адреса.

Инструкция должна выполняться с опцией "P".

Пример: программирование ANR-инструкции:



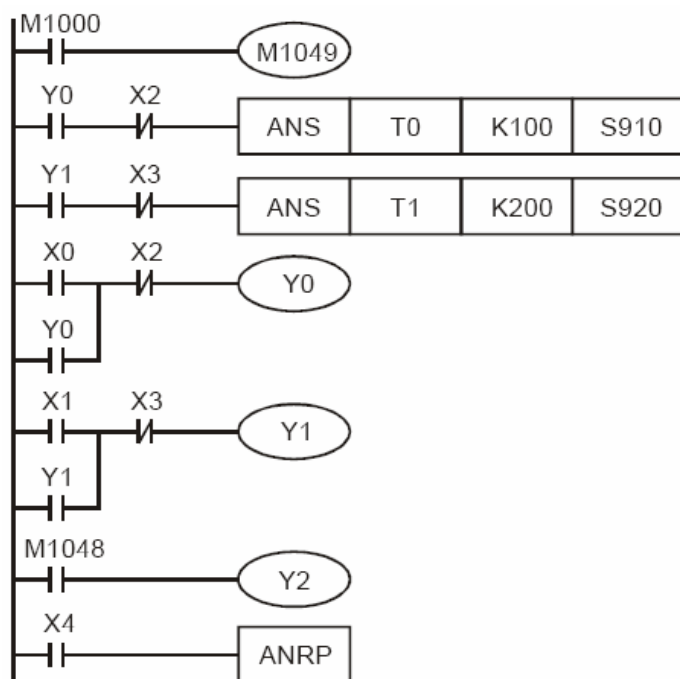
Если включен X3, используемый сигнальный бит S910 отключится.

Флаги:

Если M1049 (Активизация флага тревоги) = 1 и один из операндов S899...S1023 включен, то M1048 (Флаг тревоги) будет установлен в 1 и в специальный регистр D1049 будет записан наименьший номер выполняемой команды.

Пример применения инструкций тревожной сигнализации:

X0: кнопка "Пуск вперед", X1: кнопка "Пуск назад", X2: концевой выключатель движения вперед, X3: концевой выключатель движения назад, X4: кнопка сброса тревожной сигнализации; Y0: движение вперед; Y1: движение назад; Y3: индикация аварии; S910: сигнализация превышения времени движения вперед; S920: сигнализация превышения времени движения назад.



Если Y0 будет включен более 10 сек и концевой выключатель за это время не будет достигнут, S910 будет = 1, и включится Y2. Аналогичным образом Y2 будет включен и если Y1 будет включен более 20 сек.

Сброс аварии можно осуществить кнопкой X4.

API		SQR		<div>S</div> <div>D</div>	Вычисление корня квадратного								DVP-		
48	D		P										ES/EX/SS	SA/SX	EH
													-	+	+

Оп-ранд	Биты				Слова										<u>16-ти битная инструкция</u> (5 шагов). SQR - Непрерывное выполнение. SQRП – Импл. выполн			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E				F
	S					*	*						*					
	D												*					

Примечания: Если операнды S и D используются с индексом F, возможно только 16-ти битное выполнение команды.

32-х битная инструкция
(9 шагов).
DSQR - Непрерывное выполнение.
DSQRП – Импл. выполн

Флаги: M1020 (флага ноля), M1021 (флага заема), M1022 (флага переноса)

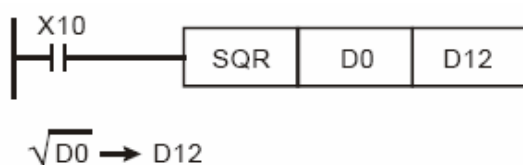
Функция

Вычисление корня квадратного, $(D) = \sqrt{(S)}$

Описание

Выполняется вычисление корня квадратного числа в (S) и с округлением до целого числа результат заносится в (D). Операция выполняется в BIN-формате.

Пример: Программирование SQR-инструкции



Если включен вход X10, рассчитывается значение корня квадратного в регистре данных D0 и результат как округленное целое число записывается в регистр данных D12.

Внимание! Корень квадратный из отрицательного числа всегда приводит к ошибке и включается флаг ошибки M1067.

API		FLT		<div>S</div> <div>D</div>	Преобразование целого числа в число с плавающей точкой								DVP-		
49	D		P										ES/EX/SS	SA/SX	EH
													-	-	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (5 шагов). FLT - Непрерывное выполнение. FLTP – Имп. выполн <u>32-х битная инструкция</u> (9 шагов). DFLT - Непрерывное выполнение. DFLTP – Имп. выполн Флаги: M1081 (Флаг направления преобраз.)			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E				F
	S												*					
	D												*					
Примечания:																		

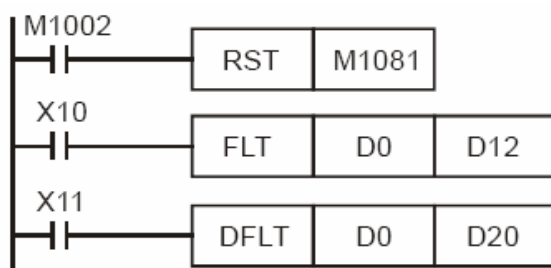
Функция

Преобразование чисел целого числа BIN-формата в формат числа с плавающей запятой.

Описание

- Целое число в (S) преобразовывается в число с плавающей запятой и заносится в (D).
- Результат преобразования (если M1081=0) будет записываться в 32-х битный регистр данных.
- Если результат преобразования = 0, то включится флаг ноля M1020.
- Когда M1081 (Флаг направления преобразования) = 1, будет выполнено обратное преобразование числа с плавающей запятой в целое. При этом (S) будет занимать 2 регистра, а (D) – 1 регистр. Инструкция будет аналогична команде INT.

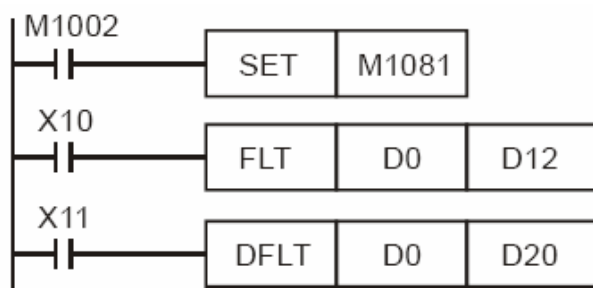
Пример 1: Программирование FLT-инструкции



Когда X10 = 1, целое число регистра D0 будет преобразовано в число с плавающей запятой и записано в регистры D12 (D13). Например, если D0=K10, то D12 (D13) будут равны H41200000.

Когда X11 = 1, целое число регистра D0(D1) будет преобразовано в число с плавающей запятой и записано в регистры D21 (D21). Например, если D0(D1) =K100000, то D12 (D13) будут равны H4735000.

Пример 1: Программирование FLT-инструкции



Когда X10 = 1, число с плавающей запятой D0(D1) будет преобразовано в целое число и записано в регистр D12. Например, если D0(D1)=H47C35000, то D12 будет равно K100000.

Когда X11 = 1, число с плавающей запятой D0(D1) будет преобразовано в целое число и записано в регистры D20(D21). Например, если D0(D1)=H47C35000, то D20(D21) будут равны K100000.

7. ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 50-99

API		REF		 	Обновление состояния ВХОДОВ/ВЫХОДОВ	DVP-		
50			P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова												16-ти битная инструкция (5 шагов). REF - Непрерывное выполнение. REFP – Имп. выполн. <u>32-х битная инструкция</u> --- Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
	D	*	*														
	n					*	*										
<u>Примечания:</u> Операнд D должен быть кратным десяти, например, X0, X10, Y20, Y30,... и т.д. Необходимое условие: n = 8...256, и должен быть кратным 8, например, 8, 16, 24, 32,... и т.д. Импульсное выполнение инструкции REFP в серии ES/EX/SS не поддерживается.																	

Функция

Немедленное обновление состояния входов и выходов

Описание

- Операнд (D) задает начальный адрес обновляемых входов/выходов и должен быть кратным 10: X0, X10, X20, и т. д. Операнд (n) задает количество обновляемых входов/выходов, которое должно быть кратным 8: 8, 16, 24, и т. д..

- Перед началом выполнения цикла программы CPU ПЛК считывает состояние сигналов входов и сохраняет их в специальной области памяти – регистрах отображения состояния входов. Таким образом, обрабатываются не реальные входы, а регистр отображения состояния входов.

После обработки цикла программы (после инструкции END) считываются данные, записанные при обработке программы, из регистров отображения состояния 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 CJ-инструкцией (наивысший номер шага) и относящейся к ней точкой маркировки (наименьший номер шага).

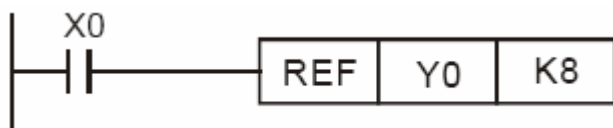
Пример применения REF-инструкции для обновления входов



Обновляется 16 адресов, т.е. входов X0...X15.

Если входы активированы примерно за 10 мс (время задержки) перед обработкой REF-инструкции, то входной регистр отображения активизируется, если выполнится REF-инструкция.

Пример применения REF-инструкции для обновления выходов



Обновляется 8 адресов, т.е. выходов Y0...Y7.

Если выходы включены, то включатся принадлежащие к ним регистры отображения выходов после выполнения REF-инструкции. Контакт реального выхода активизируется по истечению времени ответа. Время ответа является физически обусловленным временем включения активизированного выхода.

Примечание: В контроллерах ES данной командой могут обрабатываться только входы и выходы базового модуля: X0 – X17, Y0 – Y17.

API		REFF		<div>n</div>	Изменение времени входного фильтра	DVP-		
51			P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (3 шага). REFF - Непрерывное выполнение. REFFP – Импл. выполн.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
n					*	*										32-х битная инструкция --- Флаги: нет
Примечания: нет																

Функция

Установка времени фильтра для входов X0 – X17

Описание

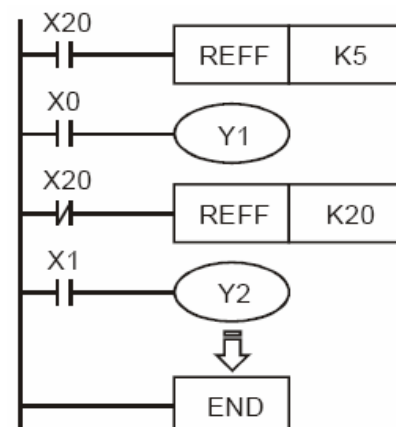
Входной фильтр в ПЛК используется для защиты от дребезга и помех на дискретных входах.

- REF-инструкция устанавливает время фильтра равное (n) мс напрямую в специальные регистры D1020, D1021.

- (n) может задаваться от 0 до 60. Уставка 0 равно времени фильтра 50 мкс.

При включении ПЛК время фильтра для задержки обработки входов определяется значениями регистров D1020, D1021

Если активен вход X20, то уставка фильтра примет значение 5мс, при отключенном – 20мс.



Когда используются входные прерывания, инструкции высокоскоростного счета или команда SPD, то для входов используемых этими командами входной фильтр не действует.

API		MTR		S D₁ D₂ n	Матричный ввод	DVP-		
52						ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (9 шагов). MTR - Непрерывное выполнение. <u>32-х битная инструкция</u> ---. Флаги: M1029 – флаг завершения выполнения
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															
D ₁		*														
D ₂		*	*	*												
n					*	*										
<u>Примечания:</u> Операнд S должен быть кратным 10: X0, X10, X20, и т. д. и занимает 8 адресов. Операнд D ₁ должен быть кратным 10: Y0, Y10, Y20, и т. д. и занимает n адресов. Операнд D ₂ должен быть кратным 10: 0, 10, 20, и т. д. n = 2 ... 8																

Функция

Чтение в ПЛК матрицы размером 8 x n

Описание

- Переключатель 8 x n-матрицы считывается по методу умножения по 8-ми входам и n выходам.
- Входы от (S) до (S+8) распределяются по каждому из (n) рядов внутренних реле. Внутренние реле для первого ряда начинаются с (D2).
- Каждый ряд соответствует выходу; Первый ряд срабатывает на выходе (D1).

Указание! Для выполнения инструкции должен использоваться ПЛК с транзисторными выходами.

Инструкция может использоваться в программе только один раз.

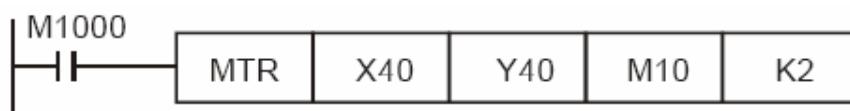
Обработка входов/выходов для каждого выхода выполняется в режиме прерывания с интервалами в 25 мс.

С помощью MTR-инструкции могут восприниматься 64 входных состояния при применении 8 входов и 8 транзисторных выходов. Все входные данные считываются в пределах 200 мс (25 x 8).

Специальное реле M1029 включится, как только матрица будет считана. M1029 отключится, если выключатся входные условия.

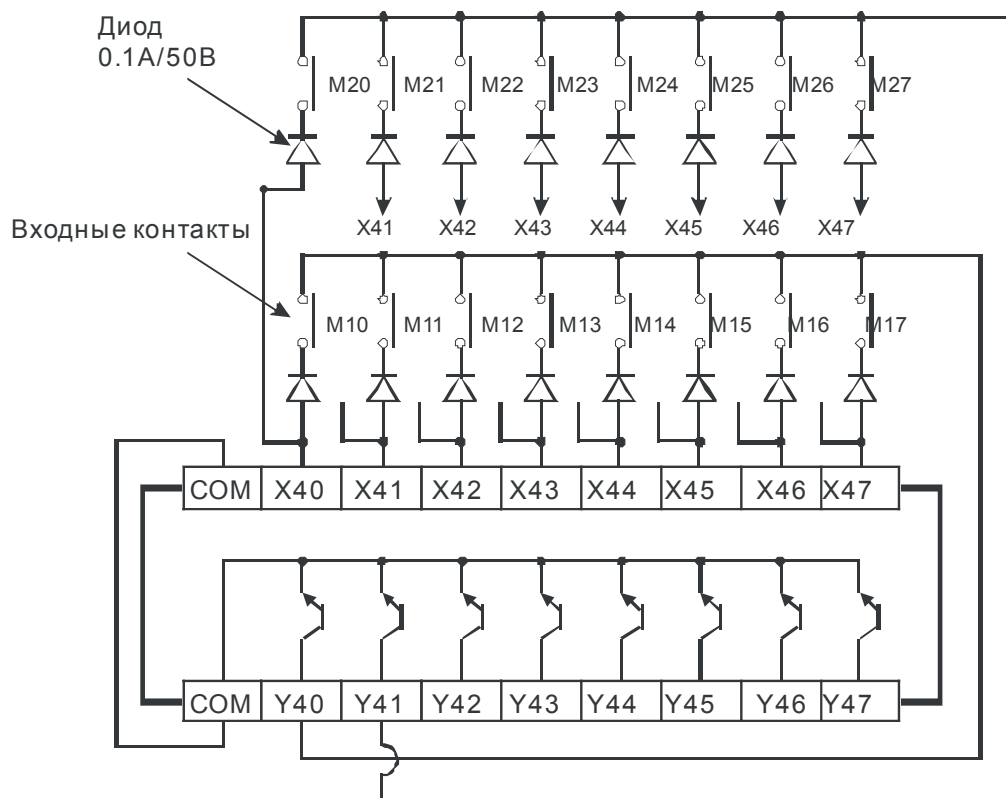
Для исключения конфликтной ситуации в программе в качестве входов по возможности не применять адреса X0...X7.

Пример программирования и применения MTR-инструкции



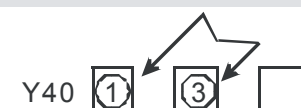
При включении ПЛК будет считываться матрица с адресами столбцов X40...X47 и адресами строк Y40, Y41. Состояния входных 16-ти входных переключающих устройств

(клавиатура 8 x 2) будут записываться во входные реле M10...M17 и M20...M27. M10...M27 остаются неизменными до тех пор, пока не включатся входные условия. Для надежности работы следует последовательно с входными контактами включать диоды (0.1A, 50B).

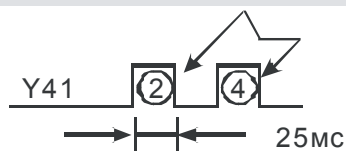


На следующем рисунке видно, что 2 выхода Y40, Y41 включаются один за другим. Этот процесс повторяется постоянно. Данные, воспринимаемые в первой и второй строках, непрерывно передаются к M10...M17 и M20...M27 и запоминаются в них.

Чтение состояния контактов первой строки



Чтение состояния контактов второй строки



API	-	HSCS			Включение от высокоскоростного счетчика	DVP-		
						ES/EX/SS	SA/SX	EH
53	D			(S ₁) (S ₂) (D)		+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂												*				32-х битная инструкция (13 шагов)
D		*	*	*												DHSCS - Непрерывное выполнение.
<u>Примечания:</u> Операнд S ₂ может быть любым доступным регистром высокоскоростного счетчика (C235 – C240, C241 – C244 и др.) Диапазон операнда D: I010...I060, могут используются с индексами E и F в DVP-EH.																Флаги: M1150-M1333

Функция

Включение выхода или внутреннего реле от команды высокоскоростного счетчика. Операнды включаются сразу по окончании выполнения инструкции до окончания цикла программы.

API	-	HSCR			Выключение от высокоскоростного счетчика	DVP-		
						ES/EX/SS	SA/SX	EH
54	D			(S ₁) (S ₂) (D)		+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂												*				32-х битная инструкция (13 шагов)
D		*	*	*								*				DHSCR - Непрерывное выполнение.
<u>Примечания:</u> Операнд S ₂ может быть любым доступным регистром высокоскоростного счетчика (C235 – C240, C241 – C244 и др.) Диапазон операнда D: I010...I060, могут используются с индексами E и F в DVP-EH.																Флаги: M1150-M1333

Функция

Выключение выхода, внутреннего реле или сброса счетчика от команды высокоскоростного счетчика. Операнды выключаются сразу по окончании выполнения инструкции до окончания цикла программы.

Описание инструкций DHSCS/DHSCR

- Высокоскоростной счетчик считает изменение состояния на входах счетчика в режиме прерывания. Каждому высокоскоростному счетчику определены жесткие входы счета с жестко определенными функциями.
- В разделе 2.7. 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.

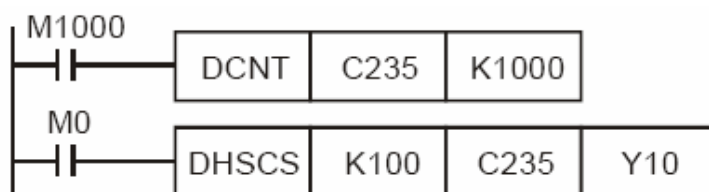
- Операнды включаются до окончания цикла программы непосредственно после выполнения инструкции.
- Инструкция выполняется, если данные в (S1) согласованы с данными в (S2). При этом активизация должна выполняться или по импульсу на счетном входе или на входе сброса. Если активизация должна выполняться по входу сброса, то должно быть включено реле M1261.
- 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 ПЛК не может использоваться больше 4-х DHSCS и DHSCR инструкций одновременно в моделях ES/EX/SS и больше 6-ти в моделях SA/SX.

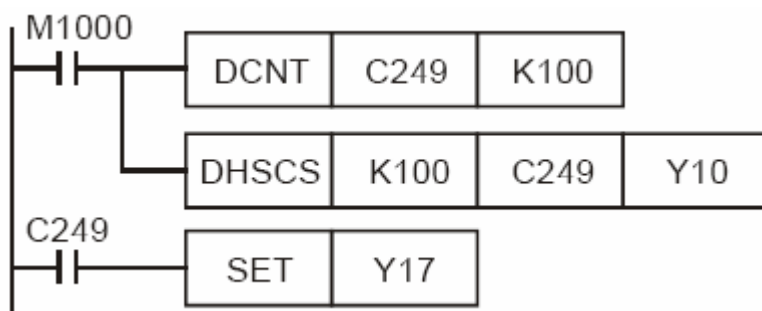
Пример 1: Применение DHSCS-инструкции

Счетным входом высокоскоростного счетчика C235 является X0.

Если M0 включено и накопленное значение счетчика C235 изменяется с 99 на 100 или с 101 на 100, то сразу включится Y10.



Пример 2: Различия между включением выхода DHSCS-инструкцией и SET-инструкцией.



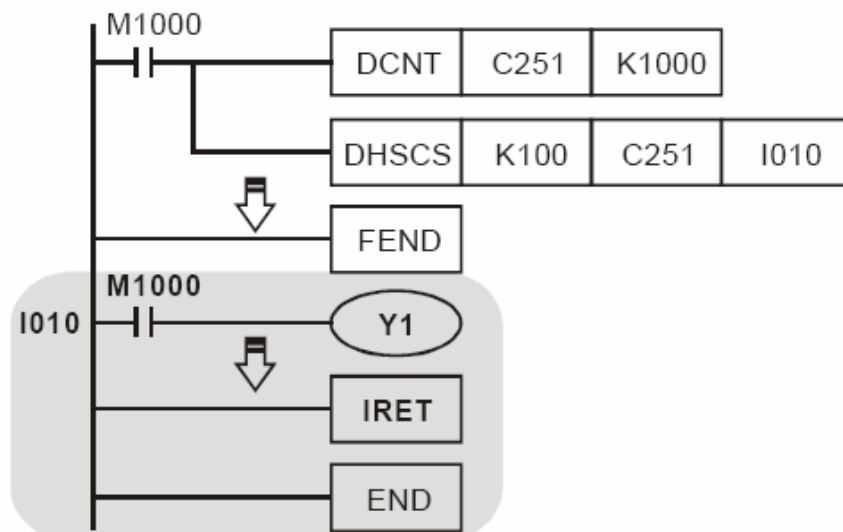
Когда накопленное значение счетчика C249 изменяется с 99 на 100 или с 101 на 100, то выход Y10 включится DHSCS-инструкцией сразу в цикле программы (время включения релейного выхода – 10мс, транзисторного выхода – 10мкс).

Когда накопленное значение счетчика C249 изменяется с 99 на 100 или с 101 на 100, то выход Y17 включится DCNT-инструкцией только в конце цикла программы после инструкции END.

Пример 3: Применение DHSCS-инструкции для обработки прерывания от высокоскоростного счетчика.

Инструкция DHSCS может использоваться для вызова подпрограммы обработки прерывания по достижению высокоскоростным счетчиком заданного значения.

Серия ES не поддерживает эту функцию.

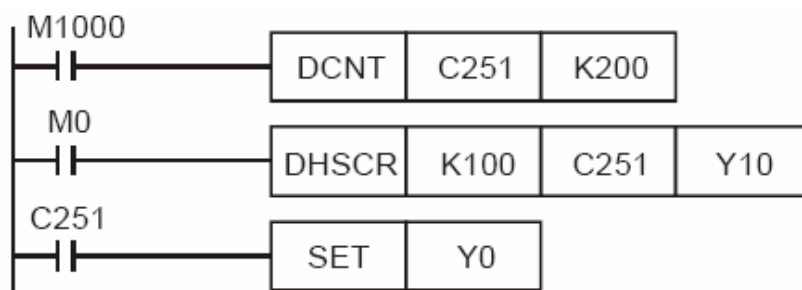


Подпрограмма обработки прерывания, вызванная точкой прерывания I010, выполняется как только значение высокоскоростного счетчика C251 достигнет заданного значения по константе K100.

Для отключения обработки прерывания от высокоскоростного счетчика нужно включать флаг M1059 (в SA/SX) и M1289-M1294 (в EH-серии):

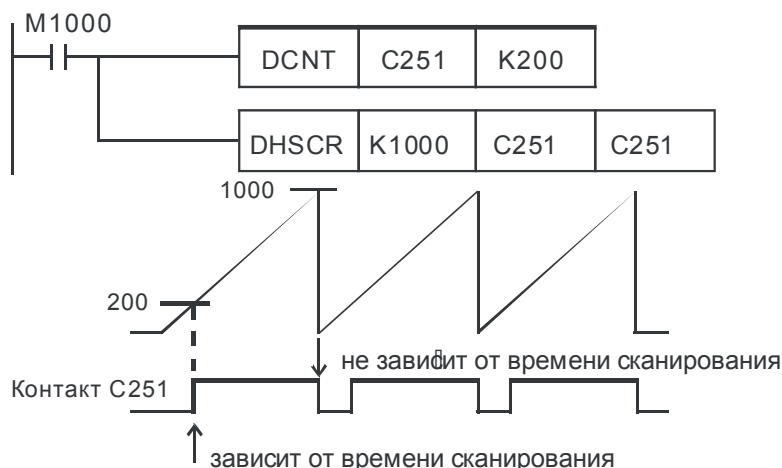
Точка прерывания	Флаг запрета прерывания
I010	M1289
I020	M1290
I030	M1291
I040	M1292
I050	M1293
I060	M1294

Пример 4: Применение DHSCR-инструкции



Когда накопленное значение счетчика C251 изменяется с 99 на 100 или с 101 на 100, то выход Y10 выключится DHSCR-инструкцией сразу в цикле программы.

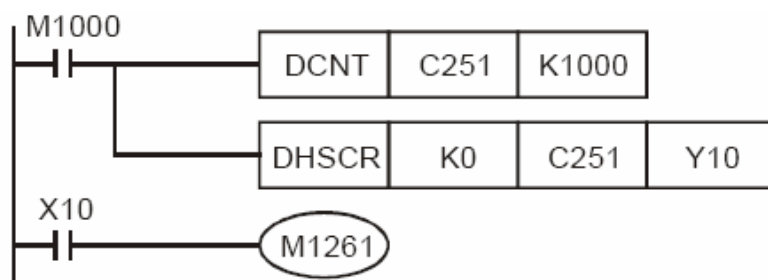
Когда накопленное значение счетчика C251 изменяется с 199 на 200 или с 201 на 200, то выход Y0 включится DCNT-инструкцией только в конце цикла программы после инструкции END.

Пример 5: Применение DHSCR-инструкции для сброса счетчика

Когда накопленное значение счетчика C251 изменится с 199 на 200, включится рабочий контакт C251.

Когда накопленное значение счетчика C251 изменится с 999 на 1000, произойдет сброс контакта C251 и регистр C251 обнулится.

Примечание: Если активизация выхода должна выполняться по входу сброса, то должно быть включено реле M1261 (см. следующий пример). Функция может выполняться только в контроллерах EH и со счетчиками C241-C254.



Если M1261=0, то при поступлении сигнала на вход X2 внешнего сброса счетчика инструкция DHSCR не выполнится и состояние выхода Y10 останется неизменным.

Если M1261=1, то при поступлении сигнала на вход X2 внешнего сброса счетчика инструкция DHSCR выполнится и выход Y10 будет выключен.

Адреса и специальные регистры высокоскоростных счетчиков

Номер	Функция
M1150	Операция группового сравнения DHSZ
M1151	Завершение операции группового сравнения DHSZ
M1152	Операция группового сравнения DHSZ в режиме контроля частоты
M1153	Завершение выполнения режима контроля частоты
M1235 - M1244	Выбор направления счета для счетчиков C235 - C244 (0: суммирование; 1: вычитание)
M1246 – M1249 M1251 – M1254	Индикация направления счета счетчиков C246 – C249 и C251 – C254. (0: суммирование; 1: вычитание)
M1264	Запрещение внешнего старта счетчика HHSC0
M1265	Запрещение внешнего сброса счетчика HHSC0
M1266	Запрещение внешнего старта счетчика HHSC1
M1267	Запрещение внешнего сброса счетчика HHSC1
M1268	Запрещение внешнего старта счетчика HHSC2

M1269	Запрещение внешнего сброса счетчика HHSC2
M1270	Запрещение внешнего старта счетчика HHSC3
M1271	Запрещение внешнего сброса счетчика HHSC3
M1272	Сброс контроля HHSC0
M1273	Старт контроля HHSC0
M1274	Сброс контроля HHSC1
M1275	Старт контроля HHSC1
M1276	Сброс контроля HHSC2
M1277	Старт контроля HHSC2
M1278	Сброс контроля HHSC3
M1279	Старт контроля HHSC3
M1289	I010 флаг запрета прерывания
M1290	I020 флаг запрета прерывания
M1291	I030 флаг запрета прерывания
M1292	I040 флаг запрета прерывания
M1293	I050 флаг запрета прерывания
M1294	I060 флаг запрета прерывания
M1312	C235 старт входа
M1313	C236 старт входа
M1314	C237 старт входа
M1315	C238 старт входа
M1316	C239 старт входа
M1317	C240 старт входа
M1320	C235 сброс входа
M1321	C236 сброс входа
M1322	C237 сброс входа
M1323	C238 сброс входа
M1324	C239 сброс входа
M1325	C240 сброс входа
M1328	C235 разрешение функции старт/сброс
M1329	C236 разрешение функции старт/сброс
M1330	C237 разрешение функции старт/сброс
M1331	C238 разрешение функции старт/сброс
M1332	C239 разрешение функции старт/сброс
M1333	C240 разрешение функции старт/сброс
D1022	Удвоенная частота для двухфазного счетчика AB (DVP-ES/EX/SS/SA/SX)
D1150	Регистр для записи результата в режиме группового сравнения
D1151	Регистр для записи результата в режиме частотного управления
D1152	Изменение значения старшего слова DHSZ
D1153	Изменение значения младшего слова DHSZ
D1225	Первая группа счетчиков (HHSC0). Счетный регистры: C241, C246, C251. Выбор режима
D1226	Вторая группа счетчиков (HHSC1). Счетный регистры: C242, C247, C252. Выбор режима
D1227	Третья группа счетчиков (HHSC2). Счетный регистры: C243, C248, C253. Выбор режима
D1228	Четвертая группа счетчиков (HHSC3). Счетный регистры: C244, C249, C254. Выбор режима
D1225 - D1228	Выбор режима счета для двухфазных высокоскоростных аппаратных счетчиков HHSC0 – HHSC3 контроллеров DVP-EH. 1: нормальная частота счета; 2: двойная частота (заводская уставка); 3: тройная частота; 4: четверная частота счета.

API	-	HSZ		Зонное сравнение для высокоскоростного счетчика	DVP-		
55	D		S₁ S₂ S D		ES/EX/SS	SA/SX	EH
					-	+	+

Оп- еранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S ₁					*	*	*	*	*	*	*	*	*	*	*	<u>32-х битная инструкция</u> (17 шагов) DHSZ - Непрерывное выполнение.
S ₂					*	*	*	*	*	*	*	*	*	*	*	
S												*				
D		*	*	*												
<u>Примечания:</u> Операнд S ₁ должен быть меньше либо равен операнду S ₂ Операнд S может быть любым доступным регистром высокоскоростного счетчика (C235 – C240, C241 – C244 и др.) Операнд D: занимает три адреса, и может использоваться с индексами E и F в DVP-EH.																Флаги: M1150-M1333 (см. описание DHSCS, DHSCR)

Функция

Сравнение текущего значения высокоскоростного счетчика с областью, ограниченной значениями, указанными в (S1/S2)

Описание

- DHSZ-инструкция сравнивает в режиме прерывания накопленное значение высокоскоростного счетчика с областью, указанной 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) = 1, если $S < S1$: текущее значение меньше заданной области S1-S2

(D+1) = 1, если $S1 \leq S \leq S2$: текущее значение внутри заданной области S1-S2

(D+2) = 1, если $S > S2$: текущее значение больше заданной области S1-S2

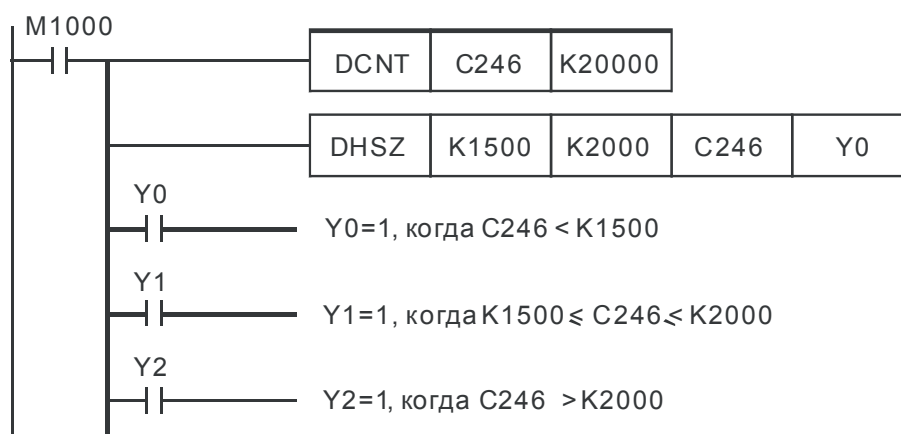
- Так как здесь рассматривается высокоскоростная инструкция, то выходы, которые указываются в (D), сразу выдаются физически.

В программе ПЛК серии SA/SX инструкции DHSCS (API 53), DHSCR (API 54) и DHSZ (API 55) должны применяться не более шести раз при одновременном использовании.

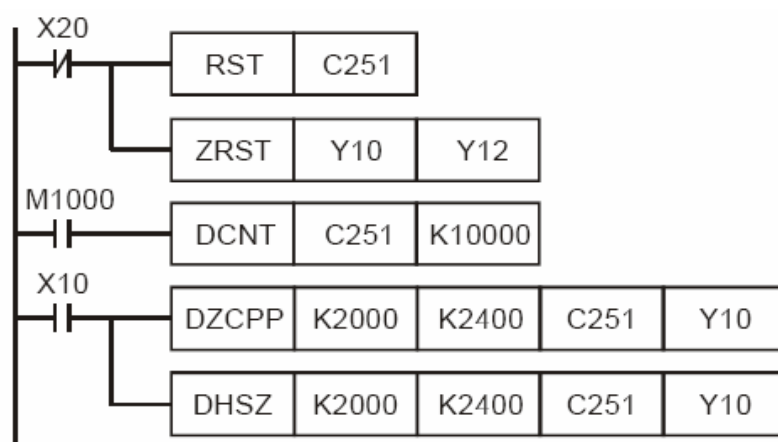
В контроллерах EH не ограничено число использования инструкций высокоскоростного счета: DHSCS (API 53), DHSCR (API 54) и DHSZ (API 55), однако ограничено число их одновременного выполнения. Инструкции DHSCS и DHSCR будут использовать по одной единице памяти, а инструкция DHSZ – две единицы памяти. Надо учесть, что при выполнении одновременно не должно использоваться более восьми единиц памяти. При превышении данного значения инструкции высокоскоростного счета расположенные в программе ниже (тех, что используют 8 ед. памяти) будут проигнорированы.

Пример: Применение DHSZ-инструкции

При попадании текущего значения счетчика C241 в заданные области будут активизироваться соответствующие выходы Y0-Y2:



Пример: Применение DHSZ-инструкции для управления тремя скоростями.



(S1): окончание быстрого хода (пуск медленного хода)

(S2): конец медленного хода (введение торможения)

(S): определение высокоскоростного счетчика (D):

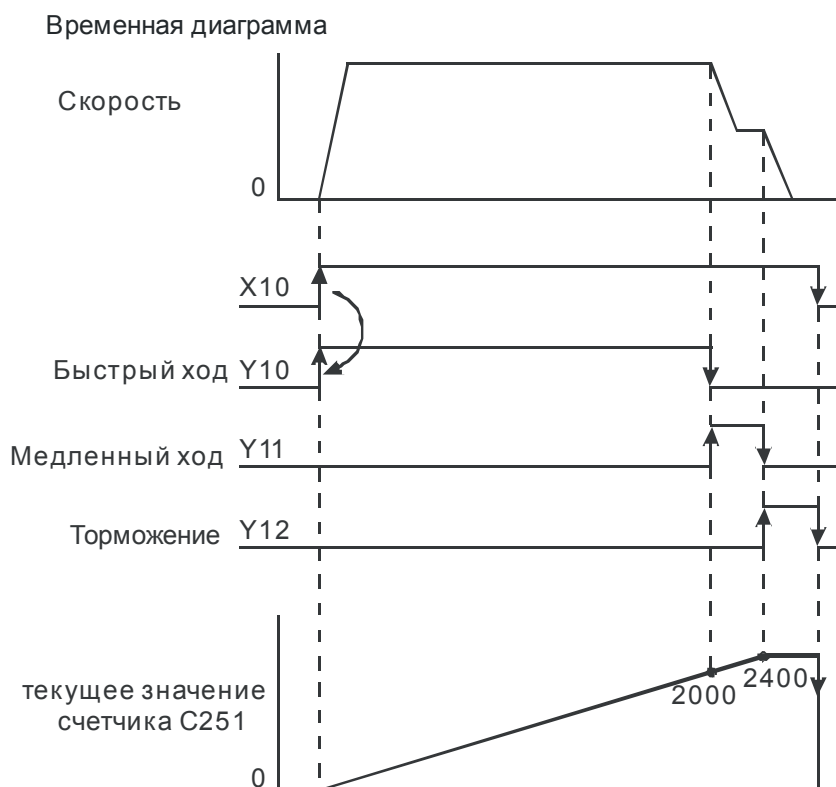
Y10-> быстрый ход

Y11 -> медленный ход

Y12 -> торможение

Процессы счета и сравнения DHSZ-инструкцией, а также внешних выходов выполняются в режиме прерывания. DZCPP-инструкция используется для запуска Y10 в начальный момент времени, когда C251=0.

Временные характеристики включения выходов Y10,Y11,Y12:



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 ZRST-функции).

Принцип функционирования DHSZ-инструкции в режиме группового сравнения (со специальным реле M1150)

- DHSZ-инструкция со специальным реле M1150 сравнивает в режиме прерывания накопленное значение высокоскоростного счетчика со значениями, указанными в табличных областях.
- Операнд (D) для этой специальной функции задается по специальному реле M1150.
- 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).

Всегда может быть применена только одна DHSZ-инструкция специального реле M1150.

DHSZ-инструкция со специальным реле M1150 первый раз выполняется по первой END-инструкции. ПЛК 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 реле M1150

После срабатывания входа X10 выполняется сравнение таблицы от регистра данных D0 в 4-х записях таблицы данных со значением в счетчике C251.

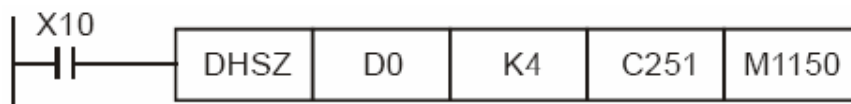
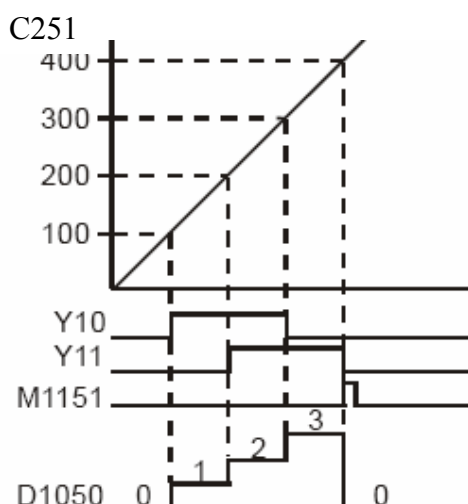


Таблица имеет следующую форму:

Данные сравнения (32 бит)		Адрес выхода Y	Индикация вкл/выкл.		Счетчик номера записи D1150
Старшее слово	Младшее слово				
D1 (K0)	D0 (K100)	D2 (K10)	D3	(K1)	0
D5 (K0)	D4 (K200)	D6 (K11)	D7	(K1)	1
D9 (K0)	D8 (K300)	D10 (K10)	D11	(K0)	2
D13 (K0)	D12 (K400)	D14 (K11)	D15	(K0)	3
		K10: Y10 K11: Y11	K1 = включение, K0 = отключение		0→1→2→3→0 цикл сканирования

Таблица сравнения, стартовый адрес D0, длина K4. Значения сравнения запоминаются в 32-х битном формате (двойное слово).

Временные процессы для включения выходов Y10, Y11



Если DHSZ-инструкция применяется со специальным реле M1150, то регистр данных D1150 определяется как счетчик номера записи. После каждого сравнения D1150 переходит к следующему номеру записи.

Если все записи таблицы обработаны, то включается флаг конца процесса M1151 и D1150 выключается внешним импульсом или импульсом из программы. D1150 стартует вновь согласно счету, если отключается флаг M1151.

Принцип функционирования DHSZ-инструкции в режиме управления частотой (со специальным реле M1152)

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 реле M1152 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 реле M1152.
- 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).

Всегда может быть применена только одна DHSZ-инструкция специального реле M1152.

DHSZ-инструкция со специальным реле M1152 первый раз выполняется по первой END-инструкции. ПЛК 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 D1151 не перескочит к началу таблицы.

Пример: Применение DHSZ-инструкции со специальным реле M1152

После срабатывания входа X10 выполняется сравнение таблицы от регистра данных D0 в 5-ти записях таблицы данных со значением в счетчике C251.

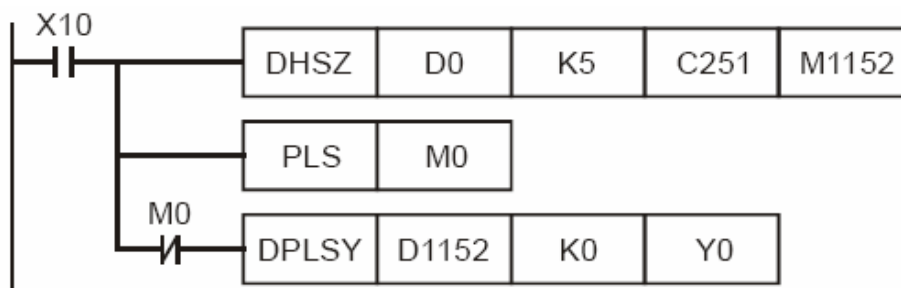
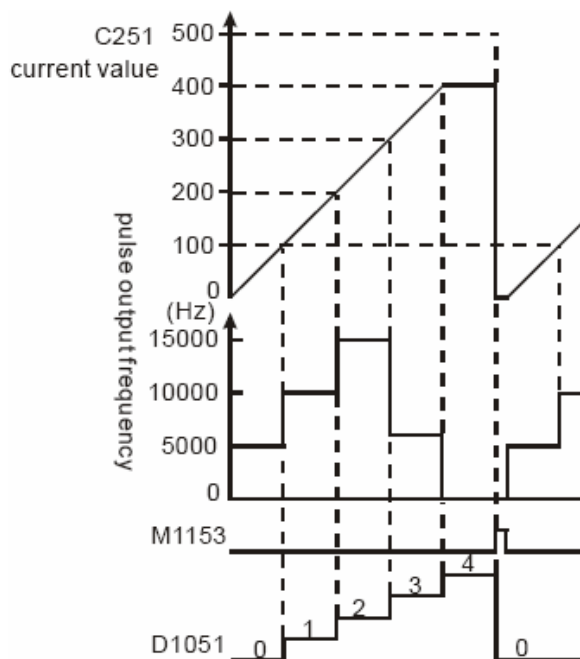


Таблица имеет следующую форму:

Данные сравнения (32 бит)		Частота на импульсном выходе 0...250 кГц	Счетчик номера записи D1151
Старшее слово	Младшее слово		
D1 (K0)	D0 (K100)	D3, D2 (K5000)	0
D5 (K0)	D4 (K200)	D7, D6 (K10000)	1
D9 (K0)	D8 (K300)	D11 D10 (K15000)	2
D13 (K0)	D12 (K400)	D15, D14 (K6000)	3
D17 (K0)	D16 (K0)	D17, D16 (K0)	4
			0→1→2→3→4 цикл сканирования

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.

Изменение частоты при применении DHSZ-инструкции со специальным реле M1152:



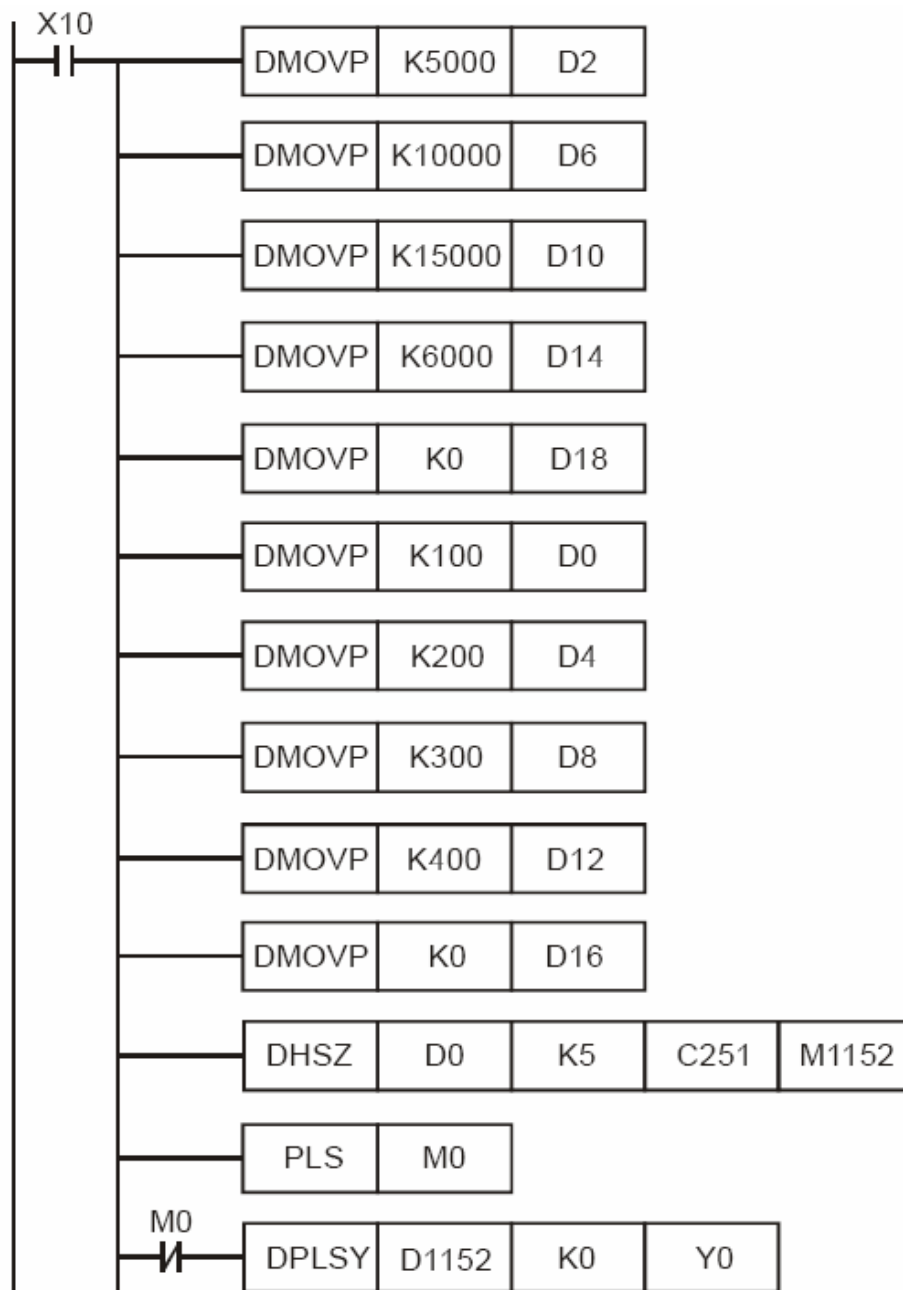
Если DHSZ-инструкция применяется со специальным реле M1152, то регистр данных D1151 определяется как счетчик номера записи. После каждого сравнения D1151 переходит к следующему номеру записи.

D1152 содержит соответствующее значение частоты записей таблицы для применения в PLSY-инструкции. Для обработки регистры данных D1336 и D1337 содержат сравниваемые значения.

Если все записи таблицы обработаны, то включается флаг конца процесса M1153 и D1151 выключается внешним импульсом или импульсом из программы. D1151 стартует вновь согласно счету, если отключается флаг M1153.

При отключении DHSZ-инструкции отключаются все значения, включая выдаваемую частоту.

Пример полной программы:



API	SPD	S₁ S₂ D	Определение скорости	DVP-		
				ES/EX/SS	SA/SX	EH
56				+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (7 шагов) SPD - Непрерывное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S ₁	*																
S ₂					*	*	*	*	*	*	*	*	*	*	*		
D											*	*	*				
<u>Примечания:</u> Операнд S ₁ может быть X1-X2 (в ES/EX/SS/SA/SX) Операнд S ₁ может быть X0-X3 (в EH) Операнд D: занимает 5 адресов																32-х битная инструкция --- Флаги: M1100	

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+3), (D+4).
 (D+1), (D): сумма импульсов после отсчета времени
 (D+3), (D+2): текущее значение импульсов внутри интервала времени
 (D+4): остающееся отсчитываемое время (32767 мс макс.)

После отсчета времени содержание (D+3), (D+2) передается в (D+1), (D), а само (D+3), (D+2) отключается.

Входы высокоскоростного счетчика, используемые в инструкции, не могут применяться в других высокоскоростных операциях.

Для каждого высокоскоростного входа можно задать максимум одну SPD-инструкцию.

Максимальная измеряемая частота импульсов:

Модель	DVP-ES/EX/SS/SA/SX	DVP-EH
Макс. частота	X1 (30кГц); X2 (10кГц)	X0/X1 (250кГц); X2/X3 (10кГц)

По следующей формуле можно рассчитать скорость вращения в об/мин:

$$N = \frac{60(D0)}{nt} \times 10^3, \text{ где}$$

N – частота вращения (об/мин)

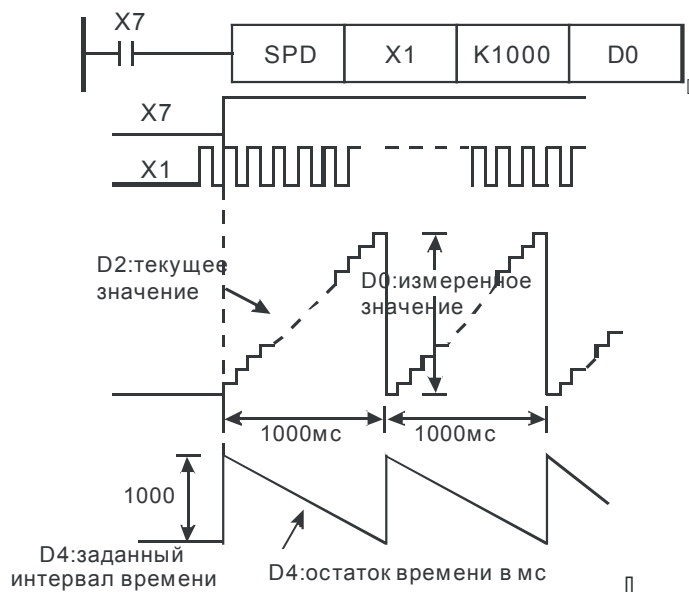
n – число импульсов на оборот

t – время выборки S2 (мс)

Флаг M1100 будет включаться каждый раз по истечении времени выборки S2.

Пример программирования SPD-инструкции

Когда X7 включено D2 считает количество включений X0. После 1000 мс результат счета сохраняется в D0. D2 обнуляется и начинает вновь счет включений X0. В D4 соответственно измеряется оставшееся время.



API		PLSY	S1	S2	D	Импульсный выход	DVP-		
57	D						ES/EX/SS	SA/SX	EH
							+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов) PLSY - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S ₁					*	*	*	*	*	*	*	*	*	*	*
S ₂					*	*	*	*	*	*	*	*	*	*	*
D		*													
<u>Примечания:</u> В серии ES/SS/EX команда PLSY может в программе использоваться только дважды и выход не должен повторяться.															32-х битная инструкция (13 шагов) DPLSY - Непрерывное выполнение.
															Флаги: M1010 – M1345

Функция

Выдача на выходе определенного числа импульсов с жестко заданной частотой и жестким соотношением ширины импульса 1:1.

Описание

- Инструкция формирует определенное число импульсов.
- В (S1) определяется частота: 1 ... 10000 Гц (в DVP-ES/EX/SS);
1... 33000 Гц (в DVP-SA/SX);
1... 200000 Гц (в DVP-EH)
- 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 неограниченный ряд импульсов (в DVP-EH). В DVP-ES/EX/SS/SA/SX для выдачи неограниченного числа импульсов должны быть включены специальные реле M1010 (Y0), M1023 (Y1)

- В (D) определяется адрес выхода. В DVP-ES/EX/SS/SA/SX может быть только Y0 или Y1; в DVP-EH может быть только Y0 или Y2. ПЛК должен иметь транзисторные выходы.
- Соотношение между включенным и отключенным состоянием: 50% ВКЛ, 50% ОТКЛ. Включенное и отключенное состояния выдаются непосредственно в режиме прерывания.
- При применении PLSY-инструкции число импульсов задается в двух следующих друг за другом регистрах данных.
- В DVP-ES/EX/SS/SA/SX если желаемое число импульсов создано на выходе Y0, то включается флаг M1029 (инструкция полностью отработана на выходе Y0) и если желаемое число импульсов создано на выходе Y1, то включается флаг M1030 (инструкция полностью отработана на выходе Y1). M1029 и M1029 отключатся, если входное условие для PLSY-инструкции будет отключено.
- В DVP-EH если желаемое число импульсов создано на выходах Y0 и Y1, то включается флаг M1029 (инструкция полностью отработана на выходах Y0 и Y1) и если желаемое число импульсов создано на выходах Y2 и Y3, то включается флаг M1030 (инструкция полностью отработана на выходах Y2 и Y3). M1029 и M1029 отключатся, если входное условие для PLSY-инструкции будет отключено.

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.

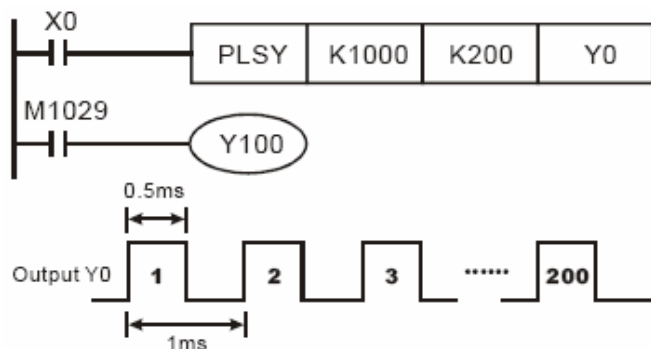
В специальных регистрах D1336-D1339 будет отображаться текущее выданное число импульсов во время выполнения PLSY-инструкции.

Возможно применение одновременно инструкций PLSY, PWM и PLSR, если данные инструкций подключены к разным выходам.

Возможно использование в подпрограммах нескольких инструкций PLSY, тем не менее должна остановиться запущенная команда, прежде чем в подпрограмме начинается следующая команда PLSY.

Пример программирования PLSY-инструкции

Если включено X0, то создается 200 импульсов с частотой 1000 Гц на выходе Y0. Создание импульсов прекращается, если выключается X0. Если X0 включается снова, то операция начинается вновь. Если X0 не включено, то выключается Y0. После создания последнего импульса флаг M1029 включит выход Y100.



Специальные реле

M1010	ES/EX/SS/SA: выбор режима PLSY импульсного выхода Y0. Когда M1010=1 импульсы идут непрерывно. Когда M1010=0 выдается число импульсов заданное в S2. EH: M1010=1 импульсы на Y0, Y1, Y2, Y3 будут идти до выполнения инструкции END. Когда начнется выдача импульсов, реле M1010 буде сброшено.
M1023	ES/EX/SS/SA: Выбор режима PLSY импульсного выхода Y1. Когда M1023=1 импульсы идут непрерывно.
M1029	ES/EX/SS/SA: Команда PLSY или PLSR для импульсного выхода Y0 полностью выполнена. EH: Первая импульсная группа CH0 (Y0, Y1) полностью выполнена.
M1030	ES/EX/SS/SA: Команда PLSY или PLSR для импульсного выхода Y1 полностью выполнена. EH: Вторая импульсная группа CH1 (Y2, Y3) полностью выполнена.
M1078	ES/EX/SS/SA: Остановка выполнения команды PLSY для Y0
M1079	ES/EX/SS/SA: Остановка выполнения команды PLSY для Y1
M1258	EH: Замена импульсного выходного сигнала на Y0 и Y1 (для PWM-инструкции)
M1259	EH: Замена импульсного выходного сигнала на Y2 и Y3 (для PWM-инструкции)
M1334	EH: Стоп импульсного выхода CH0 (Y0, Y1)
M1335	EH: Стоп импульсного выхода CH1 (Y2, Y3)
M1336	EH: Флаг передачи импульсов CH0 (Y0, Y1)
M1337	EH: Флаг передачи импульсов CH1 (Y2, Y3)
M1338	EH: Старт импульсного выхода CH0 (Y0, Y1) флаг начала
M1339	EH: Старт импульсного выхода CH1 (Y2, Y3) флаг начала
M1340	EH: Иметь прерывание (I110) после завершения передачи по CH0 (Y0, Y1)
M1341	EH: Иметь прерывание (I120) после завершения передачи по CH1 (Y2, Y3)
M1342	EH: Иметь прерывание (I130) при одновременной передаче по CH0 (Y0, Y1)
M1343	EH: Иметь прерывание (I140) при одновременной передаче по CH1 (Y2, Y3)
M1344	EH: Старт CH0 (Y0, Y1) флаг выравнивания
M1345	EH: Старт CH1 (Y2, Y3) флаг выравнивания
M1347	EH: Автоматический сброс флага прерывания для CH0 (Y0, Y1)
M1348	EH: Автоматический сброс флага прерывания для CH1 (Y2, Y3)

Специальные регистры

D1030	ES/EX/SS/SA: Выданное количество импульсов на выходе Y0 (младшее слово)
D1031	ES/EX/SS/SA: Выданное количество импульсов на выходе Y0 (старшее слово)
D1032	ES/EX/SS/SA: Выданное количество импульсов на выходе Y1 (младшее слово)
D1033	ES/EX/SS/SA: Выданное количество импульсов на выходе Y1 (старшее слово)
D1220	EH: Установка первой группы импульсных выходов: 00: 1 фазный выход (Y0); 01: 2-х фазный выход AB на Y0, Y1 (A опережает B); 02: 2-х фазный выход AB на Y0, Y1 (B опережает A); 03: 1 фазный выход (Y1);
D1221	EH: Установка второй группы импульсных выходов: 00: 1 фазный выход (Y2); 01: 2-х фазный выход AB на Y2, Y3 (A опережает B); 02: 2-х фазный выход AB на Y2, Y3 (B опережает A); 03: 1 фазный выход (Y3);
D1328	EH: CH0 (Y0, Y1) смещение импульсов (младшее слово)
D1329	EH: CH0 (Y0, Y1) смещение импульсов (старшее слово)
D1330	EH: CH1 (Y2, Y3) смещение импульсов (младшее слово)
D1331	EH: CH1 (Y2, Y3) смещение импульсов (старшее слово)
D1332	EH: CH0 (Y0, Y1) остаток импульсов (младшее слово)
D1333	EH: CH0 (Y0, Y1) остаток импульсов (старшее слово)
D1334	EH: CH1 (Y2, Y3) остаток импульсов (младшее слово)
D1335	EH: CH1 (Y2, Y3) остаток импульсов (старшее слово)

D1336	EH: CH0 (Y0, Y1) текущее значение импульсов (младшее слово)
D1337	EH: CH0 (Y0, Y1) текущее значение импульсов (старшее слово)
D1338	EH: CH1 (Y2, Y3) текущее значение импульсов (младшее слово)
D1339	EH: CH1 (Y2, Y3) текущее значение импульсов (старшее слово)
D1344	EH: CH0 (Y0, Y1) число импульсов дополнения (младшее слово)
D1345	EH: CH0 (Y0, Y1) число импульсов дополнения (старшее слово)
D1346	EH: CH1 (Y2, Y3) число импульсов дополнения (младшее слово)
D1347	EH: CH1 (Y2, Y3) число импульсов дополнения (старшее слово)

API		PWM	  	Импульсный выход с ШИМ	DVP-		
					ES/EX/SS	SA/SX	EH
58					+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов) PWM - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
D		*														
<u>Примечания:</u> В серии ES/SS/EX команда PLSY может в программе использоваться только один раз. Значение операнда S1 должно быть меньше или равно значения операнда S2																32-х битная инструкция ---. Флаги: M1010 – M1337

Функция

Выдача на выходе импульсов с жестко заданной шириной и периодом.

Описание

- Инструкция формирует последовательный ряд импульсов. Соотношение между шириной импульса t и длительностью периода T задается:

t : ширина импульса (мс)

T : продолжительность периода (мс)

Частота f : $1/T$ (кГц)

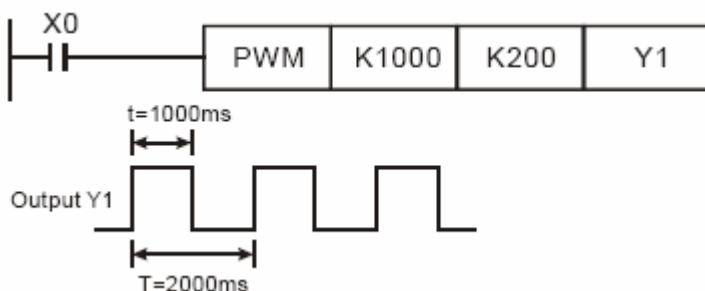
- В (S1) устанавливается ширина импульса в области t : от 1 до 32767 мс. Ширина импульса должна лежать в пределах $1 < t < T$.
- В (S2) устанавливается длительность периода T : от 1 до 32767 мс.
- В (D) указывается адрес выхода. В DVP-ES/EX/SS/SA/SX может быть только Y1; в DVP-EH может быть только Y0 или Y2. ПЛК должен иметь транзисторные выходы, чтобы избежать быстрого износа контактов.
- Возможно применение одновременно инструкций PLSY, PWM и PLSR, если данные инструкций подключены к разным выходам.
- S1 и S2 могут быть изменены во время выполнения PWM-инструкции.

Источники ошибок:

Если установленное значение ширины импульса в (S1) больше, чем жестко определенное значение для T в (S2), то появляется ошибка в работе программы M1067=1.

Пример программирования PWM-инструкции

Если включено X0, то на выходе Y1 формируются импульсы с шириной 1с и периодом 2с. Выдача импульсов прекращается, если выключается X0. Если X0 не включено, то выключается Y1.



Специальные реле

M1010	ЕН: M1010=1 импульсы на Y0, Y1, Y2, Y3 будут идти до выполнения инструкции END. Когда начнется выдача импульсов, реле M1010 буде сброшено.
M1070	ES/EX/SS/SA: Команда PWM для выхода Y0. M1070=0: дискретность задания = 1 мс M1070=1: дискретность задания = 100 мкс. ЕН: Команда PWM для выхода Y0. M1070=0: дискретность задания = 1 мс M1070=1: дискретность задания = 100 мкс.
M1071	ЕН: Команда PWM для выхода Y2. M1070=0: дискретность задания = 1 мс M1070=1: дискретность задания = 100 мкс.
M1258	ЕН: Замена импульсного выходного сигнала на Y0 и Y1 (для PWM-инструкции)
M1259	ЕН: Замена импульсного выходного сигнала на Y2 и Y3 (для PWM-инструкции)
M1334	ЕН: Стоп импульсного выхода CH0 (Y0, Y1)
M1335	ЕН: Стоп импульсного выхода CH1 (Y2, Y3)
M1336	ЕН: Флаг передачи импульсов CH0 (Y0, Y1)
M1337	ЕН: Флаг передачи импульсов CH1 (Y2, Y3)

API		PLSR		S₁ S₂ S₃ D	Импульсный выход с ускорением/замедлением	DVP-		
59	D					ES/EX/SS	SA/SX	EH
						+	+	+

Оп- код	Биты				Слова											16-ти битная инструкция (9 шагов) PLSR - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
S ₃					*	*	*	*	*	*	*	*	*	*	*	
D		*														
<u>Примечания:</u> В серии ES/SS/EX команда PLSY может в программе использоваться только дважды и выход не должен повторяться.																
32-х битная инструкция (17 шагов) DPLSR - Непрерывное выполнение. Флаги: M1029 – M1030																

Функция

Выдача на выходе определенного числа импульсов с заданной частотой и участками плавного увеличения и уменьшения частоты.

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).

- В (S1) определяется частота: 10 ... 10000 Гц (в DVP-ES/EX/SS);
10... 32000 Гц (в DVP-SA/SX);
10... 200000 Гц (в DVP-EH)

Указанная частота должна делиться на 10. Если задаваемая частота не делится на 10, она округляется до соответствующего значения.

- Ширина шага наклонной составляет 1/10 указанной выходной частоты (при применении шагового двигателя это нужно учитывать).

- Максимальное количество импульсов:

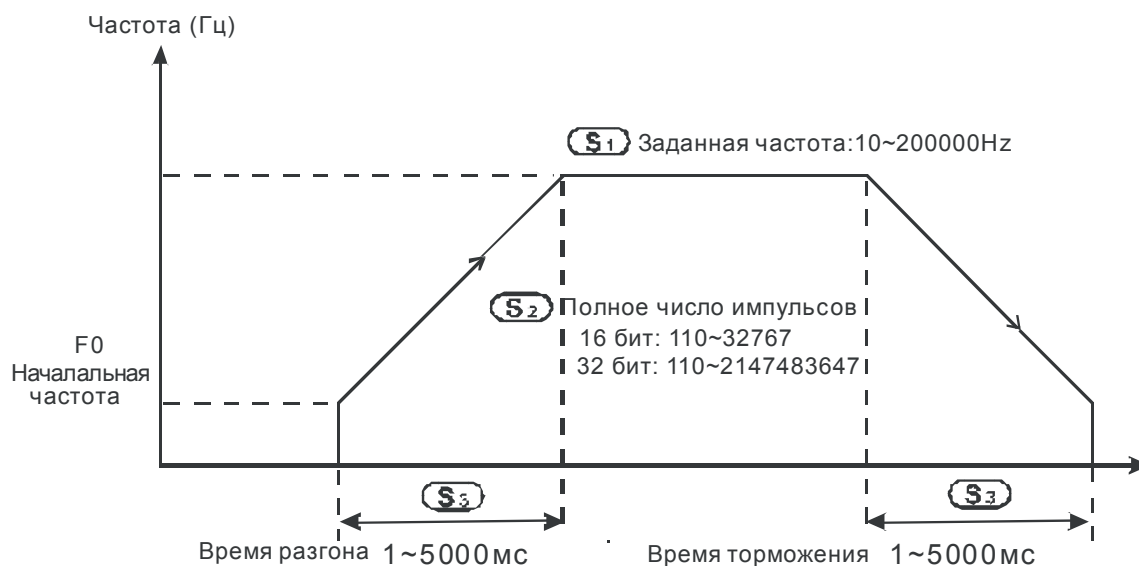
16-ти битовые инструкции: от 110 до 32767 импульсов

32-х битных инструкций: от 110 до 2147483647 импульсов

При задании менее 110 импульсов правильная работа не гарантируется.

- Время подъема ramпы должно соответствовать ниже описанным граничным значениям.

- В (D) определяется адрес выхода. В DVP-ES/EX/SS/SA/SX может быть только Y0 или Y1; в DVP-EH может быть только Y0 или Y2. ПЛК должен иметь транзисторные выходы.



В одной программе в DVP-ES/SS/SX 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.

- В DVP-ES/EX/SS/SA/SX если желаемое число импульсов создано на выходе Y0, то включается флаг M1029 (инструкция полностью отработана на выходе Y0) и если желаемое число импульсов создано на выходе Y1, то включается флаг M1030 (инструкция

полностью отработана на выходе Y1). M1029 и M1029 отключатся, если входное условие для PLSR-инструкции будет отключено.

- В DVP-EH если желаемое число импульсов создано на выходах Y0 и Y1, то включается флаг M1029 (инструкция полностью отработана на выходах Y0 и Y1) и если желаемое число импульсов создано на выходах Y2 и Y3, то включается флаг M1030 (инструкция полностью отработана на выходах Y2 и Y3). M1029 и M1029 отключатся, если входное условие для PLSR-инструкции будет отключено.

Ограничение времени наклона

Время наклона (S3) ограничивается 5000 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 (D1012).

- Минимальное значение для (S3) рассчитывается по уравнению:

$$(S3) \geq \frac{90000}{(S1)}$$

- Максимальное значение для (S3+) рассчитывается по уравнению:

$$(S3) \leq \frac{(S2)}{(S1)} \times 818$$

- Если параметр выходит за расчетные границы, то значение (S1) (частота) уменьшается.

- Подъем выходной частоты происходит в 10 шагов.

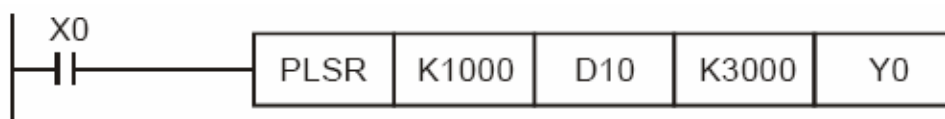
После отключения условий выполнения PLSR-инструкции выключаются сработавшие выходы. При новом включении условий выполнения снова начинается обработка инструкции.

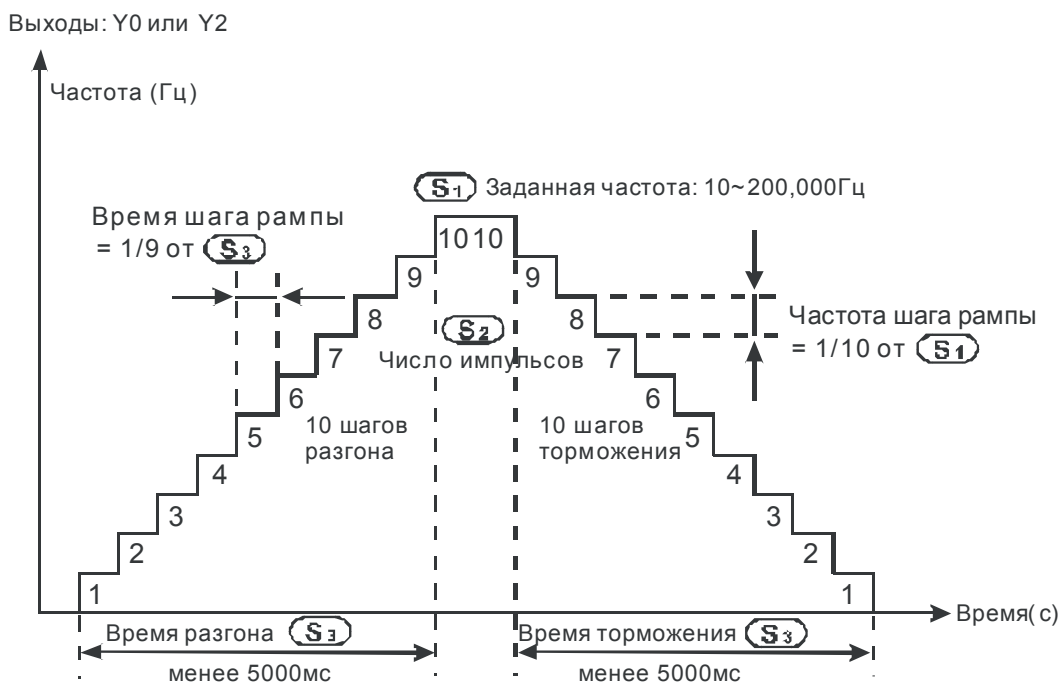
Если во время обработки изменяется операнд, профиль (характер) выхода сохраняется. Изменения операнда начинают действовать при следующей обработке инструкции.

Пример программирования PLSR-инструкции

При включении входа X0 число импульсов, указанное в D10 (S2) выдается на Y (D). Выходная частота составляет 1000 Гц (S1).

Возрастание частоты до 1000 Гц (S1) и снижение частоты до 0 выполняется соответственно за 3000 мс (S3) шагами по 100 Гц (S1 / 10).





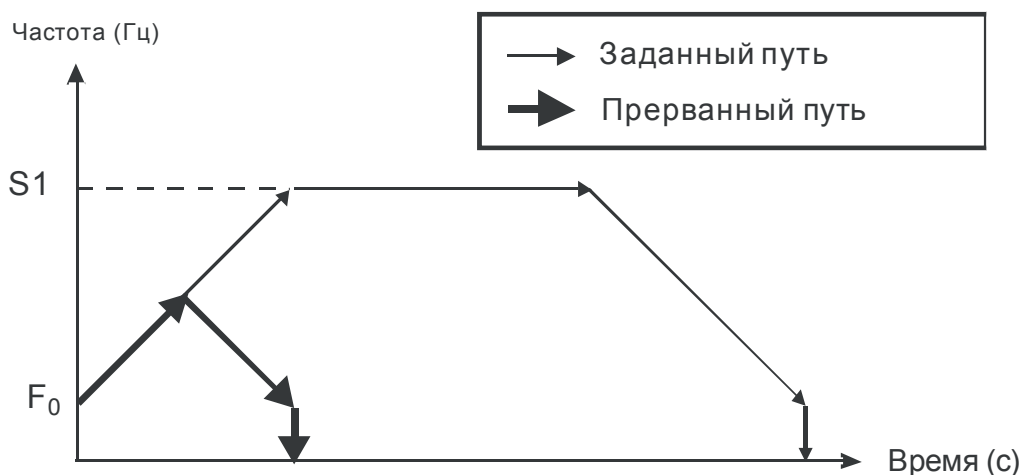
Специальные реле

M1010	EH: M1010=1 импульсы на Y0, Y1, Y2, Y3 будут идти до выполнения инструкции END. Когда начнется выдача импульсов, реле M1010 буде сброшено.
M1029	ES/EX/SS/SA: Команда PLSY или PLSR для импульсного выхода Y0 полностью выполнена. EH: Первая импульсная группа CH0 (Y0, Y1) полностью выполнена.
M1030	ES/EX/SS/SA: Команда PLSY или PLSR для импульсного выхода Y1 полностью выполнена. EH: Вторая импульсная группа CH1 (Y2, Y3) полностью выполнена.
M1334	EH: Стоп импульсного выхода CH0 (Y0, Y1)
M1335	EH: Стоп импульсного выхода CH1 (Y2, Y3)
M1336	EH: Флаг передачи импульсов CH0 (Y0, Y1)
M1337	EH: Флаг передачи импульсов CH1 (Y2, Y3)

Специальные регистры

D1030	ES/EX/SS/SA: Выданное количество импульсов на выходе Y0 (младшее слово)
D1031	ES/EX/SS/SA: Выданное количество импульсов на выходе Y0 (старшее слово)
D1032	ES/EX/SS/SA: Выданное количество импульсов на выходе Y1 (младшее слово)
D1033	ES/EX/SS/SA: Выданное количество импульсов на выходе Y1 (старшее слово)
D1220	EH: Установка первой группы импульсных выходов: 00: 1 фазный выход (Y0); 01: 2-х фазный выход AB на Y0, Y1 (A опережает B); 02: 2-х фазный выход AB на Y0, Y1 (B опережает A); 03: 1 фазный выход (Y1);
D1221	EH: Установка второй группы импульсных выходов: 00: 1 фазный выход (Y2); 01: 2-х фазный выход AB на Y2, Y3 (A опережает B); 02: 2-х фазный выход AB на Y2, Y3 (B опережает A); 03: 1 фазный выход (Y3);
D1336	EH: CH0 (Y0, Y1) текущее значение импульсов (младшее слово)
D1337	EH: CH0 (Y0, Y1) текущее значение импульсов (старшее слово)
D1338	EH: CH1 (Y2, Y3) текущее значение импульсов (младшее слово)
D1339	EH: CH1 (Y2, Y3) текущее значение импульсов (старшее слово)
D1340	Темп разгона 1
D1341	Максимальная выходная частота (младшее слово)
D1342	Максимальная выходная частота (старшее слово)

Если выполнение инструкции PLSR будет внезапно прервано, то импульсы на выходе прекратятся не сразу, а будет выполнено автоматическое торможение:



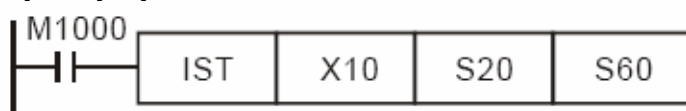
API	IST	S	D1	D2	Ручное/автоматическое управление	ПЛК		
60						ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (7 шагов). IST Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*													
D1				*												
D2				*												
<u>Примечания:</u> Операнд S может занимать 8 устройств. Используемый диапазон операндов D1 и D2: S20 – S899 и D2 > D1. Команда IST в программе может использоваться только один раз.																32-х битная инструкция --- Флаги: M1040 – M1047

Описание:

- 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).
- В (S) определяется область управляющих входов. В качестве управляющих входов могут применяться операнды X, Y или M. Они указывают стартовые адреса областей операндов. В (D1) и (D2) определяется область операндов шаговых состояний для шаговых цепей автоматического режима работы. При этом должно выполняться условие: (D1) < (D2).

Пример применения:



- Операнд (S) определяет управляющие входы, которые служат для непосредственного управления процессом работы. Ими выбирается нужный режим работы:

X10: ручной (наладочный) режим работы

X11: возврат в исходную позицию

X12: шаговый режим работы

X13: выполнение одного цикла

X14: автоматический режим работы

X15: кнопка возврата в исходную позицию

X16: кнопка запуска автоматического режима работы

X17: кнопка останова автоматического режима работы

- Следующие специальные реле благодаря IST-инструкции влияют или управляют обработкой шаговых цепей:

M1040: запрещение переходов

M1041: старт последующего перехода

M1042: импульс запуска

M1043: Возвращение в нулевую точку завершено

M1044: Нахождение в нулевой точке

M1045: Запрещение сброса всех выходов

M1046: Режим STL выполняется

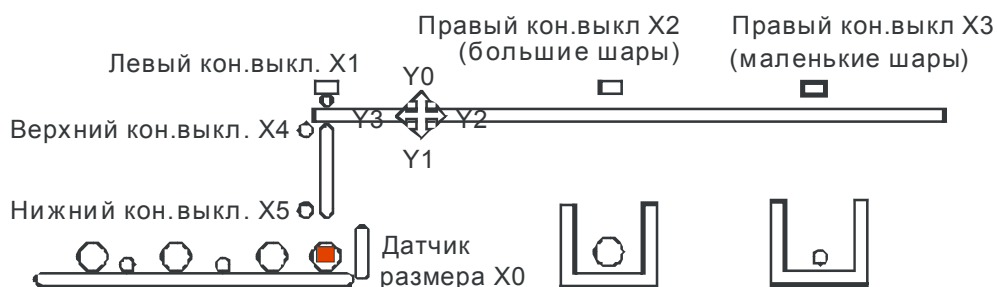
M1047: разрешение отображение состояния шагов в регистрах D1040 – D1047

- Операнды шаговых состояний S0...S2 предназначены (резервируются) для инициализации шаговых цепей:
 - S0: наладочный режим работы,
 - S1: перемещение в нулевую точку
 - S2: автоматический режим работы
- 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).

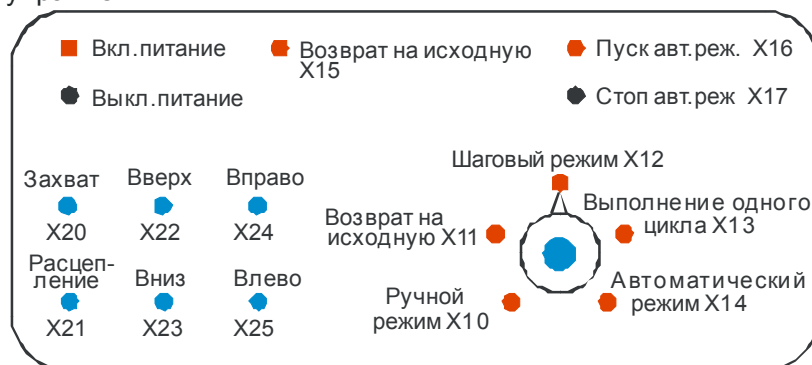
Пример управления роботом-манипулятором (с помощью IST-инструкции):

Задача: Надо произвести сортировку больших и маленьких шаров и поместить их в соответствующие коробки.

Движения робота-манипулятора: опустить манипулятор вниз, взять шар, поднять манипулятор с шаром, переместить манипулятор с шаром вправо, опустить манипулятор с шаром вниз, положить шар в коробку, поднять манипулятор, переместить манипулятор влево на исходную позицию.



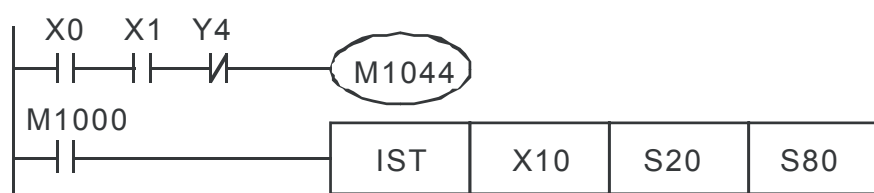
Пульт управления:



Описание датчиков и исполнительных устройств:

- Конечные выключатели: слева – X1; справа – X2 (для больших шаров) и X3 (для маленьких шаров); снизу – X5; сверху – X4
- X0 – датчик определения размера шара
- X10 – X14: переключатели выбора режима работы
- X15 – X17: кнопки пуска/стопа выполнения выбранного режима
- X20 – X25: кнопки управления манипулятором в ручном режиме
- Команды перемещения манипулятора: Y0 – вверх; Y1 – вниз; Y2 – вправо; Y3 – влево

Инициализация:



Ручной режим работы:**Режим выхода на исходную позицию:**

SFC-диаграмма:

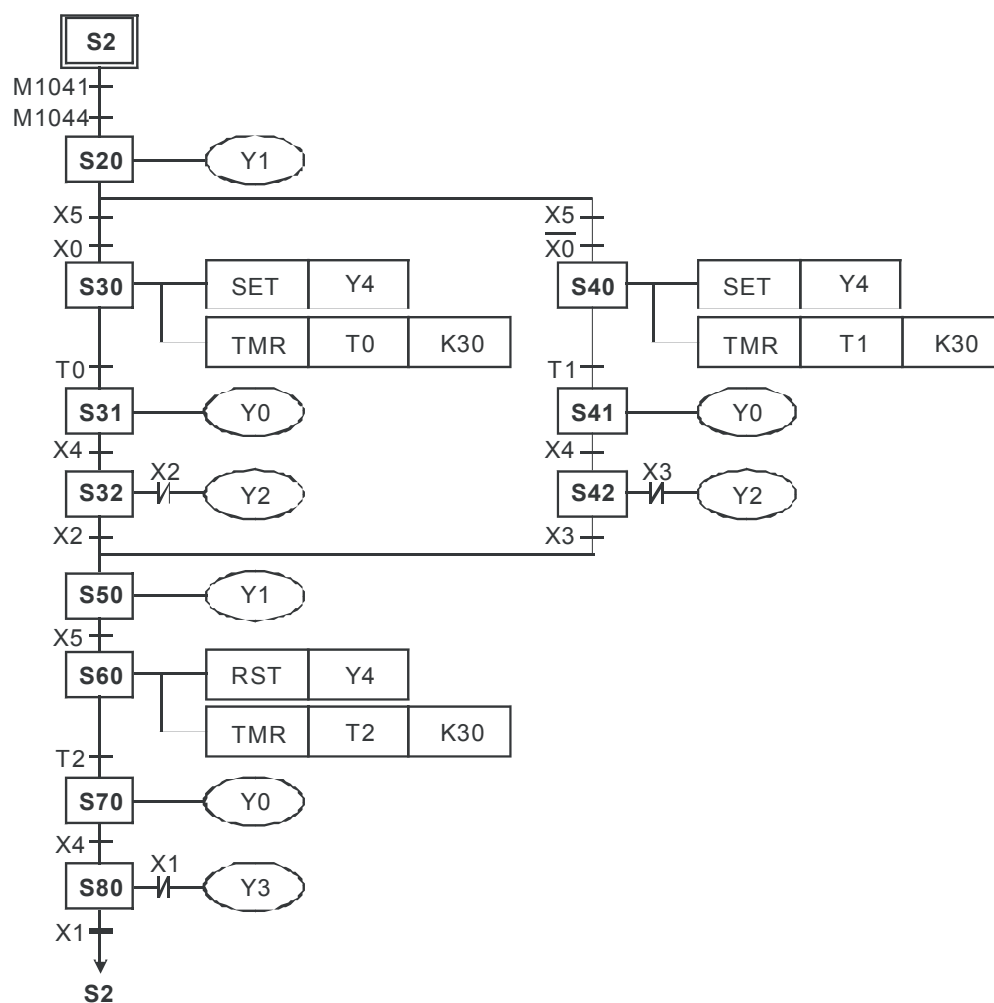


Релейно-контактная схема:

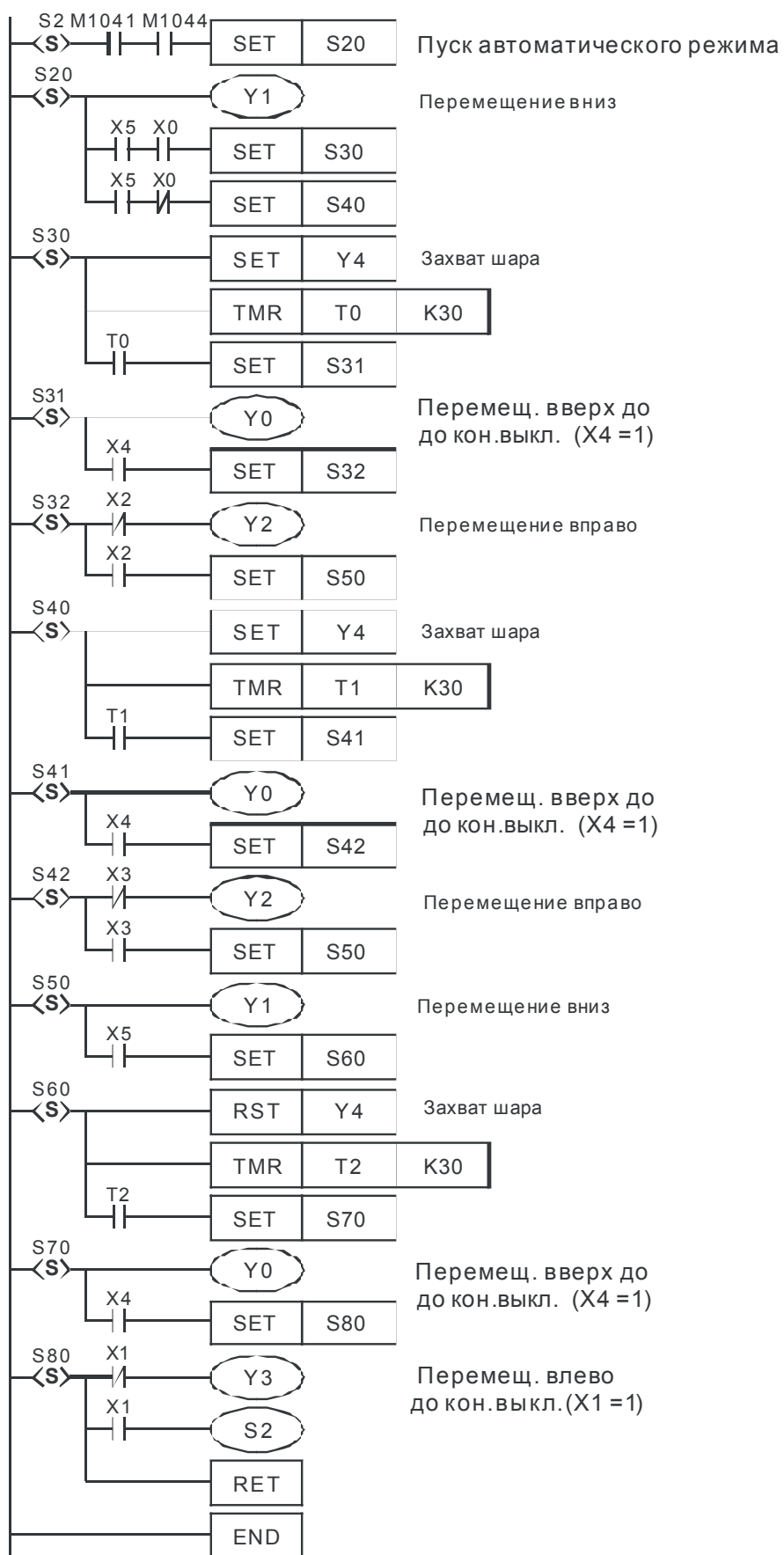


Автоматический режим (пошаговый/ один цикл/ непрерывный)

SFC-диаграмма:



Релейно-контактная схема:



API		SER		S₁ S₂ D n	Поиск данных в стеке	DVP-		
61	D		P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- еранд	Биты				Слова										<u>16-ти битная инструкция</u> (9 шагов) SER - Непрерывное выполнение. <u>32-х битная инструкция</u> (17 шагов) DSER - Непрерывное выполнение. Флаги: нет	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁							*	*	*	*	*	*	*			
S ₂					*	*	*	*	*	*	*	*	*	*		*
D								*	*	*	*	*	*			
n					*	*							*			
<u>Примечания:</u> Если операнд S ₂ используется с индексом F, возможно только 16-ти битное выполнение команды. Операнд D занимает 5 адресов n=1...256 (16 бит команда), n=1...128 (32 бит команда)																

Функция

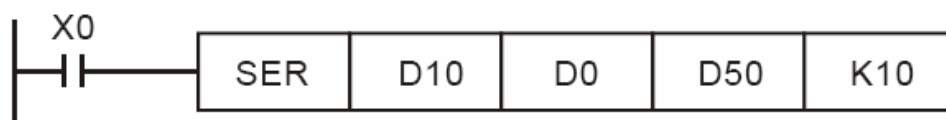
Сквозной поиск в области данных по значению поиска

Описание

- Область данных от (S₁) до (S₁+n) просматривается 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

В примере показан поиск по коду в D0 = K100, начиная с D10 при длине K10. Результат заносится в D50 до D54 (5 регистров при 16-ти битовом формате).



Область данных (от (S ₁) до (S ₁ +n))	Сравниваемое значение (ключ поиска S ₂)	Позиция	Результат сравнения
D10 = 88	D0 = 100	0	равенство
D11 = 100		1	
D12 = 110		2	

D13 = 150		3	
D14 = 100		4	равенство
D15 = 300		5	
D16 = 100		6	равенство
D17 = 5		7	минимум
D18 = 100		8	равенство
D19 = 500		9	максимум

Регистры результата (D)	Описание
D50 = 4	Полное количество эквивалентных данных
D51 = 1	Номер первого эквивалентного значения
D52 = 8	Номер последнего эквивалентного значения
D53 = 7	Номер наименьшего значения
D54 = 9	Номер наибольшего значения

API		ABSD		S₁ S₂ D n	Абсолютный многоуставочный счетчик	DVP-		
62	D					ES/EX/SS	SA/SX	EH
						-	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (9 шагов) ABSD - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S ₁							*	*	*	*	*	*	*			32-х битная инструкция (17 шагов) DABSD - Непрерывное выполнение.
S ₂												*				
D		*	*	*												
n					*	*										
<u>Примечания:</u> Если операнд S ₁ определен как KnX, KnY, KnM, KnS, то если Kn=K4: 16-ти битное выполнение команды; Kn=K8: 32-х битное выполнение команды. Диапазон n=1...64																
Флаги: нет																

Функция

Включение выходного бита в зависимости от состояния счетчика

Описание

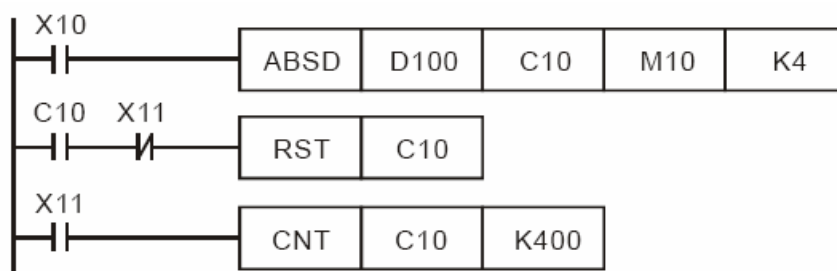
- Накопленное (текущее) значение счетчика (S₂) сравнивается с таблицей значений входов/выходов. Таблица образуется из словных операндов. Первым операндом является (S₁). Таблица содержит (n) строк.
- Если счетчик (S₂) получил накопленное значение, занесенное в таблицу, включается относящийся к нему операнд.
- Записанное в таблицу значение должно находиться в области счета от 0 до 32 767.
- Значение может, например, быть записанным с помощью MOV-инструкции.

Всегда записываются два счетчика, (S₂) и (S₂+1). В (S₁) должен применяться четный адрес операнда.

Допустимо использовать в (S₂) высокоскоростные счетчики для DABSD-инструкции.

Пример программирования ABSD-инструкции

В примере программирования на рисунке контролируется состояние ВКЛ./ОТКЛ. реле от M10 до M13.



В (S2) указывается счетчик (C10). Число (n) определяет количество включаемых и отключаемых операндов цели (D) и соответственно число операндов (S1), в которых записаны значения включения и отключения.

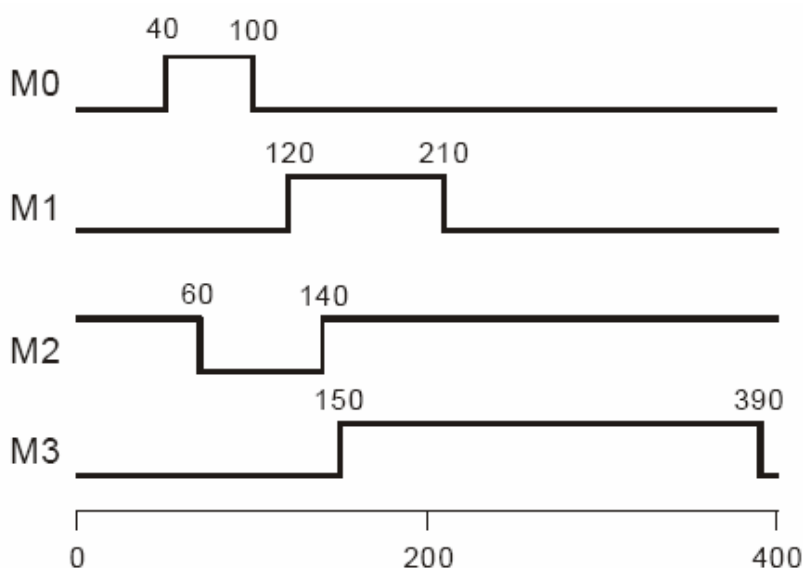
Так как $n = 4$, то для процессов включения и отключения имеются реле M10...M3.

Четыре включающих значения записываются в регистры данных D100, D102, D104 и D106. Четыре отключающих значения записываются в регистры данных D101, D103, D105 и D107.

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.

Нижний лимит	Верхний лимит	Текущее значение счетчика C10	Состояние выхода
D100 = 40	D101 = 100	$40 \leq C10 \leq 100$	M10 = 1
D102 = 120	D103 = 210	$120 \leq C10 \leq 210$	M11 = 1
D104 = 140	D105 = 170	$140 \leq C10 \leq 170$	M12 = 1
D106 = 150	D107 = 390	$150 \leq C10 \leq 390$	M13 = 1

Если включено X10, то реле M10...M13 включаются согласно графикам на Рис. Если X10 выключен, реле M10...M13 не включаются.



API		INCD		<div><div>S₁</div><div>S₂</div><div>D</div><div>n</div></div>	Инкрементный многоуставочный счетчик											DVP-		
63																ES/EX/SS	SA/SX	EH
																-	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (9 шагов) INCD - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁							*	*	*	*	*	*	*			
S ₂												*				
D		*	*	*												
n					*	*										

Примечания:

Если операнд S₁ определен как KnX, KnY, KnM, KnS, то Kn=K4.

Операнд S₂ должен быть в диапазоне C0 – C198 и занимает 2 адреса.

Диапазон n=1...64

32-х битная инструкция

Флаги: M1029 (флаг
завершения выполнения)

Функция

Включение выходного бита в зависимости от состояния счетчика

Описание

- n битовые операнды, определяемые в (D), включаются в зависимости от текущего значения счетчиков (S2)
- В (S1) предварительно задается требуемое значение для точки включения битовых операндов (D).
- Счетчик (S2) должен быть в диапазоне C0 – C198; требуемое (задаваемое) значение должно быть больше, чем самое большое значение включения в (S1).
- Счетчик (S2+1) 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 M1029.

Пример программирования INCD-инструкции:

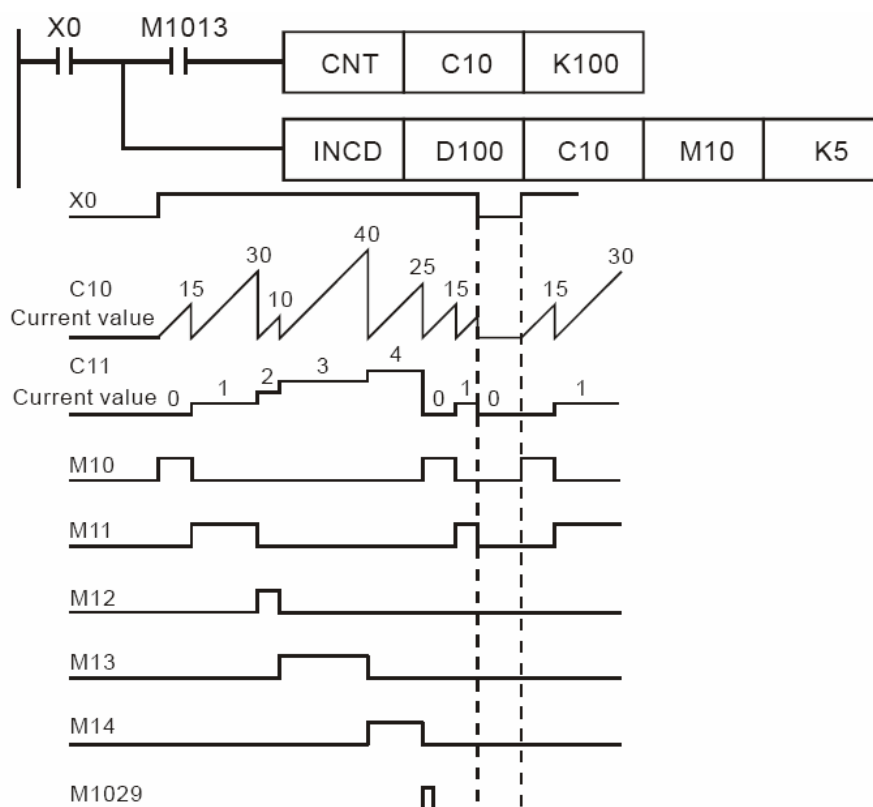
Четыре заданных значения должны быть записаны до начала выполнения инструкции в регистры данных D100-D104 (например командой MOV): D100=15, D101=30, D102=10, D103=40, D104=25,

Счетчик C10 отключается автоматически, если будут достигнуты значения, занесенные в D100... D104.

Счетчик C11 подсчитывает количество циклов отключения на C10. Реле M10...M14 включают соответствующие значения счетчика C11.

Флаг M1029 включается, если выполняется последний, т.е. n-ый цикл счета. В конце действий этот процесс повторяется.

Счетчики C10 и C11 выключаются, если выключается X0; реле M10...M14 также выключаются. Если X0 снова включается, процесс начинается вновь.



API	TTMR	(D) (n)	Обучающийся таймер	DVP-		
				ES/EX/SS	SA/SX	EH
64				-	+	+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (5 шагов) TTMR - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D													*			
n					*	*										
<u>Примечания:</u> Если операнд D будет занимать два адреса. Диапазон n=0...2 В серии SA/SX инструкция может использоваться в программе не более 8 раз. В серии EH инструкция может использоваться в программе не более 8 раз одновременно.																<u>32-х битная инструкция</u> --- Флаги: нет

Функция

Измерение и запоминание времени нажатия кнопки.

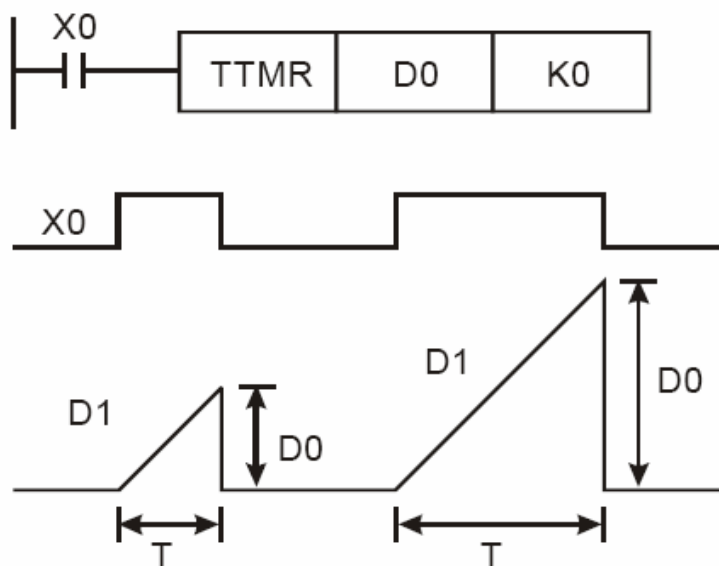
Описание

- Измеряется продолжительность (D+1) включения инструкции (время нажима кнопки), умножается на (n) и записывается в регистр данных (D).
- Число (n) определяет множитель времени.

Если n = 0: (D) измеряется в сек; если n = 1: (D) измеряется с дискретностью 100 мс (множитель 10); если n = 2: (D) измеряется с дискретностью 10 мс (множитель 100).

Указание: С помощью TTMR-инструкции создаются 2 регистра данных (D) и (D+1).

Пример формирования времен:



При включении X0 измеряется продолжительность воздействия и сохраняется в D0 и D1. Когда X0 будет выключено, значение D1 будет сброшено на 0, а D0 сохранит измеренное значение. (D) содержит умноженное время воздействия на кнопку (с). (D+1) содержит время воздействия на кнопку (с).

Отношения между D1 и D0:

n	D0	D1 (ед.: 100 мс)
K0 (ед.: с)	1 x T	D1 = D0 x 10
K1 (ед.: 100 мс)	10 x T	D1 = D0
K2 (ед.: 10 мс)	100 x T	D1 = D0 / 10

API	STMR	S m D	Специальный таймер	DVP-		
65				ES/EX/SS	SA/SX	EH
				-	+	+

Опе- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (7 шагов) STMR - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					
m					*	*										
D		*	*	*												
<u>Примечания:</u> Операнд S серии SA/SX может использовать диапазон T0...T191, а в серии EH - T0...T199. Диапазон m=1...32767 Операнд D занимает 4 адреса.																<u>32-х битная инструкция</u> --- Флаги: нет

Функция

Многофункциональное реле времени, обеспечивающее:

- 1) задержку на выключение;
- 2) формирование импульса по выключению;
- 3) формирование импульса по включению;
- 4) смещение по времени входного сигнала

Описание

- Таймер (S) с уставкой (m) управляет битовым операндом (D) следующим образом:

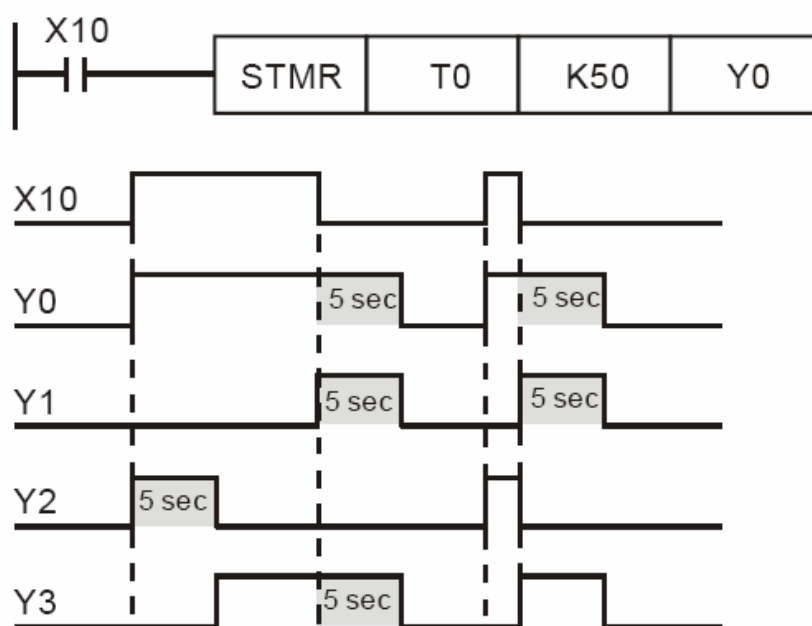
(D): включается одновременно с входным условием инструкции, а выключается после выключения входного условия через заданную в (m) задержку времени;

(D+1): включается одновременно с выключением входных условий инструкции, а выключается через время, заданное в (m);

(D+2): включается одновременно с включением входных условий инструкции, а выключается через время, заданное в (m);

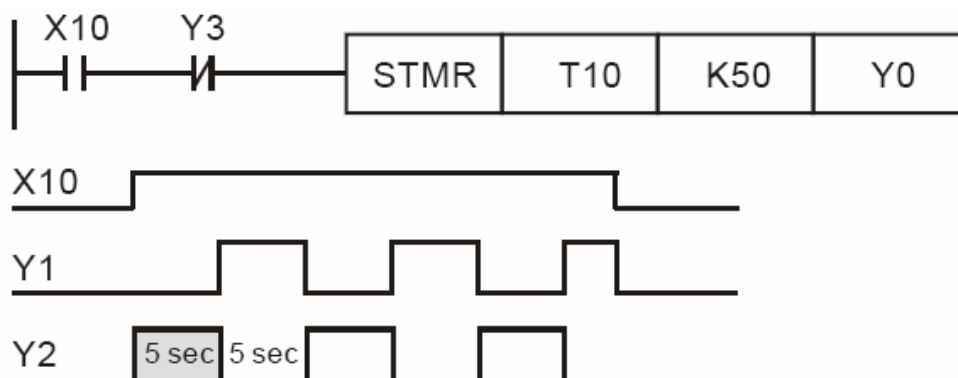
(D+3): включается через время (m) после включения входных условий инструкции, и выключается после выключения входного условия через заданную в (m) задержку времени;

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





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

Пример применения STMR-инструкции в качестве фликера (формирователя импульсов):



Пока включено реле X10 на выходах Y1 и Y2 будут формироваться импульсы шириной 5 сек. Y1 и Y2 будут противоположны по фазе.

API		ALT			Импульсное реле (Т-триггер)	DVP-		
66			P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (3 шага) ALT - Непрерывное выполнение. ALTP – имп. выполнение
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
D		*	*	*											

Примечания: Импульсное выполнение инструкции ALTP в серии ES/EX/SS не поддерживается.

32-х битная инструкция

Флаги: нет

Функция

Реализация триггерной функции. Инвертирование состояния операнда.

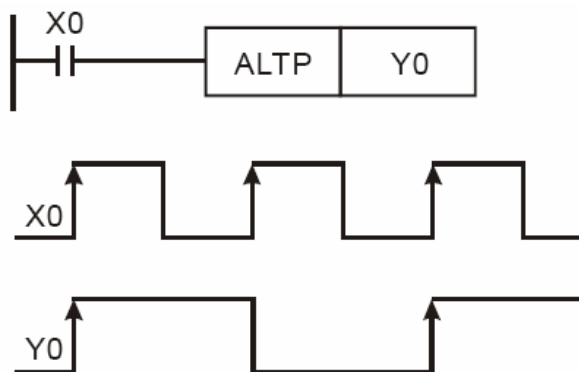
Описание

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

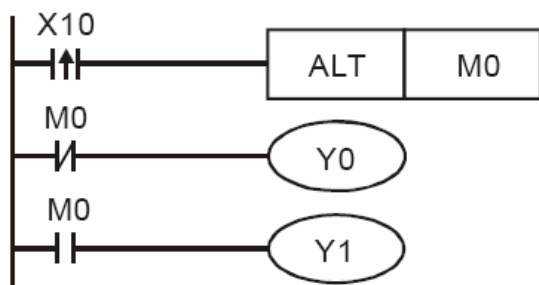
Инструкция выполняется в каждом цикле программы. Этого можно избежать последовательным включением функции импульса (PLS-инструкцией) или параметром "P".

Пример 1: Применение ALT-инструкции для инвертирования состояния операнда.

Состояние выхода Y0 каждый раз инвертируется, если на входе X0 устанавливается сигнал "1".

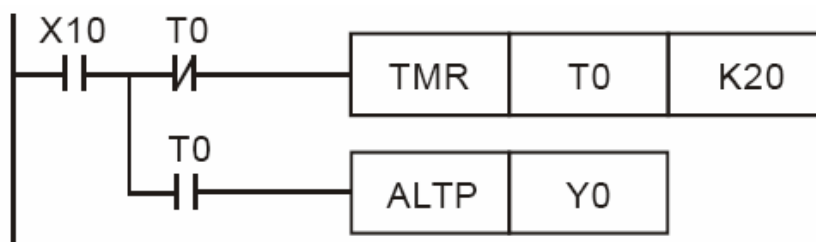


Пример 2: Применение ALT-инструкции. Старт-стоп функция:



Старт выхода Y1 активизируется нажимом кнопки X10. Стоп выхода Y1 активизируется повторным нажимом кнопки X10.

Пример 3: Применение ALT-инструкции. Генератор импульсов:



Когда X10 включено, на выходе Y0 будут генерироваться импульсы длительностью 2 сек.

API	RAMP	<div><div>S₁</div><div>S₂</div><div>D</div><div>n</div></div>	Функция наклонна характеристики (RAMP)	DVP-		
67				ES/EX/SS	SA/SX	EH
				-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (9 шагов) RAMP - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁													*			
S ₂													*			
D													*			
n					*	*										
<div><div>Примечания:</div><div>Диапазон n=1...32767</div></div>																<div><div>32-х битная инструкция</div><div>---</div><div>Флаги: M1026 (старт режима); M1029 (флаг завершения выполнения)</div></div>

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 операции

- Необходимое время выполнения инструкции T составляет: $T = (n \times \text{время цикла программы})$

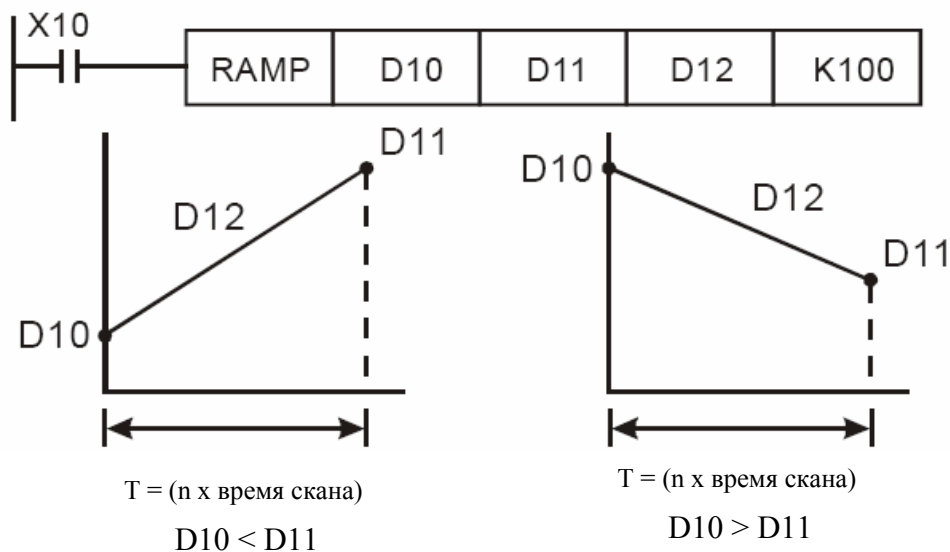
- Если в (D) достигнуто целевое значение, то включается специальный флаг M1029. Целевое значение в дальнейшем сохраняется в (D).
- Количество циклов операций после обработки инструкции запоминается в регистре данных, следующим за (D).

Время исполнения RAMP-инструкции зависит от времени цикла программы. Поэтому ПЛК должно работать с постоянным временем цикла программы, чтобы гарантировать определенное соотношение функции.

Можно повлиять на процесс RAMP-функции с помощью специального реле M1026. Если M1026 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 реле M1026 RAMP-функция сохраняется. Это означает, что до тех пор пока накопленное значение в D будет соответствовать значению в S2 RAMP-функция сохранит мгновенное состояние. M1029 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.

Если RAMP-функция прерывается перед окончанием, позиция до временной точки прерывания сохраняется столько времени, пока снова не появится сигнал запуска. Если снова появится сигнал ramпы, регистры D и D+1 снова отключатся и начнется новый цикл.

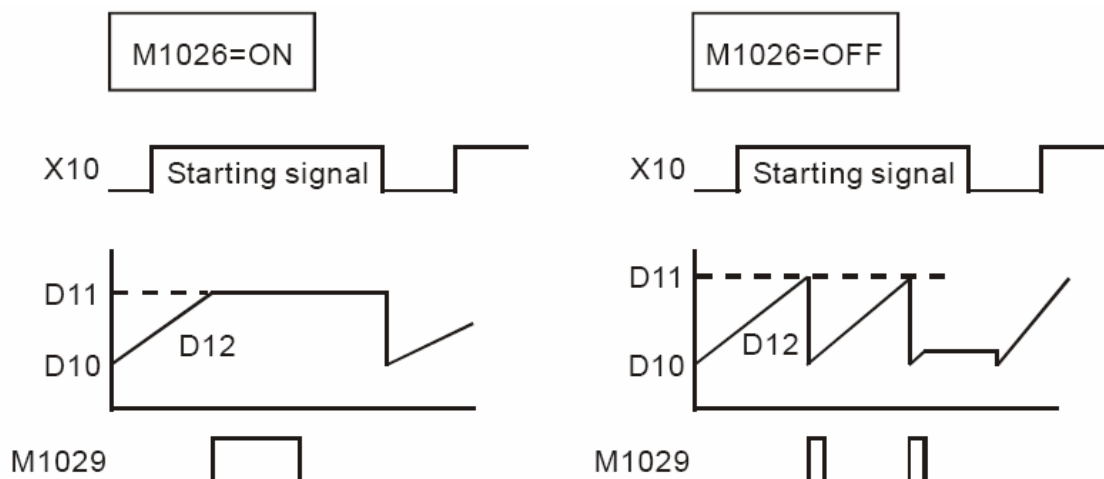
Пример программирования RAMP-инструкции



- После включения X10 D10 (S1) определяет исходное значение. Это значение в D12, последовательно возрастаая (100 раз - "n"), изменяется до тех пор, пока в не будет достигнуто целевое значение, записанное в D11 (S2).
- Время T, необходимое для этого процесса: $T = (n \times \text{время цикла программы})$.
- Количество циклов операций n записано в D13 (D+1).
- Если после определения времени цикла программы, оно несколько больше, чем текущее время цикла программы в регистре данных D1039, активизируется специальное реле M1039 и ПЛК работает с постоянным временем цикла программы.

Если, например, определенное в D1039 значение составляет 30 мс, то в регистре данных D12 потребуются изменения от исходного значения до целевого значения $T = 100 \times 30 = 3 \text{ с}$.

- Если X10 выключено, то прерывается выполнение рамповой функции. Если затем X10 снова включится, выполнение рамповой функции начнется вновь с исходного состояния.
- Если выполнение рамповой функции окончено, включается специальное реле (флаг) 1029, и D12 (D) принимает из D10 (S1) исходное значение.



API	SORT				Сортировка данных	DVP-		
						ES/EX/SS	SA/SX	EH
69				S m1 m2 D n		-	+	+

Операнд	Биты				Слова										16-ти битная инструкция (11 шагов) SORT - Непрерывное выполнение. <u>32-х битная инструкция</u> --- Флаги: M1029 (флаг завершения выполнения)	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S													*			
m ₁					*	*										
m ₂					*	*										
D													*			
n					*	*							*			
<u>Примечания:</u> Диапазоны операндов: m ₁ =1...32 m ₂ =1...6 n=1...m2																

Функция

Сортировка матрицы по значениям колонок (столбцов)

Описание

При вызове SORT-инструкции внутренняя матрица данных, характеризующаяся стартовым регистром данных (S) с величиной строк m₁ и столбцов m₂, сортируется по значениям в столбце n и записывается с новой сортировкой в регистр данных (D).

Указание: При выполнении SORT-инструкции каждая запись в соответствии с данными в выбранном поле сортировки n 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 M1029 включается опознание. Процесс сортировки заканчивается только тогда, когда в m1 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



Когда X0=1, происходит сортировка данных в матрице размером 5x5.

Матрицы данных могут быть представлены в следующей форме.

Неотсортированная матрица:

Номер столбца	1	2	3	4	5
Номер строки	№ группы	математики	физики	химики	историки
1	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
3	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90
4	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
5	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69

Матрица, отсортированная по количеству физиков (D100=K3):

Номер столбца	1	2	3	4	5
Номер строки	№ группы	математики	физики	химики	историки
1	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
3	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
4	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69
5	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90

Матрица, отсортированная по количеству историков (D100=K5):

Номер столбца	1	2	3	4	5
Номер строки	№ группы	математики	физики	химики	историки
1	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
3	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69
4	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
5	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90

API		TKY		S D1 D2	Ввод с 10-ти кнопочной клавиатуры	DVP-		
70	D					ES/EX/SS	SA/SX	EH
						-	+	+

Оп- еранд	Биты				Слова											<u>16-ти битная инструкция</u> (7 шагов) ТКУ - Непрерывное выполнение. <u>32-х битная инструкция</u> (17 шагов) ДТКУ - Непрерывное выполнение. Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												
D ₁								*	*	*	*	*	*	*	*	
D ₂		*	*	*												
<u>Примечания:</u> Операнд S занимает 10 адресов Операнд D ₂ занимает 10 адресов																

Функция

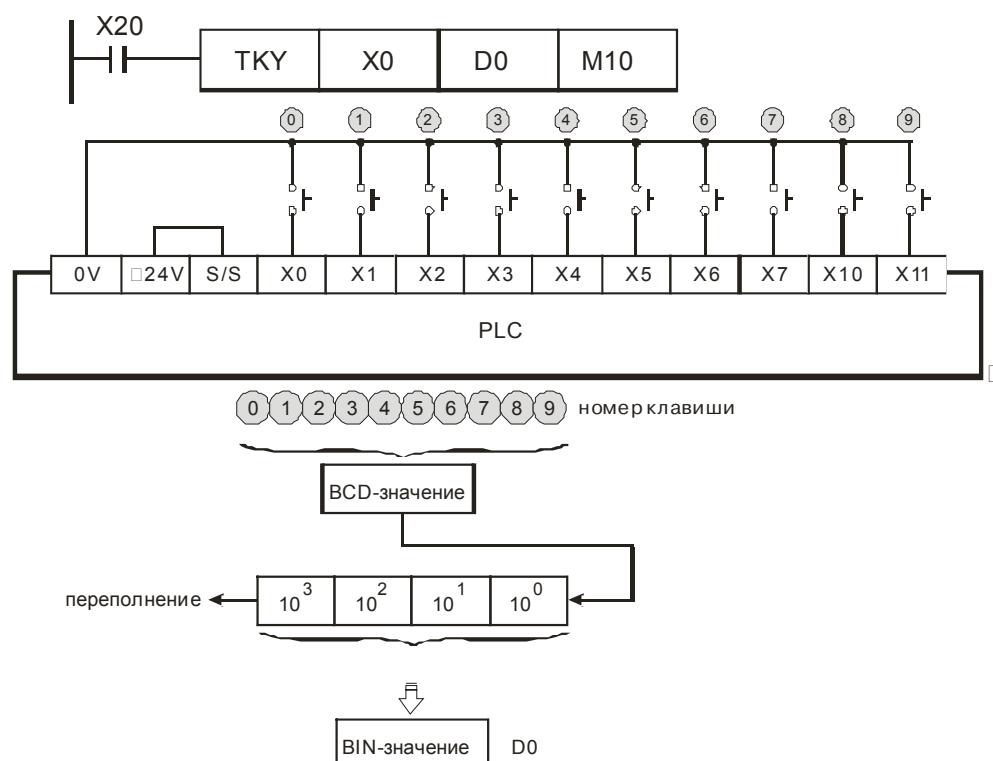
Чтение десятичной клавиатуры по входам

Описание

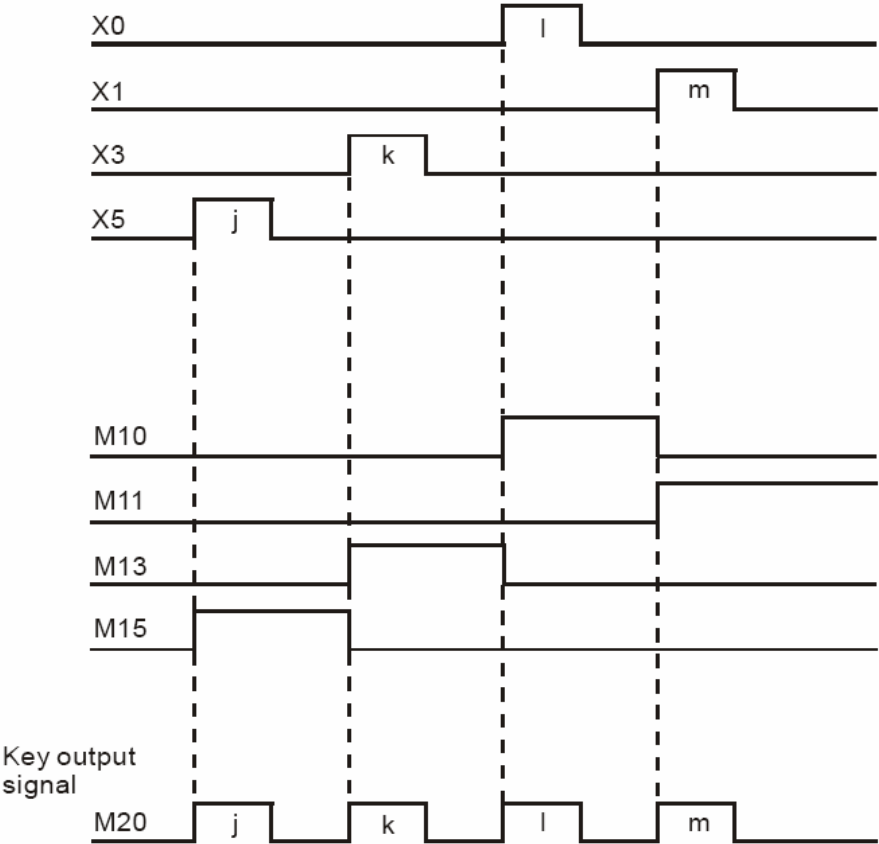
- Клавиатура с 10 клавишами читается в ПЛК по битам (S)...(S+9)
- Указанные значения один за другим заносятся в слово данных (D1). При 16-ти битовом операнде могут записываться 4 места (макс.9999) и при 32-х битовом операнде 8 мест (макс. 99999999).
- Если указываются больше возможных 4 или 8 мест, записываются только последние 4 или 8 мест.
- Биты (D2)...(D2+9) отображают состояния клавиш.

Если TKY-инструкция больше не активна, то биты (D2) стираются. Содержание (D1) сохраняется.

Пример: программирование TKY-инструкции:



Когда X20 включено, считывается состояние клавиш на входах X0 – X11 и значение записывается в регистр D0 в BIN-формате. Реле M10 – M19 включаются при включении соответствующих клавиш.



Если клавиши X0...X5 задействуются в последовательности X5 → X3 → X0 → X1 то в регистре данных D0 будет записано значение 5301. Реле M20 будет включаться при включении любой клавиши.

API		HKEY		S	D1	D2	D3	Ввод с 16-ти кнопочной клавиатуры	DVP-		
									ES/EX/SS	SA/SX	EH
71	D								-	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (9 шагов) HKEY - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															32-х битная инструкция (17 шагов) DNKEY - Непрерывное выполнение. Флаги: M1029 (флаг завершения выполнения) M1167 (переключатель режима)
D1		*														
D2											*	*	*	*	*	
D3		*	*	*												
<p>Примечания: Операнд S занимает 4 адреса Операнд D1 занимает 4 адреса Операнд D3 занимает 8 адресов</p>																

Функция

Чтение шестнадцатеричной клавиатуры по входам

Описание

- Клавиши читаются мультиплексным методом (методом умножения). Соответственно задействовано 4 входа и 4 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 внутренним реле для запоминания функций воздействия на клавиши и контрольных сигналов. Клавиши от A до F включают внутренние реле от (D3) до ((D3)+5). Реле ((D3)+6) заносится при воздействии на одну из клавиш от A до F, а реле ((D3)+6) заносится при воздействии на одну из клавиш от 0 до 9. После каждой регистрации воздействия на клавишу включается реле M1029.
- Число, указанное посредством клавиш от 0 до 9, записывается в (D2). Максимум может быть задано 4 места (макс. 9999).

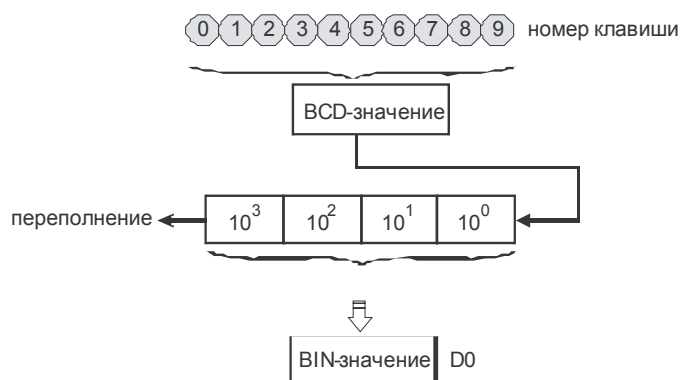
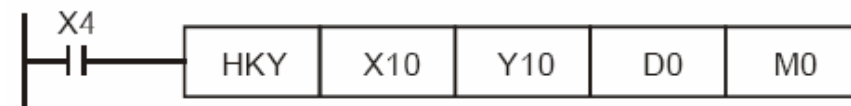
Если выполняется 32-х битный операнд, может быть задано 8 мест (макс. 99999999).

- 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.

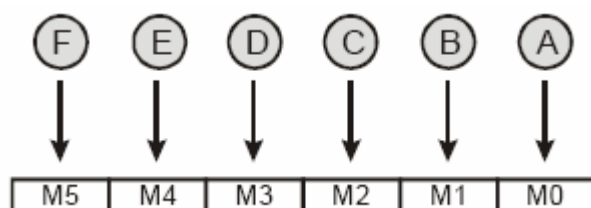
Указание: НКУ-инструкция может применяться в программе только один раз.

При применении НКУ-инструкции ПЛК должен работать с постоянным временем цикла.

Пример программирования НКУ-инструкции:



Функциональные клавиши



Если задействована клавиша A, включается реле M0. M0 остается включенным до тех пор, пока не будут задействованы другие клавиши. Если в конце будет задействована клавиша D, то M0 отключится, а M3 включится.

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

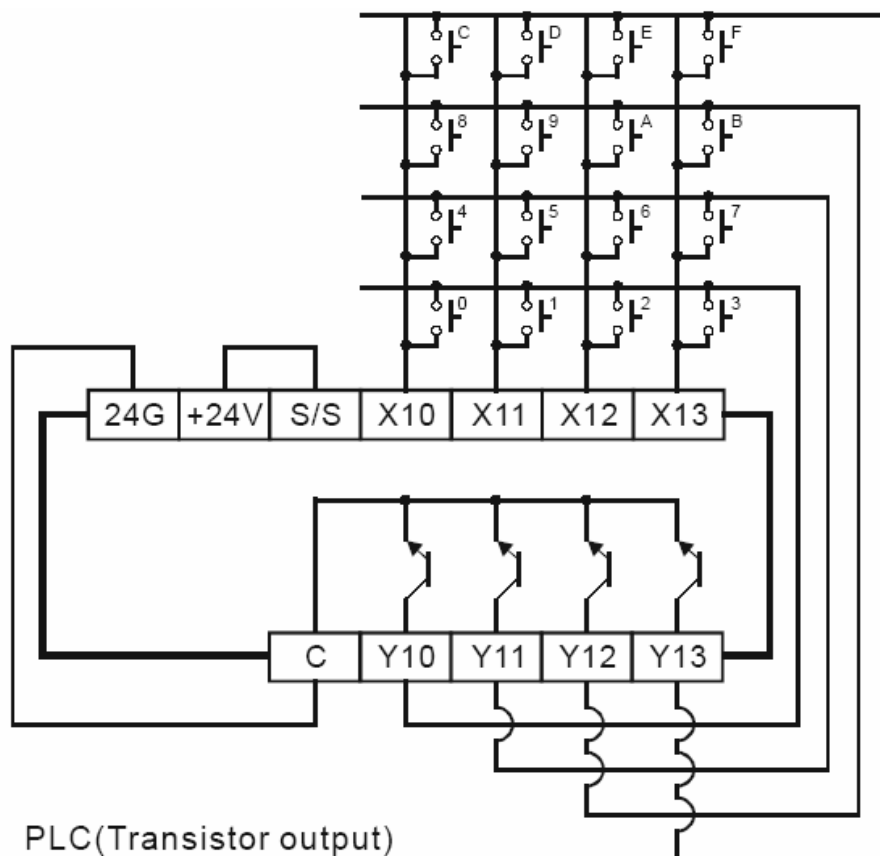
Выходы

До тех пор, пока удерживаются нажатыми клавиши A...F, M6 остается активным. До тех пор, пока удерживаются нажатыми клавиши 1...9, M7 остается активным. Если X4 отключается, данные в D0 не изменяются. Реле от M0 до M7 выключаются. Восприятие воздействия на клавиши требует 8 циклов.

Накопитель данных

Указанные значения записываются в регистр данных D0 четырехзначными.

Внешнее соединение:



Принцип действия со специальным реле M1167

Чтение шестнадцатеричной клавиатуры по входам

Описание

- Включение специального реле M1167 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. FFFFH).

Если выполняется 32-х битный операнд, может быть задано 8 мест (макс. FFFFFFFFH)

- 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: При применении НКУ-инструкции ПЛК должен работать с постоянным временем цикла. Если время цикла очень мало, нужно работать с временным прерыванием.

API		DSW		<div>S</div> <div>D₁</div> <div>D₂</div> <div>n</div>				Ввод с цифрового переключателя		DVP-		
72				ES/EX/SS		SA/SX	EH					
				-		+	+					

Оп-ранд	Биты				Слова											<div>16-ти битная инструкция (9 шагов) DSW - Непрерывное выполнение.</div> <div>32-х битная инструкция ---</div> <div>Флаги: M1029 (флаг завершения выполнения)</div>
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
	S	*														
	D ₁		*													
	D ₂										*	*	*	*	*	
n					*	*										
<div>Примечания:</div> <div>n = 1 – 2</div>																

Функция

Чтение BCD-переключателя по методу мультиплекса (умножения)

Описание

- В ПЛК 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: Для корректного выполнения DSW-инструкции ПЛК должен применяться с транзисторными выходами.

Пример программирования DSW-инструкции

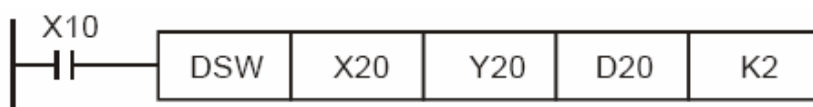
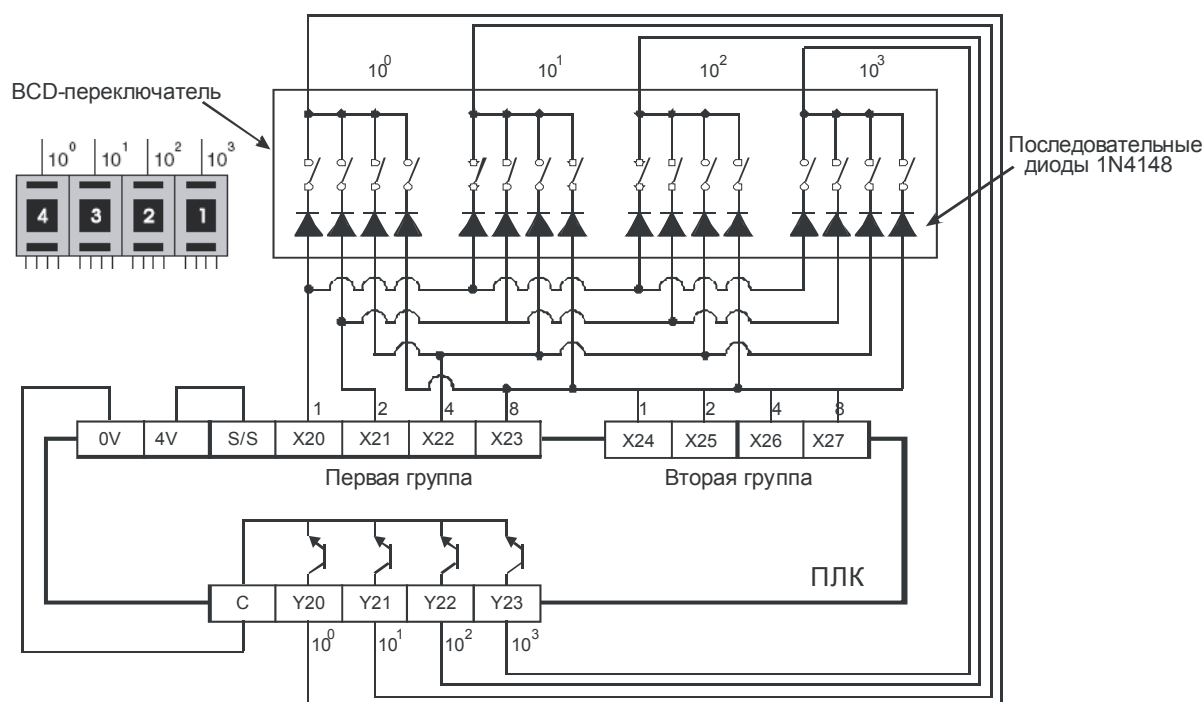


Схема соединения входов/выходов:



Чтение первого четырехзначного блока

Настройка четырехместного цифрового переключателя (BCD), который связан с входами X20...X23, считывается одно место (секция) за другим по выходам Y20...Y23 и запоминается в двоичном коде в регистре данных D20.

В этом случае произведена настройка по $n = 1$.

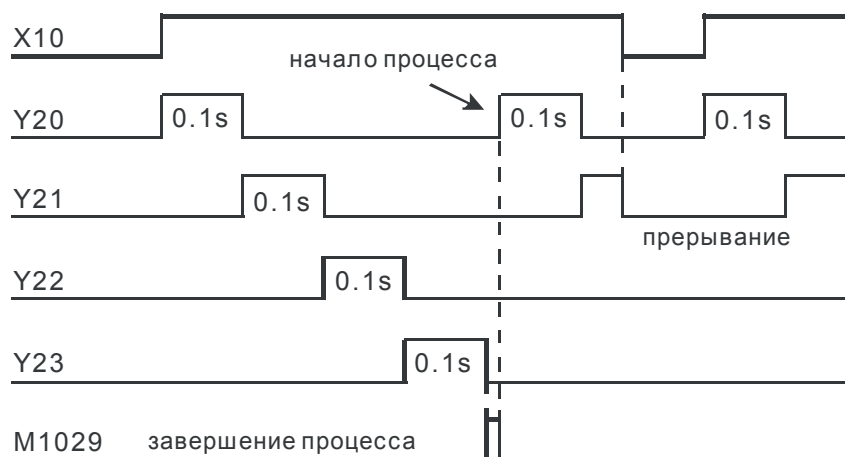
Чтение второго четырехзначного блока

Настройка переключателя (BCD), который связан с входами X24...X27, считывается одно место (секция) за другим по выходам Y20...Y23 и запоминается в двоичном коде в регистре данных D20.

В этом случае произведена настройка по $n = 2$.

Если включен X10, то выходы Y20...Y23 один за другим обрабатывают состояния соответствующих входов X.

Если процесс работы завершен, включается реле M1029.



Если X10 – кнопка без фиксации, то для активизации DSW-инструкции рекомендуется использовать реле с самоподхватом:



API		SEGD		S D	Дешифратор для 7-ми сегментного индикатора	DVP-		
73			P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (5 шагов) SEGD - Непрерывное выполнение. SEGDP - Имп. вып.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S	*																
D		*															
<u>Примечания:</u> Серия ES/SS/EX не поддерживает импульсное выполнение команды SEGDP.																	<u>32-х битная инструкция</u> --- Флаги: нет

Функция

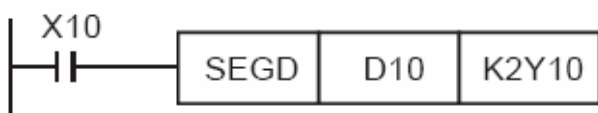
Выдача одноместного шестнадцатеричного числа на 7-ми сегментный цифровой индикатор

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:




Пример программирования:



Когда X10 включено, число записанное в младших четырех битах (b0...b3) регистра D10 будет декодировано в формат 7-ми сегментного индикатора и результат сохранен на выходах: Y10...Y17.

Таблица преобразования данных:

16 бит	Комбинация	Структура 7-ми сегм. индикатора	Состояние каждого сегмента							Данные на дисплее
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		вкл	вкл	вкл	вкл	вкл	вкл	выкл	0
1	0001		выкл	вкл	вкл	выкл	выкл	выкл	выкл	1
2	0010		вкл	вкл	выкл	вкл	вкл	выкл	вкл	2
3	0011		вкл	вкл	вкл	вкл	выкл	выкл	вкл	3
4	0100		выкл	вкл	вкл	выкл	выкл	вкл	вкл	4
5	0101		вкл	выкл	вкл	вкл	выкл	вкл	вкл	5
6	0110		вкл	выкл	вкл	вкл	вкл	вкл	вкл	6
7	0111		вкл	вкл	вкл	выкл	выкл	вкл	выкл	7
8	1000		вкл	вкл	вкл	вкл	вкл	вкл	вкл	8
9	1001		вкл	вкл	вкл	вкл	выкл	вкл	вкл	9
A	1010		вкл	вкл	вкл	выкл	вкл	вкл	вкл	A
B	1011		выкл	выкл	вкл	вкл	вкл	вкл	вкл	B
C	1100		вкл	выкл	выкл	вкл	вкл	вкл	выкл	C
D	1101		выкл	вкл	вкл	вкл	вкл	выкл	вкл	D
E	1110		вкл	выкл	выкл	вкл	вкл	вкл	вкл	E
F	1111		вкл	выкл	выкл	выкл	вкл	вкл	вкл	F

API	SEGL	  	7-ми сегментный индикатор с запоминанием	DVP-		
				ES/EX/SS	SA/SX	EH
74				+	+	+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (7 шагов) SEGL - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S					*	*	*	*	*	*	*	*	*	*	*		
D		*															
n					*	*											
<u>Примечания:</u> n = 0 – 7 В серии EH команда SEGL может быть использована дважды. В серии ES/EX/SS/SA/SX команда SEGL может быть использована один раз.															<u>32-х битная инструкция</u> --- Флаги: M1029 (флаг завершения выполнения)		

Функция

Управление четырехразрядным 7-ми сегментным индикатором с запоминанием показания

Описание

- С помощью этой инструкции можно управлять одним или двумя четырехразрядными 7-ми сегментными индикаторами. Управление выполняется по

мультиплексному методу. Для работы записываются 4 тактовые 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).

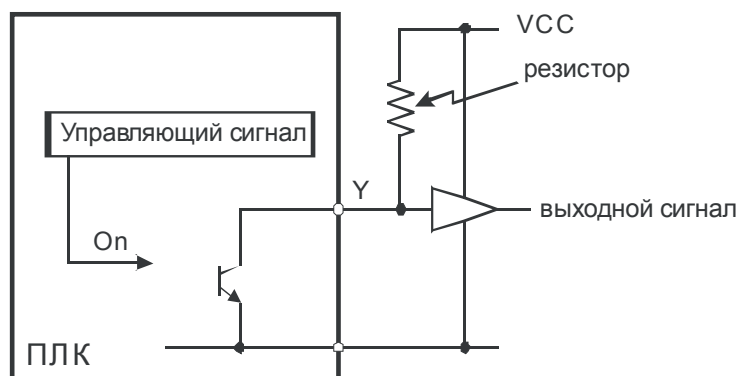
- BCD-код, находящийся соответственно на выходах данных, автоматически распределяется по тактовым выходам ((D)+4)...((D)+7) согласно местам индикатора.

- Настройка (n) зависит от четырех факторов:

- выходной логики выходов ПЛК (+/- включения)
- логики на проводе данных 7-ми сегментного индикатора
- логики тактовых входов 7-ми сегментного индикатора
- количества примененных 7-ми сегментных индикаторов

ОТРИЦАТЕЛЬНАЯ ЛОГИКА (NPN-выход)

Выход имеет низкий (LOW) уровень, если внутренняя логика равна 1.

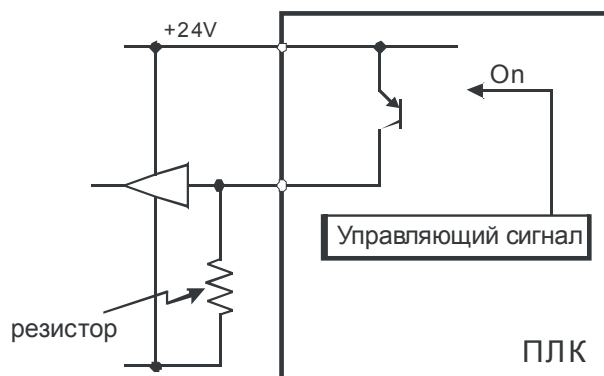


Логический тактовый сигнал: данные запоминаются, если тактовый сигнал имеет низкий уровень (LOW). Логический сигнал данных: активный провод данных является положительным.

BCD-значение				Выход Y (BCD-код)				Выходной сигнал			
b ₃	b ₂	b ₁	b ₀	8	4	2	1	A	B	C	D
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	1	0	1	1	0	1	0	0	1

ПОЛОЖИТЕЛЬНАЯ ЛОГИКА (PNP-выход)

Выход имеет высокий (HIGH) уровень, если внутренняя логика равна 1.



Логический тактовый сигнал: данные запоминаются, если тактовый сигнал высокий (HIGH). Логический сигнал данных: активный провод данных является отрицательным.

BCD-значение				Выход Y (BCD-код)				Выходной сигнал			
b ₃	b ₂	b ₁	b ₀	8	4	2	1	A	B	C	D
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1	0

Операнд (n):

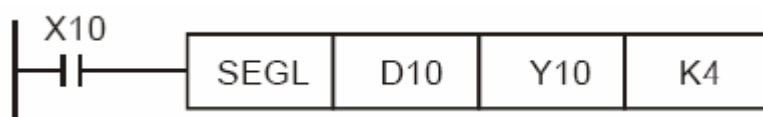
Число групп индикатора	Группа А				Две группы			
Выход Y (BCD-код)	+		-		+		-	
Логический тактовый сигнал	+	-	+	-	+	-	+	-
n	0	1	2	3	4	5	6	7

Указание: Для корректного выполнения инструкции ПЛК должен применяться с транзисторными выходами.

Может применяться только 7-ми сегментный индикатор с сохранением данных.

Время сканирования программы должно быть более 10 мс.

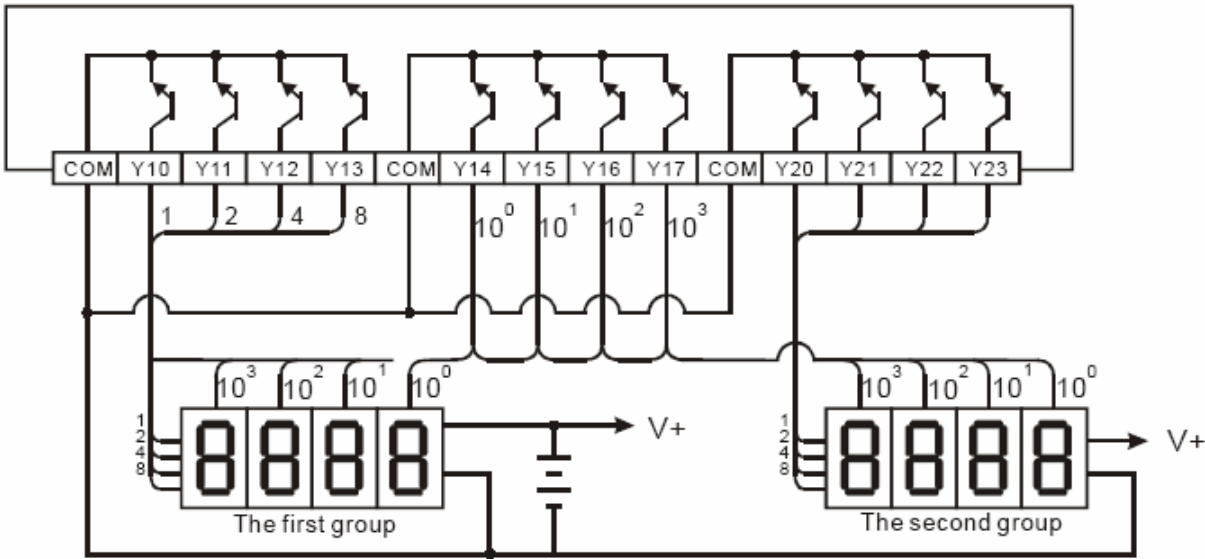
Пример программирования SEGL-инструкции



Когда X10 включено, инструкция будет выполняться. Значение D10 будет конвертироваться в BCD-код и передаваться первой группе индикаторов. Значение D11 будет конвертироваться в BCD-код и передаваться второй группе индикаторов. Если какое-нибудь значение D10 или D11 будет больше 9999, то будет ошибка выполнения.

Когда X10 включено, Y14 – Y17 будут циклически опрашиваться. Каждый цикл опроса нуждается в 12-ти циклах сканирования программы. M1029 будет включаться в конце каждого цикла опроса в течение одного скана программы.

Схема соединения:



API		ARWS		<div><div>S</div><div>D₁</div><div>D₂</div><div>n</div></div>				Ввод со стрелочной клавиатуры						DVP-		
75											ES/EX/SS		SA/SX		EH	
										-		+		+		

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (9 шагов) ARWS - Непрерывное выполнение. <u>32-х битная инструкция</u> --- Флаги: нет	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
	S	*	*	*	*												
	D ₁										*	*	*	*	*		
	D ₂		*														
n					*	*											
<u>Примечания:</u> Операнд S занимает 4 адреса. n = 0 – 3 (см. описание предыдущей инструкции) Команда может использоваться в программе только один раз. В серии SA/SX операнд D2 не может индексироваться (E, F)																	

Функция

Выбор и изменение места четырехместного BCD-числа на 7-ми сегментном индикаторе

Описание

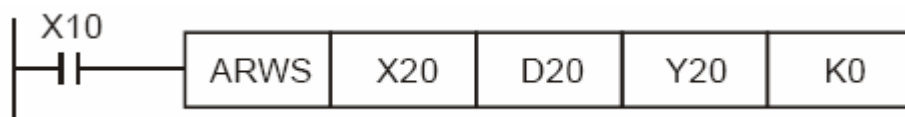
- Опрашиваются четыре клавиши от (S) до ((S)+3):

(S) = увеличение выбранного места
((S)+1) = уменьшение выбранного места
((S)+2) = курсор вправо
((S)+3) = курсор влево

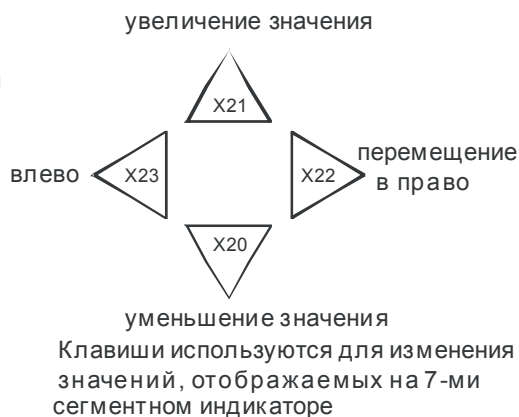
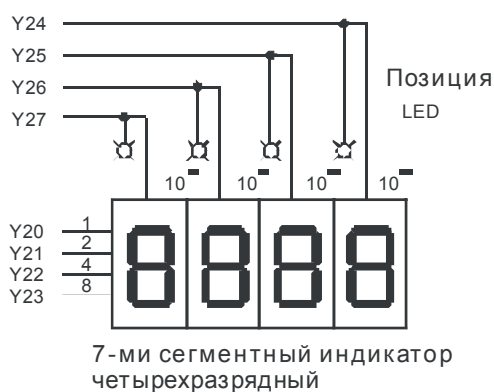
- Данные, заносимые в (D1), отображаются на четырехместном 7-ми сегментном индикаторе и изменяются с помощью клавиш (S)
- Данные, заносимые в (D1), являются двоичными данными.
- С помощью (D2) и (n) определяются выходы и вид схемы подключения, к которой подключается 7-ми сегментный индикатор (см. SEGL-инструкцию).

Для корректного выполнения инструкции ПК должно применяться с транзисторными выходами.

Пример программирования ARWS-инструкции



В 16-ти битовом регистре данных D20 записывается четыре BCD-числа. Каждое BCD-число имеет четыре бита. Максимум в D20 можно записать число 9 999.



Клавишами или входами X20...X23 могут изменяться позиции и числовые значения индикаторов).

X21: счет вверх выбранных мест 0-1-2-3

X20: счет вниз выбранных мест 0-9-8-7

X23: сдвиг влево

X22: сдвиг вправо

Если X10 включен, позиция 103 рассматривается как начальная позиция.

Каждый нажим на X22 и X23 действует так, что изменяются позиции индикаторов в предварительно задаваемой последовательности:

Воздействие на X22 (сдвиг вправо): 103-102 -101 -100 -103
Воздействие на X23 (сдвиг влево): 103-102 -101 -100 -103

Позиция, определенная по X22 и X23, может отображаться (Y24...Y27) дополнительной лампой LED в проводе строб-сигнала.

По клавишам или входам X20...X23 изменяются числовые значения на установленных позициях отображения.

С помощью X20 и X21 определяется последовательность ввода данных.

Данные в регистре данных D20 изменяются в следующей последовательности:

X21: счет вверх: 0 -1 - 2-...8 - 9 - 0 -1

X20: 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 D20 может быть записано желаемое значение и одновременно отображено на 7-ми сегментном индикаторе.

API		ASC		 	ASCII-конвертирование	DVP-		
76						ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты					Слова										<u>16-ти битная инструкция</u> (11 шагов) ASC - Непрерывное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
	S																	
	D										*	*	*					

Примечания:
Операнд S представляет собой 8 буквенно-цифровых символов, введенных из WPLSoft или программатора.

32-х битная инструкция

Флаги: M1161 (режим 8/16)

Функция

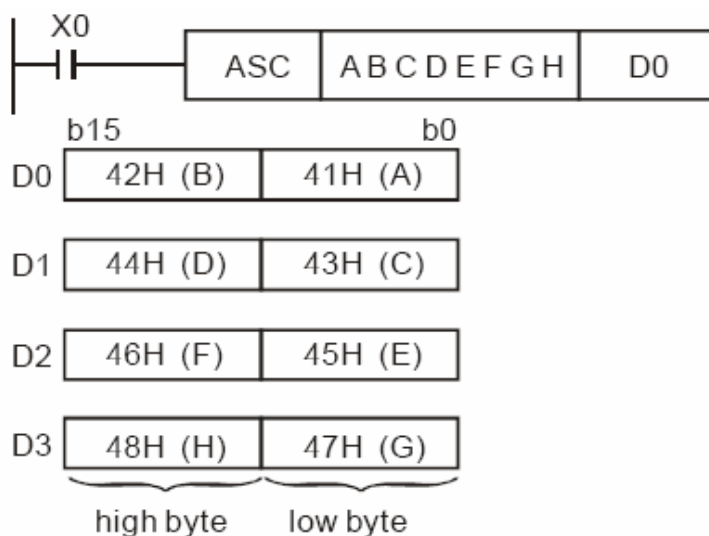
Конвертирование (преобразование) буквенно-цифровых данных в ASCII-данные

Описание

- Буквенно-цифровые данные, указанные в (S), конвертируются в ASCII-знаки и сохраняются в (D).
- Могут задаваться максимум 8 буквенно-цифровых данных.

Пример программирования ASC-инструкции

Когда X0 включено буквенные данные (ABCDEFGH) конвертируются в ASCII-коды и сохраняются в регистры D0-D3



Если M1161=1, буквенные данные (ABCDEFGH) будут конвертироваться в ASCII-коды и сохраняться в младшие байты регистров D0-D3:

	b15	b0
D0	00 H	41H (A)
D1	00 H	42H (B)
D2	00 H	43H (C)
D3	00 H	44H (D)
D4	00 H	45H (E)
D5	00 H	46H (F)
D6	00 H	47H (G)
D7	00 H	48H (H)
	high byte	low byte

Таблица ASCII-кодов:

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		

API		PR		<div>S</div> <div>D</div>		Выдача ASCII-знаков по выходам		DVP-		
77					ES/EX/SS			SA/SX	EH	
								-	+	+

Оп-ранд	Биты				Слова											<u>16-ти битная инструкция</u> (5 шагов) PR - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
	S										*	*	*				
	D		*														

Примечания:

Операнд S занимает 4 адреса. Операнд D занимает 10 адресов.

Команда PR может использоваться дважды в программе.

В серии SA/SX операнд D не может индексироваться (E, F)

32-х битная инструкция

Флаги: M1029 (флаг завершения выполнения)
M1027

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.

Указание: PR-инструкция может использоваться в программе только два раза.

Для корректного выполнения инструкции ПК должно применяться с транзисторными выходами.

Пример программирования PR-инструкции

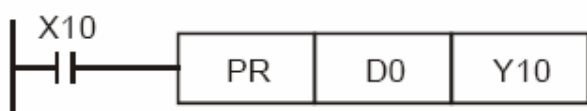
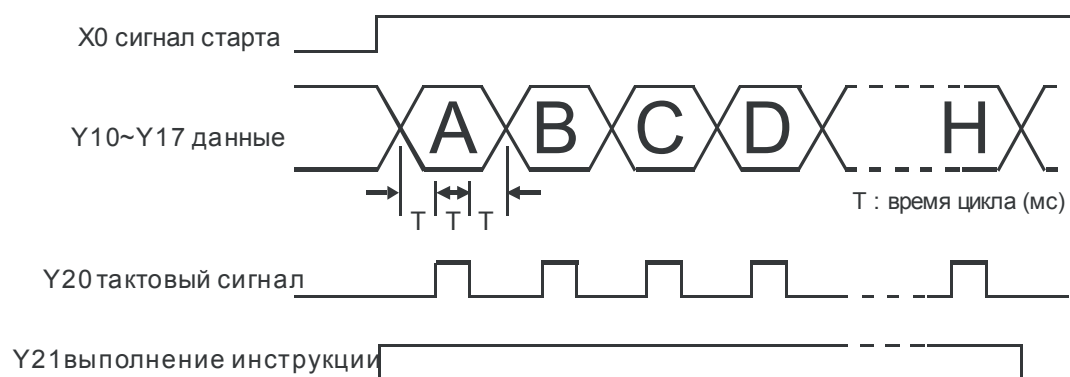


Диаграмма примера программирования для включения входов/выходов



В регистрах данных D0...D3 находятся ASCII-данные примера программирования. Согласно инструкции выдаются буквы "A"... "H".

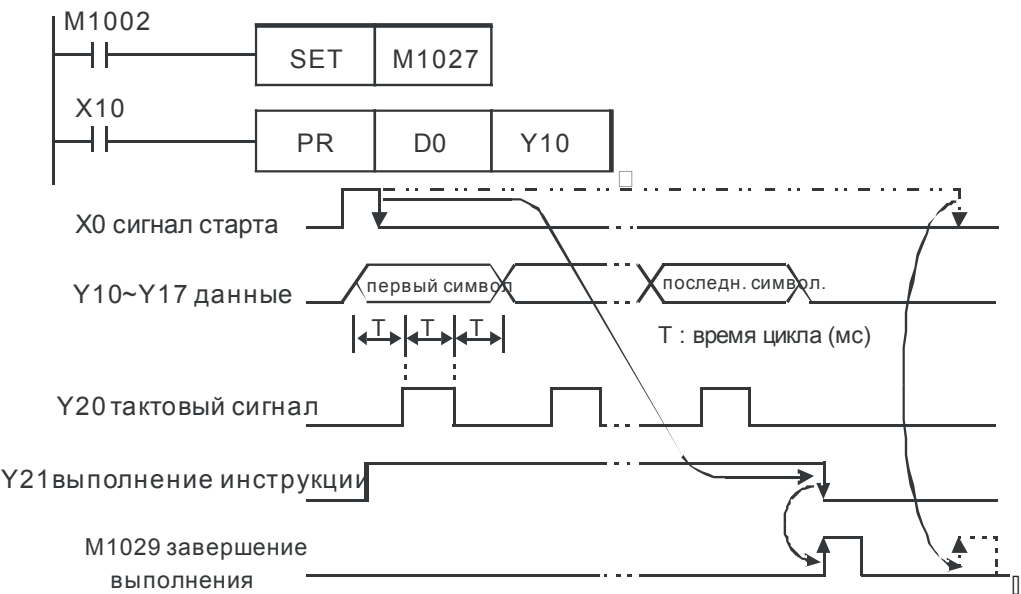
В качестве выходов используются Y10 (младший бит) и до Y17 (старший бит), а также Y20 (строб-сигнал) и Y21 (флаг выполнения).

Формат вывода:

Если X0 во время работы инструкции выключится, то передача данных прекращается. Процесс начинается вновь, как только X0 снова включится.

Если M1027=0, может выводиться от 1 до 8-и символов.

Если M1027=1, может выводиться от 1 до 16-ти символов.



API		FROM	P	<div>m1</div> <div>m2</div> <div>D</div> <div>n</div>				Чтение данных из специального модуля	DVP-		
78	D			ES/EX/SS			SA/SX		EH		
									+	+	+

Оп- еранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
m1					*	*									
m2					*	*									
D								*	*	*	*	*	*	*	*
n					*	*									

Примечания:
Диапазоны операндов: m1=0...7
m2=0...48
n=1...(49-m2)
Серия ES/SS/EX не поддерживает импульсное выполнение команды FROMP

16-ти битная инструкция
(9 шагов)
FROM - Непрерывное
выполнение.
FROMP - Имп. вып.

32-х битная инструкция
(17 шагов)
DFROM - Непрерывное
выполнение.
DFROMP - Имп. вып.

Флаги:
M1083 (флаг разрешения/
запрещения прерывания в
течении выполнения
FROM/TO)

Функция

Чтение данных из контрольных регистров (CR) подключенного специального модуля

Описание

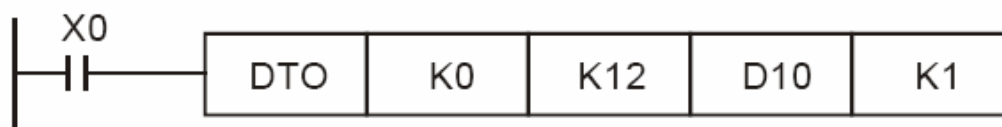
- (m1) - адрес специального модуля.
- (m2) – номер CR-регистра
- (D) – начальный адрес регистров, в которые записываются данные, полученные из (m2)
- (n) – число, регистров считываемых одновременно

Каждый специальный модуль нумеруется последовательно от 0 до 7. Нумерация начинается с модуля, который первым связан с ПЛК. Максимум можно присоединить к ПЛК 8 специальных модулей.

Каждый специальный модуль нумеруется последовательно от 0 до 7. Нумерация начинается с модуля, который первым связан с ПЛК. Максимум можно присоединить к ПЛК 8 специальных модулей.

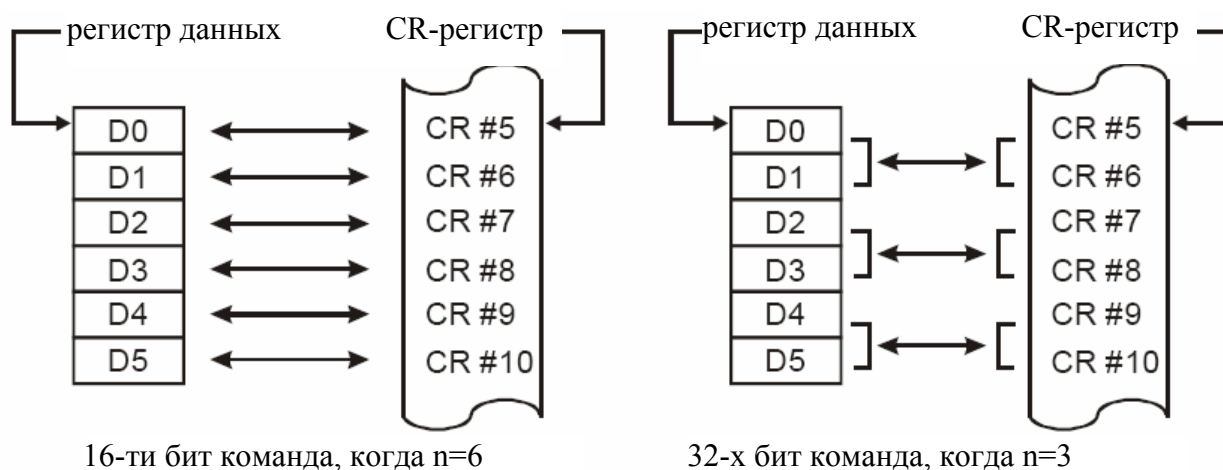
В ПЛК (кроме ES/SS/EX) имеется возможность прерывания выполнения инструкции посредством специального бита M1083.

Пример 1: Запись данных



Когда X0 включено, будут записываться данные из D10 в CR12 и из D11 в CR13 специального модуля #0.

Когда используется 32-х битная инструкция (DFROM/DTO), при n=1, передаются данные 2-х регистров. Когда используется 16-ти битная инструкция (FROM/TO), при n=1, передаются данные 1-го регистра.



Примеры применения инструкций FROM/TO

Пример 1: установка передаточной характеристики в модуле аналогового ввода DVP-04AD. Требуется установить для первого канала CH1 смещение (OFFSET) 0B (=K0_{LSB}) и усиление (GAIN) 2.5B (=K2000_{LSB}).



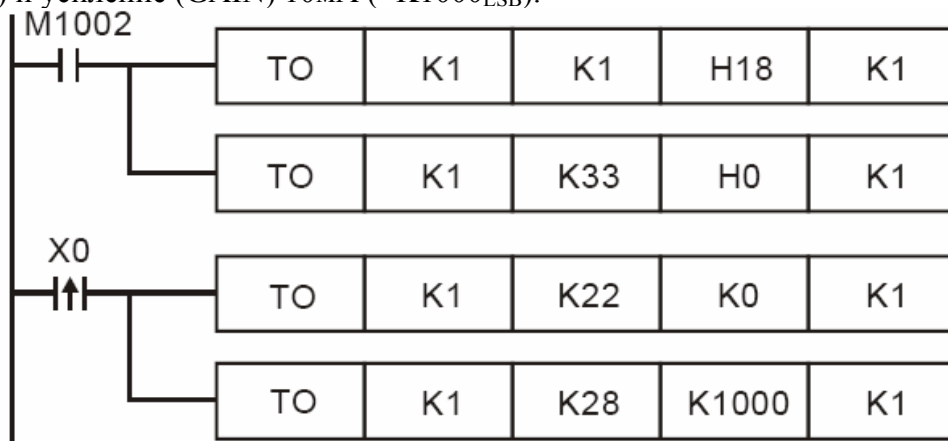
- 1) Запись H0 в CR1 устанавливает режим работы аналогового входа CH1 в диапазоне -10В ... +10В
- 2) Запись H0 в CR33 разрешает корректировку передаточной характеристики CH1
- 3) По переднему фронту включения X0 значение смещения в CR18 будет установлено равным K0 и значение усиления в CR24 будет установлено равным K2000

Пример 2: установка передаточной характеристики в модуле аналогового ввода DVP-04AD. Требуется установить для второго канала CH2 смещение (OFFSET) 2мА ($=K400_{LSB}$) и усиление (GAIN) 18мА ($=K3600_{LSB}$).



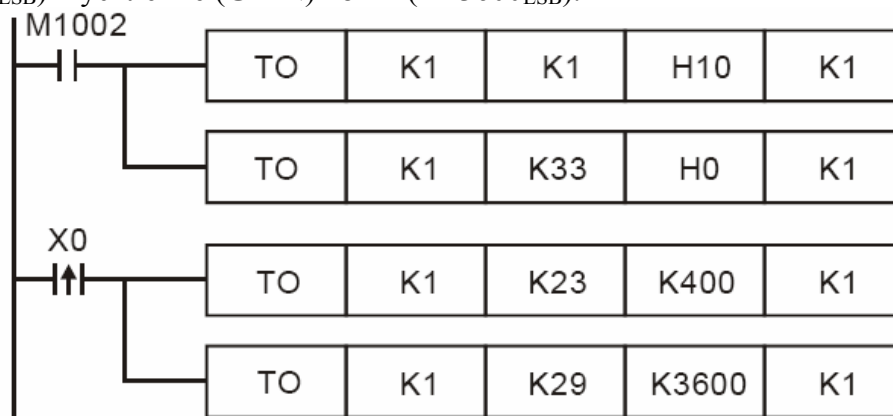
- 1) Запись H18 в CR1 устанавливает режим работы аналогового входа CH2 в диапазоне -20мА ... +20мА
- 2) Запись H0 в CR33 разрешает корректировку передаточной характеристики CH2
- 3) По переднему фронту включения X0 значение смещения в CR19 будет установлено равным K400 и значение усиления в CR25 будет установлено равным K3600

Пример 3: установка передаточной характеристики в модуле аналогового вывода DVP-02DA. Требуется установить для второго канала CH2 смещение (OFFSET) 0мА ($=K0_{LSB}$) и усиление (GAIN) 10мА ($=K1000_{LSB}$).



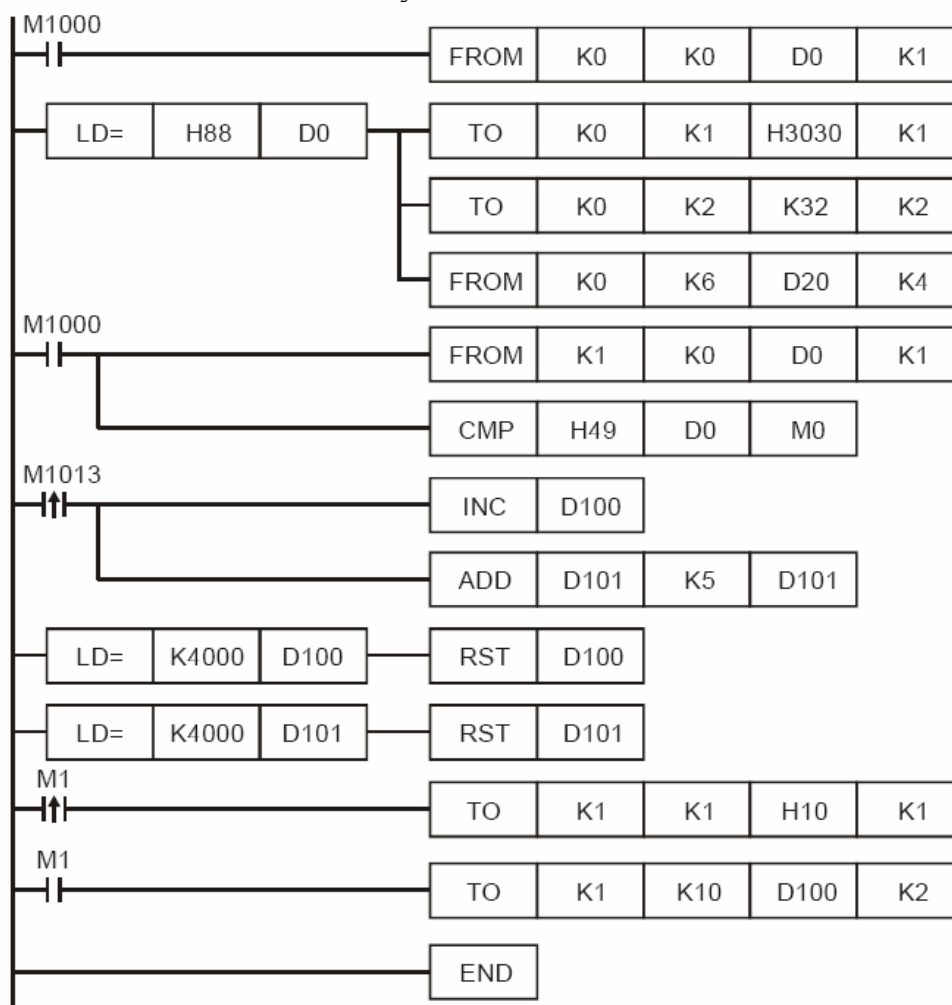
- 1) Запись H18 в CR1 устанавливает режим работы аналогового выхода CH2 в диапазоне 0мА ... 20мА
- 2) Запись H0 в CR33 разрешает корректировку передаточной характеристики CH2
- 3) По переднему фронту включения X0 значение смещения в CR22 будет установлено равным K0 и значение усиления в CR28 будет установлено равным K1000

Пример 4: установка передаточной характеристики в модуле аналогового вывода DVP-02DA. Требуется установить для второго канала CH2 смещение (OFFSET) 2мА ($=K400_{LSB}$) и усиление (GAIN) 18мА ($=K3600_{LSB}$).



- 1) Запись H10 в CR1 устанавливает режим работы аналогового выхода CH2 в диапазоне 4мА ... 20мА
- 2) Запись H0 в CR33 разрешает корректировку передаточной характеристики CH2
- 3) По переднему фронту включения X0 значение смещения в CR23 будет установлено равным K400 и значение усиления в CR29 будет установлено равным K3600

Пример 5: пример программы с одновременным использованием модуля аналогового ввода DVP-04AD и модуля аналогового вывода DVP-02DA.



- 1) Чтение данных о модели специального модуля с адресом #0 и сравнение этих данных с H88 (идентификационный код DVP-04AD)
- 2) Если D0=H88:
 - устанавливаются режимы для аналоговых входов в CR1: для CH1, CH3 – режим 0; для CH2, CH4 – режим 3;
 - в CR2, CR3 устанавливается число усреднения значений входов CH1, CH2 = K32
 - читаются средние значения сигналов с аналоговых входов CH1-CH4 и сохраняются в регистрах D20-D24
- 3) Чтение данных о модели специального модуля с адресом #1 и сравнение этих данных с H49 (идентификационный код DVP-02DA)
- 4) D100 будет увеличиваться на 1 и D101 будет увеличиваться на 5 через 1сек.
- 5) Когда значения D100 и D101 достигнут значения 4000, произойдет их обнуление
- 5) Если D0=H49 (контакт M1=1):
 - устанавливаются режимы для аналоговых выходов в CR1: для CH1 – режим 0; для CH2 – режим 2;
 - записываются значения регистров D100 и D101 в специальные регистры CR10, CR11 и тем самым текущие значения регистров D100 и D101 задают сигнал на аналоговых выходах модуля DVP-02DA

API		RS		<div><div>S</div><div>m</div><div>D</div><div>n</div></div>	Последовательная передача данных	DVP-		
80						ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (9 шагов) RS - Непрерывное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			
S													*			32-х битная инструкция ---	
m					*	*							*				
D													*				
n					*	*							*				
<u>Примечания:</u> Диапазоны операндов: m=0...256 n=0...256																Флаги: M1120 – M1131, M1140 – M1143, M1161	

Функция

Последовательная передача данных внешним устройствам по интерфейсу RS-485

Описание

С помощью RS-инструкции возможны прием и передача от большого числа приборов с последовательным интерфейсом. При этом коммуникация через последовательный интерфейс RS-485 управляется в четыре шага:

- 1) Настройка коммуникационных параметров
- 2) Выдача RS-инструкции, состоящей из:
 - (S) = стартовый адрес передаваемых данных
 - (m) = длина передаваемого сообщения
 - (D) = стартовый адрес принимаемых данных
 - (n) = длина принимаемого сообщения

3) Передача сообщения

4) Прием сообщения

- Если не нужно передавать данные, тогда $m = K0$, если не нужно принимать данные, тогда $n = K0$,
- RS-инструкция может использоваться в программе неограниченное число раз, однако одновременное выполнение двух и более RS-инструкций невозможно.
- Рекомендуется, включить реле M1122 одним импульсным сигналом, так как в противном случае реле после успешной передачи снова сработает на ВКЛ. и повторится передача данных.
- RS-инструкция автоматически управляет приемом сообщений. Как только сообщение полностью принято, данные запоминаются в буферной области приема, и включается специальное реле M1123 в состояние ВКЛ.
- Одновременно передача и прием данных невозможны. Реле M1121 (опознание приема) находится во время процесса приема в состоянии ВКЛ. И хотя реле M1122 (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 D1123 может проверяться, сколько байт актуально было передано. После полного приема сообщения показывается полная длина сообщения.
- При коммуникации данных часто требуется опознать начало и конец сообщения. Обычно это происходит с помощью определенного состояния относительно сообщения, так называемые стартовый и стоповый сигналы. С помощью RS-инструкции имеется возможность автоматически добавить в сообщение стартовый символ и/или конечный символ. Выбираются стартовый и конечные символы включением битов b8 и b9 в регистр данных D1120 параметров коммуникации.
- Для связи с устройствами, поддерживающими протокол MODBUS можно так же использовать инструкции API 100 MODRD, API 101 MODWR и API 150 MODRW

Пример программирования RS-инструкции:



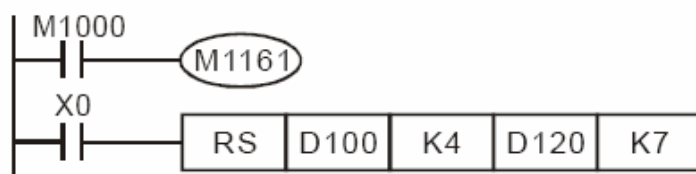
Если X10 включено и M1122=1 начнется последовательная передача содержимого десяти регистров, начиная с D100 и заканчивая D109, в порт RS-485. После завершения передачи данных реле M1122 будет сброшено (не используйте команду RST M1122). Через 1 мс начнется последовательный прием данных из порта RS-485 и сохранение их в регистрах D120 – D129. После завершения приема данных реле M1123 будет сброшено.

Пример 2: 8-бит режим (M1161=1) / 16-бит режим (M1161=0):

<8-бит режим >:

Стартовый и стоповый коды передачи данных должны быть установлены с использованием M1126 и M1130 и назначены в D1124-D1126. ПЛК будет передавать установленные стартовый и стоповый коды автоматически при выполнении RS-инструкции.

Когда M1161=1, будет установлен 8-ти битный режим передачи и 16-ти битные данные будут разделены на старший и младший байты. Старшие байты будут игнорироваться, а младшие байты будут передаваться и приниматься.



Передача данных: (от ПЛК внешним устройствам)

STX	D100L	D101L	D102L	D103L	EXT1	EXT2
Старт. код	<div> <div>\$</div> начальный адрес передаваемых данных (младший байт D100) </div> <div> <div>m</div> длина = 4 </div>				Стоп. код 1	Стоп. код 2

Прием данных: (от внешних устройств в ПЛК)

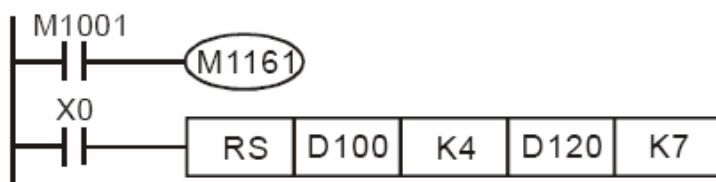
D120L	D121L	D122L	D123L	D124L	D125L	D126L
Старт. код	<div> <div>\$</div> начальный адрес принимаемых данных (младший байт D120) </div> <div> <div>n</div> длина = 7 </div>				Стоп. код 1	Стоп. код 2

ПЛК будет принимать данные, включая стартовый и стоповые коды. Учтите это при задании длины принимаемых данных.

<16-бит режим >:

Стартовый и стоповый коды передачи данных должны быть установлены с использованием M1126 и M1130 и назначены в D1124-D1126. ПЛК будет передавать установленные стартовый и стоповый коды автоматически при выполнении RS-инструкции.

Когда M1161=0, будет установлен 16-ти битный режим передачи и 16-ти битные данные будут разделены на старший и младший байты. Старшие и младшие байты будут передаваться и приниматься.



Передача данных: (от ПЛК внешним устройствам)

STX	D100L	D100H	D101L	D101H	EXT1	EXT2
Старт. код	<div> <div>\$</div> начальный адрес передаваемых данных (младший байт D100) </div> <div> <div>m</div> длина = 4 </div>				Стоп. код 1	Стоп. код 2

Прием данных: (от внешним устройств в ПЛК)

D120L	D120H	D121L	D121H	D122L	D122H	D123L
Старт. код	S начальный адрес принимаемых данных (младший байт D120)			Стоп. код 1	Стоп. код 2	
	n длина = 7					

ПЛК будет принимать данные, включая стартовый и стоповые коды. Учтите это при задании длины принимаемых данных.

Пример 3: Чтение состояния преобразователя частоты VFD-B из шести регистров с начальным адресом H2101 (ASCII режим (M1143=0) / 16-бит режим (M1161=0)):



Передаваемое сообщение от ПЛК к VFD-B: “: 01 03 2101 0006 D4 CR LF “

Принятое сообщение от VFD-B в ПЛК:

“: 01 03 0C 0100 1766 0000 0000 0136 3B CR LF “

Передаваемые данные, расписанные по регистрам:

Регистр	Данные			
D100 младший байт	:	3A H	STX	
D100 старший байт	0	30 H	ADR1	ADR (1, 0) – коммуникационный адрес VFD-B
D101 младший байт	1	31 H	ADR0	
D101 старший байт	0	30 H	CMD1	CMD (1, 0) – код команды
D102 младший байт	3	33 H	CMD0	
D102 старший байт	2	32 H	Стартовый адрес данных	
D103 младший байт	1	31 H		
D103 старший байт	0	30 H		
D104 младший байт	1	31 H		
D104 старший байт	0	30 H	Число данных (слов)	
D105 младший байт	0	30 H		
D105 старший байт	0	30 H		
D106 младший байт	6	36 H		
D106 старший байт	D	44 H	LRC CHK1	LRC CHK (1, 0) – контрольная сумма

D107 младший байт	4	3A H	LRC CHK0	
D107 старший байт	CR	A H	Конец	
D108 младший байт	LF	D H		

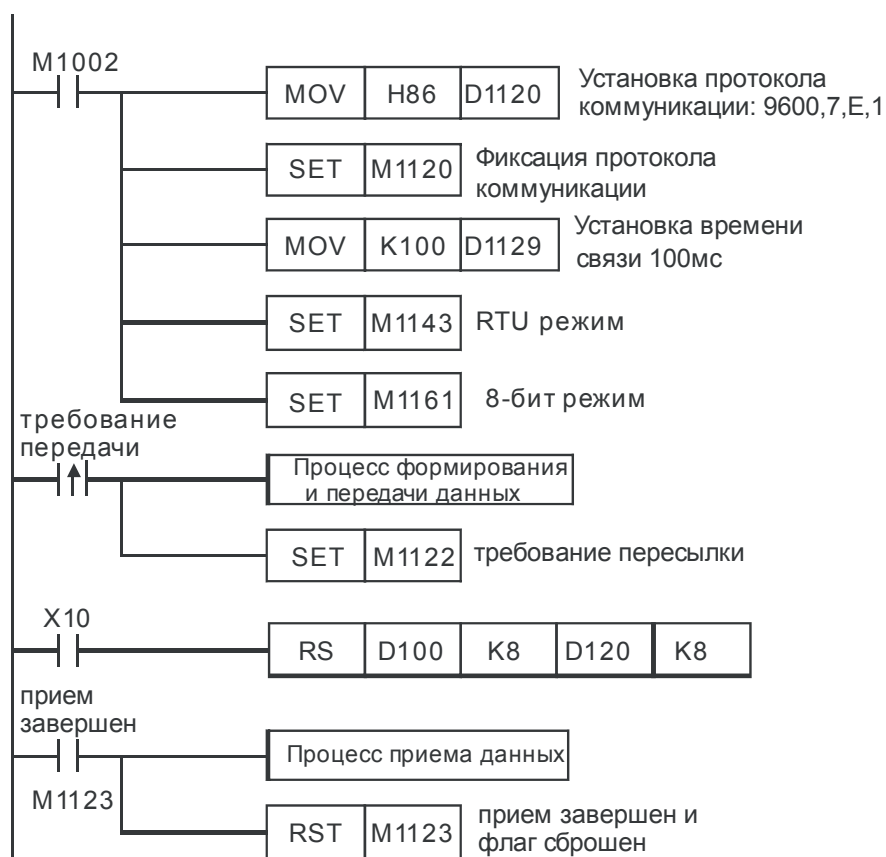
Принимаемые данные, расписанные по регистрам:

Регистр	Данные			
D120 младший байт	:	3A H	STX	
D120 старший байт	0	30 H	ADR1	ADR (1, 0) – коммуникационный адрес VFD-B
D121 младший байт	1	31 H	ADR0	
D121 старший байт	0	30 H	CMD1	CMD (1, 0) – код команды
D122 младший байт	3	33 H	CMD0	
D122 старший байт	0	30 H	Число данных (байт)	
D123 младший байт	C	43 H		
D123 старший байт	0	30 H	Содержимое регистра с адресом 2101H	
D124 младший байт	1	31 H		
D124 старший байт	0	30 H		
D125 младший байт	0	30 H		
D125 старший байт	1	31 H	Содержимое регистра с адресом 2102H	
D126 младший байт	7	37 H		
D126 старший байт	6	36 H		
D127 младший байт	6	36 H		
D127 старший байт	0	30 H	Содержимое регистра с адресом 2103H	
D128 младший байт	0	30 H		
D128 старший байт	0	30 H		
D129 младший байт	0	30 H		
D129 старший байт	0	30 H	Содержимое регистра с адресом 2104H	
D130 младший байт	0	30 H		
D130 старший байт	0	30 H		
D131 младший байт	0	30 H		
D131 старший байт	0	30 H	Содержимое регистра с адресом 2105H	
D132 младший байт	1	31 H		
D132 старший байт	3	33 H		
D133 младший байт	6	36 H		
D133 старший байт	0	30 H	Содержимое регистра с адресом 2106H	
D134 младший байт	0	30 H		
D134 старший байт	0	30 H		
D135 младший байт	0	30 H		
D135 старший байт	3	33 H	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D136 младший байт	B	42 H	LRC CHK0	
D136 старший байт	CR	A H	Конец	
D137 младший байт	LF	D H		

Пример 4: Запись в регистр с адресом H2000 преобразователя частоты VFD-B числа H12 (RTU режим (M1143=1) / 8-бит режим (M1161=1):

Передаваемое сообщение от ПЛК к VFD-B: **01 06 2000 0012 02 07**

Принятое сообщение от VFD-B в ПЛК: **01 06 2000 0012 02 07**



Передаваемые данные, расписанные по регистрам:

Регистр	Данные	
D100 младший байт	01 H	Коммуникационный адрес VFD-B
D101 младший байт	06 H	Адрес команды
D102 младший байт	20 H	Адрес данных
D103 младший байт	00 H	
D104 младший байт	00 H	Данные
D105 младший байт	12 H	
D106 младший байт	02 H	CRC CHK Low
D107 младший байт	07 H	CRC CHK High

Принимаемые данные, расписанные по регистрам:

Регистр	Данные	
D120 младший байт	01 H	Коммуникационный адрес VFD-B
D121 младший байт	06 H	Адрес команды
D122 младший байт	20 H	Адрес данных
D123 младший байт	00 H	
D124 младший байт	00 H	Данные
D125 младший байт	12 H	
D126 младший байт	02 H	CRC CHK Low
D127 младший байт	07 H	CRC CHK High

Примечания:

- Флаги, относящиеся к инструкциям последовательной передачи данных по RS-485 интерфейсу: RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW

M1120	Сохранение протокола коммуникации. Если M1120=0, то коммуникационный протокол установленный в D1120 будет сбрасываться на значение по умолчанию в конце каждого скана. M1120=1 фиксирует заданное пользователем значение протокола коммуникации.	Устанавливается и сбрасывается пользователем
M1121	Готовность передачи по RS-485. Когда M1121=0, идет передача информации по RS-485	Устанавливается и сбрасывается системой
M1122	Требование передачи данных по RS-485. Для начала передачи и приема данных по RS-485 командами RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW пользователь должен установить M1122=1 (импульс). M1122 будет сброшено автоматически после завершения передачи данных.	Устанавливается пользователем, сбрасывается системой
M1123	Прием данных по RS-485 завершен. M1123 будет автоматически включено после выполнения инструкций RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW. Когда прием данных завершится, пользователь должен сбросить M1123.	Устанавливается системой, сбрасывается пользователем
M1124	Ожидание приема данных по RS-485. Когда M1124=1, ПЛК ожидает прием данных.	Устанавливается и сбрасывается системой
M1125	Сброс коммуникации. Когда M1125=1, связь по RS-485 будет сброшена. После этого надо выключить M1125.	Устанавливается и сбрасывается пользователем
M1126	Выбор STX/CTX (пользователь/система). См. следующую таблицу	
M1127	Прием данных по RS-485 завершен (от команд MODRD, RDST, MODRW)	Устанавливается системой, сбрасывается пользователем
M1128	Индикация приема/передачи по RS-485	Устанавливается и сбрасывается системой
M1129	Время приема по RS-485 вышло. M1129 будет включено, если время ожидания приема превысит значение, установленное в D1129.	Устанавливается системой, сбрасывается пользователем
M1130	Выбор STX/CTX. См. следующую таблицу	Устанавливается и сбрасывается пользователем
M1131	M1131=1, когда данные конвертируются в HEX (от команд MODRD, RDST, MODRW), иначе M1131=0	Устанавливается и сбрасывается системой
M1140	Ошибка приема данных по RS-485 (от команд MODRD, RDST, MODRW)	
M1141	Ошибка команд MODRD, RDST, MODRW	
M1142	Ошибка приема данных по RS-485 (от команды VFD-A)	
M1143	Выбор режима ASCII/RTU (для команд MODRD, MODWR, MODRW): M1143 = 1 – режим RTU M1143 = 0 – режим ASCII	Устанавливается и сбрасывается пользователем
M1161	Режим 8/16 бит (M1161 = 1 – 8 бит)	

- Специальные регистры, относящиеся к инструкциям последовательной передачи данных по RS-485 интерфейсу: RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW

D1038	Для SA/SX/EH: время задержки ответа, когда ПЛК – ведомый (slave) (ед. = 0.1 мс)
D1050 ... D1055	После выполнения MODRD/RDST ПЛК будет автоматически конвертировать ASCII данные из D1070-D1085 в шестнадцатеричный формат (HEX) и сохранять их в D1050-D1055
D1070 ... D1085	Регистры для хранения данных принятых по RS-485. Сообщения, принятые от ведомых устройств по RS-485 будут сохраняться в этих регистрах.

D1089 ... D1099	Регистры для хранения данных отправленных по RS-485. Сообщения, отправленные ведомым устройствам по RS-485 будут сохраняться в этих регистрах.
D1120	Протокол коммуникации по RS-485 (H86 = 9600, 7, E, 1). См. следующую таблицу
D1121	Коммуникационный адрес ПЛК, когда он ведомый (slave)
D1122	Остаточные слова от переданных данных
D1123	Остаточные слова от принятых данных
D1124	Определение стартового слова (STX)
D1125	Определение стопового слова 1 (EXT1)
D1126	Определение стопового слова 2 (EXT2)
D1129	Время ожидания ответа по RS-485 (мс)
D1130	Код ошибки отклика MODBUS
D1256 ... D1295	Регистры для хранения данных принятых/переданных по RS-485 командой MODRW
D1296 ... D1311	ПЛК будет автоматически конвертировать ASCII данные, сохраненные в D1256-D1295 в шестнадцатеричный формат и сохранять их в D1296-D1311

- Установка протокола коммуникации (D1120)

Каждый протокол последовательного обмена должен сначала конфигурироваться, чтобы гарантировать полную совместимость с внешним коммутационным устройством. Протокол обмена конфигурируется с помощью специального регистра D1120. Однако это возможно только для неактивной RS-инструкции. На приведенной ниже таблице показано соответствие специального регистра D1120 и его значение для порта RS-485.

	Содержимое	0	1
b0	Длина данных	7	8
b1	Паритет	00: нет паритета (none); 01: паритет по чету (odd); 10 паритет по нечету (even)	
b2			
b3	Стоповый бит	1 бит	2 бита
b4	Скорость (бит/с)	0001 (H1): 110, 0010 (H2): 150, 0011 (H3): 300, 0100 (H4): 600, 0101 (H5): 1200, 0110 (H6): 2400, 0111 (H7): 4800, 1000 (H8): 9600, 1001 (H9): 19200, 1010 (HA): 38400, 1011 (HB): 57600 (только для DVP-SA/SX/EH) 1100 (HC): 115200 (только для DVP-SA/SX/EH)	
b5			
b6			
b7			
b8	Выбор стартового слова	нет	D1124
b9	Выбор первого стопового слова	нет	D1125
b10	Выбор второго стопового слова	нет	D1126
b15 – b11	не определены		

Специальный регистр D1124 содержит значение стартового слова, 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 D1125 содержит значение стопового слова 1, в случае если он был выбран. Значением по умолчанию является ASCII "ETX" или 03H. Однако оно может быть изменено пользователем перед началом сеанса связи.

Специальный регистр D1126 содержит значение стопового слова 2, в случае если он был выбран. По умолчанию оно не установлено. Однако оно может быть изменено пользователем перед началом сеанса связи.

		M1130	
		0	1
M1126	0	D1124: определено пользователем D1125: определено пользователем D1126: определено пользователем	D1124: H 0002 D1125: H 0003 D1126: H 0000 (не установлено)
	1	D1124: определено пользователем D1125: определено пользователем D1126: определено пользователем	D1124: H 003A (':') D1125: H 000D (CR) D1126: H 000A (LF)

Пример установки коммуникационного формата:

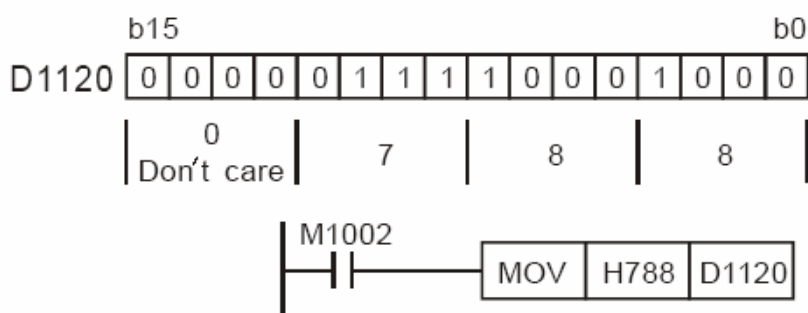
9600 7, N, 2

STX : “:”

ETX1 : “CR”

EXT2 : “LF”

В D1120 должно быть записано H788:



M1126=1, M1130=1

- Выбор режима ASCII или RTU (M1143)

ASCII режим (M1143=0):

STX	Стартовый символ “:” (3AH)
ADR 1	Коммуникационный адрес: 8-bit адрес, состоящий из 2 ASCII кодов
ADR 0	
CMD 1	Код команды: 8-bit адрес, состоящий из 2 ASCII кодов
CMD 0	
DATA (0)	Содержание данных: n x 8-bit данных, состоящих из 2-х ASCII кодов. n<=37, максимум 74 ASCII кодов
DATA (1)	
.....	
DATA (n-1)	
LRC CHK 1	LRC контрольная сумма: 8-bit контрольная сумма, состоящая из 2 ASCII кодов
LRC CHK 0	
END 1	Конечный символ: END1= CR (ODH), ENDO= LF(OAH)
END 0	

Каждый 8-bit блок данных есть комбинация двух ASCII символов. Для примера, 1-байт данных: 64 Hex, показан как '64' в ASCII, состоит из '6' (36 Hex) и '4' (34Hex).

Символ	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII код	30H	31H	32H	33H	34H	35H	36H	37H

Символ	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
ASCII код	38H	39H	41H	42H	43H	44H	45H	46H

ADR (Коммуникационный адрес):

Допустимый коммуникационный адрес должен быть выбран из диапазона 0 ... 254. Коммуникационный адрес равный 0 – средство трансляции всем ведомым устройствам одновременно, в этом случае, ведомые устройства не будут отвечать ни на какое сообщение ведущему устройству.

Для примера, связь VFD с адресом 16 decimal:

(ADR 1, ADR 0)='1','0' => '1'=31H, '0'=30H

CMD (код команды) и DATA (символы данных):

Формат символов данных зависит от командных кодов. Доступные командные коды - 03H, чтение N слов. Максимальное значение N это 12. Код команды: 06H, запись 1 слово.

LRC (продольная проверка избыточности) рассчитана в итоге, модуль 256, значение байтов от ADR1 до последнего символа данных, тогда вычисление шестнадцатеричного представления 2-ух дополнений отрицание суммы. Для примера, читая 1 слово с адреса 0401H преобразователя с адресом 01H. 01H+03H+04H+01H+00H+01H=0AH,

2-ух дополнений отрицание 0AH есть **F6H**.

Для примера, чтение непрерывных 2 слов от начального адреса 2102H VFD-B с адресом 01H.

ASCII режим:

Командное сообщение:	
STX	':'
ADR 1	'0'
ADR 0	'1'
CMD 1	'0'
CMD 0	'3'
Стартовый адрес данных	'2' '1' '0' '2'
Число (в словах)	'0' '0' '0' '2'

Ответное сообщение:	
STX	':'
ADR 1	'0'
ADR 0	'1'
CMD 1	'0'
CMD 0	'3'
Число данных (в байтах)	'0' '4'
Содержание данных по стартовому адресу	'1' '7' '7'
2102H	'0'
Содержание данных по адресу 2103H	'0' '0'

LRC CHK	'D'		'0'
LRC CHK	'7'		'0'
END 1	CR	LRC CHK 1	'7'
END 0	LF	LRC CHK 0	'1'
		END 1	CR
		END 0	LF

RTU режим (M1143=1):

START	интервал молчания
ADR	Адрес коммуникации: 8-bit адрес
CMD	Код команды: 8-bit команда
DATA (n-1)	Содержание данных:
.....	n x 8-bit данных. n<=25
DATA0	
CRC CHK Low	CRC контрольная сумма:
CRC CHK High	16-bit контрольная сумма из 2-ух 8-bit символов
END	интервал молчания - более 10 мс

Каждый 8-bit блок данных - комбинация двух 4-битных шестнадцатеричных символов. Для примера, 64 Hex.

START: DVP-ES/EX/SS/SA/SX не могут иметь интервал молчания менее 10 мс.

EH серия:

Скорость передачи (бит/с)	Интервал молчания RTU (мс)	Скорость передачи (бит/с)	Интервал молчания RTU (мс)
300	40	9600	2
600	21	19200	1
1200	10	38400	1
2400	5	57600	1
4800	3	115200	1

ADR (Коммуникационный адрес):

Допустимый коммуникационный адрес должен быть выбран из диапазона 0 ... 254. Коммуникационный адрес равный 0 – средство трансляции всем ведомым устройствам одновременно, в этом случае, ведомые не будут отвечать ни на какое сообщение ведущему устройству.

Для примера, связь VFD с адресом 16 decimal:

RTU режим: (ADR)=10H

CMD (код команды) и DATA (символы данных):

Формат символов данных зависит от командных кодов. Доступные командные коды - 03H, чтение N слов. Максимальное значение N это 12. Код команды: 06H, запись 1 слово.

CRC (циклическая проверка по избыточности) рассчитанная следующими шагами:

Шаг 1 : Загрузка 16-bit регистра (называемого CRC регистром) с FFFFH;

Шаг 2: Исключающее ИЛИ первому 8-bit байту из командного сообщения с байтом младшего порядка из 16-bit регистра CRC, помещение результата в CRC регистр.

Шаг 3: Сдвиг одного бита регистра CRC вправо с MSB нулевым заполнением. Извлечение и исследование LSB.

Шаг 4: Если LSB CRC регистра равно 0, повторите шаг 3, в противном случае исключающее ИЛИ CRC регистра с полиномиальным значением A001H.

Шаг 5: Повторяйте шаг 3 и 4, до тех пор, пока восемь сдвигов не будут выполнены. Затем, полный 8-bit байт будет обработан.

Шаг 6: Повторите шаг со 2 по 5 для следующих 8-bit байтов из командного сообщения.

Продолжайте пока все байты не будут обработаны. Конечное содержание CRC регистра CRC значение. При передачи значения CRC в сообщении, старшие и младшие байты значения CRC должны меняться, то есть сначала будет передан младший байт.

Для примера, чтение непрерывных 2 слов от начального адреса 2102H VFD-B с адресом 01H.

RTU режим:

Командное сообщение:	
ADR	01H
CMD	03H
Стартовый адрес данных	21H
	02H
Число данных в словах	00H
	02H
CRC CHK Low	6FH
CRC CHK High	F7H

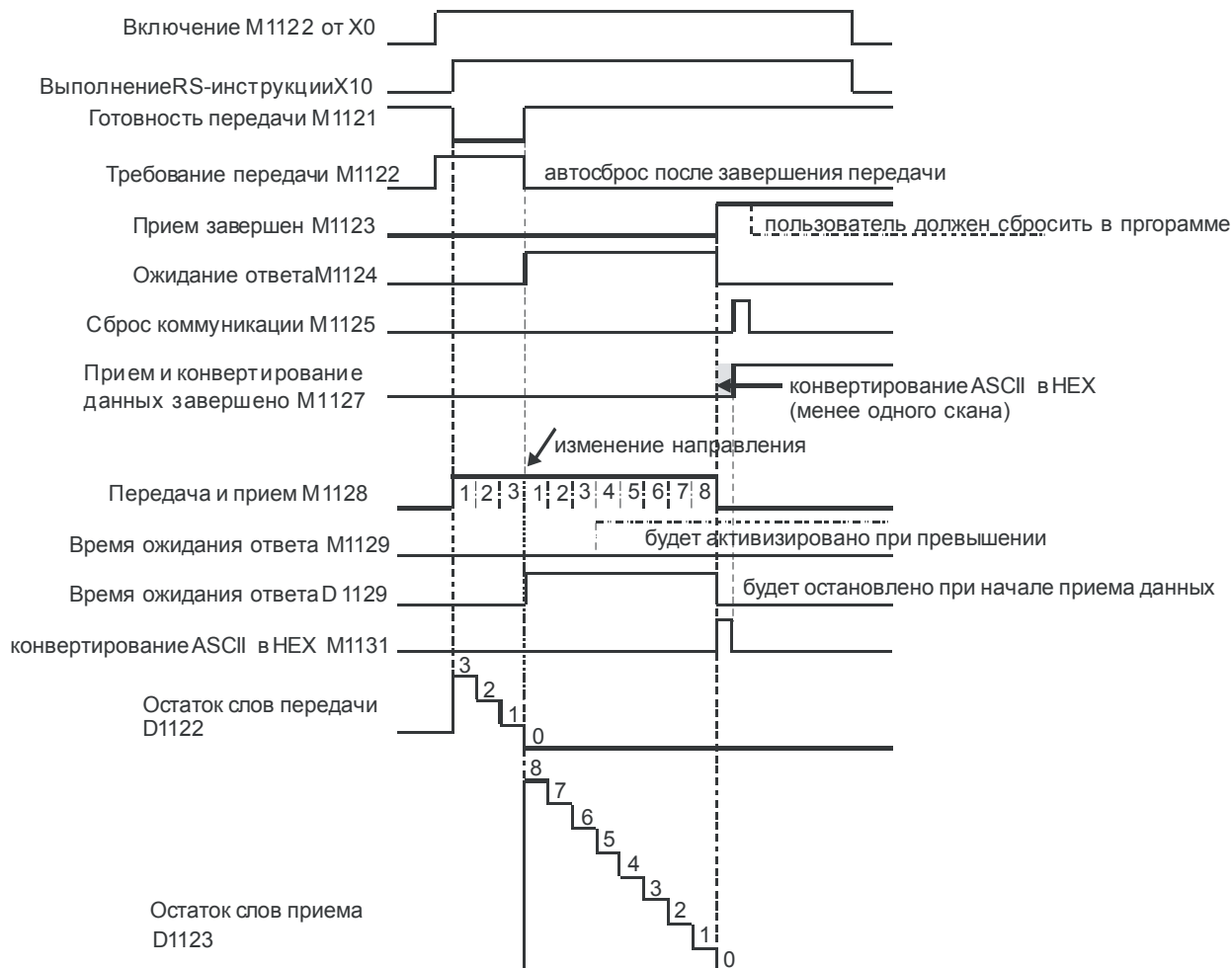
Ответное сообщение:	
ADR	01H
CMD	03H
Число данных в байтах	04H
Содержание данных по адресу 2102H	17H
	70H
Content of data address 2103H	00H
	00H
CRC CHK Low	FEH
CRC CHK High	5CH

END: DVP-ES/EX/SS/SA/SX не могут иметь интервал молчания менее 10 мс.

EH серия:

Скорость передачи (бит/с)	Интервал молчания RTU (мс)	Скорость передачи (бит/с)	Интервал молчания RTU (мс)
300	40	9600	2
600	21	19200	1
1200	10	38400	1
2400	5	57600	1
4800	3	115200	1

- Временная диаграмма коммуникации по RS-485



API		PRUN														DVP-		
																ES/EX/SS	SA/SX	EH
81	D															-	+	+

Оп- еранд	Биты				Слова										16-ти битная инструкция (5 шагов)			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E				
S							*		*						PRUN - Непрерывное выполнение. PRUNP - Имп. вып.			
D								*	*									

Примечания: Когда операнд S определен как KnX, операнд D должен быть KnM. Когда операнд S определен как KnM, операнд D должен быть KnY.															32-х битная инструкция (9 шагов) DPRUN - Непрерывное выполнение. DPRUNP - Имп. вып. Флаги: нет			
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--

Функция

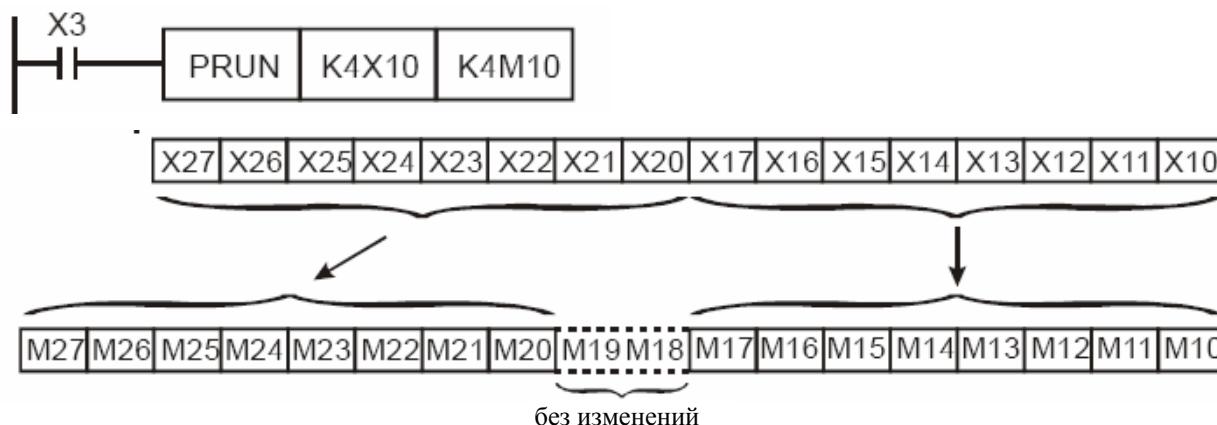
Перенос состояния входов или внутренних реле в область данных, автоматически передаваемую в параллельном режиме.

Описание

- PRUN-инструкция отличается от MOV-инструкции тем, что она работает в восьмеричном коде.
- (S) – источник данных, (D) приемник данных
- Так как передача выполняется восьмерично, (S) должно начинаться с X0/M0, 10,20,30...

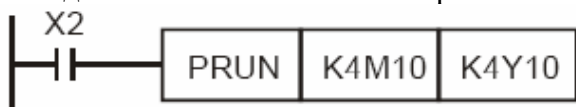
Пример 1: программирование PRUN-инструкции

Когда X3 включено, состояния входов X10-X17 и X20-X27 будут переданы внутренним реле M10-M17 и M20-M27.



Пример 2: программирование PRUN-инструкции

Когда X3 включено, состояния внутренних реле M10-M17 и M20-M27 будут переданы выходам Y10-Y17 и Y20-Y27. При этом состояния M18-M19 передаваться не будут.



API		ASCII		S D n	Преобразование HEX в ASCII	DVP-		
82			P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- еранд	Биты				Слова											16-ти битная инструкция (7 шагов) ASCII - Непрерывное выполнение. ASCIP - Импульс. вып.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*			
n					*	*										
<u>Примечания:</u> Операнд n = 1...256. DVP-ES/EX/SS не поддерживают импульсное выполнение ASCIP																<u>32-х битная инструкция</u> --- Флаги: M1161 (8/16 бит режим выполнения)

Функция

Преобразование шестнадцатеричные значений в ASCII-символы

Описание

ASCII-инструкция позволяет преобразование шестнадцатеричные значений из регистров данных в ASCII-символы.

ASCII-инструкция составляется следующим образом:

- Стартовым адресом (S), где содержатся шестнадцатеричные данные.
- Адресом цели (D), где должны храниться преобразованные ASCII-символы.
- Вводом по (n) числа знаков, т.е. количество шестнадцатеричные цифр, преобразуемых в ASCII-знаки.
- Стартовый адрес

Определенное здесь значение обозначает первый словный операнд, который содержит преобразуемое шестнадцатеричное число. Если нужно преобразовывать больше двух чисел, то считываются следующие словные операнды, пока не будут преобразованы все желаемые цифры.

- Адрес цели (конечный)

Определенное здесь значение обозначает первый словный операнд, который должен содержать ASCII-знаки. Каждый словный операнд может содержать два знака (2 байта). Стартовые адреса следующих словных операндов применяются столько, пока не будут запомнены все знаки.

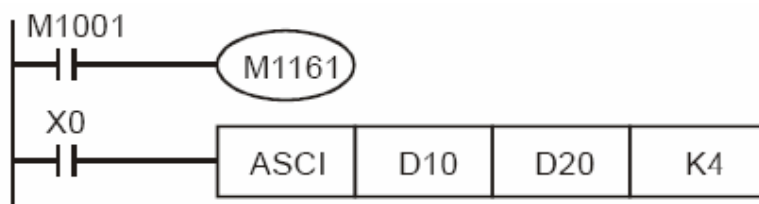
- Количество знаков

Заданное здесь значение может быть только десятичным (K) или шестнадцатеричным (H). Оно означает количество шестнадцатеричных цифр, которые должны быть преобразованы и сколько должно быть запомнено ASCII-знаков. Количество знаков может быть от 1 до 256 цифр

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 специальное реле M1161. При этом при ВКЛ. выбран 8-ми битовый режим.

Инструкция "SET M1161" требуется только тогда, если должен применяться 8-ми битовый режим.

Пример: программирование ASCII-инструкции в 16-ти битовом режиме (M1161=0)

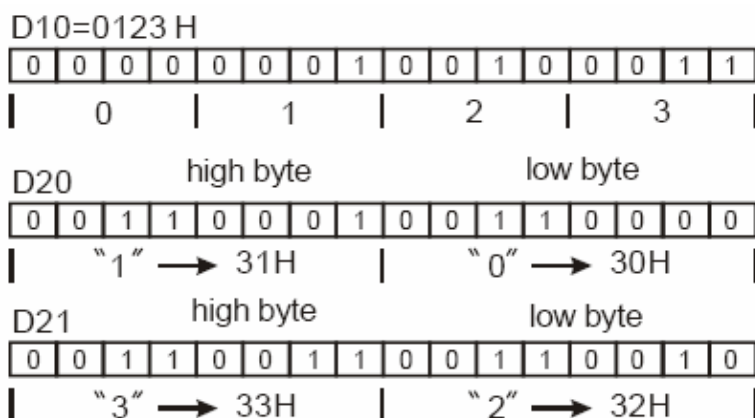


Приведенная программа выполняет преобразование, если вход X0 включен. Во время активного состояния ASCII-инструкции четыре (K4) шестнадцатеричные цифр из регистра данных D10 преобразуются в ASCII-знаки и запишутся в регистры данных со стартовым адресом D20.

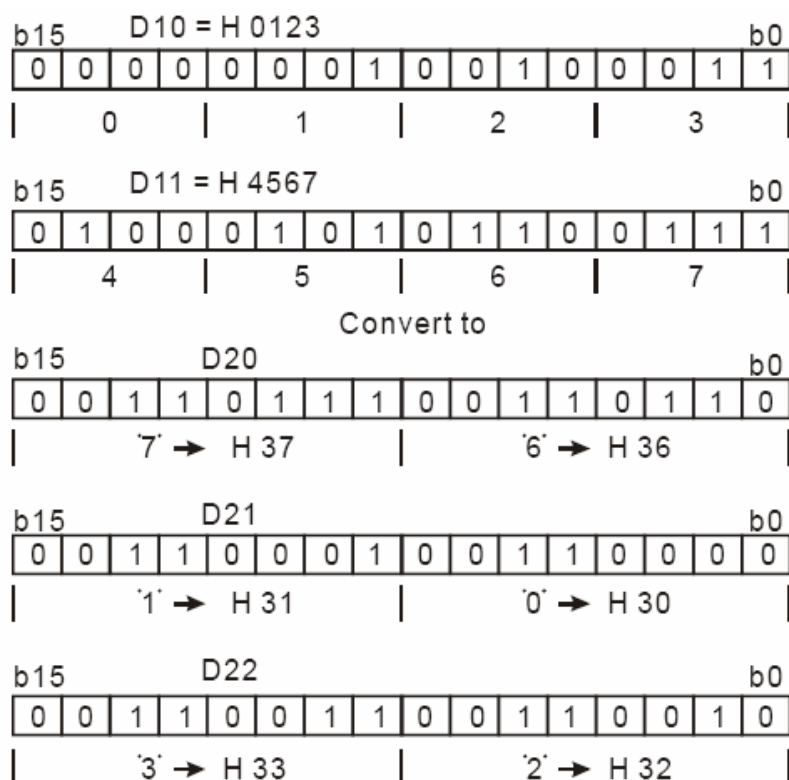
Допустим, что значения регистров следующие:

(D10) = 0123 H	'0' = 30H	'4' = 34H	'8' = 38H
(D11) = 4567 H	'1' = 31H	'5' = 35H	'9' = 39H
(D12) = 89AB H	'2' = 32H	'6' = 36H	'A' = 41H
(D13) = CDEFH	'3' = 33H	'7' = 37H	'B' = 42H

Когда n=4, структура преобразования данных будет следующая:



Когда n=6, структура преобразования данных будет следующая:



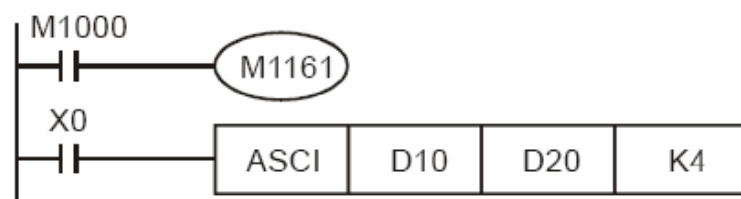
Когда n = 1...16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20 low byte	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D20 high byte		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D21 low byte			"3"	"2"	"1"	"0"	"7"	"6"
D21 high byte				"3"	"2"	"1"	"0"	"7"
D22 low byte					"3"	"2"	"1"	"0"
D22 high byte						"3"	"2"	"1"
D23 low byte							"3"	"2"
D23 high byte								"3"
D24 low byte								
D24 high byte								
D25 low byte								
D25 high byte								
D26 low byte								
D26 high byte								
D27 low byte								
D27 high byte								

no change

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20 low byte	"B"	"A"	"9"	"8"	"F"	"E"	"D"	"C"
D20 high byte	"4"	"B"	"A"	"9"	"8"	"F"	"E"	"D"
D21 low byte	"5"	"4"	"B"	"A"	"9"	"8"	"F"	"E"
D21 high byte	"6"	"5"	"4"	"B"	"A"	"9"	"8"	"F"
D22 low byte	"7"	"6"	"5"	"4"	"B"	"A"	"9"	"8"
D22 high byte	"0"	"7"	"6"	"5"	"4"	"B"	"A"	"9"
D23 low byte	"1"	"0"	"7"	"6"	"5"	"4"	"B"	"A"
D23 high byte	"2"	"1"	"0"	"7"	"6"	"5"	"4"	"B"
D24 low byte	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D24 high byte		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D25 low byte			"3"	"2"	"1"	"0"	"7"	"6"
D25 high byte				"3"	"2"	"1"	"0"	"7"
D26 low byte					"3"	"2"	"1"	"0"
D26 high byte						"3"	"2"	"1"
D27 low byte							"3"	"2"
D27 high byte								"3"

Пример: программирования ASCII-инструкции в 8-ми битовом режиме (M1161=1)



Приведенная программа выполняет преобразование, если вход X0 включен. Во время активного состояния ASCII-инструкции четыре (K4) шестнадцатеричные цифр из регистра данных D10 преобразуются в ASCII-знаки и запишутся в регистры данных со стартовым адресом D20. При этом каждый знак является байтом.

Допустим, что значения регистров следующие:

(D10) = 0123 H	'0' = 30H	'4' = 34H	'8' = 38H
(D11) = 4567 H	'1' = 31H	'5' = 35H	'9' = 39H
(D12) = 89AB H	'2' = 32H	'6' = 36H	'A' = 41H
(D13) = CDEFH	'3' = 33H	'7' = 37H	'B' = 42H

Когда n=2, структура преобразования данных будет следующая:

D10=0123 H

0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1
	0		1		2		3									

ASCII code of D20=2 is 32H

0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
										3		2			

ASCII code of D21=3 is 33H

0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
										3		3			

Когда n=4, структура преобразования данных будет следующая:

b15 D10 = H 0123 b0

0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1
	0		1		2		3									

Convert to

b15	D20										b0					
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	
										'0' → H 30						

b15	D21															b0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
											1' → H 31					

b15	D22															b0
0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	
										2' → H 32						

b15	D23										b0				
0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
										3' → H 33					

Когда n = 1...16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D21		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D22			"3"	"2"	"1"	"0"	"7"	"6"
D23				"3"	"2"	"1"	"0"	"7"
D24					"3"	"2"	"1"	"0"
D25						"3"	"2"	"1"
D26							"3"	"2"
D27								"3"
D28								
D29								
D30								
D31								
D32								
D33								
D34								
D35								

no change

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20	"B"	"A"	"9"	"8"	"F"	"E"	"D"	"C"
D21	"4"	"B"	"A"	"9"	"8"	"F"	"E"	"D"
D22	"5"	"4"	"B"	"A"	"9"	"8"	"F"	"E"
D23	"6"	"5"	"4"	"B"	"A"	"9"	"8"	"F"
D24	"7"	"6"	"5"	"4"	"B"	"A"	"9"	"8"
D25	"0"	"7"	"6"	"5"	"4"	"B"	"A"	"9"
D26	"1"	"0"	"7"	"6"	"5"	"4"	"B"	"A"
D27	"2"	"1"	"0"	"7"	"6"	"5"	"4"	"B"
D28	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D29		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D30			"3"	"2"	"1"	"0"	"7"	"6"
D31				"3"	"2"	"1"	"0"	"7"
D32					"3"	"2"	"1"	"0"
D33						"3"	"2"	"1"
D34							"3"	"2"
D35								"3"

no change

API		HEX		Преобразование ASCII в HEX	DVP-		
83			S D n		ES/EX/SS	SA/SX	EH
					+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (7 шагов) HEX - Непрерывное выполнение. HEXP - Имп. вып.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*			
n					*	*										
<u>Примечания:</u> Операнд n = 1...256. DVP-ES/EX/SS не поддерживают импульсное выполнение HEXP																<u>32-х битная инструкция</u> --- Флаги: M1161 (8/16 бит режим выполнения)

Функция

Преобразование ASCII-знаков в шестнадцатеричные значения

Описание

HEX-инструкция составляется, 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.
- 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

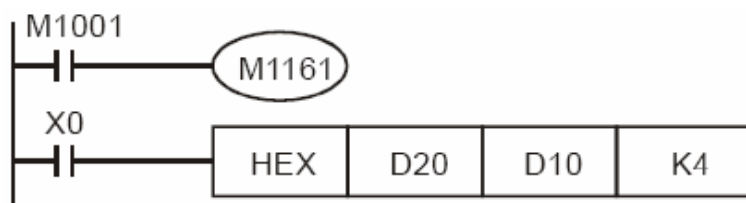
Функция "RST M1161" переключает формат хранения данных на значение по умолчанию в 16 бит.

Функция "RST M1161" 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.

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 специальное реле M1161. Здесь при ВКЛ. выбран 8-ми битовый режим.

Инструкция "SET M1161" требуется только тогда, если должен применяться 8-ми битовый режим.

Пример: программирование HEX-инструкции в 16-ти битовом режиме (M1161=0)

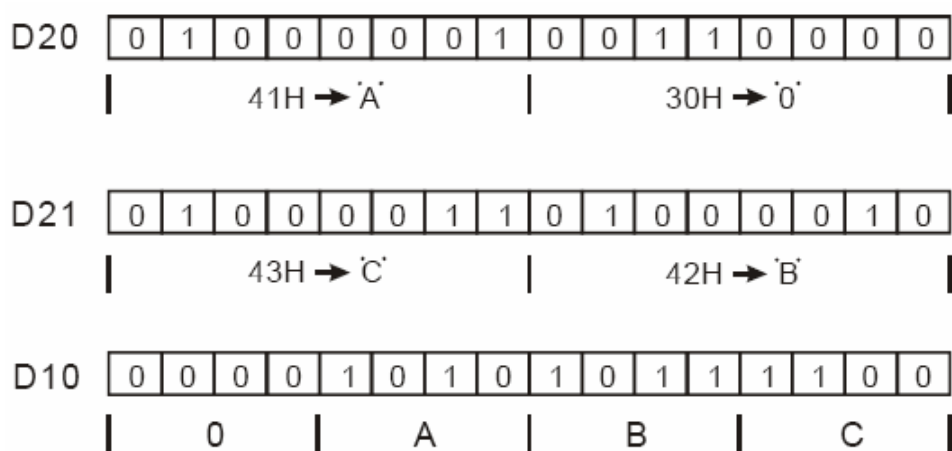


Приведенная программа выполняет преобразование, если вход X0 включен. Во время активного состояния HEX-инструкции четыре (К4) ASCII-знака из регистров данных со стартовым адресом D20 преобразуются в шестнадцатиричные цифры и запишутся в регистры данных со стартовым адресом D10.

Допустим, что значения регистров следующие:

S	ASCII код	HEX	S	ASCII код	HEX
D20 младший байт	H 43	C	D24 младший байт	H 34	4
D20 старший байт	H 44	D	D24 старший байт	H 35	5
D21 младший байт	H 45	E	D25 младший байт	H 36	6
D21 старший байт	H 46	F	D25 старший байт	H 37	7
D22 младший байт	H 38	8	D26 младший байт	H 30	0
D22 старший байт	H 39	9	D26 старший байт	H 31	1
D23 младший байт	H 41	A	D27 младший байт	H 32	2
D23 старший байт	H 42	B	D27 старший байт	H 33	3

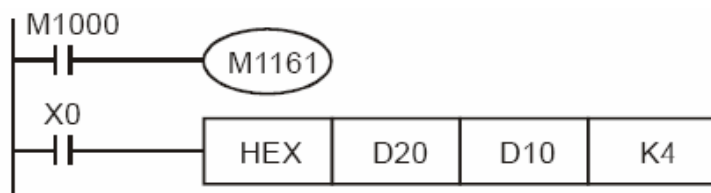
Когда n=4, структура преобразования данных будет следующая:



Когда n = 1...16:

п	D	D13	D12	D11	D10
1	Регистры не определены или равны 0.				***C H
2					**CD H
3					*CDE H
4					CDEF H
5				***C H	DEF8 H
6				**CD H	EF89 H
7				*CDE H	F89A H
8				CDEF H	89AB H
9				***C H	DEF8 H
10				**CD H	EF89 H
11				*CDE H	F89A H
12				CDEF H	89AB H
13		***C H	DEF8 H	9AB4 H	5670 H
14		**CD H	EF89 H	AB45 H	6701 H
15		*CDE H	F89A H	B456 H	7012 H
16		CDEF H	89AB H	4567 H	0123 H

Пример: программирование HEX-инструкции в 8-ми битовом режиме (M1161=1)

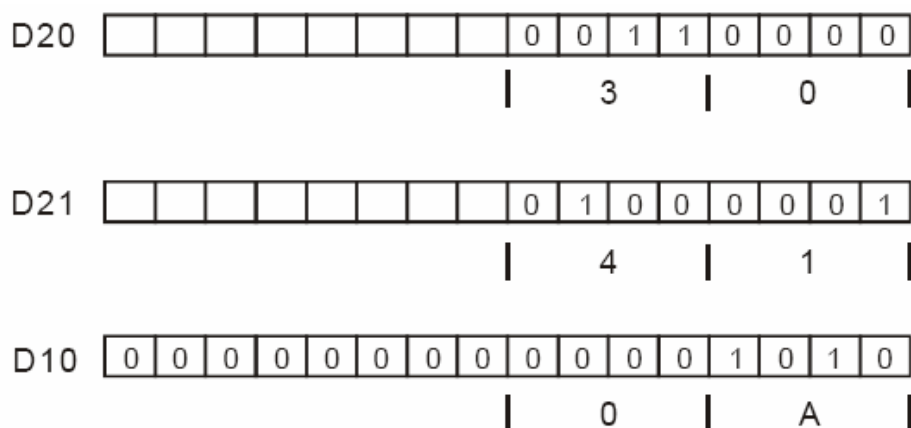


Приведенная программа выполняет преобразование, если вход X0 включен. Во время активного состояния HEX-инструкции четыре (K4) ASCII-знака из регистров данных со стартовым адресом D20 преобразуются в шестнадцатеричные цифры и запишутся в регистры данных со стартовым адресом D10. При этом каждая цифра является байтом.

Допустим, что значения регистров следующие:

S	ASCII код	HEX	S	ASCII код	HEX
D20	H 43	C	D28	H 34	4
D21	H 44	D	D29	H 35	5
D22	H 45	E	D30	H 36	6
D23	H 46	F	D31	H 37	7
D24	H 38	8	D32	H 30	0
D25	H 39	9	D33	H 31	1
D26	H 41	A	D34	H 32	2
D27	H 42	B	D35	H 33	3

Когда n=2, структура преобразования данных будет следующая:



Когда n = 1...16:

n \ D	D13	D12	D11	D10
1	Регистры не определены или равны 0.			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

API	CCD	S D n	Расчет контрольной суммы	DVP-		
				ES/EX/SS	SA/SX	EH
84		P		-	+	+

Оп- еранд	Биты				Слова											16-ти битная инструкция (7 шагов) CCD - Непрерывное выполнение. CCDP - Имп. вып.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			
D									*	*	*	*	*			
n					*	*							*			
<u>Примечания:</u> Операнд n = 1...256.																32-х битная инструкция --- Флаги: M1161 (8/16 бит режим выполнения)

Функция

Расчет контрольной суммы

Описание

CCD-инструкция используется для выполнения расчета контрольной суммы передаваемых по RS-485 интерфейсу данных, что бы исключить ошибки передачи.

CCD-инструкция составляется, 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.

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 специальное реле M1161. Инструкция "SET M1161" требуется только тогда, если должен применяться 8-ми битовый режим.

- 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.

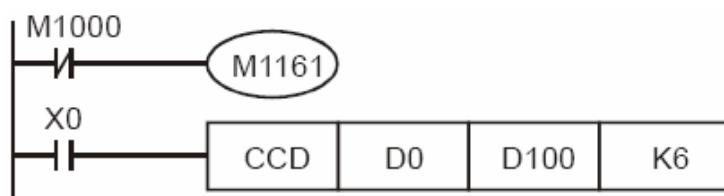
- Адрес цели (конечный)

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.

Пример: программирование CCD-инструкции в 16-ти битовом режиме (M1161=0)



Во время активного состояния CCD-инструкции (X0=1) суммируются шесть (K6) байт данных из регистров данных D0...D2 (16-ми битового формата) и суммируемое значение и проверка паритета сохраняются в регистре данных D100 или D101.

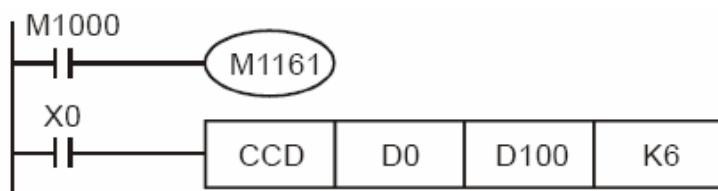
(S)	Значения регистров
D0 младший байт	K100 = 0 1 1 0 0 1 0 0
D0 старший байт	K111 = 0 1 1 0 1 1 1 1 ←
D1 младший байт	K120 = 0 1 1 1 1 0 0 0
D1 старший байт	K202 = 1 1 0 0 1 0 1 0
D2 младший байт	K123 = 0 1 1 1 1 0 1 1 ←
D2 старший байт	K211 = 1 1 0 1 0 0 1 1 ←
D100	K867
D101	0 0 0 1 0 0 0 1 при четном результате индицируется 0, при нечетном - 1.

D100 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1

D101 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 ←паритет

Пример: программирование CCD-инструкции в 8-ми битовом режиме (M1161=1)

Во время активного состояния CCD-инструкции (X0=1) суммируются шесть (K6) байт данных из регистров данных D0...D4 (8-ми битового формата) и суммируемое значение и проверка паритета сохраняются в регистре данных D100 или D101.



(S)	Значения регистров
D0 младший байт	K100 = 0 1 1 0 0 1 0 0
D1 младший байт	K111 = 0 1 1 0 1 1 1 1 ←
D2 младший байт	K120 = 0 1 1 1 1 0 0 0
D3 младший байт	K202 = 1 1 0 0 1 0 1 0
D4 младший байт	K123 = 0 1 1 1 1 0 1 1 ←
D5 младший байт	K211 = 1 1 0 1 0 0 1 1 ←
D100	K867
D101	0 0 0 1 0 0 0 1 при четном результате индицируется 0, при нечетном - 1.

D100 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1

D101 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 ← паритет

API	VRRD			Чтение значения с потенциометра	DVP-		
85		P	S D		ES/EX/SS	SA/SX	EH
					-	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция (5 шагов) VRRD - Непрерывное выполнение. VRRDP - Имп. вып.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										
D								*	*	*	*	*	*	*	*	
<u>Примечания:</u> Операнд S = 0...7.																32-х битная инструкция ---
																Флаги: M1178, M1179

Функция

Чтение значения в 8-ми битном формате, с двух потенциометров (№0 и №1) встроенных в базовых модулях DVP-SA/EH и с шести потенциометров (№2 и №7) функциональной карты DVP-F6VR

Описание

С помощью VRRD-инструкции преобразовывается настройка потенциометра с номером (S) и запоминается в (D) в диапазоне от 0 до 255 (8 бит).

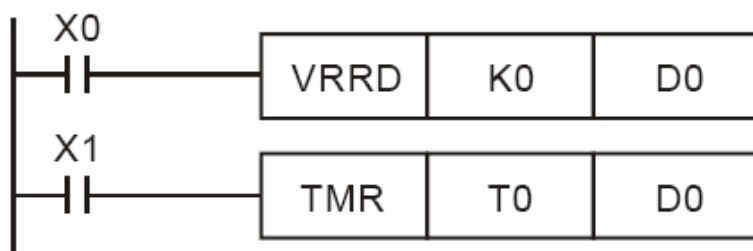
Указание: Потенциометры нумеруются от 0 до 7.

Модуль DVP-F6VR вставляются в слот расширения DVP-EH.

Пример 1:

Когда X0 включен, считываются показания потенциометра №0 и установленное значение запоминается в регистре данных D0.

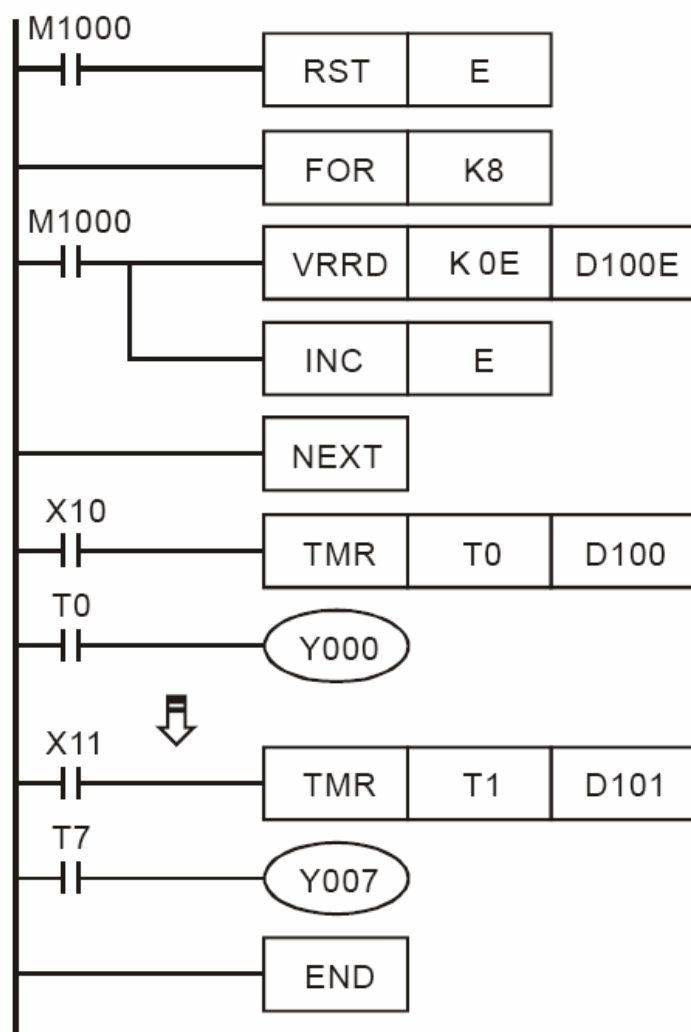
Если X1 включен, значение считываемое с потенциометра №0 является уставкой таймера T0.



Пример 2:

В нижеприведенной программе считываются показания потенциометра №0 - №7 и установленные значения запоминаются в регистрах данных D0 – D7. Считывание происходит в цикле FOR-NEXT

Значения, считываемые с потенциометров №0 - №7 являются уставками таймеров T0-T7.



Примечание: Значения потенциометров встроенных в базовых модулях DVP-SA/EH можно считывать с помощью специальных реле и регистров:

M1178 – старт VR0; M1179 – старт VR1;

D1178 – значение VR0; D1179 – значение VR1

API		VRSC		<div><div>S</div><div>D</div></div>	Чтение положения потенциометра	DVP-		
86			P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (5 шагов) VRSC - Непрерывное выполнение. VRSCP - Имп. вып.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S					*	*										
D								*	*	*	*	*	*	*	*	

Примечания:

Операнд S = 0...7.

32-х битная инструкция

Флаги: нет

Функция

Чтение положения, с двух потенциометров (№0 и №1) встроенных в базовых модулях DVP-SA/EH и с шести потенциометров (№2 и №7) функциональной карты DVP-F6VR

Описание

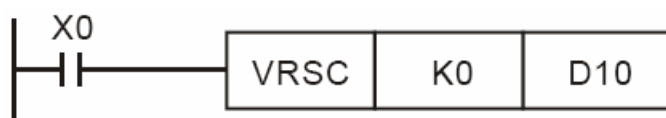
С помощью VRSC-инструкции преобразовывается настройка потенциометра с номером (S) и запоминается в (D) в диапазоне значений от 0 до 10.

Указание: Потенциометры нумеруются от 0 до 7.

Модуль DVP-F6VR вставляются в слот расширения DVP-EH.

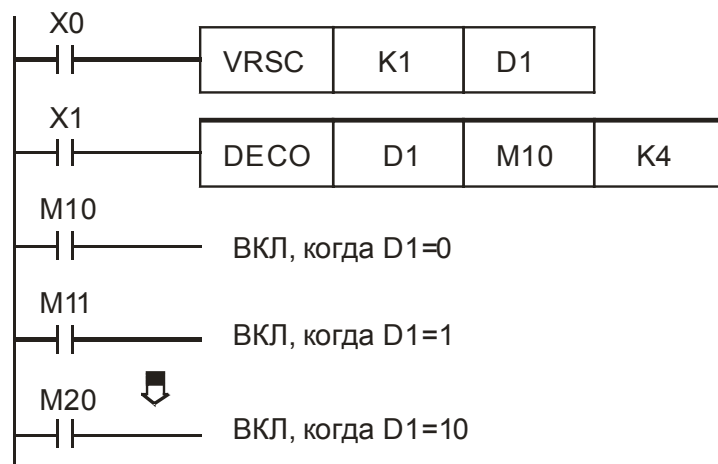
Пример 1:


Когда X0 включен, считываются показания потенциометра №0 (в масштабе 1...10) и установленное значение запоминается в регистре данных D0.



Пример 2:

В нижеприведенной программе потенциометр №0 используется в качестве цифрового переключателя на 10 положений. В зависимости от положения потенциометра включается соответствующее внутреннее реле M10-M20.



API		ABS	☺		Абсолютное значение	DVP-		
87	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (3 шагов) ABS - Непрерывное выполнение. ABSP - Имп. вып.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
D					*	*	*	*	*	*	*	*	*	*	*
<div><div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div></div></div>															

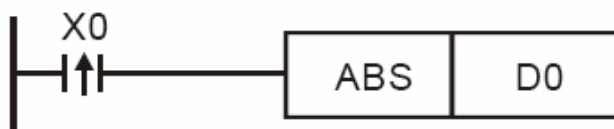
Функция

Нахождение модуля числа и запись его абсолютного значения.

Описание

Если значение в (D) имело отрицательное значение, то после выполнение ABS-инструкции (D) знак "-" отбрасывается и число становится положительным. Если (D) имело положительное значение, то никаких изменений не происходит.

Указание: Рекомендуется выполнять команду в импульсном режиме.



API		PID		   	ПИД-регулятор	DVP-		
88	D					ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (9 шагов) PID - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S ₁													*			32-х битная инструкция DPID - Непрерывное выполнение. Флаги: нет
S ₂													*			
S ₃													*			
D													*			
<u>Примечания:</u> Операнд S ₃ использует 14 последовательных адресов. В серии ES данная инструкция поддерживается только в версии 5.7 и выше.																

Функция

Программирование замкнутого контура регулирования при задании 15-ти параметров регулирования.

Описание

- С помощью PID-инструкции можно проследить регулирование текущего значения в замкнутом контуре регулирования.

- (S1) указывает заданное значение контура регулирования (SV).
- (S2) считывает действительное значение как обратную связь для регулятора (PV).
- (S3) является начальным адресом области регистра, в которой запоминаются параметры контура регулирования. Для этого имеется 15 один за другим следующих регистров данных для команды PID (16 бит) и 21 один за другим следующих регистров данных для команды DPID (32 бит). Таблица содержит обзор параметров.
- В (D) записываются расчетные выходные значения, которые выдаются по процессу регулирования (MV).
- В программе PID-инструкцию можно применять неограниченное число раз, однако адреса регистров операнда (S) не должны повторяться.

Пример программирования PID-инструкции

PID-инструкция начнет выполняться, как только включится X0. Задаваемое значение записано в D0, текущее значение считывается в D1, выходные значения записываются в D150. Параметры регулирования хранятся в регистрах данных D100...D114.

Если X0 выключен, PID-инструкция выполняться не будет и значения в регистрах останутся неизменными.

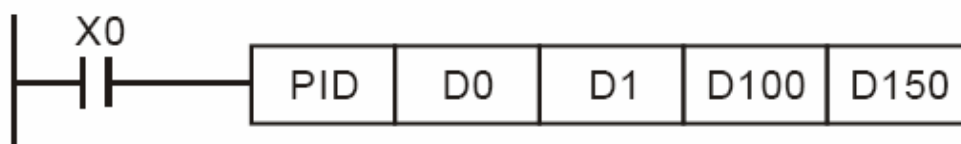


Таблица параметров регулирования (S3) для команды PID (16 бит):

Адрес регистра	Функция	Значения	Описание
(S3)	Время выборки (Ts)	1~2000 (ед.изм: 10мс)	Интервал проверки процесса изменения действительного значения. Если Ts меньше времени скана, PID-инструкция будет выполняться в каждом программном цикле. Если Ts = 0, PID-инструкция не будет выполняться.
(S3) + 1	Коэффициент пропорциональности (Kp)	0~30000(%)	Коэффициент умножения для пропорционального регулирования. При значениях больше 30000%, Kp = 30000%
(S3) + 2	Интегральный коэффициент (Ki)	0~30000(%)	Коэффициент для умножения обратных значений при интегральном регулировании. Выбор значения нуля препятствует интегральному регулированию. При значениях больше 30000%, Ki = 30000%
(S3) + 3	Дифференциальный коэффициент (Kd)	0~30000(%)	Коэффициент умножения дифференциального регулирования. Выбор значения нуля препятствует дифференциальному. При значениях больше 30000%, Kd = 30000%
(S3) + 4	Метод регулирования (Dir)	0: нормальное 1: прямое (E=SV-PV) 2: обратное (E=PV-SV)	
(S3) + 5	Зона нечувствительности	0~32767	Определяется минимальное рассогласование (E) при котором выход не работает. Пример: если зона нечувствительности = 5, то при -5 < E < 5, выходное значение MV = 0

(S3) + 6	Верхнее ограничение выхода MV.	-32768~32767	Определяется максимальное выходное значение (MV). Пример: параметр = 1000, то при $MV > 1000$, выходное значение $MV = 1000$. $((S3) + 6)$ должно быть больше чем $((S3) + 7)$
(S3) + 7	Нижнее ограничение выхода MV	-32768~32767	Определяется минимальное выходное значение (MV). Пример: параметр = -500, то при $MV < -500$, выходное значение $MV = 500$.
(S3) + 8	Верхнее ограничение интегрирования	-32768~32767	Определяется максимальное значение интегрирования. Пример: параметр = 1000, то при $MV > 1000$, выходное значение $MV = 1000$ и интегрирование не выполняется. $((S3) + 8)$ должно быть больше чем $((S3) + 9)$
(S3) + 9	Нижнее ограничение интегрирования	-32768~32767	Определяется минимальное значение интегрирования. Пример: параметр = 500, то при $MV > -500$, выходное значение $MV = 500$ и интегрирование не выполняется.
(S3) + 10, 11	Накопленное временное значение интегрирования	32-бит число с плавающей точкой	Это промежуточное значение. Может использоваться для последующей обработки.
(S3) + 12	Значение предыдущего PV	-	Это промежуточное значение. Может использоваться для последующей обработки.
(S3) + 13 (S3) + 14	Используются системой. Не используйте их!		

- Количество PID-инструкций в программе неограниченно. Однако нужно следить за тем, чтобы не была пересечена область данных (S3), чтобы избежать конфликтов данных внутри управления.
- PID-инструкция может включаться в подпрограммы обработки прерывания, в STL-программу или в программу перехода (CJ).
- Интервал времени выборки (T_s) должен быть выбран больше, чем время цикла программы, так как иначе это приведет к ошибке с кодом K6740 (ошибка ПИД-регулирования). Если это не выполнить, время проверки автоматически станет равным времени цикла. При использовании инструкции прерывания от I6XX до I8XX время выборки должно быть не меньше, чем время цикла прерывания.
- Интервал времени выборки (T_s) может варьироваться на основе скана программы. Максимальная область отклонения лежит в пределах от (T_s - (время цикла программы)) до (T_s + (время цикла программы)). Это отклонение может минимизироваться благодаря введению PID-инструкции внутрь тактируемой программы прерывания.
- Модули аналогового ввода типа DVP-04AD / DVP-04XA / DVP-04PT / DVP-04TC, преобразуют аналоговые данные в цифровые, т.е. оцифровываются аналоговые данные процесса. Для стабильной работы ПИД-регулятора внимательно выставьте время преобразования сигналов в указанных модулях. Базовое устройство (ПЛК) применяет FROM-инструкцию, чтобы занести эти данные в (S2). ПЛК серии ES не имеют аналоговых входов/выходов, поэтому необходимо использовать RS-485 порт.
- Параметры регулирования зависят от специфических данных системы регулирования.

Таблица параметров регулирования (S3) для команды DPID (32 бит):

Адрес регистра	Функция	Значения	Описание
(S3)	Время выборки (T_s)	1~2000 (ед.изм: 10мс)	Интервал проверки процесса изменения действительного значения. Если T_s меньше времени скана, PID-инструкция будет выполняться в каждом программном цикле. Если $T_s=0$, PID-инструкция не будет выполняться.
(S3) + 1	Коэффициент пропорциональности (K_P)	0~30000(%)	Коэффициент умножения для пропорционального регулирования. При значениях больше 30000%, $K_P = 30000\%$
(S3) + 2	Интегральный коэффициент (K_I)	0~30000(%)	Коэффициент для умножения обратных значений при интегральном регулировании. Выбор значения нуля препятствует интегральному регулированию. При значениях больше 30000%, $K_I = 30000\%$
(S3) + 3	Дифференциальный коэффициент (K_D)	0~30000(%)	Коэффициент умножения дифференциального регулирования. Выбор значения нуля препятствует дифференциальному. При значениях больше 30000%, $K_D = 30000\%$
(S3) + 4	Метод регулирования (Dir)	0: нормальное 1: прямое ($E=SV-PV$) 2: обратное ($E=PV-SV$)	
(S3) + 5, 6	Зона нечувствительности	0~2147483647	Определяется минимальное рассогласование (E) при котором выход не работает. Пример: если зона нечувствительности = 5, то при $-5 < E < 5$, выходное значение MV = 0
(S3) + 7, 8	Верхнее ограничение выхода MV.	-2,147,483,648~ 2,147,483,647	Определяется максимальное выходное значение (MV). Пример: параметр = 1000, то при $MV > 1000$, выходное значение MV = 1000. ((S3) + 7, 8) должно быть больше чем ((S3) + 9, 10)
(S3) + 9, 10	Нижнее ограничение выхода MV	-2,147,483,648~ 2,147,483,647	Определяется минимальное выходное значение (MV). Пример: параметр = -500, то при $MV < -500$, выходное значение MV = 500.
(S3) + 11, 12	Верхнее ограничение интегрирования	-2,147,483,648~ 2,147,483,647	Определяется максимальное значение интегрирования. Пример: параметр = 1000, то при $MV > 1000$, выходное значение MV = 1000 и интегрирование не выполняется. ((S3) + 11, 12) должно быть больше чем ((S3) + 13, 14)
(S3) + 13, 14	Нижнее ограничение интегрирования	-2,147,483,648~ 2,147,483,647	Определяется минимальное значение интегрирования. Пример: параметр = 500, то при $MV > -500$, выходное значение MV = 500 и интегрирование не выполняется.
(S3) + 15, 16	Накопленное временное значение интегрирования	32-бит число с плавающей точкой	Это промежуточное значение. Может использоваться для последующей обработки.
(S3) + 17, 18	Значение предыдущего PV	-	Это промежуточное значение. Может использоваться для последующей обработки.
(S3) + 19 (S3) + 20	Используются системой. Не используйте их!		

Уравнение ПИД-регулятора:

PID-инструкция работает со следующей математической формуле для расчета выходных значений:

$$MV = K_P * E(t) + K_I * E(t) \frac{1}{S} + K_D * PV(t)S$$

где:

MV – выходное значение ПИД-регулятора

K_P – коэффициент пропорциональности

$E(t)$ – отклонение: $E(t) = SV - PV$ (при прямом регулировании); $E(t) = PV - SV$ (при инверсном регулировании)

PV – текущее измеренное значение (сигнал обратной связи)

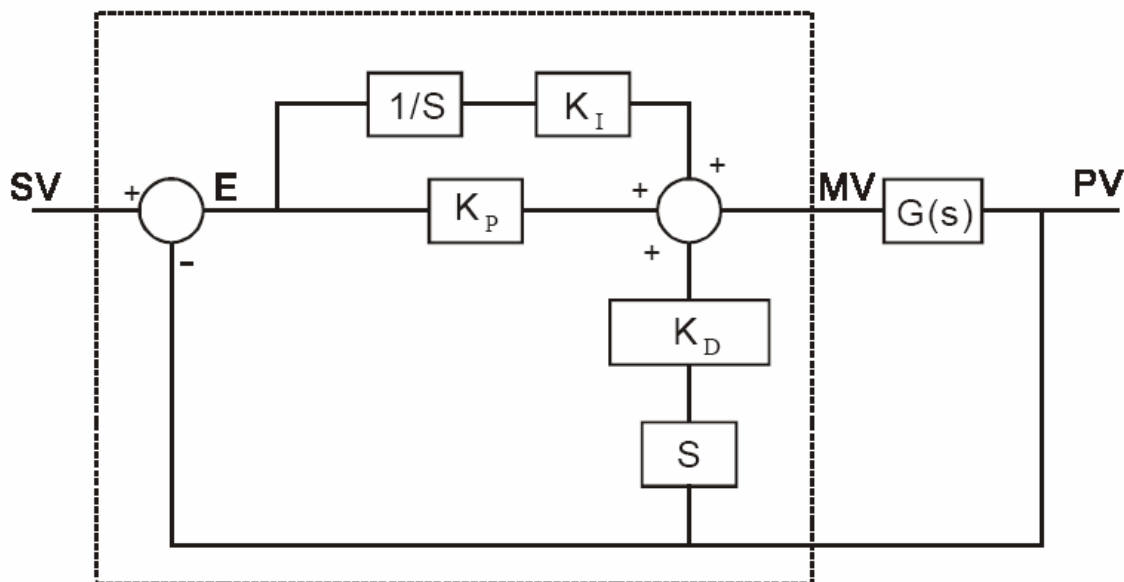
SV – заданное значение

K_D – дифференциальный коэффициент

$PV(t)S$ – значение дифференцирования $PV(t)$

K_I – интегральный коэффициент

$E(t)I/S$ – значение интегрирования $E(t)$

Схема ПИД-регулятора:

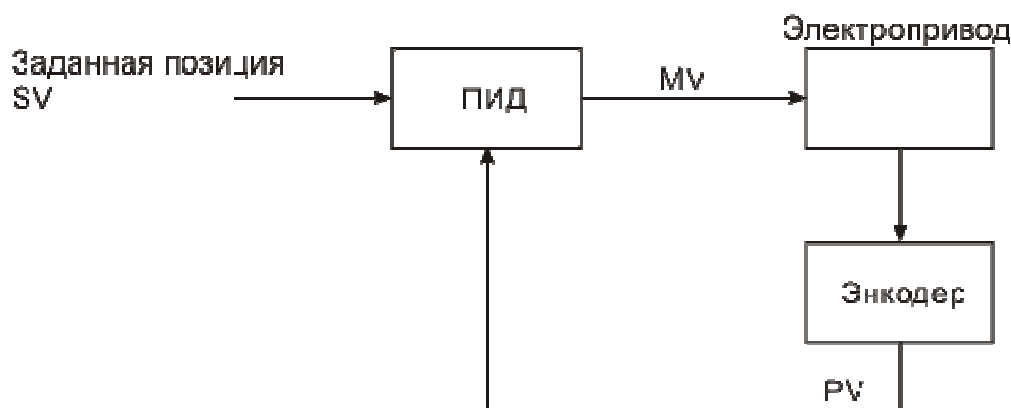
Пунктирной линией выделена PID-инструкция.

Замечания по настройке:

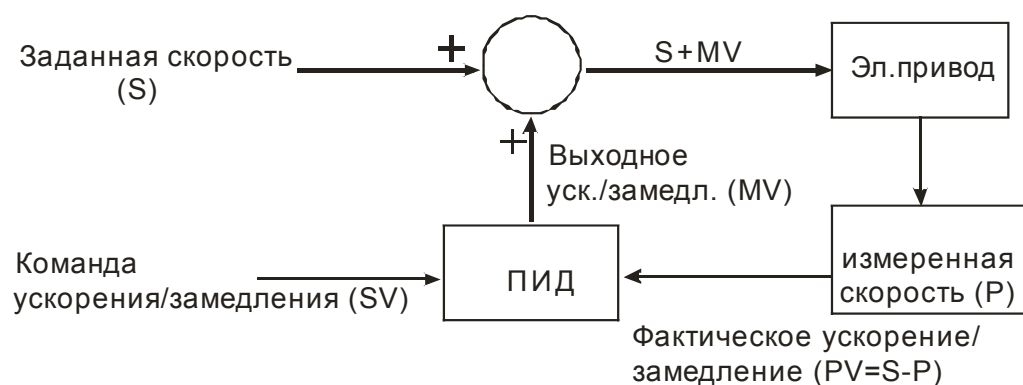
- При настройке основных параметров (K_P , K_I , K_D) ПИД-регулятора всегда начинайте с корректировки пропорционального коэффициента K_P , а K_I , K_D установите равными 0. Далее корректируйте интегральный коэффициент K_I (начинайте с небольшого значения постепенно его увеличивая) и дифференциальный коэффициент K_D (начинайте с небольшого значения постепенно его увеличивая). См. пример 4. Когда

$K_p < 100\%$, значение отклонения $E(t)$ будет уменьшаться, когда $K_p > 100\%$, значение отклонения $E(t)$ будет усиливаться.

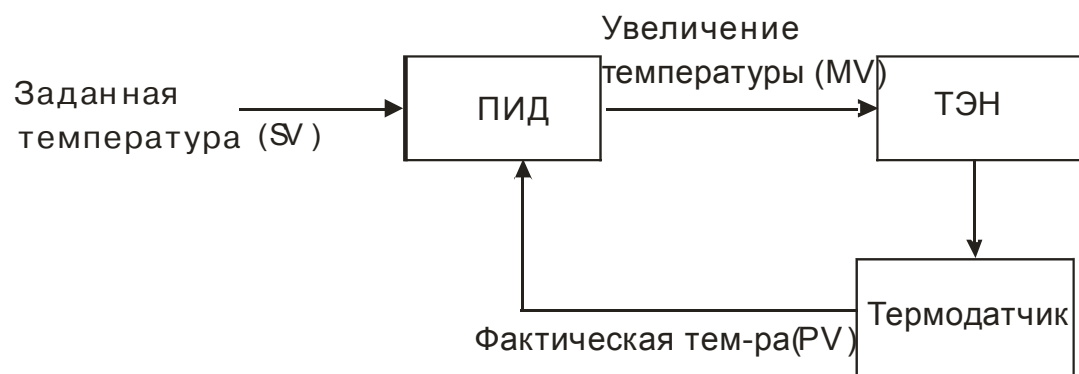
Пример 1: Блок-диаграмма для использования ПИД-регулятора для управления позиционированием.



Пример 2: Блок-диаграмма для использования ПИД-регулятора для управления скоростью ((S3) + 4 должно быть = 0).



Пример 3: Блок-диаграмма для использования ПИД-регулятора для управления температурой ((S3) + 4 должно быть = 1).

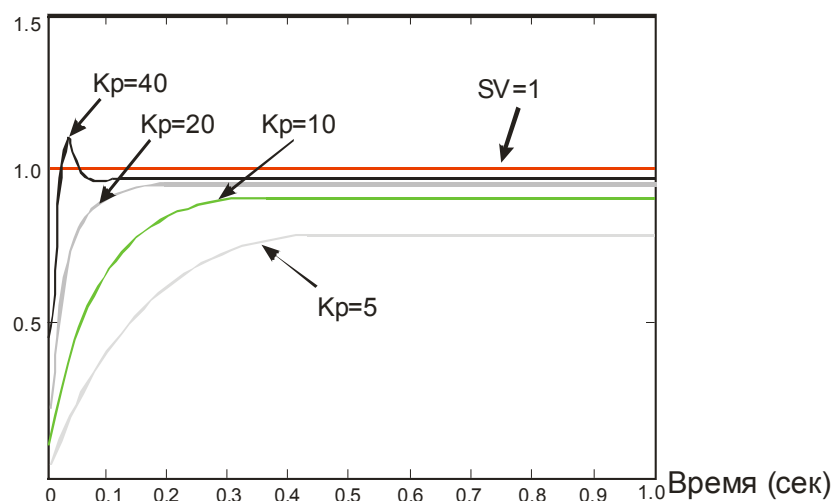


Пример 4: Последовательность настройки ПИД-регулятора

Рассмотрим систему регулирования, описываемую уравнением $G(s) = \frac{b}{s+a}$ (функция подходит для большинства моделей электропривода переменного тока).

Заданное значение $SV=1$, время выборки $T_s = 10\text{мс}$.

Шаг 1. Установим K_I и K_D равными 0, а K_P поочередно установим равным 5, 10, 20, 40 и запишем при этом состояния PV.



Шаг 2. Из приведенных кривых выберем кривую, полученную при $K_P=10$ на основании следующих рассуждений:

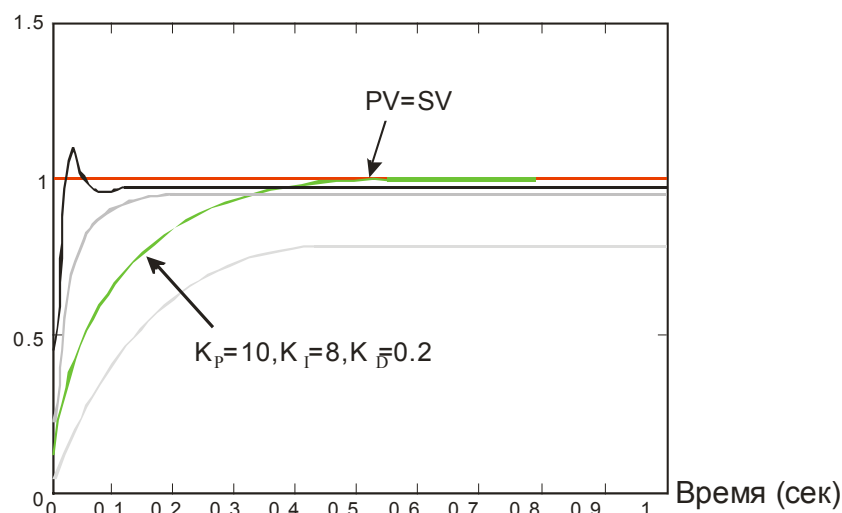
при $K_P=40$ система имеет малое время отклика и как следствие перерегулирование;

при $K_P=20$ система также имеет быстрый отклик, но без перерегулирования, однако кратковременные импульсы MV будут приводить к частым стартам/стоповым режимам и пусковым нагрузкам;

при $K_P=10$ система имеет достаточно быстрый и плавный отклик;

при $K_P=5$ система имеет очень медленный отклик и большое остаточное отклонение;

Шаг 3. При выбранном $K_P=10$ проведем корректировку K_I и K_D . Сначала будем плавно увеличивать значения K_I (1, 2, 4, 8), но не больше чем K_P . Затем будем плавно увеличивать значения K_D (0.01, 0.05, 0.1, 0.2), но не больше чем 10% от K_P . В итоге мы должны получить систему с быстрым и плавным откликом без перерегулирования и с минимальным остаточным отклонением.



Примечание: В этом примере только показан порядок настройки параметров ПИД-регулятора. В каждой реальной системе регулирования параметры ПИД-регулятора надо подбирать индивидуально.

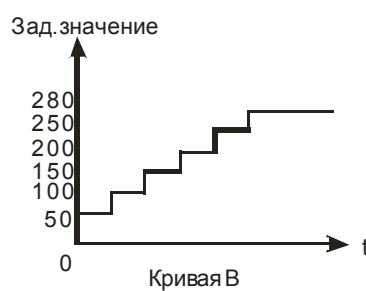
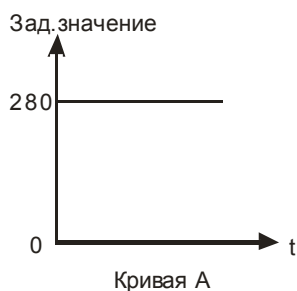
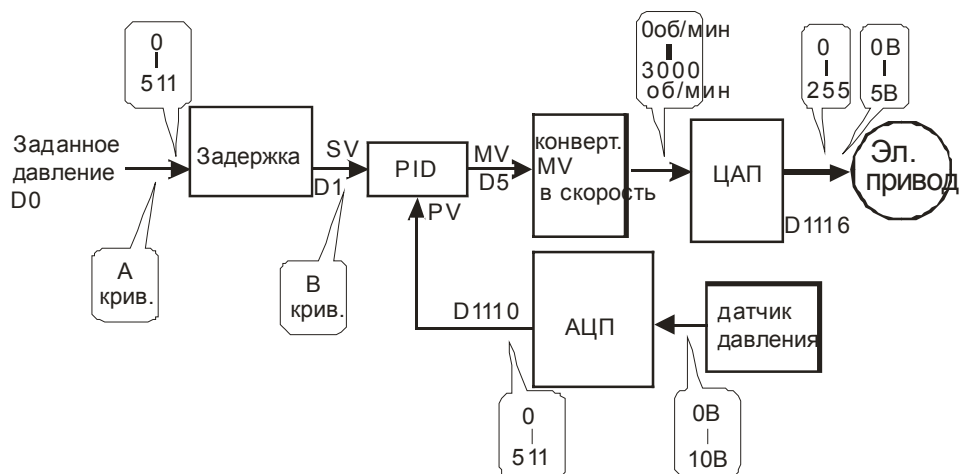
Пример применения 1: Использование ПИД-регулятора для системы регулирования давления (блок-диаграмма примера 1).

Характеристики системы: заданное значение должно достигаться ступенчато для избежания перегрузок при быстрой реакции.

Рекомендуемые методы решения:

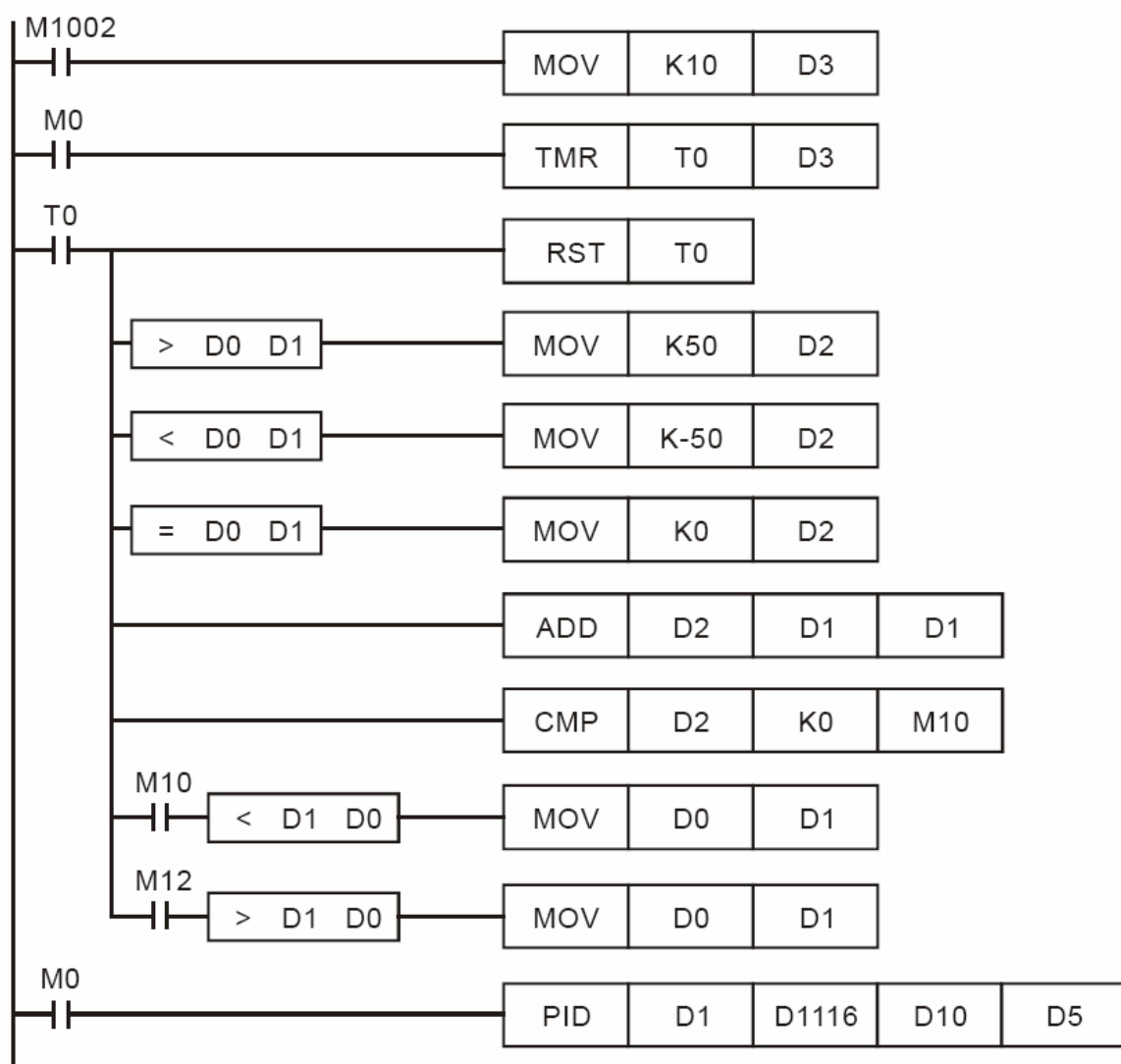
Метод 1: использовать большое время выборки.

Метод 2: использовать команду задержки и блок-диаграмму показанную ниже.



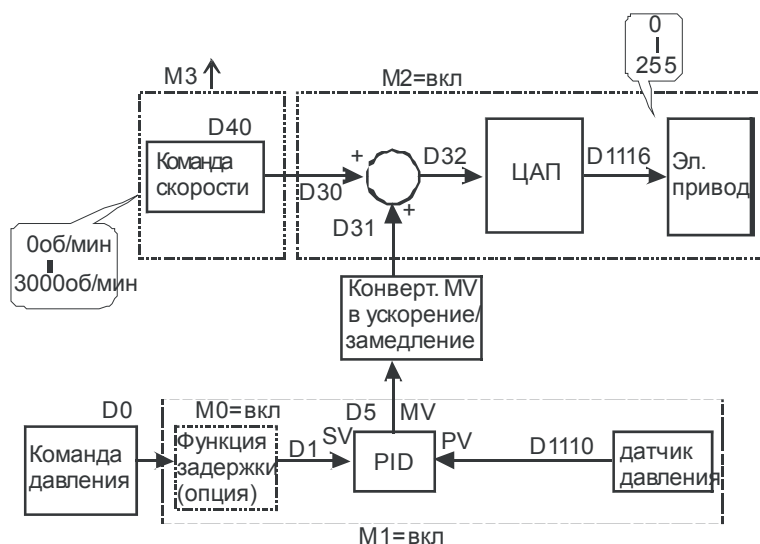
D2 - шаг ступени
D3 - время ступеньки

Программа, реализующая команду задержки:

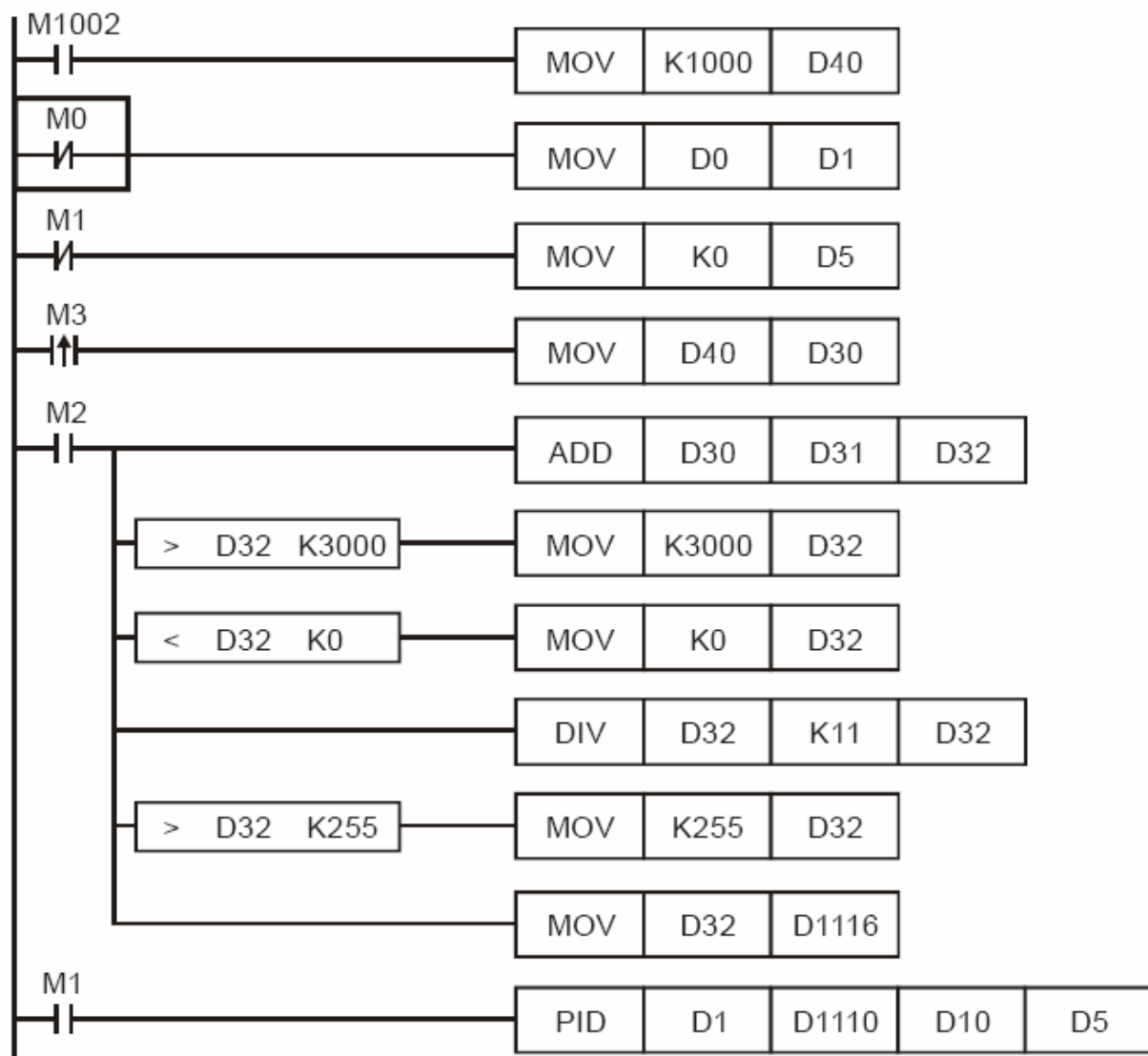


Пример применения 2: Использование ПИД-регулятора для одновременного управления скоростью и давлением (блок-диаграмма примера 2).

Система имеет разомкнутый контур скорости и замкнутый контур давления. Так же как в предыдущем примере может быть использована функция задержки.



Программа:



8. ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 100-149

API		MODRD		<div>S₁</div> <div>S₂</div> <div>n</div>	Чтение данных MODBUS	DVP-		
100						ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова												16-ти битная инструкция (7 шагов) MODRD - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
	S ₁				*	*							*				
	S ₂				*	*							*				
	n				*	*							*				

Примечания:

Диапазоны операндов: S₁ = K0...K256
K1 < n ≤ K6

32-х битная инструкция

Флаги:
M1120 – M1131,
M1140 – M1143 (см.
инструкцию RS)

Функция

Чтение данных от внешних устройств, поддерживающих протокол MODBUS (ASCII или RTU) по интерфейсу RS-485.

Описание

С помощью MODRD-инструкции возможно чтение данных от преобразователей частоты серии VFD (за исключение VFD-A). См. так же инструкции на преобразователи VFD.

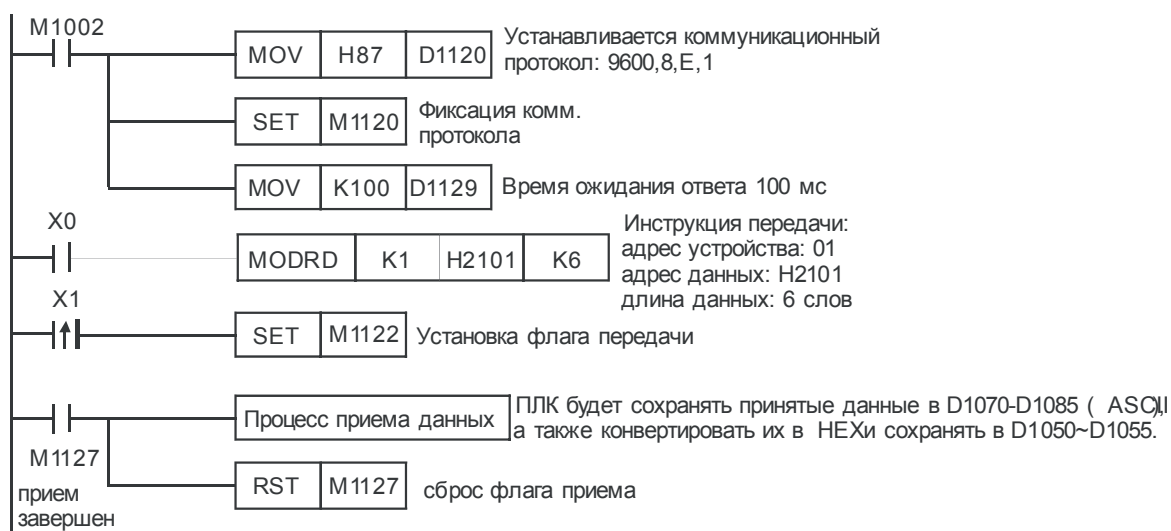
(S_1) – коммуникационный адрес устройства в сети

(S_2) – адрес считываемых данных

(n) – длина считываемых данных

- Если адрес данных установлен неверно, то будет записан соответствующий код ошибки в спец. регистре D1130, пока M1141 будет включено.
- Нельзя использовать в качестве условия выполнения MODRD-инструкции импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF). Иначе данные будут приняты не корректно.
- Принятые данные запоминаются в буферной области приема: регистрах D1070 – D1085, и проверяются на корректность. Если данные приняты с ошибками, включается специальное реле M1140.
- Если используется ASCII-режим, принятые данные будут автоматически конвертироваться в HEX-значения и сохраняться в регистрах D1050 – D1055. В RTU-режиме регистры D1050 – D1055 использоваться не будут.
- После M1140 или M1141 = 1, запрос на чтение данных будет отправлен повторно и, если данные будут приняты без ошибок, флаги M1140 и M1141 будут сброшены.

Пример применения 1: связь ПЛК с преобразователем частоты VFD-S (ASCII-режим, M1143 = 0)



Передаваемое сообщение (ПЛК→VFD-S): "01 03 2101 0006 D4"

Получаемое сообщ-е (VFD-S→ ПЛК): "01 03 0C 0100 1766 0000 0000 0136 0000 3B"

Передаваемые данные, расписанные по регистрам:

Регистр	Данные			
D1089 младший байт	0	30 Н	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1089 старший байт	1	31 Н	ADR0	
D1090 младший байт	0	30 Н	CMD1	CMD (1, 0) – код команды
D1090 старший байт	3	33 Н	CMD0	
D1091 младший байт	2	32 Н	Стартовый адрес данных	
D1091 старший байт	1	31 Н		
D1092 младший байт	0	30 Н		
D1092 старший байт	1	31 Н		
D1093 младший байт	0	30 Н	Число данных (слов)	
D1093 старший байт	0	30 Н		
D1094 младший байт	0	30 Н		
D1094 старший байт	6	36 Н		
D1095 младший байт	D	44 Н	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1095 старший байт	4	34 Н	LRC CHK0	

Получаемые данные, расписанные по регистрам:

Регистр	Данные			
D1070 младший байт	0	30 H	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1070 старший байт	1	31 H	ADR0	
D1071 младший байт	0	30 H	CMD1	CMD (1, 0) – код команды
D1071 старший байт	3	33 H	CMD0	
D1072 младший байт	0	30 H	Число данных (байт)	
D1072 старший байт	C	43 H		
D1073 младший байт	0	30 H	Содержимое регистра с адресом 2101H	D1050 = 0100 H
D1073 старший байт	1	31 H		
D1074 младший байт	0	30 H		
D1074 старший байт	0	30 H		
D1075 младший байт	1	31 H	Содержимое регистра с адресом 2102H	D1051 = 1766 H
D1075 старший байт	7	37 H		
D1076 младший байт	6	36 H		
D1076 старший байт	6	36 H		
D1077 младший байт	0	30 H	Содержимое регистра с адресом 2103H	D1052 = 0000 H
D1077 старший байт	0	30 H		

D1078 младший байт	0	30 H	Содержимое регистра с адресом 2104H	D1053 = 0000 H
D1078 старший байт	0	30 H		
D1079 младший байт	0	30 H		
D1079 старший байт	0	30 H		
D1080 младший байт	0	30 H	Содержимое регистра с адресом 2105H	D1054 = 0136 H
D1080 старший байт	0	30 H		
D1081 младший байт	0	30 H		
D1081 старший байт	1	31 H		
D1082 младший байт	3	33 H	Содержимое регистра с адресом 2106H	D1055 = 0000 H
D1082 старший байт	6	36 H		
D1083 младший байт	0	30 H		
D1083 старший байт	0	30 H		
D1084 младший байт	0	30 H	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1084 старший байт	0	30 H		
D1085 младший байт	3	33 H	LRC CHK0	
D1085 старший байт	B	42 H		

Пример применения 2: связь ПЛК с преобразователем частоты VFD-S (RTU-режим, M1143 = 1)



Передаваемое сообщение (ПЛК→VFD-S): "01 03 2102 0002 6F F7"

Принимаемое сообщ-е (VFD-S→ ПЛК): "01 03 04 1700 0000 FE 5C"

Передаваемые данные, расписанные по регистрам:

Регистр	Данные	
D1089 младший байт	01 H	Коммуникационный адрес VFD-S
D1090 младший байт	03 H	Адрес команды
D1091 младший байт	21 H	Адрес данных
D1092 младший байт	02 H	
D1093 младший байт	00 H	Данные
D1094 младший байт	02 H	
D1095 младший байт	6F H	CRC CHK Low
D1096 младший байт	F7 H	CRC CHK High

Принимаемые данные, расписанные по регистрам:

Регистр	Данные	
D1070 младший байт	01 H	Коммуникационный адрес VFD-S
D1071 младший байт	03 H	Адрес команды
D1072 младший байт	04 H	Число данных (байт)
D1073 младший байт	17 H	Содержимое регистра с адресом 2102H
D1074 младший байт	70 H	
D1075 младший байт	00 H	Содержимое регистра с адресом 2103H
D1076 младший байт	00 H	
D1077 младший байт	FE H	CRC CHK Low
D1078 младший байт	5C H	CRC CHK High

Пример применения 3: связь ПЛК с преобразователем частоты VFD-S (ASCII-режим, M1143 = 0). Повторная передача при превышении времени ожидания ответа.

Когда X0 включено, данные читаются со стартового адреса H2100 и сохраняются в регистрах D1070 – D1085, а также конвертироваться в HEX и сохраняться в регистрах D1050 – D1055.

Флаг M1129 будет включен, если время ожидания ответа от VFD-S превысит 100 мс. После чего здесь M1129 будет инициализировать повторную передачу данных, включив M1122.

Флаг M1140 будет включен, если принятые данные будут содержать ошибки. После чего здесь M1140 будет инициализировать повторную передачу данных, включив M1122.

Флаг M1141 будет включен, если адрес данных будет задан некорректно. После чего здесь M1141 будет инициализировать повторную передачу данных, включив M1122.



API		MODWR		<div><div>S₁</div><div>S₂</div><div>n</div></div>			Запись данных MODBUS	DVP-					
101				ES/EX/SS	SA/SX	EH							
				+	+	+							

Оп- еранд	Биты				Слова										16-ти битная инструкция (7 шагов) MODWR - Непрерывное выполнение.			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E				F
	S ₁					*	*						*					
	S ₂					*	*						*					
n					*	*							*			32-х битная инструкция ---		
<u>Примечания:</u> Диапазоны операндов: S ₁ = K0...K256																		

Флаги: M1120 – M1131, M1140 – M1143 (см. инструкцию RS)		
--	--	--

Функция

Запись данных во внешние устройства, поддерживающие протокол MODBUS (ASCII или RTU) по интерфейсу RS-485.

Описание

С помощью MODWR-инструкции возможна запись данных в преобразователи частоты серии VFD (за исключение VFD-A). См. так же инструкции на преобразователи VFD.

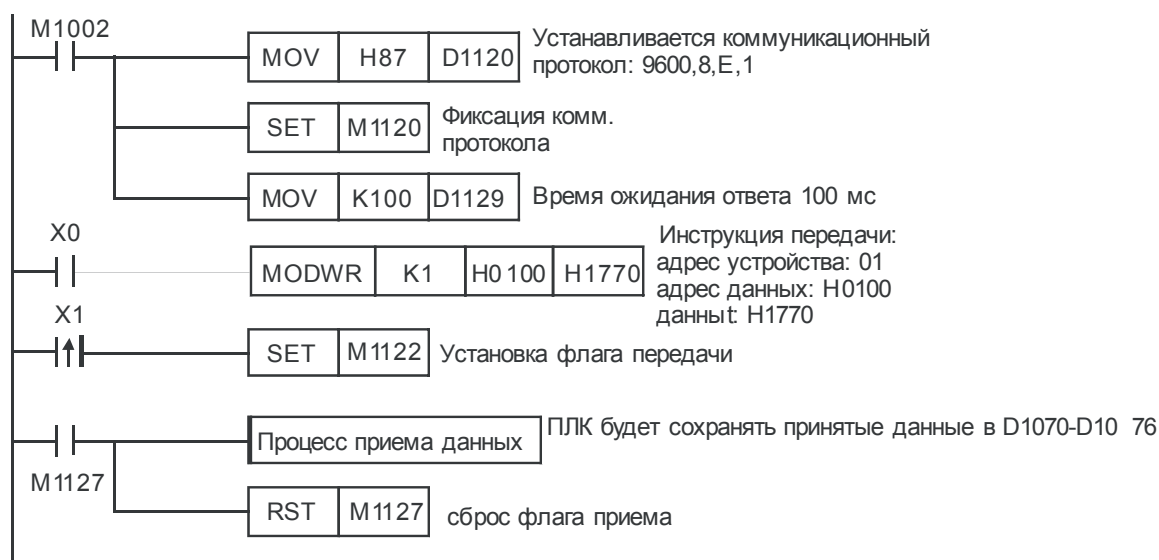
(S₁) – коммуникационный адрес устройства в сети

(S₂) – адрес записи данных

(n) – данные

- Если адрес данных установлен неверно, то будет записан соответствующий код ошибки в спец. регистре D1130, пока M1141 будет включено.
- Не используйте в качестве условия выполнения MODWR-инструкции импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF).
- Принятый ответ запоминаются в буферной области приема: регистрах D1070 – D1076, и проверяется на корректность. Если обнаружены ошибки, включается специальное реле M1140.
- Если используется ASCII-режим, принятые данные будут автоматически конвертироваться в HEX-значения и сохраняться в регистрах D1050 – D1055. В RTU-режиме регистры D1050 – D1055 использоваться не будут.
- После M1140 или M1141 = 1, данные будут отправлены повторно и, если не будет ошибок, флаги M1140 и M1141 будут сброшены.

Пример применения 1: связь ПЛК с преобразователем частоты VFD-S (ASCII-режим, M1143 = 0)



Передаваемое сообщение (ПЛК→VFD-S): "01 06 0100 1770 71"

Принимаемое сообщ-е (VFD-S→ ПЛК): "01 06 0100 1770 71"

Передаваемые данные, расписанные по регистрам:

Регистр	Данные			
D1089 младший байт	0	30 Н	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1089 старший байт	1	31 Н	ADR0	
D1090 младший байт	0	30 Н	CMD1	CMD (1, 0) – код команды
D1090 старший байт	6	36 Н	CMD0	
D1091 младший байт	0	30 Н	Адрес данных	
D1091 старший байт	1	31 Н		
D1092 младший байт	0	30 Н		
D1092 старший байт	0	30 Н		
D1093 младший байт	1	31 Н	Данные	
D1093 старший байт	7	37 Н		
D1094 младший байт	7	37 Н		
D1094 старший байт	0	30 Н		
D1095 младший байт	7	37 Н	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1095 старший байт	1	31 Н	LRC CHK0	

Принимаемые данные, расписанные по регистрам:

Регистр	Данные			
D1070 младший байт	0	30 Н	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1070 старший байт	1	31 Н	ADR0	
D1071 младший байт	0	30 Н	CMD1	CMD (1, 0) – код команды
D1071 старший байт	6	36 Н	CMD0	
D1072 младший байт	0	30 Н	Адрес данных	
D1072 старший байт	1	31 Н		
D1073 младший байт	0	30 Н		
D1073 старший байт	0	30 Н		
D1074 младший байт	1	31 Н	Данные	
D1074 старший байт	7	37 Н		
D1075 младший байт	7	37 Н		
D1075 старший байт	0	30 Н		
D1076 младший байт	7	37 Н	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1076 старший байт	1	31 Н	LRC CHK0	

Пример применения 2: связь ПЛК с преобразователем частоты VFD-S (RTU-режим, M1143 = 1)



Передаваемое сообщение (ПЛК→VFD-S): **"01 06 2000 0012 02 07"**

Принимаемое сообщение (VFD-S→ ПЛК): **"01 06 2000 0012 02 07"**

Передаваемые данные, расписанные по регистрам:

Регистр	Данные	
D1089 младший байт	01 H	Коммуникационный адрес VFD-S
D1090 младший байт	06 H	Адрес команды
D1091 младший байт	20 H	Адрес данных
D1092 младший байт	00 H	
D1093 младший байт	00 H	Данные
D1094 младший байт	12 H	
D1095 младший байт	02 H	CRC CHK Low
D1096 младший байт	07 H	CRC CHK High

Принимаемые данные, расписанные по регистрам:

Регистр	Данные	
D1070 младший байт	01 H	Коммуникационный адрес VFD-S
D1071 младший байт	06 H	Адрес команды
D1072 младший байт	20 H	Адрес данных
D1073 младший байт	00 H	
D1074 младший байт	00 H	Данные
D1075 младший байт	12 H	
D1076 младший байт	02 H	CRC CHK Low
D1077 младший байт	07 H	CRC CHK High

Пример применения 3: связь ПЛК с преобразователем частоты VFD-S (ASCII-режим, M1143 = 0). Повторная передача при превышении времени ожидания ответа.

Когда X0 включено, ПЛК будет писать данные H1770 (K6000) в адрес H0100 преобразователя VFD-S с коммуникационным адресом 1.

Флаг M1129 будет включен, если время ожидания ответа от VFD-S превысит 100 мс. После чего здесь M1129 будет инициализировать повторную передачу данных, включив M1122.

Флаг M1140 будет включен, если данные будут содержать ошибки. После чего здесь M1140 будет инициализировать повторную передачу данных, включив M1122.

Флаг M1141 будет включен, если адрес данных будет задан некорректно. После чего здесь M1141 будет инициализировать повторную передачу данных, включив M1122.



API		FWD		Команда "Пуск вперед" для привода VFD-A	DVP-		
102			S₁ S₂ n		ES/EX/SS	SA/SX	EH
					+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов) FWD - Непрерывное выполнение. 32-х битная инструкция ---	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S ₁					*	*							*			
S ₂					*	*							*			
n					*	*							*			
<u>Примечания:</u> Диапазоны операндов: S ₁ = K0...K31 n = K1 или K2															Флаги: M1120 – M1131, M1140 – M1143 (см. инструкцию RS)	

API	REV	S₁ S₂ n	Команда "Пуск реверсивно" для привода VFD-A	DVP-		
				ES/EX/SS	SA/SX	EH
103				+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (7 шагов) REV - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*							*			
S ₂					*	*							*			
n					*	*							*			
<u>Примечания:</u> Диапазоны операндов: S ₁ = K0...K31 n = K1 или K2																<u>32-х битная инструкция</u> --- Флаги: M1120 – M1131, M1140 – M1143 (см. инструкцию RS)

API	STOP	S₁ S₂ n	Команда "Стоп" для привода VFD-A	DVP-		
				ES/EX/SS	SA/SX	EH
104				+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов) STOP - Непрерывное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S ₁					*	*							*				
S ₂					*	*							*				
n					*	*							*				
<u>Примечания:</u> Диапазоны операндов: S ₁ = K0...K31 n = K1 или K2																32-х битная инструкция --- Флаги: M1120 – M1131, M1140 – M1143 (см. инструкцию RS)	

Функция

Управление преобразователем частоты VFD-A/H по интерфейсу RS-485.

Описание

(S₁) – коммуникационный адрес ПЧ в сети

(S₂) – заданная частота, Гц:

для VFD-A серии S2=K0000...K4000, что соответствует 0.0 Гц ...400.0 Гц;

для VFD-H серии S2=K0000...K1500, что соответствует 0 Гц ...1500 Гц

(n) – объект команды: если n=1, команда относится к одному ПЧ; если n=2, команда относится ко всем ПЧ, связанным с ПЛК

- Принятый ответ запоминается в буферной области приема: регистрах D1070 – D1080, и проверяется на корректность. Если обнаружены ошибки, включается специальное реле M1142. Если (n)=2, ПЛК не будет принимать ответ от ПЧ.

Пример применения: связь ПЛК с преобразователем частоты VFD-A. Повторная передача при превышении времени ожидания ответа и ошибке данных.



Передаваемое сообщение (ПЛК→VFD-A): "C ♥ ☺ 0001 0500"

Принимаемое сообщение (VFD-A→ПЛК): "C ♥ ♠ 0001 0500"

Передаваемые данные, расписанные по регистрам:

Регистр	Данные		
D1089 младший байт	C	43 H	Стартовое слово команды
D1090 младший байт	♥	03 H	Контрольная сумма
D1091 младший байт	☺	01 H	Объект команды
D1092 младший байт	0	30 H	Коммуникационный адрес VFD-A
D1093 младший байт	0	30 H	
D1094 младший байт	0	30 H	
D1095 младший байт	1	31 H	
D1096 младший байт	0	30 H	Заданная частота
D1097 младший байт	5	35 H	
D1098 младший байт	0	30 H	
D1099 младший байт	0	30 H	

Принимаемые данные, расписанные по регистрам:

Регистр	Данные		
D1070 младший байт	C	43 H	Стартовое слово команды
D1071 младший байт	♥	03 H	Контрольная сумма
D1072 младший байт	♠	06 H	Авторизация ответа (06H – корректно, 07H - ошибка)
D1073 младший байт	0	30 H	Коммуникационный адрес VFD-A
D1074 младший байт	0	30 H	
D1075 младший байт	0	30 H	
D1076 младший байт	1	31 H	
D1077 младший байт	0	30 H	Заданная частота
D1078 младший байт	5	35 H	
D1079 младший байт	0	30 H	
D1080 младший байт	0	30 H	

API	RDST		 	Чтение текущего состояния VFD-A	DVP-		
105					ES/EX/SS	SA/SX	EH
					+	+	+

Оп- еранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S ₁					*	*							*		
n					*	*							*		
<div>Примечания: Диапазоны операндов: S₁ = K0 ... K31 n = K0 ... K3</div>															

16-ти битная инструкция
(5 шагов)
RDST - Непрерывное
выполнение.

32-х битная инструкция

Флаги:
M1120 – M1131,
M1140 – M1143 (см.
инструкцию RS)

Функция

Чтение текущего состояния преобразователя частоты VFD-A по интерфейсу RS-485.

Описание

(S_1) – коммуникационный адрес ПЧ в сети

(n) – объект состояния:

$n = 0$ – заданная частота;

$n = 1$ – выходная частота;

$n = 2$ – выходной ток;



$n = 3$ – режим работы

Принятый ответ запоминается в буферной области приема: регистрах D1070 – D1080, максимум 11 слов. См. инструкцию на преобразователь частоты VFD-A.

Нельзя использовать в качестве условия выполнения RSTEF-инструкции импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF). Иначе данные будут приняты не корректно.

Ответ	Описание	
Q	Стартовое слово: 51H	
S	Код контрольной суммы: 03H	
B	Авторизация ответа (06H – корректно, 07H - ошибка)	
U	Коммуникационный адрес VFD-A: "Uu" = "00" – "31" (ASCII)	
u		
N	Объект состояния: "Nn" = "00" – "03" (ASCII)	
n		
A	Данные состояния. Содержимое "ABCD" будет различным в зависимости от заданного объекта состояния. См. ниже.	
B		
C		
D		
	Nn = 00	Заданная частота = ABC.D (Гц) Выходная частота = ABC.D (Гц) Выходной ток = ABC.D (А) Принятые данные будут автоматически конвертироваться в HEX-значения и сохраняться в D1050. Например, "ABCD" = "0600" → D1050 = 0258H (K600)
	Nn = 01	
	Nn = 02	
	Nn = 03	Режим работы
	‘A’ =	‘0’ Stop, ‘5’ JOG(FWD)

		‘1’ FWD работа, ‘6’ JOG(REV) ‘2’ Stop, ‘7’ JOG(REV) ‘3’ REV работа, ‘8’ Сбой привода ‘4’ JOG(FWD) Принятые данные будут автоматически конвертироваться в HEX-значения и сохраняться в D1051.					
	‘B’ =	b7	b6	b5	b4	Источник команды управления	
		0	0	0	0	Встроенный пульт управления	
		0	0	0	1	Фиксированная скорость 1	
		0	0	1	0	Фиксированная скорость 2	
		0	0	1	1	Фиксированная скорость 3	
		0	1	0	0	Фиксированная скорость 4	
		0	1	0	1	Фиксированная скорость 5	
		0	1	1	0	Фиксированная скорость 6	
		0	1	1	1	Фиксированная скорость 7	
		1	0	0	0	Фиксированная JOG-скорость	
		1	0	0	1	Аналоговый сигнал	
		1	0	1	0	RS-485	
		1	0	1	1	Команда Up/Down	
		b3 = 1: идет торможение пост. током при останове					
		b2 = 1: идет торможение пост. током при старте					
		b1 = 0: FWD; b1 = 1: REV					
		b1 = 0: Стоп; b1 = 1: Работа					
		Принятые данные "B" сохраняться в M1168(b0)~M1175(b7).					
	“CD” =	00	Нет аварии			10	OcA
		01	oc			11	Ocd
		02	ov			12	Ocn
		03	oH			13	GFF
		04	oL			14	Lv
		05	oL1			15	Lv1
		06	EF			16	cF2
		07	cF1			17	bb
		08	cF3			18	oL2
		09	HPF			19	
		Принятые данные будут автоматически конвертироваться в HEX-значения и сохраняться в D1052.					

API		RSTEF		 	Команда "Сброс" для привода VFD-A	DVP-		
						ES/EX/SS	SA/SX	EH
106						+	+	+

Оп- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (5 шагов) RSTEF - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S ₁					*	*							*			
n					*	*							*			
<u>Примечания:</u> Диапазоны операндов: S ₁ = K0 ... K31 n = K1 или K2															<u>32-х битная инструкция</u> ---	
															Флаги: M1120 – M1131, M1140 – M1143 (см. инструкцию RS)	

Функция

Команда "Сброс" аварийной блокировки для преобразователя частоты VFD-A по интерфейсу RS-485 после сбоя в работе.

Описание

(S₁) – коммуникационный адрес ПЧ в сети

(n) – объект команды:

если n=1, команда относится к одному ПЧ; если n=2, команда относится ко всем ПЧ, связанным с ПЛК

- Принятый ответ запоминается в буферной области приема: регистрах D1070 – D1089, и проверяется на корректность. Если (n)=2, ПЛК не будет принимать ответ от ПЧ.
- Нельзя использовать в качестве условия выполнения RSTEF-инструкции импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF). Иначе данные будут приняты не корректно.
- Для детальной информации см. описание RS-инструкции.

API		LRC		<div>S</div> <div>n</div> <div>D</div>			Расчет контрольной суммы LRC	DVP-		
107			P					ES/EX/SS	SA/SX	EH
								-	+	+

Оп-ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S													*		
n					*	*							*		
D													*		
Примечания: нет															

16-ти битная инструкция
(7 шагов)
LRC - Непрерывное выполнение.
LRCP - Импульс. вып-е

32-х битная инструкция

Флаги:
M1161 (установка режима 8/16 бит)

Функция

Вычисление контрольной суммы последовательных данных по методу LRC (продольная проверка избыточности). Используется для ASCII-режима передачи данных.

Описание

(S₁) – стартовый адрес проверяемых данных

(n) – число операндов

(D) – стартовый адрес результата

- Число операндов (n) должно быть четным и быть в диапазоне K1...K256. При выходе за указанный диапазон будет выдана ошибка с кодом 0E1A и записана в спец. регистр D1067, спец. реле M1067, M1068 будут включены.
- 16-бит режим (M1161=0): шестнадцатеричные данные, начиная с адреса (S) будут разделены на старшие 8 бит и младшие 8 бит, и LRC-инструкция будет

выполняться для (n) адресов. Результат будет сохранен в старшие и младшие байты операнда (D).

- 8-бит режим (M1161=1): LRC-инструкция будет выполняться для младших байтов (n) регистров начиная с адреса (S). Результат будет сохранен в младшие байты операндов (D) и (D)+1. В старшие байты регистров результата будут записаны ноли.

Пример: Чтение состояния преобразователя частоты VFD-B из шести регистров с начальным адресом H2101 (ASCII режим (M1143=0) / 8-бит режим (M1161=1):

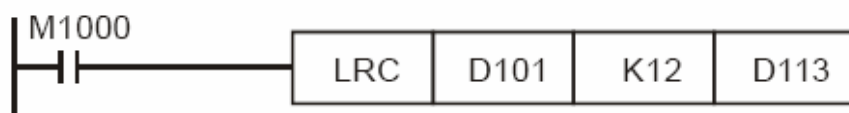


Передаваемое сообщение от ПЛК к VFD-B: “: 01 03 2101 0006 D4 CR LF “

Передаваемые данные, расписанные по регистрам:

Регистр	Данные			
D100 младший байт	:	3A H	STX	
D101 младший байт	0	30 H	ADR1	ADR (1, 0) – коммуникационный адрес VFD-B
D102 младший байт	1	31 H	ADR0	
D103 младший байт	0	30 H	CMD1	CMD (1, 0) – код команды
D104 младший байт	3	33 H	CMD0	
D105 младший байт	2	32 H	Стартовый адрес данных	
D106 младший байт	1	31 H		
D107 младший байт	0	30 H		
D108 младший байт	1	31 H		
D109 младший байт	0	30 H	Число данных (слов)	
D110 младший байт	0	30 H		
D111 младший байт	0	30 H		
D112 младший байт	6	36 H		
D113 младший байт	D	44 H	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D114 младший байт	4	34 H	LRC CHK0	
D115 младший байт	CR	A H	Конец	
D116 младший байт	LF	D H		

Контрольная сумма LRC CHK (1, 0) может быть рассчитана с помощью LRC-инструкции (8-бит режим, M1161=1) следующим образом:



$$01\text{ H} + 03\text{ H} + 21\text{ H} + 01\text{ H} + 00\text{ H} + 06\text{ H} = 2\text{C H},$$

2-ух дополнений отрицание 3CH есть **D4H** ($100 - 3\text{C} = \text{D4}$). При этом результат будет сохранен в младшие байты регистров D113, D114.

API		CRC		<div>S</div> <div>n</div> <div>D</div>			Расчет контрольной суммы CRC	DVP-		
108			P					ES/EX/SS	SA/SX	EH
								-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (7 шагов) CRC - Непрерывное выполнение. CRCP - Импл. вып-е
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S													*		
n					*	*							*		
D													*		

Примечания: нет

32-х битная инструкция

Флаги:
M1161 (установка
режима 8/16 бит)

Функция

Вычисление контрольной суммы последовательных данных по методу CRC (циклическая проверка избыточности). Используется для RTU-режима передачи данных.

Описание

(S₁) – стартовый адрес проверяемых данных

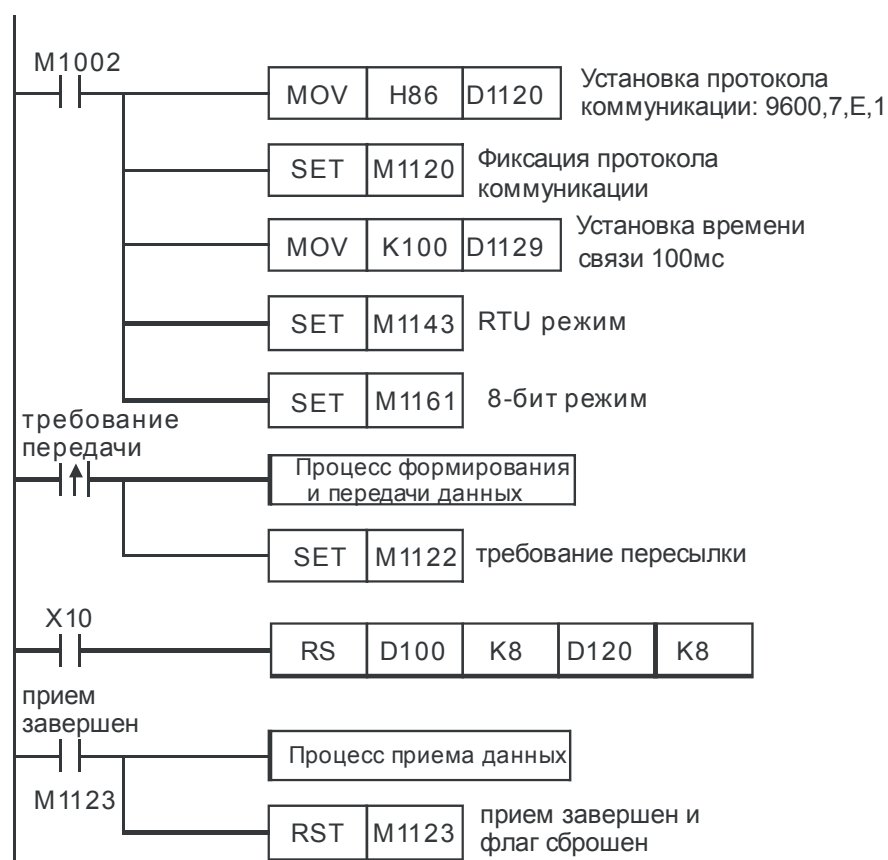
(n) – число операндов

(D) – стартовый адрес результата

- Число операндов (n) должно быть в диапазоне K1...K256. При выходе за указанный диапазон будет выдана ошибка с кодом 0E1A и записана в спец. регистр D1067, спец. реле M1067, M1068 будут включены.
- 16-бит режим (M1161=0): шестнадцатеричные данные, начиная с адреса (S) будут разделены на старшие 8 бит и младшие 8 бит, и CRC-инструкция будет выполняться для (n) адресов. Результат будет сохранен в старшие и младшие байты операнда (D).
- 8-бит режим (M1161=1): CRC-инструкция будет выполняться для младших байтов (n) регистров начиная с адреса (S). Результат будет сохранен в младшие байты операндов (D) и (D)+1. В старшие байты регистров результата будут записаны ноли.

Пример: Запись в регистр с адресом H2000 преобразователя частоты VFD-S числа H12 (RTU режим (M1143=1) / 8-бит режим (M1161=1):

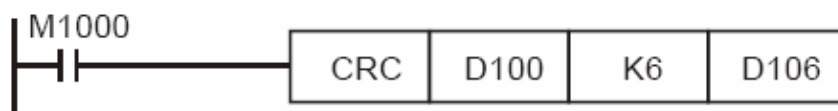
Передаваемое сообщение от ПЛК к VFD-B: **01 06 2000 0012 02 07**



Передаваемые данные, расписанные по регистрам:

Регистр	Данные	
D100 младший байт	01 H	Коммуникационный адрес VFD-S
D101 младший байт	06 H	Адрес команды
D102 младший байт	20 H	Адрес данных
D103 младший байт	00 H	
D104 младший байт	00 H	Данные
D105 младший байт	12 H	
D106 младший байт	02 H	CRC CHK Low
D107 младший байт	07 H	CRC CHK High

Контрольная сумма CRC CHK (1, 0) может быть рассчитана с помощью CRC-инструкции (8-бит режим, M1161=1) как показано в примечании и результат записан в младшие байты регистров D106 и D107:



Примечание: Порядок расчета контрольной суммы по методу CRC

RTU Режим:

ADR	01H
CMD	03H
Начальный адрес	21H
	02H
Число данных (Индекс слова)	00H
	02H
CRC CHK Low	6FH
CRC CHK High	F7H

CRC (циклическая проверка по избыточности) рассчитанная следующими шагами:

Шаг 1 : Загрузка 16-bit регистра (называемого CRC регистром) с FFFFH;

Шаг 2: Исключающее ИЛИ первому 8-bit байту из командного сообщения с байтом младшего порядка из 16-bit регистра CRC, помещение результата в CRC регистр.


Шаг 3: Сдвиг одного бита регистра CRC вправо с MSB нулевым заполнением. Извлечение и исследование LSB.

Шаг 4: Если LSB CRC регистра равно 0, повторите шаг 3, в противном случае исключающее ИЛИ CRC регистра с полиномиальным значением A001H.

Шаг 5: Повторяйте шаг 3 и 4, до тех пор, пока восемь сдвигов не будут выполнены. Затем, полный 8-bit байт будет обработан.

Шаг 6: Повторите шаг со 2 по 5 для следующих 8-bit байтов из командного сообщения.

Продолжайте пока все байты не будут обработаны. Конечное содержание CRC регистра CRC значение. При передаче значения CRC в сообщении, старшие и младшие байты значения CRC должны меняться, то есть сначала будет передан младший байт.

API				Чтение значения с цифровых микропереключателей	DVP-		
109		SWRD	P		ES/EX/SS	SA/SX	EH
					-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (2 шага) SWRD - Непрерывное выполнение. SWRDP - Имп. вып-е	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
D								*	*	*	*	*	*	*	*	
<div>Примечания: нет</div>																<div>32-х битная инструкция --- Флаги: M1104 – M1111 (состояния микропереключателей)</div>

Функция

Чтение значения с DIP-переключателей функциональной карты DVP-F8ID и запись результата в младший байт операнда (D)

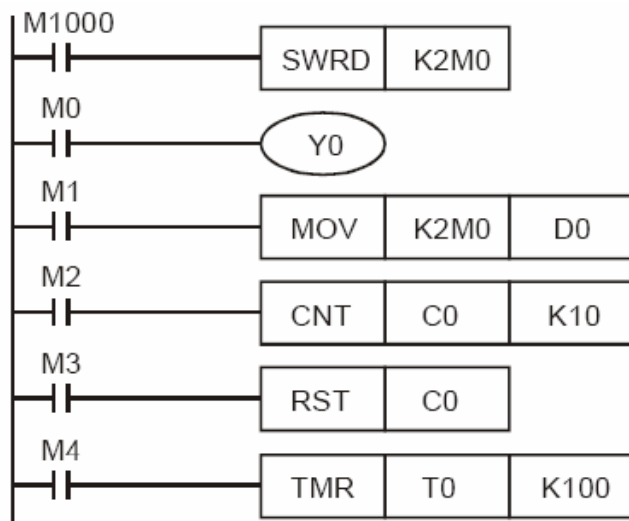
Описание

(D) –адрес операнда результата

- Каждый DIP-переключатель имеет ассоциированный с ним бит.
- Если SWRD-инструкция выполняется без функциональной карты DVP-F8ID, то результат будет = 0, сообщения об ошибке при этом не будет.

Пример применения:

Состояния 8-ми DIP-переключателей функциональной карты DVP-F8ID будут переданы внутренним реле M0 – M7.



Реле будут установлены в считанное состояние после команды END. Инструкция REF не будет действовать.

Минимальное значение бит считываемых одновременно может быть 4 (K1Y, K1M и т.д.).

Примечание: Индивидуально считать состояние каждого DIP-переключателя функциональной карты DVP-F8ID можно с помощью спец. реле M1104 – M1111.

API	-	ECMP		<div>S₁</div> <div>S₂</div> <div>D</div>			Сравнение чисел с плавающей запятой	DVP-		
110	D		P	ES/EX/SS	SA/SX	EH				
								+	+	+

Оп-еранд	Биты				Слова										16-ти битная инструкция --- 32-х битная инструкция (13 шагов) DECMP - Непрерывное выполнение. DECMP - Импульсное выполнение. Флаги: нет		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
	S ₁				*	*							*				
	S ₂				*	*							*				
	D		*	*	*												

Примечания: Операнд D занимает 3 непрерывных адреса.
Эта инструкция может работать только 32-х битном режиме (DECMP, DECMP).

Серия ES/EX/SS не поддерживает импульсное выполнение DECMP.

Функция

Сравнение двух двоичных чисел с плавающей запятой с выдачей результата сравнения

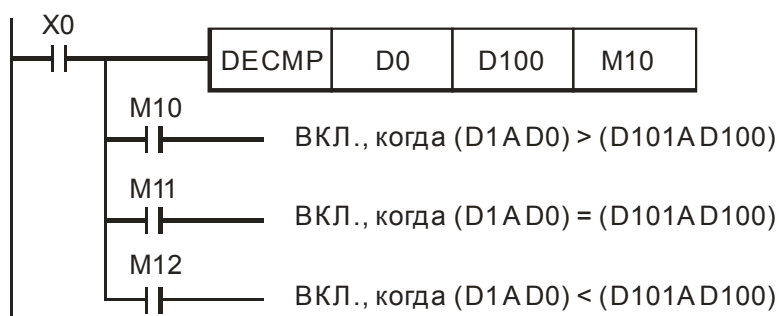
Описание

- 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).

! Опрошенные операнды выходов остаются включенными после отключения условий выполнения DECMP-инструкции.

! Если в качестве операнда (S₁) или (S₂) используется константа K/H, она автоматически из целого конвертируется в двоичный формат с плавающей запятой.

Пример программирования DECMP-инструкции:



При включении контакта X0 сравниваются число с плавающей запятой, указанное в D100 (S₂), с числом с плавающей запятой, указанным в D0 (S₁).

Если число в D100 меньше числа D0, то включается реле M10. Если число в D100 равно числу D0, то включается реле M11. Если число в D100 больше числа D0, то включается реле M12.

Для получения результатов сравнения в виде: \leq \geq \neq можно использовать параллельные комбинации контактов M10 – M12.

Для сброса результата можно использовать команды RST, ZRST.

API	-	EZCP	P	S₁ S₂ S D	Зонное сравнение чисел с плавающей запятой	DVP-		
						ES/EX/SS	SA/SX	EH
111	D					+	+	+

Операнд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	--- 32-х битная инструкция (17 шагов) DEZCP - Непрерывное выполнение. DEZCPP - Импульсное выполнение. Флаги: нет
S ₁					*	*							*			
S ₂					*	*							*			
S ₃					*	*							*			
D		*	*	*												

Примечания: Операнд D занимает 3 непрерывных адреса.
 Операнд S1 должен быть меньше чем S2.
 Эта инструкция может работать только 32-х битном режиме (DEZCP, DEZCPP).
 Серия ES/EX/SS не поддерживает импульсное выполнение DEZCPP.

Функция

Сравнение числа с плавающей запятой с выделенной (указанной) областью с выдачей результата сравнения

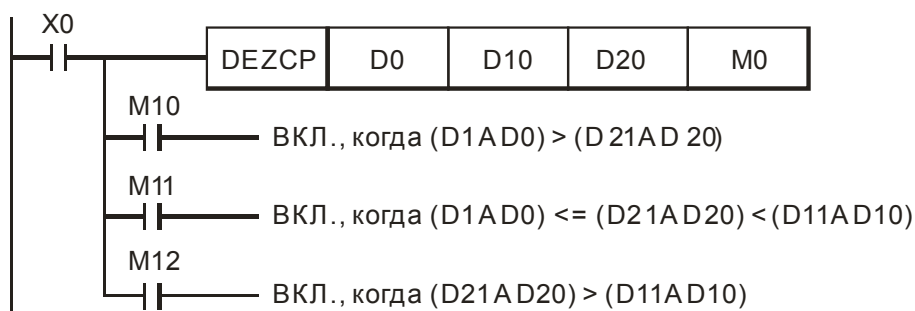
Описание

- 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).

! Опрошенные операнды выходов остаются включенными после отключения условий выполнения DEZCP -инструкции.

! Если в качестве операнда (S1) или (S2) используется константа К/Н, она автоматически из целого конвертируется в двоичный формат с плавающей запятой.

Пример программирования DEZCP-инструкции:



При включении контакта X0 сравниваются число с плавающей запятой D20 (S3) с числами в области между D0 (S1) и D10 (S2).

Если число в D20 меньше числа между D0 и D10, то включается реле M10. Если число в D20 равно числу между D0 и D10, то включается реле M11. Если число в D20 больше числа между D0 и D10, то включается реле M12.

Для сброса результата можно использовать команды RST, ZRST.

API	-	RAD		Перевод градусов в радианы	DVP-		
116	D		S D		ES/EX/SS	SA/SX	EH
					-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			
D													*			
<div>Примечания:</div> <div>Эта инструкция может работать только 32-х битном режиме (DRAD, DRADP).</div>																<div>---</div> <div>32-х битная инструкция (9 шагов)</div> <div>DRAD - Непрерывное выполнение.</div> <div>DRADP - Импульсное выполнение.</div> <div>Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)</div>

Функция

Преобразование градусов в радианы по формуле: $\text{рад} = \text{грд} \times (\pi / 180)$

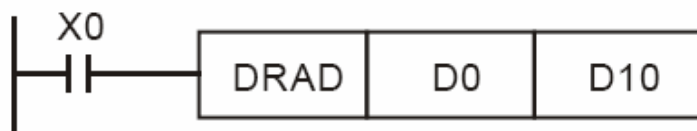
Описание

(S) – исходные данные (грд)

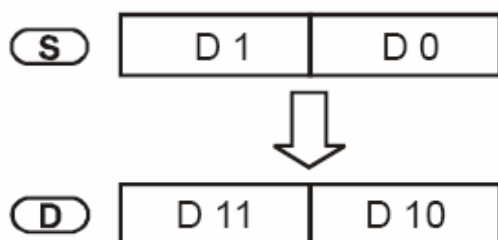
(D) – результат преобразования (рад)

- Если абсолютное значение (D) > макс. значения числа с плавающей запятой, то M1022 = 1.
- Если абсолютное значение (D) < мин. значения числа с плавающей запятой, то M1021 = 1.
- Если абсолютное значение (D) = 0, то M1020 = 1.

Пример



Когда X0 включено, происходит преобразование градусов записанных в регистрах D0, D1 в радианы и результат сохраняется в регистрах D10, D11. Данные регистров D0, D1 и D10, D11 - в двоичном формате с плавающей запятой



API	-	DEG		Перевод радианов в градусы	DVP-		
117	D		P		ES/EX/SS	SA/SX	EH
					-	+	+

Оп- ранд	Биты				Слова								16-ти битная инструкция			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S					*	*							*			32-х битная инструкция (9 шагов)
D													*			DDEG - Непрерывное выполнение. DDEGP - Импульсное выполнение.
Примечания: Эта инструкция может работать только 32-х битном режиме (DDEG, DDEGP).																Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)

Функция

Преобразование радианов в градусы по формуле: $\text{грд} = \text{рад} \times (180 / \pi)$

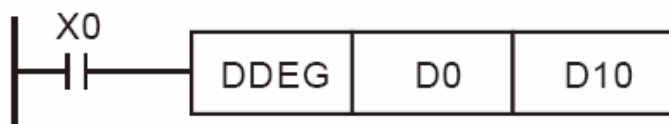
Описание

(S) – исходные данные (рад)

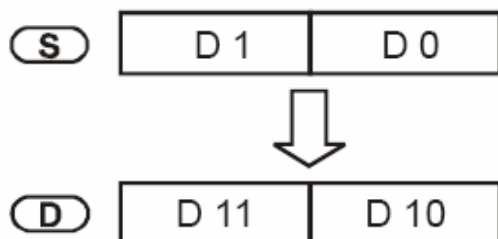
(D) – результат преобразования (грд)

- Если абсолютное значение (D) > макс. значения числа с плавающей запятой, то M1022 = 1.
- Если абсолютное значение (D) < мин. значения числа с плавающей запятой, то M1021 = 1.
- Если абсолютное значение (D) = 0, то M1020 = 1.

Пример



Когда X0 включено, происходит преобразование радианов записанных в регистрах D0, D1 в градусы и результат сохраняется в регистрах D10, D11. Данные регистров D0, D1 и D10, D11 - в двоичном формате с плавающей запятой



API	-	EBCD	P	S D	Перевод двоичного числа с плавающей запятой в десятичный формат с плавающей запятой	DVP-		
						ES/EX/SS	SA/SX	EH
118	D					+	+	+

Операнд	Биты				Слова											16-ти битная инструкция
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			---
D													*			32-х битная инструкция (9 шагов)
Примечания: Эта инструкция может работать только 32-х битном режиме (DEBCD, DEBCDP). Серия ES/EX/SS не поддерживает импульсное выполнение DEBCDP. Форматы чисел с плавающей запятой описаны в главе 5.3.2																DEBCD - Непрерывное выполнение. DEBCDP - Импульсное выполнение. Флаги: нет

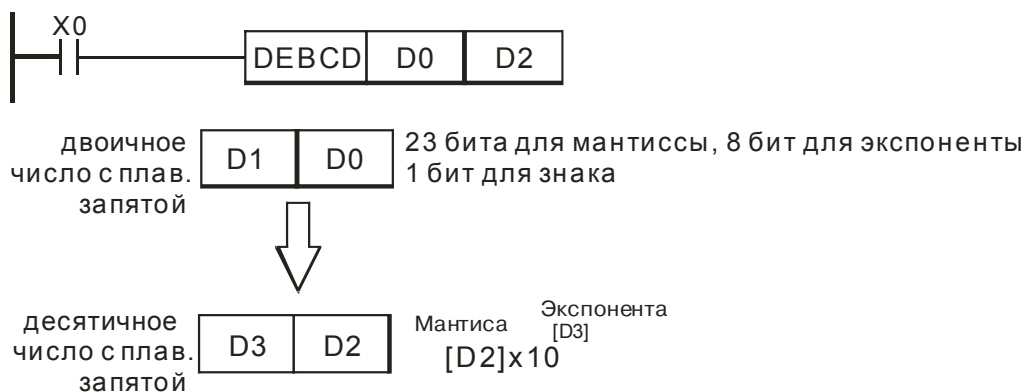
Функция

Преобразование числа в двоичном формате с плавающей запятой в десятичный формат с плавающей запятой.

Описание

- Число, заданное в двоичном формате с плавающей запятой, в (S) конвертируется и запоминается в (D).
- Мантисса запоминается в (D).
- Экспонента запоминается в ((D)+1).

Пример



При включении входа X0 число, записанное в D0 и D1, в двоичном формате с плавающей запятой конвертируется в десятичный формат с плавающей запятой и записывается в D2 и D3.

Мантисса указывается в D2. Экспонента указывается в D3.

Примечание: Форматы чисел с плавающей запятой подробно описаны в главе 5.3.2

API	-				Перевод десятичного числа с плавающей запятой в двоичный формат с плавающей запятой	DVP-		
119	D	EBIN	P	S D		ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			32-х битная инструкция (9 шагов) DEBIN - Непрерывное выполнение. DEBINP - Импульсное выполнение. Флаги: нет
D													*			
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DEBIN, DEBINP). Серия ES/EX/SS не поддерживает импульсное выполнение DEBINP. Форматы чисел с плавающей запятой описаны в главе 5.3.2																

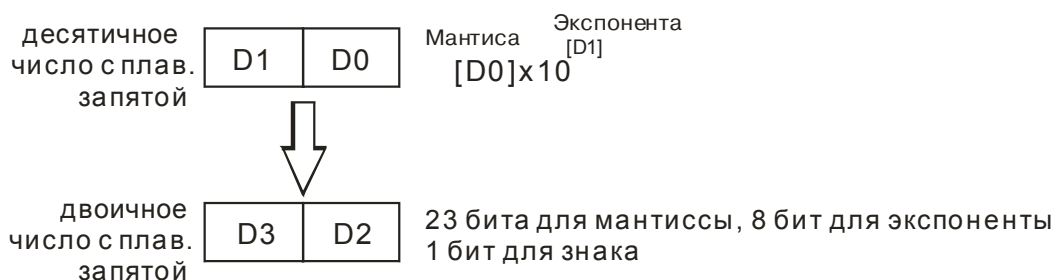
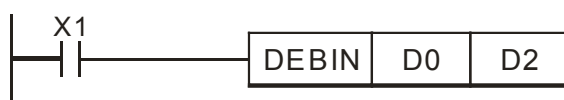
Функция

Преобразование числа в десятичном формате с плавающей запятой в двоичный формат с плавающей запятой.

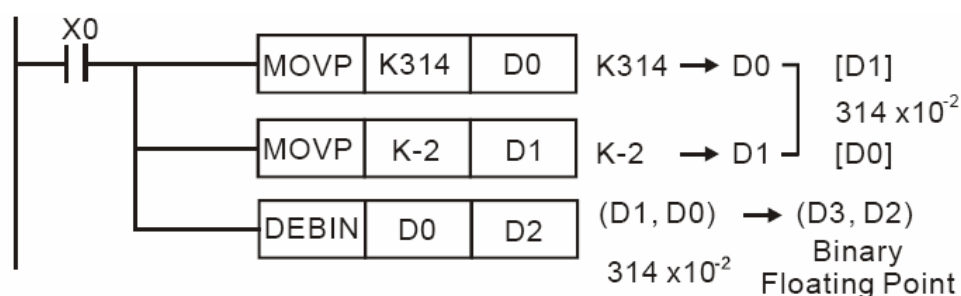
Описание

- Число, заданное в десятичном формате с плавающей запятой, в (S) конвертируется и запоминается в (D).
- Мантисса запоминается в (D).
- Экспонента запоминается в ((D)+1).

Пример 1:



Пример 2:



Примечание: Форматы чисел с плавающей запятой подробно описаны в главе 5.3.2

API	-	EADD	P	(S1) (S2) (D)	Сложение чисел с плавающей запятой	DVP-		
120	D					ES/EX/SS	SA/SX	EH
						+	+	+

Оп- еранд	Биты				Слова											16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---	
S ₁					*	*							*			<u>32-х битная инструкция</u> (13 шагов) DEADD - Непрерывное выполнение. DEADDP - Импульсное выполнение. Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)	
S ₂					*	*							*				
D													*				
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DEADD, DEADDP). Серия ES/EX/SS не поддерживает импульсное выполнение DEADDP.																	

Функция

Вычисление суммы двух чисел в двоичном формате с плавающей запятой

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 два следующих друг за другом регистра.
- Задаваемые константы (K, H) 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 ноля (zero) M1020.
- Если результат сложения больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.
- Если результат сложения меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M1021.

Пример 1: программирование DEADD-инструкции

При включении входа X0, к числу с плавающей запятой, записанному в (D1, D0) будет прибавлено число с плавающей запятой, записанное в (D3, D2).

Результат сохранится в (D11, D10).



Пример 2: программирование DEADD-инструкции

При включении входа X2, к константе K1234 прибавится число с плавающей запятой, записанное в (D11, D10).

Результат сохранится в (D21, D20).



API	-	ESUB		Вычитание чисел с плавающей запятой	DVP-		
121	D		(S1) (S2) (D)		ES/EX/SS	SA/SX	EH
					+	+	+

Опе- ранд	Биты				Слова											16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---	
S ₁					*	*							*			32-х битная инструкция (13 шагов) DESUB - Непрерывное выполнение. DESUBP - Импульсное выполнение. Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)	
S ₂					*	*							*				
D													*				
<div>Примечания: Эта инструкция может работать только 32-х битном режиме (DESUB, DESUBP). Серия ES/EX/SS не поддерживает импульсное выполнение DESUBP.</div>																	

Функция

Вычисление разности двух чисел в двоичном формате с плавающей запятой

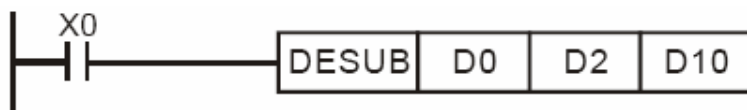
Описание

- Число с плавающей запятой, заданное в (S2), вычитается из числа с плавающей запятой в (S1). Результат запоминается в (D).
- Для каждого операнда используется по два следующих друг за другом регистра.
- Задаваемые константы (K, H) 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 нуля (zero) M1020.
- Если результат сложения больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.
- Если результат сложения меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M1021.

Пример 1: программирование DESUB-инструкции

При включении входа X0, из числа с плавающей запятой, записанного в (D1, D0) будет вычтено число с плавающей запятой, записанное в (D3, D2).

Результат сохранится в (D11, D10).



Пример 2: программирование DESUB-инструкции

При включении входа X2, из константы K1234 будет вычтено число с плавающей запятой, записанное в (D1, D0).

Результат сохранится в (D11, D10).



API	-	EMUL		<div>S1S2D</div>			Умножение чисел с плавающей запятой							DVP-			
122	D		P	ES/EX/SS		SA/SX								EH			
														+		+	+

Операнд	Биты				Слова												16-ти битная инструкция ---		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F				
	S1					*	*							*					
	S2					*	*							*					
	D													*					
<div>Примечания:</div> <div>Эта инструкция может работать только 32-х битном режиме (DEMUL, DEMULP).</div> <div>Серия ES/EX/SS не поддерживает импульсное выполнение DEMULP.</div>																		<div>32-х битная инструкция (13 шагов)</div> <div>DEMUL - Непрерывное выполнение.</div> <div>DEMULP - Импульсное выполнение.</div> <div>Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)</div>	

Функция

Вычисление произведения двух чисел в двоичном формате с плавающей запятой

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 два следующих друг за другом регистра.
- Задаваемые константы (K,H) 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.
- Если результат умножения равен 0, то включается флаг ноля (zero) M1020.
- Если результат умножения больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.

- Если результат умножения меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M1021.

Пример 1: программирование DEMUL-инструкции

При включении входа X1, число с плавающей запятой, записанное в (D1, D0) будет умножено на число с плавающей запятой, записанное в (D11, D10).

Результат сохранится в (D21, D20).



Пример 2: программирование DEMUL-инструкции

При включении входа X2, константа K1234 будет умножена на число с плавающей запятой, записанное в (D1, D0).

Результат сохранится в (D11, D10).



API	-	EDIV		Деление чисел с плавающей запятой	DVP-		
123	D	P	(S1) (S2) (D)		ES/EX/SS	SA/SX	EH
					+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S ₁					*	*							*			32-х битная инструкция (13 шагов) DEDIV - Непрерывное выполнение. DEDIVP - Импульсное выполнение. Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)
S ₂					*	*							*			
D													*			
Примечания: Эта инструкция может работать только 32-х битном режиме (DEDIV, DEDIVP). Серия ES/EX/SS не поддерживает импульсное выполнение DEDIVP.																

Функция

Вычисление частного от деления двух чисел в двоичном формате с плавающей запятой

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 два следующих друг за другом регистра.
- Задаваемые константы (K, H) 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 деления равен 0, то включается флаг ноля (zero) M1020.
- Если результат деления больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.
- Если результат деления меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M1021.

Источник ошибки

Операнд (S2) не может быть равен нулю т.к. деление на нуль не допустимо.

Пример 1: программирование DEDIV-инструкции

При включении входа X1, число с плавающей запятой, записанное в (D1, D0) будет разделено на число с плавающей запятой, записанное в (D11, D10).

Результат сохранится в (D21, D20).





Пример 2: программирование DEDIV-инструкции

При включении входа X2, число с плавающей запятой, записанное в (D1, D0) будет разделено на константу K10.

Результат сохранится в (D11, D10).



API	-	EXP		 		Вычисление экспоненты в формате с плавающей запятой								DVP-		
124	D		P	ES/EX/SS										SA/SX	EH	
														+	+	+

Оп- ранд	Биты				Слова											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S					*	*							*			
D													*			

Примечания:

Эта инструкция может работать только 32-х битном режиме (DEXP, DEXPP).
Серия ES/EX/SS не поддерживает импульсное выполнение DEXPP.

16-ти битная инструкция

32-х битная инструкция
(13 шагов)
DEXP - Непрерывное выполнение.
DEXPP - Импульсное выполнение.

Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)

Функция

Вычисление экспоненты числа в двоичном формате с плавающей запятой

Описание

- Находится экспонента числа, заданного в (S). Результат запоминается в (D).

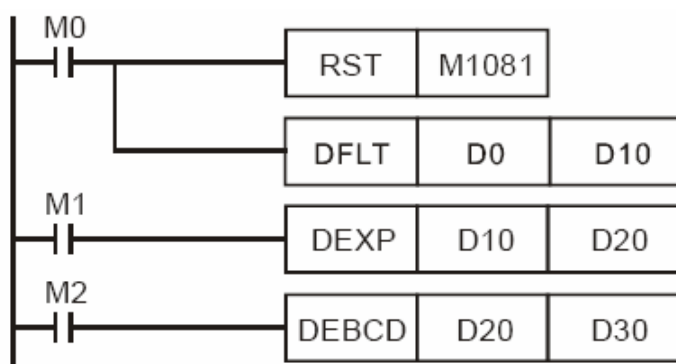
$$e^{[S]} = [D], \text{ где } e = 2.71828$$
- Для каждого операнда используется по два следующих друг за другом регистра.
- Задаваемые константы (K, H) перед делением автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.
- Если результат равен 0, то включается флаг нуля (zero) M1020.
- Если результат больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.
- Если результат меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M1021.



Пример: программирование DEXP-инструкции

Когда M0=1, целое число, записанное в D0, D1 конвертируется в двоичное число с плавающей запятой и записывается в (D10, D11).

Когда M1=1, находится экспонента числа, записанного в D10, D11 и результат записывается в (D20, D21).

Когда M2=1, двоичное число с плавающей запятой, записанное в (D20, D21) конвертируется в десятичное число с плавающей запятой и записывается в (D30, D31).



API	-	LN		 	Вычисление логарифма натурального в формате с плавающей запятой	DVP-		
125	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			
D													*			
<div>Примечания:</div> <div>Эта инструкция может работать только 32-х битном режиме (DLN, DLNP). Серия ES/EX/SS не поддерживает импульсное выполнение DLNP.</div>															---	
															32-х битная инструкция (13 шагов) DLN - Непрерывное выполнение. DLNP - Импульсное выполнение.	
															Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)	

Функция

Вычисление логарифма натурального (с основанием "е") числа в двоичном формате с плавающей запятой

Описание

- Находится логарифм натуральный числа, заданного в (S). Результат запоминается в (D).

$$\ln[S] = [D]$$

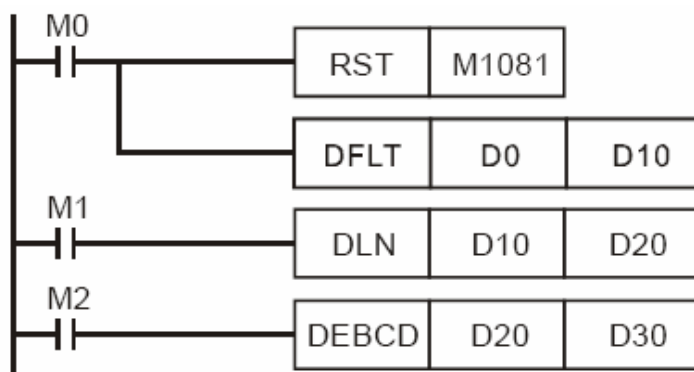
- Для каждого операнда используется по два следующих друг за другом регистра.
- Задаваемые константы (K, H) перед делением автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.
- Операнд (S) может иметь только положительное значение.
- Если результат равен 0, то включается флаг ноля (zero) M1020.
- Если результат больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.
- Если результат меньше, чем минимально допустимое значение, то включается флаг заема (Borrow) M1021.

Пример: программирование DLN-инструкции

Когда M0=1, целое число, записанное в D0, D1 конвертируется в двоичное число с плавающей запятой и записывается в (D10, D11).

Когда M1=1, находится натуральный логарифм числа, записанного в D10, D11 и результат записывается в (D20, D21).

Когда M2=1, двоичное число с плавающей запятой, записанное в (D20, D21) конвертируется в десятичное число с плавающей запятой и записывается в (D30, D31).



API	-	LOG		Вычисление логарифма в формате с плавающей запятой	DVP-		
126	D		(S1) (S2) (D)		ES/EX/SS	SA/SX	EH
					+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S ₁					*	*							*			32-х битная инструкция (13 шагов) DLOG - Непрерывное выполнение. DLOGP - Импульсное выполнение. Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)
S ₂					*	*							*			
D													*			
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DLOG, DLOGP). Серия ES/EX/SS не поддерживает импульсное выполнение DLOGP.																

Функция

Вычисление логарифма числа в двоичном формате с плавающей запятой

Описание

- Находится логарифм с основанием (S1) числа, заданного в (S2). Результат запоминается в (D).

$$\log_{[S1]}[S2] = [D]$$

- Для каждого операнда используется по два следующих друг за другом регистра.
- Задаваемые константы (K, H) 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.
- Операнд (S2) может иметь только положительное значение.
- Если результат равен 0, то включается флаг ноля (zero) M1020.
- Если результат больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.

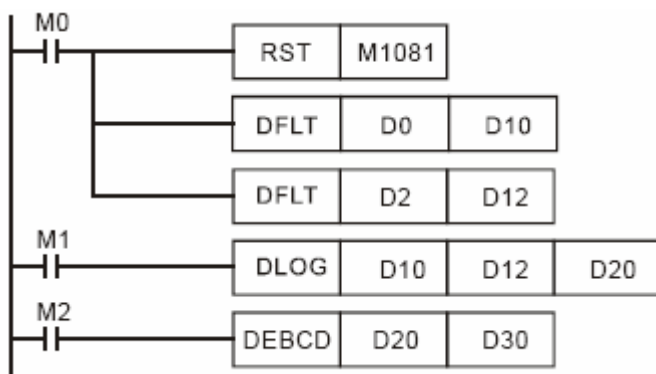
- Если результат меньше, чем минимально допустимое значение, то включается флаг заимствования (Borrow) M1021.

Пример: программирование DLOG-инструкции

Когда M0=1, целое число, записанное в D0, D1 конвертируется в двоичное число с плавающей запятой и записывается в (D10, D11); целое число, записанное в D2, D3 конвертируется в двоичное число с плавающей запятой и записывается в (D12, D13)

Когда M1=1, находится логарифм с основанием (D10, D11) числа, записанного в (D12, D13) и результат записывается в (D20, D21).

Когда M2=1, двоичное число с плавающей запятой, записанное в (D20, D21) конвертируется в десятичное число с плавающей запятой и записывается в (D30, D31).



API	-	ESQR			Корень квадратный в формате с плавающей запятой	DVP-		
127	D		P	S D		ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция ---	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S					*	*							*			32-х битная инструкция (9 шагов) DESQR - Непрерывное выполнение. DESQRP - Импульсное выполнение. Флаги: M1020 (флаг ноля), M1067 (флаг ошибки)
D													*			
<div>Примечания: Необходимое условие: $(S) \geq 0$ Эта инструкция может работать только 32-х битном режиме (DESQR, DESQRP). Серия ES/EX/SS не поддерживает импульсное выполнение DESQRP.</div>																

Функция

Вычисление квадратного корня числа в двоичном формате с плавающей запятой

Описание

- Из числа с плавающей запятой, заданного в (S), вычисляется корень квадратный.
- Для каждого операнда используется по два следующих друг за другом регистра.
- Задаваемые константы (K,H) 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 нуля (zero) M1020.

Источник ошибки

Если операнд (S) отрицательное число, то активизируется флаг ошибки M1067.

Пример 1: программирование DESQR-инструкции

При включении входа X0 вычисляется корень квадратный из числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10).



Пример 2: программирование DESQR-инструкции

При включении входа X0 вычисляется корень квадратный из константы K1234. Результат сохраняется в (D11, D10).



API	-	POW		<div><div>S1</div><div>S2</div><div>D</div></div>			Степень числа в формате с плавающей запятой	DVP-		
128	D		P	ES/EX/SS	SA/SX	EH				
								+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---	
	S ₁					*	*						*			32-х битная инструкция	
	S ₂					*	*						*			(13 шагов)	
D													*			DPOW - Непрерывное	
Примечания:																	выполнение.
Эта инструкция может работать только 32-х битном режиме (DPOW, DPOWP).																	DPOWP - Импульсное
Серия ES/EX/SS не поддерживает импульсное выполнение DPOWP.																	выполнение.
																	Флаги: нет

Функция

Возведение числа в степень в двоичном формате с плавающей запятой

Описание

- Число (S1) возводится в степень (S2). Результат запоминается в (D).

$$[S1]^{[S2]} = [D]$$

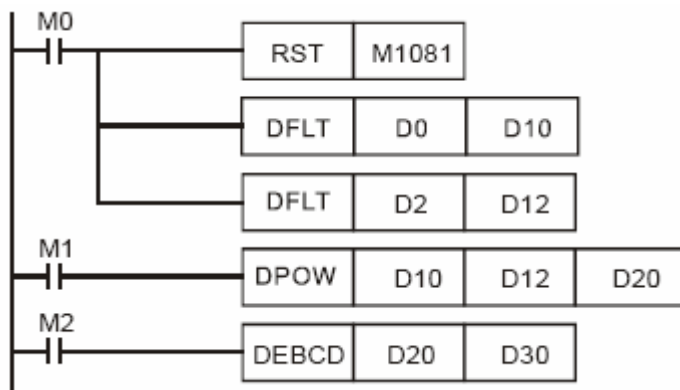
- Для каждого операнда используется по два следующих друг за другом регистра.
- Задаваемые константы (K, H) перед операцией автоматически преобразовываются в числа с плавающей запятой.
- Может применяться один и тот же операнд для источника и для цели. В этом случае рассчитанный результат снова запоминается в операнде-источнике и может использоваться для следующего расчета. Этот процесс повторяется в каждом цикле программы.
- Операнды (S1) и (S2) могут иметь только положительные значения.
- Если результат равен 0, то включается флаг ноля (zero) M1020.
- Если результат больше, чем максимально допустимое значение, то включается флаг переноса (Carry) M1022.
- Если результат меньше, чем минимально допустимое значение, то включается флаг заимствования (Borrow) M1021.

Пример: программирование DPOW-инструкции

Когда M0=1, целое число, записанное в D0, D1 конвертируется в двоичное число с плавающей запятой и записывается в (D10, D11); целое число, записанное в D2, D3 конвертируется в двоичное число с плавающей запятой и записывается в (D12, D13)

Когда M1=1, число (D10, D11) возводится в степень (D12, D13) и результат записывается в (D20, D21).

Когда M2=1, двоичное число с плавающей запятой, записанное в (D20, D21) конвертируется в десятичное число с плавающей запятой и записывается в (D30, D31).



API		INT		S D	Преобразование числа в двоичном формате с плавающей запятой в целое	DVP-		
						ES/EX/SS	SA/SX	EH
129	D		P			+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (5 шагов) INT - Непрерывное выполнение. INTP - Импульсное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S													*				
D													*				
<div>Примечания:</div> <div>Серия ES/EX/SS не поддерживает импульсное выполнение DESQRP.</div>																32-х битная инструкция (9 шагов) DINT - Непрерывное выполнение. DINTP - Импульсное выполнение. Флаги: M1020 (флаг ноля), M1021 (флаг заема), M1022 (флаг переноса)	

Функция

Преобразование числа в двоичном формате с плавающей запятой в целое

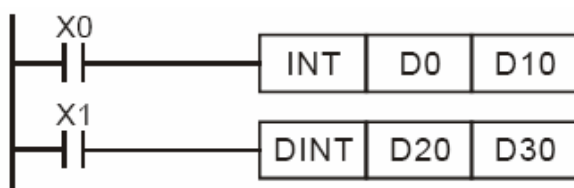
Описание

- 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 (API 49).
- Если результат конвертирования является 0, то включается флаг нуля (zero) M1020.
- Если в (S) указывается не целое число, то это число округляется до ближайшего меньшего целого значения и включается флаг заимствования (borrow) M1021.
- Если конвертируемое целое значение находится вне области памяти операнда цели, появляется превышение и включается флаг переноса (carry) M1022.

Пример: программирование INT-инструкции

При включении входа X0 число с плавающей запятой в (D0, D1) округляется до ближайшего меньшего целого значения. Результат сохранится в D10.

При включении входа X1 число с плавающей запятой в (D20, D21) округляется до ближайшего меньшего целого значения. Результат сохранится в (D30, D31).



API	-	SIN		S D	Синус числа в формате с плавающей запятой	DVP-		
130	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

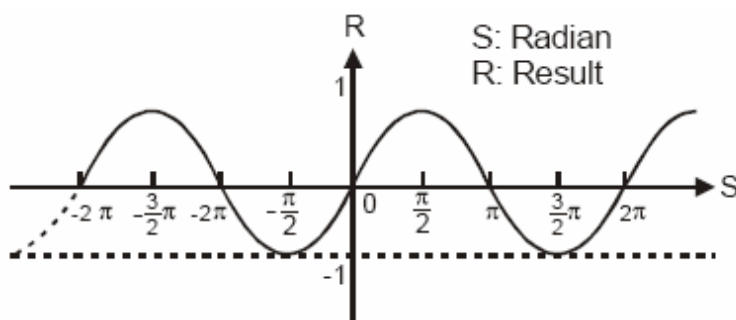
Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S					*	*							*			32-х битная инструкция
D													*			(9 шагов)
<p><u>Примечания:</u> Необходимое условие: $0^0 \leq (S) < 360^0$</p> <p>Эта инструкция может работать только 32-х битном режиме (DSIN, DSINP). Серия ES/EX/SS не поддерживает импульсное выполнение DSINP.</p>																<p>DSIN - Непрерывное выполнение. DSINP - Импульсное выполнение.</p> <p>Флаги: M1018 (флаг для радианов/градусов)</p>

Функция

Расчет синуса числа в двоичном формате с плавающей запятой

Описание

- Рассчитывается синус числа с плавающей запятой, заданного в (S). Результат запоминается в (D).
- Для каждого операнда применяется соответственно два 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 2π радианами, если M1018 = 0. Значение угла, указанного в (S), должно быть углом между 0 и 360 градусами, если M1018 = 1.



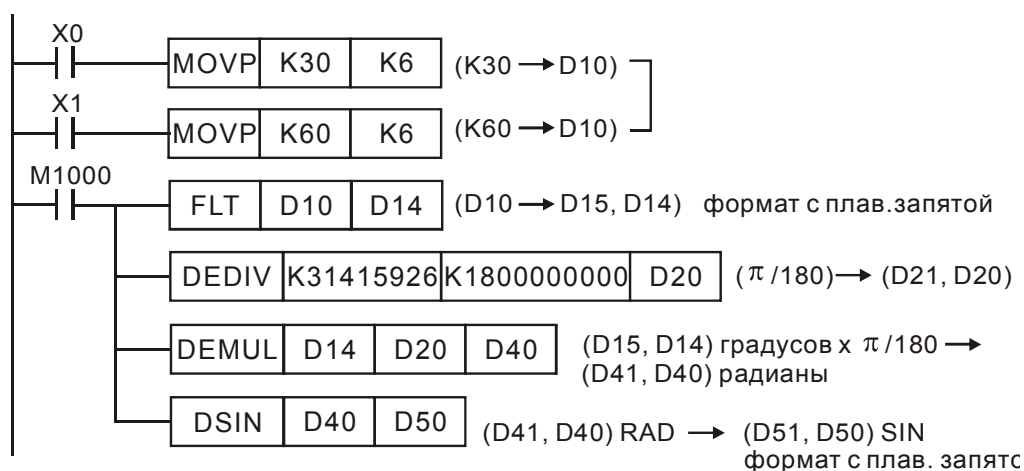
Пример 1: программирование DSIN-инструкции

При включении входа X0 рассчитывается синус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10). Так как M1018 = 0: вычисление выполняется в радианах.

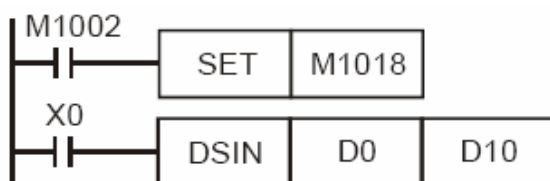


Пример 2: программирование DSIN-инструкции

Входами X0 и X1 выбирается угол в радианах, конвертируется в двоичный формат с плавающей запятой и рассчитывается синус заданного угла. Результат сохраняется в (D51, D50).

**Пример 3:** программирование DSIN-инструкции

При включении входа X0 рассчитывается синус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10). Так как M1018 = 1: вычисление выполняется в градусах.



API	-	COS			Косинус числа в формате с плавающей запятой	DVP-		
131	D		P	S D		ES/EX/SS	SA/SX	EH
						+	+	+

Оп- код	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			
D													*			
<p><u>Примечания:</u> Необходимое условие: $0^0 \leq (S) < 360^0$ Эта инструкция может работать только 32-х битном режиме (DCOS, DCOSP). Серия ES/EX/SS не поддерживает импульсное выполнение DCOSP.</p>															<p><u>32-х битная инструкция</u> (9 шагов) DCOS - Непрерывное выполнение. DCOSP - Импульсное выполнение.</p> <p>Флаги: M1018 (флаг для радианов/градусов)</p>	

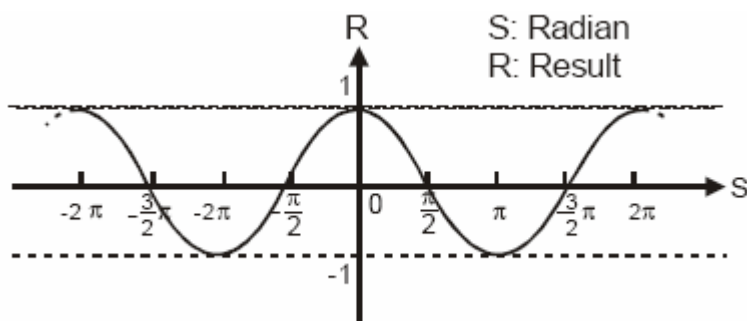
Функция

Расчет косинуса числа в двоичном формате с плавающей запятой

Описание

- Рассчитывается косинус числа с плавающей запятой, заданного в (S). Результат запоминается в (D).

- Для каждого операнда применяется соответственно два 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 2π радианами, если $M1018 = 0$. Значение угла, указанного в (S), должно быть углом между 0 и 360 градусами, если $M1018 = 1$.



Пример 1: программирование DCOS-инструкции

При включении входа X0 рассчитывается косинус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10). Так как $M1018 = 0$: вычисление выполняется в радианах.



Пример 2: программирование DCOS-инструкции

При включении входа X0 рассчитывается косинус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10). Так как $M1018 = 1$: вычисление выполняется в градусах.



API	-	TAN		S D	Тангенс числа в формате с плавающей запятой	DVP-		
132	D		P			ES/EX/SS	SA/SX	EH
						+	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S					*	*							*			32-х битная инструкция (9 шагов) DTAN - Непрерывное выполнение. DTANP - Импульсное выполнение. Флаги: M1018 (флаг для радианов/градусов)
D													*			

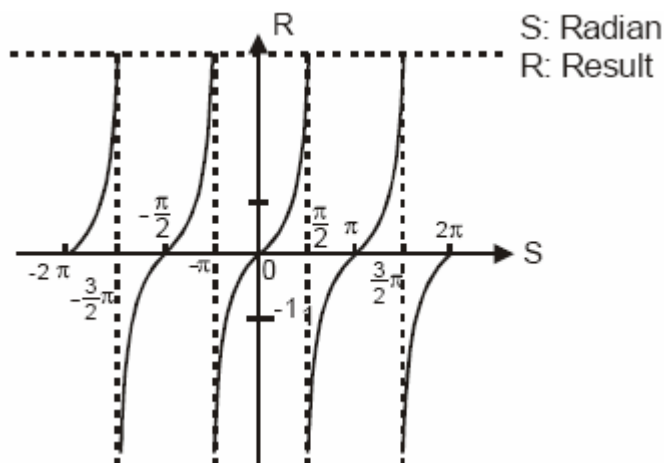
Примечания: Необходимое условие: $0^0 \leq (S) < 360^0$
 Эта инструкция может работать только 32-х битном режиме (DTAN, DTANP).
 Серия ES/EX/SS не поддерживает импульсное выполнение DTANP.

Функция

Расчет тангенса числа в двоичном формате с плавающей запятой

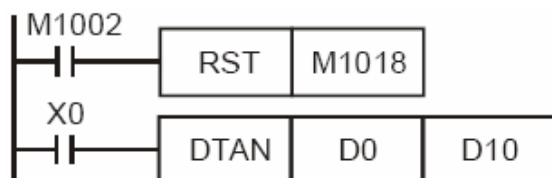
Описание

- Рассчитывается тангенс числа с плавающей запятой, заданного в (S). Результат запоминается в (D).
- Для каждого операнда применяется соответственно два 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 2π радианами, если M1018 = 0. Значение угла, указанного в (S), должно быть углом между 0 и 360 градусами, если M1018 = 1.



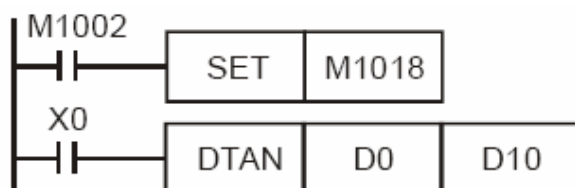
Пример 1: программирование DTAN-инструкции

При включении входа X0 рассчитывается тангенс числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10). Так как M1018 = 0: вычисление выполняется в радианах.



Пример 2: программирование DTAN-инструкции

При включении входа X0 рассчитывается тангенс числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10). Так как M1018 = 1: вычисление выполняется в градусах.



API	-	ASIN		S	D	Арксинус числа в формате с плавающей запятой	DVP-		
133	D		P				ES/EX/SS	SA/SX	EH
							-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S					*	*							*			32-х битная инструкция (9 шагов) DASIN - Непрерывное выполнение. DASINP - Импульсное выполнение.
D													*			
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DASIN, DASINP).																Флаги: нет

Функция

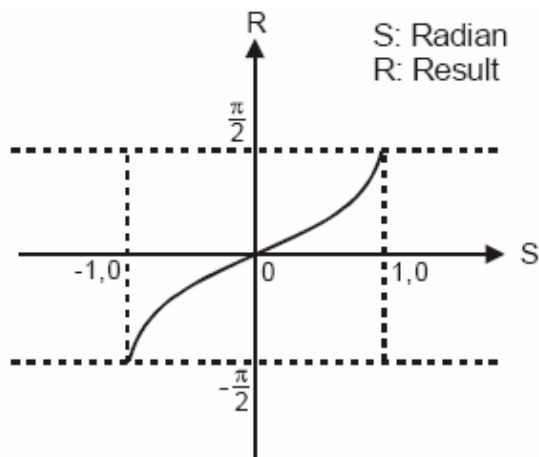
Расчет арксинуса числа в двоичном формате с плавающей запятой

Описание

- Рассчитывается арксинус числа с плавающей запятой, заданного в (S). Результат запоминается в (D).

$$ASIN = SIN^{-1}$$

- Для каждого операнда применяется соответственно два следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.



Пример: программирование DASIN-инструкции

При включении входа X0 рассчитывается арксинус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10).



API	-	ACOS			Arccosine of a number in floating-point format	DVP-		
134	D		P	S D	с плавающей запятой	ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			
D													*			
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DACOS, DACOSP).															<u>32-х битная инструкция</u> (9 шагов) DACOS - Непрерывное выполнение. DACOSP - Импульсное выполнение. Флаги: нет	

Функция

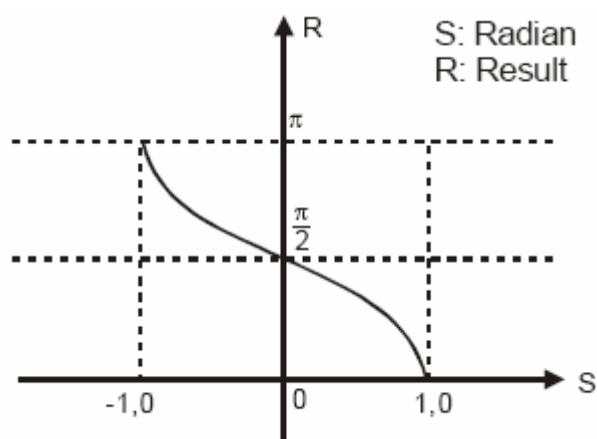
Расчет арккосинуса числа в двоичном формате с плавающей запятой

Описание

- Рассчитывается арккосинус числа с плавающей запятой, заданного в (S). Результат запоминается в (D).

$$\text{ACOS} = \text{COS}^{-1}$$

- Для каждого операнда применяется соответственно два следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.

**Пример:** программирование DACOS-инструкции

При включении входа X0 рассчитывается арккосинус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10).



API	-	ATAN	P	S D	Арктангенс числа в формате с плавающей запятой	DVP-		
						ES/EX/SS	SA/SX	EH
135	D					-	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S					*	*							*			32-х битная инструкция (9 шагов)
D													*			DATAN - Непрерывное выполнение. DATANP - Импульсное выполнение.
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DATAN, DATANP).																Флаги: нет

Функция

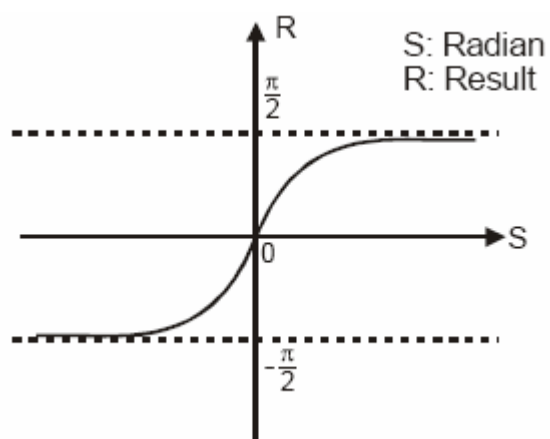
Расчет арктангенса числа в двоичном формате с плавающей запятой

Описание

- Рассчитывается арктангенс числа с плавающей запятой, заданного в (S). Результат запоминается в (D).

$$\text{ATAN} = \text{TAN}^{-1}$$



- Для каждого операнда применяется соответственно два следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.



Пример: программирование DATAN-инструкции

При включении входа X0 рассчитывается арктангенс числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10).



API	-	SINH	P	 	Гиперболический синус числа в формате с плавающей запятой	DVP-		
						ES/EX/SS	SA/SX	EH
136	D					-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S					*	*							*			32-х битная инструкция (9 шагов)
D													*			DSINH - Непрерывное выполнение. DSINHP - Импульсное выполнение.
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DSINH, DSINHP).																Флаги: нет

Функция

Расчет гиперболического синуса числа в двоичном формате с плавающей запятой

Описание

- Рассчитывается гиперболический синус числа с плавающей запятой, заданного в (S). Результат запоминается в (D).



$$\text{SINH} = (e^s - e^{-s})/2$$

- Для каждого операнда применяется соответственно два следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.

Пример: программирование DSINH-инструкции

При включении входа X0 рассчитывается гиперболический синус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10).



API	-	COSH	P	 	Гиперболический косинус числа в формате с плавающей запятой	DVP-		
						ES/EX/SS	SA/SX	EH
137	D					-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S					*	*							*			32-х битная инструкция
D													*			(9 шагов)
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DCOSH, DCOSHP).																DCOSH - Непрерывное выполнение.
																DCOSHP - Импульсное выполнение.
																Флаги: нет

Функция

Расчет гиперболического косинуса числа в двоичном формате с плавающей запятой

Описание

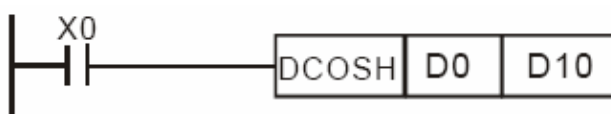
- Рассчитывается гиперболический косинус числа с плавающей запятой, заданного в (S). Результат запоминается в (D).

$$\text{COSH} = (e^s + e^{-s})/2$$

- Для каждого операнда применяется соответственно два следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.

Пример: программирование DCOSH-инструкции

При включении входа X0 рассчитывается гиперболический косинус числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10).



API	-	TANH	P	S	D	Гиперболический тангенс числа в формате с плавающей запятой	DVP-		
							ES/EX/SS	SA/SX	EH
138	D						-	+	+

Операнд	Биты				Слова											16-ти битная инструкция ---
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			
D													*			
<u>Примечания:</u> Эта инструкция может работать только 32-х битном режиме (DTANH, DTANHP).																32-х битная инструкция (9 шагов) DTANH - Непрерывное выполнение. DTANHP - Импульсное выполнение. Флаги: нет

Функция

Расчет гиперболического тангенса числа в двоичном формате с плавающей запятой

Описание

- Рассчитывается гиперболический тангенс числа с плавающей запятой, заданного в (S). Результат запоминается в (D).

$$\text{TANH} = (e^s - e^{-s}) / (e^s + e^{-s})$$

- Для каждого операнда применяется соответственно два следующих друг за другом регистра.
- Значения операндов источника и цели имеют формат с плавающей запятой.

Пример программирования DTANH-инструкции

При включении входа X0 рассчитывается гиперболический тангенс числа с плавающей запятой в (D1, D0). Результат сохраняется в (D11, D10).



API				Общая команда генерации импульсов с ШИМ	DVP-		
144	GPWM	(S1)	(S2)	(D)	ES/EX/SS	SA/SX	EH
					-	+	+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (7 шагов) GPWM - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S ₁													*			<u>32-х битная инструкция</u> --- Флаги: нет	
S ₂													*				
D		*	*	*													
<u>Примечания:</u> Операнд S2 занимает три адреса Операнд S1 должен быть меньше или равен S2																	

Функция

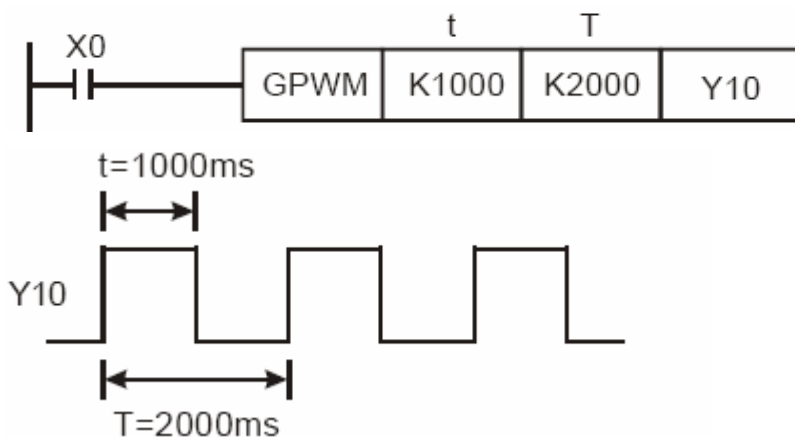
Генерация на выходе ПЛК импульсов с заданной шириной и периодом.

Описание

- На выходе (D) генерируются импульсы с шириной заданной в (S1) и периодом в (S2)
- Ширина импульса (S1) может быть задана в диапазоне: $t = 0 \dots 32767$ мс
- Период следования импульсов (S2) может быть задан в диапазоне: $T = 1 \dots 32767$ мс, причем должно выполняться условие: $S1 \leq S2$
- Регистры ((S1)+1) и ((S2)+1) используются системой. Не используйте их в программе!
- Если: $S1 \leq 0$, импульсы генерироваться не будут
- Если: $S1 \geq S2$, выход (D) будет постоянно включен
- Значения (S1) и (S2) можно изменять во время выполнения инструкции.

Пример программирования GPWM-инструкции

Когда вход X0 включен, на выходе Y10 будут генерироваться импульсы длительностью 1сек через каждые 2 сек. Если X0=0, импульсы генерироваться не будут.



Примечания:

Эта инструкция работает в цикле программы. Значения S1, S2 и S1-S2 должны быть больше времени одного цикла выполнения программы. Иначе будет ошибка.

Если GPWM-инструкция записана в подпрограмме, она может работать не правильно.

API	FTC	(S1) (S2) (S3) (D)	Температурный контроллер	DVP-		
				ES/EX/SS	SA/SX	EH
145				-	+	+

Оп- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (9 шагов) FTC - Непрерывное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			
S ₁													*			<u>32-х битная инструкция</u> --- Флаги: нет	
S ₂													*				
S ₃													*				
D													*				
Примечания: Операнд S ₃ использует 6 последовательных адресов.																	

Функция

Поддержание заданной температуры в контуре замкнутого регулирования по алгоритму с нечеткой (fuzzy) логикой.

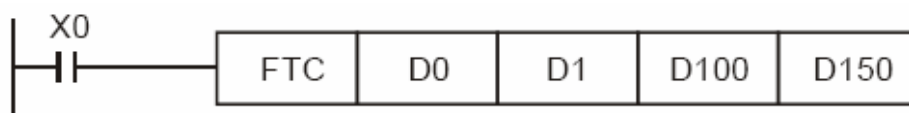
Описание

- (S1) указывает заданное значение контура регулирования (SV).
Диапазон: 1 ... 5000, что соответствует 0.1 °C ... 500 °C
- (S2) считывает действительное значение как обратную связь для регулятора (PV).
Диапазон: 1 ... 5000, что соответствует 0.1 °C ... 500 °C
- (S3) является начальным адресом области регистра, в которой запоминаются параметры контура регулирования. Для этого имеется 6 один за другим следующих регистров данных. Таблица содержит обзор параметров.
 - В (D) записываются расчетные выходные значения, которые выдаются по процессу регулирования (MV). Диапазон: 0 ... 100, что соответствует 0% ... 100%
 - В программе FTC-инструкцию можно применять неограниченное число раз, однако адреса регистров операнда (S3) не должны повторяться.

Пример программирования FTC-инструкции

FTC-инструкция начнет выполняться, как только включится X0. Задаваемое значение записано в D0, текущее значение считывается в D1, выходные значения записываются в D150. Параметры регулирования хранятся в регистрах данных D100...D105.

Если X0 выключен, FTC-инструкция выполняться не будет и значения в регистрах останутся неизменными.

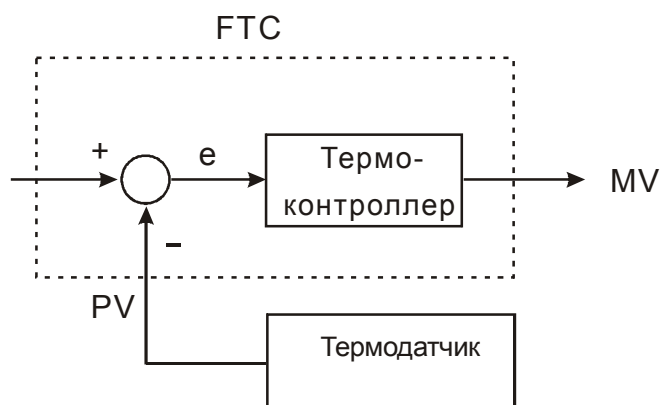


До начала выполнения FTC-инструкции установите параметры регулирования в D100, D101

Таблица параметров регулирования (S3) для команды FTC:

Адрес регистра	Функция	Значения	Описание
(S3)	Время выборки (T_s)	1~200 (ед.изм: 100мс)	Интервал проверки процесса изменения действительного значения. Если T_s меньше времени скана, FTC-инструкция будет выполняться в каждом программном цикле. Если $T_s = 0$, FTC-инструкция не будет выполняться.
(S3) + 1	b0: единицы измерения температуры; b1: функция фильтра; b2-b5: выбор режима работы нагревателя; b6-b15: зарезервированы	b0=0: $^{\circ}\text{C}$ b0=1: $^{\circ}\text{F}$	Значение по умолчанию: K0
		b1=0: нет фильтра b1=1: есть фильтр	Без фильтра: $PV = \text{текущему измеренному значению}$. С фильтром: $PV = (\text{текущее измеренное значение} + \text{предыдущее измеренное значение})/2$
		b2 = 1	Медленный нагрев
		b3 = 1	Нормальный нагрев
		b4 = 1	Быстрый нагрев
		b5 = 1	Очень быстрый нагрев
(S3) + 2 ... (S3) + 5	Используются системой. Не используйте их!		

Схема FTC-регулятора:



Пунктирной линией выделена FTC-инструкция.

Замечания по настройке:

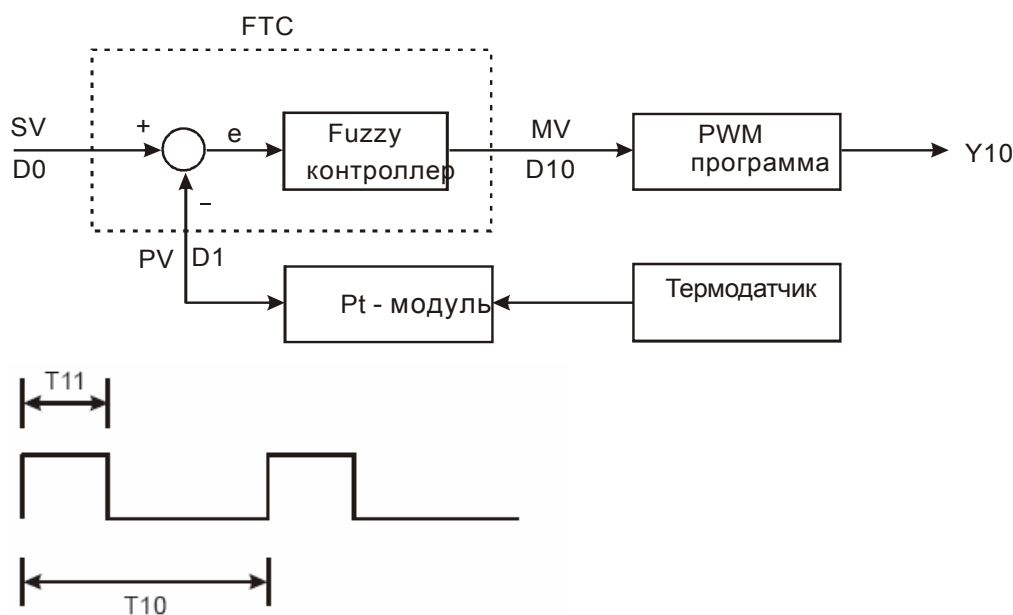
Модули аналогового ввода типа DVP-04AD / DVP-04XA / DVP-04PT / DVP-04TC, преобразуют аналоговые данные в цифровые, т.е. оцифровываются аналоговые данные процесса. Для стабильной работы интервал времени выборки (T_s) должен быть выбран больше, чем двукратное время преобразования сигналов в указанных модулях. Базовое устройство (ПЛК) применяет FROM-инструкцию, чтобы занести эти данные в (S2).

Значения параметра (S3)+1 b2-b5, определяют скорость реакции регулятора на изменения измеренного значения. По умолчанию установлена нормальная скорость нагрева. Если система регулирования не имеет резких изменений температуры, можно выбрать режим медленного нагрева и тем самым увеличить точность поддержания

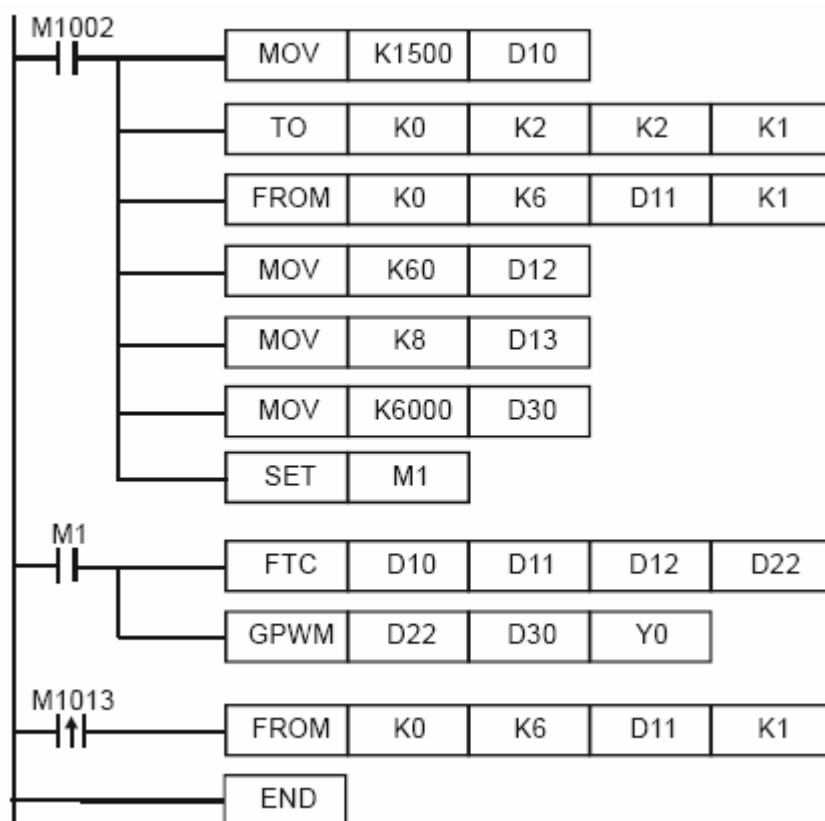
заданной температуры. Напротив, если внешняя среда имеет чрезмерно колебательный характер, нужно выбрать режим быстрого нагрева и тем самым реагировать на быстрые изменения температуры.

Пример применения:

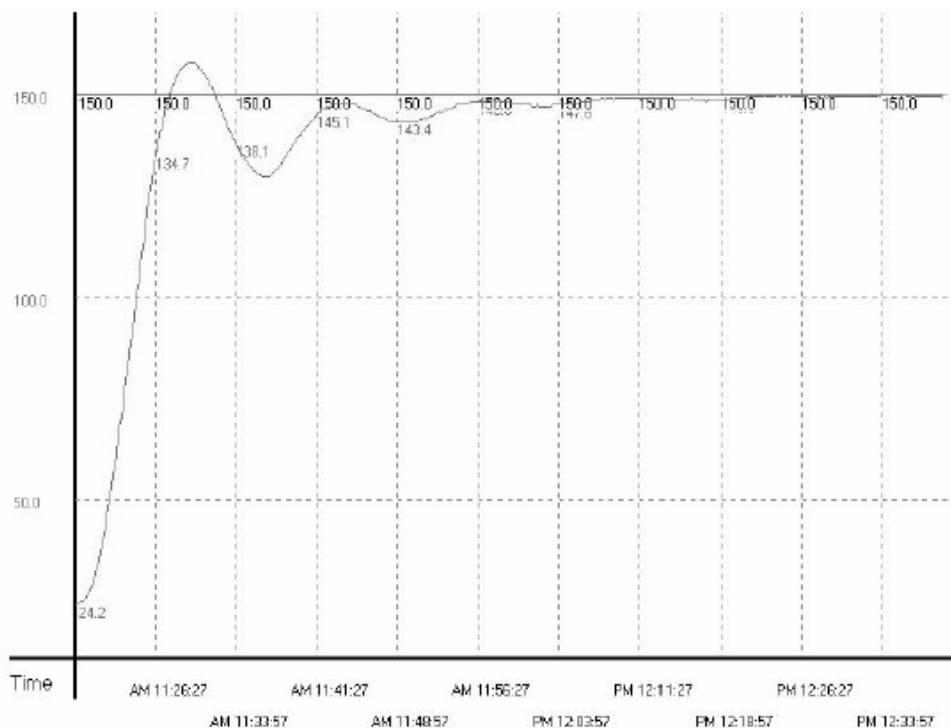
FTC-инструкцию удобно использовать совместно с GPWM-инструкцией, выходное значение (MV) FTC-инструкции будет определять ширину импульсов (t) GPWM-инструкции и тем самым регулировать выходную мощность на нагревательном элементе.



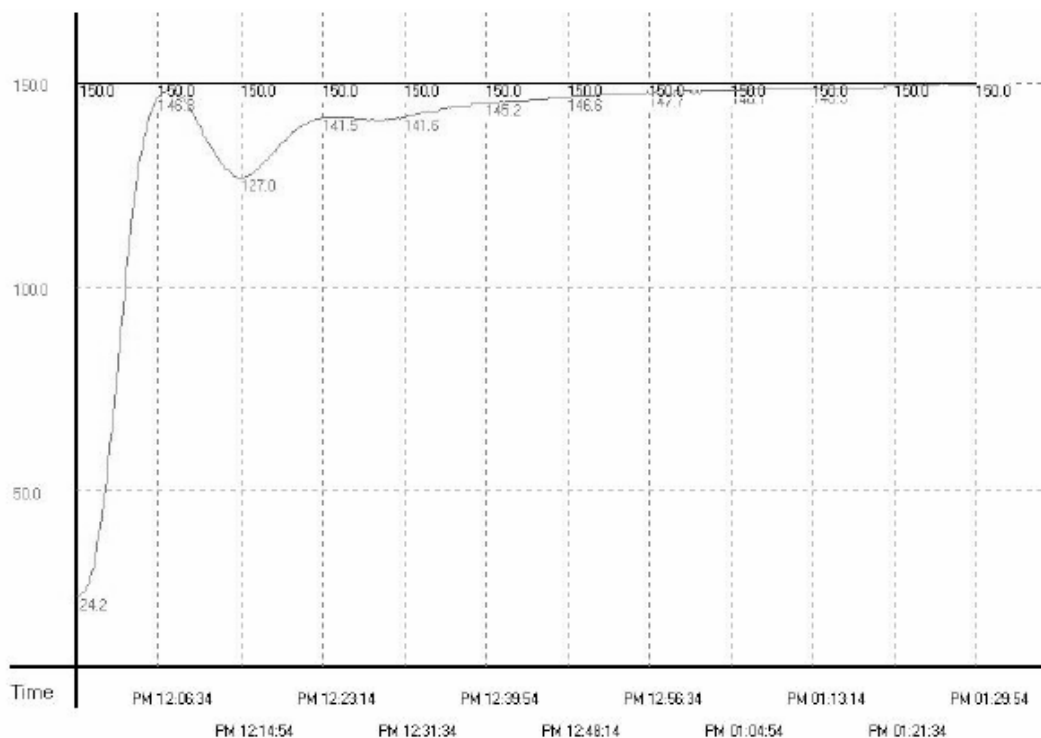
Заданное значение температуры D10=K1500 (150 °C). Время выборки установлено в D12 = K60 (6 сек); D13=K8 (bit 3 = 1); D30 = K6000 (=D12x100). Программа показана ниже:



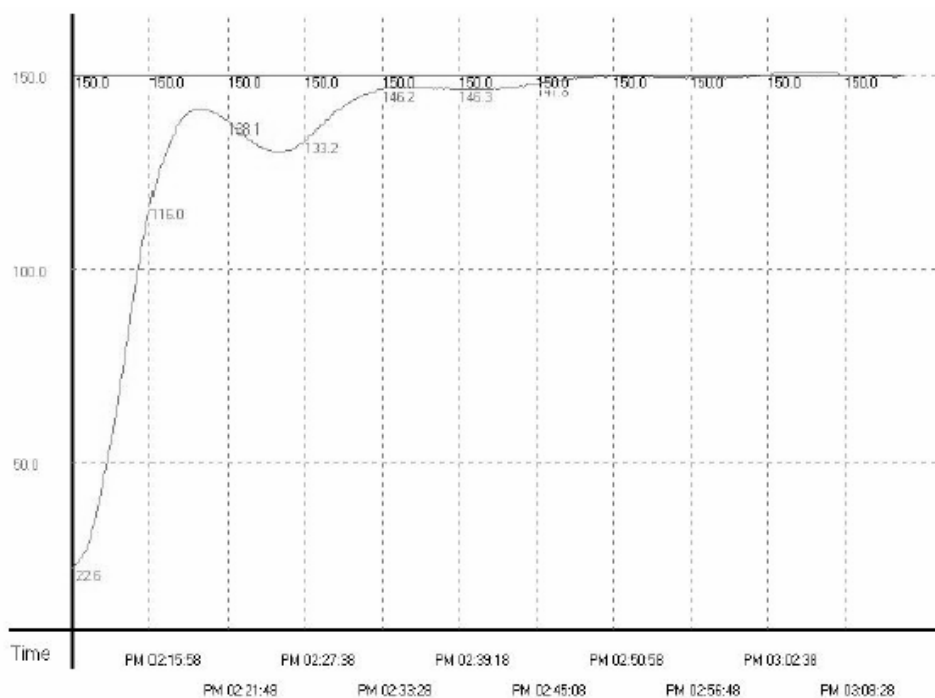
Эксперимент 1: Следим за нагревом печи и отмечаем заданную и текущую температуру. На зафиксированном графике изменения температуры видно перерегулирование в первом цикле примерно на 10°C . Температура достигает заданной и устанавливается с точностью $\pm 1^{\circ}\text{C}$ примерно за 48 мин:



Эксперимент 2: Изменим режим нагрева на быстрый (D13=K16). На зафиксированном графике изменения температуры видно что перерегулирования не стало. Температура достигает заданной и устанавливается с точностью $\pm 1^{\circ}\text{C}$ примерно за 1 час 15 мин:

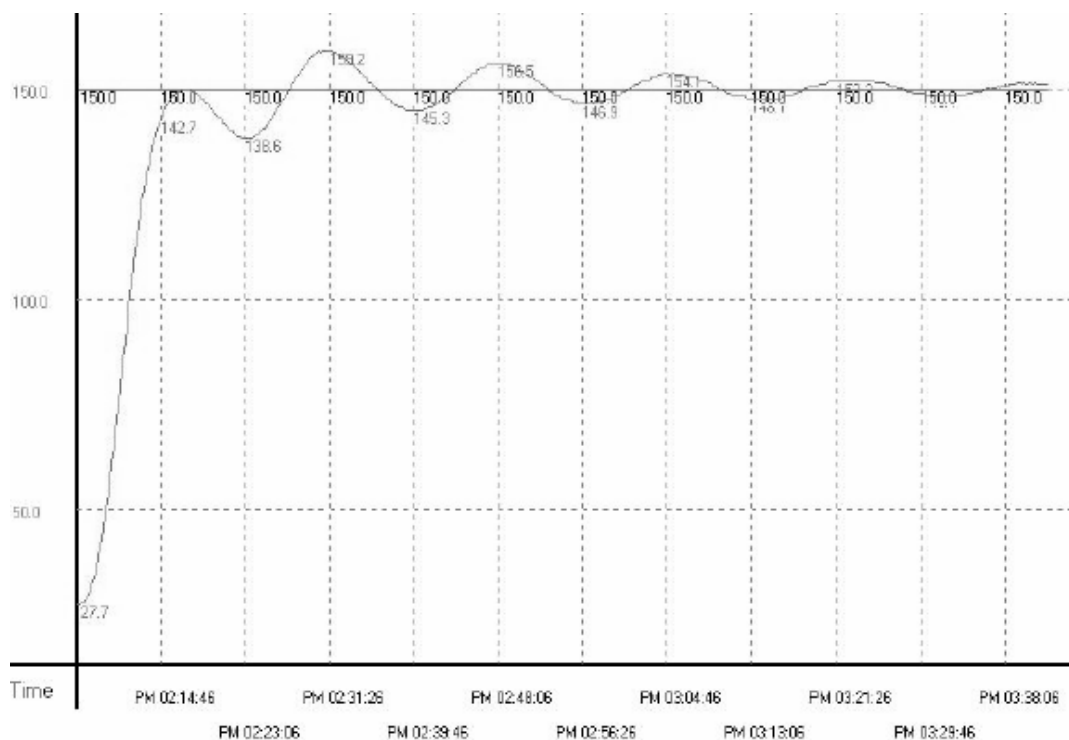


Эксперимент 3: Изменим время выборки (D12=K40) и период импульсов ШИМ (D30=K4000). На зафиксированном графике изменения температуры видно, что температура стала достигать заданной быстрее и устанавливается с точностью $\pm 1^{\circ}\text{C}$ примерно за 37 мин:



Эксперимент 4: Попробуем еще изменить время выборки (D12=K20) и период импульсов ШИМ (D30=K2000). На зафиксированном графике изменения температуры

видно, что температура не стала выходить на заданную, а стала иметь постоянный колебательный характер с перерегулированием и система не стабильна:



Вывод: система имела стабильность, точность и оптимальную динамику с параметрами заданными в эксперименте 3.

API		SWAP			Перестановка байтов в регистре								DVP-		
147	D		P										ES/EX/SS	SA/SX	EH
													+	+	+

Оп- ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S								*	*	*	*	*	*	*	*

Примечания:

Если в качестве операнда (D) используется регистр F, возможно только 16-ти битное выполнение инструкции.

В серии ES/EX/SS инструкция SWAP и DSWAP возможна только начиная с версии v.4.9

16-ти битная инструкция
(5 шагов)
SWAP - Непрерывное выполнение.
SWAPP - Импульсное выполнение.

32-х битная инструкция
(9 шагов)
DSWAP - Непрерывное выполнение.
DSWAPP - Импульсное выполнение.

Флаги: нет

Функция

Перестановка местами старшего и младшего байтов операнда

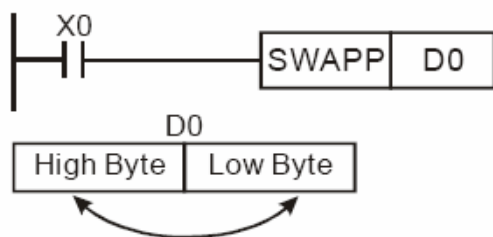
Описание

- При применении SWAP-инструкции старший и младший байты операнда (D) обмениваются местами.

- При применении DSWARE-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.

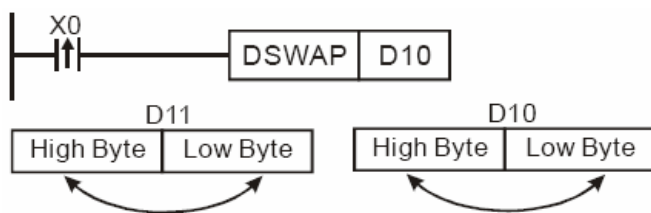
Пример программирования SWAPP-инструкции

При включении входа X0 старший и младший байты регистра D0 обмениваются местами.



Пример программирования DSWAP-инструкции

При включении входа X0 старший и младший байты регистров D10 и D11 обмениваются местами.



API		MEMR		m D n	Чтение данных из файловых регистров	DVP-		
148	D		P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (7 шагов) MEMR - Непрерывное выполнение. MEMRP - Импульсное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
m					*	*							*				
D													*				
n					*	*							*				
<u>Примечания:</u> Диапазон операнда (m): SA/SX – m=0...1599; EH – m=0...9999 Диапазон операнда (D): SA/SX – D2000...D4999; EH – D2000...D9999 Диапазон операнда (n): 16-бит: SA/SX – n=1...1600; EH – n=1...8000 32-бит: SA/SX – n=1...800; EH – n=1...4000																<u>32-х битная инструкция</u> (13 шагов) DMEMR - Непрерывное выполнение. DMEMRP - Импульсное выполнение. Флаги: M1101	

Функция

Копирование данных из файловых регистров в регистры данных общего назначения

Описание

(m) – начальный адрес считываемых файловых регистров

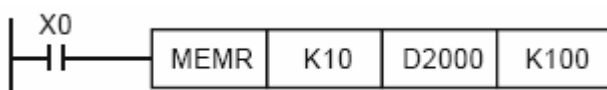
(D) – начальный адрес регистров данных для записи считанных файловых регистров

(n) – количество считываемых данных

- Операнды (m) и (n) в контроллерах SA/SX не поддерживают индексацию E и F
- Если операнды (m), (D) или (n) выйдут из своего диапазона, будет зафиксирована в D1067 ошибка с кодом 0E1A и включаться флаги M1067, M1068.

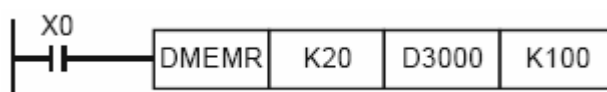
Пример программирования MEMR-инструкции

Когда вход X0 включен, будут считываться данные из ста файловых регистров начиная с адреса 10 и помещаться в регистры данных D2000 – D2099. Если X0 выключить, когда инструкция полностью не выполнена, несчитанные регистры останутся неизменными.



Пример программирования DMEMR-инструкции

Когда вход X0 включен, будут считываться 32-х битные данные из ста файловых регистров начиная с адреса 20 и помещаться в регистры данных D3000 – D2199. Если X0 выключить, когда инструкция полностью не выполнена, несчитанные регистры останутся неизменными.



API		MEMW		<div>S</div> <div>m</div> <div>n</div>			Запись данных в файловые регистры							DVP-		
149	D		P								ES/EX/SS			SA/SX		EH
											-			+		+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (7 шагов) MEMW - Непрерывное выполнение. MEMWP - Импульсное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
	S												*					
	m				*	*							*					
	n				*	*							*					

Примечания:
Диапазон операнда (S): SA/SX – D2000...D4999; EH – D2000...D9999
Диапазон операнда (m): SA/SX – m=0...1599; EH – m=0...9999
Диапазон операнда (n):
16-бит: SA/SX – n=1...1600; EH – n=1...8000
32-бит: SA/SX – n=1...800; EH – n=1...4000

32-х битная инструкция
(13 шагов)
DMEMW - Непрерывное
выполнение.
DMEMWP - Импульсное
выполнение.

Флаги: M1101

Функция

Копирование данных из регистров данных общего назначения в файловые регистры

ОПИСАНИЕ

(S) – начальный адрес копируемых регистров данных

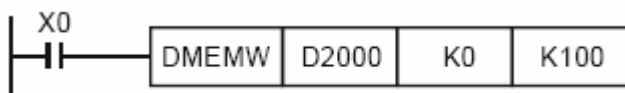
(m) – начальный адрес файловых регистров в которые будут записываться данные

(n) – количество записываемых данных

- Операнды (m) и (n) в контроллерах SA/SX не поддерживают индексацию E и F
- Если операнды (m), (D) или (n) выйдут из своего диапазона, будет зафиксирована в D1067 ошибка с кодом 0E1A и включатся флаги M1067, M1068.

Пример программирования DMEW-инструкции

Когда вход X0 включен, будут копироваться 32-х битные данные из регистров D2000...D2199 в файловые регистры с адресами 0...199. Если X0 выключить, когда инструкция полностью не выполнена, несчитанные регистры останутся неизменными.



Файловые регистры

Файловые регистры есть в ПЛК серий SA/SX/EH. Они отличаются от регистров общего назначения тем, что к ним не возможно прямое обращение, а только с помощью команд MEMR (API 147), MEMW (API 148) или программатора, или WPLSoft.

Область чтения и записи данных в файловый регистр задается в командах MEMR (API 147), MEMW (API 148).

Для автоматической передачи данных при включении ПЛК или переключении его из режима STOP в режим RUN область чтения и записи данных в файловый регистр может быть задана в специальных регистрах:

D1101 определяет стартовый адрес файлового регистра (в DVP-SA/SX: K0...K1599; в DVP-EH: K0...K9999).

D1102 определяет число читаемых файловых регистров (в DVP-SA/SX: K0...K1600; в DVP-EH: K0...K10000).

D1103 определяет стартовый адрес для сохранения в файловый регистр (должен быть больше 2000).

M1101 разрешает/запрещает запись/чтение файловых регистров.

Если адрес файлового регистра при чтении выходит из пользовательского диапазона, в назначенные регистры будут записаны ноли.

9. ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 150-199

API		MODRW		<div><div>S₁</div><div>S₂</div><div>S₃</div><div>S</div><div>n</div></div>										Чтение/запись данных MODBUS				DVP-		
150															ES/EX/SS		SA/SX	EH		
															+		+	+		

Оп- ранд	Биты				Слова												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S ₁					*	*							*				
S ₂					*	*							*				
S ₃					*	*							*				
S													*				
n					*	*							*				

Примечания:

Диапазоны операндов: S₁ = K0...K255
S₂ = K3(H3) или K6(H6) или K16(H10)
n = K1 ... K8

16-ти битная инструкция
(11 шагов)
MODRW - Непрерывное
выполнение.

32-х битная инструкция

Флаги:
M1120 – M1131,
M1140 – M1143 (см.
инструкцию RS)

Функция

Обмен данными (чтение/запись) с внешними устройствами, поддерживающих протокол MODBUS (ASCII или RTU) по интерфейсу RS-485.

Описание

С помощью MODRW-инструкции возможно чтение и запись данных от преобразователей частоты серии VFD (за исключение VFD-A). См. так же инструкции на преобразователи VFD.

(S₁) – коммуникационный адрес устройства в сети

(S₂) – функциональный код команды. Например, для VFD-S доступные командные коды: 03H – чтение n слов; 06H – запись 1 слова; 10H – запись n слов

(S₃) – адрес данных внешнего устройства. Например, для VFD-S адрес данных 2001H: команда задания выходной частоты (Гц)

(S) – начальный адрес регистров источника данных (при записи) или регистров для сохранения считываемых данных

(n) – длина считываемых/записываемых данных

- Если адрес данных установлен неверно, то будет записан соответствующий код ошибки в спец. регистре D1130, пока M1141 будет включено.
- Нельзя использовать в качестве условия выполнения MODRW-инструкции импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF). Иначе данные будут приняты не корректно.
- После M1140 или M1141 = 1, запрос на чтение данных будет отправлен повторно и, если данные будут приняты без ошибок, флаги M1140 и M1141 будут сброшены.

Пример применения 1: чтение n данных (код команды = 03H) из преобразователя частоты VFD-S (ASCII-режим, когда M1143 = 0; RTU-режим, когда M1143 = 1).

Когда X0 включено, данные читаются со стартового адреса H2100 и сохраняются в соответствующих регистрах:

В ASCII-режиме принятые данные сохраняются в 16-ти последовательных регистрах начиная с D0. ПЛК будет автоматически конвертировать принятые данные в Hex-формат и сохранять их в D1296 – D1311. Во время конвертирования будет включено реле M1131.

Пользователь может использовать команды MOV, DMOV или BMOV для копирования данных из D1296 – D1311 в регистры общего назначения для последующей обработки. Другие команды (в ES-серии) для этих регистров не применимы.

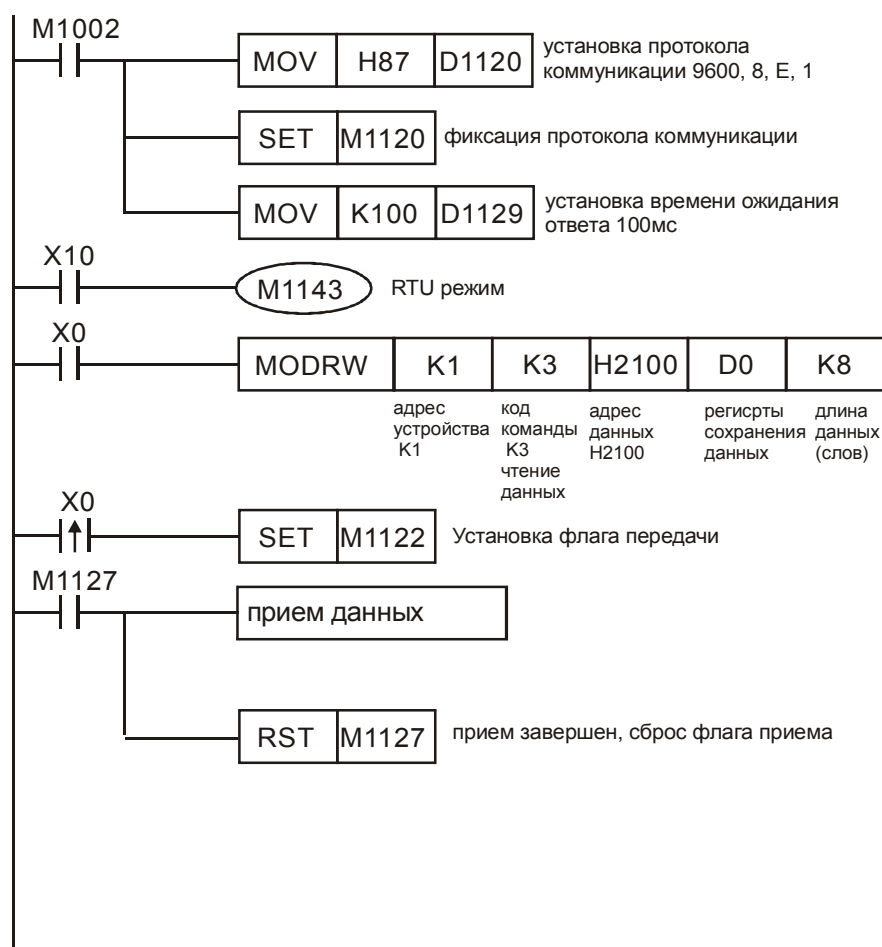
В RTU-режиме принятые данные сохраняются в 8 последовательных регистров начиная с D0 в Hex-формате. В регистры D1296 – D1311 ничего записываться не будет.

В ASCII-режиме и RTU-режиме переданные данные будут сохраняться в регистрах D1256 – D1295. Пользователь может использовать команды MOV, DMOV или BMOV для копирования данных из D1256 – D1295 в регистры общего назначения для последующей обработки. Другие команды для этих регистров не применимы.

Флаг M1129 будет включен, если время ожидания ответа от VFD-S превысит 100 мс.

Флаг M1140 будет включен, если принятые данные будут содержать ошибки.

Флаг M1141 будет включен, если адрес данных будет задан некорректно.



- ASCII-режим

Передаваемое сообщение (ПЛК→VFD-S): "01 03 2100 0006 D5"

Принимаемое сообщ-е (VFD-S→ ПЛК): "01 03 0C 0100 1766 0000 0000 0136 0000 3B"

Передаваемые данные, расписанные по регистрам:

Регистр	Данные			
D1256 младший байт	0	30 Н	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1256 старший байт	1	31 Н	ADR0	
D1257 младший байт	0	30 Н	CMD1	CMD (1, 0) – код команды
D1257 старший байт	3	33 Н	CMD0	
D1258 младший байт	2	32 Н	Стартовый адрес данных	
D1258 старший байт	1	31 Н		
D1259 младший байт	0	30 Н		
D1259 старший байт	0	30 Н		
D1260 младший байт	0	30 Н	Число данных (слов)	
D1260 старший байт	0	30 Н		
D1261 младший байт	0	30 Н		
D1261 старший байт	6	36 Н		
D1262 младший байт	D	44 Н	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1262 старший байт	5	35 Н	LRC CHK0	

Принимаемые данные, расписанные по регистрам:

Регистр	Данные			
D0 младший байт	0	30 H	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D0 старший байт	1	31 H	ADR0	
D1 младший байт	0	30 H	CMD1	CMD (1, 0) – код команды
D1 старший байт	3	33 H	CMD0	
D2 младший байт	0	30 H	Число данных (байт)	
D2 старший байт	C	43 H		
D3 младший байт	0	30 H	Содержимое регистра с адресом 2100H	D1296 = 0100 H
D3 старший байт	1	31 H		
D4 младший байт	0	30 H		
D4 старший байт	0	30 H		
D5 младший байт	1	31 H	Содержимое регистра с адресом 2101H	D1297 = 1766 H
D5 старший байт	7	37 H		
D6 младший байт	6	36 H		
D6 старший байт	6	36 H		
D7 младший байт	0	30 H	Содержимое регистра с адресом 2102H	D1298 = 0000 H
D7 старший байт	0	30 H		
D8 младший байт	0	30 H		
D8 старший байт	0	30 H		
D9 младший байт	0	30 H	Содержимое регистра с адресом 2103H	D1299 = 0000 H
D9 старший байт	0	30 H		
D10 младший байт	0	30 H		
D10 старший байт	0	30 H		
D11 младший байт	0	30 H	Содержимое регистра с адресом 2104H	D1300 = 0136 H
D11 старший байт	1	31 H		
D12 младший байт	3	33 H		
D12 старший байт	6	36 H		
D13 младший байт	0	30 H	Содержимое регистра с адресом 2105H	D1301 = 0000 H
D13 старший байт	0	30 H		
D14 младший байт	0	30 H		
D14 старший байт	0	30 H		
D15 младший байт	3	33 H	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D15 старший байт	B	42 H	LRC CHK0	

- RTU-режим

Передаваемое сообщение (ПЛК→VFD-S): **"01 03 2100 0006 CF F4"**

Принимаемое сообщ-е (VFD-S→ ПЛК): **"01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5"**

Передаваемые данные, расписанные по регистрам:

Регистр	Данные	
D1256 младший байт	01 H	Коммуникационный адрес VFD-S
D1257 младший байт	03 H	Адрес команды
D1258 младший байт	21 H	Адрес данных
D1259 младший байт	00 H	
D1260 младший байт	00 H	Данные
D1261 младший байт	06 H	
D1262 младший байт	CF H	CRC CHK Low
D1263 младший байт	F4 H	CRC CHK High

Принимаемые данные, расписанные по регистрам:

Регистр	Данные	
D0 младший байт	01 H	Коммуникационный адрес VFD-S
D1 младший байт	03 H	Адрес команды
D2 младший байт	0C H	Число данных (байт)
D3 младший байт	00 H	Содержимое регистра с адресом 2100H
D4 младший байт	00 H	
D5 младший байт	05 H	Содержимое регистра с адресом 2101H
D6 младший байт	03 H	
D7 младший байт	0B H	Содержимое регистра с адресом 2102H
D8 младший байт	B8 H	
D9 младший байт	0B H	Содержимое регистра с адресом 2103H
D10 младший байт	B8 H	
D11 младший байт	00 H	Содержимое регистра с адресом 2104H
D12 младший байт	00 H	
D13 младший байт	01 H	Содержимое регистра с адресом 2105H
D14 младший байт	2D H	
D15 младший байт	8E H	CRC CHK Low
D16 младший байт	C5 H	CRC CHK High

Пример применения 2: запись одного слова (код команды = 06H) в преобразователь частоты VFD-S (ASCII-режим, когда M1143 = 0; RTU-режим, когда M1143 = 1)

В ASCII-режиме данные будут передаваться из регистра D50. Ответные данные будут записаны в D1070 – D1076.

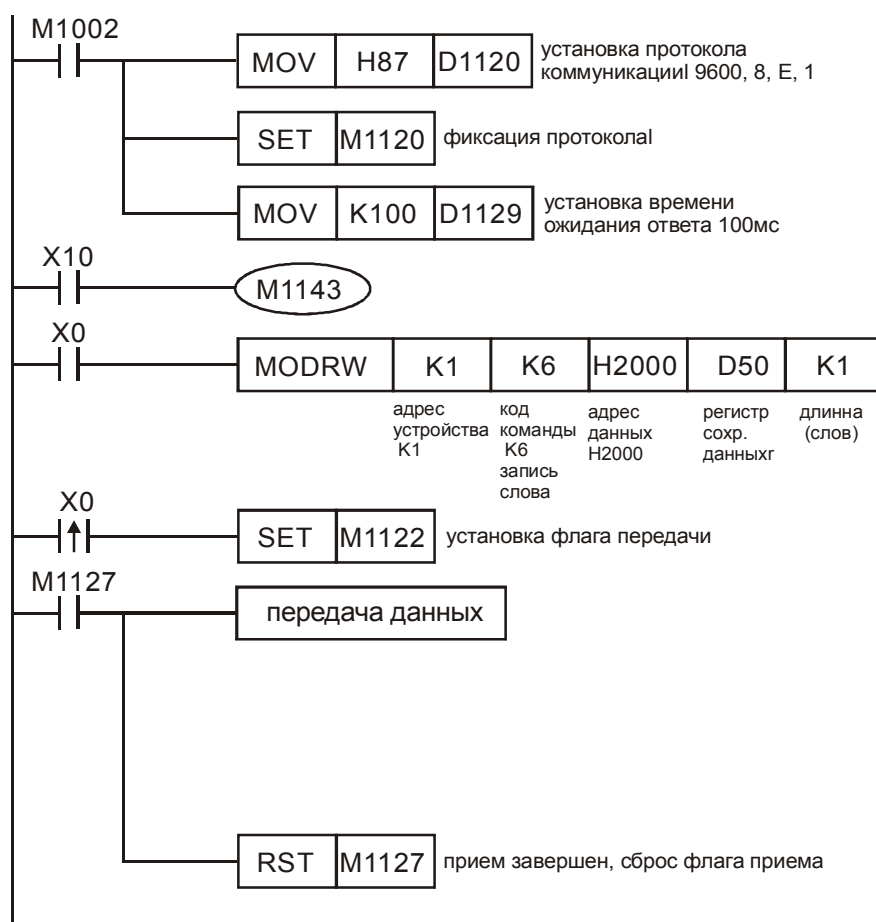
В RTU-режиме данные будут передаваться из регистра D50. Ответные данные будут записаны в D1070 – D1077.

В ASCII-режиме и RTU-режиме переданные данные будут сохраняться регистрах D1256 – D1295. Пользователь может использовать команды MOV, DMOV или BMOV для копирования данных из D1256 – D1295 в регистры общего назначения для последующей обработки. Другие команды для этих регистров не применимы.

Флаг M1129 будет включен, если время ожидания ответа от VFD-S превысит 100 мс.

Флаг M1140 будет включен, если принятые данные будут содержать ошибки.

Флаг M1141 будет включен, если адрес данных будет задан некорректно.



- ASCII-режим

Передаваемое сообщение (ПЛК→VFD-S): "01 06 2000 0012 C7"

Принимаемое сообщ-е (VFD-S→ ПЛК): "01 06 2000 0012 C7"

Передаваемые данные, расписанные по регистрам:

Регистр	Данные			
D1256 младший байт	0	30 Н	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1256 старший байт	1	31 Н	ADR0	
D1257младший байт	0	30 Н	CMD1	CMD (1, 0) – код команды
D1257 старший байт	6	36 Н	CMD0	
D1258 младший байт	2	32 Н	Адрес данных	
D1258 старший байт	0	30 Н		
D1259 младший байт	0	30 Н		
D1259 старший байт	0	30 Н		
D1260 младший байт	0	30 Н	Данные	
D1260 старший байт	0	30 Н		
D1261 младший байт	1	31 Н		
D1261 старший байт	2	32 Н		
D1262 младший байт	C	43 Н	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1262 старший байт	7	37 Н	LRC CHK0	

Принимаемые данные, расписанные по регистрам:

Регистр	Данные			
D1070 младший байт	0	30 H	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1070 старший байт	1	31 H	ADR0	
D1071 младший байт	0	30 H	CMD1	CMD (1, 0) – код команды
D1071 старший байт	6	36 H	CMD0	

D1072 младший байт	2	32 Н	Адрес данных	
D1072 старший байт	0	30 Н		
D1073 младший байт	0	30 Н		
D1073 старший байт	0	30 Н		
D1074 младший байт	0	30 Н	Данные	
D1074 старший байт	0	30 Н		
D1075 младший байт	1	31 Н		
D1075 старший байт	2	32 Н		
D1076 младший байт	C	43 Н	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1076 старший байт	7	37 Н	LRC CHK0	

- RTU-режим

Передаваемое сообщение (ПЛК→VFD-S): **"01 06 2000 0012 02 07"**

Принимаемое сообщение (VFD-S→ ПЛК): **"01 06 2000 0012 02 07"**

Передаваемые данные, расписанные по регистрам:

Регистр	Данные	
D1256 младший байт	01 Н	Коммуникационный адрес VFD-S
D1257 младший байт	06 Н	Адрес команды
D1258 младший байт	20 Н	Адрес данных
D1259 младший байт	00 Н	
D1260 младший байт	00 Н	Данные
D1261 младший байт	12 Н	
D1262 младший байт	02 Н	CRC CHK Low
D1263 младший байт	07 Н	CRC CHK High

Принимаемые данные, расписанные по регистрам:

Регистр	Данные	
D1070 младший байт	01 Н	Коммуникационный адрес VFD-S
D1071 младший байт	06 Н	Адрес команды
D1072 младший байт	20 Н	Адрес данных
D1073 младший байт	00 Н	
D1074 младший байт	00 Н	Данные
D1075 младший байт	12 Н	
D1076 младший байт	02 Н	CRC CHK Low
D1077 младший байт	07 Н	CRC CHK High

Пример применения 3: запись n слов (код команды = 10H) в преобразователь частоты VFD-S (ASCII-режим, когда M1143 = 0; RTU-режим, когда M1143 = 1)

В ASCII-режиме данные будут передаваться из регистров, начиная с D50. Ответные данные будут записаны в D1070 – D1078.

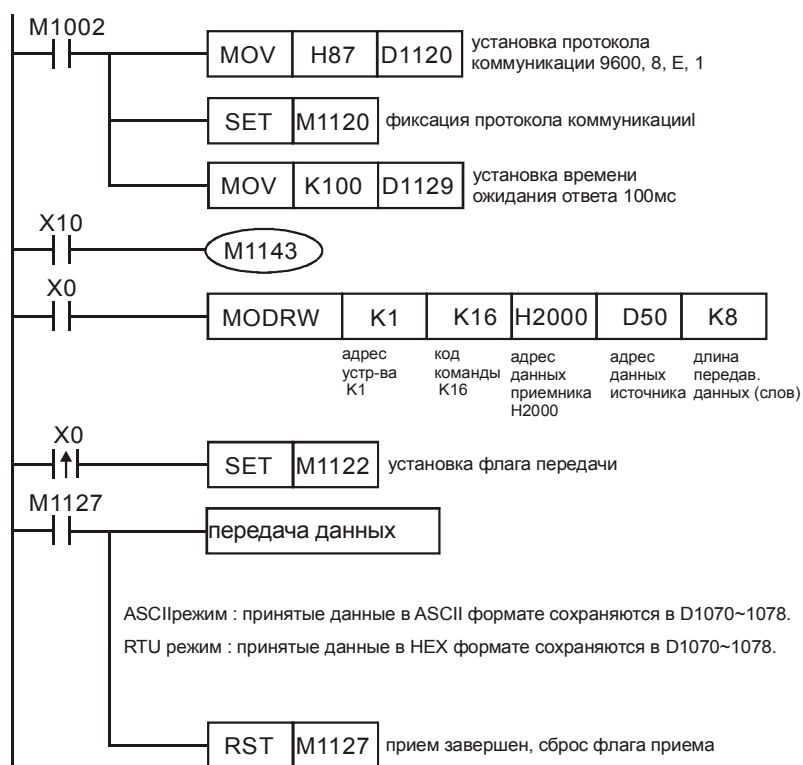
В RTU-режиме данные будут передаваться из регистров, начиная с D50. Ответные данные будут записаны в D1070 – D1078.

В ASCII-режиме и RTU-режиме переданные данные будут сохраняться регистрах D1256 – D1295. Пользователь может использовать команды MOV, DMOV или BMOV для копирования данных из D1256 – D1295 в регистры общего назначения для последующей обработки. Другие команды для этих регистров не применимы.

Флаг M1129 будет включен, если время ожидания ответа от VFD-S превысит 100 мс.

Флаг M1140 будет включен, если принятые данные будут содержать ошибки.

Флаг M1141 будет включен, если адрес данных будет задан некорректно.



- ASCII-режим

Передаваемое сообщение (ПЛК→VFD-S): "01 10 2000 0002 04 0012 1770 30"

Принимаемое сообщ-е (VFD-S→ПЛК): "01 10 2000 0002 CD "

Передаваемые данные, расписанные по регистрам:

Регистр	Данные			
D1256 младший байт	0	30 Н	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1256 старший байт	1	31 Н	ADR0	
D1257 младший байт	1	31 Н	CMD1	CMD (1, 0) – код команды
D1257 старший байт	0	30 Н	CMD0	
D1258 младший байт	2	32 Н	Адрес данных приемника	
D1258 старший байт	0	30 Н		
D1259 младший байт	0	30 Н		
D1259 старший байт	0	30 Н		
D1260 младший байт	0	30 Н	Число регистров	
D1260 старший байт	0	30 Н		
D1261 младший байт	0	30 Н		
D1261 старший байт	2	32 Н		
D1262 младший байт	0	30 Н	Счет байтов	
D1262 старший байт	4	34 Н		
D1263 младший байт	0	30 Н	Данные 1 (D50 = H12)	
D1263 старший байт	0	30 Н		
D1264 младший байт	1	31 Н		
D1264 старший байт	2	32 Н		
D1265 младший байт	1	31 Н	Данные 2 (D50 = H1770)	
D1265 старший байт	7	37 Н		
D1266 младший байт	7	37 Н		
D1266 старший байт	0	30 Н		
D1267 младший байт	3	33 Н	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1267 старший байт	0	30 Н	LRC CHK0	

Принимаемые данные, расписанные по регистрам:

Регистр	Данные			
D1070 младший байт	0	30 H	ADR1	ADR (1, 0) – коммуникационный адрес VFD-S
D1070 старший байт	1	31 H	ADR0	
D1071 младший байт	1	31 H	CMD1	CMD (1, 0) – код команды
D1071 старший байт	0	30 H	CMD0	
D1072 младший байт	2	32 H	Адрес данных	
D1072 старший байт	0	30 H		
D1073 младший байт	0	30 H		
D1073 старший байт	0	30 H		
D1074 младший байт	0	30 H	Число регистров	
D1074 старший байт	0	30 H		
D1075 младший байт	0	30 H		
D1075 старший байт	2	32 H		
D1076 младший байт	C	43 H	LRC CHK1	LRC CHK (1, 0) – контрольная сумма
D1076 старший байт	D	44 H	LRC CHK0	

- RTU-режим

Передаваемое сообщение (ПЛК→VFD-S): "01 10 2000 0002 04 0012 1770 C4 7F "

Принимаемое сообщение (VFD-S→ ПЛК): "01 10 2000 0002 4A 08"

Передаваемые данные, расписанные по регистрам:

Регистр	Данные		
D1256 младший байт	01 H		Коммуникационный адрес VFD-S
D1257 младший байт	10 H		Адрес команды
D1258 младший байт	20 H	Адрес данных	
D1259 младший байт	00 H		
D1260 младший байт	00 H	Число регистров	
D1261 младший байт	02 H		
D1262 младший байт	04 H	Счет байтов	
D1263 младший байт	00 H	Данные 1 (D50 = H12)	
D1264 младший байт	12 H		
D1265 младший байт	17 H	Данные 2 (D50 = H1770)	
D1266 младший байт	70 H		
D1267 младший байт	C4 H	CRC CHK Low	
D1268 младший байт	7F H	CRC CHK High	

Принимаемые данные, расписанные по регистрам:

Регистр	Данные		
D1070 младший байт	01 H		Коммуникационный адрес VFD-S
D1071 младший байт	10 H		Адрес команды
D1072 младший байт	20 H	Адрес данных	
D1073 младший байт	00 H		
D1074 младший байт	00 H	Число регистров	
D1075 младший байт	02 H		
D1076 младший байт	4A H	CRC CHK Low	
D1077 младший байт	08 H	CRC CHK High	

API		PWD		S D	Импульсная ловушка	DVP-		
151						ES/EX/SS	SA/SX	EH
						-	-	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (5 шагов)			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
S	*																	
D													*					
<u>Примечания:</u> Диапазон операнда (S): X10 ... X17 Диапазон операнда (D): D0...D9999 Операнд (D) занимает два последовательных регистра. Возможно только однократное использование в программе.															32-х битная инструкция --- Флаги: нет			

Функция

Регистрация длительности входного импульса

Описание

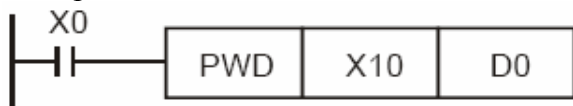
(S) – вход ПЛК

(D) – регистр для записи длительности импульса на входе ПЛК. Ед. измерения 100 мкс

- Максимальная длительность регистрируемого импульса: 214748.3647 сек, или 3579.139 мин, или 59.652 часа
- Минимальная длительность регистрируемого импульса: 100 мкс. Если длительность импульса меньше 100мкс, то (D) = 0.

Пример программирования PWD-инструкции

Когда вход X0 включен, будет регистрироваться время включенного состояния входа X10 и сохраняться в D0, D1



API		RTMU		S D	Начало измерения времени выполнения подпрограммы внешнего прерывания	DVP-		
152						ES/EX/SS	SA/SX	EH
						-	-	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (9 шагов) RTMU - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S					*	*							*			
D					*	*							*			
<u>Примечания:</u> Диапазон операнда (S): K0 ... K9 Диапазон операнда (D): K10...K500														32-х битная инструкция ---		
														Флаги: нет		

Функция


Определение начала измерения времени выполнения подпрограммы от внешнего прерывания I.

Описание

(S) – определение адреса сохранения результата измерения

(D) – базовое время измерения. Ед. измерения 1 мкс

- Операнд (S) определяет адрес специального регистра (D1156 – D1165) для сохранения измеренного времени. Например, если (S) = K5, определяется адрес специального регистра D1161.
- После выполнения RTMU-инструкции, если (S) и (D) введены пользователем правильно, будет запущен таймер, измеряющий время выполнения подпрограммы от внешнего прерывания I, и одновременно будет обнулен специальный регистр, определенный в (S). После выполнения RTMD-инструкции таймер будет остановлен, и результат измерения будет записан в специальный регистр, определенный в (S).
- Инструкции RTMU и RTMD должны использоваться в паре. Если не будет RTMD-инструкции то измерение времени не будет прекращено.
- Инструкции RTMU и RTMD могут применяться для служебных целей при разработке новой программы для измерения и оценки времени обработки внешних прерываний. После завершения разработки программы их рекомендуется убрать.

API					DVP-		
153	RTMD			Конец измерения времени выполнения подпрограммы внешнего прерывания	ES/EX/SS	SA/SX	EH
					-	-	+

Оп- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (9 шагов) RTMD - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S					*	*							*			<u>32-х битная инструкция</u> --- Флаги: нет
<u>Примечания:</u> Диапазон операнда (S): K0 ... K9																

Функция

Определение конца измерения времени выполнения подпрограммы от внешнего прерывания I.

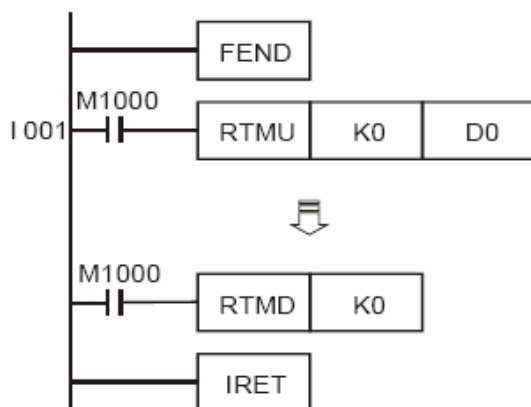
Описание




(S) – определение адреса сохранения результата измерения

- Операнд (S) определяет адрес специального регистра (D1156 – D1165) для индикации времени. Например, если (S) = K5, определяется адрес специального регистра D1161.

Пример применения

По переднему фронту включения входа X0, начнется выполнение подпрограммы прерывания с указателем I001. После выполнения RTMU-инструкции запустится 8-ми битный таймер (с дискретностью 10 мкс). После выполнения RTMD-инструкции измерение времени прекратится и результат будет записан в D1156.



API	RAND	  	Генератор случайных чисел	DVP-		
154				ES/EX/SS	SA/SX	EH
				-	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (7 шагов) RAND - Непрерывное выполнение. RANDP - Импульсное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	
<u>Примечания:</u> Диапазон операндов (S1), (S2): K0 ... K32767 (S1) должно быть меньше или равно (S2)																<u>32-х битная инструкция</u> --- Флаги: нет

Функция

Генерация случайного числа в заданном диапазоне.

Описание

(S1) – минимальное значение случайного числа

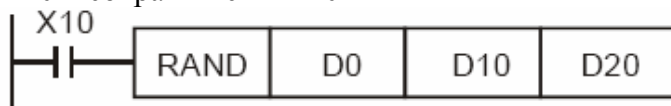
(S1) – максимальное значение случайного числа




(D) – операнд для хранения сгенерированного случайного числа

- Если операнд ввести $(S1) > (S2)$, то будет зафиксирована в D1067 ошибка с кодом 0E1A и включаться флаги M1067, M1068.

Пример применения

Когда вход X10 включен, будет генерироваться случайное число в диапазоне от D0 до D10 и сохраняться в D20



API	-	ABSR	  	Чтение абсолютной текущей позиции	DVP-		
					ES/EX/SS	SA/SX	EH
155	D				-	-	+

Оп- ранд	Биты				Слова											16-ти битная инструкция
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	---
S	*	*	*	*												32-х битная инструкция
D ₁		*	*	*												(13 шагов)
D ₂								*	*	*	*	*	*	*	*	DABSR - Непрерывное
<u>Примечания:</u> Операнд (S) занимает 3 последовательных адреса Операнд (D1) занимает 3 последовательных адреса Операнд (D2) занимает 2 последовательных адреса																выполнение.
																Флаги: M1010, M1029,
																M1030, M1334, M1335,
																M1336, M1337, M1346

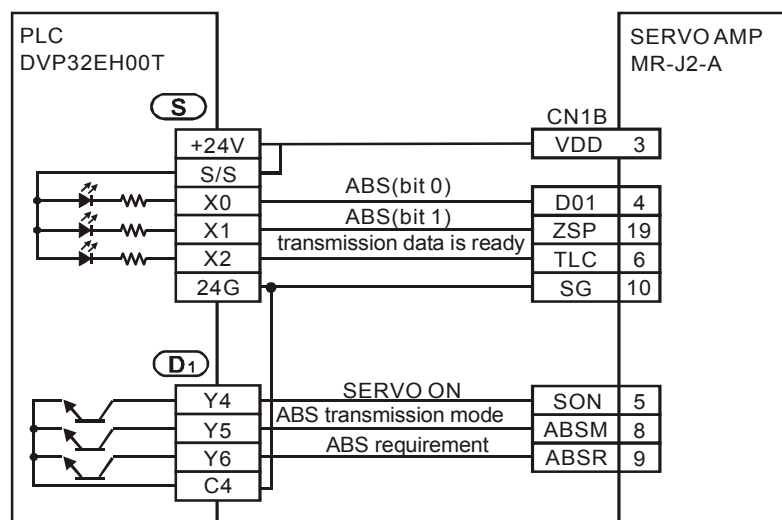
Функция

С помощью этой инструкции может восприниматься абсолютная позиция сервоусилителей MITSUBISHI типа MR-H и MR-J2. Сервопривод должен быть настроен на режим работы в абсолютных координатах.

Описание

- ABSR-инструкция является 32-х битной инструкцией. Поэтому всегда указывайте ее как "DABSR".
- Чтение позиции начинается, если выполнено стартовое условие ABS-инструкции. При окончании чтения включается реле M1029, M1030 (они должны быть сброшены пользователем). Если во время чтения стартовые условия отключаются, процесс чтения прекращается.
- При проектировании нужно следить за тем, чтобы сервоусилитель включался раньше или одновременно с ПЛК.
- 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 D1337, D1336 (канал CH0) и D1339, D1338 (канал CH1).

Схема соединения должна быть следующей:



- Стартовые условия ABSR-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 программирования DABSR-инструкции

Когда вход X7 включен, значение абсолютной позиции (32 бита) будет читаться от сервопривода и сохраняться в регистрах (D1337, D1336). Одновременно, запустится таймер T10 с уставкой 5 сек. Если за 5 сек абсолютное положение не будет считано, включится контакт M10, сигнализирующий о сбое чтения позиции привода.

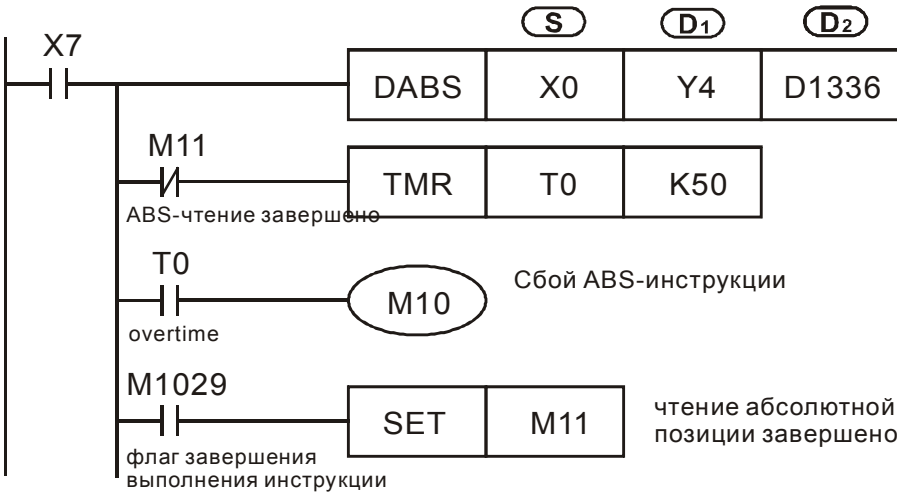
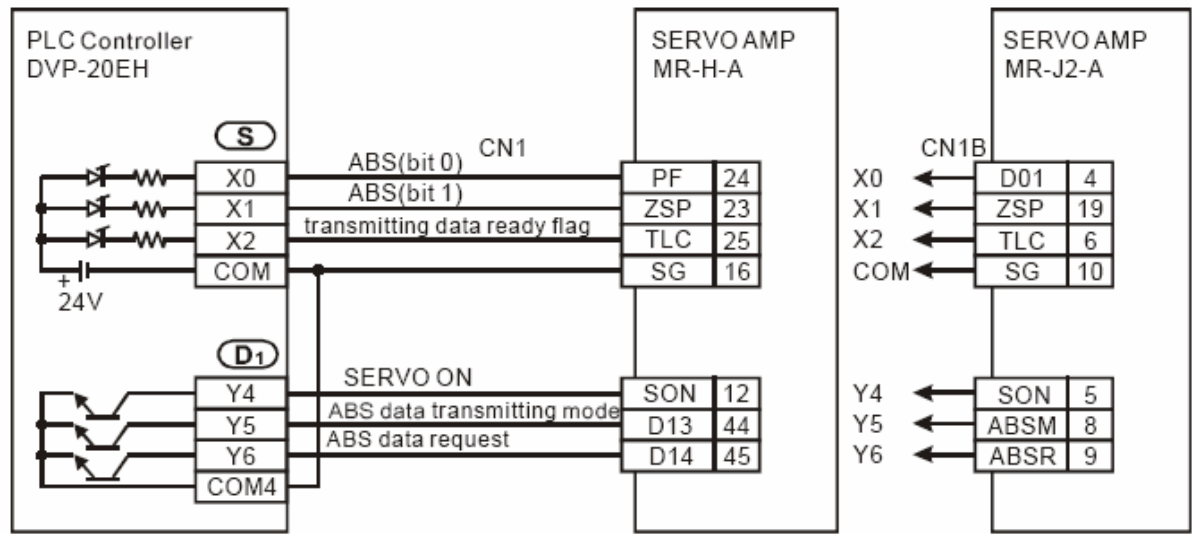
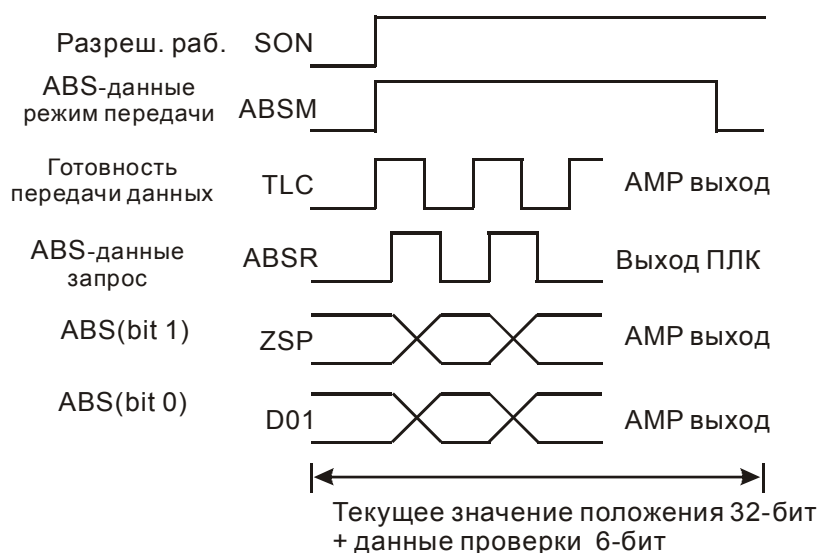


Схема соединения:



Временная диаграмма:



Описание флагов и специальных регистров

M1010	ЕН: M1010=ON импульсы на CH0 (Y0, Y1) и CH1 (Y2, Y3) будут идти до выполнения инструкции END. С первым импульсом на выходе флаг будет автоматически сброшен.
M1029	ЕН: Первая импульсная группа CH0 (Y0, Y1) полностью выполнена.
M1030	ЕН: Вторая импульсная группа CH1 (Y2, Y3) полностью выполнена.
M1334	Команда - стоп импульсного выхода CH0 (Y0, Y1)
M1335	Команда - стоп импульсного выхода CH1 (Y2, Y3)
M1336	Флаг передачи импульсов CH0 (Y0, Y1)
M1337	Флаг передачи импульсов CH1 (Y2, Y3)
M1346	Флаг разрешения очистки выходного сигнала от команды ZRN

D1336	CH0 (Y0, Y1) текущее значение импульсов (младшее слово)
D1337	CH0 (Y0, Y1) текущее значение импульсов (старшее слово)
D1338	CH1 (Y2, Y3) текущее значение импульсов (младшее слово)
D1339	CH1 (Y2, Y3) текущее значение импульсов (старшее слово)
D1341	Максимальная выходная частота (младшее слово)
D1342	Максимальная выходная частота (старшее слово)
D1343	Время разгона/замедления. Диапазон: 50...5000 мс. Заводская установка 100мс

API		ZRN	S₁ S₂ S₃ D	Выход в ноль	DVP-		
156	D				ES/EX/SS	SA/SX	EH
					-	-	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (9 шагов) ZRN - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
S ₃	*	*	*	*												
D		*														
<u>Примечания:</u> описание флагов см. в описании ABSR (API155)																
Флаги: M1010, M1029, M1030, M1334, M1335, M1336, M1337, M1346																

Функция

С помощью этой инструкции можно управлять перемещением к нулевой (исходной) точке сервопривода. Во время позиционирования в ПЛК запоминается текущая позиция и в зависимости от направления движения повышается или снижается по мере перемещения. Благодаря этому позиция машины в ПЛК всегда актуальна. При отключении напряжения ПЛК теряет это значение позиции. Поэтому после включения управления или для отладки машины выполняется возврат в исходное положение к определенной позиции.

Описание

- В (S1) указывается скорость, с которой должен быть выполнен возврат к нулевой точке. При словном операнде возможно от 10 до 32767 Гц, при 32-х битовом операнде возможна область от 10 до 200 кГц.
- В (S2) указывается ползучая (медленная) скорость от 10 до 32,767 Гц.
- В (S3) определяется переключатель (DOG) для перехода на ползучую скорость. Если в (S3) указан другой операнд, чем вход (X), то на временную точку торможения окажет влияние время цикла ПЛК (программы) и отклонение от нулевой точки может оказаться очень большим.
- В (D) определяется выход ПЛК для выдачи импульсов. Могут применяться только выходы Y0 или Y2. Так как импульсы выдаются с большой частотой, то ПЛК должно применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны.
- Если включено специальное реле M1346, то на сервоусилитель будет передан сигнал сброса, когда достигается нулевая точка.

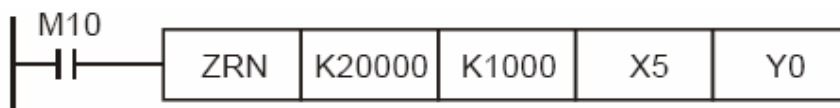
Для канала CH0 (Y0, Y1) – сигнал сброса Y4;

Для канала CH1 (Y2, Y3) – сигнал сброса Y5;

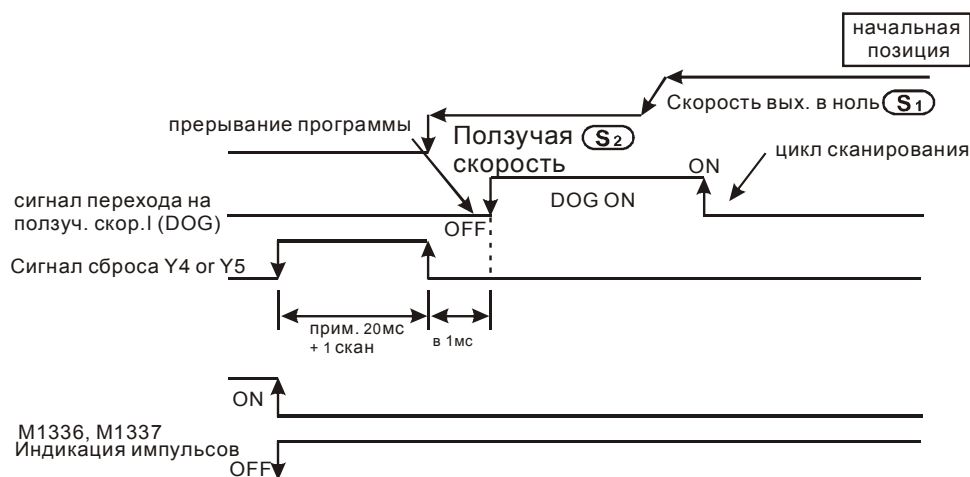
ВНИМАНИЕ! К нулевой точке нельзя подойти с любого направления. При перемещении в исходное положение всегда перемещаются назад в направлении нулевой точки.

Пример программирования ZRN-инструкции

Когда M10 включено, на выходе начнут выдаваться импульсы с частотой 20 кГц. По достижении точки (DOG) перехода на ползучую скорость, когда замкнется X5, выходные импульсы будут выдаваться с частотой 1 кГц. Когда X5 разомкнется, выход в ноль будет выполнен и выдача импульсов прекратится.

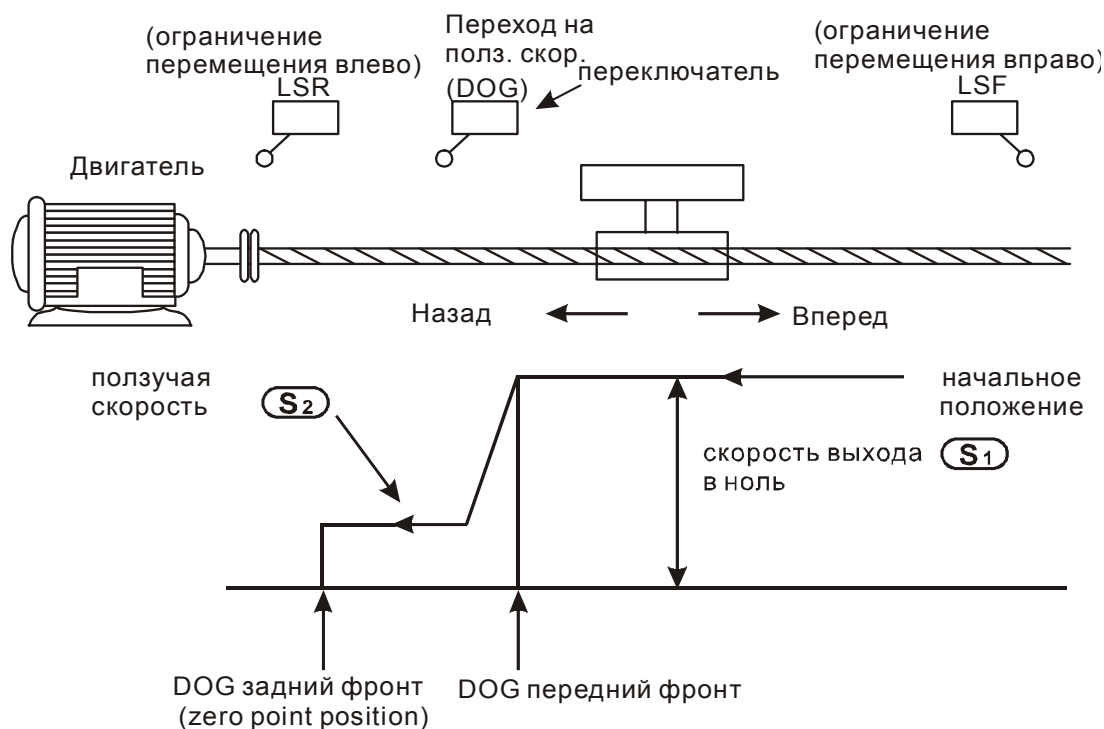


Временная диаграмма выхода в ноль:



При окончании выдачи импульсов включается реле M1029, M1030. Флаги индикации выдачи импульсов M1336 для CH0 и M1337 для CH1 будут выключены.

При перемещении в исходное положение всегда перемещаются назад в направлении нулевой точки и элементы должны быть расположены как показано на рисунке:



Когда ZRN-инструкция выполняется, текущее значение выданных импульсов можно прочитать в D1337, D1336 (канал CH0) и D1339, D1338 (канал CH1)

Когда ZRN-инструкция выполняется, текущее значение выходной частоты можно прочитать в (D1394, D1395) и (D1396, D1397).

API		PLSV		S D1 D2	Выдача импульсов с переменной частотой	DVP-		
157	D					ES/EX/SS	SA/SX	EH
						-	-	+

Оп- еранд	Биты				Слова											16-ти битная инструкция (7 шагов) PLSV - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	
D ₁		*														
D ₂		*	*	*												
<u>Примечания:</u> описание флагов см. в описании ABSR (API155)																32-х битная инструкция (13 шагов) DPLSV - Непрерывное выполнение.
																Флаги: M1010, M1029, M1030, M1334, M1335, M1336, M1337, M1346

Функция

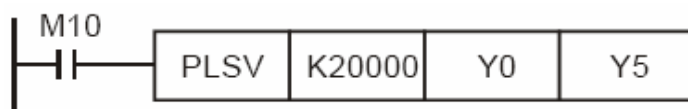
Выдача импульсов с переменной частотой на Y0 или Y2 и сигнала для определения направления вращения серводвигателя.

Описание

- В (S) указывается частота выходных импульсов. При словном операнде охватывается область от 10 до 32767 Гц, при 32-х битовом операнде выдается частота от 10 до 200 кГц.
- В (D1) определяется выход ПЛК для выдачи импульсов. Могут применяться только выходы Y0 и Y2. Так как импульсы выдаются с большой частотой, то ПЛК должен применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны.
- 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 реле M1336 (контроль Y0) и M1337 (контроль Y2).

Пример программирования PLSV-инструкции

Когда M10 включено, на выходе Y0 будут выдаваться импульсы с частотой 20 кГц. Y5=1 – представляет положительное направление вращения.



API					Позиционирование в относительных координатах	DVP-		
		DRVI		S₁ S₂ D₁ D₂		ES/EX/SS	SA/SX	EH
158	D					-	-	+

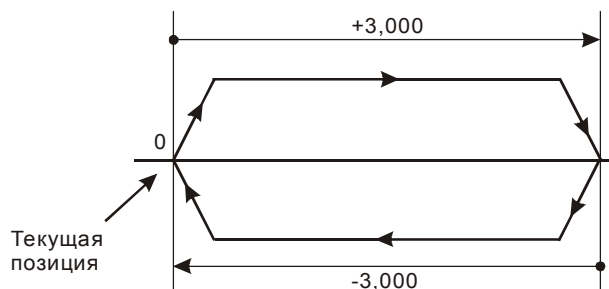
Оп- еранд	Биты				Слова											16-ти битная инструкция (9 шагов) DRVI - Непрерывное выполнение. 32-х битная инструкция (17 шагов) DDRVI - Непрерывное выполнение. Флаги: M1010, M1029, M1030, M1334, M1335, M1336, M1337, M1346
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
D ₁		*														
D ₂		*	*	*												
Примечания: описание флагов см. в описании ABSR (API155)																

Функция

Команда перемещения в заданное положение в относительных координатах.

Описание

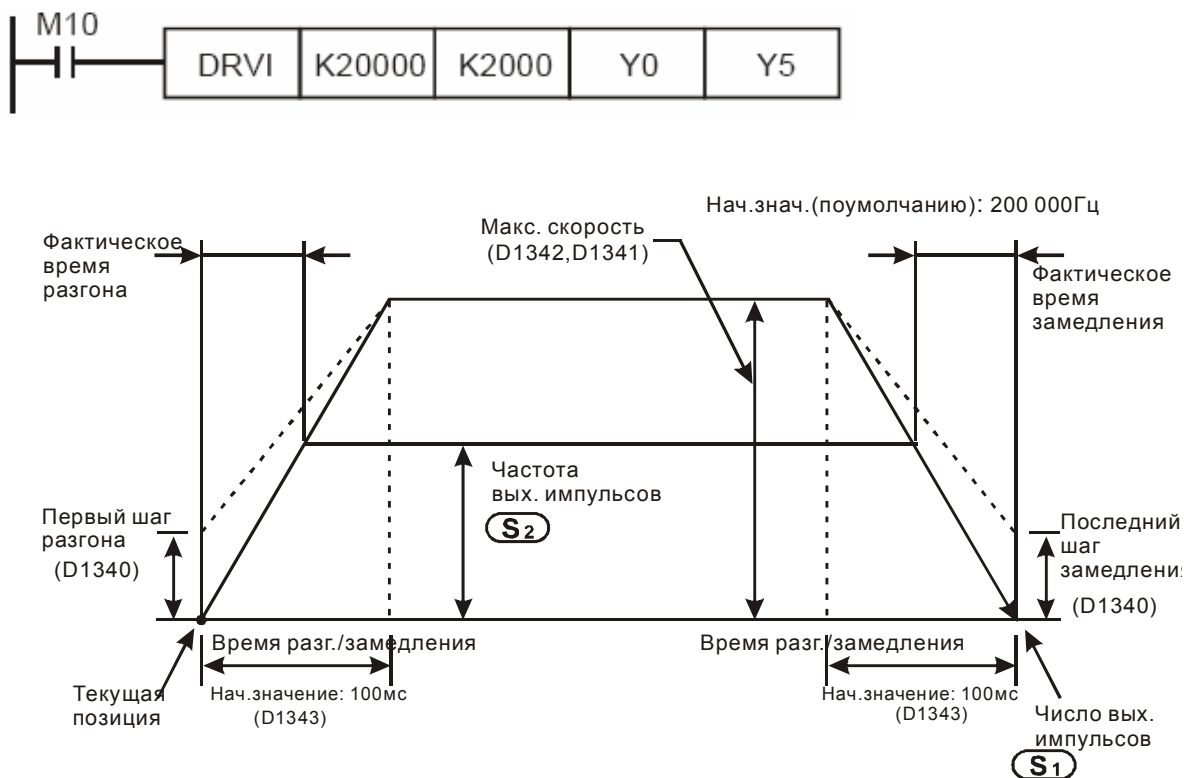
- В (S₁) указывается число импульсов (приращений), на которое надо переместиться (относительное перемещение). Область значений при словном операнде охватывает - 32767...32767 приращений, а при 32-х битном операнде -999999...999999 приращений.
- В (S₂) указывается частота импульсов выхода. При словном операнде охватывается область от 10 до 32767 Гц, при 32-х битном операнде возможна выдача частоты от 10 до 200 кГц.
- В (D₁) определяется выход ПЛК для выдачи импульсов. Могут применяться только выходы Y0 и Y2. Так как импульсы выдаются с большой частотой, то ПЛК должен применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны.
- 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.



- Когда DRVI-инструкция выполняется, текущее значение выданных импульсов можно прочитать в D1337, D1336 (канал CH0) и D1339, D1338 (канал CH1)
- Если во время выполнения инструкции изменяется содержание операнда, то это не влияет на позиционирование. Измененный операнд будет принят во внимание при следующем выполнении инструкции.
- Если стартовые условия DRVI-инструкции во время выполнения инструкции больше не выполняются, то замедления обрабатывается и привод останавливается. В этом случае специальный флаг M1029, M1030 который показывает конец обработки, не включается.
- После отключения стартовых условий DRVI-инструкция запускается вновь тогда, когда выключены специальные реле M1336 (контроль Y0) и M1337 (контроль Y2)..

Пример программирования DRVI-инструкции

Когда M10 включено, на выходе Y0 будет выдано 2000 импульсов с частотой 2 кГц. Y5=1 – представляет положительное направление вращения.

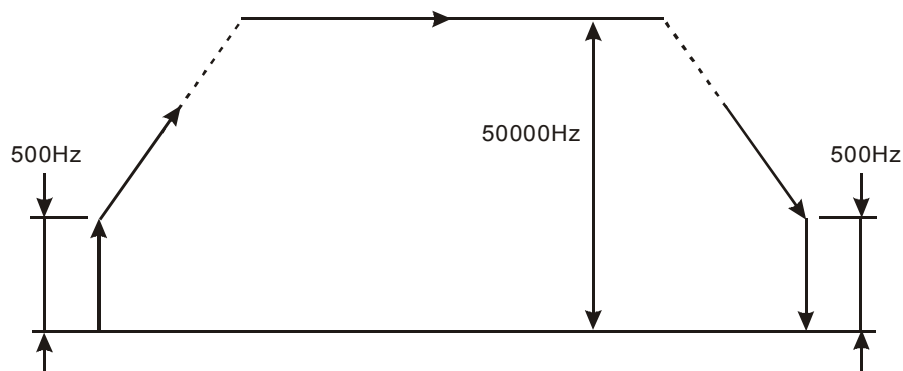


- Минимальная выходная частота может быть рассчитана по следующей формуле:

$$F_{\min} = \sqrt{\text{Макс. скор. [D1342, D1341]Гц} / (2 \times (\text{Разгон/Торможение [D1343]мс} / 1000))}$$

Пример расчета:

$$F_{\min} = \sqrt{50000\text{Гц} / (2 \times (100\text{мс} / 1000))} = 500\text{Гц}$$



Описание флагов и специальных регистров

M1010	ЕН: M1010=ON импульсы на CH0 (Y0, Y1) и CH1 (Y2, Y3) будут идти до выполнения инструкции END. С первым импульсом на выходе флаг будет автоматически сброшен.
M1029	ЕН: Первая импульсная группа CH0 (Y0, Y1) полностью выполнена.
M1030	ЕН: Вторая импульсная группа CH1 (Y2, Y3) полностью выполнена.
M1334	Команда - стоп импульсного выхода CH0 (Y0, Y1)
M1335	Команда - стоп импульсного выхода CH1 (Y2, Y3)
M1336	Флаг передачи импульсов CH0 (Y0, Y1)
M1337	Флаг передачи импульсов CH1 (Y2, Y3)
M1346	Флаг разрешения очистки выходного сигнала от команды ZRN

D1336	CH0 (Y0, Y1) текущее значение импульсов (младшее слово)
D1337	CH0 (Y0, Y1) текущее значение импульсов (старшее слово)
D1338	CH1 (Y2, Y3) текущее значение импульсов (младшее слово)
D1339	CH1 (Y2, Y3) текущее значение импульсов (старшее слово)
D1340	Минимальная выходная частота. Диапазон: $\leq 1/10$ от макс. выходной частоты
D1341	Максимальная выходная частота (младшее слово)
D1342	Максимальная выходная частота (старшее слово)
D1343	Время разгона/замедления. Диапазон: 50...5000 мс. Заводская установка 100мс

API		DRVA		S₁ S₂ D₁ D₂	Позиционирование в абсолютных координатах	DVP-		
159	D					ES/EX/SS	SA/SX	EH
						-	-	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (9 шагов) DRVA - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
D ₁		*														
D ₂		*	*	*												
<u>Примечания:</u> описание флагов см. в описании DRVA (API158)																32-х битная инструкция (17 шагов) DDRVA - Непрерывное выполнение. Флаги: M1010, M1029, M1030, M1334, M1335, M1336, M1337, M1346

Функция

Команда перемещения в заданное положение в абсолютных координатах. При таком виде позиционирования должна быть определена нулевая (исходная) точка. Текущая действительная позиция измеряется как абсолютное значение по отношению к этой нулевой точке.

Описание

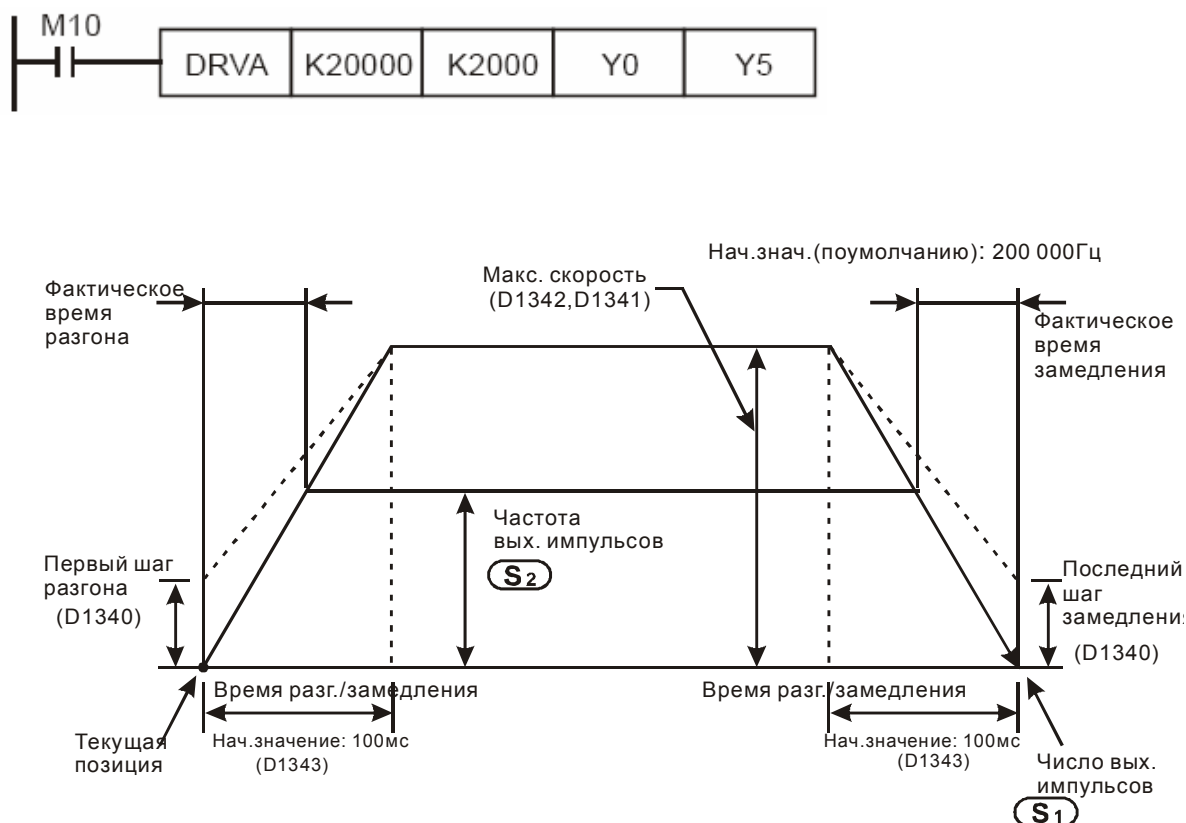
- В (S1) указывается абсолютная позиция (заданное положение), т.е. число импульсов относительно нулевой точки, куда надо переместиться. Область значений при словном операнде охватывает -32767...32767, а при 32-х битном операнде -2147483648 ~ +2147483647.
- В (S2) указывается частота импульсов выхода. При словном операнде охватывается область от 10 до 32767 Гц, при 32-х битном операнде возможна выдача частоты от 10 до 200 кГц.
- В (D1) определяется выход ПЛК для выдачи импульсов. Могут применяться только выходы Y0 и Y2. Так как импульсы выдаются с большой частотой, то ПЛК должен применяться с транзисторными выходами. Релейные контакты в этом случае изнашиваются преждевременно и поэтому непригодны.
- 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.



- Когда DRVA-инструкция выполняется, текущее значение выданных импульсов можно прочитать в D1337, D1336 (канал CH0) и D1339, D1338 (канал CH1)
- 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 DRVA-инструкции во время выполнения инструкции больше не выполняются, то замедления отрабатывается и привод останавливается. В этом случае специальный флаг M1029, M1030 который показывает конец обработки, не включается.
- После отключения стартовых условий DRVA-инструкция запускается вновь тогда, когда выключены специальные реле M1336 (контроль Y0) и M1337 (контроль Y2).

Пример программирования DRVI-инструкции

Когда M10 включено, на выходе Y0 будет выдано такое число импульсов с частотой 2 кГц, которое приведет сервопривод в позицию на расстоянии 20000 импульсов от нулевой точки. Y5=1 – представляет положительное направление вращения.



• Отношение между фактической частотой и временем разгона/замедления следующее:

Tg: время дискретизации разгона/замедления

T: время разгона/замедления

Fa: наклон разгона/замедления

F: Макс. частота

F0: Мин. частота

P: полное число импульсов

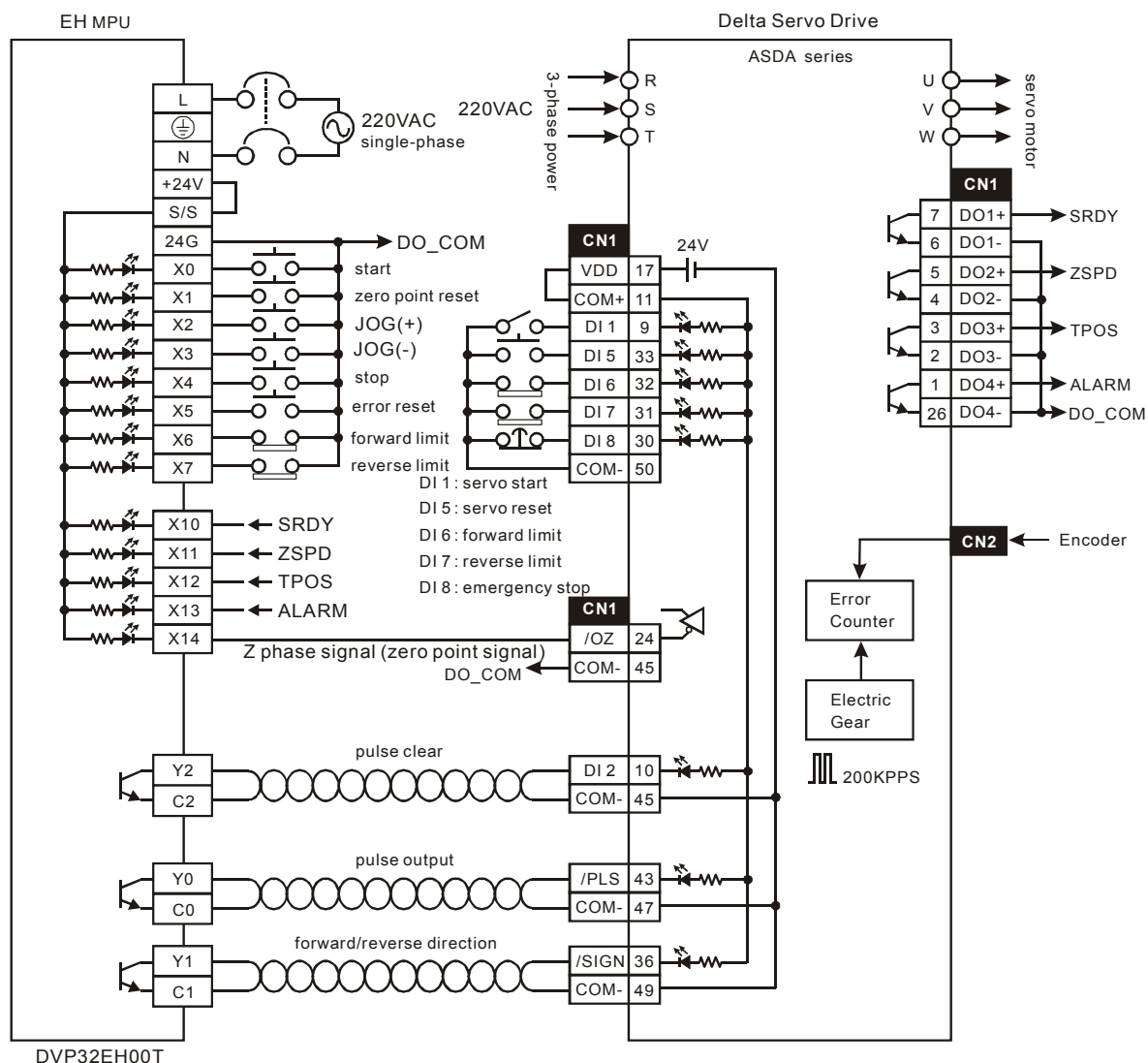
1. $Tg = T / (60 * 1000)$
2. $Fa = (F - F0) / 60$
3. P0 (число вых. импульсов первого шага) = 1
4. Каждый сегмент частоты:
 - i. $F(n) = F0 + Fa * n$ ($n = 1 \sim 60$)
5. Число импульсов в каждом сегменте:

$$P(n) = \lfloor Tg * F(n) \rfloor$$

Ограничения:

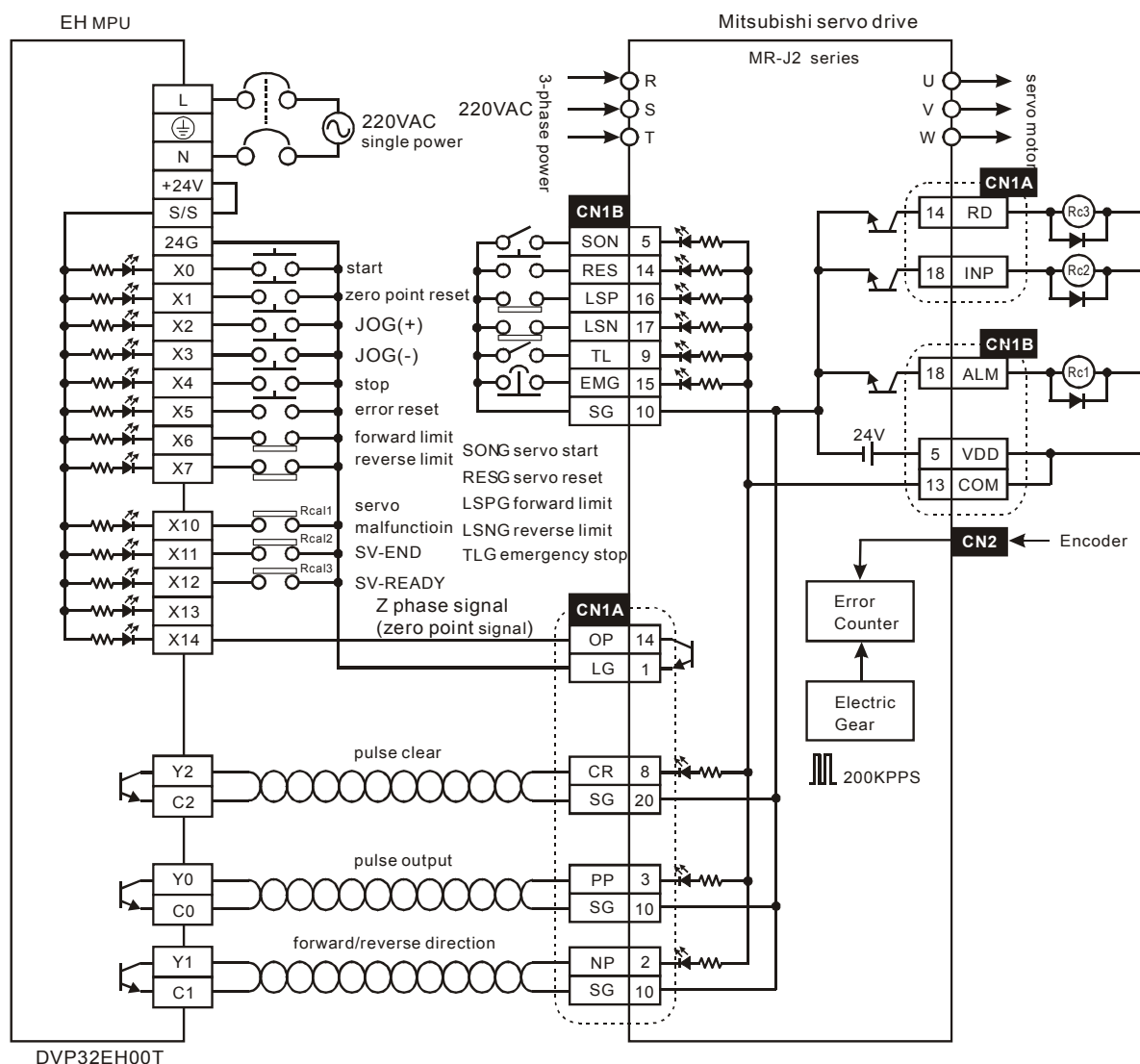
1. Когда $P(n) < 1$, ПЛК не будет выдавать импульсы и перейдет к следующему сегменту.
2. Мин. частота не может быть меньше 10 Гц.
3. При полном числе импульсов $P \leq 3$, разгон/замедление не будет выполняться.

- Пример соединения DVP-EH и сервопривода ASDA (Delta)



Рекомендуется обязательно использовать конечные выключатели для ограничения крайних левого (DI7) и правого (DI6) перемещений.

- Пример соединения DVP-EH и сервопривода MR-J2 (Mitsubishi)



Рекомендуется обязательно использовать конечные выключатели для ограничения крайних левого (LSNG) и правого (LSPG) перемещений.

Предупреждения при использовании инструкций позиционирования.

Инструкции позиционирования ZRN (API 156), PLSV (API 157), DRVI (API 158) и DRVA (API 159) работают только с выходами Y0, Y1 (CH0) и Y2, Y3 (CH1).

Инструкции позиционирования могут использоваться в программе многократно. Но,

- 1) нельзя одновременно активизировать несколько инструкций с одинаковыми выходами (Y0 и Y2). При работе одинаковых выходов ПЛК воспринимает их, как двойной выход, и данная инструкция будет работать не корректно.
- 2) рекомендуется использовать режим пошагового управления (STL)
 - Перед повторной активизацией инструкции убедитесь, что специальные реле M1336 (контроль Y0) и M1337 (контроль Y2) выключены.

Предупреждение при использовании инструкций импульсных выходов FNC 57 (PLSY) и FNC 59 (PLSR)

- Если инструкции позиционирования и инструкции импульсных выходов используют одновременно одинаковый выход, то ПЛК 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).

Выходы ПЛК:

- 1) Для работы инструкций позиционирования необходимо использовать ПЛК, только с транзисторными выходами.
- 2) Диапазон напряжения: от 5 до 24 В постоянного тока.
- 3) Для более устойчивой работы на больших частотах рекомендуется нагрузка на транзисторный выход от 10 до 100 мА.
- 4) Максимальная частота для Y0, Y2 = 200 кГц; для Y1, Y3 = 10 кГц;

Формат выходных импульсов:

1. 1 фаза 1 выхода + направление (рекомендуется использовать этот режим)



2. 1 фаза 2 выхода (макс. 10 кГц)



3. 2 фазы 2 выхода (макс. 10 кГц)



Выбирайте соответствующий формат импульсов и в сервоприводе или шаговом приводе.

Описание флагов и специальных регистров

M1010	ЕН: M1010=ON импульсы на CH0 (Y0, Y1) и CH1 (Y2, Y3) будут идти до выполнения инструкции END. С первым импульсом на выходе флаг будет автоматически сброшен.
M1029	ЕН: Первая импульсная группа CH0 (Y0, Y1) полностью выполнена.
M1030	ЕН: Вторая импульсная группа CH1 (Y2, Y3) полностью выполнена.
M1334	Команда - стоп импульсного выхода CH0 (Y0, Y1)
M1335	Команда - стоп импульсного выхода CH1 (Y2, Y3)
M1336	Флаг передачи импульсов CH0 (Y0, Y1)
M1337	Флаг передачи импульсов CH1 (Y2, Y3)
M1346	Флаг разрешения очистки выходного сигнала от команды ZRN

D1220	Режим работы первой группа импульсных выходов (определяют последние два бита): 00: 1 фазн. (Y0); 01: (Y0, Y1) двухфазный - фаза А опережает В; 02: (Y0, Y1) двухфазный - фаза В опережает А 03: 1 фазн. (Y1)
-------	--

D1221	Режим работы второй группы импульсных выходов (определяют последние два бита): 00: 1 фазн. (Y2); 01: (Y2, Y3) двухфазный - фаза А опережает В; 02: (Y2, Y3) двухфазный - фаза В опережает А 03: 1 фазн. (Y3)
D1336	CH0 (Y0, Y1) текущее значение импульсов (младшее слово)
D1337	CH0 (Y0, Y1) текущее значение импульсов (старшее слово)
D1338	CH1 (Y2, Y3) текущее значение импульсов (младшее слово)
D1339	CH1 (Y2, Y3) текущее значение импульсов (старшее слово)
D1340	Минимальная выходная частота. Диапазон: $\leq 1/10$ от макс. выходной частоты
D1341	Максимальная выходная частота (младшее слово)
D1342	Максимальная выходная частота (старшее слово)
D1343	Время разгона/замедления. Диапазон: 50...5000 мс. Заводская установка 100мс

Для инструкций (PLSV), (DRVI) и (DRVA) текущее значение уменьшается или увеличивается в соответствии с направлением вращения. Для инструкций PLSY) и (PLSR) используются те же регистры, но в них представлено суммарное выданное число импульсов.

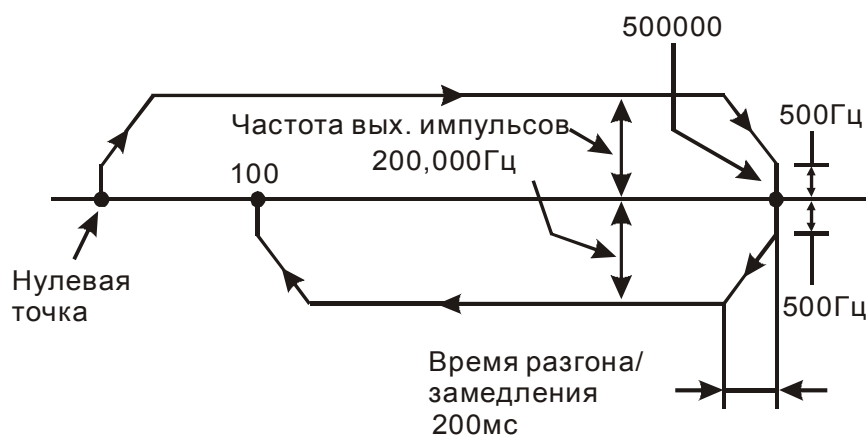
Задается время разгона до максимальной частоты (D1343).

Задается время торможения с максимальной частоты (D1343) до нулевой скорости.

Пример программирования операции ВПЕРЕД/НАЗАД

Используется DVP-EH с сервоприводом Mitsubishi MR-J2-□A.

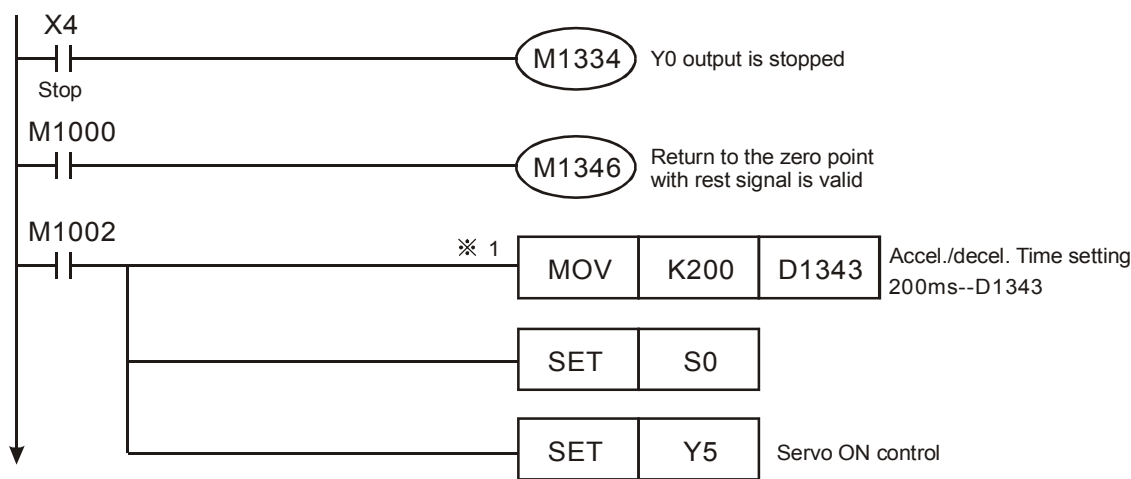
Позиционирование выполняется абсолютным способом.



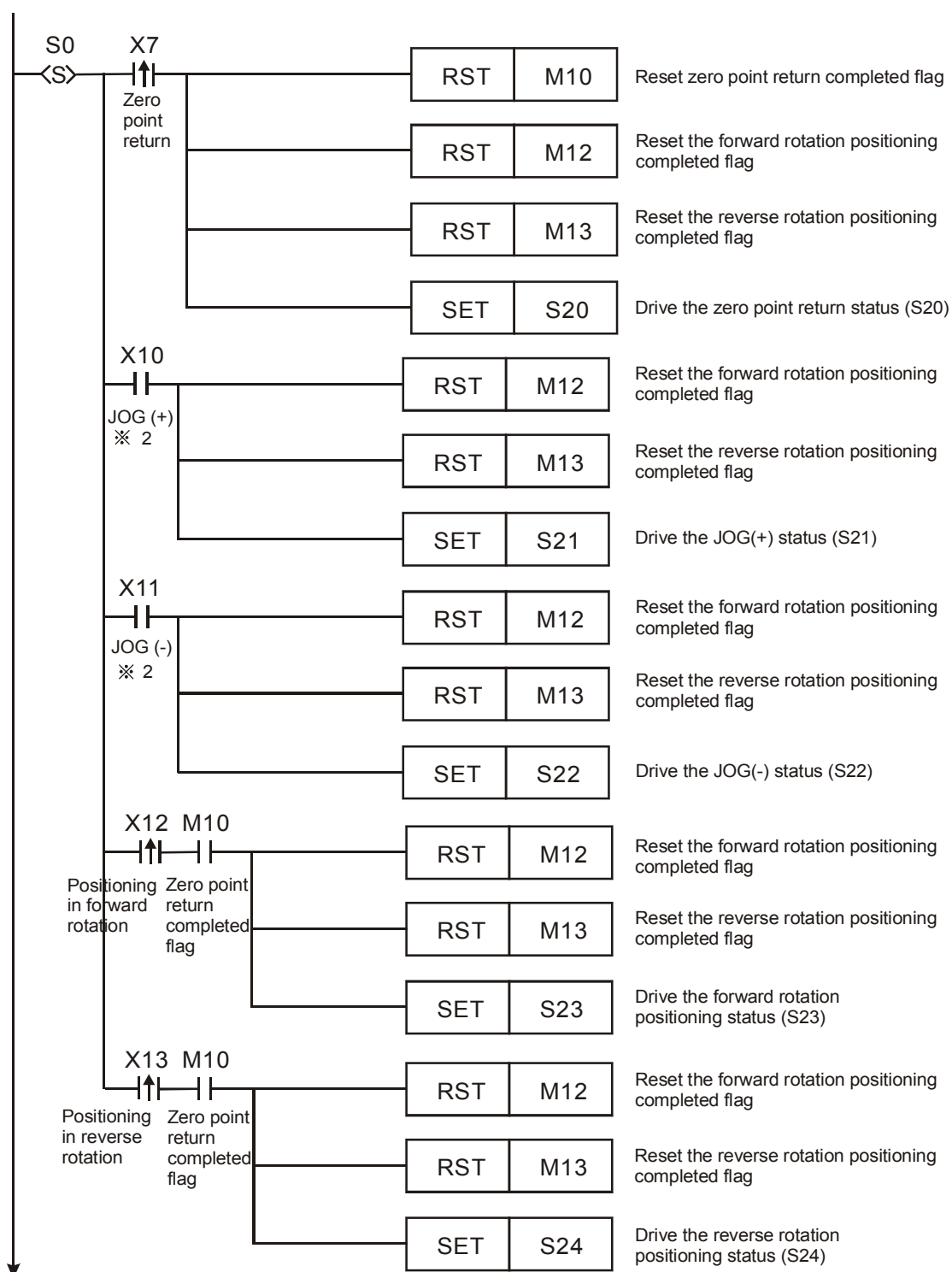
Минимальная вых. частота:

$$\sqrt{200,000 \text{ Hz} \div (2 \times (100 \text{ ms} \div 1000))} = 1,000 \text{ Hz}$$

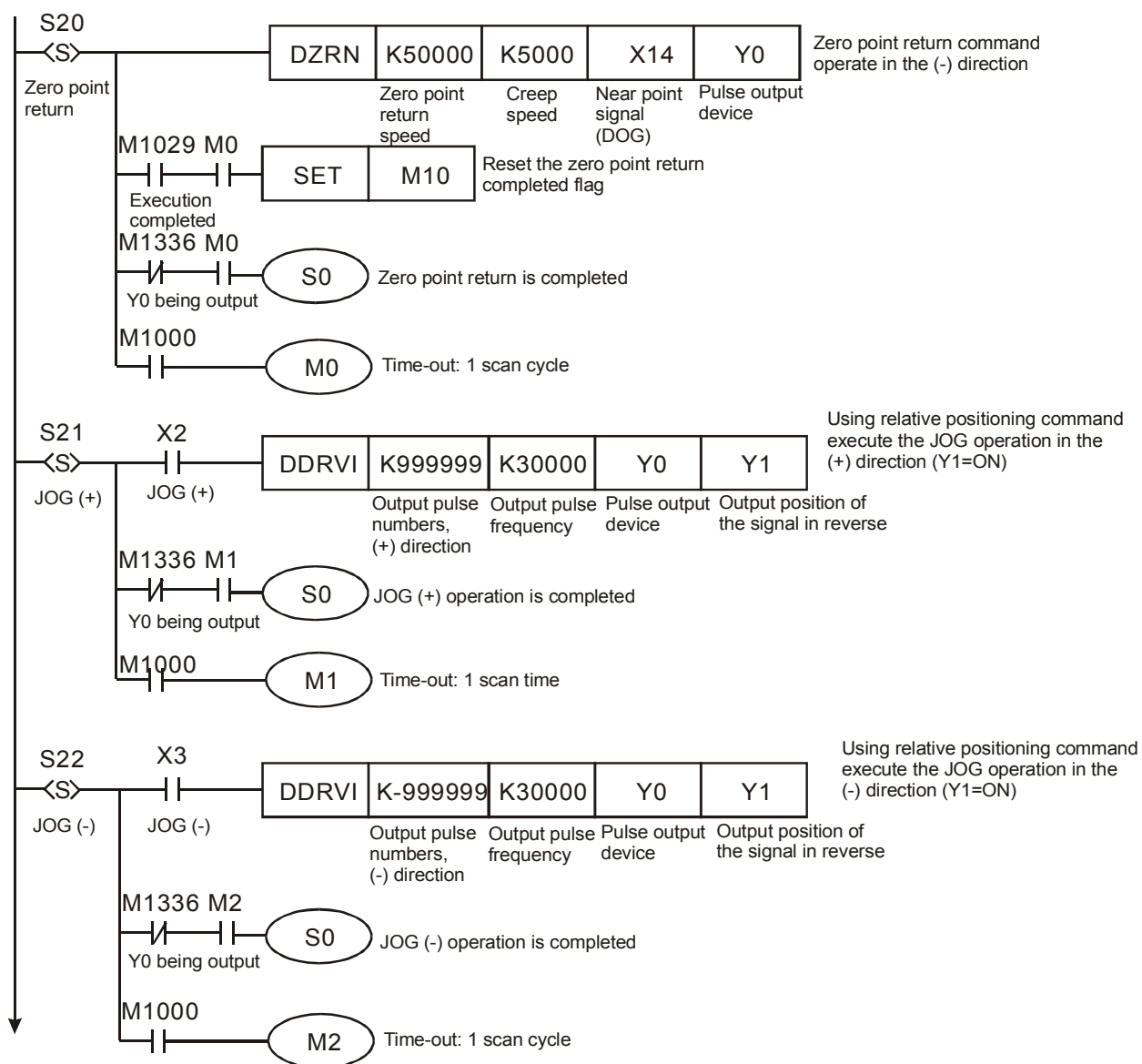
Программа с использованием пошаговых команд (STL):

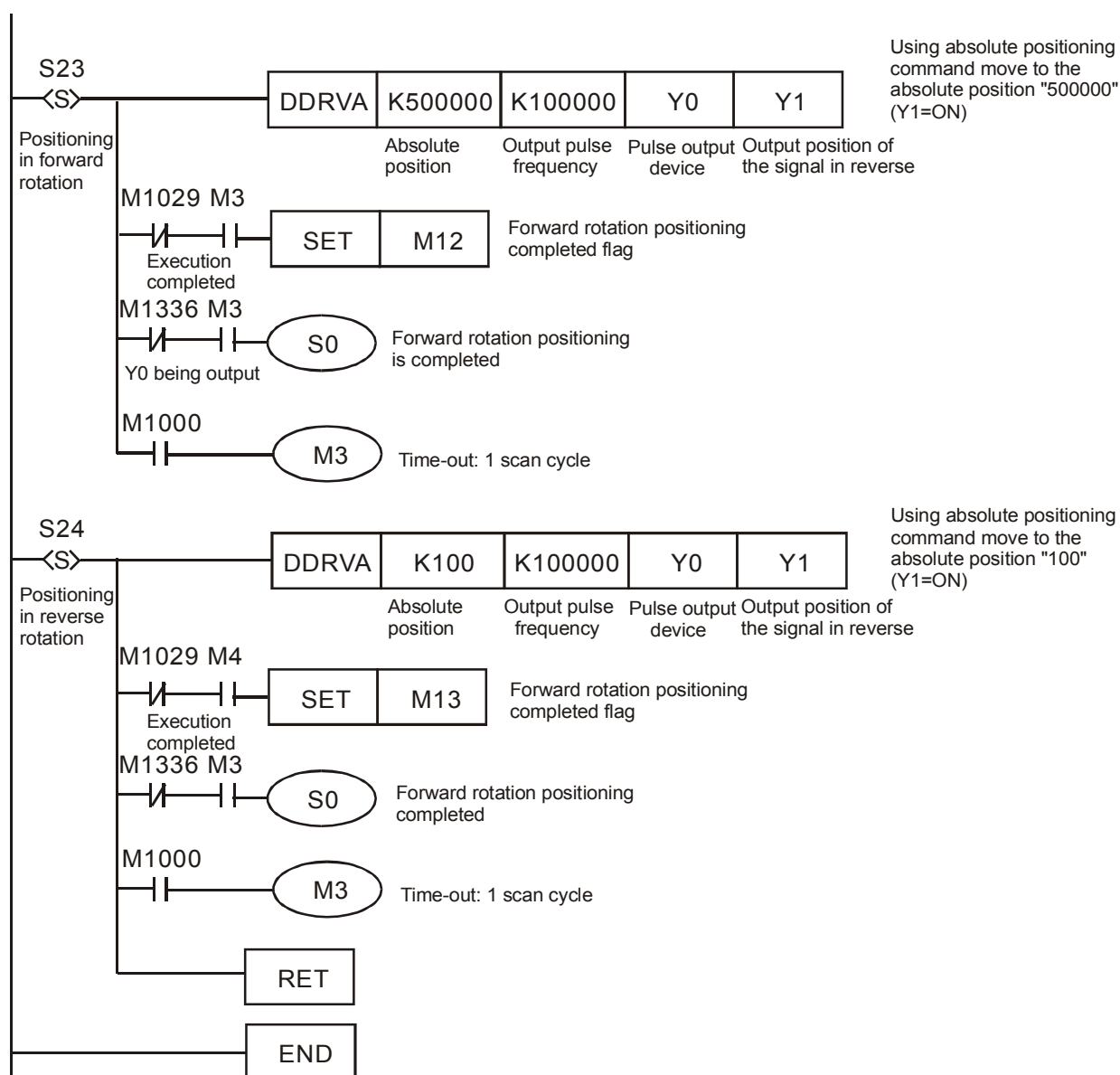







*1 D1343 должно быть больше 100 мс.



*2 Максимальное перемещение от JOG-команд равно максимальному значению DDRVI-инструкции. Для большего перемещения нажмите JOG повторно.





API	TCMP	P	    	Сравнение данных времени	DVP-		
160					ES/EX/SS	SA/SX	EH
					-	+	+

Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S ₁					*	*	*	*	*	*	*	*	*	*	*
S ₂					*	*	*	*	*	*	*	*	*	*	*
S ₃					*	*	*	*	*	*	*	*	*	*	*
S											*	*	*		
D		*	*	*											
<u>Примечания:</u> Диапазон операндов S ₁ , S ₂ , S ₃ : S ₁ =0~23, S ₂ =S ₃ =0~59 Операнд S занимает три последовательных адреса. Операнд D занимает три последовательных адреса.															

<u>16-ти битная инструкция</u> (11 шагов) TCMP - Непрерывное выполнение. TCMPP - Импульсное выполнение.
<u>32-х битная инструкция</u> ---
Флаги: нет

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 три 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).

Внимание! Сработавший выходной операнд остается после отключения условий выполнения TCMР-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).

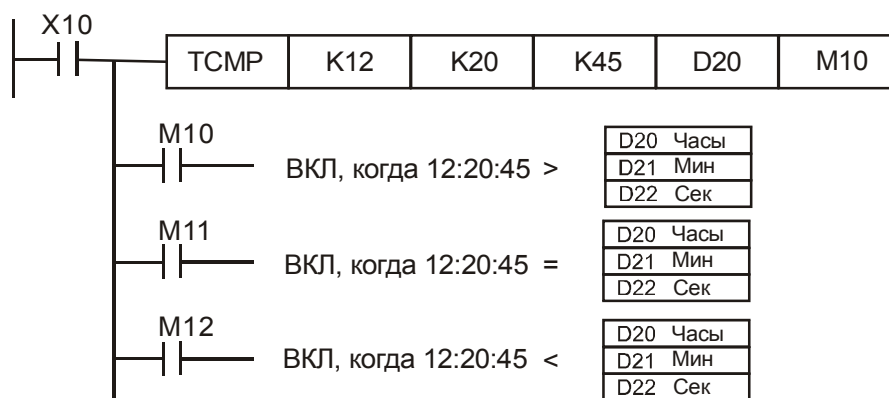
Для сравнения текущих данных часов реального времени быть использованы регистры D1315 (часы), D1314 (минуты), D1313 (секунды) в качестве исходных данных (S1), (S2) и (S3).

Если значения **(S)** превысят заданный диапазон, будет зафиксирована ошибка с кодом 0E1A (HEX) в регистре D1067. Одновременно прекратится выполнение инструкции и включатся M1067=1, M1068=1.

Пример программирования TCMР-инструкции

При включении входа X10 сравниваются заданные по K12, K20 и K45 соответственно 12 часов, 20 минут и 45 секунд с данными времени в D20 до D22.

Если значение в D20 до D22 меньше чем значение 12:20:45, включается реле M10. Если значение в D20 до D22 равно значению 12:20:45, включается реле M11. Если значение в D20 до D22 больше чем значение 12:20:45, включается реле M12.



API		TZCP		(S₁) (S₂) (S) (D)	Зонное сравнение данных времени	DVP-		
161			P			ES/EX/SS	SA/SX	EH
						-	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (9 шагов) TZCP - Непрерывное выполнение. TZCPP - Импульсное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁											*	*	*			
S ₂											*	*	*			
S											*	*	*			
D		*	*	*												
<u>Примечания:</u> Необходимое условие: S ₁ ≤ S ₂ Операнды S ₁ , S ₂ , S занимают по три последовательных адреса. Операнд D занимает три последовательных адреса.																32-х битная инструкция --- Флаги: нет

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).

Внимание! 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).

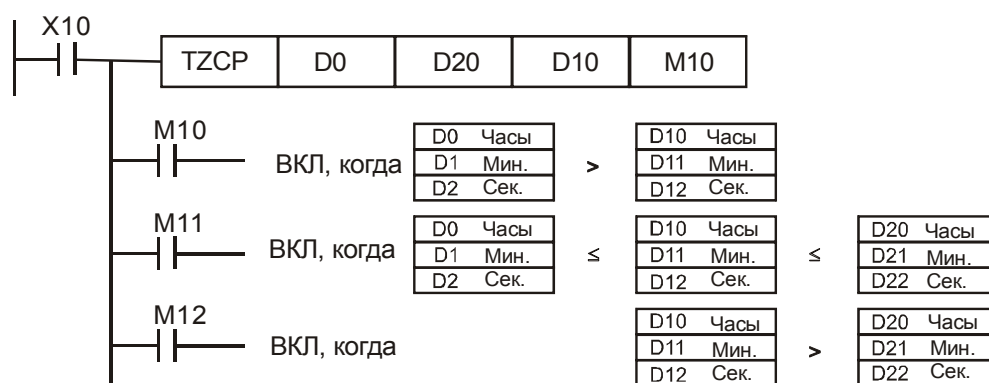
В операндах ((S₁)+1), ((S₂)+1) и ((S)+1) могут указываться значения от 0 до 59 (минут).

В операндах ((S₁)+2), ((S₂)+2) и ((S)+2) могут указываться значения от 0 до 59 (секунд).

Если значения в S, S₁, S₂ превысят заданный диапазон, будет зафиксирована ошибка с кодом 0E1A (HEX) в регистре D1067. Одновременно прекратится выполнение инструкции, и включатся M1067=1, M1068=1.

Пример программирования TZCP-инструкции

При включении входа X10 сравниваются данные времени и по результату включаются соответствующие внутренние реле M10 – M12.



API		TADD		S₁ S₂ D	Сложение данных времени	DVP-		
162			P			ES/EX/SS	SA/SX	EH
						-	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (7 шагов) TADD - Непрерывное выполнение. TADDP - Импульсное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S ₁											*	*	*				
S ₂											*	*	*				
D											*	*	*				
<u>Примечания:</u> Операнды S ₁ , S ₂ , D занимают по три последовательных адреса.																32-х битная инструкция --- Флаги: M1020 (Zero flag) M1022 (Carry flag)	

Функция

Сложение данных времени и сохранение результата

Описание

- Данные времени в (S₁) до ((S₁)+2) суммируются с данными времени в (S₂) до ((S₂)+2) и результат сохраняется в (D) до ((D)+2)

Расчет выполняется безошибочно относительно перехода времени (секунды-минуты и минуты-часы).

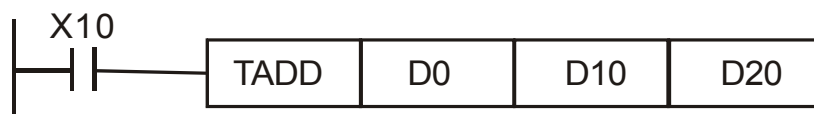
- Если результат сложения больше, чем 24 часа, выполняется переключение на 0 часов ("следующий день"), и включается флаг передачи (carry) M1022.
- Если результат сложения 0 (00:00:00, 0 часов, 0 минут, 0 секунд), то включается флаг нуля (zero) 1020.

Если значения в S₁, S₂ превышают заданный диапазон, будет зафиксирована ошибка с кодом 0E1A (HEX) в регистре D1067. Одновременно прекратится выполнение инструкции, и включатся M1067=1, M1068=1.

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.

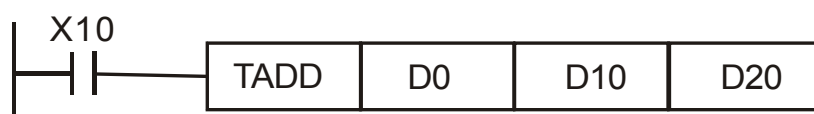
Пример программирования TADD-инструкции

При включении входа X10 к данным времени в D0 до D2 прибавляются данные времени из D10 до D12. Результат сохраняется в D20 до D22.



D0 8 час	+	D10 6 час	→	D20 14 час
D1 10 мин		D11 40 мин		D21 50 мин
D2 20 сек		D12 6 сек		D22 26 сек
08:10:20		06:40:06		14:50:26

Если результат сложения больше, чем 24 часа, выполняется переключение на 0 часов ("следующий день"), и включается флаг передачи (carry) M1022.



D0 18 час	+	D10 11 час	→	D20 6 час
D1 40 мин		D11 30 мин		D21 10 мин
D2 30 сек		D12 8 сек		D22 38 сек
18:40:30		11:30:08		06:10:38

API		TSUB		S₁ S₂ D	Вычитание данных времени	DVP-		
163			P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (7 шагов) TSUB - Непрерывное выполнение. TSUBP - Импульсное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁											*	*	*			
S ₂											*	*	*			
D											*	*	*			
<div><div>Примечания:</div><div>Операнды S₁, S₂, D занимают по три последовательных адреса.</div></div>																<div><div>32-х битная инструкция</div><div>---</div><div>Флаги: M1020 (Zero flag) M1021 (Borrow flag)</div></div>

Функция

Вычитание данных времени и сохранение результата

Описание

- Данные времени в (S2) до ((S2)+2) вычитаются из данных времени в (S1) до ((S1)+2) и результат сохраняется в (D) до ((D)+2)
- Расчет выполняется безошибочно относительно перехода времени (секунды-минуты и минуты-часы).
- Если результат вычитания меньше, чем 0 часов (00:00:00), то остаток от 24 часов заимствуется ("предыдущий день"), и включается флаг заимствования (borrow) M1021.

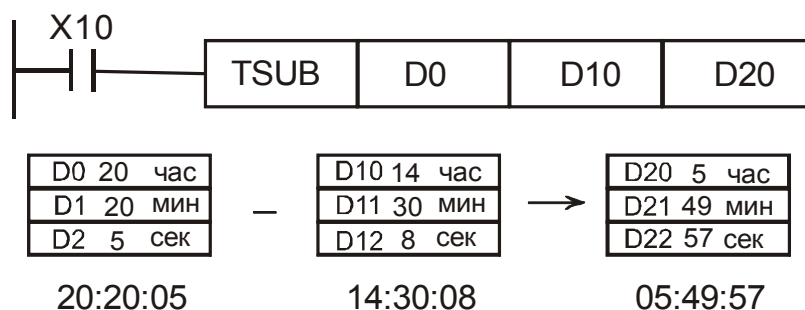
- Если результат вычитания 0 (00:00:00, 0 часов, 0 минут, 0 секунд), то включается флаг нуля (zero) M1020.

Если значения в S1, S2 превышают заданный диапазон, будет зафиксирована ошибка с кодом 0E1A (HEX) в регистре D1067. Одновременно прекратится выполнение инструкции, и включатся M1067=1, M1068=1.

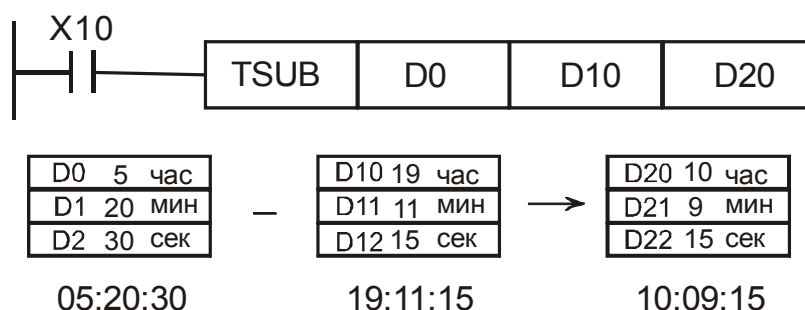
В качестве источника и цели может применяться один и тот же операнд ((S1) до ((S1)+2), (S2) до ((S2)+2). В этом случае расчетный результат снова запишется в операнд источника и может использоваться для следующих расчетов. Этот процесс повторяется в каждом цикле. Чтобы гарантировать одноразовое выполнение, применяется импульсная инструкция или блокировки.

Пример программирования TSUB-инструкции

При включении входа X10 из данных времени в D0 до D2 вычитутся данные времени D10 до D12. Результат сохраняется в D20 до D22.



Если результат вычитания меньше, чем 0 часов (00:00:00), то остаток от 24 часов заимствуется ("предыдущий день"), и включается флаг заимствования (borrow) M1021.



API		TRD	P		Чтение данных времени	DVP-		
						ES/EX/SS	SA/SX	EH
166						-	+	+

Оп- еранд	Биты				Слова										16-ти битная инструкция (7 шагов)				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TRD - Непрерывное выполнение.			
D											*	*	*			TRDP - Импульсное выполнение.			
<u>Примечания:</u> Операнд D занимает 7 последовательных адресов.																<u>32-х битная инструкция</u> --- Флаги: M1016, M1017, M1076			

Функция

Чтение текущего значения часов реального 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 адресов операнда (D).

Пример программирования TRD-инструкции

С включением входа X0 считываются данные реального времени и запоминаются в регистрах D0...6.



Примечания

- Число лет читается как двухзначное число. Четырехзначное представление достигается включением специального реле M1016. Например, 2006 г.



- Производить коррекцию часов реального времени можно с помощью инструкции TWR (API 167) или WPLSoft.
- Специальные реле и регистры:

M1016	Отображение года: OFF – 2 цифры (напр, 06 год), ON – 4 цифры (напр, 2006 год)
M1017	Корректировка секунд (± 30 сек). При включении M1017, если D1313 = 0...29, то D1313 будет обнулен, если D1313 = 30...59, то D1313 будет обнулен, а в D1314 будет добавлена 1 мин.
M1076	Ошибка часов реального времени. Включается когда превышен диапазон или при низком заряде батареи. Проверяется при включении ПЛК.
D1313	Часы реального времени (RTC): отображение и коррекция секунд (00...59)
D1314	Часы реального времени (RTC): отображение и коррекция минут (00...59)
D1315	Часы реального времени (RTC): отображение и коррекция часов (00...23)
D1316	Часы реального времени (RTC): отображение и коррекция дня (01...31)
D1317	Часы реального времени (RTC): отображение и коррекция месяца (01...12)
D1318	Часы реального времени (RTC): отображение и коррекция дня недели (1...7)
D1319	Часы реального времени (RTC): отображение и коррекция года (00...99)

API		TWR		S	Запись данных времени	DVP-		
167			P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- еранд	Биты				Слова										16-ти битная инструкция (7 шагов) TWR - Непрерывное выполнение. TWRP - Импульсное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S											*	*	*				
<div>Примечания:</div> <div>Операнд D занимает 7 последовательных адресов.</div>																32-х битная инструкция --- Флаги: M1016, M1017, M1076	

Функция

Изменение текущего значения часов реального времени

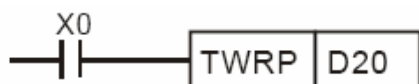
Описание

- 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).

Если значения в S превышают заданный диапазон, будет зафиксирована ошибка с кодом 0E1A (HEX) в регистре D1067. Одновременно прекратится выполнение инструкции, и включатся M1067=1, M1068=1.

Пример программирования TWR-инструкции

С включением входа X0 записываются данные времени из регистра D20...D26 как реальное время.



	Спец. регистр	Назначение	Значен ие		Регистр	Назначение	
Новые значения времени	D20	Год	00~99	→	D1319	Год	Часы реального времени
	D21	День (Пн.~Вск.)	1~7	→	D1318	День (Пн.~Вск.)	
	D22	Месяц	1~12	→	D1317	Месяц	
	D23	День месяца	1~31	→	D1316	День месяца	
	D24	Часы	0~23	→	D1315	Часы	
	D25	Минуты	0~59	→	D1314	Минуты	
	D26	Секунды	0~59	→	D1313	Секунды	

Пример 2. Требуется установить следующие значения календаря и часов реального времени: 2002/03/23, вторник, 15:27:30

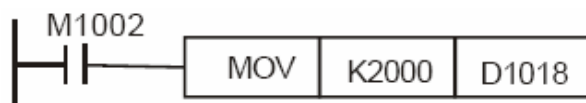
С включением входа X10 требуемые данные записываются в часы реального времени.

С помощью входа X11 можно скорректировать секунды. При включении M1017, если D1313 = 0...29, то D1313 будет обнулен, если D1313 = 30...59, то D1313 будет обнулен, а в D1314 будет добавлена 1 мин.



Примечания

- Число лет отображается как двухзначное число. Четырехзначное представление достигается включением записью в специальный регистр D1018 константы 2000.



- Производить коррекцию часов реального времени так же можно с помощью WPLSoft.

API		HOUR		S₁ S₂ D	Счетчик часов	DVP-		
169	D					ES/EX/SS	SA/SX	EH
						-	+	+

Оп- еранд	Биты				Слова										16-ти битная инструкция (7 шагов) HOUR - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
S					*	*	*	*	*	*	*	*	*	*	*
D1													*		
D2		*	*	*											
<u>Примечания:</u> Операнд D занимает 2 последовательных адреса. Инструкция может использоваться в программе не более 4 раз.															32-х битная инструкция (13 шагов) DHOUR - Непрерывное выполнение.
															Флаги: нет

Функция

По окончанию отсчета установленного времени выдается сигнал

Описание

- В (S) указывается время в часах, после отсчета которого включается операнд, указанный в (D).
- 16-ти битовая инструкция: (D1) содержит текущее значение полных часов. В ((D1+1)+1) заносится оставшееся время в секундах.
- 32-х битовая инструкция: (D1) и (D1+1) содержит данные полных часов. В ((D1+1)+2) заносится оставшееся время в секундах.
- Операнд, указанный в (D2), включается, если текущее (действительное) значение в (D1) больше, чем заданное значение в (S).

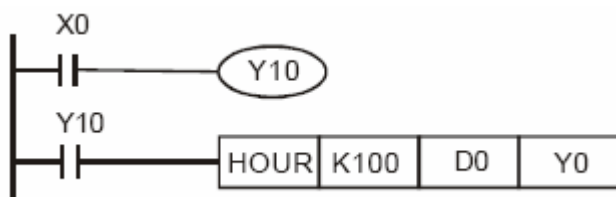
Чтобы действительное прошедшее время сохранить и при отключении напряжения, для (D1) должен быть применен регистр с памятью.

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)+1) (16-бит) или (D1) до ((D1+1)+2) (32-бита) стираются, если будет получено заданное значение.

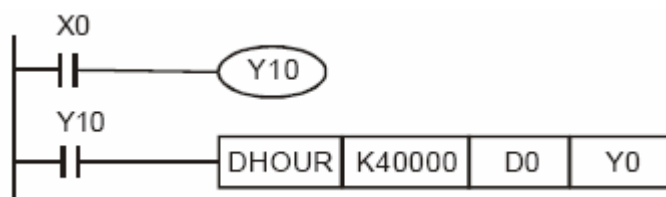
Пример программирования HOUR-инструкции

После отсчета 100 часов и секунд, после того как будет включен вход X0, включится Y0, в D0 будет записываться текущее время (ч), однако если D0 < 1 ч, то в D0 будут записаны секунды в диапазоне 0 - 3599.



Пример программирования D HOUR-инструкции

После отсчета 40000 часов и секунд, после того как будет включен вход X0, включится Y0, в D0 и D1 будет записываться текущее время (ч), однако если D0 < 1 ч, то в D2 будут записаны секунды.



API		GRY		S D	Преобразование целого числа в код Грея	DVP-		
170	D		P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (5 шагов)
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GRY - Непрерывное выполнение. GRYP - Имп. вып.
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	
<div><div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div></div></div>																

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).

Диапазон (S): 16-бит команда: 0~32767

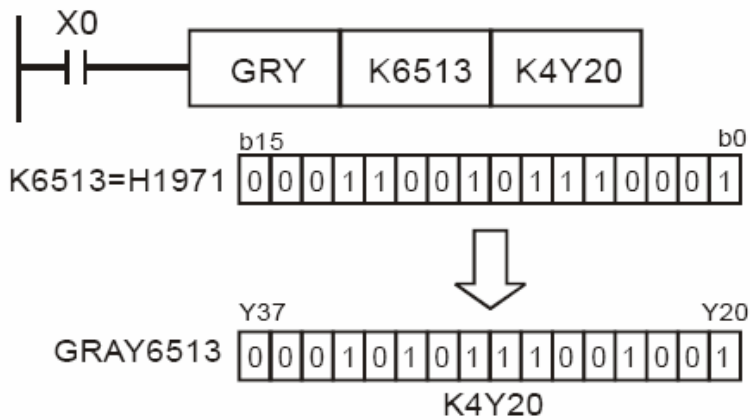
32-бит команда: 0~2147483,647

Если значения в (S) превышают заданный диапазон, будет зафиксирована ошибка с кодом 0E1A (HEX) в регистре D1067. Одновременно прекратится выполнение инструкции, и включатся M1067=1, M1068=1.

Используя характеристику кода Грея можно числовое значения посредством инкрементирования (приращения) исходных данных выдать в каждом цикле программы, не применяя стробоскопные сигналы.

Пример программирования GRY-инструкции

При включении входа X0 целое значение K6513 конвертируется в код Грея. Результат выдается на выходы Y20...Y37.



API		GBIN		<div>S</div> <div>D</div>	Преобразование кода Грея в целое число	DVP-		
171	D		P			ES/EX/SS	SA/SX	EH
						-	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (5 шагов) BIN - Непрерывное выполнение. BINP - Имп. вып.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	
Примечания:																32-х битная инструкция (9 шагов) DBIN - Непрерывное выполнение. DBINP - Имп. вып. Флаги: нет

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).

Диапазон (S): 16-бит команда: 0~32767
32-бит команда: 0~2147483,647

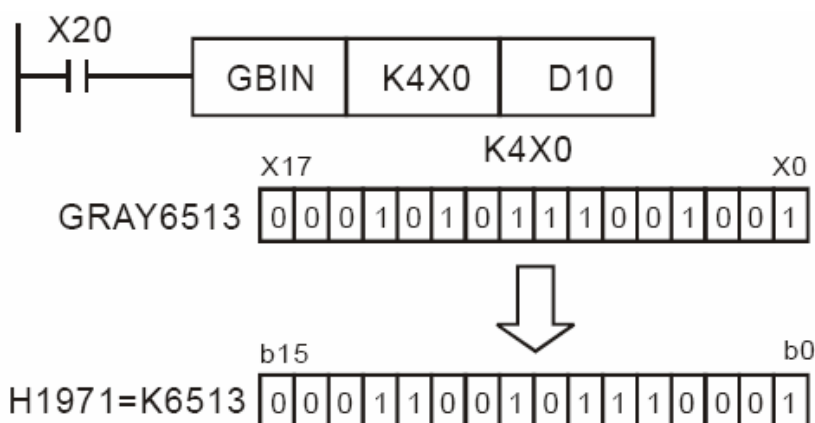
Если значения в (S) превышают заданный диапазон, будет зафиксирована ошибка с кодом 0E1A (HEX) в регистре D1067. Одновременно прекратится выполнение инструкции, и включатся M1067=1, M1068=1.

Указание: GBIN-инструкция может применяться для чтения данных с абсолютного энкодера.

Если для операндов источника применяются входы X0...X17, то время чтения может быть сокращено посредством настройки фильтров активизации (API51, REFF) или D1020.

Пример программирования GBIN-инструкции

При включении входа X20 значение в коде Грея на входах X0...X17 конвертируется в значение целого числа. Результат заносится в D10.



API		MAND		<div>S1</div> <div>S2</div> <div>D</div> <div>n</div>				Логическое умножение матриц (И)		DVP-		
180			P	ES/EX/SS		SA/SX	EH					
					-		+		+			

Оп-ранд	Биты				Слова										16-ти битная инструкция (9 шагов) MAND - Непрерывное выполнение. MANDP - Импл. вып.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S1							*	*	*	*	*	*	*			
S2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			
<div>Примечания:</div> <div>Диапазон операнда (n): 1...256</div> <div>Для серий SA/SX если операнды S1, S2, D назначены как KnX, KnY, KnM, KnS, то n может быть только равен 4.</div>																32-х битная инструкция ---
																Флаги: нет

Функция

Операция логического умножения двух матриц (логическое И)

Описание

S₁: Исходная матрица 1

S₂: Исходная матрица 2

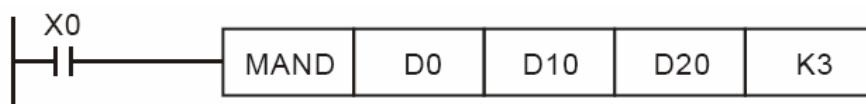
D: Результирующая матрица

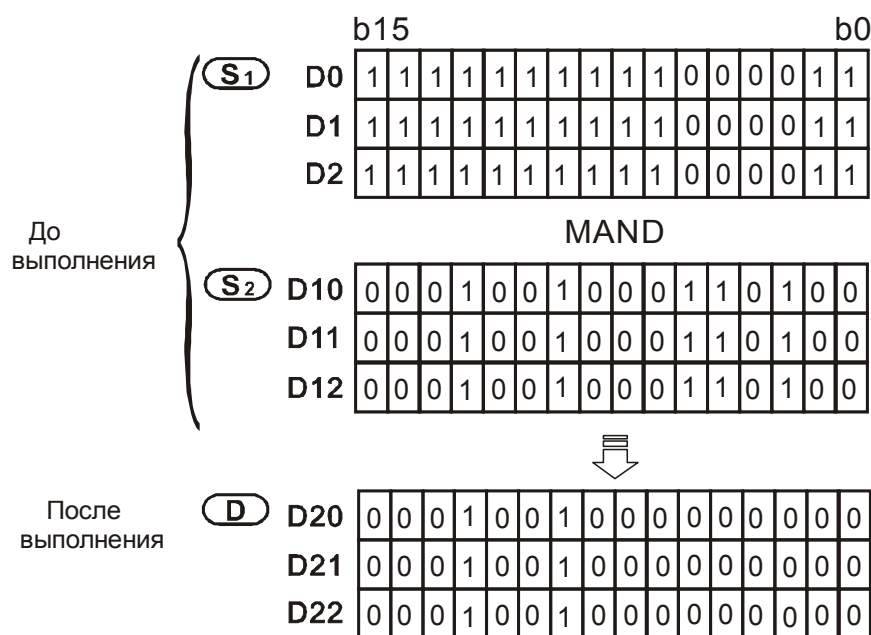
n: Число строк матрицы (n=K1~K256)

- Производится операция логического умножения между матрицами 1 и 2 длиной n и результат сохраняется в D.

Пример программирования:

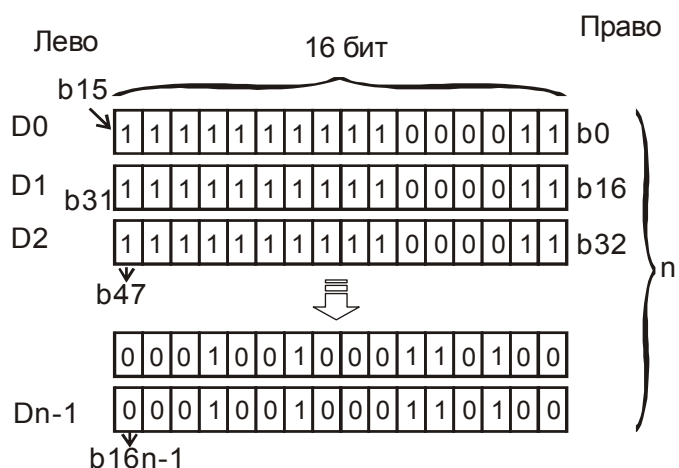
Когда X0=1, производится логическое умножение матриц состоящих из трех строк по 16 бит (D0-D2) и (D10-D12). Результат сохраняется в (D20-D22).





Примечания для матричных команд:

- Значения матрицы нумеруются с права на лево. Это важно для операции сдвига и вращения.



- В 16-бит регистре содержатся 16 значений строки матрицы ($b_0 - b_{16n-1}$). Операнд (n) определяет число строк матрицы, а следовательно количество последовательных 16-ти битных регистров. Матрица: 16 x n бит (точек).
- Матрица имеет фиксированную ширину 16 бит и длину от 1 до 256
- Пример матрицы определенной как: D0 и n=3, D0=HAAAAA, D1=H55555, D2=HAAFF

	C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
R ₀	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	D0
R ₁	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D1
R ₂	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	D2

- Пример матрицы определенной как: K2X0 и n=3, K2X0=H37, K2X10=H68, K2X20=H45

	C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
R ₀	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X ₀ ~X ₇
R ₁	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X ₁₀ ~X ₁₇
R ₂	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X ₂₀ ~X ₂₇

API		MOR		S ₁	S ₂	D	n	Логическое сложение матриц (ИЛИ)	DVP-		
181			P						ES/EX/SS	SA/SX	EH
									-	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (9 шагов) MOR - Непрерывное выполнение. MORP - Имп. вып.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S ₁							*	*	*	*	*	*	*				
S ₂							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*							*				
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S1, S2, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.															32-х битная инструкция --- Флаги: нет		

Функция

Операция логического сложения двух матриц (логическое ИЛИ)

Описание

S₁: Исходная матрица 1

S₂: Исходная матрица 2

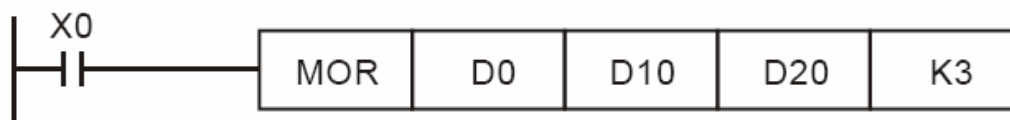
D: Результирующая матрица

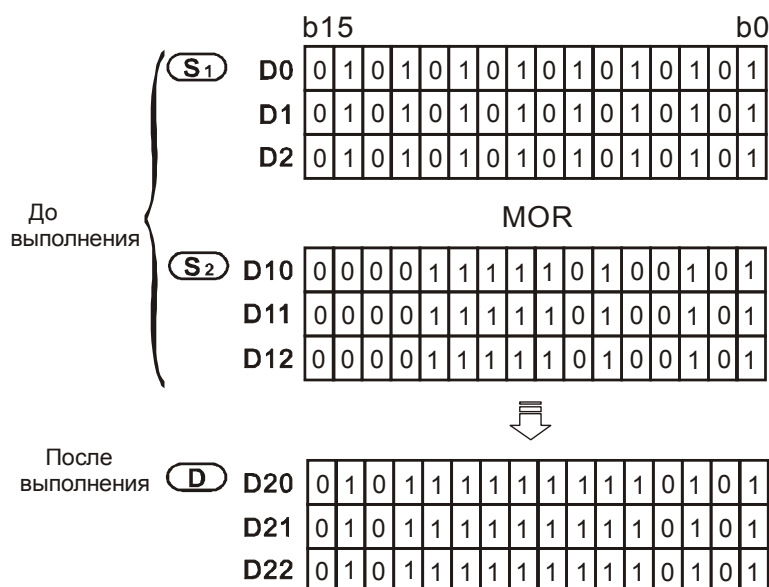
n: Число строк матрицы (n=K1~K256)

- Производится операция логического сложения между матрицами 1 и 2 длиной n и результат сохраняется в D.

Пример программирования:

Когда X0=1, производится логическое сложение матриц состоящих из трех строк по 16 бит (D0-D2) и (D10-D12). Результат сохраняется в (D20-D22).





API	MXOR	P	$\textcircled{S_1}$ $\textcircled{S_2}$ \textcircled{D} \textcircled{n}	"Исключающее ИЛИ" для матриц	DVP-		
182					ES/EX/SS	SA/SX	EH
					-	+	+

Оп- ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S ₁							*	*	*	*	*	*	*		
S ₂							*	*	*	*	*	*	*		
D								*	*	*	*	*	*		
n					*	*							*		
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S1, S2, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.															

16-ти битная инструкция
(9 шагов)
MXOR - Непрерывное
выполнение.
MXORP - Импл. вып.

32-х битная инструкция

Флаги: нет

Функция

Операция "исключающее ИЛИ" для двух матриц

Описание

S₁: Исходная матрица 1

S₂: Исходная матрица 2

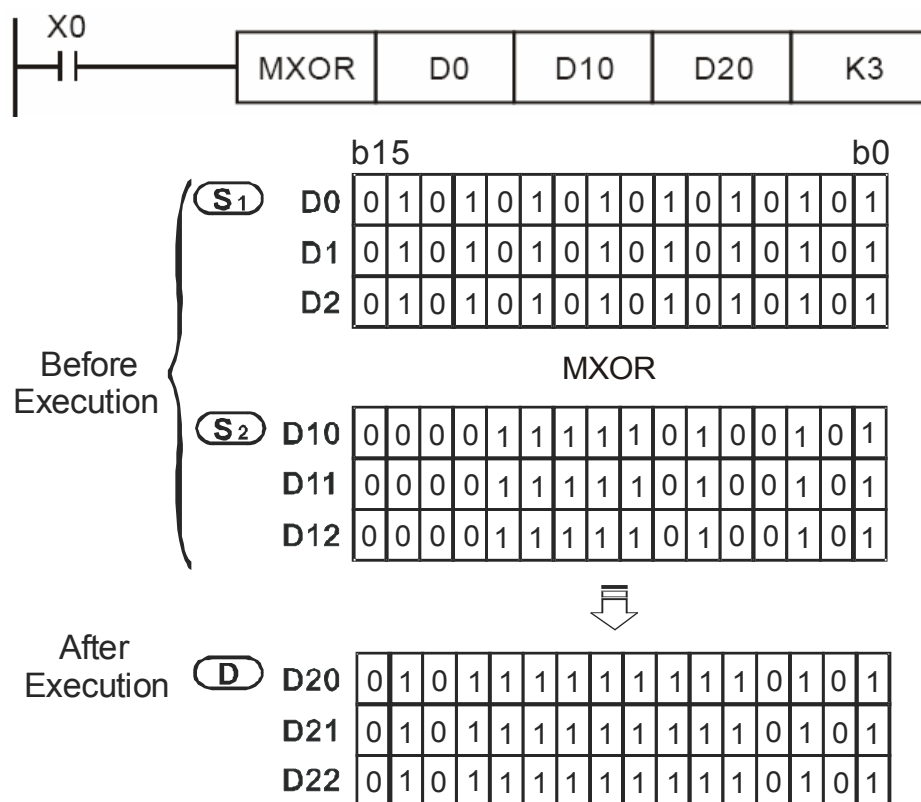
D: Результирующая матрица

n: Число строк матрицы (**n**=K1~K256)

- Производится операция "исключающее ИЛИ" между матрицами 1 и 2 длиной **n** и результат сохраняется в **D**.

Пример программирования:

Когда X0=1, производится "исключающее ИЛИ" для матриц состоящих из трех строк по 16 бит (D0-D2) и (D10-D12). Результат сохраняется в (D20-D22).



API										"Исключающее НЕ-ИЛИ"	DVP-		
		MXNR		S₁	S₂	D	n			для матриц	ES/EX/SS	SA/SX	EH
183			P								-	+	+

Опе- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (9 шагов) MXNR - Непрерывное выполнение. MXNRP - Импл. вып.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S ₁							*	*	*	*	*	*	*				
S ₂							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*							*				
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S1, S2, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.																<u>32-х битная инструкция</u> --- Флаги: нет	

Функция

Операция "исключающее НЕ-ИЛИ" для двух матриц

Описание

S₁: Исходная матрица 1

S₂: Исходная матрица 2

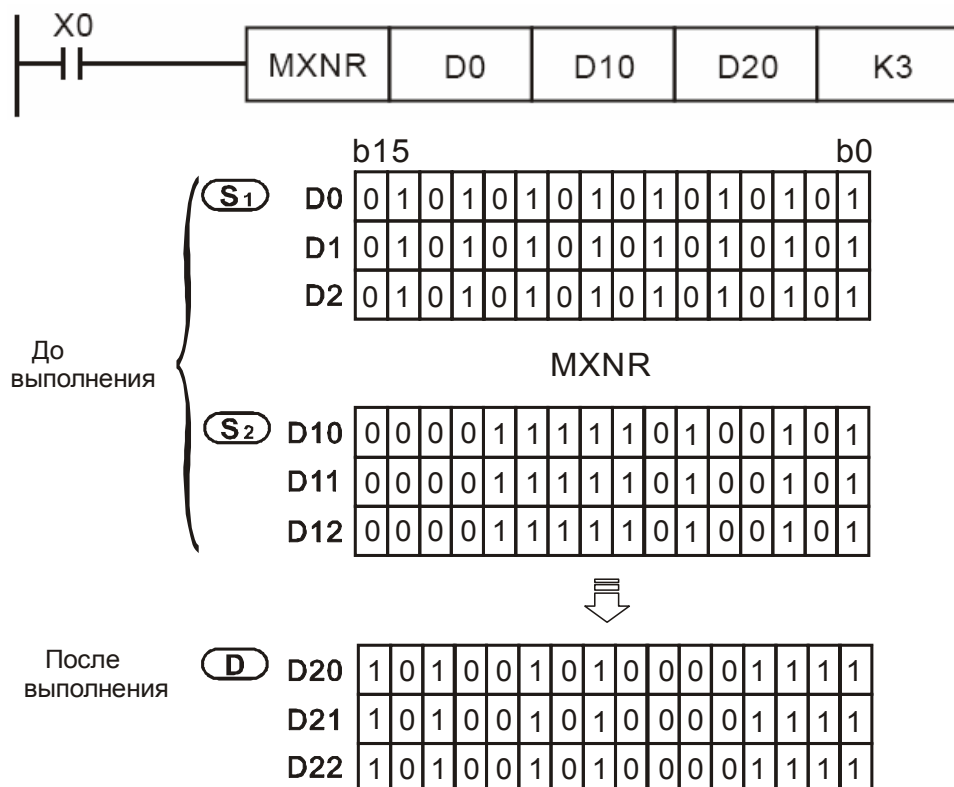
D: Результирующая матрица

n: Число строк матрицы (n=K1~K256)

- Производится операция "исключающее НЕ-ИЛИ" между матрицами 1 и 2 длиной **n** и результат сохраняется в **D**.

Пример программирования:

Когда X0=1, производится "исключающее НЕ-ИЛИ" для матриц состоящих из трех строк по 16 бит (D0-D2) и (D10-D12). Результат сохраняется в (D20-D22).



API	MINV	P	S	D	n	Инверсия матрицы	DVP-		
							ES/EX/SS	SA/SX	EH
184							-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (9 шагов) MINV - Непрерывное выполнение. MINVP - Импл. вып.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*							*				
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S1, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.																32-х битная инструкция --- Флаги: нет	

Функция

Изменение значения каждого бита в матрице на противоположное

Описание

S: Исходная матрица

D: Результирующая матрица

n: Число строк матрицы (n=K1~K256)

- Производится операция инверсии матрицы длиной **n** и результат сохраняется в **D**.

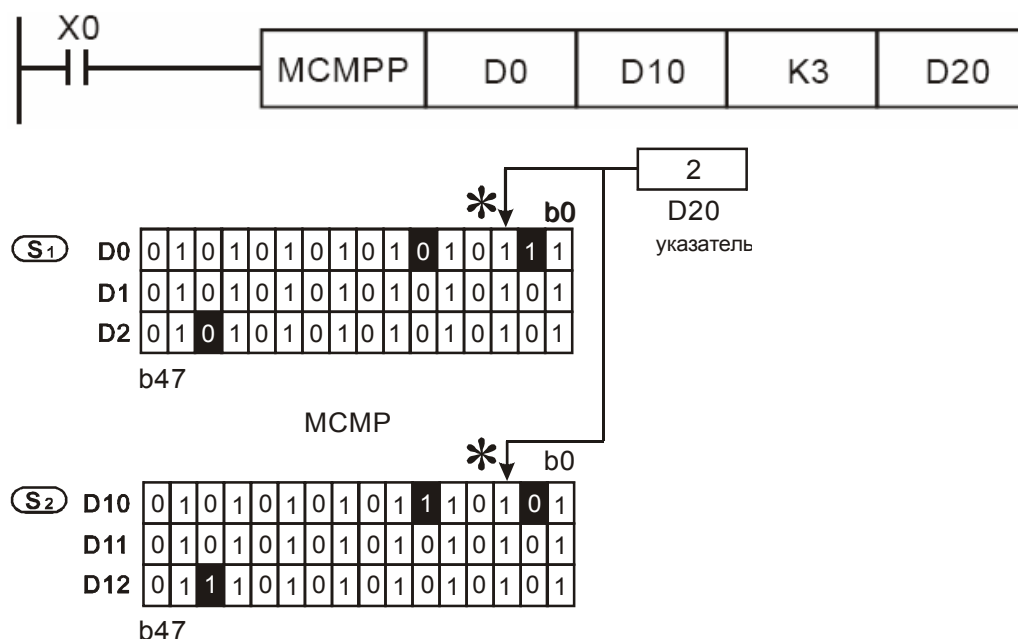
- Сразу после завершения сравнения матриц будет выставлен флаг M1091=1. Сравнение будет завершаться после каждого несовпадения битов в матрицах.
- Когда сравнение достигает последнего бита, выставляется флаг M1089=1 и номер последнего бита будет записан в операнд **D**
- Когда идет сравнение первого бита, выставляется флаг M1090=1
- Если указатель (D), превышает заданный диапазон, то выставляется флаг M1092=1 и команда не будет выполнена.

Пример программирования:

Когда X0 переключится с 0 на 1, начнется побитовое сравнение (M1090=0). (M1088=0 если один или несколько битов в матрицах различаются)

Когда текущее значение указателя D20=2, можно получить следующие результаты (1, 2, 3, 4) при переключении X0 с 0 на 1 четыре раза.

- ① D20=5, M1091=1, M1089=0.
- ② D20=45, M1091=1, M1089=0.
- ③ D20=47, M1091=0, M1089=1.
- ④ D20=1, M1091=1, M1089=0.



API	MBRD	P	<div> <div>S</div> <div>n</div> <div>D</div> </div>	Чтение битов в матрице	DVP-		
186					ES/EX/SS	SA/SX	EH
					-	+	+

Оп- ранд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S							*	*	*	*	*	*	*		
n					*	*							*		
D								*	*	*	*	*	*	*	*
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.															

16-ти битная инструкция
(5 шагов)
MBRD - Непрерывное
выполнение.
MBRDP - Имп. вып.

32-х битная инструкция

Флаги: M1089-M1095

API	MBWR	P	<div> <div>S</div> <div>n</div> <div>D</div> </div>	Запись битов в матрицу	DVP-		
187					ES/EX/SS	SA/SX	EH
					-	+	+

Опе- ранд	Биты				Слова										16-ти битная инструкция (5 шагов) MBWR - Непрерывное выполнение. MBWRP - Импл. вып.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		
S							*	*	*	*	*	*	*			
n					*	*							*			
D								*	*	*	*	*	*	*	*	
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.															32-х битная инструкция --- Флаги: M1089-M1095	

Функция

Запись значений отдельных битов в матрицу

Описание

S: Исходная матрица

n: Число строк матрицы (**n**=K1~K256)

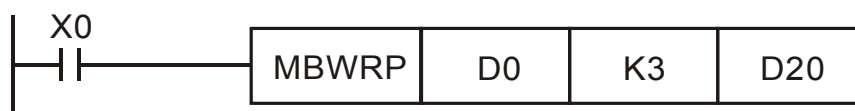
D: Указатель записи

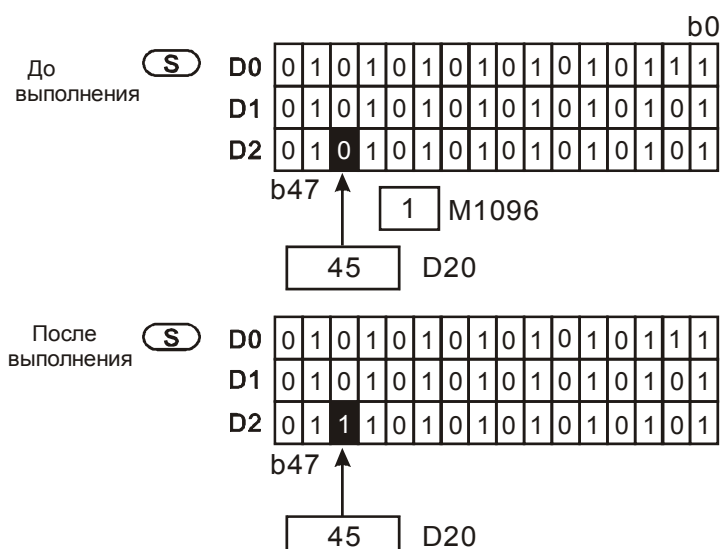
- Когда команда будет выполнена, то для следующего старта надо, чтобы M1094 (флаг сброса указателя) = 1. Тем самым, значение указателя **D** будет равно 0 и начнется запись **S** с 0-бита и состояние (0/1) каждого бита будет зависеть от состояния специального реле M1096. Если M1093 (флаг увеличения указателя) = 1, то каждый считанный бит будет увеличивать значение **D** на единицу. Когда будет записан последний бит в матрице, M1089 (флаг конца матрицы) будет =1, в указателе **D** будет записано число записанных бит и выполнение команды завершится.
- Если указатель (D), превышает заданный диапазон, то выставляется флаг M1092=1 и команда не будет выполнена.

Пример программирования:

2. Когда X0 переключится с 0 на 1, флаг сброса указателя M1094=1, флаг увеличения указателя M1093=1, и после каждого записанного бита, значение указателя будет увеличиваться на единицу.

Когда D20=45, M1094=1, будет получен следующий результат при переключении X0 с 0 на 1 один раз.





API	MBS	P	(S) (D) (n)	Сдвиг битов в матрице	DVP-		
					ES/EX/SS	SA/SX	EH
188					-	+	+

Оп- ранд	Биты				Слова										16-ти битная инструкция (5 шагов) MBS - Непрерывное выполнение. MBSP - Имп. вып.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*	*	*		
n					*	*							*				
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.																32-х битная инструкция --- Флаги: M1095-M1097	

Функция

Сдвиг значений битов в матрице

Описание

S: Исходная матрица

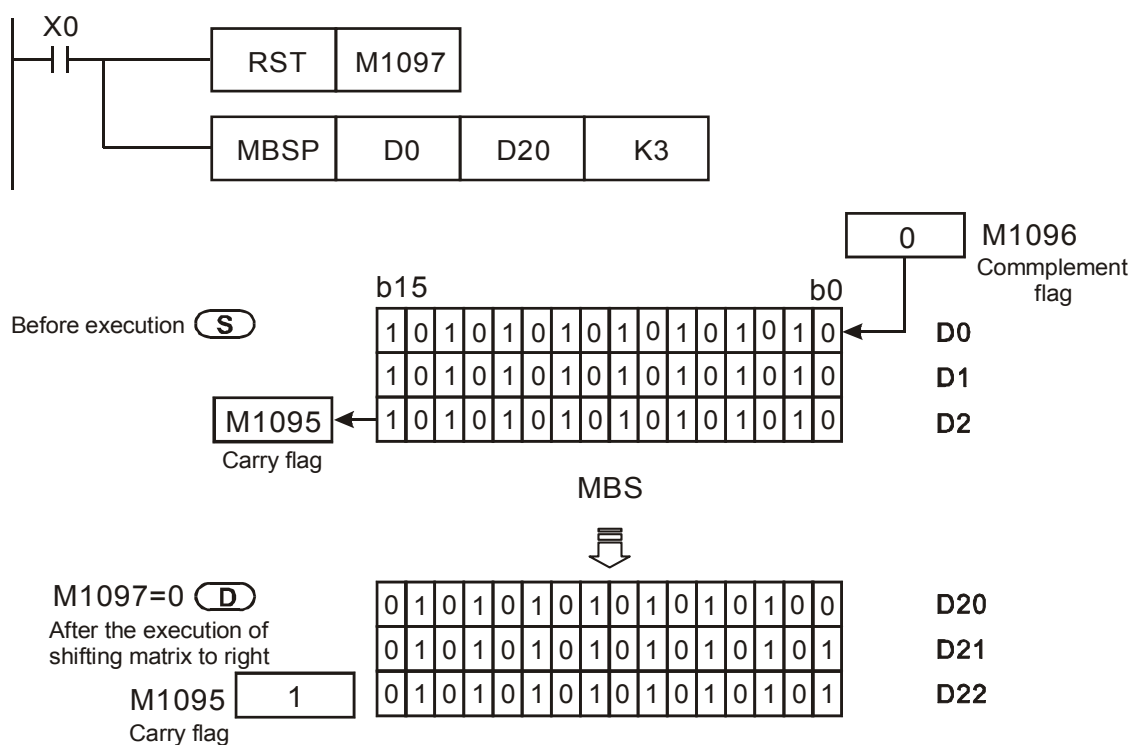
D: Результирующая матрица

n: Число строк матрицы (n=K1~K256)

- Направление сдвига определяется флагом M1097: если M1097=0 – сдвиг влево; если M1097=1 – сдвиг вправо.

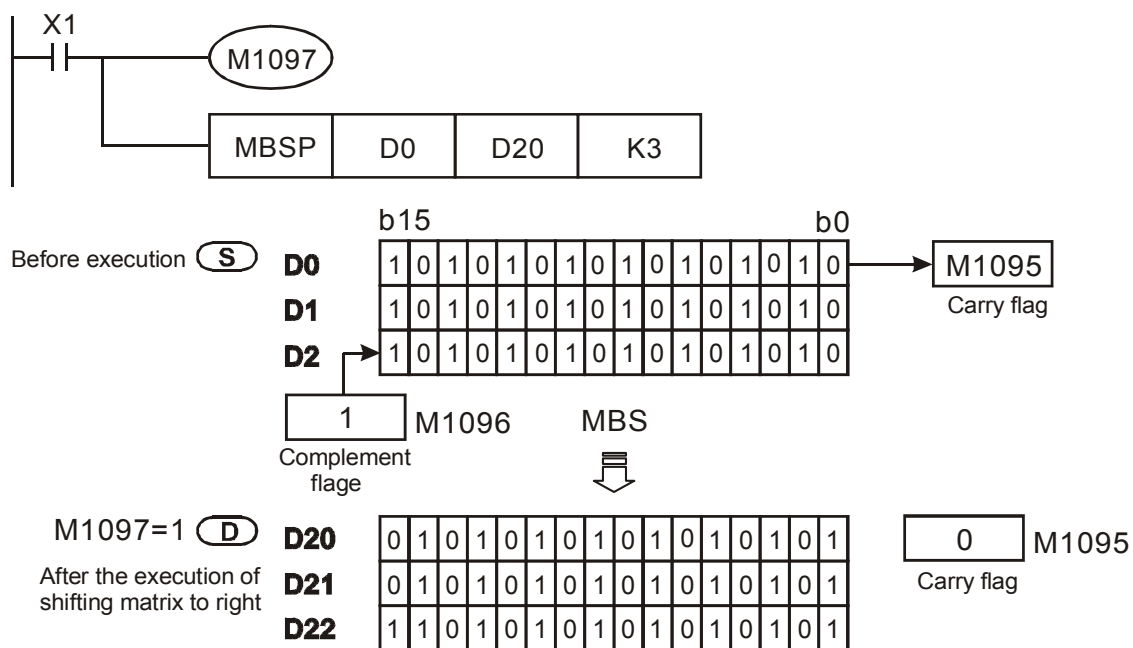
Пример программирования 1:

Когда X0 = 1, флаг направления M1097=0. Установим флаг дополнения M1096=0, произойдет смещение битов влево в регистрах D0-D2 и результат будет сохранен в D20-D22 и флаг переноса (carry) M1095 будет установлен в 1.



Пример программирования 2:

Когда X0 = 1, флаг направления M1097=1. Установим флаг дополнения M1096=1, произойдет смещение битов влево в регистрах D0-D2 и результат будет сохранен в D20-D22 и флаг переноса (carry) M1095 будет установлен в 0.



API		MBR	☺	(S) (D) (n)	Вращение битов в матрице	DVP-		
189			P			ES/EX/SS	SA/SX	EH
						-	+	+

Оп- еранд	Биты				Слова										<u>16-ти битная инструкция</u> (5 шагов) MBR - Непрерывное выполнение. MBRP - Имп. вып.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*	*	*		
n					*	*							*				
<u>Примечания:</u> Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.																<u>32-х битная инструкция</u> --- Флаги: M1095, M1097	

ФУНКЦИЯ

Вращение (кольцевой сдвиг) значений битов в матрице

ОПИСАНИЕ

S: Исходная матрица

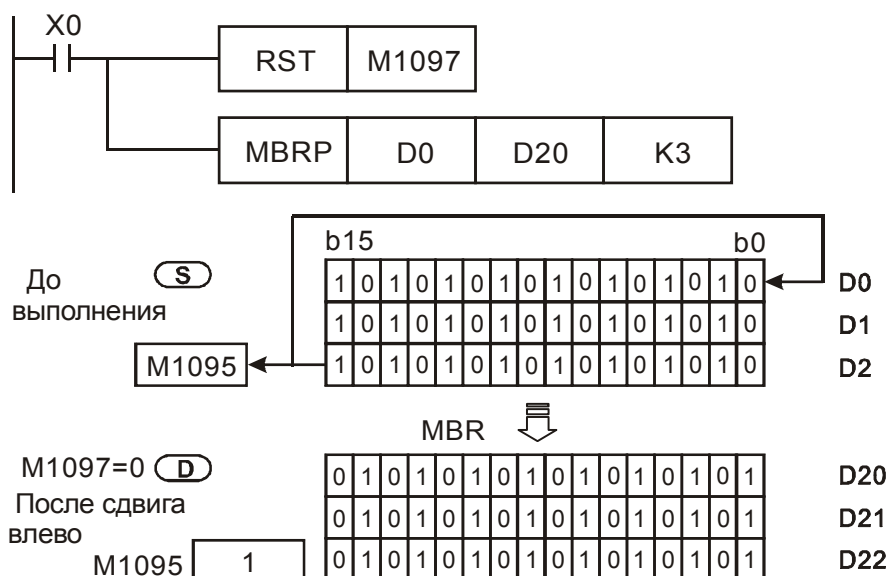
D: Результирующая матрица

n: Число строк матрицы ($n=K1 \sim K256$)

- Направление вращения определяется флагом M1097: если M1097=0 – вращение влево; если M1097=1 – вращение вправо.

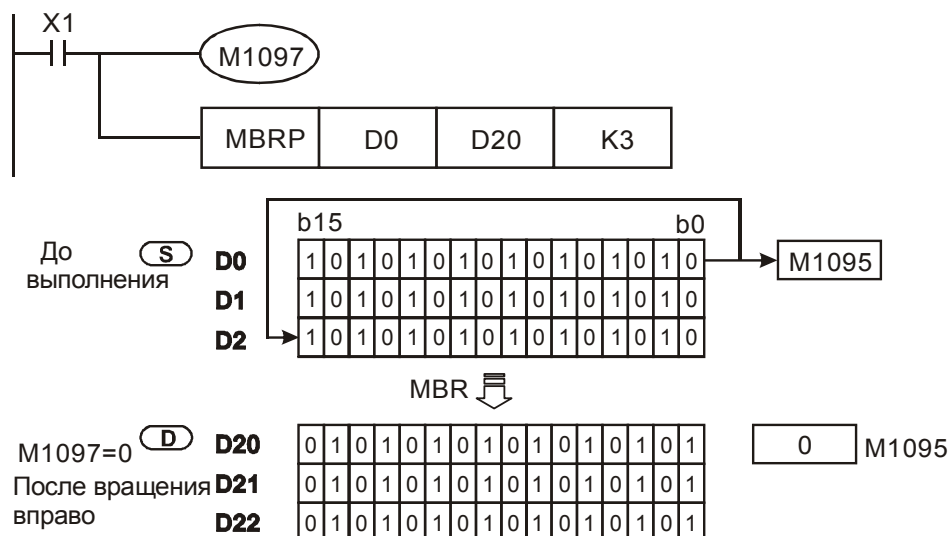
Пример программирования 1:

Когда X0 = 1, флаг направления M1097=0. Произойдет кольцевой сдвиг битов влево в регистрах D0-D2 и результат будет сохранен в D20-D22 и флаг переноса (carry) M1095 будет установлен в 1.



Пример программирования 2:

Когда $X0 = 1$, флаг направления $M1097=1$. Произойдет кольцевой сдвиг битов вправо в регистрах D0-D2 и результат будет сохранен в D20-D22 и флаг переноса (carry) M1095 будет установлен в 0.



API	MBC	P	(S) (n) (D)	Счетчик битов в матрице	DVP-		
					ES/EX/SS	SA/SX	EH
190					-	+	+

Оп-ранд	Биты				Слова										16-ти битная инструкция (7 шагов) MBC - Непрерывное выполнение. MBCP - Имп. вып.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S							*	*	*	*	*	*	*				
n					*	*							*				
D								*	*	*	*	*	*	*	*		
Примечания: Диапазон операнда (n): 1...256 Для серий SA/SX если операнды S, D назначены как KnX, KnY, KnM, KnS, то n может быть равен только 4.																32-х битная инструкция --- Флаги: M1098, M1099	

Функция

Подсчет единичных или нулевых битов в матрице и сохранение результата в (D)

Описание

S: Исходная матрица

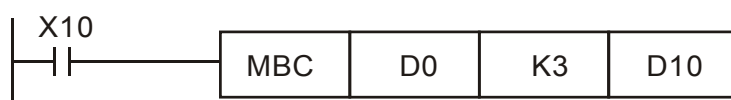
n: Число строк матрицы ($n=K1 \sim K256$)

D: Результат

- Если $M1098=1$ – подсчет единичных битов; если $M1098=0$ – подсчет нулевых битов.
- Если результат подсчета $=0$, будет выставлен флаг $M1099=1$.

Пример программирования:

Когда $X10 = 1$, произойдет подсчет битов в регистрах D0-D2 и результат будет сохранен в D10.



D0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1

D10 12 M1098=0

D10 36 M1098=1

API		HST		S	Высокоскоростной таймер	DVP-		
						ES/EX/SS	SA/SX	EH
196			P			-	+	+

Оп- ранд	Биты				Слова											<u>16-ти битная инструкция</u> (3 шага) NST - Непрерывное выполнение. NSTP - Имп. вып.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										<u>32-х битная инструкция</u> --- Флаги: M1015
<u>Примечания:</u> Возможные значения операнда (S): 1, 2																

Функция

Запуск и останов высокоскоростного таймера с дискретностью 100мкс

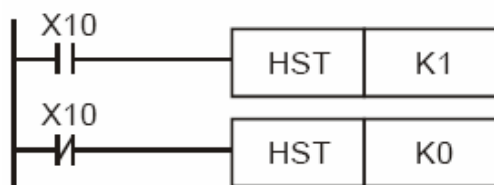
Описание

S: Условие запуска и останова высокоскоростного таймера

- Когда S=1, одновременно включается флаг индикации работы таймера M1015, и начинается отсчет времени в регистре D1015 с дискретностью 100 мкс.
- Диапазон регистра D1015: K0-K32767. Когда время достигнет верхнюю границу K32767, следующим значением будет 0 и отсчет начнется заново.
- Когда S=0, одновременно выключится флаг индикации работы таймера M1015, и отсчет времени в регистре D1015 немедленно будет остановлен.

Пример программирования:

Когда X10 = 1, M1015=1 и в значение в регистре D1015 начнет увеличиваться на 1 через каждые 100 мкс. Когда X10 = 0, M1015=0 и отсчет времени будет остановлен.



10. ПРИКЛАДНЫЕ ИНСТРУКЦИИ API 215-246

API						DVP-		
215~ 217	D	LD#		S₁ S₂	Логические операции контактного типа	ES/EX/SS	SA/SX	EH
						-	+	+

Оп- еранд	Биты				Слова											16-ти битная инструкция (5 шагов) LD# - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
Примечания: Символ #: это &, , ^																32-х битная инструкция (9 шагов) DLD# - Непрерывное выполнение. Флаги: нет

Функция

Выполнение логической операции "И", "ИЛИ", "Исключающее ИЛИ" над операндами S₁ и S₂, и включение LD-контакта в зависимости от результата операции.

Описание

- Инструкции LD# в программе располагаются крайними слева и открывают логическую связь или являются условиями выполнения правосторонних команд.
- Таблица отражает соответствие логической операции с соответствующим номером функции.

API	16 –бит команда	32 - бит команда	Контакт замкнут, если:	Контакт разомкнут, если:
215	LD&	DLD&	S ₁ & S ₂ ≠0	S ₁ & S ₂ =0
216	LD	DLD	S ₁ S ₂ ≠0	S ₁ S ₂ =0
217	LD^	DLD^	S ₁ ^ S ₂ ≠0	S ₁ ^ S ₂ =0

&: логическое умножение (И)

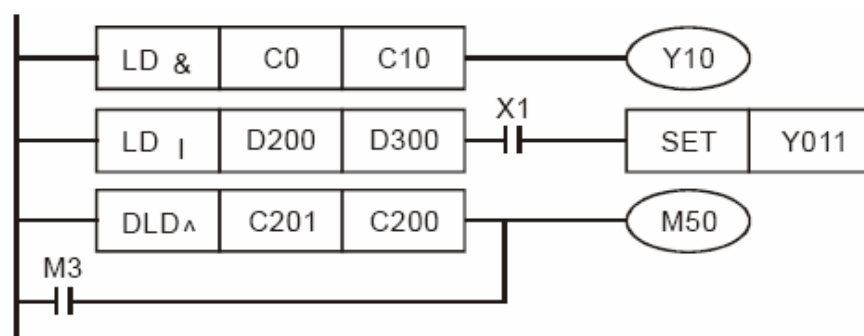
|: логическое сложение (ИЛИ)

^: исключающее ИЛИ (XOR)

- Если в качестве операнда S₁ или S₂ используется счетный регистр C200-..., используйте 32-х битную команду DLD, иначе это вызовет программную ошибку и светодиод "ERROR" будет мигать.

Пример программирования:

- Когда результат операции LD& (логическое умножение) между регистрами C0 и C10 не равен 0, Y10=ВКЛ.
- Когда результат операции LD| (логическое сложение) между значениями регистров D200 и D300 не равен 0 и X1=ВКЛ, Y10=ВКЛ с самоудержанием.
- Когда результат операции LD^ (исключающее ИЛИ) между значениями регистров C201 и C200 не равен 0 или M3=ВКЛ, M50=ВКЛ.



API				Логические операции контактного типа. Последовательный контакт.	DVP-		
218~ 220	D	AND#	S₁ S₂		ES/EX/SS	SA/SX	EH
					-	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (5 шагов) AND# - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
Примечания: Символ #: это &, , ^																32-х битная инструкция (9 шагов) DAND# - Непрерывное выполнение.
																Флаги: нет

Функция

Выполнение логической операции "И", "ИЛИ", "Исключающее ИЛИ" над операндами S₁ и S₂, и включение AND-контакта в зависимости от результата операции.

Описание

- Инструкции AND# в программе располагаются после команд LD и образуют с ними логическую связь "И".
- Таблица отражает соответствие логической операции с соответствующим номером функции.

API	16 –бит команда	32 – бит команда	Контакт замкнут, если:	Контакт разомкнут, если:
218	AND&	DAND&	S ₁ & S ₂ ≠0	S ₁ & S ₂ =0
219	AND	DAND	S ₁ S ₂ ≠0	S ₁ S ₂ =0
220	AND^	DAND^	S ₁ ^ S ₂ ≠0	S ₁ ^ S ₂ =0

&: логическое умножение (И)

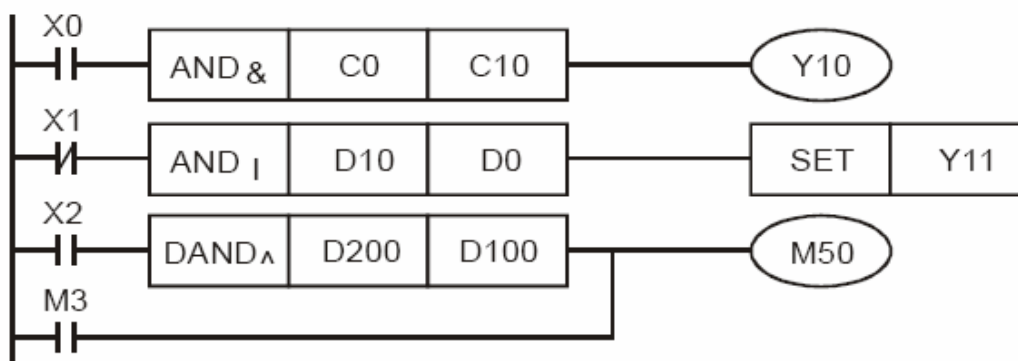
|: логическое сложение (ИЛИ)

^: исключающее ИЛИ (XOR)

- Если в качестве операнда S₁ или S₂ используется счетный регистр C200-..., используйте 32-х битную команду DAND, иначе это вызовет программную ошибку и светодиод "ERROR" будет мигать.

Пример программирования:

1. Когда X0=БКЛ, и результат операции AND& (И) между регистрами C0 и C10 не равен 0, Y10=БКЛ.
2. Когда X1=ВЫКЛ, и результат операции AND| (ИЛИ) между регистрами D10 и D0 не равен 0, Y11=БКЛ с самоудержанием.
3. Когда X2=БКЛ, и результат операции AND^ (XOR) между регистрами D200(D201) и D100(D101) не равен 0 или M3=БКЛ, M50=БКЛ.



API		OR#		Логические операции контактного типа. Параллельный контакт.	DVP-		
					ES/EX/SS	SA/SX	EH
221~223	D		S₁ S₂		-	+	+

Оп- ранд	Биты				Слова										<u>16-ти битная инструкция</u> (5 шагов) OR# - Непрерывное выполнение.	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
<u>Примечания:</u> Символ #: это &, , ^																<u>32-х битная инструкция</u> (9 шагов) DOR# - Непрерывное выполнение. Флаги: нет

Функция

Выполнение логической операции "И", "ИЛИ", "Исключающее ИЛИ" над операндами S₁ и S₂, и включение OR-контакта в зависимости от результата операции.

Описание

- Инструкция OR# в программе располагается слева, параллельно команде LD и образует с ней логическую связь "ИЛИ".
- Таблица отражает соответствие логической операции с соответствующим номером функции.

API	16 –бит команда	32 – бит команда	Контакт замкнут, если:	Контакт разомкнут, если:
221	OR&	DOR&	S ₁ & S ₂ ≠0	S ₁ & S ₂ =0
222	OR	DOR	S ₁ S ₂ ≠0	S ₁ S ₂ =0
223	OR^	DOR^	S ₁ ^ S ₂ ≠0	S ₁ ^ S ₂ =0

&: логическое умножение (И)

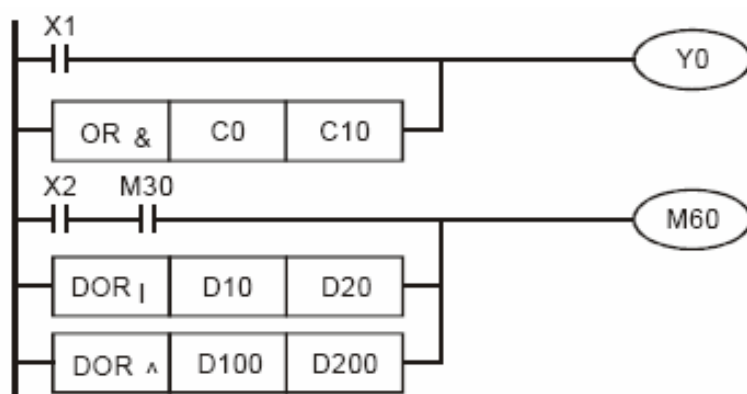
|: логическое сложение (ИЛИ)

^: исключающее ИЛИ (XOR)

- Если в качестве операнда **S1** или **S2** используется счетный регистр C200-..., используйте 32-х битную команду DOR, иначе это вызовет программную ошибку и светодиод “ERROR” будет мигать.

Пример программирования:

- Когда X1=ВКЛ, или результат операции OR& (И) между регистрами C0 и C10 не равен 0, Y0=ВКЛ.
- Если X2 и M30 = “ВКЛ”, или результат операции OR| (ИЛИ) между регистрами D10 и D20 не равен 0, или результат операции OR^ (XOR) между регистрами D100 и D200 не равен 0, M60=ВКЛ.



API		LD*		<div>S1</div> <div>S2</div>	Операции сравнения контактного типа.	DVP-		
224~ 230	D		ES/EX/SS			SA/SX	EH	
			+			+	+	

Опе- ранд	Биты				Слова										16-ти битная инструкция (5 шагов) LD* - Непрерывное выполнение.		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F	
	S1					*	*	*	*	*	*	*	*	*		*	*
	S2					*	*	*	*	*	*	*	*	*		*	*

Примечания: Символ *: это =, >, <, <>, ≤, ≥

32-х битная инструкция
(9 шагов)
DLD* - Непрерывное
выполнение.

Флаги: нет

Функция

Сравнение значений операндов **S1** и **S2**, и включение LD-контакта в зависимости от результата операции.

Описание

- Инструкции LD* в программе располагаются крайними слева и начинают логическую связь или являются условиями выполнения правосторонних команд.

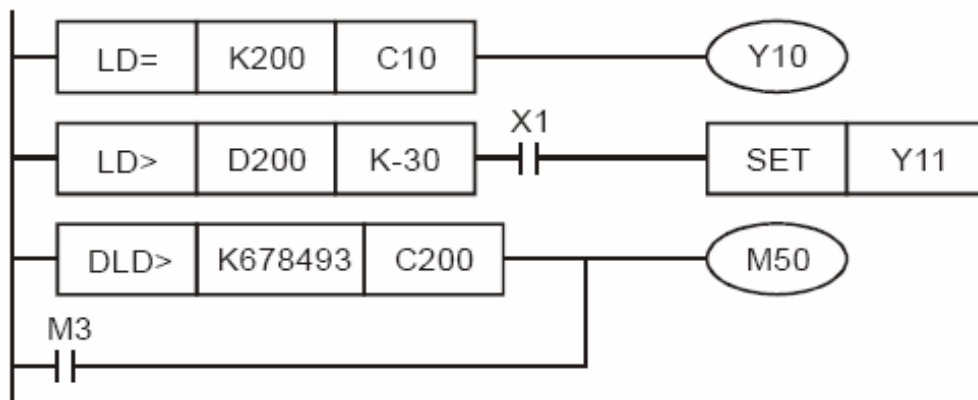
- Если результат сравнения истинен, включается LD-контакт.
- Если результат сравнения ложен, LD-контакт не включается.
- Таблица отражает соответствие операций сравнения с соответствующим номером инструкции.


API	16 –бит команда	32 - бит команда	Контакт замкнут, если:	Контакт разомкнут, если:
224	LD =	DLD =	$(S1) = (S2)$	$(S1) \neq (S2)$
225	LD >	DLD >	$(S1) > (S2)$	$(S1) < (S2)$
226	LD <	DLD <	$(S1) < (S2)$	$(S1) > (S2)$
228	LD <>	DLD <>	$(S1) \neq (S2)$	$(S1) = (S2)$
229	LD ≤	DLD ≤	$(S1) \leq (S2)$	$(S1) \geq (S2)$
230	LD ≥	DLD ≥	$(S1) \geq (S2)$	$(S1) \leq (S2)$

- Когда самый старший бит, MSB (16-бит команда: b15, 32-бит команда: b31), для S_1 и $S_2 = 1$, это будет означать в операции сравнения, что число отрицательное.
- Если в качестве операнда S_1 или S_2 используется счетный регистр C200-..., используйте 32-х битную команду DLD, иначе это вызовет программную ошибку и светодиод “ERROR” будет мигать.

Пример программирования:

1. Если значение счетчика C10 = K200, Y10=BKЛ.
2. Когда значение регистра D200 больше K-30, и X1=BKЛ, Y11=BKЛ с самоудержанием.
3. Если значение счетчика C200 меньше K678493 или когда M3=BKЛ, M50=BKЛ.



API		AND*		Операции сравнения контактного типа. Последовательное соединение	DVP-		
					ES/EX/SS	SA/SX	EH
232~ 238	D				+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (5 шагов) AND* - Непрерывное выполнение.
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
Примечания: Символ *: это =, >, <, <>, ≤, ≥																32-х битная инструкция (9 шагов) DAND* - Непрерывное выполнение.
																Флаги: нет

Функция

Сравнение значений операндов **S1** и **S2**, и включение AND-контакта в зависимости от результата операции.

Описание

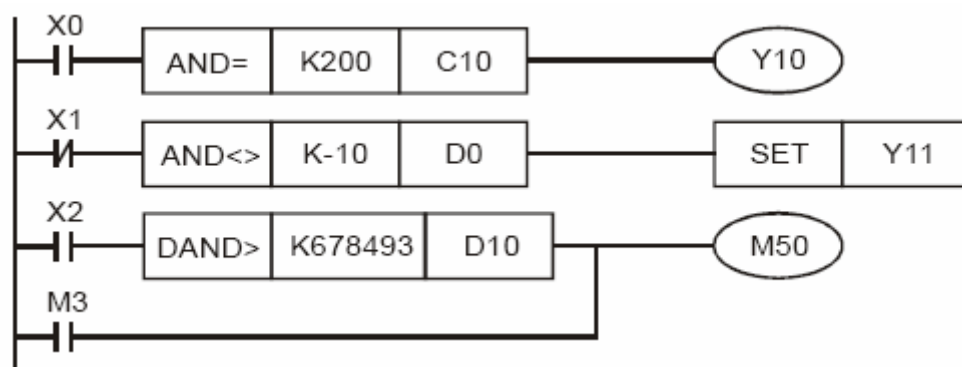
- Инструкции AND* в программе располагаются после LD-инструкций и образуют с ними логическую связь "И".
- Если результат сравнения истинен, включается AND-контакт.
- Если результат сравнения ложен, AND-контакт не включается.
- Таблица отражает соответствие операций сравнения с соответствующим номером инструкции.

API	16 –бит команда	32 - бит команда	Контакт замкнут, если:	Контакт разомкнут, если:
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)

- Когда самый старший бит, MSB (16-бит команда: b15, 32-бит команда: b31), для **S1** и **S2** = 1, это будет означать в операции сравнения, что число отрицательное.
- Если в качестве операнда **S1** или **S2** используется счетный регистр C200-..., используйте 32-х битную команду DAND, иначе это вызовет программную ошибку и светодиод "ERROR" будет мигать.

Пример программирования:

1. Если X0=BKЛ и значение счетчика C10 = K200, Y10=BKЛ.
2. Если X1=ВЫКЛ и значение регистра D0 не равно K-10, Y11=BKЛ с самоудержанием.
3. Если X2=BKЛ и значение регистров D10 (D11) меньше K678493, или M3=BKЛ, M50=BKЛ.



API				Операции сравнения контактного типа. Параллельное соединение.	DVP-		
240~ 246	D	OR*	S₁ S₂		ES/EX/SS	SA/SX	EH
					+	+	+

Оп- ранд	Биты				Слова											16-ти битная инструкция (5 шагов) OR* - Непрерывное выполнение. 32-х битная инструкция (9 шагов) DOR* - Непрерывное выполнение. Флаги: нет
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S ₁					*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	
Примечания: Символ *: это =, >, <, <>, ≤, ≥																

Функция

Сравнение значений операндов **S₁** и **S₂**, и включение OR-контакта в зависимости от результата операции.

Описание

- Инструкция OR* в программе располагается параллельно LD-инструкции и образует с ними логическую связь "ИЛИ".
- Если результат сравнения истинен, включается OR-контакт.
- Если результат сравнения ложен, OR-контакт не включается.
- Таблица отражает соответствие операций сравнения с соответствующим номером инструкции.

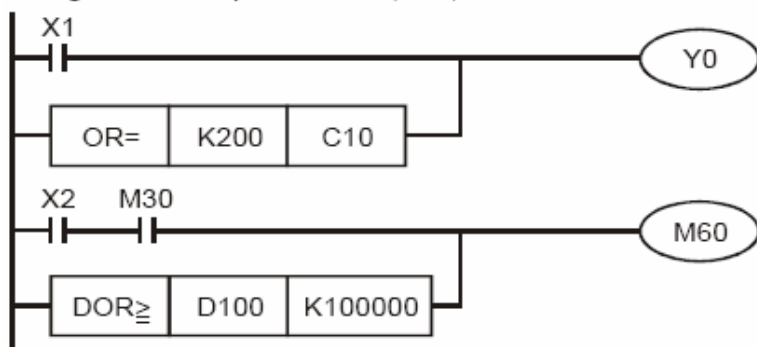
API	16 –бит команда	32 - бит команда	Контакт замкнут, если:	Контакт разомкнут, если:
240	OR =	DOR =	(S1) = (S2)	(S1) ≠ (S2)
241	OR >	DOR >	(S1) > (S2)	(S1) < (S2)
242	OR <	DOR <	(S1) < (S2)	(S1) > (S2)
244	OR <>	DOR <>	(S1) ≠ (S2)	(S1) = (S2)
245	OR ≤	DOR ≤	(S1) ≤ (S2)	(S1) ≥ (S2)
246	OR ≥	DOR ≥	(S1) ≥ (S2)	(S1) ≤ (S2)

- Когда самый старший бит, MSB (16-бит команда: b15, 32-бит команда: b31), для **S₁** и **S₂** = 1, это будет означать в операции сравнения, что число отрицательное.

- Если в качестве операнда **S1** или **S2** используется счетный регистр C200-..., используйте 32-х битную команду DOR, иначе это вызовет программную ошибку и светодиод “ERROR” будет мигать.

Пример программирования:

1. Если X1=ON, или значение счетчика C10 = K200, Y0=ВКЛ.
2. Если X2 и M30 = “ВКЛ”, или значение регистров D100 (D101) больше или равно K1000000, M60=ВКЛ.



11. ПАРАМЕТРЫ КОММУНИКАЦИИ

11.1. Встроенные коммуникационные порты

Контроллеры DVP имеют два встроенных последовательных независимых коммуникационных порта (COM1 и COM2) и дополнительный COM3 (в EH серии). Порты могут использоваться для соединения ПЛК с персональным компьютером, операторской панелью и другими периферийными или сетевыми устройствами. Спецификация коммуникационных портов дана в таблице.

Тип ПЛК		ES/EX/SS		SA/SX		EH	
COM1	Тип порта	RS-232		RS-232		RS-232	
	Режим работы	Ведомый (Slave)		Ведомый (Slave)		Ведомый (Slave)	
	Протокол	Modbus ASCII		Modbus ASCII/RTU		Modbus ASCII/RTU	
	Скорость (бит/с)	9600		9600 – 115200		9600 – 115200	
	Подкл. устройства	ПК, панель оператора		ПК, панель оператора		ПК, панель оператора	
COM2	Тип порта	RS-485		RS-485		RS-485 (может быть изменен на RS-232, RS-422)	
	Режим работы	Ведущий	Ведомый	Ведущий	Ведомый	Ведущий	Ведомый
	Протокол	Modbus или формат определенный польз. в RS инструкциях	Как в COM1	Modbus или формат определенный польз. в RS инструкциях	Как в COM1	Modbus или формат определенный польз. в RS инструкциях	Как в COM1
	Скорость (бит/с)	9600/19200/38400	9600	9600 – 115200		9600 – 115200	
	Подкл. устройства	ПЛК, ПЧ и др.	Как в COM1	ПЛК, ПЧ и др.	Как в COM1	ПЛК, ПЧ и др.	Как в COM1
COM3	Тип порта	-		-		RS-422, RS-232	
	Режим работы					Ведомый/ Modbus	
	Скорость (бит/с)					9600/19200/ 38400	
	Подкл. устройства					ПК, панель оператора	

11.2. Описание протоколов коммуникации

1. Коммуникационный интерфейс: RS-232C. Коммуникационный протокол: режим ASCII, 9600 бит/сек, протокол <7, E, 1 >
2. Коммуникационный блок данных

STX	Стартовый символ ‘:’ (3AH)
ADR 1	Коммуникационный адрес: 8-bit адрес, состоящий из 2 ASCII кодов
ADR 0	
CMD 1	Код команды: 8-bit адрес, состоящий из 2 ASCII кодов
CMD 0	
DATA (0)	Содержание данных: n x 8-bit данных, состоящих из 2-х ASCII кодов. n<=37, максимум 74 ASCII кодов
DATA (1)	
.....	
DATA (n-1)	
LRC CHK 1	LRC контрольная сумма:
LRC CHK 0	8-bit контрольная сумма, состоящая из 2 ASCII кодов

END 1	Конечный символ:
END 0	END1= CR (ODH), ENDO= LF(OAH)

■ ADR (Коммуникационный адрес)

Допустимый коммуникационный адрес должен быть выбран из диапазона 0 ... 31.

Коммуникационный адрес равный 0 – средство трансляции всем ПЛК одновременно, в этом случае, ПЛК не будут отвечать ни на какое сообщение ведущему устройству.

Для примера, связь ПЛК с адресом 16 (десятичное):

(ADR 1, ADR 0)='1', '0' ⇒ '1'=31H, '0'=30H

■ CMD (код команды) и DATA (символы данных)

Формат символов данных зависит от командных кодов. Для примера, чтение непрерывных 8 слов с начального адреса 0614H от ПЛК с адресом 01H.

Формат символов данных зависит от командных кодов. Для примера, чтение непрерывных 8 слов с начального адреса 0614H от ПЛК с адресом 01H.	(Hex)
Командное сообщение	
Стартовый символ	3A
Адрес ПЛК	01
Код команды	03
Начальный адрес данных (старший байт)	06
Начальный адрес данных (младший байт)	14
Число данных в словах (старший байт)	00
Число данных в словах (младший байт)	08
LRC контрольная сумма	DA

Максимальное число данных:

= 18 (для 16 bit регистров)

= 9 (для 32 bit регистров)

Пример чтения данных регистров T20~T27 от ПЛК с адресом 01

PC→PLC

“: 01 03 06 14 00 08 DA CR LF”

PLC→PC

“: 01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 B8 CR LF”

Ответное сообщение	(Hex)
Адрес ПЛК	01
Код команды	03
Число данных в байтах	10
Данные (T20 старший байт)	00
Данные (T20 младший байт)	01
Данные (T21 старший байт)	00
Данные (T21 младший байт)	02
Данные (T22 старший байт)	00
Данные (T22 младший байт)	03
Данные (T23 старший байт)	00
Данные (T23 младший байт)	04

Данные (T24 старший байт)	00
Данные (T24 младший байт)	05
Данные (T25 старший байт)	00
Данные (T25 младший байт)	06
Данные (T26 старший байт)	00
Данные (T26 младший байт)	07
Данные (T27 старший байт)	00
Данные (T27 младший байт)	08
LRC контрольная сумма	C8

■ CHK (проверка суммы)

LRC (продольная проверка избыточности) рассчитана в итоге, модуль 256, значение байтов от ADR1 до последнего символа данных, тогда вычисление шестнадцатеричного представления 2-ух дополнений отрицание суммы. Для примера, читая 1 слово с адреса 0401H ПЛК с адресом 01H.

STX	‘.’
ADR 1	‘0’
ADR 0	‘1’
CMD 1	‘0’
CMD 0	‘3’
Начальный адрес данных	‘0’
	‘4’
	‘0’
	‘1’
Число данных в словах	‘0’
	‘0’
	‘0’
	‘1’
LRC CHK 1	‘F’
LRC CHK 0	‘6’
END 1	CR
END 0	LF

$$01H+03H+04H+01H+00+01H = 0AH$$

2-ух дополнений отрицание 0AH есть **F6H**

Исключительная ситуация по ответу:

Ниже приводятся ситуации, в которых ПЛК не дает нормального ответа управляющему устройству, например, компьютеру.

Если ПЛК не принимает сообщения из-за ошибки связи и не отвечает компьютеру, то компьютер исчерпает лимит времени ожидания.

ПЛК принимает сообщение без ошибки, но не может его обработать, ответ исключения возвратится ведущему устройству.

В ответе исключения, старший значащий бит первоначального кода команды установлен в 1, и код исключения объясняет условие, которое вызвало исключение.

Пример ответа исключения с кодом команды 01H и кодом исключения 02H:

Командное сообщение:

Стартовый символ	3A
Адрес ПЛК	01
Код команды	01
Начальный адрес данных (старший байт)	04
Начальный адрес данных (младший байт)	00

Число данных в словах (старший байт)	00
Число данных в словах (младший байт)	10
LRC контрольная сумма	EA

Ответное сообщение:

Стартовый символ	3A
Адрес ПЛК	01
Код команды	81
Код исключения	02
LRC контрольная сумма	7C

Значение кода исключения:

Коды ошибки	Описание
01	Код запрещенной команды: Код команды, полученный в командном сообщении, не доступный для понимания ПЛК.
02	Недоступный адрес данных: Адрес данных, полученный в командном сообщении, не доступный для понимания ПЛК.
03	Не допустимое значение данных: Значение данных, полученное в командном сообщении, не доступное для понимания ПЛК.
07	Ошибка контрольной суммы Ошибочно сформирован блок данных Длина блока данных выходит из допустимого диапазона.

- ◆ Формат символьных данных зависит от кода команды. Возможные коды команд описаны в следующей таблице.

Код	Имя	Обозначение
01	Чтение состояния выходов	S, Y, M, T, C
02	Чтение состояния входов	S, X, Y, M, T, C
03	Чтение значений регистров	T, C, D
05	Установка состояния одного реле	S, Y, M, T, C
06	Установка значения одного регистра	T, C, D
15	Установка состояния нескольких реле	S, Y, M, T, C
16	Установка значения нескольких регистров	T, C, D
17	Чтение информации о модели и состоянии ПЛК	None

Адреса внутренних устройств в ПЛК серии DVP-ES

Устройство	Диапазон	Рабочий диапазон	Адрес
S	000~255	000~127	0000~00FF
S	246~511		0100~01FF
S	512~767		0200~02FF
S	768~1023		0300~03FF
X	000~377 (Octal)	000~177 (Octal)	0400~04FF
Y	000~377 (Octal)	000~177 (Octal)	0500~05FF
T	000~255	000~127	0600~06FF
M	000~255	0000~1279	0800~08FF
M	256~511		0900~09FF
M	512~767		0A00~0AFF
M	768~1023		0B00~0BFF
M	1024~1279		0C00~0CFF
C	000~255	000~127 232~255	0E00~0EFF
D	000~255	000~599 1000~1143	1000~10FF
D	256~511		1100~11FF
D	512~767		1200~12FF
D	768~1023		1300~13FF
D	1024~1279		1400~14FF

- **Код команды: 01**, чтение состояния выходных битовых устройств (катушек реле).

Командное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	01
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	14
Число точек (старший байт)	00
Число точек (младший байт)	25
LRC контрольная сумма	BF

Максимальное число точек = 255 = 0x00FF

Пример чтения состояния катушек реле T20~T56 от ПЛК с адресом 01

PC→PLC “: 01 01 06 14 00 25 BF CR LF”

PLC→PC “: 01 01 05 CD 6B B2 0E 1B D6 CR LF”

Ответное сообщение	(Hex)
Адрес ПЛК	01
Код команды	01
Число данных в байтах	05
Данные (T27...T20)	CD
Данные (T35...T38)	6B
Данные (T43...T36)	B2
Данные (T51...T44)	0E
Данные (T56...T52)	1B
LRC контрольная сумма	E6

- **Код команды: 02**, чтение состояния входных битовых устройств (контактов реле).

Командное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	02
Начальный адрес (старший байт)	05
Начальный адрес (младший байт)	14
Число точек (старший байт)	00
Число точек (младший байт)	25
LRC контрольная сумма	BF

Пример чтения состояния контактов реле Y024~Y070 от ПЛК с адресом 01

PC→PLC “: 01 02 05 14 00 25 BF CR LF”

PLC→PC “: 01 01 05 CD 6B B2 0E 1B E5 CR LF”

Ответное сообщение	(Hex)
Адрес ПЛК	01
Код команды	02
Число данных в байтах	05

Данные (Y033...Y024)	CD
Данные (Y043...Y034)	6B
Данные (Y053...Y044)	B2
Данные (Y063...Y054)	0E
Данные (Y070...Y064)	1B
LRC контрольная сумма	E5

■ **Код команды: 03**, чтение значений регистров T, C, D

Командное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	03
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	14
Число точек (старший байт)	00
Число точек (младший байт)	08
LRC контрольная сумма	DA

Максимальное число точек

= 18 (для 16 bit регистров)

= 9 (для 32 bit регистров)

Пример чтения состояния регистров T20~T27 от ПЛК с адресом 01

PC→PLC

“: 01 03 06 14 00 08 DA CR LF”

PLC→PC

“: 01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 B8 CR LF”

Ответное сообщение	(Hex)
Адрес ПЛК	01
Код команды	03
Число данных в байтах	10
Данные (T20 старший байт)	00
Данные (T20 младший байт)	01
Данные (T21 старший байт)	00
Данные (T21 младший байт)	02
Данные (T22 старший байт)	00
Данные (T22 младший байт)	03
Данные (T23 старший байт)	00
Данные (T23 младший байт)	04
Данные (T24 старший байт)	00
Данные (T24 младший байт)	05
Данные (T25 старший байт)	00
Данные (T25 младший байт)	06
Данные (T26 старший байт)	00
Данные (T26 младший байт)	07
Данные (T27 старший байт)	00
Данные (T27 младший байт)	08
LRC контрольная сумма	C8

■ **Код команды: 05**, установка состояния одного реле

Командное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	05
Адрес реле (старший байт)	05
Адрес реле (младший байт)	00
Установка состояния (старший байт)	FF
Установка состояния (младший байт)	00
LRC контрольная сумма	F6

MMNN = 0xFF00....реле включить (ON)

MMNN = 0x0000.... реле выключить (OFF)

Пример включения реле Y000 (установить состояние ON)

PC→PLC “: 01 05 05 00 FF 00 F6 CR LF”

PLC→PC “: 01 05 05 00 FF 00 F6 CR LF”

Ответное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	05
Адрес реле (старший байт)	05
Адрес реле (младший байт)	00
Установка состояния (старший байт)	FF
Установка состояния (младший байт)	00
LRC контрольная сумма	F6

■ **Код команды: 06**, установка значения одного регистра

Командное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	06
Адрес регистра (старший байт)	06
Адрес регистра (младший байт)	00
Значение регистра (старший байт)	12
Значение регистра (младший байт)	34
LRC контрольная сумма	AD

Пример записи значения 00 03 в регистр T0

PC→PLC “: 01 06 06 00 12 34 AD CR LF”

PLC→PC “: 01 06 06 00 12 34 AD CR LF”

Ответное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	06
Адрес регистра (старший байт)	06
Адрес регистра (младший байт)	00
Значение регистра (старший байт)	12

Значение регистра (младший байт)	34
LRC контрольная сумма	AD

T0: Q → : 01 06 06 00 12 34 AD CR LF

C0: Q → : 01 06 0E 00 12 34 AF CR LF

C232: Q → : 01 06 0E E8 12 34 56 78 EF CR LF

D10: Q → : 01 06 10 0A 12 34 99 CR LF

D1000: Q → : 01 06 13 E8 12 34 BA CR LF

■ **Код команды: 15, установка состояния нескольких реле**

Командное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	0F
Начальный адрес (старший байт)	05
Начальный адрес (младший байт)	00
Число точек (старший байт)	00
Число точек (младший байт)	0A
Число данных в байтах	02
Установка состояния (старший байт)	CD
Установка состояния (младший байт)	01
LRC контрольная сумма	11

Максимальное число точек = 255

Пример установки состояния реле: Y007...Y000 = 1100 1101, Y011...Y010 = 01.

PC→PLC “: 01 0F 05 00 00 0A 02 CD 01 11 CR LF”

PLC→PC “: 01 0F 05 00 00 0A E1 CR LF”

Ответное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	0F
Начальный адрес (старший байт)	05
Начальный адрес (младший байт)	00
Установка состояния (старший байт)	00
Установка состояния (младший байт)	0A
LRC контрольная сумма	E1

■ **Код команды: 16**, установка значения нескольких регистров

Командное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	10
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	00
Число регистров (старший байт)	00
Число регистров (младший байт)	02
Число данных в байтах	04
Данные (старший байт)	00
Данные (младший байт)	0A
Данные (старший байт)	01
Данные (младший байт)	02
LRC контрольная сумма	C6

Максимальное число регистров

= 16 (для 16 bit регистров)

= 8 (для 32 bit регистров)

Пример записи значения 00 0A в регистр T0, 01 02 в T1.

PC→PLC “: 01 10 06 00 02 00 04 00 0A 01 02 D6 CR LF”

PLC→PC “: 01 10 06 00 00 02 E7 CR LF”

Ответное сообщение	(Hex)
Стартовый символ	3A
Адрес ПЛК	01
Код команды	10
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	00
Число регистров (старший байт)	00
Число регистров (младший байт)	02
LRC контрольная сумма	E7

■ **Код команды: 17**, информация о модели и состоянии ПЛК

Командное сообщение:

Стартовый символ	3A
Адрес ПЛК	01
Код команды	11
LRC контрольная сумма	EE

Ответное сообщение:

Стартовый символ	3A
Адрес ПЛК	01
Код команды	11
Число данных в байтах	04
Идентификационный номер ПЛК	01
Состояние индикатора "RUN" 00 = OFF FF = ON	FF
Данные 0 (D1001 старший байт)	40
Данные 1 (D1001 младший байт)	10
LRC контрольная сумма	9A

■ Адресный список внутренних устройств контроллеров DVP

Устройство	Диапазон		Тип	Адрес	Рабочий диапазон		
					ES/EX/SS	SA/SX/SH	EH
S	000~255		bit	0000~00FF	0~127	0~1023	0~1023
S	246~511		bit	0100~01FF			
S	512~767		bit	0200~02FF			
S	768~1023		bit	0300~03FF			
X	000~377 (Octal)		bit	0400~04FF	0~177	0~177	000~377
Y	000~377 (Octal)		bit	0500~05FF			
T	000~255		bit/word	0600~06FF	0~127	000~255	000~255
M	000~255		bit	0800~08FF	0~1279	0~4095	0000~4095
M	256~511		bit	0900~09FF			
M	512~767		bit	0A00~0AFF			
M	768~1023		bit	0B00~0BFF			
M	1024~1279		bit	0C00~0CFF			
M	1280~1535		bit	0D00~0DFF			
M	1536~1791		bit	B000~B0FF			
M	1792~2047		bit	B100~B1FF			
M	2048~2303		bit	B200~B2FF			
M	2304~2559		bit	B300~B3FF			
M	2560~2815		bit	B400~B4FF			
M	2816~3071		bit	B500~B5FF			
M	3072~3327		bit	B600~B6FF			
M	3328~3583		bit	B700~B7FF			
M	3584~3839		bit	B800~B8FF			
M	3840~4095		bit	B900~B9FF			
C	0~199	16-bit	bit/word	0E00~0EC7	0~127	0~199	0~199
	200~255	32-bit	bit/Dword	0EC8~0EFF	232~255	200~255	200~255

Устройство	Диапазон	Тип	Адрес	Рабочий диапазон		
				ES/EX/SS	SA/SX/SH	EH
D	000~256	word	1000~10FF	0~1311	0~4999	0000~9999
D	256~511	word	1100~11FF			
D	512~767	word	1200~12FF			
D	768~1023	word	1300~13FF			
D	1024~1279	word	1400~14FF			
D	1280~1535	word	1500~15FF			
D	1536~1791	word	1600~16FF			
D	1792~2047	word	1700~17FF			
D	2048~2303	word	1800~18FF			
D	2304~2559	word	1900~19FF			
D	2560~2815	word	1A00~1AFF			
D	2816~3071	word	1B00~1BFF			
D	3072~3327	word	1C00~1CFF			
D	3328~3583	word	1D00~1DFF			
D	3584~3839	word	1E00~1EFF			
D	3840~4095	word	1F00~1FFF			
D	4096~4351	word	9000~90FF			
D	4352~4607	word	9100~91FF			
D	4608~4863	word	9200~92FF			
D	4864~5119	word	9300~93FF			
D	5120~5375	word	9400~94FF			
D	5376~5631	word	9500~95FF			
D	5632~5887	word	9600~96FF			
D	5888~6143	word	9700~97FF			
D	6144~6399	word	9800~98FF			
D	6400~6655	word	9900~99FF			
D	6656~6911	word	9A00~9AFF			
D	6912~7167	word	9B00~9BFF			
D	7168~7423	word	9C00~9CFF			
D	7424~7679	word	9D00~9DFF			
D	7680~7935	word	9E00~9EFF			
D	7936~8191	word	9F00~9FFF			
D	8192~8447	word	A000~A0FF			
D	8448~8703	word	A100~A1FF			
D	8704~8959	word	A200~A2FF			
D	8960~9215	word	A300~A3FF			
D	9216~9471	word	A400~A4FF			
D	9472~9727	word	A500~A5FF			
D	9728~9983	word	A600~A6FF			
D	9984~9999	word	A700~A70F			



ASIA

Delta Electronics, Inc.

Taoyuan1

31-1, Xingbang Road, Guishan Industrial Zone,
Taoyuan County 33370, Taiwan, R.O.C.

TEL: 886-3-362-6301 / FAX: 886-3-362-7267

Delta Electronics (Jiang Su) Ltd.

Wujiang Plant3

1688 Jiangxing East Road,
Wujiang Economy Development Zone,
Wujiang City, Jiang Su Province,
People's Republic of China (Post code: 215200)

TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

EUROPE

Deltronics (The Netherlands) B.V.

Eindhoven Office

De Witbogt 15, 5652 AG Eindhoven, The Netherlands

TEL: 31-40-2592850 / FAX: 31-40-2592851