

ПЛК

Основы программирования

Данный курс предназначен для участников, которые будут создавать программы для программируемых контроллеров впервые.

В этом курсе приводится информация по программированию ПЛК серии MELSEC. Одним из основных языков программирования ПЛК является язык релейно-контактных схем или, иначе говоря, релейной логики (Ladder Diagram, сокращенно - LD). В этом курсе рассматриваются основные аспекты программирования на языке релейно-контактных схем, включая его основные команды.

Некоторые разделы данного курса основаны на базовых курсах по программируемым контроллерам MELSEC. Рекомендуется до начала прохождения данного курса пройти соответствующий базовый курс.

Данный курс содержит указанную ниже информацию.

Глава 1. Программирование ПЛК

В этой главе разъясняются основные аспекты программирования на языке релейной логики.

Глава 2. Команды, содержащие битовые операнды

В этой главе приводятся разъяснения в отношении команд, содержащих битовые операнды (ВКЛ./ВЫКЛ.).

Глава 3. Команды, содержащие словные операнды





В этой главе приводятся разъяснения в отношении команд, содержащих словные (числовые) операнды.

Глава 4. Команды ветвления программы

В этой главе приводятся разъяснения в отношении команд ветвления программ.

Заключительный тест

Проходной балл: требуется 60% или выше.

Переход к следующей странице		Переход к следующей странице.
Возврат к предыдущей странице		Возврат к предыдущей странице.
Переход к требуемой странице		Появится экран «Содержание», на котором вы сможете перейти к требуемой странице.
Завершение обучения		Завершение обучения.

Правила техники безопасности

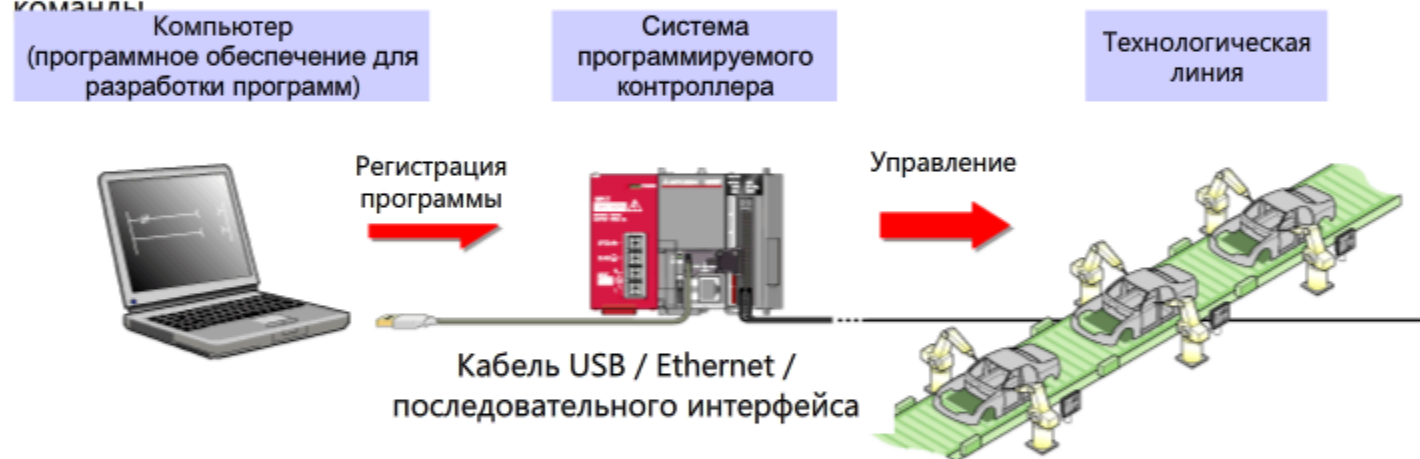
Если вы обучаетесь с использованием реальных изделий, внимательно изучите правила техники безопасности, приведенные в соответствующих руководствах.

Глава 1 Управляющая программа

Операции, выполняемые программируемым контроллером, записываются в виде управляющих программ. Эти программы регистрируются в модуле ЦП, осуществляющем управление различными входными и выходными сигналами.

В число языков программирования, используемых для программируемых контроллеров, входят релейно-контактная схема (LD), список инструкций (IL), язык функциональных блоков (FBD), структурированный текст (ST) и последовательная функциональная схема (SFC).

В этом курсе разъясняются основные аспекты программирования на языке релейной логики, включая его основные команды.



В этом курсе для создания программ используется техническое программное обеспечение для программируемых контроллеров GX Works2 или GX Works3.

Для получения необходимых знаний по применению программного обеспечения для программируемого контроллера вам следует пройти курс «GX Works2 Basics» (Основные сведения о GX Works2) или «Engineering Software MELSOFT GX Works3 (Ladder)» (Инженерное программное обеспечение MELSOFT GX Works3 (программы на языке релейной логики)).

GX Works2 поддерживает ПЛК устройства серий MELSEC-Q/L/F.

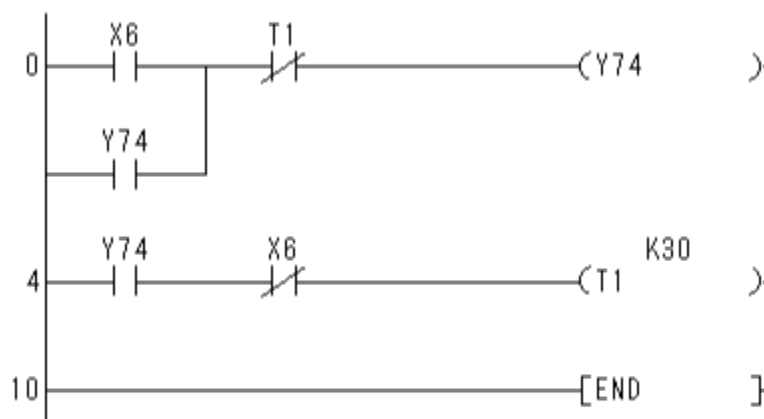
GX Works3 поддерживает ПЛК устройства серий MELSEC iQ-R/iQ-F.

1.1

Языки программирования

Программа на языке релейной логики - это графическая логическая схема, основанная на логике электрической цепи.

В программах на языке релейной логики символы, представляющие команды, соединяются линиями, подобно тому, как это делается на принципиальной электрической схеме, и это делает такие программы легкими для понимания. Кроме того, программирование на языке релейной логики не требует специальных знаний в области программирования (например, как в случае с языками C и BASIC), и его могут легко освоить те, кто имеет опыт работы с электрическими цепями и реле.



В приведенной ниже таблице показана та же программа на языке IL.

IL требует некоторого знания программирования для выражения операций в форме команд.

Номер шага	Команда	Операнд
0	LD	X6
1	OR	Y74
2	ANI	T1
3	OUT	Y74
4	LD	Y74
5	ANI	X6
6	OUT	T1 K30
10	END	

Программы для программируемых контроллеров могут обрабатывать данные двух типов.

Бит	Бит определяется двумя состояниями - ON (ВКЛ.) и OFF (ВЫКЛ.). Они также могут быть выражены как «1» (ВКЛ.) и «0» (ВЫКЛ.). Битовые значения часто используются для отображения состояний устройств ввода/вывода, например переключателей и ламп.
Слово	Числа и символы. Словные значения часто используются для отображения количества и времени. * В этом курсе будут приводиться разъяснения только в отношении чисел. Для ознакомления с дополнительной информацией о символах, используемых в качестве словных значений, см. соответствующее руководство.

Для отображения значений используются следующие числовые форматы.

- Десятичное представление
- Двоичное представление
- Шестнадцатеричное представление
- Восьмеричное представление

1.2.1**Десятичное представление**

В десятичном представлении величина (количество) числа представляется с использованием 10 цифр от «0 до 9».

В программируемых контроллерах MELSEC перед десятичными числами ставится буква «K».
Например, «K153» представляет собой десятичное число «153».

1.2.2

Двоичное представление

В то время как десятичное представление обычно используется для выражения количеств и времени, программируемые контроллеры и персональные компьютеры используют двоичные данные, которые представляют собой комбинации цифр «0» и «1».

В нижеприведенной таблице показано соответствие между десятичными и двоичными значениями вплоть до десятичного числа «8».

Десятичное представление	Двоичное представление
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
⋮	⋮

Когда в программе используется какая-либо команда из 1 слова, она будет сохраняться и обрабатываться реальным программируемым контроллером как 16-битовые двоичные данные. Эти 16-битовые двоичные данные являются синонимом «1 слова».

Например, десятичное число «157» выражается в двоичном формате как «000000010011101».

В десятичном представлении биты записываются справа налево. (Стартовым битом является крайний бит справа.)

b15	...	b8	b7	b6	b5	b4	b3	b2	b1	b0	← Позиция бита
0	...	0	1	0	0	1	1	1	0	1	← Двоичное представление
2^{16}		2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	← Степень числа 2
32768		256	128	64	32	16	8	4	2	1	← Десятичное значение

Для преобразования двоичных значений в десятичные умножьте состояние каждого бита («0» или «1») на соответствующий весовой коэффициент и сложите все произведения.

$$\begin{aligned}
 &= \underline{1} \times 128 + 0 \times 64 + 0 \times 32 + \underline{1} \times 16 + 1 \times 8 + \underline{1} \times 4 + 0 \times 2 + \underline{1} \times 1 \\
 &= 128 + 16 + 8 + 4 + 1 \\
 &= 157
 \end{aligned}$$

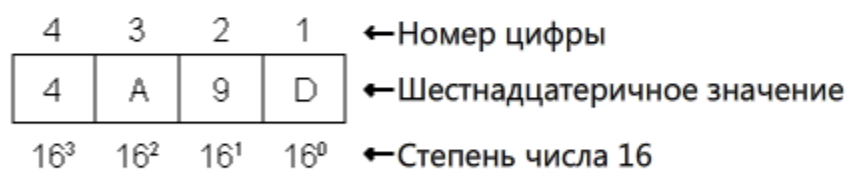
1.2.3 Шестнадцатеричное представление

В шестнадцатеричном представлении величина числа представляется с использованием шестнадцатеричной системы счисления или указывается с использованием 16 буквенно-цифровых символов: 0–9 и A–F. Цифры в шестнадцатеричном представлении возрастают в следующем порядке: 0, 1...9, A...F. Когда значение превышает основание системы счисления «F», происходит переход на следующий шестнадцатеричный разряд. В программируемых контроллерах MELSEC перед шестнадцатеричными числами ставится буква «H».

Например, «H4A9D» представляет собой шестнадцатеричное число «4A9D».

Двоичное представление могло быть длинным, и его могло бы быть трудно использовать в программах и на дисплеях мониторов. В таком случае полезным является шестнадцатеричное представление. Одна цифра шестнадцатеричного значения может выражать 4 бита (4 цифры) двоичных значений.

На приведенной ниже иллюстрации показано, как шестнадцатеричное значение выражается в виде десятичного значения.



$$= 4 \times 16^3 + A \times 16^2 + 9 \times 16^1 + D \times 16^0$$

$$= 4(4096) + 10(256) + 9(16) + 13(1)$$

$$= 19101$$

* Одна цифра шестнадцатеричного значения может выражать 4 бита двоичного значения.

Десятичное представление	Двоичное представление	Шестнадцатеричное представление
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

16	10000	10
17	10001	11
18	10010	12
⋮	⋮	⋮
⋮	⋮	⋮
19101	0100 1010 1001 1101	4 A 9 D

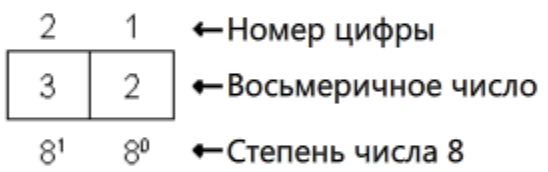
1.2.4 Восьмеричное представление

В восьмеричном представлении величина числа представляется с использованием 8 цифр («0–7»).

Когда значение возрастает с «0», «1», «2» до «7», получается значение «10».

Данное восьмеричное представление используется для номеров ввода/вывода ПЛК устройств серии MELSEC iQ-F/F.

На приведенной ниже иллюстрации показано, как восьмеричное значение выражается в виде десятичного значения.



$$= 3 \times 8^1 + 2 \times 8^0$$

$$= 3 \times (8) + 2 \times (1)$$

$$= 26$$

Десятичное представление	Двоичное представление	Шестнадцатеричное представление	Восьмеричное представление
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
<hr style="border-top: 1px dashed black;"/>			
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
16	1101	D	15
14	1110	E	16
15	1111	F	17
<hr style="border-top: 1px dashed black;"/>			
16	1000	10	20
17	10001	11	21
18	10010	12	22
.	.	.	.
.	.	.	.
.	.	.	.
26	011 010	1A	32

* Одна цифра восьмеричного значения может выражать 3 бита двоичного значения.

1.3

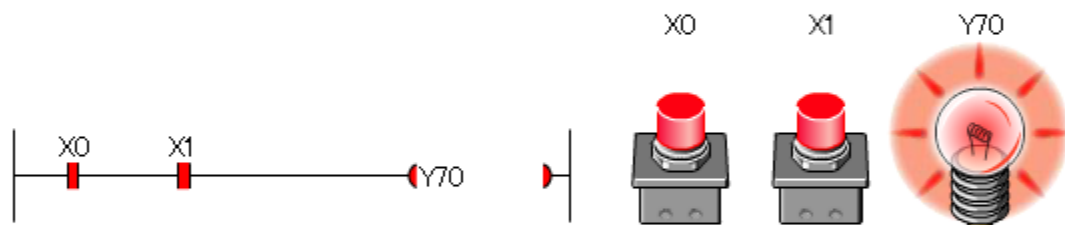
ПЛК - основы программирования

При последовательном управлении серия операций выполняется на основании сигналов ВКЛ./ВЫКЛ., получаемых от оборудования, которое подключено к модулю входов, а затем результаты выполнения операций выводятся на оборудование, подключенное к модулю выходов.

Для осуществления такого управления управляющая программа должна иметь входные условия и выходы, которые будут выполняться при удовлетворении входных условий.

Приведенная ниже программа содержит команды для выполнения следующих операций:

- Когда оба кнопочных выключателя, подключенных к клеммам X1 и X2, включены, требуется включить клемму Y70
- Результат выполнения операции выводится на клемму Y70 для включения подключенной лампы



При одновременном нажатии переключателей X0 и X1 включается лампа Y70.

В программах, для идентификации сигналов ввода/вывода используются буквенно-цифровые символы, например X0, X1 и Y70. Эти буквенно-цифровые символы называются номерами ввода/вывода.

В этой главе приводятся разъяснения в отношении номеров ввода/вывода и операндов, которые требуются для создания управляющих программ.

Для выражения адресов операндов в устройствах серий MELSEC iQ-R/Q/L/iQ-F и серии MELSEC-F используются разные форматы.

Сводная информация о различиях приводится в нижеследующей таблице.

Серия MELSEC	Бит			Слово
	X (номер ввода)	Y (номер вывода)	M (внутренний маркер)	D (регистр данных)
Серия iQ-R/Q/L	Шестнадцатеричное представление	Шестнадцатеричное представление	Десятичное представление	Десятичное представление
Серия iQ-F/F	Восьмеричное представление	Восьмеричное представление	Десятичное представление	Десятичное представление

1.4.1 Номера ввода/вывода и сигналы ввода/вывода (серии MELSEC iQ-R/Q/L)

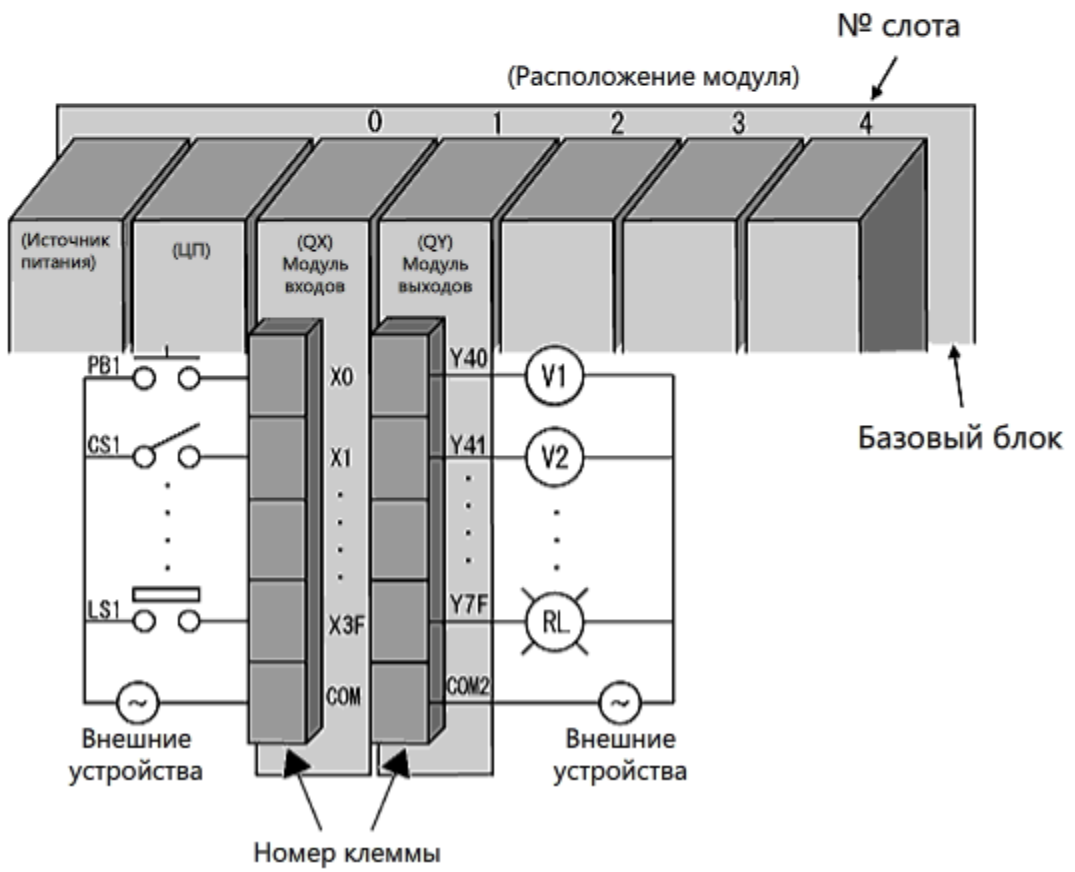
Серия MELSEC iQ-R/Q/L

Номер ввода/вывода состоит из буквы, которая обозначает вход (X) или выход (Y), и шестнадцатеричного значения, которое обозначает номер клеммы.

Сначала определяется номер ввода/вывода на основании положения установки модуля на базовом блоке.

Затем определяется диапазон номеров ввода/вывода на основании количества используемых точек ввода/вывода модуля.

На приведенной ниже иллюстрации показано, как назначаются номера ввода/вывода для 64-точечного модуля входов и 64-точечного модуля выходов, которые установлены в слотах № 0 и № 1 соответственно.



1.4.2 Номера ввода/вывода и сигналы ввода/вывода (серия MELSEC iQ-F/F)

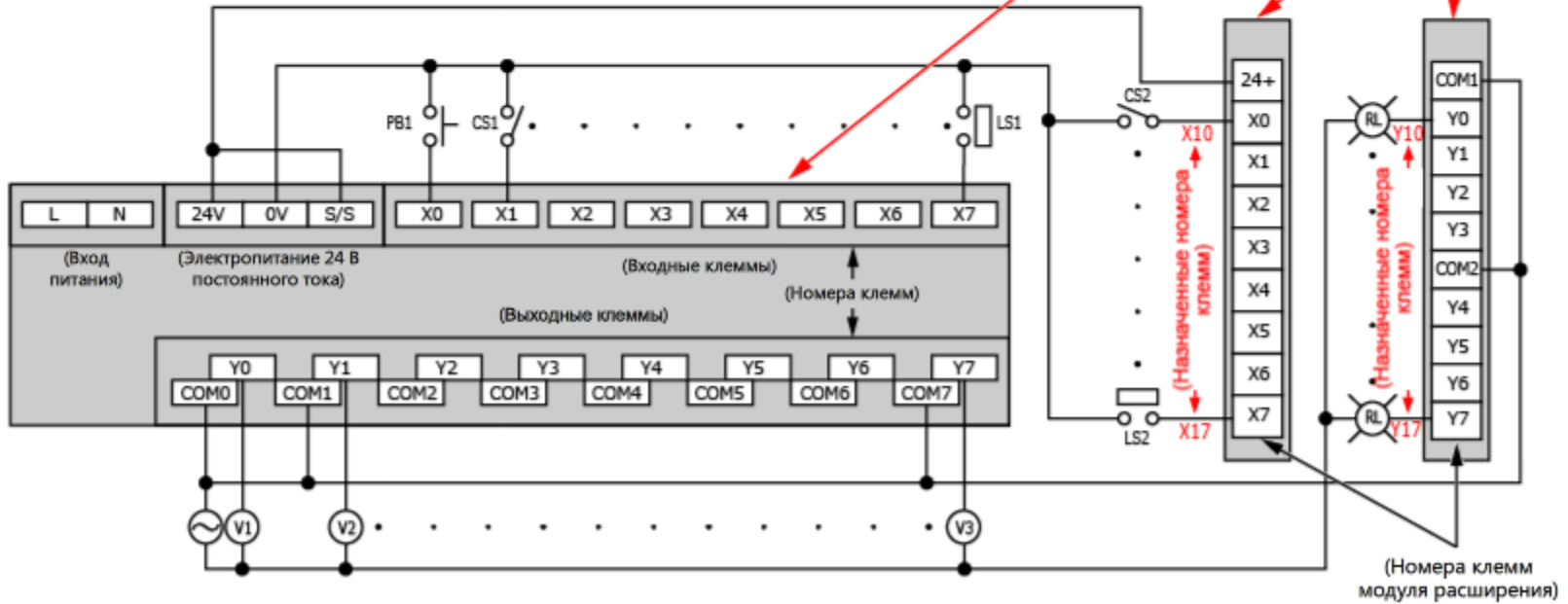
Серии MELSEC iQ-F/F

Номер ввода/вывода состоит из буквы, которая обозначает вход (X) или выход (Y), и восьмеричного значения, которое обозначает номер клеммы.

Сначала определяется номер ввода/вывода с учетом номеров, занятых входами/выходами основного блока или предыдущего модуля расширения ввода/вывода. Затем определяется диапазон номеров ввода/вывода на основании количества используемых точек ввода/вывода модуля.

Номер ввода/вывода для нового модуля расширения всегда начинается с цифры «0» следующего 8-ричного разряда. Например, если последний номер ввода/вывода предшествующего модуля - X7, то первый номер ввода/вывода следующего модуля - X10.

На приведенной ниже иллюстрации показано, как назначаются номера ввода/вывода для 8-точечного модуля расширения ввода и 8-точечного модуля расширения вывода, которые добавлены к модулю ЦП серии MELSEC-F.

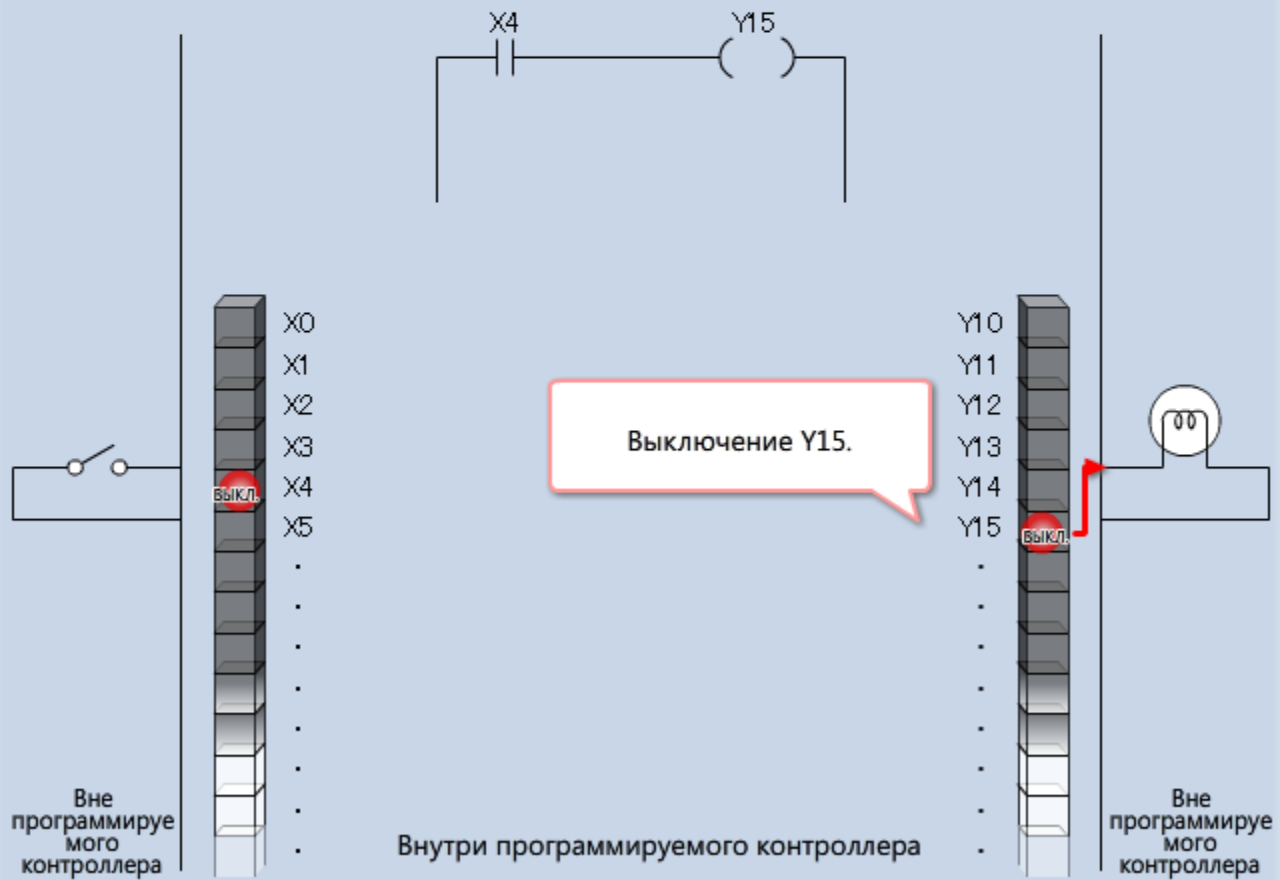


1.4.3 Номера и операнды ввода/вывода

Состояния сигналов оборудования, подключенного ко входам контроллера, сохраняются в его области памяти, называемой «devices» (операнды). Точно так же в отношении вывода выходное оборудование работает в соответствии с состояниями операндов.

Операнды, в которых хранится битовая информация (ВКЛ./Выкл.), например вход (X) и выход (Y), называются «битовыми операндами».

Адреса операндов соответствуют номерам ввода/вывода. Например, состояние клеммы, которой назначен номер ввода/вывода X0, сохраняется в операнде X0. Точно так же состояние операнда Y10 соответствует клемме, которой назначен номер ввода/вывода Y10.



1.4.4

Внутренние маркеры

Мы узнали, что битовые операнды, например X (вход) и Y (выход), соответствуют номерам, назначенным физическим клеммам ввода/вывода модуля.

Существует другая группа битовых операндов, которая не имеет отношения к «физическим» входам/выходам и клеммам ввода/вывода модулей контроллера. Такие операнды называются «внутренними маркерами (M)».

Внутренние маркеры (M) нумеруются в десятичном формате, хотя операнды ввода (X) и вывода (Y) нумеруются в шестнадцатеричном формате для ПЛК серий MELSEC iQ-R/Q/LQ/L и в восьмеричном формате для ПЛК серии MELSEC iQ-F/F.

Внутренние маркеры (M), главным образом, используются для хранения временных битовых данных.

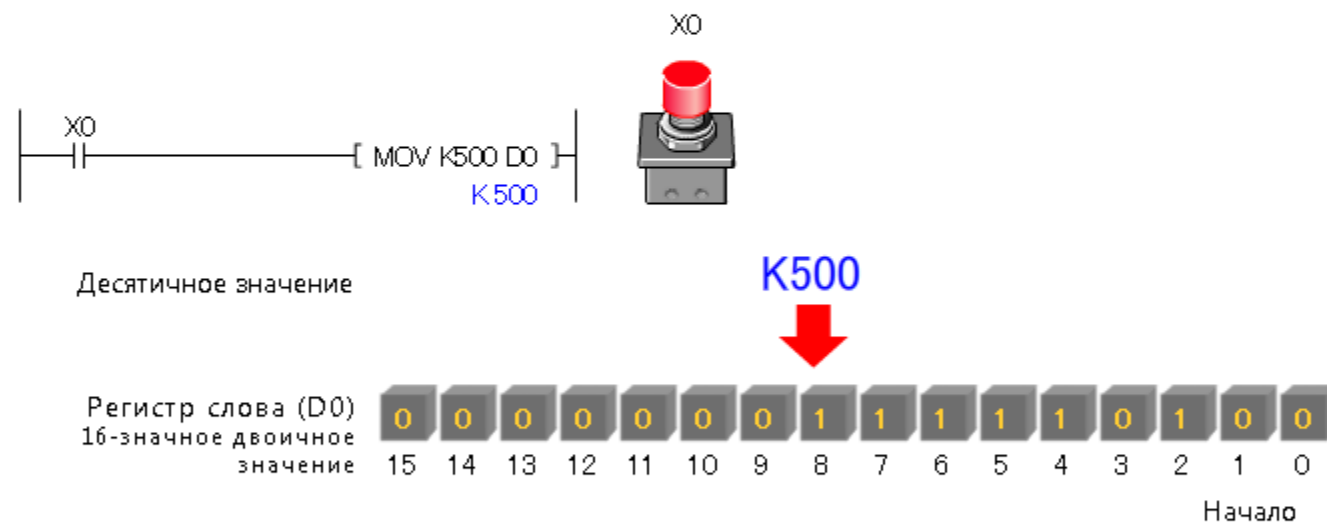
Например, внутренние маркеры (M) могут использоваться для хранения вычисленного результата операции в программе, чтобы его можно было использовать в другом месте программы

1.4.5 Словные операнды

Мы узнали, что операнды, в которых хранится битовая информация (ВКЛ./ВЫКЛ.), называются «битовыми операндами», а операнды, в которых хранится словная информация, называются «словными операндами».

Одними из часто используемых словных операндов являются «регистры данных» (D). В одном регистре данных (D) может храниться 1 слово (16 битов) данных.

На приведенной ниже анимации показано, как использовать регистры данных (D). В программе при включении X0 в D0 сохраняется «K500» (десятичное значение). Команда MOV копирует некоторое значение в заданный операнд. (Дополнительная информация будет приведена в главе 3.1.)



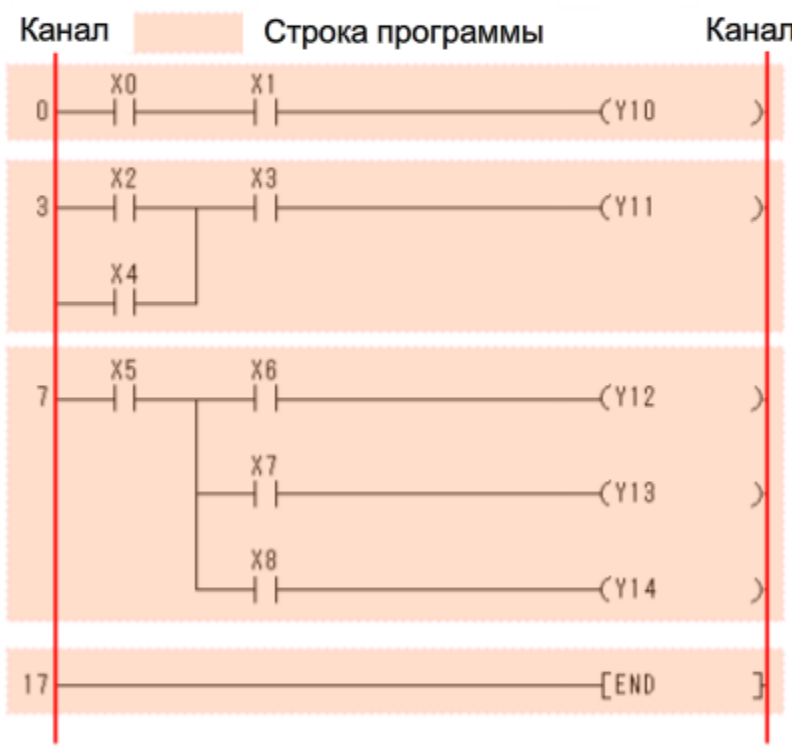
Даже при отпускании кнопки и выключении X0 значение «K500», сохраненное в регистре данных D0, удерживается.

1.5 Создание управляющих программ

Управляющие программы на языке LD состоят из вертикальных каналов с левого и правого краев, а также символов команд, соединенных между собой линиями

Каждая такая последовательность команд и линий между ними от левого канала до правого называется строкой программы

Несколько строк образуют управляющую программу, которая всегда заканчивается командой `-[END]-` или `-[FEND]-`.



■ Различия в инженерном программном обеспечении

команды `-()-` и `-[]-` визуально отличаются в инженерном программном обеспечении GX Works2 и GX Works3. В описаниях, используемых в данном курсе, применяются команды GX Works2.

	GX Works2	GX Works3
<code>-()-</code>	<code>{Y10 }</code>	Y10
<code>-[]-</code>	<code>{MOV K500 D0 }</code>	<code>MOV K500 D0</code>

1.5.1

Символы команд

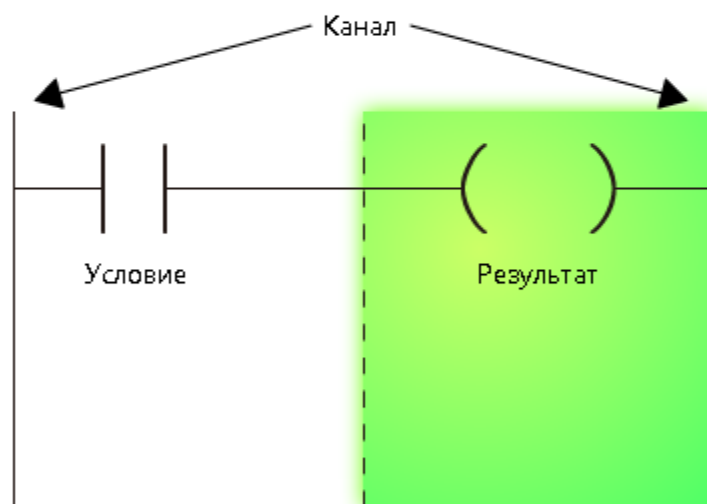
Как объясняется в главе 1.3, программируемый контроллер должен получать команды для выполнения определенных операций при соблюдении предварительно заданных входных условий. В таких командах для выражения входных условий и сведений о выводе используются символы команд.

Символ команды часто содержит адрес операнда.

Адрес операнда указывает на область (операнд), где хранится информация о состоянии, используемая для оценки соблюдения условия или в качестве выходного результата.

Каждая строка программы содержит условия и результаты вывода. Условия размещаются слева, а выходные результаты - справа.

Выходные результаты могут представлять собой простой сигнал ВКЛ./ВЫКЛ. или какую-либо специальную команду, например для выполнения расчета или операции копирования.



В релейно-контактной схеме имеются два параллельных канала.

Слева записываются условия.

Справа записываются результаты.

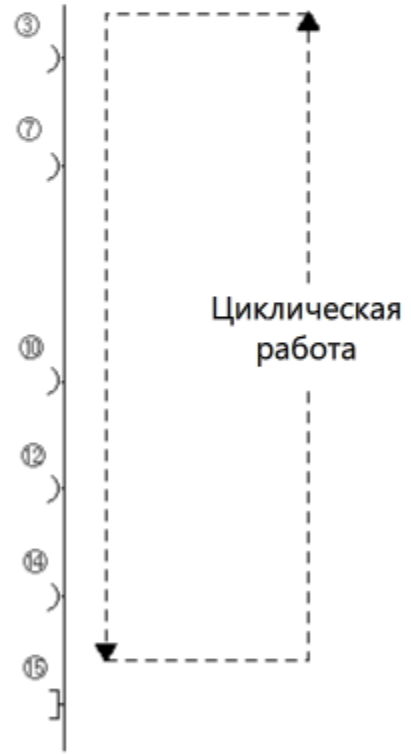
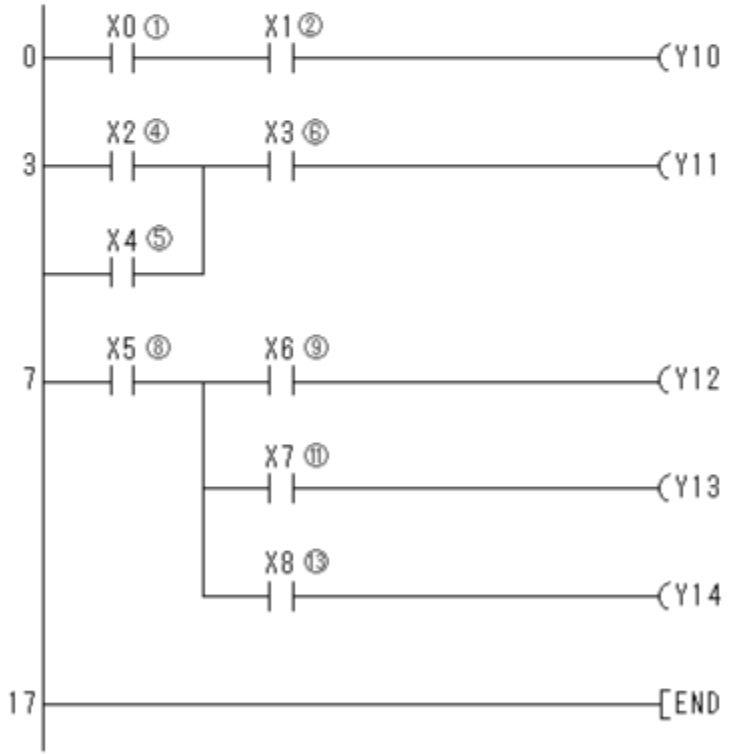
1.6 Процедура выполнения программы

Программа начинается с левой начальной команды первой по порядку строки программы и заканчивается командой -[END]-.

После достижения команды -[END]- выполнение программы начинается снова с начальной команды. Такое повторяющееся выполнение называется «циклическим опросом программы».

Период времени, которое занимает обработка одного цикла, называется «временем цикла».

Процедура выполнения программы показана на приведенной ниже иллюстрации. Команды выполняются слева направо поочередно в каждой строке



1.7 Время обновления

Как объяснялось ранее, время цикла - это период времени, которое занимает однократное выполнение программы. Время цикла также может быть выражено следующим образом:

$$\text{Время цикла} = \text{время обновления} + \text{время выполнения программы} + \text{время обработки команды END}$$

Время обновления - это период времени, которое занимает чтение данных из модуля входов в операнды ввода (X), плюс период времени, которое занимает запись данных в модуль выходов из операндов вывода (Y).

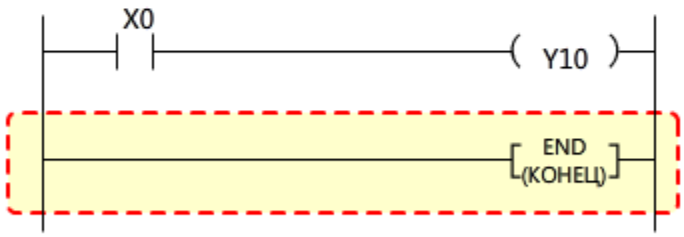


(1) Обновление ввода/вывода

- Передача информации о состоянии ВКЛ./ВЫКЛ. устройств вывода в подключенное оборудование вывода
- Сохранение информации о состоянии ВКЛ./ВЫКЛ., полученной от подключенного оборудования ввода, в устройствах ввода

(2) Выполнение программы

(3) Обработка команды END (КОНЕЦ)



Обработка команды END (КОНЕЦ)
(Подробности здесь опущены.)

Обратите внимание, что информация о состояниях ВКЛ./ВЫКЛ. фактического переключателя считывается сразу и сохраняется в операндах ввода (X), а новые данные записываются поверх существующих значений во время обновления. Точно так же данные в операндах вывода (Y) записываются сразу в модуль выходов при выполнении команды.

Это означает, что если сигнал переключается из состояния ВЫКЛ. в состояние ВКЛ., а затем снова в состояние ВЫКЛ., состояние данного сигнала может не распознаться как ВКЛ. Однако время цикла является очень малым по сравнению с длительностью сигнала. Программируемый контроллер редко может пропустить изменение состояния сигнала.

Глава 2 Команды, содержащие битовые операнды

В этой главе приводятся разъяснения в отношении команд, в которых используются битовые операнды (ВКЛ./ВЫКЛ.).

Операции, в которых используются битовые операнды, являются самыми главными операциями в управляющих программах.

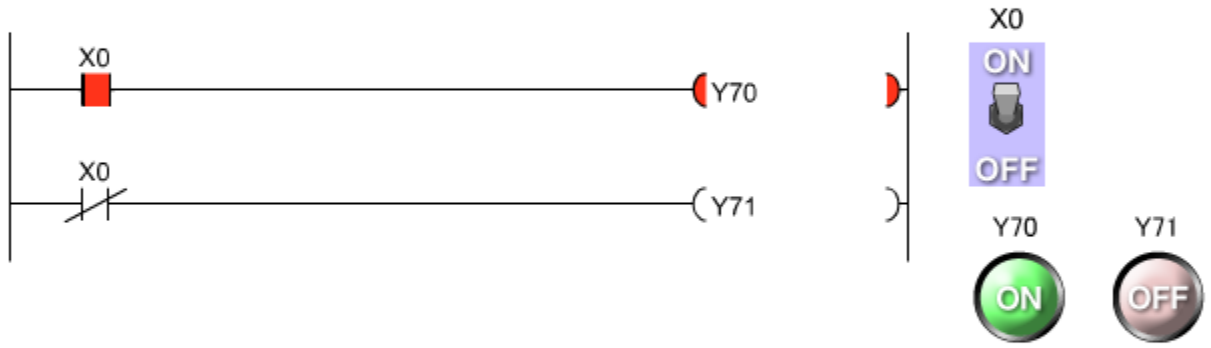
Входные сигналы от оборудования ввода используются в качестве условий для управления оборудованием вывода.

2.1 Входные условия и выходные сигналы

В качестве входных условий используются нормально разомкнутые (NO) и нормально замкнутые (NC) контакты.
 При соблюдении входных условий подается команда для выхода катушки (команда OUT).
 Если входные условия не соблюдаются, команда для выхода катушки не подается.
 Команда для НО/НЗ контакта и команда OUT является основной комбинацией команд, используемой в управляющих программах.

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу команд NO, NC и OUT, щелкая показанный справа входной переключатель.

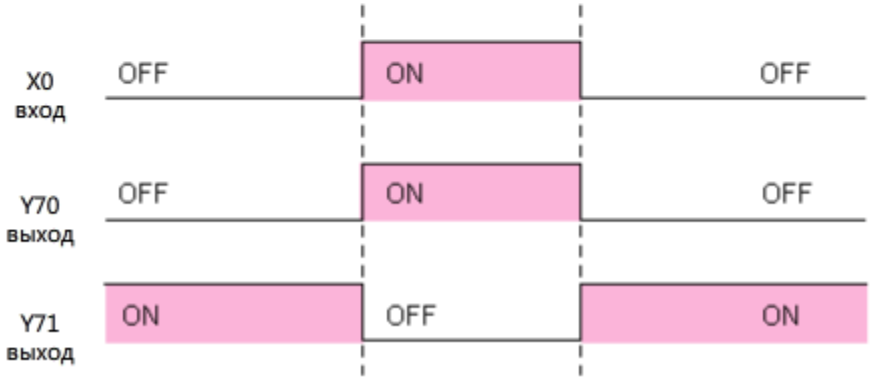


2.1 Входные условия и выходные сигналы

Коды и функции команд

Символ	Функция
	<p>НО контакт Выполняется, когда устройство имеет состояние ВКЛ.</p>
	<p>НЗ контакт Выполняется, когда устройство имеет состояние ВЫКЛ. (в противоположность НО контакту).</p>
	<p>Выход катушки (OUT) Когда предшествующее входное условие соблюдается, выводятся данные, хранящиеся в предустановленном операнде.</p>
	<p>Конечная команда (END) Указывает на конец программы. В любой программе обязательно должна присутствовать команда END.</p>

Временная диаграмма

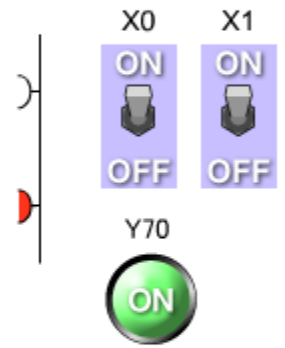
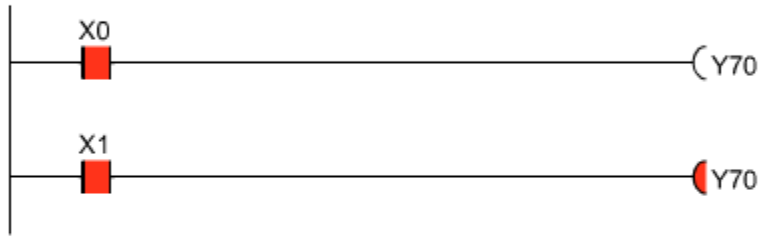


2.1.1 Использование одинакового адреса операнда для команд

В каждой строке программы может использоваться только одна команда OUT с одним адресом операнда. Если используются несколько команд OUT с одним и тем же адресом операнда, только последняя команда OUT становится действительной, делая первую команду OUT недействительной.

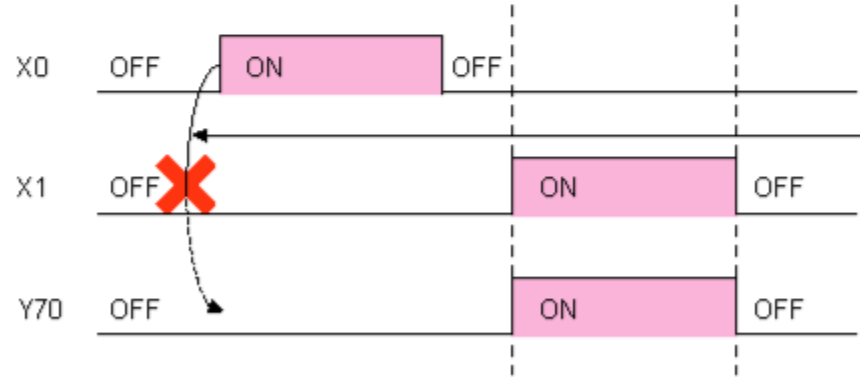
■ Программа на языке релейной логики

Имитируйте работу двух команд с одинаковым адресом операнда, щелкая показанный справа входной переключатель. Такой тип использования (использование OUT Y70 для двух команд) называется «дублированием катушки».



Когда X1 = ВКЛ., Y70 = ВКЛ.

■ Временная диаграмма



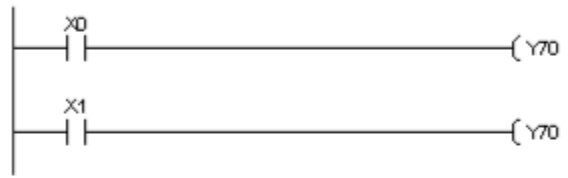
Первое входное условие X0 игнорируется, поскольку последнее входное условие имеет приоритет.

2.1.1 Использование одинакового адреса операнда для команд

■ Пример исправления

В этом примере «X1» имеет более высокий приоритет, а «X0» игнорируется.

После исправления ступеньки лестницы на показанную на рис. В операнд Y70 будет включаться при соблюдении любого из двух входных условий, что позволяет избежать конфликта между двумя командами OUT.



(Рис. А)



(Рис. В)

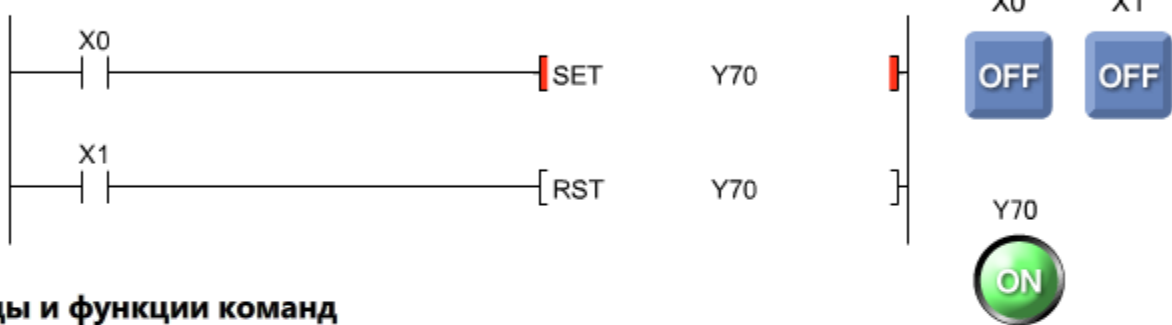
2.2 Удержание/отмена выходов

В отличие от команды OUT, команда сохранения операции (команда SET) удерживает состояние выхода даже после того, как входное условие перестает соблюдаться.

Для отмены выхода (ВЫКЛ.) можно выполнить команду отмены удержания операции (команду RST).

■ Программа на языке релейной логики и выполняемая операция

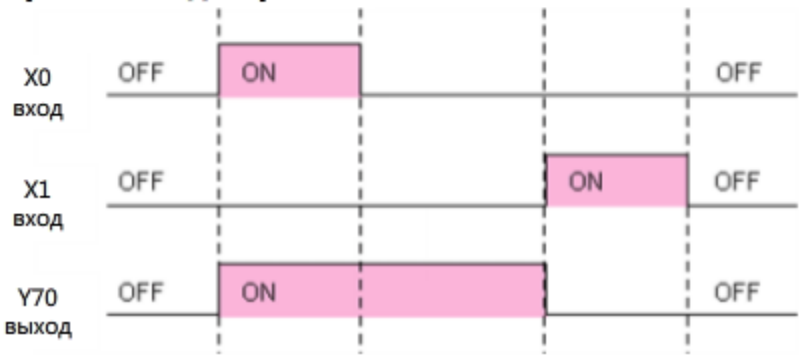
Имитируйте работу команд SET и RST, щелкая показанные справа входные переключатели.



■ Коды и функции команд

Символ	Функция
	Команда удержания операции (SET) Включает устройство и удерживает состояние (выхода) ВКЛ. Выход удерживается, даже если входное условие перестает соблюдаться.
	Команда отмены удержания операции (RST) Отменяет состояние ВКЛ. и отменяет вывод на заданное устройство.

■ Временная диаграмма



2.2.1 Различия между командами OUT и SET

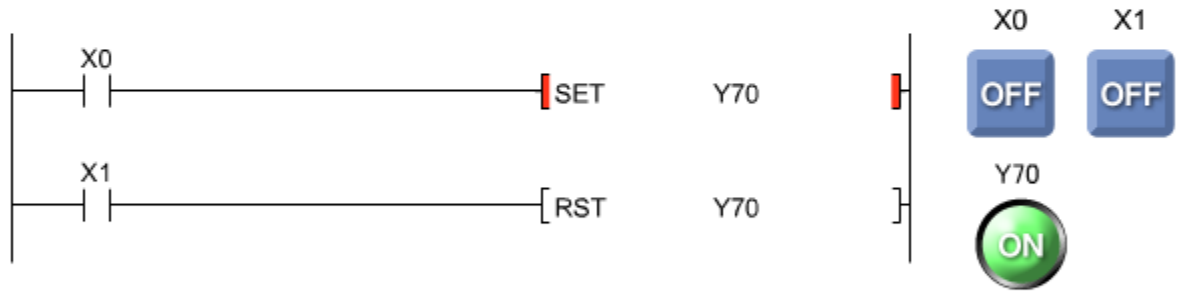
Имитируйте различия в работе команд OUT и SET, щелкая показанные справа входные переключатели.

■ Команда OUT



Y70 включен при соблюдении входного условия.

■ Команды SET/RST



Когда входное условие соблюдается, Y70 включен до тех пор, пока не будет выполнена команда RST.

2.2.2 Замена удержания командой SET

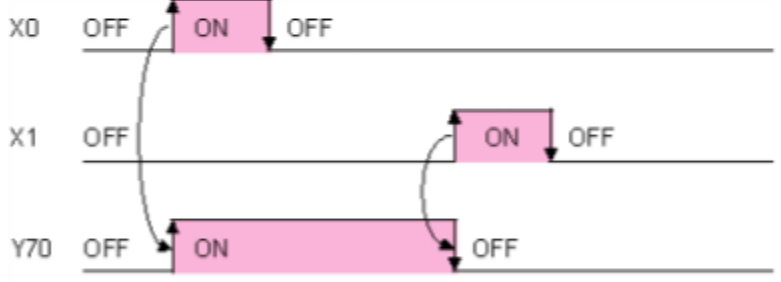
Программа на языке релейной логики и выполняемая операция

Имитируйте работу удержания, щелкая показанные справа входные переключатели.



Когда X0 = ВКЛ. и X1 = ВЫКЛ., Y70 = ВКЛ.
Y70 = ВКЛ. (удерживается) до тех пор, пока X1 не ВКЛ.

■ Временная диаграмма

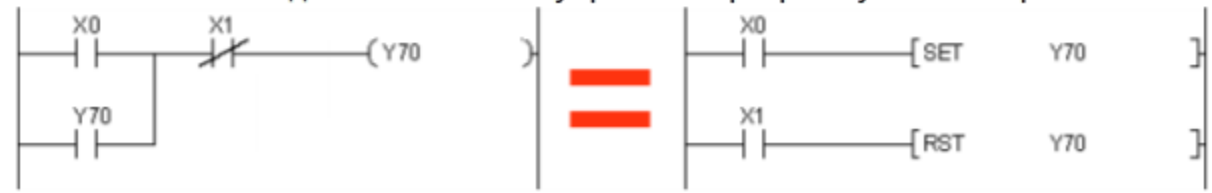


Даже после выключения X10 Y70 (катушка) остается включенной (удерживается)

■ Замена командой SET

Программа на языке релейной логики для удержания может быть переписана в виде программы на языке релейной логики с командой SET.

Использование команды SET позволяет упростить программу на языке релейной логики.



2.3 Добавление условий (логики AND)

Для реализации логики AND НО/НЗ контакты располагаются последовательно
В логике AND условие соблюдается, когда включены несколько НР/НЗ контактов, подсоединенных последовательно.

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу логики AND, щелкая показанные справа входные переключатели.



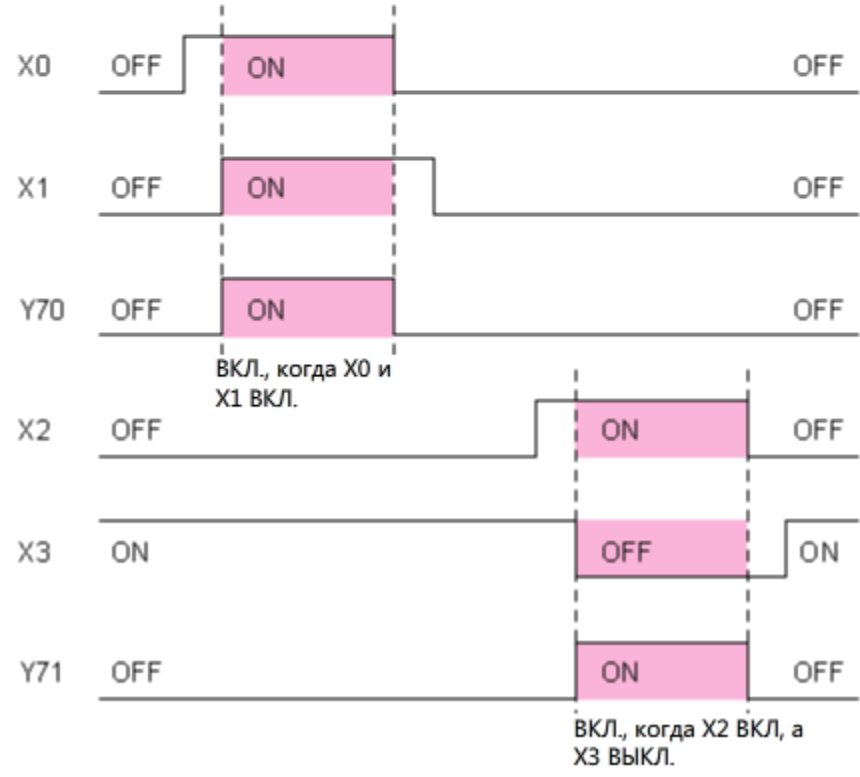
Когда X0 и X1 ВКЛ., Y70 ВКЛ.
Когда X2 ВКЛ. и X3 ВЫКЛ., Y71 ВКЛ.

2.3 Добавление условий (логики AND)

Коды и функции команд

Символ	Функция
	Последовательное подключение НО контакта НО контакт подключен последовательно (горизонтально).
	Последовательное подключение НЗ контакта НЗ контакт подключен последовательно (горизонтально).

Временная диаграмма



2.4 Добавление условий (логики OR)

Для реализации логики OR НО/НЗ контакты располагаются параллельно. В логике OR условие соблюдается, когда включен один из НО/НЗ контактов, подсоединенных параллельно.

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу логики OR, щелкая показанные справа входные переключатели.



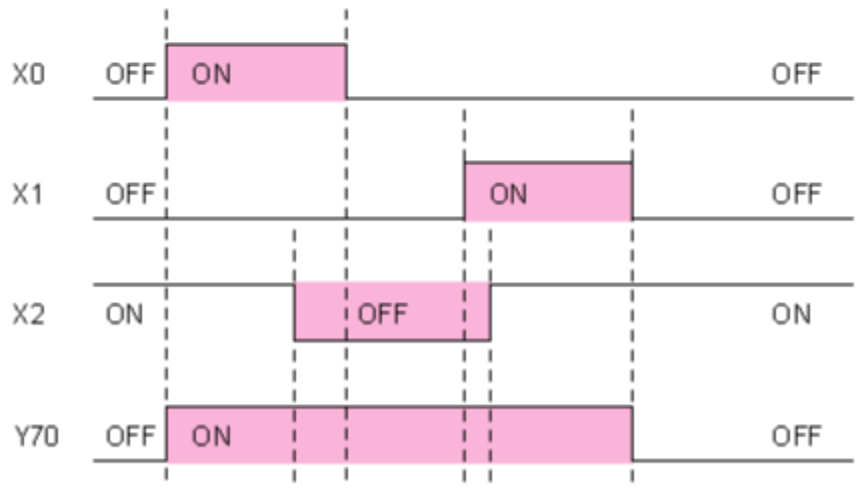
Y70 включен при соблюдении любого из следующих условий: X0 ВКЛ., X1 ВКЛ. или X2 ВЫКЛ.

2.4 Добавление условий (логики OR)

Коды и функции команд

Символ	Функция
	Параллельное подключение НО контактов НО контакт подключен параллельно (вертикально).
	Параллельное подключение НЗ контакта НЗ контакт подключен параллельно (вертикально).

Временная диаграмма



2.5

Вывод в виде импульсов

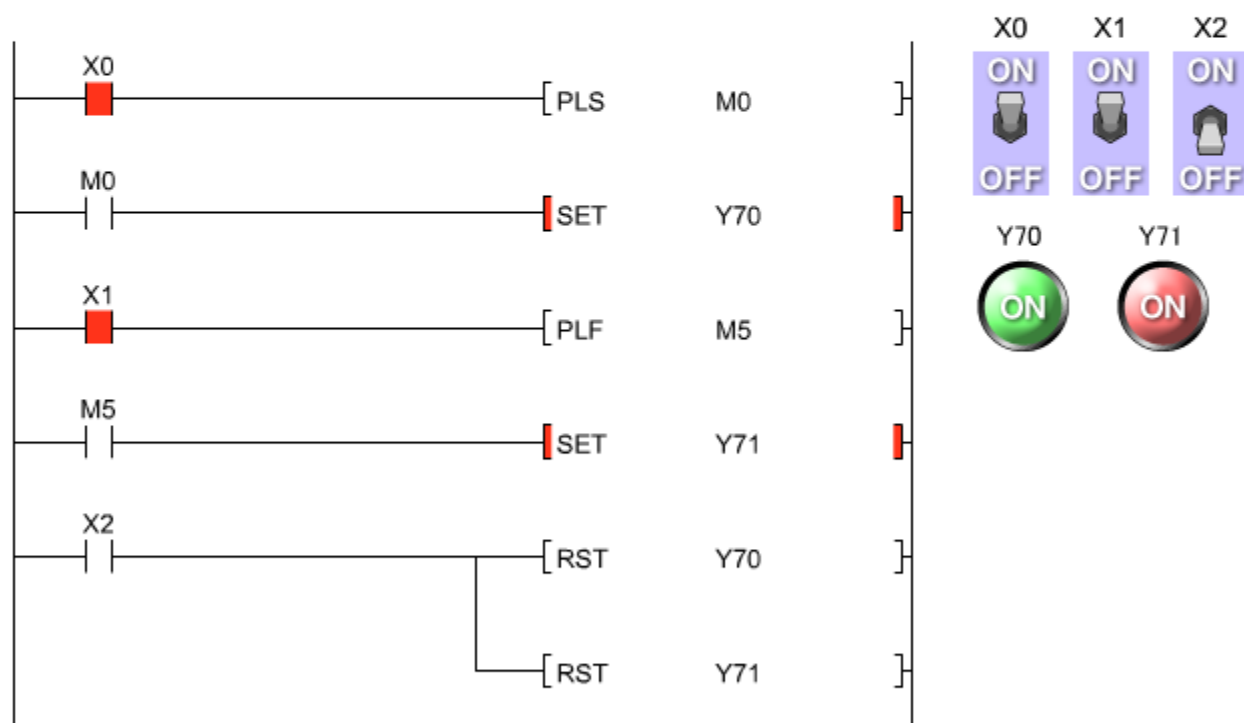
В отличие от команды OUT, команда переднего фронта (команда PLS) включает катушку на один цикл после соблюдения входного условия.

В противоположность команде PLS, команда заднего фронта (команда PLF) включает катушку на один цикл после того, как входное условие перестает соблюдаться.

По истечении одного цикла катушка, включенная командой PLS/PLF, возвращается в состояние ВЫКЛ.

■ Программа на языке релейной логики и выполняемая операция



Имитируйте работу команд PLS и PLF, щелкая показанные справа входные переключатели.



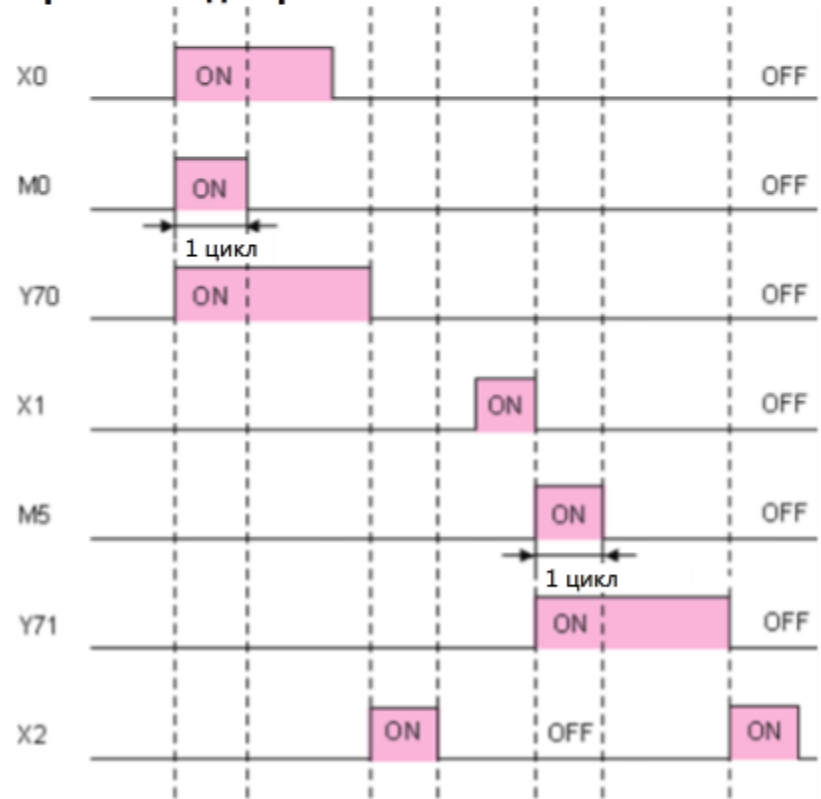
На переднем фронте X0 (при переходе из состояния ВЫКЛ. в состояние ВКЛ.) M0 включается на 1 цикл
 На заднем фронте X1 (при переходе из состояния ВКЛ. в состояние ВЫКЛ.) M5 включается на 1 цикл

2.5 Вывод в виде импульсов

Коды и функции команд

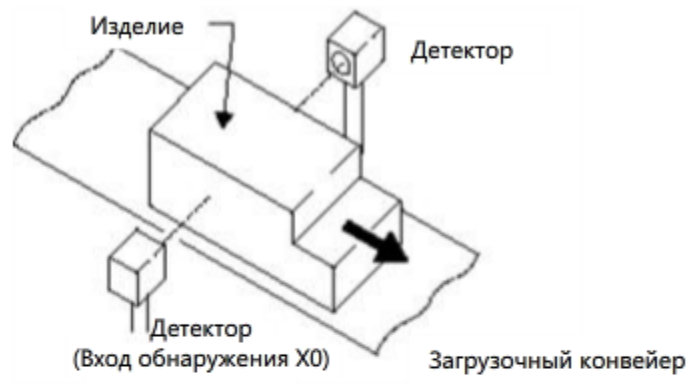
Символ	Функция
	<p>Вывод на переднем фронте (PLS) Данные выводятся на заданное устройство при 1-м цикле после соблюдения входного условия.</p>
	<p>Вывод на заднем фронте (PLF) Данные выводятся на заданное устройство при 1-м цикле после того, как входное условие перестает соблюдаться.</p>

Временная диаграмма



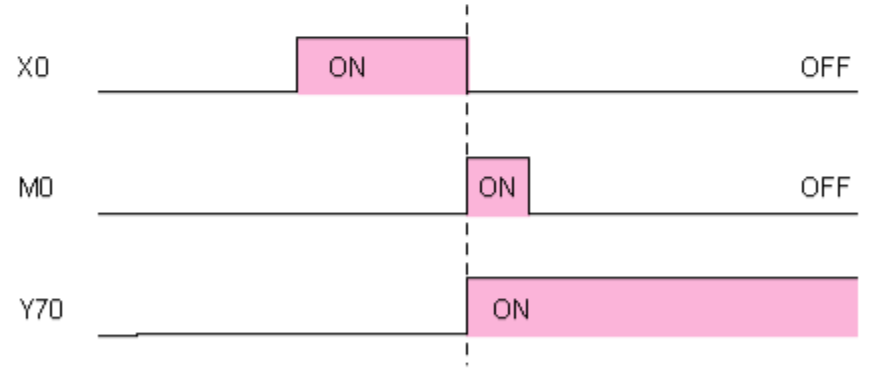
2.5.1 Пример применения импульсных выходов

■ Программа на языке релейной логики



Импульсный выход используется для определения прохождения движущихся объектов. При обнаружении прохождения изделия инициируется следующий процесс.

■ Временная диаграмма



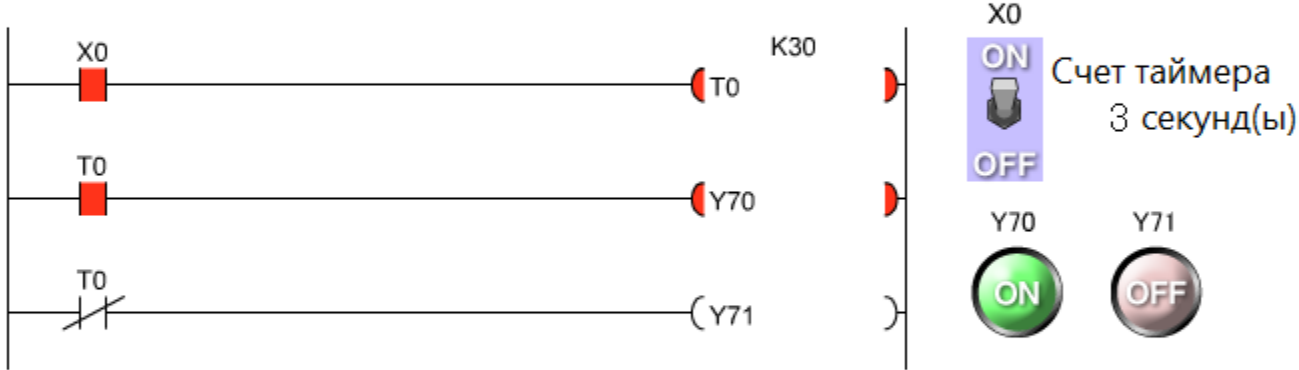
2.6 Измерение времени

Для измерения времени используются команда OUT и операнд таймера (T).
 Измерение времени начинается при соблюдении входного условия (ВКЛ.). Когда продолжительность периода времени достигает заданного значения, включается операнд таймера (T).
 Если входное условие не соблюдается (ВЫКЛ.) или если операнд таймера (T) сбрасывается с помощью команды RST, выполняется инициализация истекшего времени и выхода.

Состояние операнда таймера (T) может использоваться в качестве входного условия в других частях программы.

■ Программа на языке релейной логики и выполняемая операция

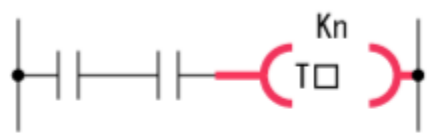
Имитируйте работу таймера, щелкая показанный справа входной переключатель.



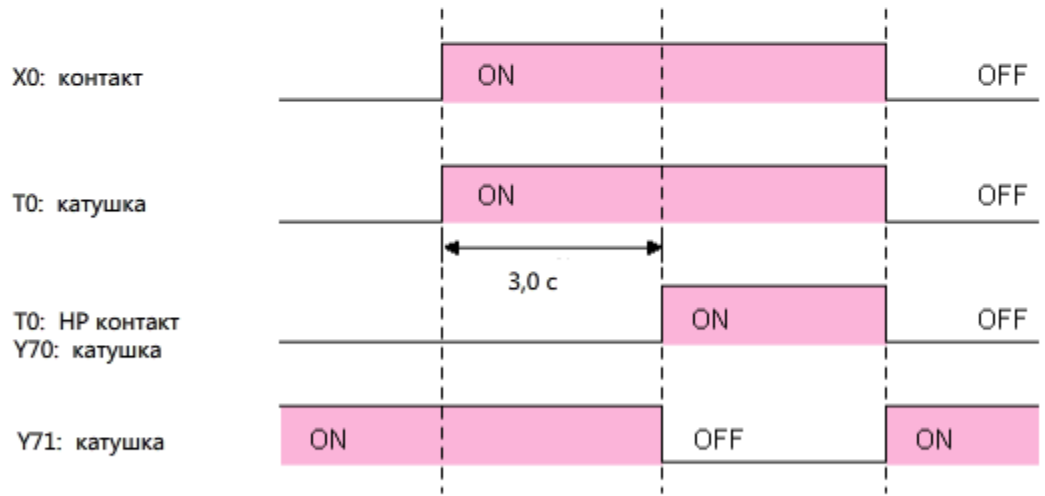
X0 включается, а затем через 3 секунды Y70 включается, а Y71 выключается.

2.6 Измерение времени

Код и функция команды

Символ	Функция
 <p>□ : адрес операнда</p>	<p>Работа таймера</p> <p>Операнд таймера (Т) используется с выходом катушки (OUT) для измерения продолжительности соблюдения условия (пребывания в состоянии ВКЛ.). Тайм-аут возникает по истечении заданного периода времени. Одновременно с тайм-аутом включается таймер (Т0). Установленное значение таймера указывается в виде «Кп» (п: десятичное число). Таймеры часто используются в качестве таймеров задержки включения, которая определяет время с момента соблюдения определенного условия.</p>

Временная диаграмма

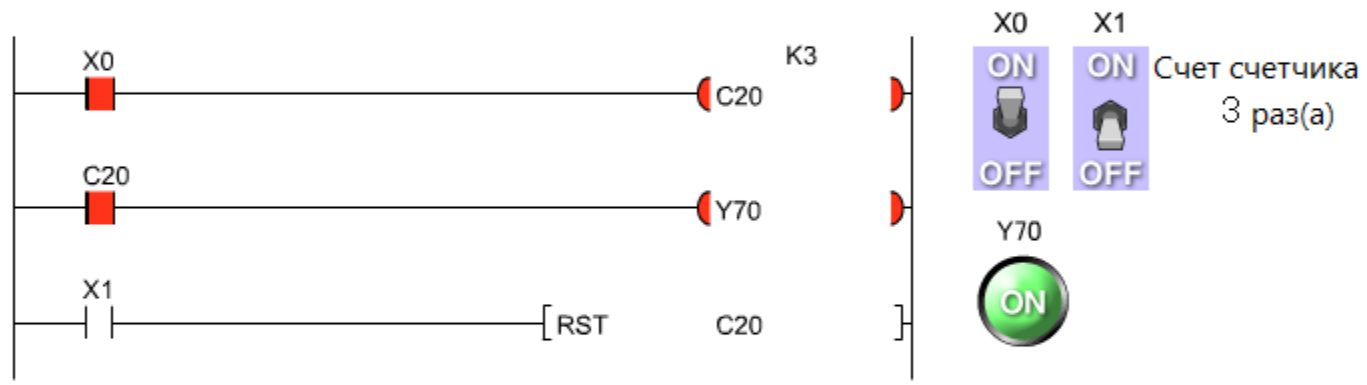


2.7 Подсчет

Для подсчета используются команда OUT и операнд счетчика (C).
 При соблюдении определенного входного условия счет увеличивается, а когда счет достигает заданного значения, включается указанный операнд счетчика (C).
 Если операнд счетчика (C) сбрасывается с помощью команды RST, выполняется инициализация счета и состояния операнда.
 Состояние операнда счетчика (C) может использоваться в качестве входного условия в других частях программы.

■ Программа на языке релейной логики и выполняемая операция


Имитируйте работу счетчика, щелкая показанный справа входной переключатель.



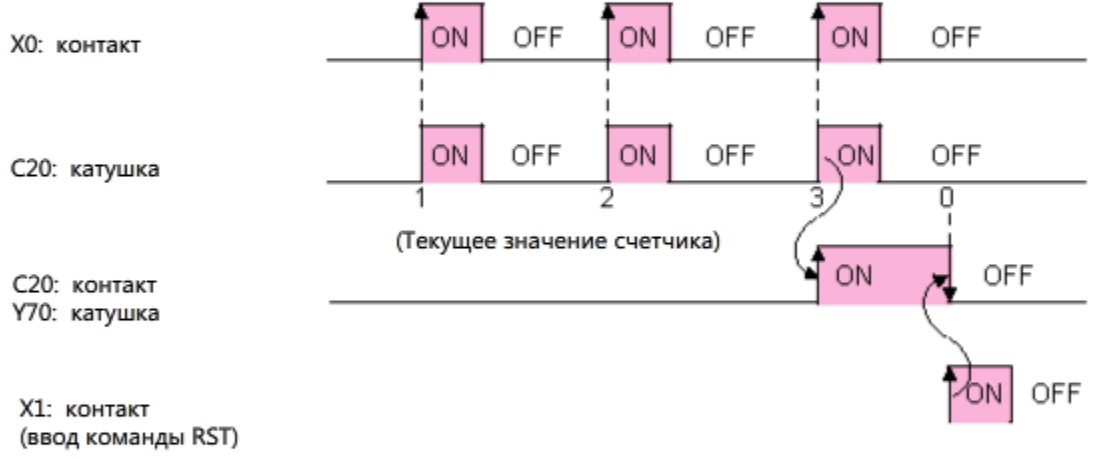
Значение в C20 увеличивается при каждом включении X0. Когда счет достигает 3 (завершения счета), включается Y70.

2.7 Подсчет

Код и функция команды

Символ	Функция
 <p>□ : адрес операнда</p>	<p>Счетчик</p> <p>В сочетании с выходом катушки (OUT) счетчик подсчитывает (по одному), сколько раз соблюдается условие.</p> <p>Завершение подсчета происходит, когда счет достигает заданного числа, и включается контакт счетчика.</p> <p>Установленное значение счетчика указывается в виде «Kn» (n: десятичное число).</p>

Временная диаграмма



Глава 3**Команды, содержащие словные операнды**

В этой главе приводятся разъяснения в отношении команд, содержащих словные операнды.

Словные операнды полезны для контроля времени, счета и ввода значений из внешнего оборудования или передачу во внешнее оборудование

- Имитируйте базовые операции программы для знакомства с работой основных команд
- Познакомьтесь в имитации с ролями команд и обработкой, выполняемой в программируемом контроллере

3.1 Перенос данных в словный операнд

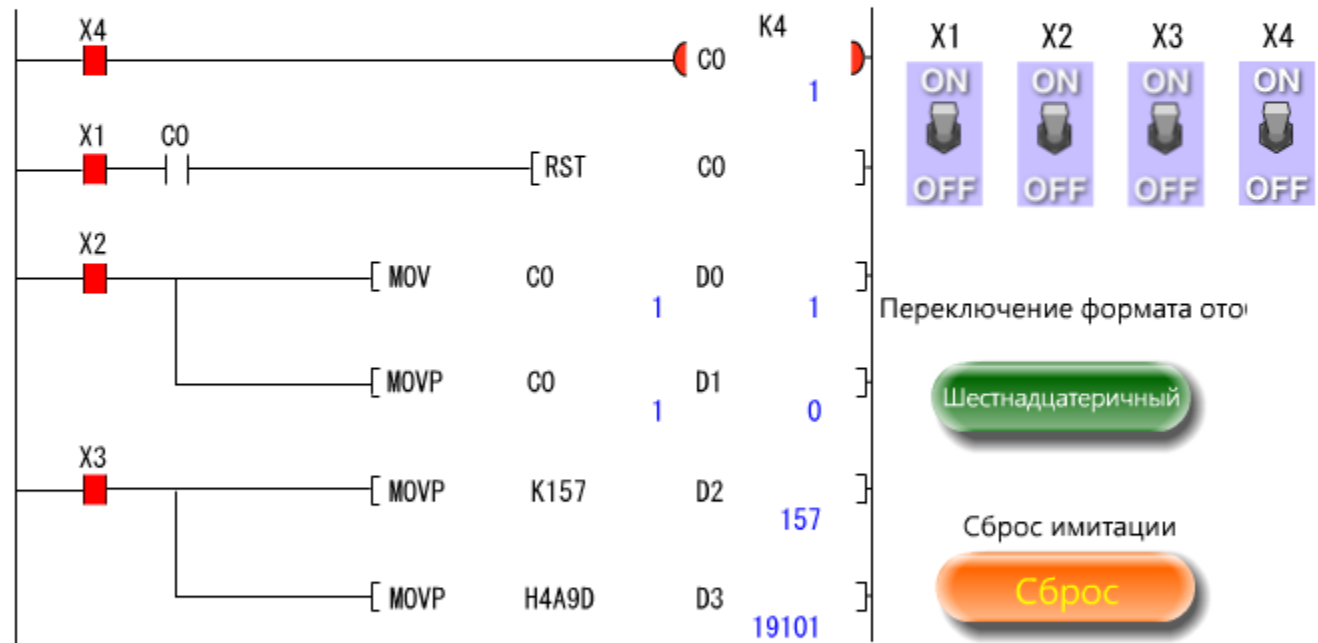
Команда передачи 16-битовых данных (MOV) переносит (копирует) блок данных из 1 слова (16 битов) в заданный словный операнд.

Переносить можно либо значение в каком-либо операнде, либо заданные данные. Для переноса доступны данные в десятичном и шестнадцатеричном формате.

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу указанных ниже команд, щелкая показанные справа входные переключатели.

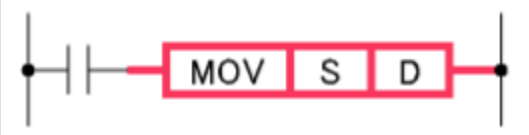
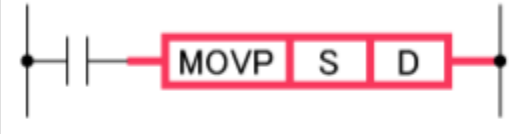
Каждое число, выделенное синим, указывает на значение (текущее значение), хранящееся в операнде.



При многократном включении/выключении X4 текущее значение CO увеличивается. (0, 1...4->0...).

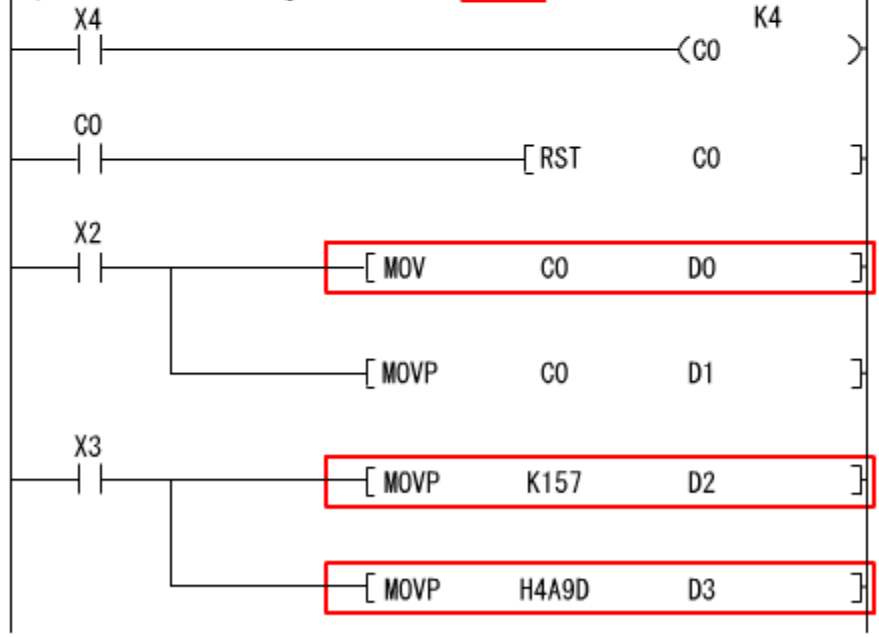
3.1 Перенос данных в словный операнд

Коды и функции команд

Символ	Функция
	<p>Передача 16-битовых данных (MOV) При соблюдении входного условия данные, указанные в источнике (S), передаются (копируются) в операнд, указанный в назначении (D).</p>
	<p>Передача 16-битовых данных (импульсных) (MOV P) На переднем фронте условия (при переходе из состояния ВЫКЛ. в состояние ВКЛ.) данные, указанные в источнике (S), передаются (копируются) в операнд, указанный в назначении (D).</p>

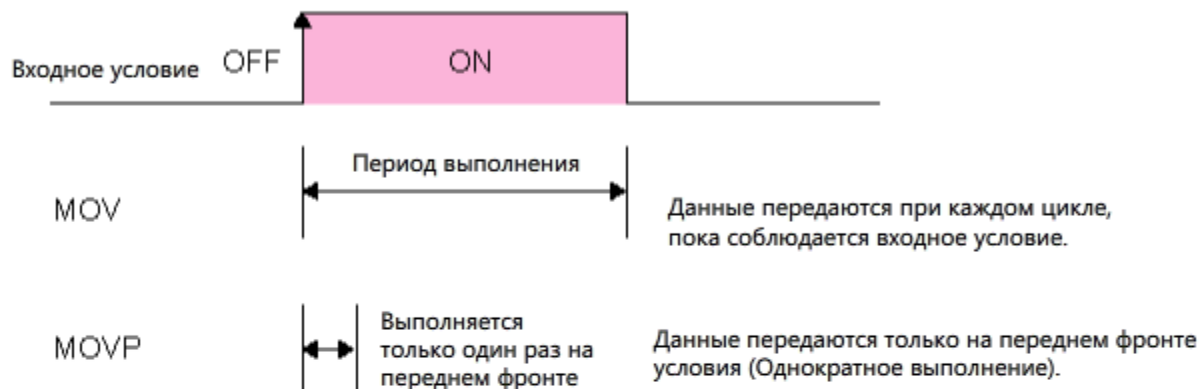
Программа на языке релейной логики

Щелкните мигающую область



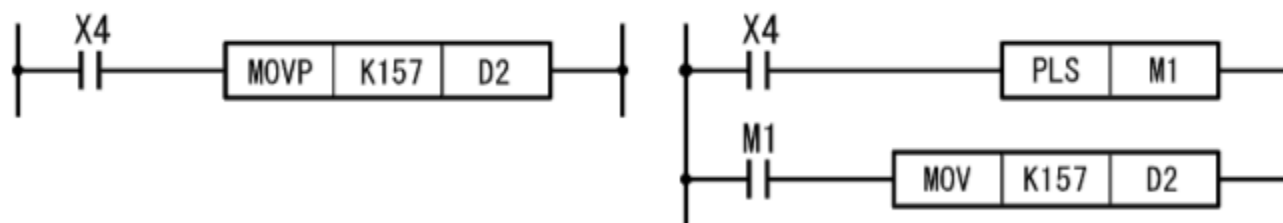
3.1.1 Различия между MOV и MOVP

Команда MOV используется для непрерывного считывания изменяющихся данных. В то же время команда MOVP используется для однократной передачи данных, например для настройки данных или считывания данных при возникновении какой-либо ошибки.



На приведенных ниже иллюстрациях показаны две программы, которые будут приводить к выполнению одной и той же операции с помощью команд MOV и MOVP. Передача данных выполняется при включении X4.

При использовании команды MOVP операция может быть выполнена без помощи команды PLS, которая задает выполнение операции на переднем фронте.



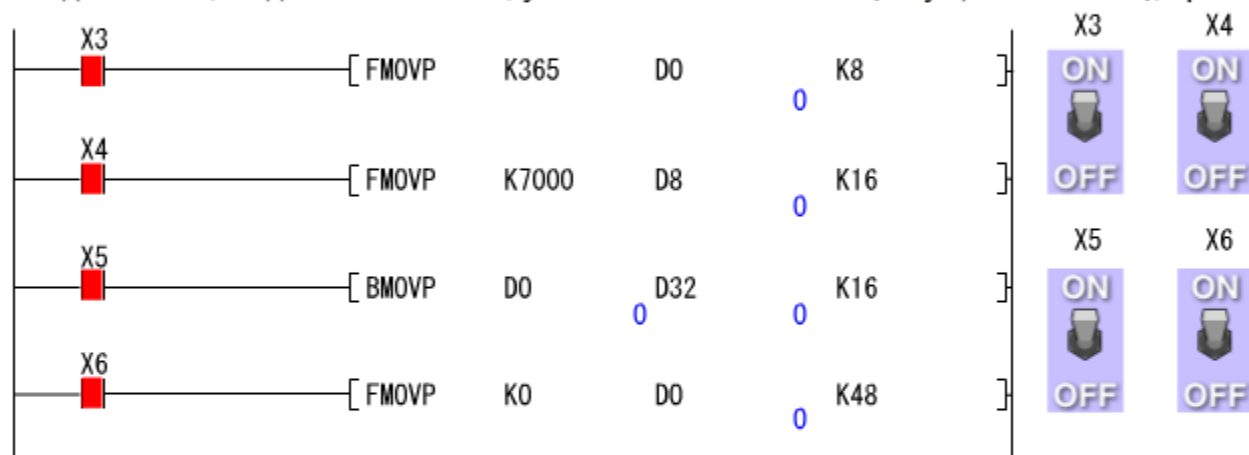
3.1.2

Одновременный перенос данных в несколько словных операндов

Команды MOV/MOVP используются для передачи данных в некоторый операнд. Для передачи данных в несколько операндов, адреса которых следуют подряд, можно использовать «команду групповой передачи одинаковых данных» (FMOV) или «команду передачи одинаковых блоков данных» (BMOV).

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу указанных ниже команд, щелкая показанные справа входные переключатели. Каждое число, выделенное синим, указывает на значение (текущее значение), хранящееся в операнде.



Монитор операндов

D0	0	D8	0	D32	0
D1	0	D9	0	D33	0
D7	0	D23	0	D47	0

При включении каждого входного сигнала сразу осуществляется передача заданных данных.

ПРИМЕЧАНИЕ. На третьей ступеньке лестницы, начинающейся с X5, данные передаются с помощью команды BMOV.

3.1.2

Одновременный перенос данных в несколько словных операндов

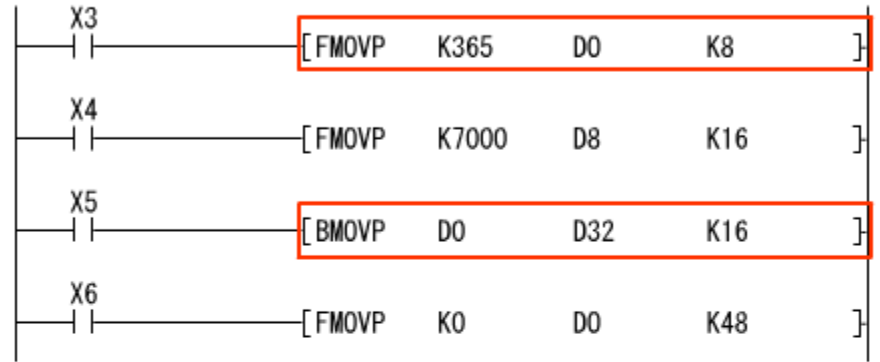
■ Коды и функции команд

Символ	Функция
	<p>Групповая передача одинаковых данных (FMOV) При соблюдении входного условия данные, указанные в источнике (S), передаются (копируются) в операнд, указанный в назначении (D), и в «n» операндов, адреса которых следуют за D.</p>
	<p>Групповая передача одинаковых данных (импульсных) (FMOVP) На переднем фронте условия данные, указанные в источнике (S), передаются (копируются) в операнд, указанный в назначении (D), и в «n» операндов, адреса которых следуют за D.</p>
	<p>Групповая передача блоков данных (BMOV) При соблюдении входного условия данные в операнде, заданном источником (S), и в «n» следующих операндов передаются в операнд, заданный операндом (D), и в «n» следующих операндов.</p>
	<p>Групповая передача блоков данных (импульсных) (BMOVP) На переднем фронте условия данные в операнде, заданном источником (S), и в «n» следующих операндов передаются в операнд, заданный операндом (D), и в «n» следующих операндов.</p>

3.1.2 Одновременный перенос данных в несколько словных операндов

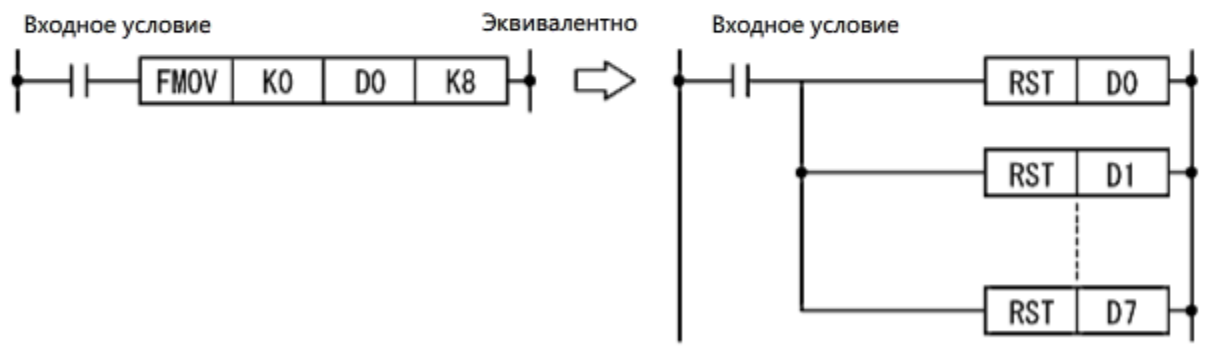
■ Программа на языке релейной логики и выполняемая операция

Щелкните мигающую область .



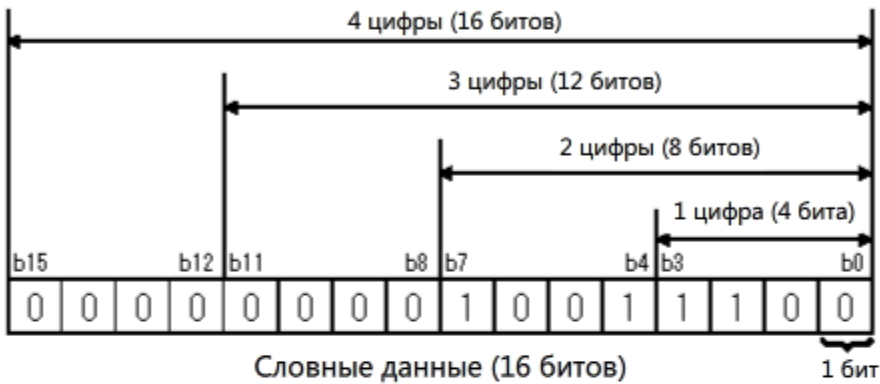
■ Применение команд FMOV и BMOV

Команда FMOV удобна для одновременного удаления большого объема данных.



3.1.3 Количество битовых операндов в командах

Четырехбитовые операнды группируются в однобитовые цифры операндов для контроля битовой информацией в определенном диапазоне (передачи данных и т. д.).



■ Порядок задания цифры битового операнда

Количество битовых операндов выражается как «константа операнда» + «адрес первого операнда». Количество бит является кратным 4. Некоторые примеры приведены в нижеследующей таблице. Ниже приводятся примеры того, когда адрес первого операнда - «M0».

Диапазон битов	Метод задания
16-битовые данные	K4M0 (16 битов, M0–M15)
32-битовые данные	K8M0 (32 бита, M0–M31)

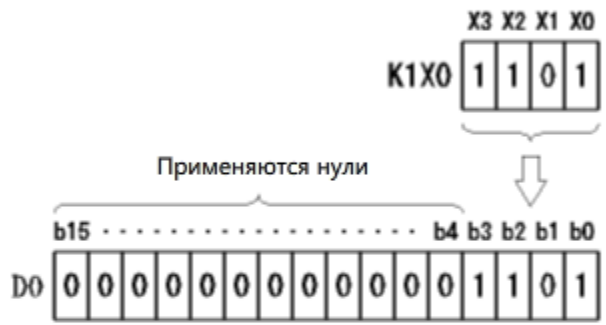
Константа операнда	Диапазон доступных для использования числовых значений
K1 (4 бита)	0–15
K2 (8 битов)	0–255
K3 (12 битов)	0–4095
K4 (16 битов)	От –32768 до 32767 16-й бит может использоваться для знака плюс/минус в целях выражения отрицательных значений.

3.1.3 Примеры передачи групп битовых операндов

Для передачи (копирования) чисел из источника в операнд назначения используются команды передачи данных. Ниже приводятся примеры того, как передаются заданные данные.

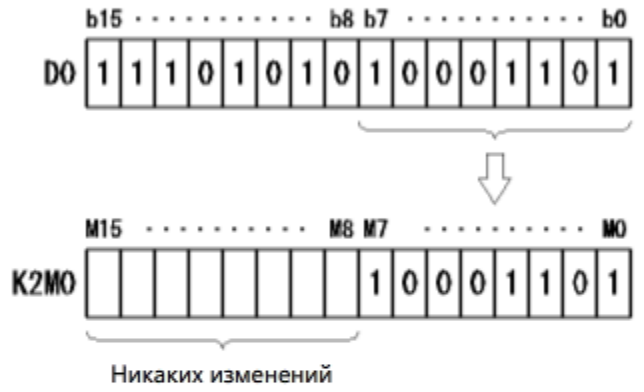
(a) Битовые операнды, заданные константами → Словные операнды

Пример: MOV K1X0 D0



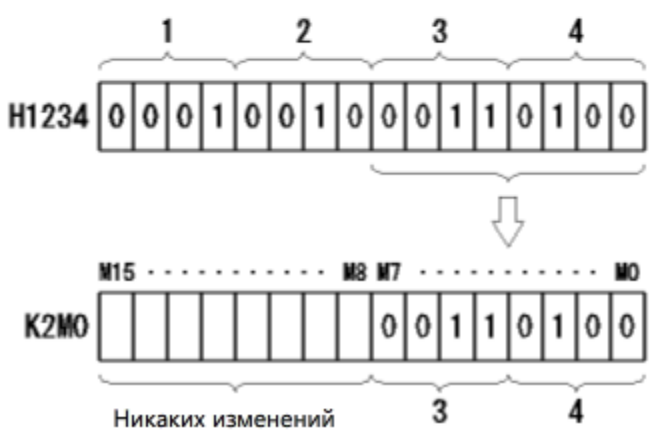
(b) Словные операнды → битовые операнды, заданные константами

Пример: MOV D0 K2M0



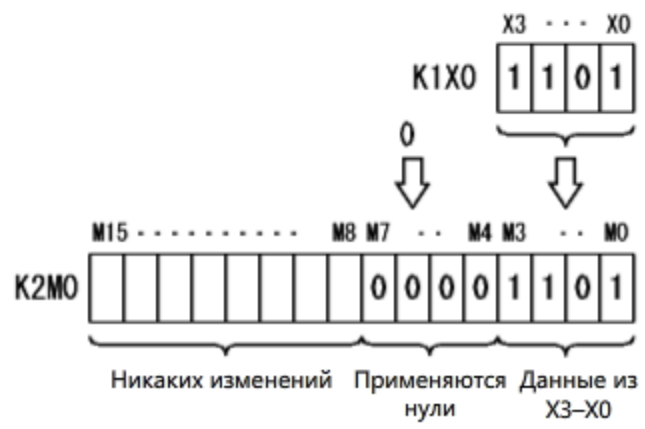
(c) Константы (числа, заданные напрямую) → битовые операнды, заданные цифрами

Пример: MOV H1234 K2M0



(d) Битовые операнды, заданные цифрами → битовые операнды, заданные цифрами

Пример: MOV K1X0 K2M0

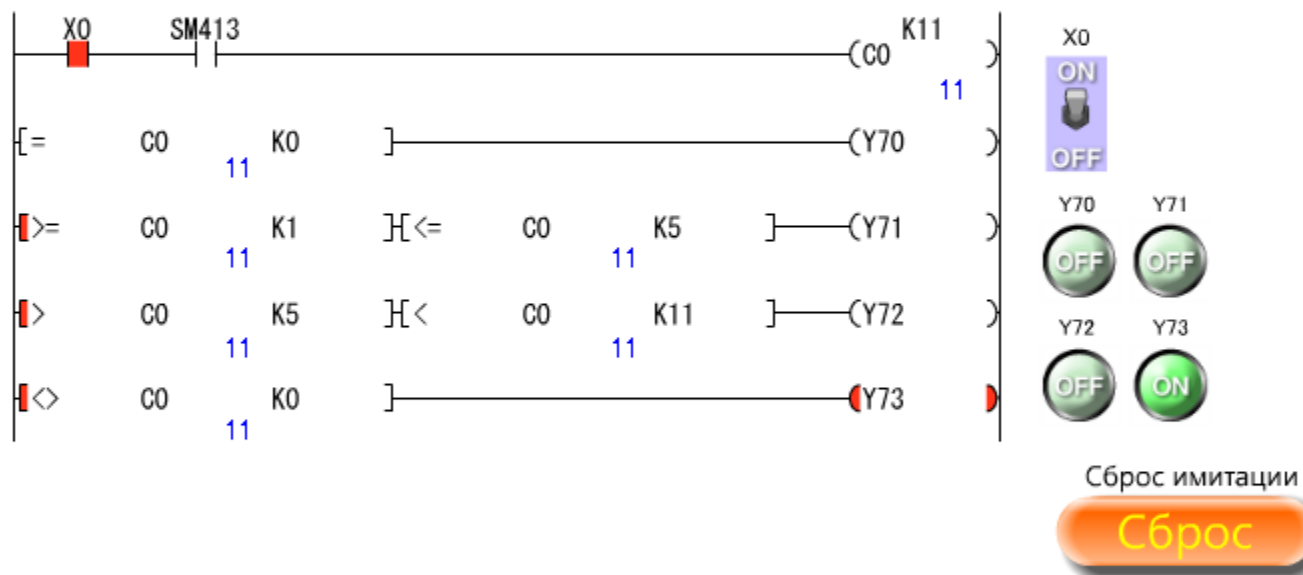


3.2 Сравнение числовых значений

Для сравнения словных данных и данных, хранящихся в словных операндах, используются команды операций сравнения. При соблюдении некоторого условия (\neq) выполняется следующая команда.

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу указанных ниже команд, щелкая показанные справа входные переключатели. Каждое число, выделенное синим, указывает на значение (текущее значение), хранящееся в операнде.



Y70–Y73 включаются/выключаются в зависимости от текущего значения C0.

SM413 - это специальное реле, которое включается или выключается с интервалом в 1 секунду модулем ЦП. (2-секундные часы)

Пока X0 включен, C0 увеличивает счет через каждые 2 секунды.

* SM413 - это специальное реле, которое включается/выключается с интервалом в 1 секунду (2-секундные часы). SM403 может использоваться для устройств серий MELSEC iQ-R/Q/L/iQ-F. Устройства серии MELSEC-F не имеют реле 2-секундных часов, но имеют M8011 (часы 0,01 с), M8012 (часы 0,1 с), M8013 (часы 1 с) и M8014 (часы 1 мин).

3.2

Сравнение числовых значений

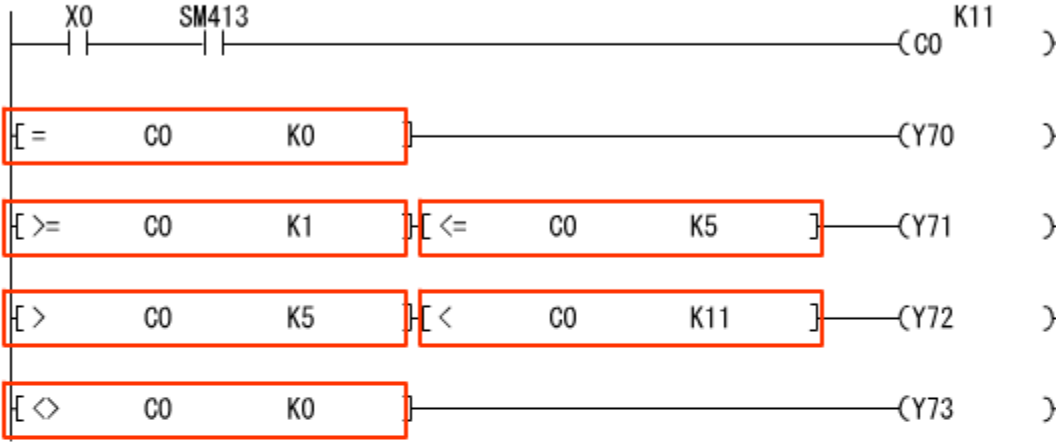
■ Коды и функции команд

Символ	Функция
	Сравнение 16-битовых двоичных данных. (=) Условие соблюдается, когда ИСТОЧНИК 1 равен ИСТОЧНИКУ 2.
	Сравнение 16-битовых двоичных данных. (<) Условие соблюдается, когда ИСТОЧНИК 1 меньше ИСТОЧНИКА 2.
	Сравнение 16-битовых двоичных данных. (>) Условие соблюдается, когда ИСТОЧНИК 1 больше ИСТОЧНИКА 2.
	Сравнение 16-битовых двоичных данных. (<=) Условие соблюдается, когда ИСТОЧНИК 1 равен ИСТОЧНИКУ 2 или меньше него.
	Сравнение 16-битовых двоичных данных. (>=) Условие соблюдается, когда ИСТОЧНИК 1 равен ИСТОЧНИКУ 2 или больше него.
	Сравнение 16-битовых двоичных данных. (<>) Условие соблюдается, когда ИСТОЧНИК 1 не равен ИСТОЧНИКУ 2.

3.2 Сравнение числовых значений

■ Программа на языке релейной логики и выполняемая операция

Щелкните мигающую область



SM413 - это специальное реле, которое включается или выключается с интервалом в 1 секунду модулем ЦП (2-секундные часы). Специальные реле (SM) представляют собой операнды реле в модуле ЦП. Каждое специальное реле выполняет определенную функцию.

В этом разделе приводятся разъяснения в отношении основных арифметических операций со словными (числовыми) операндами.

■ **Сложение и вычитание**

Арифметические операции с использованием знаков сложения (+) и вычитания (-).

■ **Умножение и деление**

Арифметические операции с использованием знаков умножения (*) и деления (/).

Для устройств серий MELSEC iQ-R/Q/L/iQ-F и устройств серии MELSEC-F используются разные команды, но основные принципы являются одинаковыми.

Разъяснения в этом разделе приводятся с применением команд, используемых в устройствах серий MELSEC iQ-R/Q/L /iQ-F.

3.3.1

Сложение и вычитание

На приведенной ниже схеме показаны команды, которые выполняют сложение и вычитание, а также сохраняют полученный результат в заданных операндах.

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу указанных ниже команд, щелкая показанные справа входные переключатели. Каждое число, выделенное синим, указывает на значение (текущее значение), хранящееся в операнде.

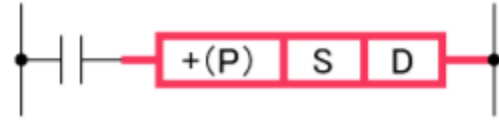

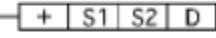
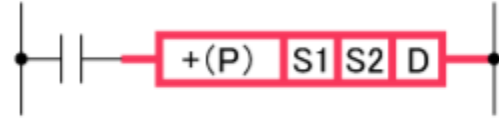
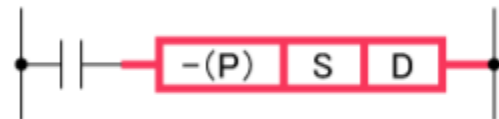
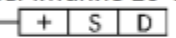
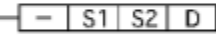
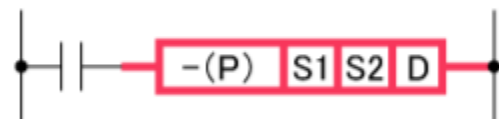


При включении каждого входного сигнала выполняется арифметическая операция.

- Пример основывается на устройствах серий MELSEC iQ-R/Q/L/iQ-F.

3.3.1 Сложение и вычитание

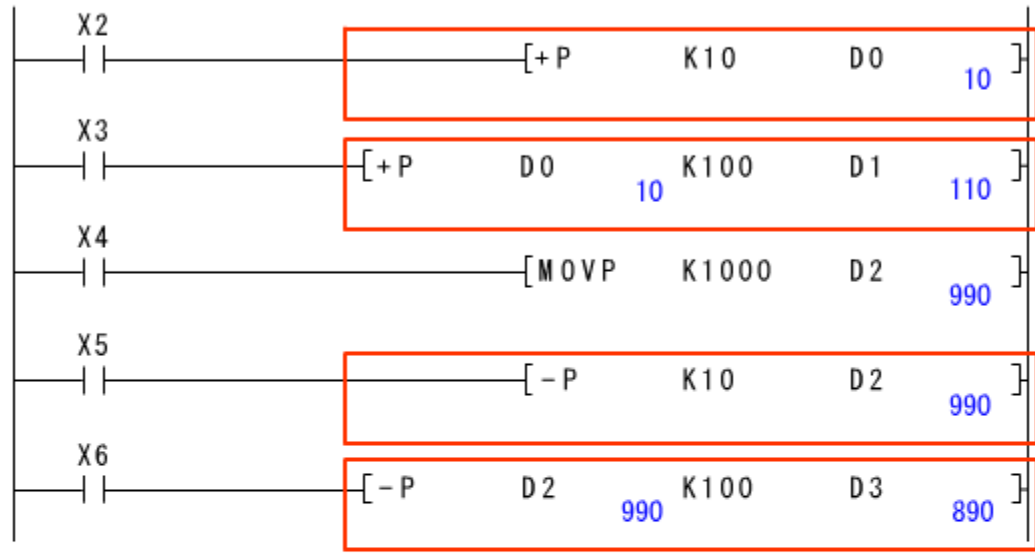
Коды и функции команд

Символ	Функция
	<p>Сложение 16-битовых двоичных данных</p> <p>-  : выполняется операция «D + S = D».</p> <p>-  : выполняется операция «S1 + S2 = D».</p>
	
	<p>Вычитание 16-битовых двоичных данных</p> <p>-  : выполняется операция «D - S = D».</p> <p>-  : выполняется операция «S1 - S2 = D».</p>
	

3.3.1 Сложение и вычитание

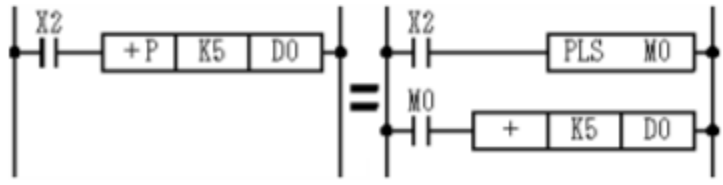
■ Программа на языке релейной логики и выполняемая операция

Щелкните мигающую область



■ Примечание о командах сложения и вычитания

В обычных условиях используйте для выполнения сложения/вычитания команду +P/-P.
 При использовании команды +/- сложение/вычитание выполняется многократно, пока соблюдается входное условие.
 При использовании любой из представленных ниже ступенек лестницы сложение выполняется только один раз при включении X2.



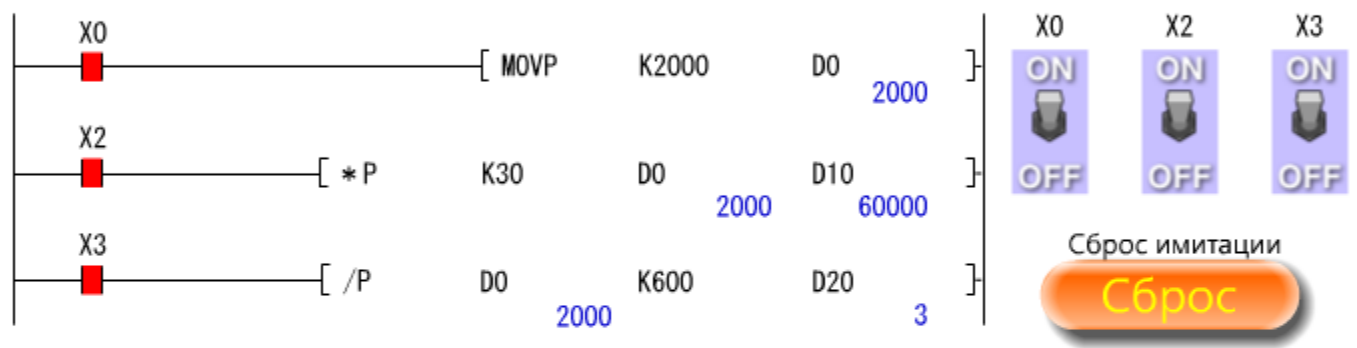
* Пример основывается на устройствах серий MELSEC iQ-R/Q/L/iQ-F.

3.3.2 Умножение и деление

На приведенной ниже схеме показаны команды, которые выполняют умножение и деление, а также сохраняют полученный результат в заданных операндах.

■ Программа на языке релейной логики и выполняемая операция

Имитируйте работу указанных ниже команд, щелкая показанные справа входные переключатели. Каждое число, выделенное синим, указывает на значение (текущее значение), хранящееся в операнде.



При включении каждого входного сигнала выполняется арифметическая операция.

* Пример основывается на устройствах серий MELSEC iQ-R/Q/L/iQ-F.

3.3.2

Умножение и деление



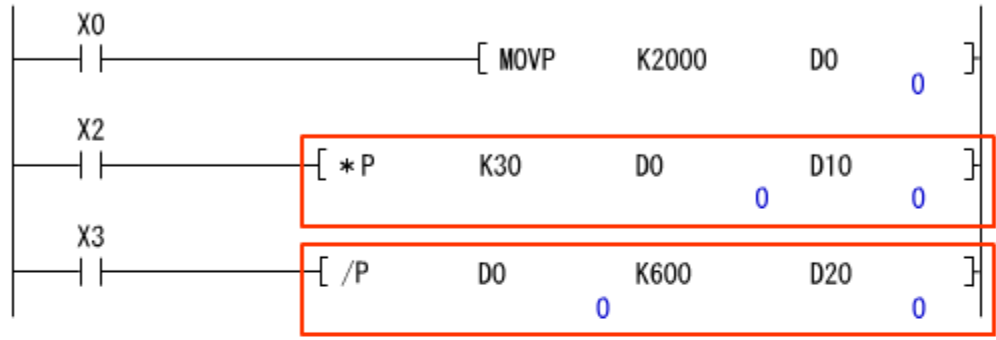
■ Коды и функции команд

Символ	Функция
	<p>Умножение 16-битовых двоичных данных (*) Выполняется операция «$S1 \times S2 = (D + 1 D)$». («D+1» - это операнд, следующий за D. Если адрес D - D100, тогда адрес «D + 1» - D101.) Результатом операции являются 32-битовые данные, которые состоят из 2 слов («D» и «D + 1»).</p>
	<p>Деление 16-битовых двоичных данных Выполняется операция «$S1/S2 = (D \text{ [частное]}, D + 1 \text{ [остаток]})$». («D + 1» - это операнд, который следует за D. Если адрес D - D100, тогда адрес «D + 1» - D101.) Результатом операции является целое число.</p>

3.3.2 Умножение и деление

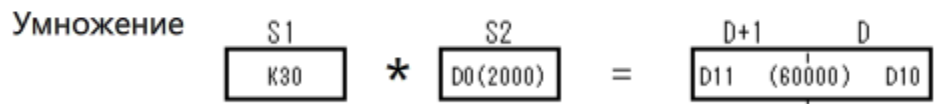
■ Программа на языке релейной логики и выполняемая операция

Щелкните мигающую область



■ Примечание о командах умножения и деления

Для выполнения команды умножения или деления требуются два словных операнда подряд (D, D+1) в качестве назначения (D).



* Пример основывается на устройствах серий MELSEC iQ-R/Q/L/iQ-F.

3.3.3 Различия между ПЛК MELSEC iQ-R/Q/L/iQ-F и MELSEC-F

В ПЛК серий MELSEC iQ-R/Q/L/iQ-F и MELSEC-F используются разные обозначения. Основные различия приведены в нижеследующей таблице.

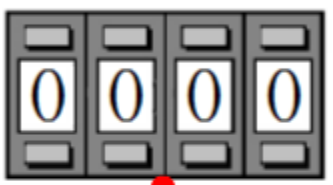
Арифметическая операция	Команда, используемая в устройствах Серия MELSEC iQ-R/Q/L/iQ-F	Команда, используемая в устройствах серии MELSEC-F	Различия
Сложение (+)			Серия MELSEC iQ-R/Q/L/iQ-F: +(P) Серия MELSEC-F: ADD(P)
Вычитание (-)			Серия MELSEC iQ-R/Q/L/iQ-F: -(P) Серия MELSEC-F: SUB(P)
Умножение (*)			Серия MELSEC iQ-R/Q/L/iQ-F: *(P) Серия MELSEC-F: MUL(P)
Деление (/)			Серия MELSEC iQ-R/Q/L/iQ-F: /(P) Серия MELSEC-F: DIV(P)

3.4 Передача/прием данных между ПЛК и устройствами ввода/вывода

Выключатель дискретного входа - это устройство ввода, обеспечивающее ввод в программируемый контроллер данных в виде числовых значений. Устройство цифровой индикации - это устройство вывода, обеспечивающее отображение полученных от программируемого контроллера данных в виде числовых значений.

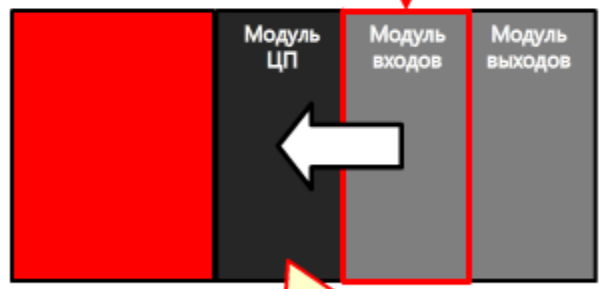
Данные, полученные от выключателя дискретного входа, должны форматироваться таким образом, чтобы их можно было обработать в программируемом контроллере. Аналогичным образом, данные, передаваемые на цифровой дисплей, должны форматироваться так, чтобы их мог считывать цифровой дисплей.

Выключатель дискретного входа



Ввод числовых значений

ПЛК



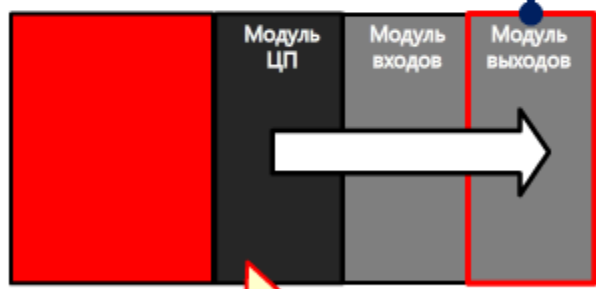
Формат для обработки программируемым контроллером

Цифровой дисплей



Отображение числовых значений

ПЛК



Формат для отображения цифровым дисплеем

3.4.1 Получение входных сигналов от выключателя дискретного входа

В программируемом контролере используется инструкция BIN для получения входных сигналов от выключателя дискретного входа.

Коды и функции команд

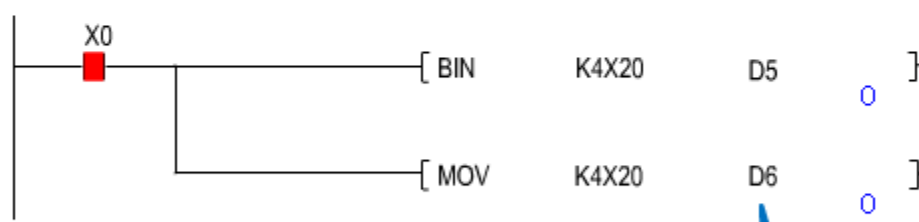
Символ	Функция
	Данные в операнде (S) преобразуются в формат, позволяющий обработать их в программируемом контроллере, а затем сохранить в операнде (D).

Программа на языке релейной логики и выполняемая операция

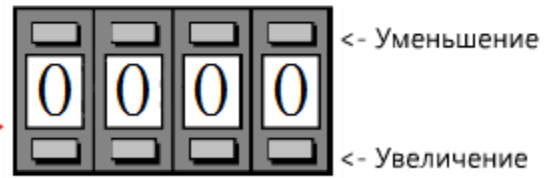
Имитируйте работу указанных ниже команд, щелкая показанные справа входные переключатели. Каждое число, выделенное синим, указывает на значение (текущее значение), хранящееся в операнде.

В D5 хранятся данные, полученные от выключателя дискретного входа после форматирования с использованием команды BIN.

В D6 хранятся неформатированные данные, полученные от выключателя дискретного входа.

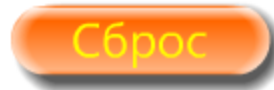


Выключатель дискретных входов



X2F až X20

Сброс имитации

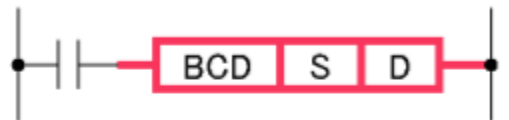


Если используется команда MOV, числа не совпадают.

3.4.2 Отображение данных ПЛК на цифровом дисплее

Для отображения данных программируемого контроллера на цифровом дисплее используется команда BCD.

Коды и функции команд

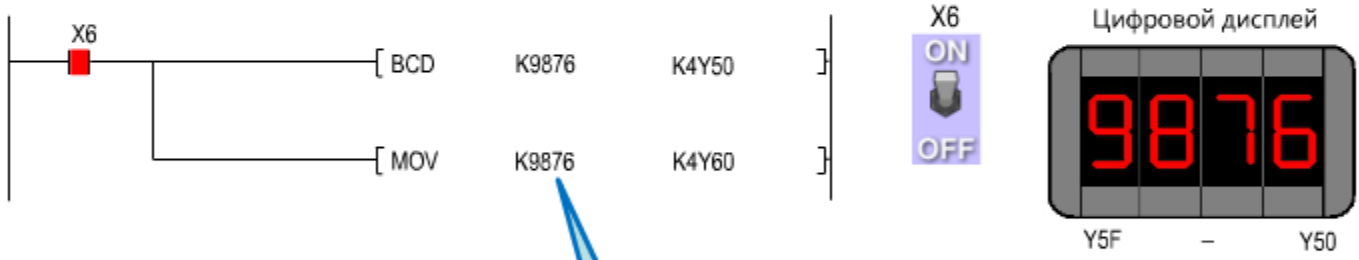
Символ	Функция
	Данные в операнде (S) преобразуются в формат, позволяющий отобразить их на цифровом дисплее, а затем сохранить в операнде (D).

Программа на языке релейной логики и выполняемая операция

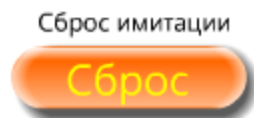
Имитируйте работу указанных ниже команд, щелкая показанные справа входные переключатели.

На верхнем цифровом дисплее отображаются данные, отформатированные с использованием команды BCD.

На нижнем цифровом дисплее отображаются неформатированные данные.



Если используется команда MOV, числа не совпадают.



В этом курсе вы узнали следующее:

- Концепция входных и выходных сигналов, подаваемых в программируемые контроллеры / из программируемых контроллеров
- Основные команды управления программируемыми контроллерами
- Информация, получаемая программируемым контроллером MELSEC, выполняется в программах на языке релейной логики в программируемом контроллере, а результаты выполнения передаются на внешние устройства в виде выходных сигналов
- Различия между форматами битовых и словных данных
- Основы управляющих программ

Пройдите курс «Engineering Software MELSOFT GX Works3 (Ladder)» (Инженерное программное обеспечение MELSOFT GX Works3 (программы на языке релейной логики)) для получения необходимых знаний о порядке редактирования и регистрации программ в модуле ЦП устройств серий MELSEC iQ-R/iQ-F.

Чтобы узнать о порядке редактирования и регистрации программ в модуле ЦП устройств серии MELSEC-Q/L/F, пройдите курс «GX Works2 Basics» (Основные сведения о GX Works2).

Теперь вы завершили все уроки курса **ПЛК - основы программирования** и готовы к прохождению заключительного теста. Если вам неясны какие-либо из рассмотренных тем, воспользуйтесь возможностью еще раз посмотреть информацию по этим темам прямо сейчас.

Данный заключительный тест содержит всего 11 вопросов (54 пунктов).

Вы можете проходить заключительный тест любое количество раз.

Порядок подсчета баллов за тест

После выбора ответа обязательно щелкните кнопку **Ответить**. Если вы продолжите, не нажав кнопку «Ответить», ваш ответ будет потерян. (Будет считаться, что вы не ответили на вопрос.)

Результаты теста

Количество правильных ответов, количество вопросов, процент правильных ответов и результат (успешно ли пройден тест) будут отображаться на странице результатов.

Правильные ответы: 11
Всего вопросов: 11
Процент: 100%

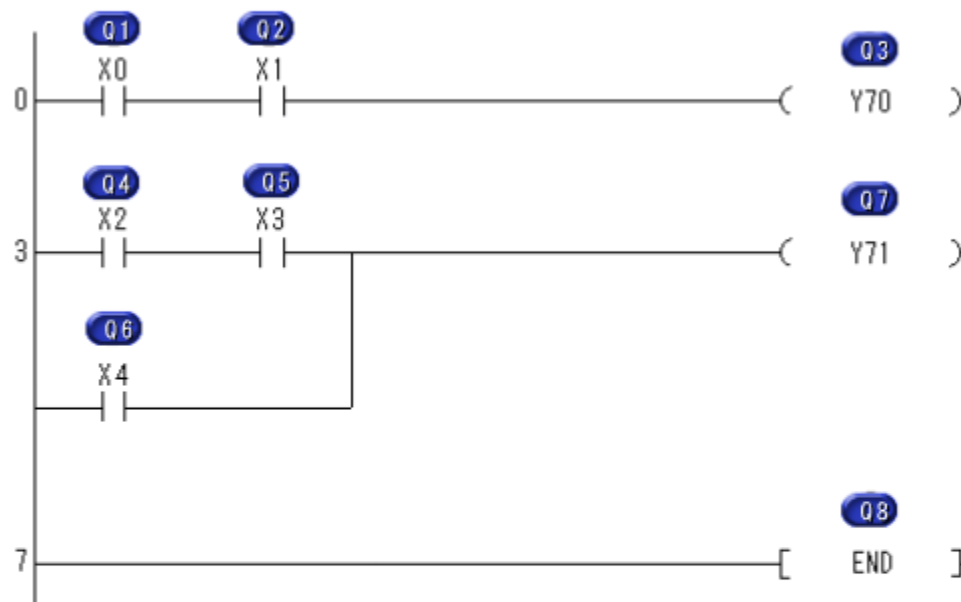
Для успешного прохождения теста вы должны правильно ответить на **60%** вопросов.

Продолжить

Просмотреть

- Щелкните кнопку **Продолжить**, чтобы завершить тест.
- Щелкните кнопку **Просмотреть**, чтобы просмотреть и проанализировать тест. (Правильные ответы будут отмечены.)
- Щелкните кнопку **Повторить попытку**, чтобы пройти тест еще раз.

Пронумеруйте указанные команды в порядке их обработки.



B1 --Select-- ▾ B2 --Select-- ▾ B3 --Select-- ▾ B4 --Select-- ▾

B5 --Select-- ▾ B6 --Select-- ▾ B7 --Select-- ▾ B8 --Select-- ▾

Ответить

Назад

Тест **Заключительный тест 2**

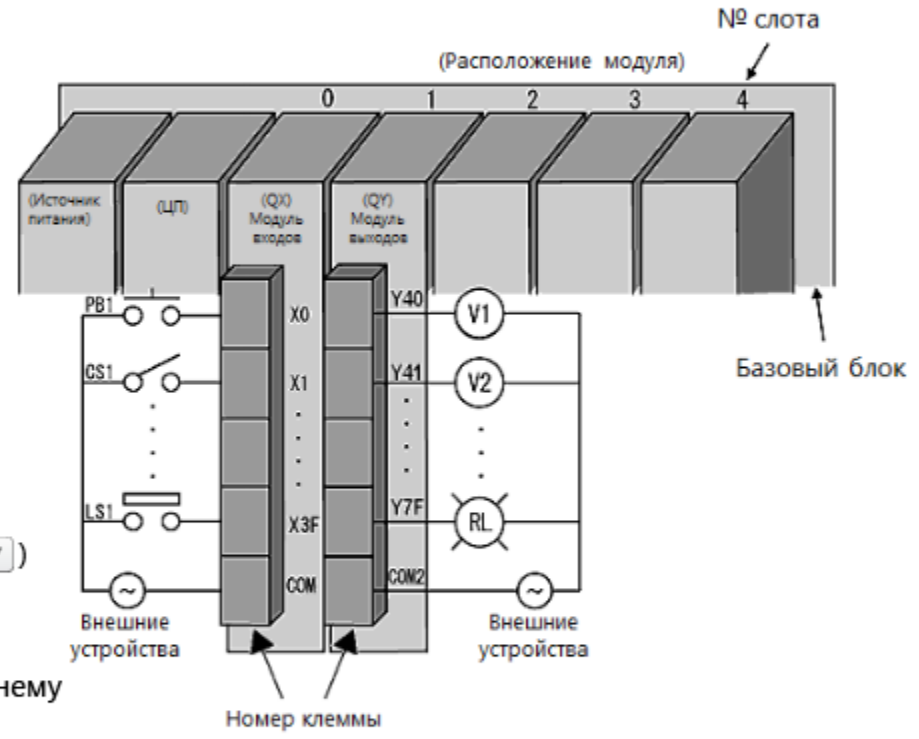
В приведенных ниже предложениях описываются внешнее оборудование ввода/вывода и сигналы ввода/вывода, передаваемые в программируемые контроллеры / из программируемых контроллеров. Выберите надлежащие слова для завершения этих предложений.

1) Номера ввода и вывода для программируемых контроллеров серии Q начинаются с (--Select--) и содержатся в (--Select--) значениях.

2) Для входных и выходных сигналов используются одинаковые номера. Поэтому перед входными сигналами ставится буква (--Select--), а перед выходными сигналами буква (--Select--).

3) Номера, назначаемые для ввода сигналов от внешнего оборудования, определяются следующими условиями:
- Где в базовом блоке установлен (--Select--)
- Номер клеммы

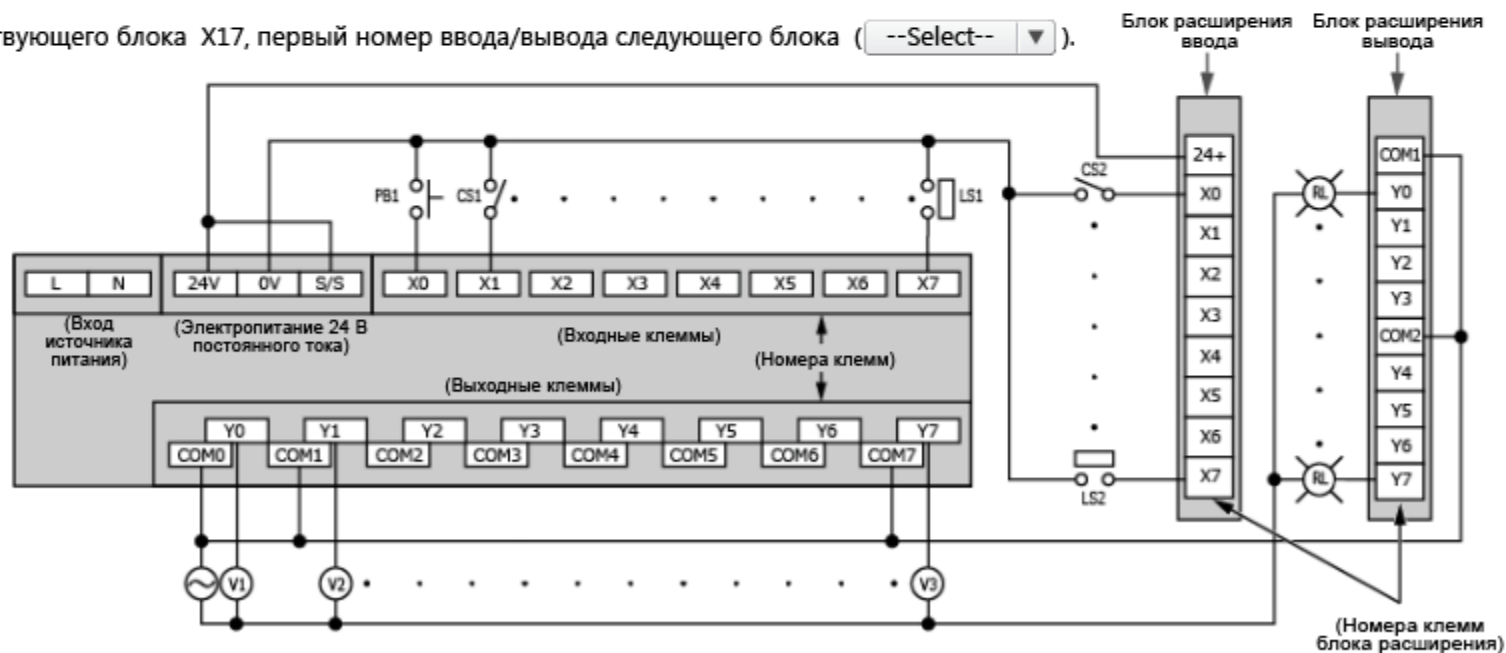
4) Номера, назначаемые выходным сигналам (катушкам) к внешнему оборудованию, определяются следующими условиями:
- Где в базовом блоке установлен (--Select--)
- Номер клеммы



Тест **Заключительный тест 3**

В приведенных ниже предложениях описываются внешнее оборудование ввода/вывода и номера ввода/вывода, назначаемые программируемым контроллерам. Выберите надлежащие слова для завершения этих предложений. (Серия MELSEC-F)

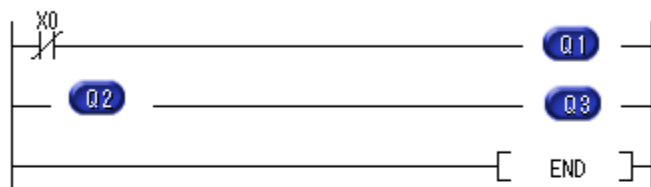
- Номера ввода и вывода для программируемых контроллеров серии MELSEC-F начинаются с (--Select--) и содержатся в (--Select--) значениях.
- Для входных и выходных сигналов используются одинаковые номера. Поэтому перед входными сигналами ставится буква (--Select--), а перед выходными сигналами буква (--Select--).
- При добавлении блока расширения ввода/вывода данному блоку будет назначен номер, который следует после номера, назначенного предшествующему (--Select--).
- Номера ввода/вывода блока всегда начинаются с номера, имеющего «0» в качестве первой цифры. Если последний номер ввода/вывода предшествующего блока X17, первый номер ввода/вывода следующего блока (--Select--).



Перетащите соответствующие команды, чтобы создать программу, выполняющую следующие операции:

При выключении переключателя X0 загорается лампа A. (Y70 ВКЛ.)

При включении данного переключателя загорается лампа B. (Y71 ВКЛ.)



B1 --Select-- ▼

B2 --Select-- ▼

B3 --Select-- ▼

Ответить

Назад

Тест **Заключительный тест 5**

Перетащите соответствующие команды, чтобы создать программу, выполняющую следующие операции:

Во время обработки материалов включен «сигнал выполнения обработки» (X0)

На переднем фронте «сигнала выполнения обработки» (X0) лампа А включена (Y70 ВКЛ.), а лампа В выключена (Y71 ВЫКЛ.).

На заднем фронте «сигнала выполнения обработки» (X0) лампа В включена (Y70 ВКЛ.), а лампа А выключена (Y71 ВЫКЛ.).



B1 --Select-- B2 --Select-- B3 --Select-- B4 --Select--

Ответить Назад

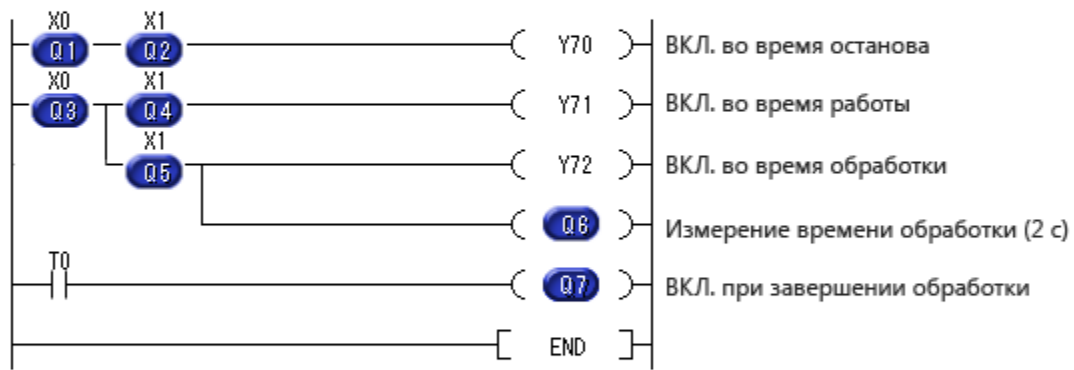
Тест **Заключительный тест 6**

Перетащите соответствующие команды, чтобы создать программу, выполняющую следующие операции:

Лампа включается при включении/выключении переключателя запуска операции (X0) и переключателя запуска обработки (X1).

Через 2 секунды после включения обоих переключателей загорается лампа D.

[Запуск операции (X0)]	[Переключатель запуска обработки (X1)]	[Лампа]
ВЫКЛ.	ВЫКЛ.	Лампа А (Y70 ВКЛ.)
ВКЛ.	ВЫКЛ.	Лампа В (Y71 ВКЛ.)
ВКЛ.	ВКЛ.	Лампа С (Y72 ВКЛ.) и через 2 с лампа D (Y73 ВКЛ.)



B1 --Select-- B2 --Select-- B3 --Select-- B4 --Select--

B5 --Select-- B6 --Select-- B7 --Select--

Ответить Назад

Тест **Заключительный тест 7**

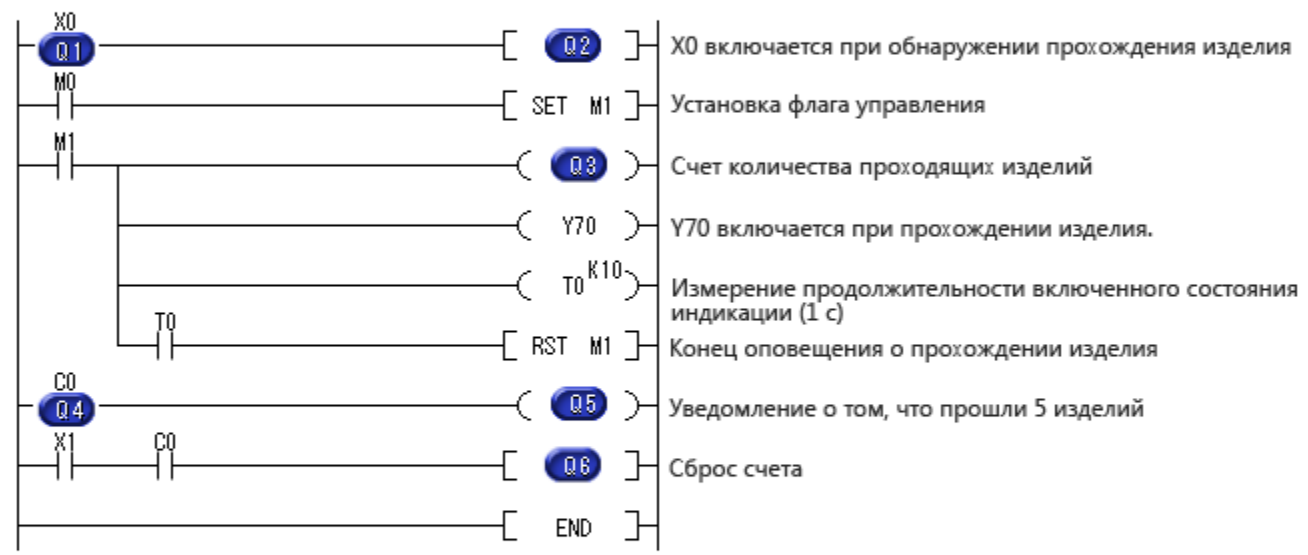
Перетащите соответствующие команды, чтобы создать программу, выполняющую следующие операции:

Пока изделие проходит по конвейеру, сигнал X0 включен.

После завершения прохождения изделия (через 3 с) включается лампа А. (Y70 ВКЛ. в течение 1 с.)

После прохождения 5 изделий включается лампа В. (Y71 ВКЛ.)

После включения лампы В включается переключатель подтверждения (X1).



B1 --Select-- B2 --Select-- B3 --Select-- B4 --Select--

B5 --Select-- B6 --Select--

Ответить Назад

Тест **Заключительный тест 8**

Перетащите соответствующие команды, чтобы создать программу, выполняющую следующие операции:

- 1) При запуске операции включается лампа А. (Y70 ВКЛ.)
- 2) Запланированное количество производимой продукции вводится с помощью дискретных переключателей BCD (X20X2F). Это количество передается как есть в регистр данных D0 при вводе.
- 3) Значения BCD, сохраняемые в регистре данных (D0, D1), непрерывно передаются на цифровой дисплей BCD. Y40Y4F: указывают на запланированное количество производимой продукции (D0)
Y50Y5F: указывают на фактическое количество произведенной продукции (D1)
- 4) Для ввода фактического количества произведенной продукции используются дискретные переключатели BCD X30X3F. При включении переключателя завершения настройки (X0) фактическое количество произведенной продукции передается в регистр данных D1.



* В этой программе для передачи данных используется команда MOV.
 * Для контроля D0 и D1 используйте шестнадцатеричные значения.

B1 B2 B3 B4

Перетащите соответствующие команды, чтобы создать программу, выполняющую следующие операции:

- 1) При запуске операции включается лампа А. (Y70 ВКЛ.)
- 2) При включении переключателя завершения настройки (X0) один раз выполняются следующие операции.
 - Запланированное количество производимой продукции А, которое было введено с помощью дискретных переключателей BCD (X20X2F), форматируется в BIN и передается в регистр данных D0.
 - Запланированное количество производимой продукции В, которое было введено с помощью дискретных переключателей BCD (X30X3F), форматируется в BIN и передается в регистр данных D1.
 - Регистры данных D0 и D1 сравниваются друг с другом, а результат показывается с помощью лампы.
 - D0 > D1: лампа В (Y71 ВКЛ./ВЫКЛ.)
 - D0 = D1: лампа С (Y72 ВКЛ./ВЫКЛ.)
 - D0 < D1: лампа D (Y73 ВКЛ./ВЫКЛ.)

B1 B2 B3

Перетащите соответствующие команды, чтобы создать программу, выполняющую следующие операции:

- 1) При запуске операции включается лампа А. (Y70 ВКЛ.)
- 2) В начале в регистре данных D0 сохраняется запланированное количество производимой продукции 100.
- 3) При завершении производства каждого изделия в регистрах данных сохраняется следующая информация.

D1: фактическое количество произведенной продукции (считается на переднем фронте X0)

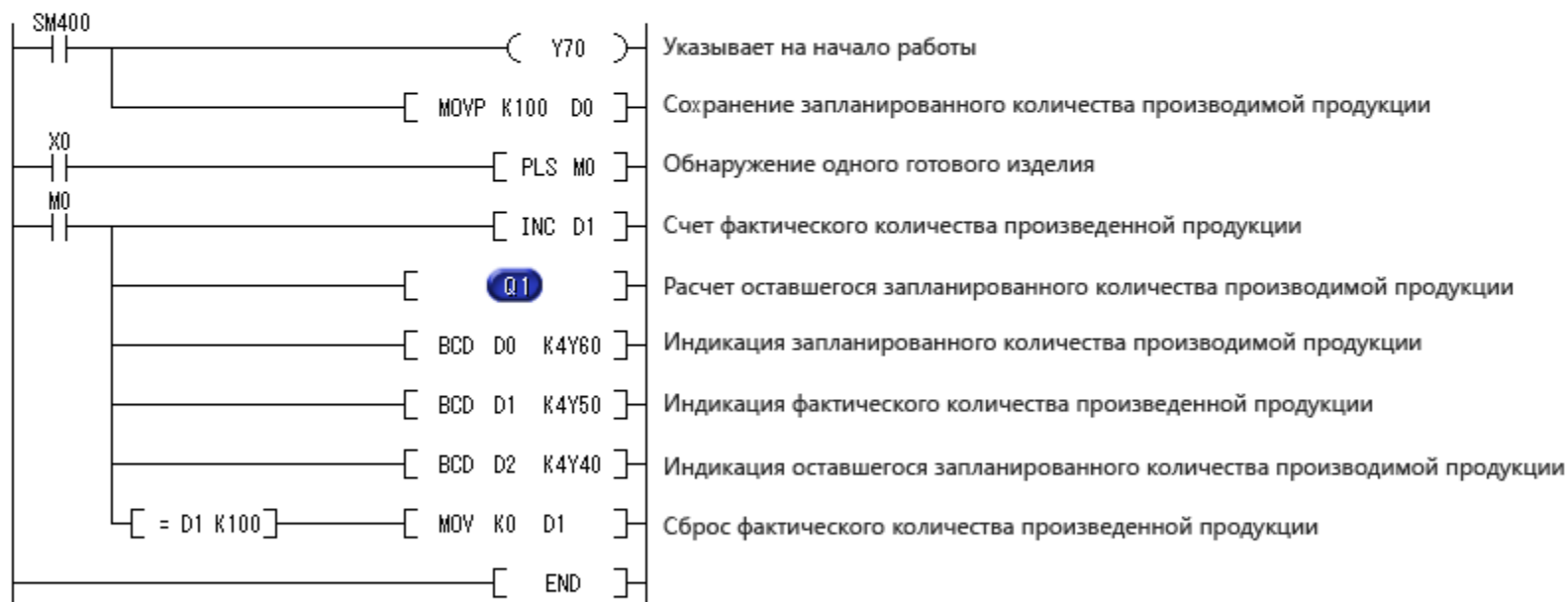
D2: оставшееся запланированное количество производимой продукции ($D2 = D0 - D1$)

На цифровом дисплее BCD отображаются следующие данные:

Y40Y4F: значение в D2 (оставшееся запланированное количество производимой продукции (0100))

Y50Y5F: значение в D1 (фактическое количество произведенной продукции (0100))

Y60Y6F: значение D0 (запланированное количество производимой продукции (100))



B1 --Select-- ▾

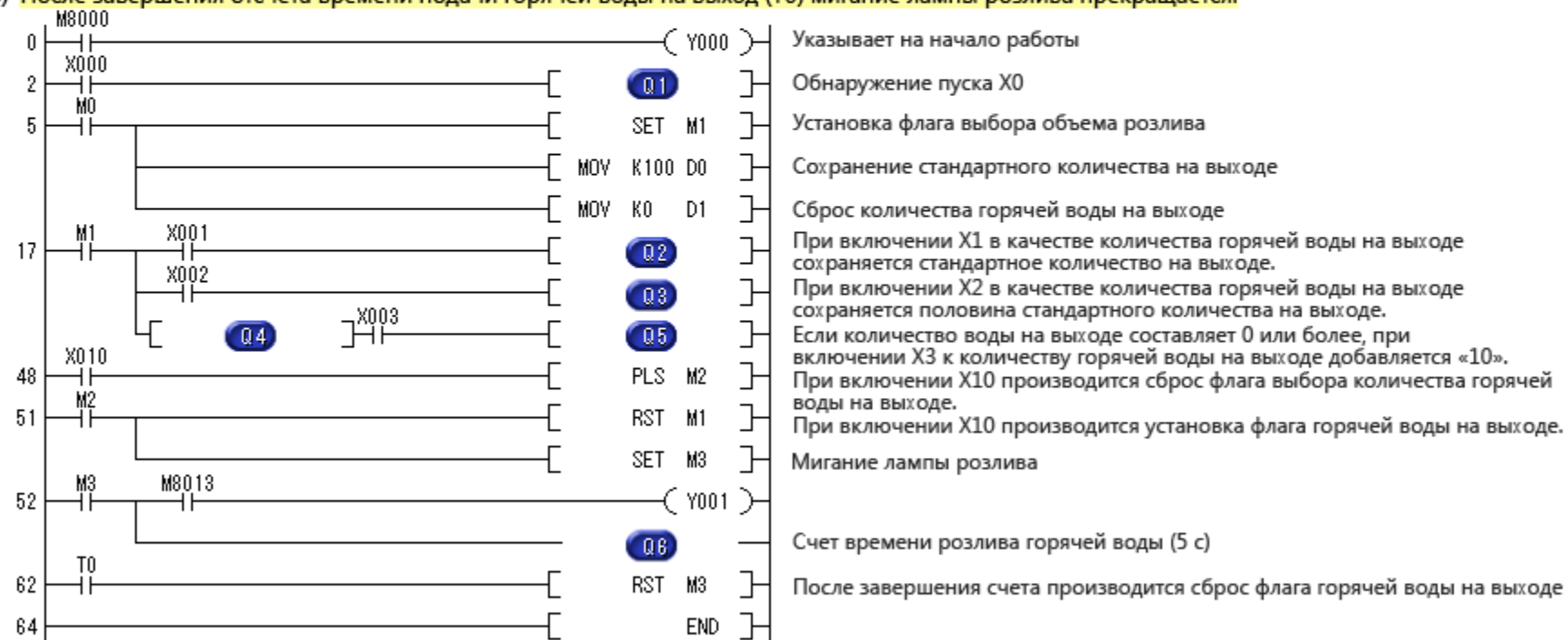
Ответить

Назад

Тест Заключительный тест 11

Показанная ниже управляющая программа предназначена для устройств серии MELSEC-F и содержит команды и специальные реле. Перетащите соответствующие команды, чтобы создать программу для розлива горячей воды из дозатора горячей воды:

- 1) При запуске операции включается лампа. (Y0 ВКЛ.)
- 2) На переднем фронте запуска работы дозатора горячей воды (при включении X0) «100» сохраняется в качестве стандартного объема дозирования воды D0, а «0» сохраняется в качестве количества горячей воды на выходе D1. (Сброс данных)
- 3) Выбор количества горячей воды на выходе.
На переднем фронте X1 в качестве количества горячей воды на выходе D1 сохраняется стандартное количество на выходе D0.
На переднем фронте X2 в качестве количества горячей воды на выходе D1 сохраняется половина стандартного количества на выходе D0.
- 4) Если выбирается количество горячей воды на выходе D1, и оно составляет 0 или более, на переднем фронте X3 к количеству горячей воды на выходе D1 добавляется «+10», а затем добавленное значение сохраняется в качестве количества горячей воды на выходе.
- 5) На переднем фронте подачи горячей воды на выход (X10) лампа розлива мигает с периодичностью 1 с (Y1 периодически включается/выключается), и ведется отсчет 5 с времени подачи горячей воды на выход (T0).
- 6) После завершения отсчета времени подачи горячей воды на выход (T0) мигание лампы розлива прекращается.

B1 B2 B3 B4 B5 B6

Вы завершили заключительный тест. Ваша область результатов является следующей.
Чтобы закончить заключительный тест, перейдите к следующей странице.

Правильные ответы: **11**

Всего вопросов: **11**

Процент: **100%**

Продолжить

Просмотреть

Поздравляем! Вы успешно прошли тест.

Вы завершили курс **ПЛК - основы программирования**.

Благодарим вас за прохождение этого курса.

Надеемся, что вам понравились уроки, а информация, полученная в рамках этого курса, окажется полезной в будущем.

Вы можете проходить данный курс любое количество раз.

Просмотреть

Закреть