

PLC

Podstawy programowania

Ten kurs przeznaczony jest dla uczestników, którzy będą po raz pierwszy tworzyć programy dla sterowników programowalnych.

Wprowadzenie **Cel kursu**



Ten kurs opisuje programowanie, którego można używać do sterowników programowalnych typu MELSEC. Jednym z głównych języków programowania jest schemat drabinkowy (LD). Kurs obejmuje najważniejsze zagadnienia dotyczące programowania drabinkowego, w tym jego główne instrukcje.

Niektóre części tego kursu opierają się na kursie podstawowym dotyczącym sterowników programowalnych MELSEC. Zaleca się przeprowadzenie stosownych kursów podstawowych przed przeprowadzeniem niniejszego kursu.

Wprowadzenie **Struktura kursu**



Kurs składa się z następujących części.

Rozdział 1 - Programowanie PLC

W tym rozdziale opisano najważniejsze zagadnienia dotyczące programowania drabinkowego.

Rozdział 2 - Operacje bitowe

W tym rozdziale opisano operacje bitowe (WŁ./WYŁ.).

Rozdział 3 - Instrukcje na rejestrach

W tym rozdziale opisano instrukcje na rejestrach (numeryczne).

Rozdział 4 - Instrukcje rozgałęziające programu

W tym rozdziale opisano instrukcje tworzące rozgałęzione programy.

Test końcowy

Wynik pozytywny: 60% lub więcej.

Wprowadzenie Jak korzystać z tego narzędzia do e-learningu

Przejdź do następnej strony		Przejdź do następnej strony.
Wróć do poprzedniej strony		Wróć do poprzedniej strony.
Przejdź do żądanej strony		Wyświetli się „Spis treści”, umożliwiający przejście do żądanej strony.
Zakończ naukę		Zakończ naukę.

Wprowadzenie Wskazówki dotyczące użytkowania



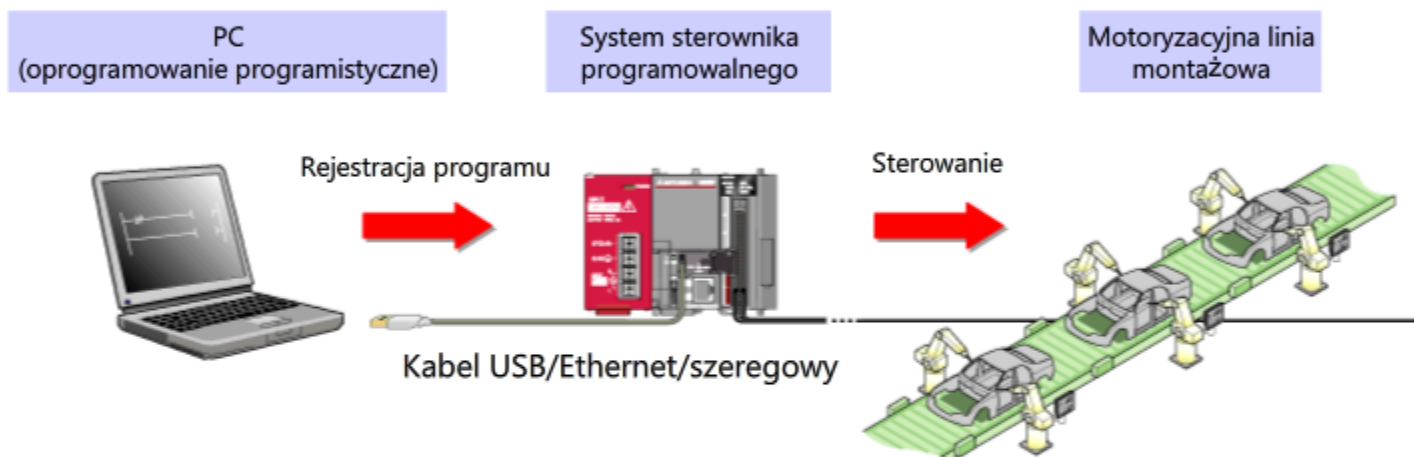
Środki bezpieczeństwa

Jeśli uczysz się przy użyciu rzeczywistych produktów, przeczytaj dokładnie zalecenia dotyczące środków bezpieczeństwa znajdujące się w odpowiednim podręczniku.

Rozdział 1 Program sterujący

Operacje wykonywane przez sterownik programowalny są zapisywane jako programy sterujące. Te programy są rejestrowane w module CPU, sterując różnymi sygnałami wejścia i wyjścia (we/wy). Języki programowania stosowane w przypadku sterowników programowalnych to między innymi logika drabinkowa, lista instrukcji (IL) i sieć działań (SFC).

Kurs wyjaśnia najważniejsze zagadnienia dotyczące programowania drabinkowego, w tym jego główne instrukcje.



Dla celów tego kursu do tworzenia programów używane jest oprogramowanie programistyczne do sterowników programowalnych GX Works2 lub GX Works3.

Aby dowiedzieć się, w jaki sposób korzystać z oprogramowania programistycznego do sterowników programowalnych, należy przeprowadzić szkolenie „GX Works2 Basics” (Podstawy GX Works2) lub „Engineering Software MELSOFT GX Works3 (Ladder)” (Oprogramowanie programistyczne MELSOFT GX Works3 (program drabinkowy)).

Oprogramowanie GX Works2 obsługuje modele MELSEC-Q/L/F.

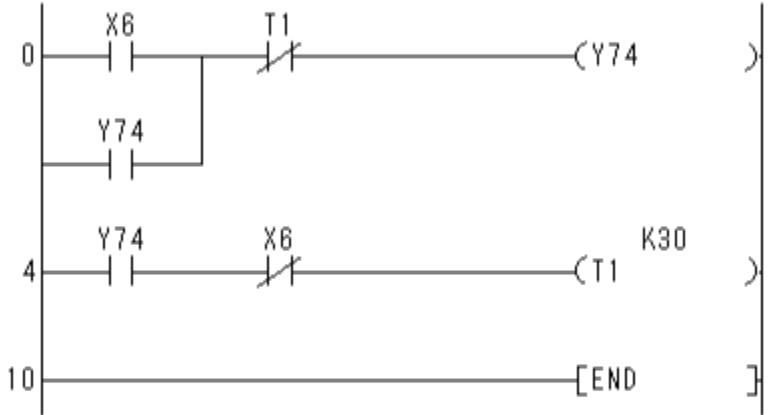
Oprogramowanie GX Works3 obsługuje modele MELSEC iQ-R/iQ-F.

1.1 Języki programowania

Języki programowania stosowane w przypadku sterowników programowalnych to między innymi logika drabinkowa, lista instrukcji (IL) i sieć działań (SFC).

Program drabinkowy jest graficznym diagramem logicznym bazującym na obwodzie elektrycznym. W programach drabinkowych symbole oznaczające instrukcje są połączone liniami, podobnymi do schematu obwodu, dzięki czemu przepływy operacji są łatwo rozpoznawalne.

Dodatkowo, programowanie drabinkowe nie wymaga specjalnej wiedzy w zakresie programowania, na przykład znajomości języków C i BASIC, i może być łatwo zrozumiane przez osoby posiadające doświadczenie w pracy z obwodami elektrycznymi i przekaźnikami.



Poniższa tabela przedstawia ten sam program w języku IL. Język IL wymaga pewnej wiedzy programistycznej, aby przedstawić operacje w formie instrukcji.

Krok nr	Instrukcja	Urządzenie
0	LD	X6
1	OR	Y74
2	ANI	T1
3	OUT	Y74
4	LD	Y74
5	ANI	X6
6	OUT	T1 K30
10	END	

Programy dla sterowników programowalnych mogą obsługiwać dwa typy wartości.

Bit	Bit jest wyrażany przez dwa typy sygnałów elektrycznych, WŁ. i WYŁ. Mogą być również przedstawione jako „1” (WŁ.) i „0” (WYŁ.). Wartości bitowe są często stosowane do określania statusów urządzeń we/wy, takich jak przełączniki i lampki.
Słowo	Liczby i znaki. Wartości słowa są często stosowane do określania wielkości i czasu. * W tym kursie są opisywane jedynie liczby. Szczegóły dotyczące znaków używanych jako wartości słowa można znaleźć w podręcznikach odpowiednich produktów.

Poniższe formaty numeryczne są stosowane do określenia wartości.

- Dziesiętny
- Binarny
- Szesnastkowy
- Ósemkowy

1.2.1**Zapis dziesiętny**

W zapisie dziesiętnym wielkość (wartość) liczby jest reprezentowana za pomocą formatu o podstawie 10 „od 0 do 9”.

W sterownikach programowalnych typu MELSEC liczby dziesiętne są poprzedzone literą „K”.

Na przykład „K153” oznacza liczbę dziesiętną „153”.

1.2.2

Zapis binarny

Podczas gdy zapis dziesiętny jest zwykle używany do zapisu wielkości i czasu, sterowniki programowalne i sterowniki osobiste używają danych binarnych, które są kombinacjami „0” i „1”.

Poniższa tabela przedstawia odpowiedniość pomiędzy wartościami dziesiętnymi i binarnymi do wartości liczby dziesiętnej „8”.

Zapis Dziesiętny	Zapis Binarny
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
⋮	⋮

Jeśli instrukcja składająca się z 1 słowa jest używana w programie, będzie zapisana i przetwarzana jako 16-bitowy element danych binarnych przez właściwy sterownik programowalny. Taki 16-bitowy element danych binarnych jest synonimem „1 słowa”.

Na przykład liczba dziesiętna „157” jest wyrażana jako „0000000010011101” w zapisie binarnym.

W zapisie dziesiętnym bity są zapisywane od prawej strony. (Bit najbardziej na prawo jest bitem początkowym.)

b15	~	b8	b7	b6	b5	b4	b3	b2	b1	b0	← Pozycja bitu
0	...	0	1	0	0	1	1	1	0	1	← Zapis Binarny
2^{16}		2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	← Potęga liczby 2
32768		256	128	64	32	16	8	4	2	1	← Waga dla wartości dziesiętnych

Aby zamienić wartości binarne na dziesiętne, pomnóż każdy status bitu („0” lub „1”) przed odpowiednią wagę i dodaj wszystkie iloczyny.

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

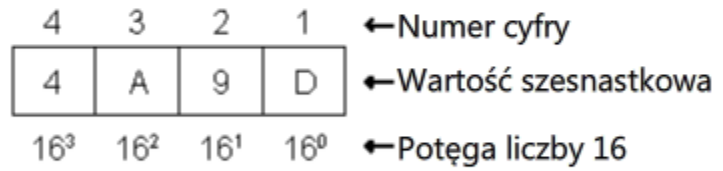
Jak widać na powyższym przykładzie, system binarny można traktować jako format liczbowy opierający się na wagach.

1.2.3 Zapis szesnastkowy

W zapisie szesnastkowym wielkość (wartość) liczby jest reprezentowana za pomocą formatu o podstawie 16 lub określana za pomocą 16 znaków alfanumerycznych: od 0 do 9 i od A do F. Każda cyfra w zapisie szesnastkowym wzrasta w kolejności 0, 1...9, A...E oraz F. Jeśli wartość przekroczy podstawę „F”, przenoszona jest na lewą stronę i staje się „10”. W sterownikach programowalnych typu MELSEC liczby szesnastkowe są poprzedzone literą „H”.
 Na przykład „H4A9D” oznacza liczbę szesnastkową „4A9D”.

Zapis binarny może być długi i skomplikowany w użyciu w programach i podczas wyświetlania monitorowanych danych. W takim przypadku korzystniejsze jest użycie zapisu szesnastkowego. Jedna cyfra wartości szesnastkowej może odpowiadać 4 bitom (4 cyfry) wartości binarnych.

Poniższy rysunek przedstawia, w jaki sposób wartość szesnastkowa jest wyrażana jako wartość dziesiętna.



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

(4096) (10) (256) (16) (13) (1)

$$= 19101$$

* Jedna cyfra wartości szesnastkowej może odpowiadać 4 bitom wartości binarnych.

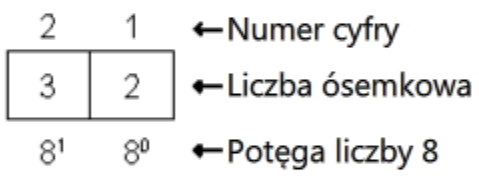
Zapis dziesiętny	Zapis binarny	Zapis szesnastkowy
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

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

1.2.4 Zapis ósemkowy

W zapisie dziesiętnym wielkość (wartość) liczby jest reprezentowana za pomocą formatu o podstawie 8 („od 0 do 7”). Gdy wartość wzrasta od „0”, „1”, „2” i przekroczy „7”, przenoszona jest na lewą stronę i staje się „10”. Zapis ósemkowy jest używany do określania ilości we/wy modelu MELSEC iQ-F/F.

Poniższy rysunek przedstawia, w jaki sposób wartość ósemkowa jest wyrażana jako wartość dziesiętna.



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

(8) (1)

$$= 26$$

* Jedna cyfra wartości ósemkowej może odpowiadać 3 bitom wartości binarnych.

Zapis dziesiętny	Zapis binarny	Zapis szesnastkowy	Zapis ósemkowy
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/>			
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/>			
16	1000	10	20
17	10001	11	21
18	10010	12	22
.	.	.	.
.	.	.	.
.	.	.	.
26	011 010	1A	32

Diagram showing arrows from the binary '011 010' and hexadecimal '1A' pointing to the octal '32'.

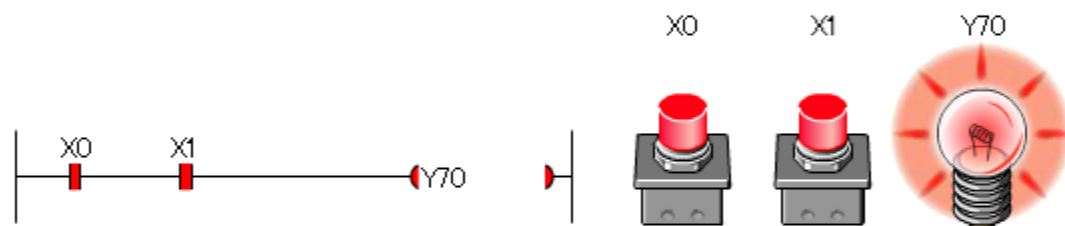
1.3 PLC - podstawy programowania

W przypadku sterowania sekwencyjnego wykonywany jest ciąg operacji w oparciu o sygnały WŁ./WYŁ. otrzymane z urządzeń podłączonych do modułu wejściowego, a następnie wyniki operacji są przekazywane do urządzeń podłączonych do modułu wyjściowego.

W celu przeprowadzenia takiego procesu sterowania, program sterujący musi posiadać warunki wejściowe i sygnały wyjściowe, które zostaną wykonane przy odpowiednich warunkach wejściowych.

Poniższy program przesyła informacje dotyczące następujących operacji:

- Jeśli oba wyłączniki przycisku podłączone do zacisków X1 i X2 znajdują się w stanie WŁ., WŁĄCZ zacisk Y70
- Wynikiem operacji jest wysłanie instrukcji wyjściowej do zacisku Y70, dotyczącej WŁĄCZENIA podłączonej lampki



Jednoczesne naciśnięcie przełączników X0 i X1 powoduje WŁĄCZENIE lampki Y70.

Programy opisane w rozdziale 1.3 używały symboli alfanumerycznych, takich jak X0, X1 i Y70 w celu określenia urządzeń we/wy. Takie znaki alfanumeryczne są nazywane adresem we/wy.

W tym rozdziale opisano adresy i urządzenia we/wy, które są wymagane do stworzenia programów sterujących.

Model MELSEC iQ-R/Q/L/iQ-F i model MELSEC-F używają odmiennych formatów do wyrażania adresów urządzeń. Poniższa tabela przedstawia te różnice.

Model MELSEC	Bit			Słowo
	X (adres wejścia)	Y (adres wyjścia)	M (przełącznik wewnętrzny)	D (rejestr danych)
Model iQ-R/Q/L	Zapis szesnastkowy	Zapis szesnastkowy	Zapis dziesiętny	Zapis dziesiętny
Model iQ-F/F	Zapis ósemkowy	Zapis ósemkowy	Zapis dziesiętny	Zapis dziesiętny

1.4.1 Adresy we/wy i sygnały we/wy (model MELSEC iQ-R/Q/L)

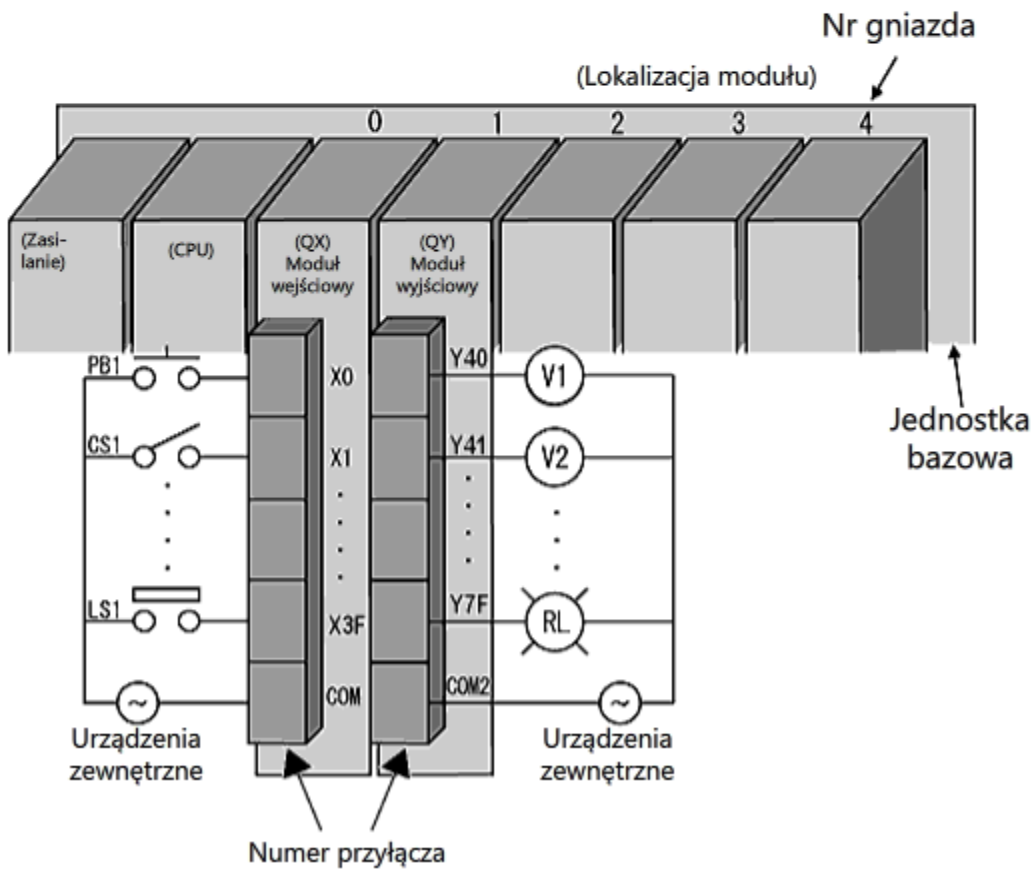
Model MELSEC iQ-R/Q/L

Adres we/wy składa się z litery alfabetu, która oznacza wejście (X) lub wyjście (Y), oraz wartości szesnastkowej oznaczającej numer przyłącza.

Adres we/wy jest najpierw określany na podstawie pozycji instalacyjnej modułu.

Zakres adresów we/wy jest następnie wyznaczany w oparciu o liczbę zajętych punktów we/wy danego modułu. (Liczba zajętych punktów we/wy jest proporcjonalna do liczby przyłączy we/wy modułu.)

Poniższy rysunek przedstawia, w jaki sposób adresy we/wy są przypisywane do 64-punktowego modułu wejściowego i 64-punktowego modułu wyjściowego, które są zainstalowane odpowiednio na gnieździe nr 0 i nr 1.



1.4.2 Adresy we/wy i sygnały we/wy (model MELSEC iQ-F/F)

Model MELSEC iQ-F/F

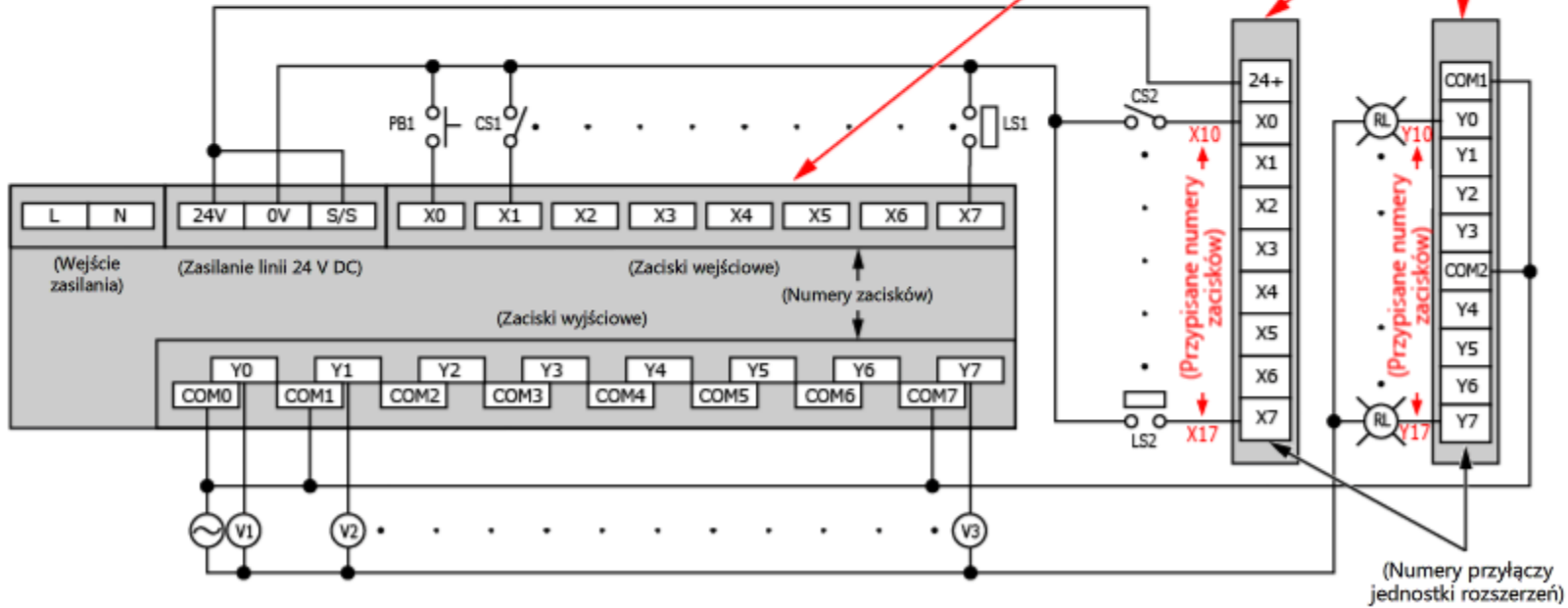
Adres we/wy składa się z litery alfabetu, która oznacza wejście (X) lub wyjście (Y), oraz wartości ósemkowej, oznaczającej numer przyłącza.

Adres we/wy jest najpierw określany na podstawie końcowego adresu we/wy jednostki głównej lub poprzedzającego bloku rozszerzeń we/wy. Zakres adresów we/wy jest następnie wyznaczany w oparciu o liczbę zajętych punktów we/wy danego tej jednostki.

(Liczba zajętych punktów we/wy jest proporcjonalna do liczby punktów we/wy posiadanych przez jednostkę rozszerzeń we/wy.)

Pierwsza cyfra adresu we/wy zawsze zaczyna się od „0” dla nowej jednostki rozszerzeń. Na przykład jeśli adres we/wy poprzedniej jednostki kończy się wartością X7, adres we/wy następnej jednostki zaczyna się od X10.

Poniższy rysunek przedstawia, jak adresy we/wy są przypisywane do 8-punktowej wejściowej jednostki rozszerzeń i 8-punktowej wyjściowej jednostki rozszerzeń, które są dołączane do jednostki głównej modelu MELSEC-F.



1.4.3 Adresy i urządzenia we/wy

Statusy urządzeń podłączonych do jednostki są zapisywane w obszarze pamięci sterownika programowalnego zwanym „rejestr”. Podobnie dla wyjścia, urządzenia wyjściowe działają zgodnie ze statusem tych rejestrów.

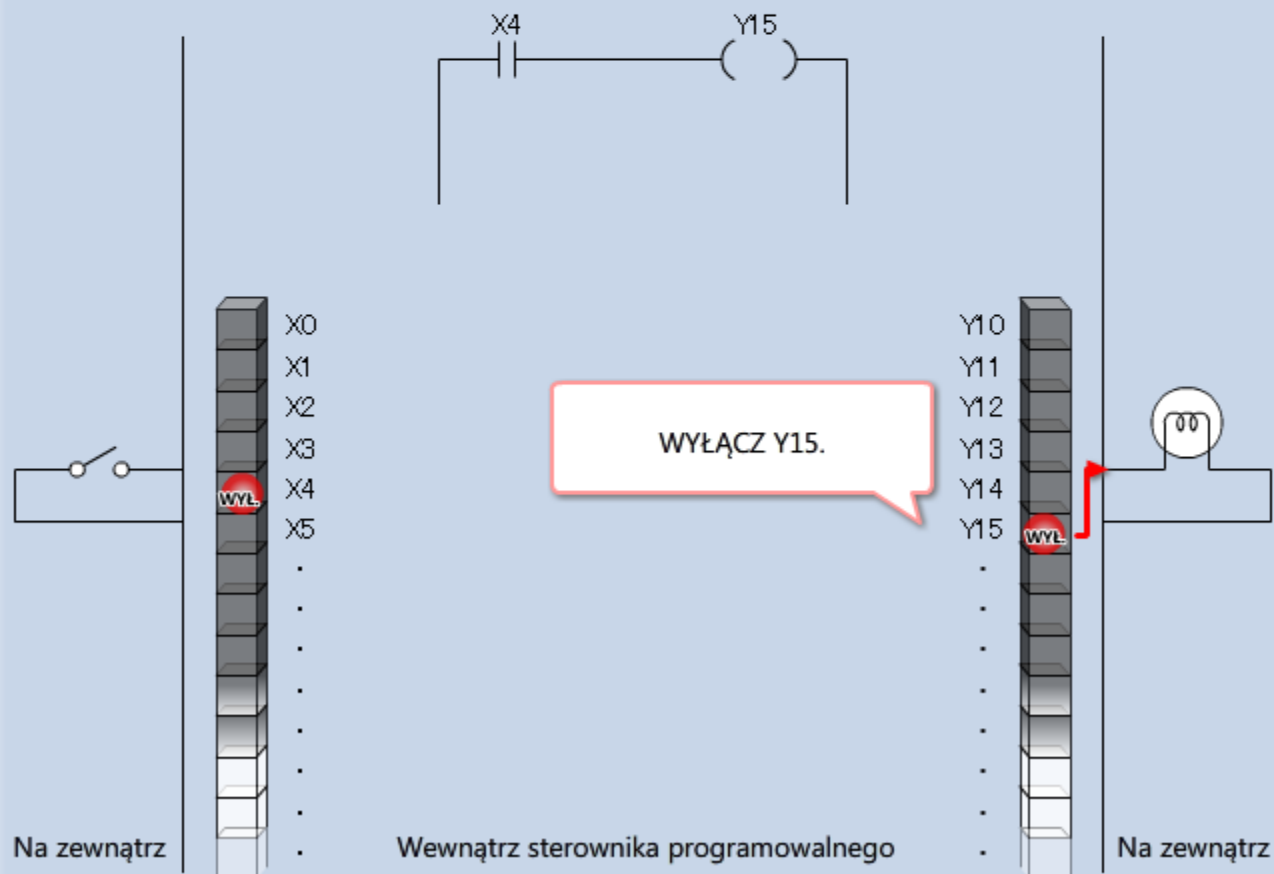
Jak wyjaśniono powyżej, programy sterujące są często wykonywane w oparciu o statusy urządzeń.

Rejestry przechowujące informacje bitowe (Wł./WYł.), takie jak wejście (X) i wyjście (Y), są nazywane „operandami bitowymi”.

Adresy urządzenia odpowiadają numerom we/wy.

Na przykład status przyłącza przypisanego do numeru we/wy X0 jest zapisywany w rejestrze X0.

Analogicznie, status urządzenia Y10 odpowiada zaciskowi przypisanemu do numeru we/wy Y10.



1.4.4

Przełączniki wewnętrzne

Dowiedzieliśmy się, że operandy bitowe, takie jak X (wejście) i Y (wyjście), odpowiadają adresom przypisanym przyłączom we/wy fizycznego modułu.

Istnieje inna grupa operandów bitowych, które nie mają związku z zaciskami we/wy modułu, a jeden z nich jest nazywany „przełącznikiem wewnętrznym (M)”.

Przełączniki wewnętrzne (M) są wyrażane w formacie dziesiętnym, mimo że urządzenia wejściowe (X) i wyjściowe (Y) są przedstawiane w formacie szesnastkowym w przypadku modelu MELSEC iQ-R/Q/L oraz ósemkowym w przypadku modelu MELSEC iQ-F/F.

Przełączniki wewnętrzne (M) są głównie używane do przechowywania tymczasowych danych bitowych.

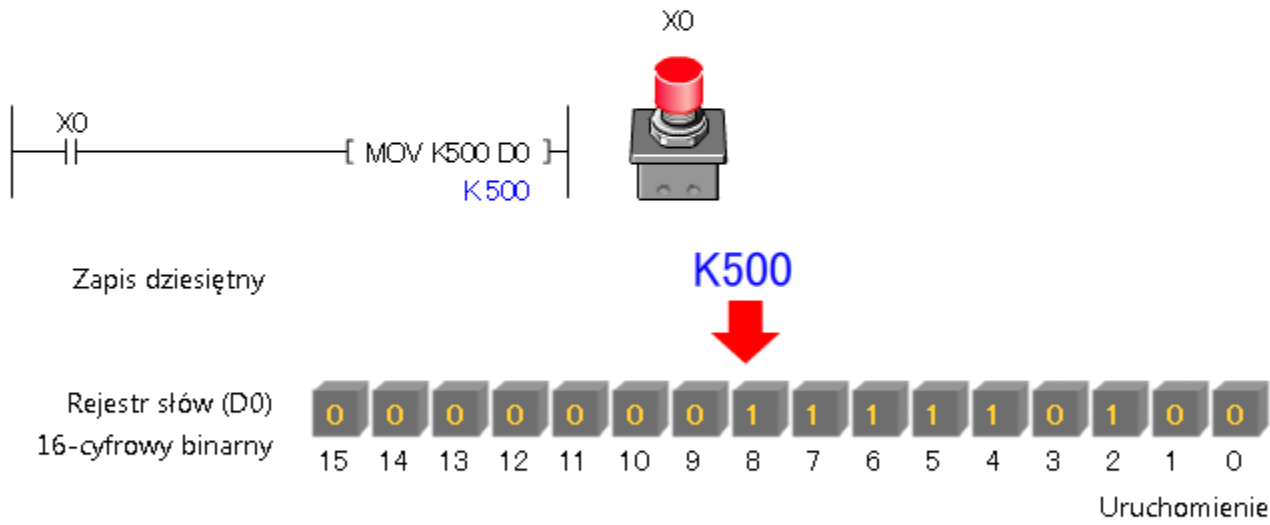
Na przykład przełączniki wewnętrzne (M) można stosować do przechowywania wyniku obliczeniowego operacji, dzięki czemu może być on użyty na innym szczeblu drabinki.

1.4.5 Rejestry

Dowiedzieliśmy się, że urządzenia przechowujące informacje bitowe (WŁ./WYŁ.) są nazywane „operandami bitowymi”, a urządzenia przechowujące pamięć podzieloną na słowa danych są nazywane „rejestrami”.

„Rejestry danych” (D) są jednymi z zazwyczaj stosowanych rejestrów. Jeden rejestr danych (D) może przechowywać 1 słowo (16 bitów) danych.

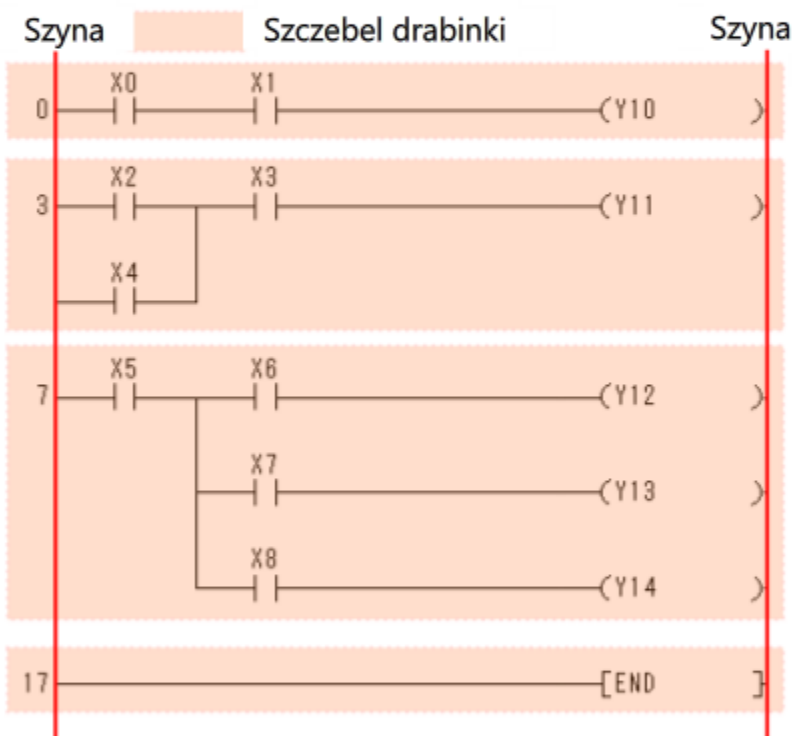
Poniższa animacja przedstawia wykorzystanie rejestrów danych (D).
W tym programie WŁĄCZENIE X0 powoduje zapisanie „K500” (wartości dziesiętnej) w D0.
Instrukcja MOV powoduje skopiowanie wartości do określonego rejestru. (Dalsze szczegóły zostaną wyjaśnione w punkcie 3.1.)
W tym podpunkcie numery są zapisywane w rejestrach danych.



Nawet jeśli przycisk jest zwolniony i X0 zostaje WYŁĄCZONY, utrzymywana jest wartość „K500” zapisana w rejestrze danych D0.

1.5 Tworzenie programów sterujących

Programy sterujące składają się z szyn na prawej i lewej krawędzi oraz symboli instrukcji połączonych z liniami. Obszar pomiędzy instrukcją -| |- połączoną z lewą szyną i instrukcją -()- lub -[]- połączoną z prawą szyną jest nazywany szczelbem drabinki. Kilka takich szczelbli drabinki stanowi program sterujący, który kończy się instrukcją -[END]- lub -[FEND]-.



■ Różnice pomiędzy oprogramowaniem programistycznym

instrukcja -()- oraz -[]- różnią się pomiędzy oprogramowaniem programistycznym GX Works2 and GX Works3. W ramach tego kursu opis stosuje instrukcje GX Works2.

	GX Works2	GX Works3
-()-	(Y10)	Y10
-[]-	{MOV K500 D0 }	

1.5.1 Symbole instrukcji

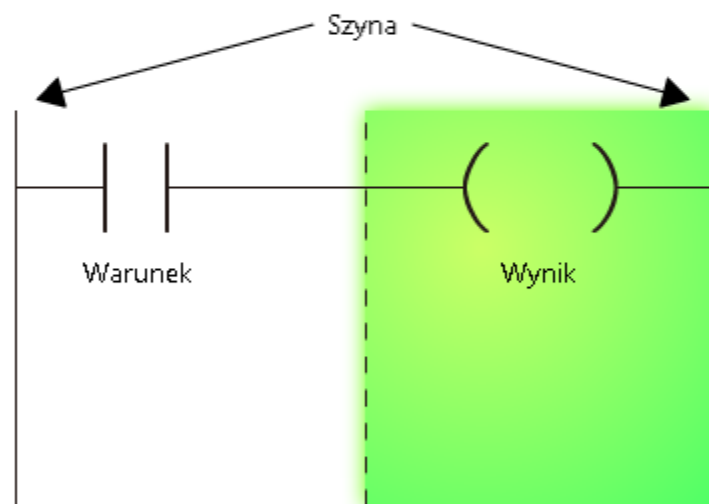
Jak wyjaśniono w punkcie 1.3, sterownik programowalny musi otrzymać instrukcję, aby wykonać pewne operacje, jeśli wstępnie określone warunki wejściowe są spełnione. Dla takich instrukcji, symbole instrukcji są stosowane do wyrażenia warunków wejściowych i szczegółów dotyczących sygnałów wyjściowych.

Symbol instrukcji często zawiera numer urządzenia.

Numer urządzenia określa obszar (urządzenie) przechowujący status, który jest używany w celu oceny warunku lub jako wynik wyjściowy.

Szczelbel drabinki zawiera warunki i wyniki wyjściowe. Warunki są umieszczone po lewej stronie, a wyniki wyjściowe znajdują się po prawej stronie.

Wyniki wyjściowe mogą być zwykłym sygnałem WŁ./WYŁ. lub dedykowaną instrukcją, taką jak operacja obliczenia lub kopiowania.



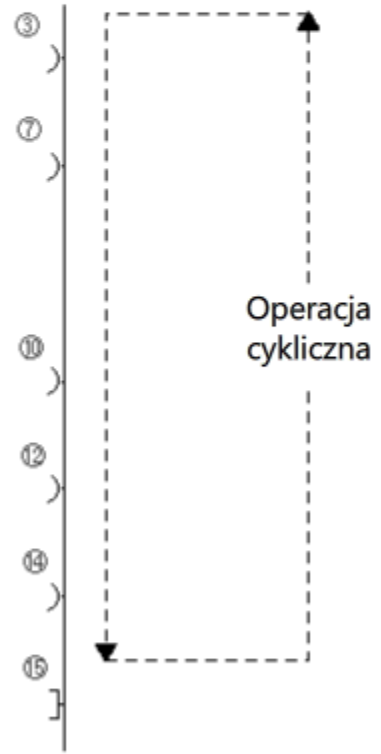
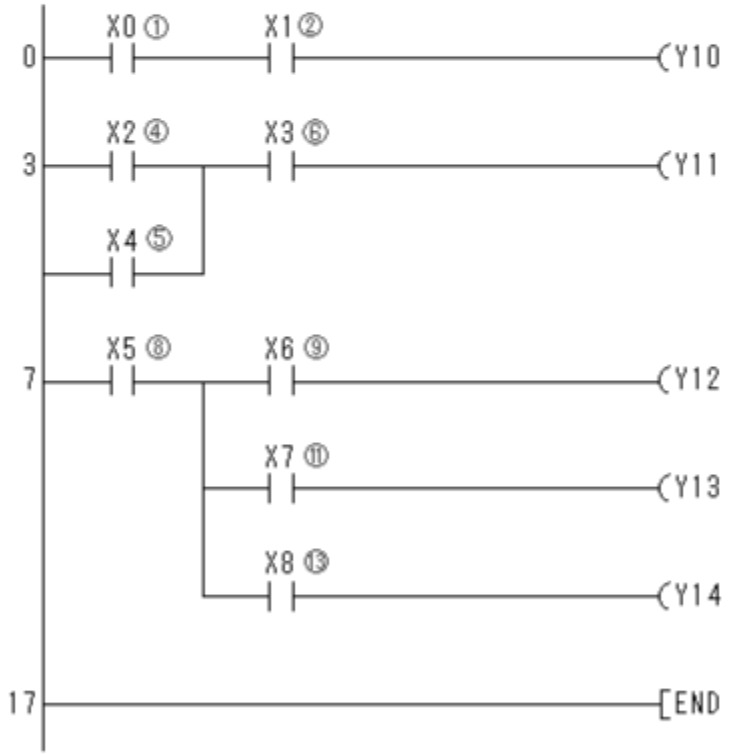
Na schemacie drabinkowym znajdują się dwie równoległe szyny.
Z lewej strony zapisywane są warunki.
Z prawej strony zapisywane są wyniki.

1.6 Procedura wykonywania programu

Program rozpoczyna się od lewej instrukcji rozpoczęcia i kończy się instrukcją -[END]-. Po osiągnięciu instrukcji -[END]- wykonywanie programu rozpoczyna się ponownie od instrukcji rozpoczęcia. To powtarzalne wykonywanie jest nazywane „operacją cykliczną”.

Jeden cykl takiej operacji cyklicznej jest nazywane „skanowaniem”, a przedział czasu, który jest potrzebny do przetworzenia jednego skanowania, jest nazywany „czasem skanowania”.

Poniższy rysunek przedstawia procedurę wykonania instrukcji. Instrukcje są wykonywane od lewej do prawej na każdym szczeblu drabinki, a następnie od górnego szczebla drabinki (nr 1, 2,...15 -> 1...).



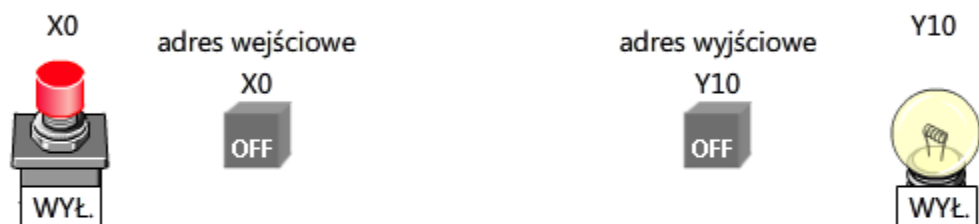
1.7

Czas odświeżania

Jak wyjaśniono wcześniej, czas skanowania to przedział czasu, który jest potrzebny do wykonania jednokrotnego ciągu operacji. Czas skanowania można również wyrazić jako:

$$\text{Czas skanowania} = \text{czas odświeżania} + \text{czas wykonywania programu} + \text{czas przetwarzania END}$$

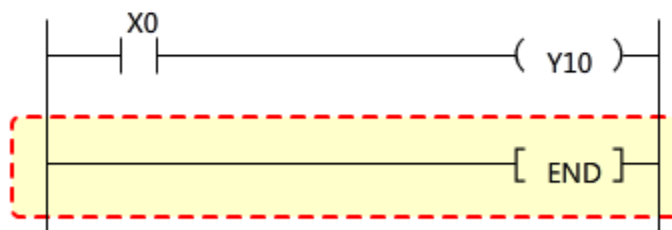
Czas odświeżania to przedział czasu, który jest potrzebny do odczytania danych z modułu wejściowego przez adresy wejściowe (X) plus przedział czasu, który jest potrzebny do zapisania danych w module wyjściowym pochodzących z adresów wyjściowych (Y).



(1) Odświeżanie we/wy

- Przesyłany jest status WŁ./WYŁ. adresów wyjściowych do podłączonych urządzeń wyjściowych
- Zapisywany jest status WŁ./WYŁ. otrzymany z podłączonych urządzeń wejściowych w adresach wejściowych

(2) Wykonywanie programu



(3) Przetwarzanie END

Przetwarzanie END przez instrukcję END
(Pominięto tutaj szczegóły).

Należy zauważyć, że statusy WŁ./WYŁ. bieżącego przełącznika są natychmiast odczytywane i zapisywane w adresach wejściowych (X), a nowe dane nadpisują istniejące wartości podczas odświeżania. Analogicznie, dane w adresach wyjściowych (Y) są natychmiast zapisywane w module wyjściowym podczas wykonywania instrukcji.

Oznacza to, że jeśli sygnał przełącza się z WYŁ. na WŁ., a następnie ponownie na WYŁ., sygnał nigdy nie zostanie rozpoznany jako WŁ. Jednak czas skanowania jest bardzo krótki w porównaniu do długości sygnału. Rzadko zdarza się, aby sterownik programowalny ominął zmianę statusu sygnału.

Rozdział 2 Operacje na bitach

W tym rozdziale opisano operacje na bitach(Wł./WYł.).

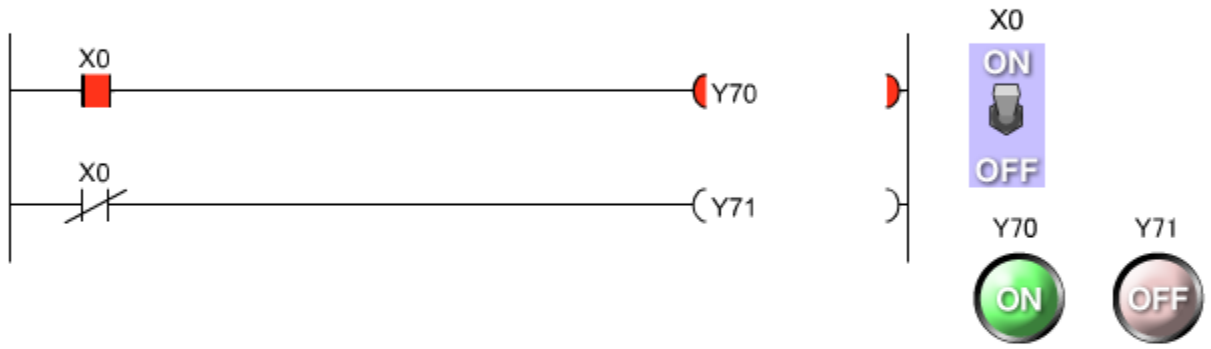
Operacje, które używają operandów bitowych, są najbardziej podstawowymi operacjami w programach sterujących. Wartości wejściowe z urządzeń wejściowych są używane jako warunki do sterowania urządzeniami wyjściowymi.

2.1 Warunki wejściowe i sygnały wyjściowe

Styki normalnie otwarte (NO) i normalnie zamknięte (NC) są używane jako warunki wejściowe.
 Gdy warunki wejściowe są spełnione, instrukcja wyjściowa cewki (instrukcja OUT) jest sygnałem wyjściowym.
 Gdy warunki wejściowe nie są spełnione, instrukcja wyjściowa cewki nie jest sygnałem wyjściowym.
 Instrukcja styku NO/NC i instrukcja OUT są główną kombinacją instrukcji używaną w programach sterujących.

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania instrukcji NO, NC i OUT, klikając przełącznik wejściowy znajdujący się po prawej stronie.

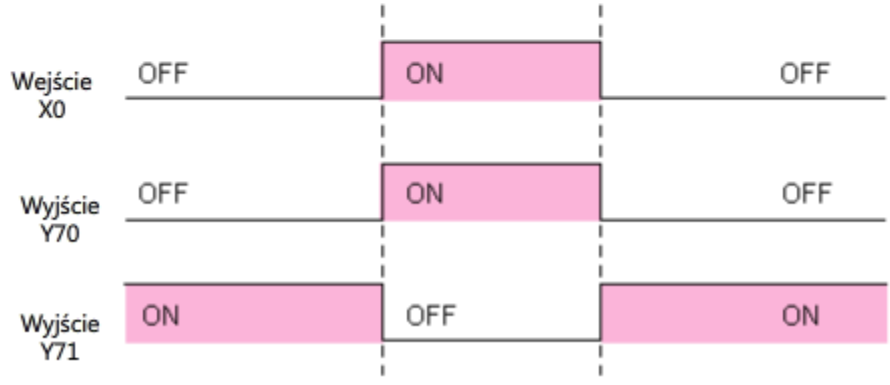


2.1 Warunki wejściowe i sygnały wyjściowe

■ Kody i funkcje instrukcji

Symbol	Funkcja
	<p>Styk NO Realizowany, jeśli status urządzenia jest WŁ.</p>
	<p>Styk NC Realizowany, jeśli status urządzenia jest WYŁ. (odwrotnie do styku NO).</p>
	<p>Wyjście cewki (OUT) Jeśli poprzedni warunek wejściowy jest spełniony, dane w bieżącym urządzeniu są sygnałem wyjściowym.</p>
	<p>Instrukcja końcowa (END) Oznacza koniec programu. Program wymaga instrukcji END.</p>

■ Wykres przebiegu czasowego

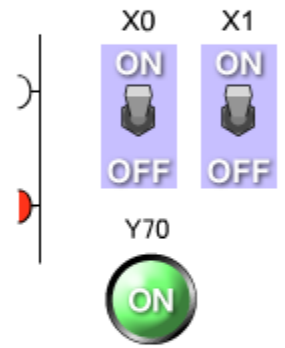
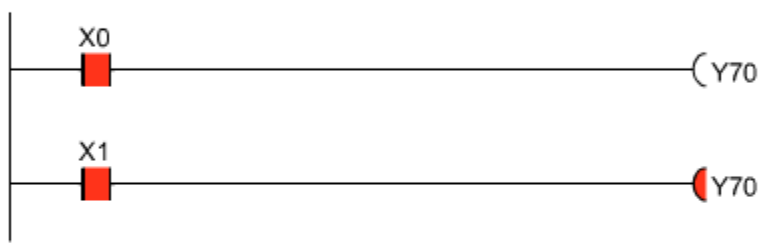


2.1.1 Użycie tego samego numeru urządzenia dla instrukcji

Na szczeblu drabinki tylko jedna instrukcja OUT może zostać użyta z jednym numerem urządzenia. Jeśli więcej niż jedna instrukcja OUT zostanie użyta z tym samym numerem urządzenia, jedynie ostatnia instrukcja OUT będzie obowiązywać, a pierwsza instrukcja OUT stanie się nieważna.

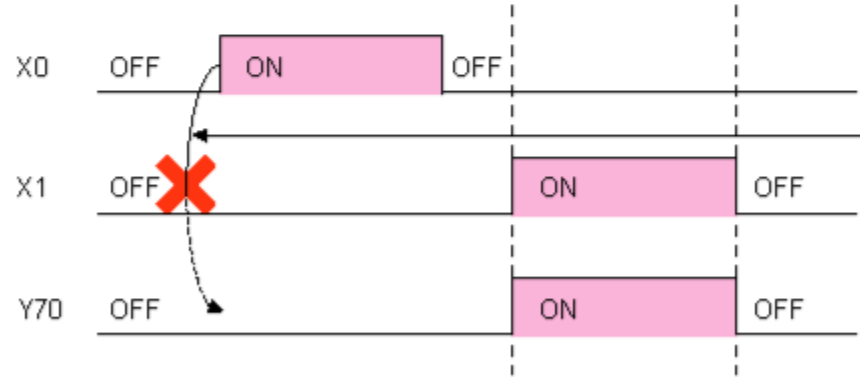
Program drabinkowy

Wykonaj symulację działania dwóch instrukcji posiadających ten sam adres, klikając przełącznik wejściowy znajdujący się po prawej stronie. Ten typ wykorzystania (używający OUT Y70 dla dwóch instrukcji) jest nazywany „zdublowaną cewką”.



Jeśli X1 = Wł., Y70 = Wł.

Wykres przebiegu czasowego



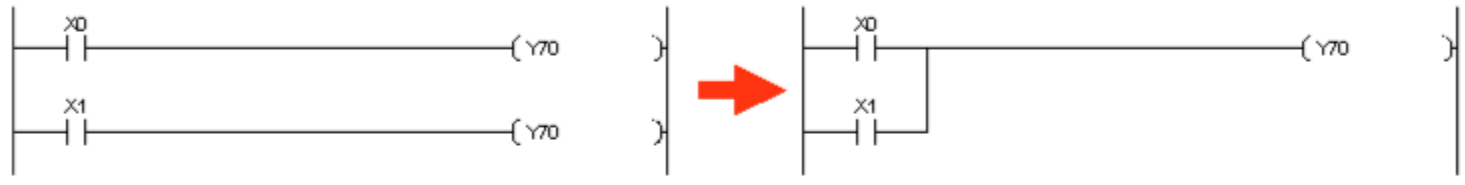
Pierwszy warunek wejściowy X0 jest ignorowany, ponieważ ostatni warunek wejściowy ma priorytet.

2.1.1 Użycie tego samego adresu dla instrukcji

Przykład korekty

W tym przykładzie warunek wejściowy „X1” ma wyższy priorytet, a „X0” jest ignorowany.

Korygując szczelbel drabinki na przedstawiony na rys. B, narzędzie Y70 zostaje WŁĄCZONE, gdy jeden z warunków wejściowych jest spełniony, zapobiegając kolizji pomiędzy dwoma instrukcjami OUT.



(Rys. A)

(Rys. B)

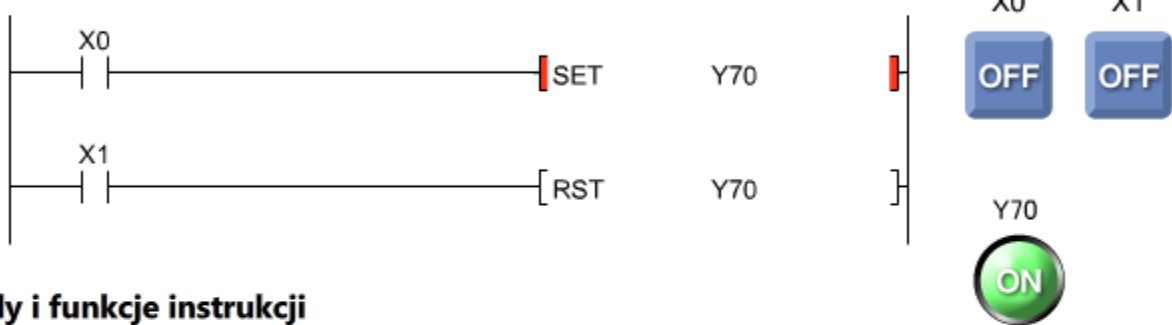
2.2 Zachowanie/kasowanie sygnałów wyjściowych

W odróżnieniu od instrukcji OUT, instrukcja przechowywania operacji (instrukcja SET) powoduje utrzymanie statusu wyjścia, nawet jeśli warunek wejściowy stanie się niespełniony.

Aby anulować wyjście (WY), należy wykonać instrukcję anulowania przechowywania operacji (instrukcję RST).

Program drabinkowy i operacja drabinkowa

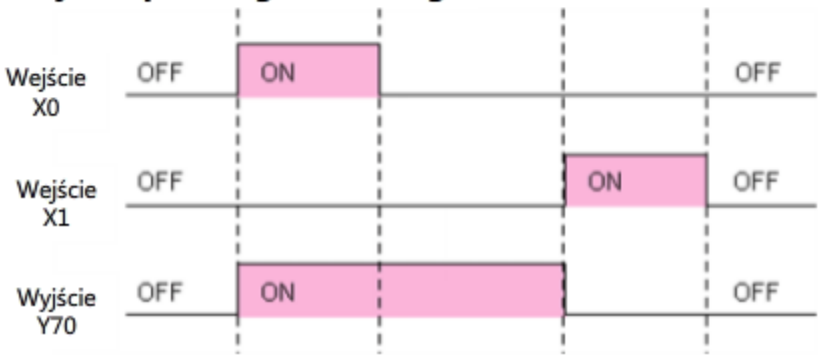
Wykonaj symulację działania instrukcji SET i RST, klikając przełączniki wejściowe znajdujące się po prawej stronie.



Kody i funkcje instrukcji

Symbol	Funkcja
	<p>Instrukcja przechowywania operacji (SET) Powoduje WŁĄCZENIE urządzenia i utrzymanie statusu Wł. (wyjście). Wyjście jest utrzymywane, nawet jeśli warunek wejściowy stanie się niespełniony.</p>
	<p>Instrukcja anulowania przechowywania operacji (RSET) Powoduje anulowanie statusu Wł. i anulowanie sygnału wyjściowego dla określonego urządzenia.</p>

Wykres przebiegu czasowego



2.2.1 Różnice pomiędzy instrukcjami OUT i SET

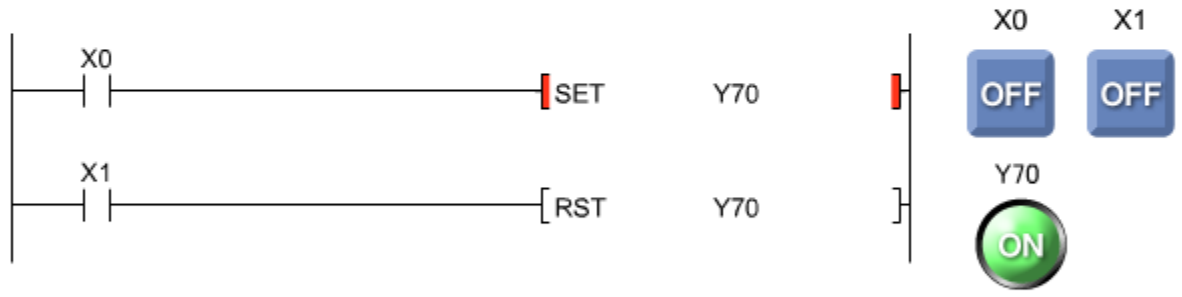
Wykonaj symulację różnic operacyjnych pomiędzy instrukcjami OUT i SET, klikając przełączniki wejściowe znajdujące się po prawej stronie.

■ Instrukcja OUT



Y70 jest Wł., jeśli warunek wejściowy jest spełniony.

■ Instrukcje SET/RST

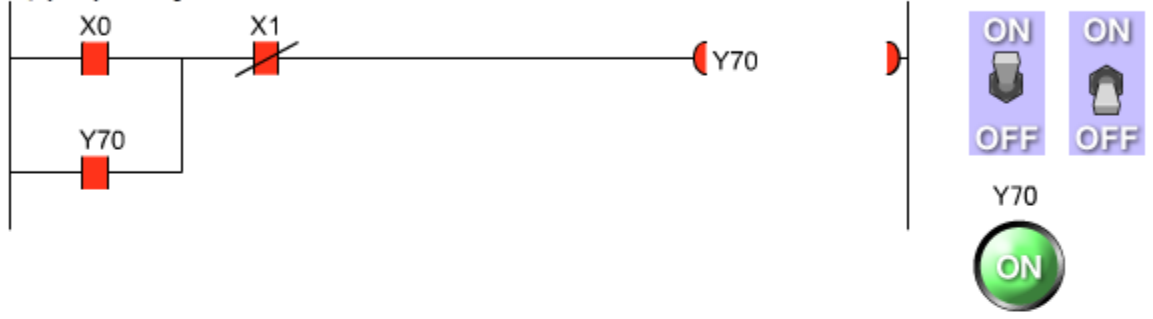


Jeśli warunek wejściowy jest spełniony, Y70 jest Wł. do momentu wykonania instrukcji RST.

2.2.2 Zamiana drabinek przechowywania na instrukcję SET

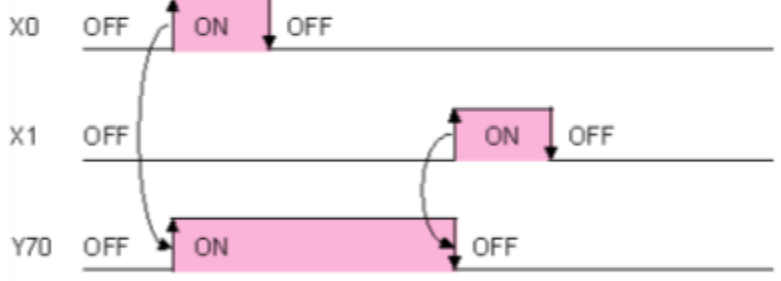
Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania drabinki przechowywania, klikając przełączniki wejściowe znajdujące się po prawej stronie.



Jeśli X0 = Wł. i X1 = WYł., Y70 = Wł.
Y70 = Wł. (przechowywanie) do momentu wystąpienia warunku X1 = Wł.

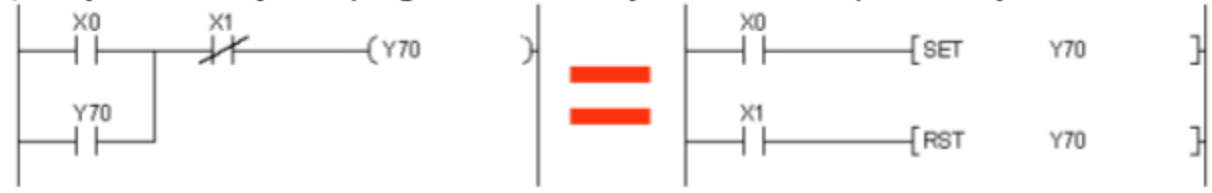
Wykres przebiegu czasowego



Nawet po tym, jak X10 przejdzie w stan WYł., Y70 (cewka) jest Wł. (przechowywanie)

Zamiana na instrukcję SET

Program drabinki przechowywania można przepisać na program drabinkowy z instrukcją SET. Dzięki użyciu instrukcji SET, program drabinkowy może zostać uproszczony.



2.3 Dołączanie warunków (logika AND)

Aby uzyskać logikę AND, styki NO/NC są umieszczone szeregowo.
W logice AND warunek ten jest spełniony, gdy więcej elementów niż styki NO/NC połączone szeregowo jest Wł.

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania logiki AND, klikając przełączniki wejściowe znajdujące się po prawej stronie.



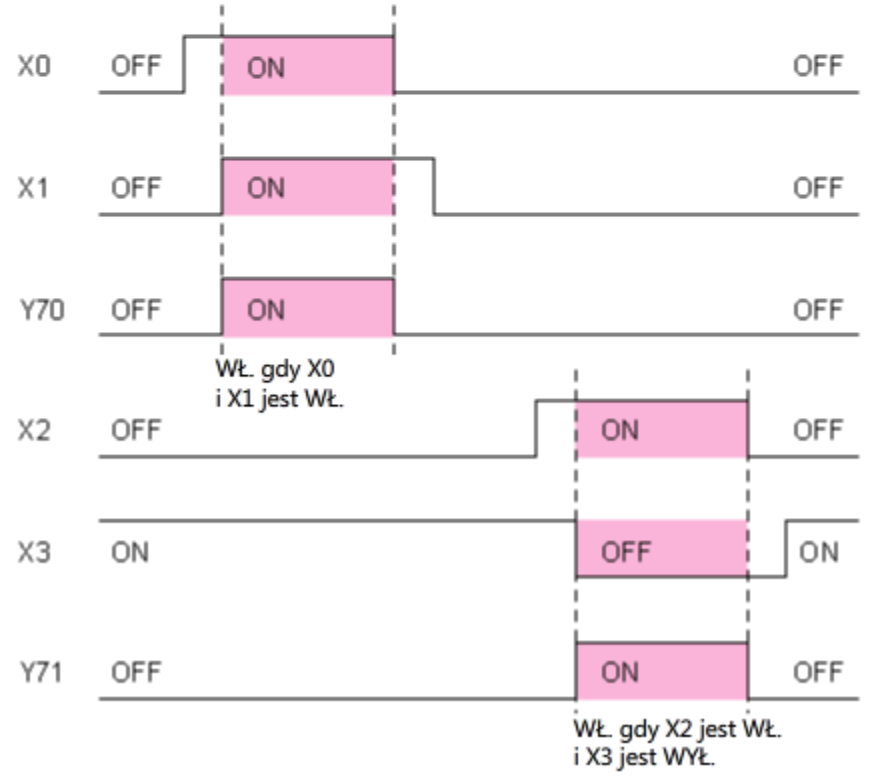
Jeśli X0 i X1 są Wł., Y70 jest Wł.
Jeśli X2 jest Wł. i X3 jest WYł., Y71 jest Wł.

2.3 Dołączanie warunków (logika AND)

Kody i funkcje instrukcji

Symbol	Funkcja
	Połączenie szeregowe styku NO Styk NO jest podłączony szeregowo (poziomo).
	Połączenie szeregowe styku NC Styk NC jest podłączony szeregowo (poziomo).

Wykres przebiegu czasowego



2.4 Dołączanie warunków (logika OR)

Aby uzyskać logikę OR, styki NO/NC są umieszczone równolegle.
W logice OR warunek ten jest spełniony, gdy jeden ze styków NO/NC połączonych równolegle jest Wł.

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania logiki OR, klikając przełączniki wejściowe znajdujące się po prawej stronie.



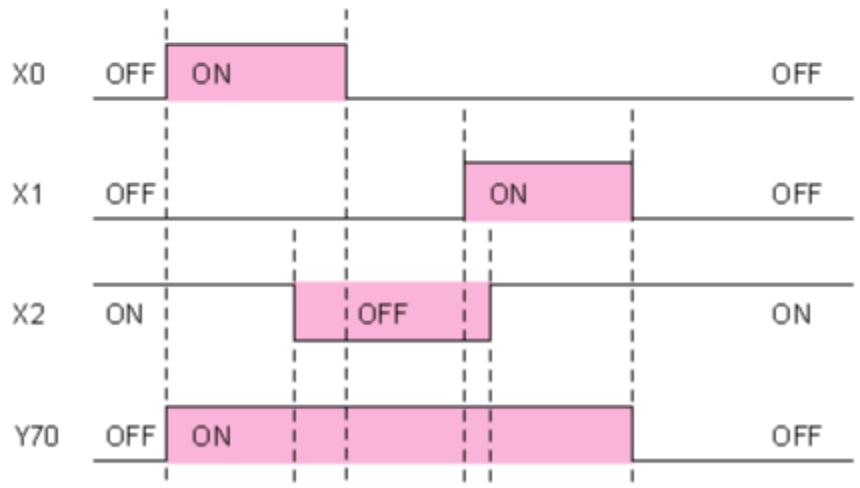
Y70 jest Wł., jeśli którykolwiek z poniższych warunków jest spełniony: X0 jest Wł., X1 jest Wł. lub X2 jest WYł.

2.4 Dołączanie warunków (logika OR)

■ Kody i funkcje instrukcji

Symbol	Funkcja
	Połączenie równoległe styku NO Styk NO jest podłączony równoległe (pionowo).
	Połączenie równoległe styku NC Styk NC jest podłączony równoległe (pionowo).

■ Wykres przebiegu czasowego



2.5 Sygnały wyjściowe jako impulsy

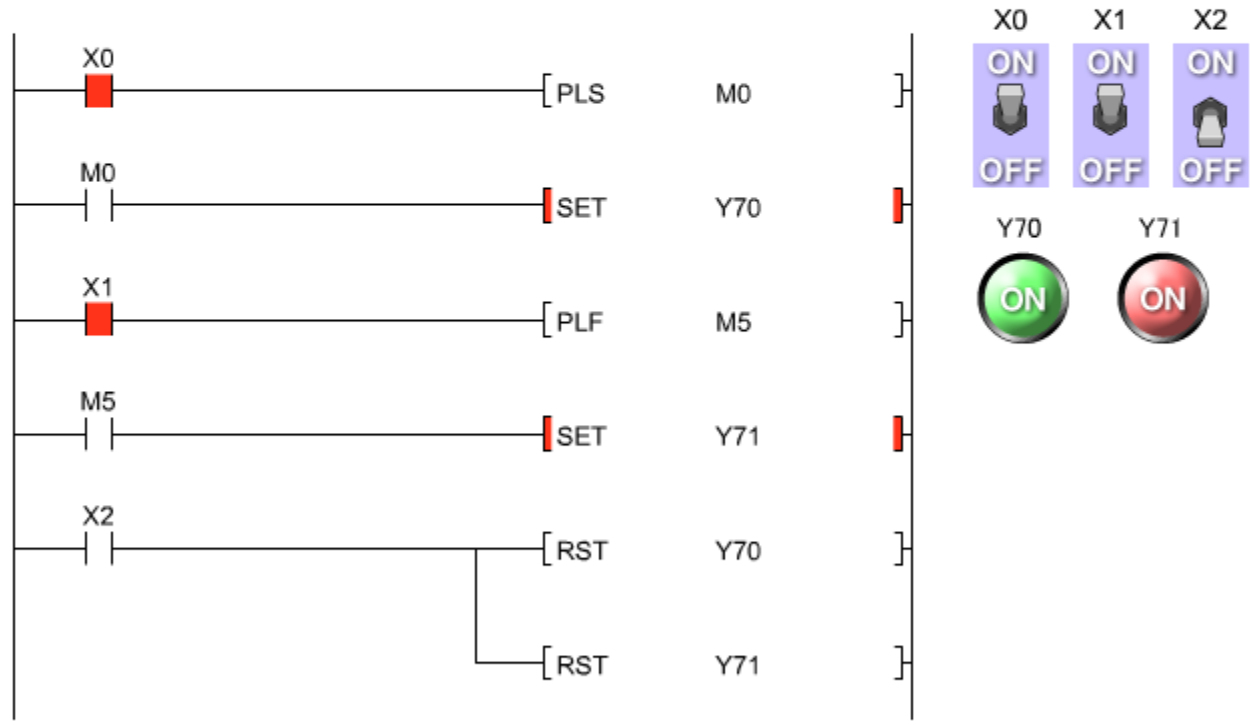
W odróżnieniu od instrukcji OUT, instrukcja zbocza narastającego(instrukcja PLS) powoduje WŁĄCZENIE cewki na jedno skanowanie po spełnieniu warunku wejściowego.

W przeciwieństwie do instrukcji PLS, instrukcja zbocza opadającego(instrukcja PLF) powoduje WŁĄCZENIE cewki na jedno skanowanie po niespełnieniu warunku wejściowego.

Cewka WŁĄCZONA przez instrukcję PLS/PLF wraca do stanu WYŁ. po jednym skanowaniu.

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania instrukcji PLS i PLF, klikając przełączniki wejściowe znajdujące się po prawej stronie.

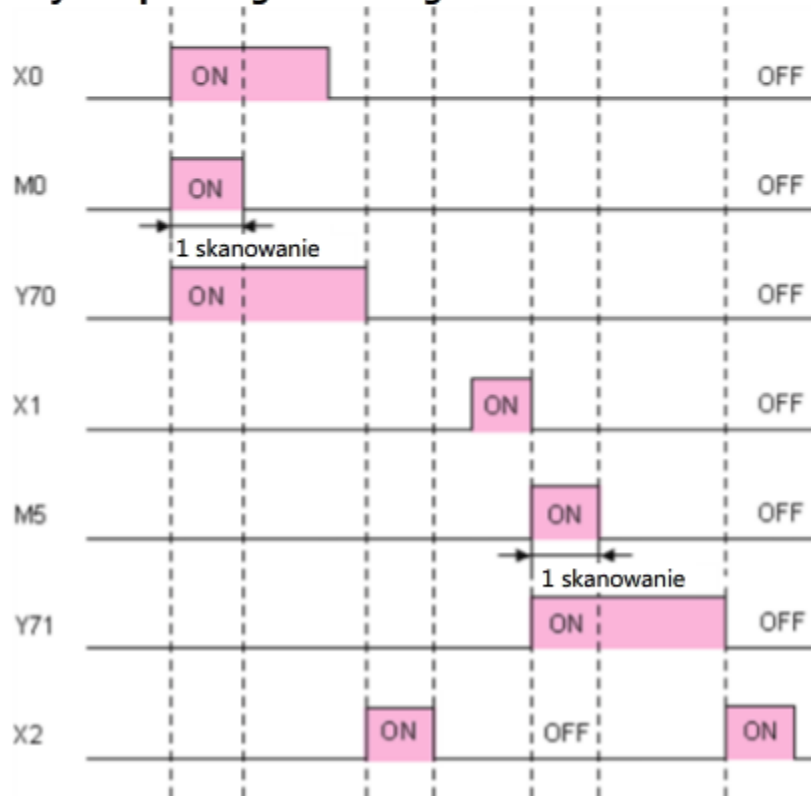


Przy wystąpieniu krawędzi rosnącej X0 (WYŁ. na WŁ.), M0 zostaje WŁĄCZONE na 1 skanowanie
Przy wystąpieniu krawędzi malejącej X1 (WŁ. na WYŁ.), M5 zostaje WŁĄCZONE na 1 skanowanie

Kody i funkcje instrukcji

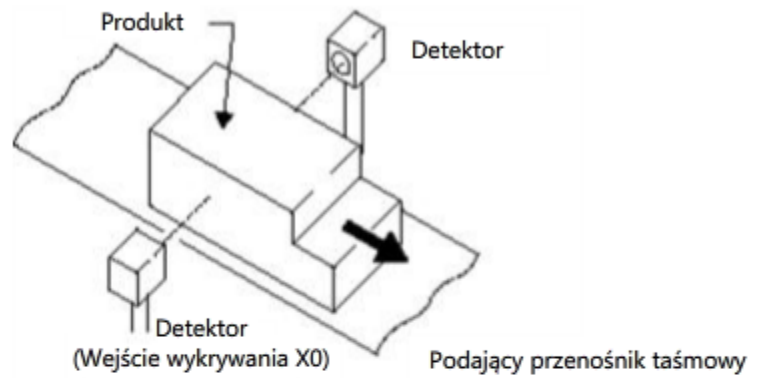
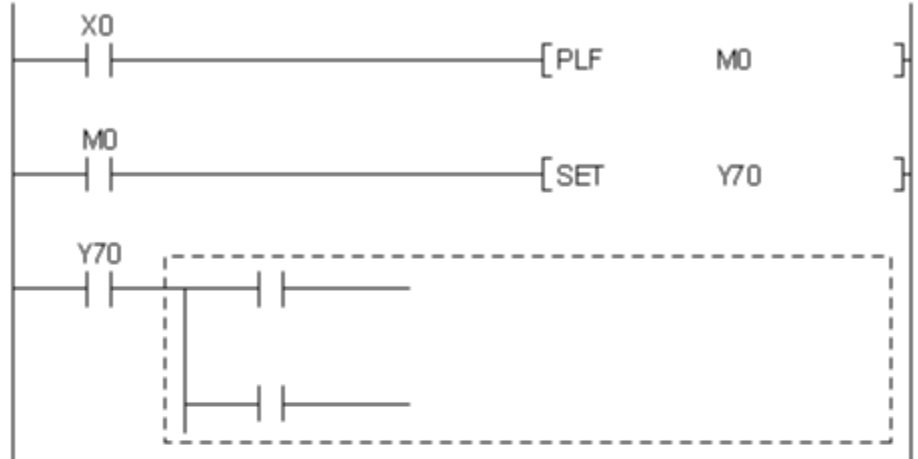
Symbol	Funkcja
<p>The symbol shows a normally open contact on the left, followed by a coil with a red 'P' and 'L' and a red 'S' in a square box on the right.</p>	<p>Wyjście w przypadku zbocza narastającego(PLS) Dane są wysyłane do określonego urządzenia przy pierwszym skanowaniu po spełnieniu warunku wejściowego.</p>
<p>The symbol shows a normally open contact on the left, followed by a coil with a red 'P' and 'L' and a red 'F' in a square box on the right.</p>	<p>Wyjście w przypadku zbocza opadającego(PLF) Dane są wysyłane do określonego urządzenia przy pierwszym skanowaniu po niespełnieniu warunku wejściowego.</p>

Wykres przebiegu czasowego



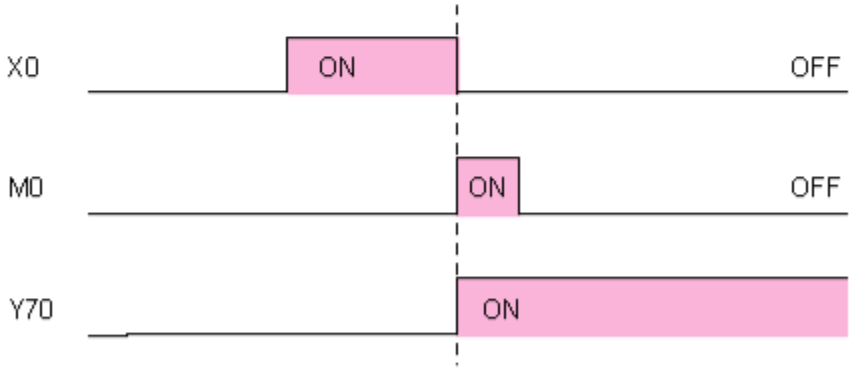
2.5.1 Przykład zastosowania wyjść impulsowych

Program drabinkowy



Wyjście impulsowe jest używane do wykrywania przejścia poruszającego się obiektu. Jeśli zostanie wykryte przejście produktu, inicjowany jest następny proces.

Wykres przebiegu czasowego

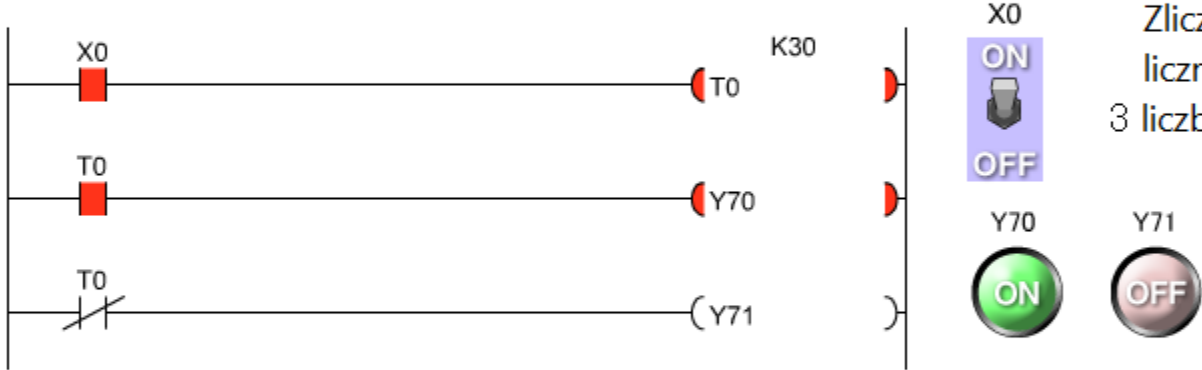


2.6 Pomiar czasu

Instrukcja OUT i urządzenie licznika czasu (T) są używane do pomiaru czasu.
 Jeśli warunek wejściowy jest spełniony (WŁ.), rozpoczyna się pomiar czasu. Gdy przedział czasu osiąga określoną wartość, urządzenie licznika czasu (T) zostaje WŁĄCZONE.
 Jeśli warunek wejściowy jest niespełniony (WYŁ.) lub urządzenie licznika czasu (T) zostanie zresetowane za pomocą instrukcji RST, dotychczasowy czas i wyjście zostają zainicjowane.
 Status urządzenia licznika czasu (T) można wykorzystać jako warunek wejściowy w innych częściach programu.

Program drabinkowy i operacja drabinkowa

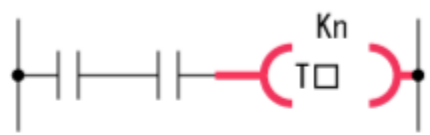
Wykonaj symulację licznika czasu, klikając przełącznik wejściowy znajdujący się po prawej stronie.



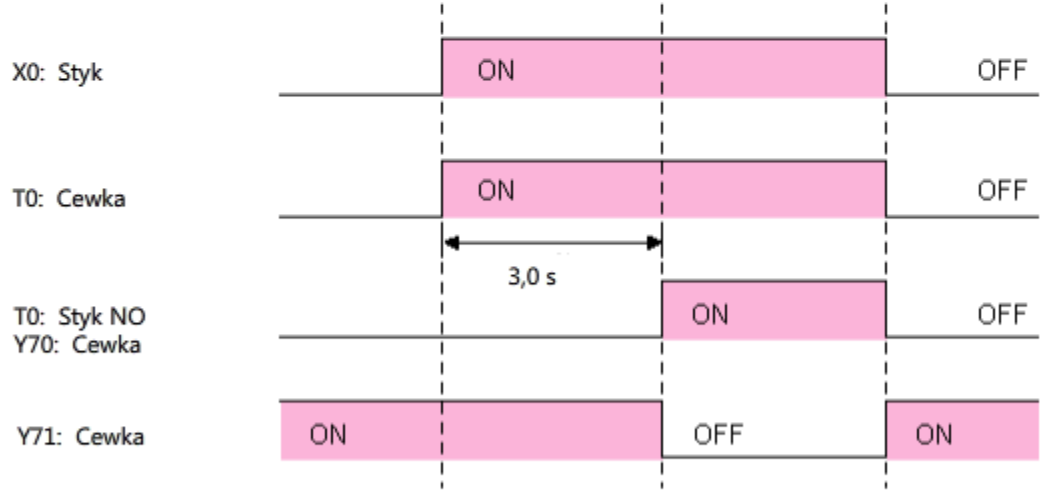
X0 zostaje WŁĄCZONY, następnie po 3 sekundach Y70 zostaje WŁĄCZONY, a Y71 zostaje WYŁĄCZONY.

2.6 Pomiar czasu

Kod i funkcja instrukcji

Symbol	Funkcja
 <p>□ : Numer urządzenia</p>	<p>Działanie licznika czasu</p> <p>Urządzenie licznika czasu (T) jest używane z wyjściem cewki (OUT) do pomiaru trwania spełnienia warunku (stanu Wł.).</p> <p>Przekroczenie dozwolonego czasu ma miejsce po upływie określonego przedziału czasu. Jednocześnie z przekroczeniem dozwolonego czasu licznik czasu (T0) zostaje Wł.</p> <p>Ustawiona wartość licznika czasu jest określana przez „Kn” (zapis dziesiętny).</p> <p>Liczniki czasu są często stosowane jako timer z opóźnionym załączeniem, który określa czas, po upływie którego określony warunek jest spełniony.</p>

Wykres przebiegu czasowego

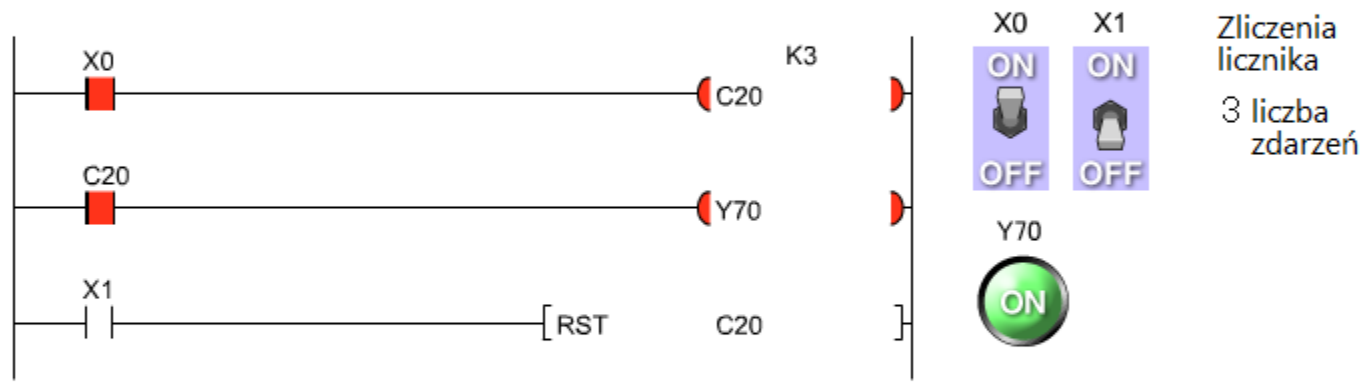


2.7 Zliczanie

Instrukcja OUT i adres licznika (C) są używane do pomiaru zliczeń.
 Jeśli warunek wejściowy jest spełniony, liczba zliczeń wzrasta, a jeśli liczba zliczeń osiąga zadaną wartość, określony adres licznika (C) zostaje WŁĄCZONY.
 Jeśli adres licznika (C) jest resetowany za pomocą instrukcji RST, inicjowane jest liczenie i status adresu.
 Status adresu licznika (C) można wykorzystać jako warunek wejściowy w innych częściach programu.

Program drabinkowy i operacja drabinkowa


Wykonaj symulację licznika, klikając przełącznik wejściowy znajdujący się po prawej stronie.



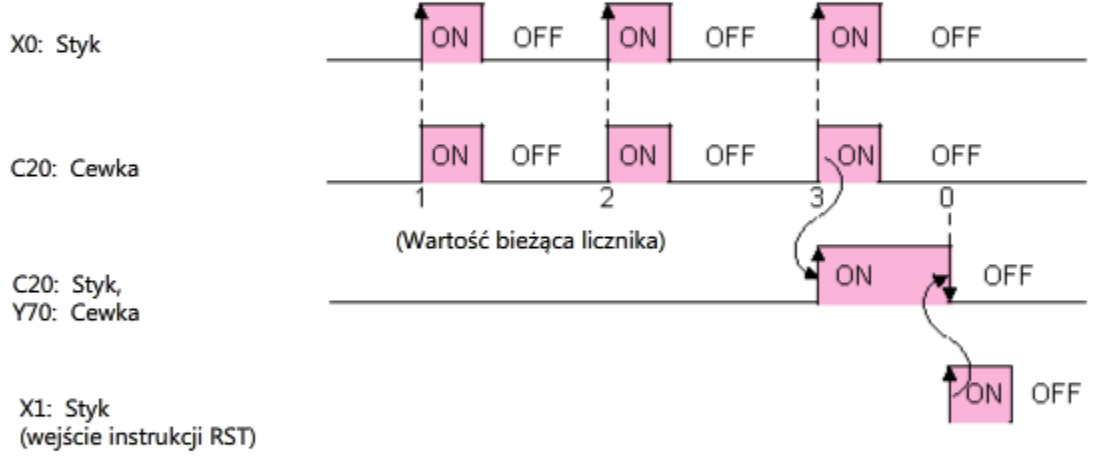
Wartość w C20 jest zwiększana za każdym razem, gdy X0 zostaje WŁĄCZONY. Gdy wartość wynosi 3 (zakończenie liczenia), Y70 zostaje WŁĄCZONY.

2.7 Zliczanie

Kod i funkcja instrukcji

Symbol	Funkcja
 <p>□ : Numer urządzenia</p>	<p>Licznik</p> <p>W połączeniu z wyjściem cewki (OUT) licznik zlicza (pojedynczo), ile razy warunek został spełniony. Zakończenie liczenia ma miejsce, gdy liczba zliczeń osiąga zadaną wartość, a styk licznika zostaje WŁĄCZONY.</p> <p>Ustawiona wartość licznika czasu jest określana przez „Kn” (zapis dziesiętny).</p>

Wykres przebiegu czasowego



Rozdział 3 Instrukcje na rejestrach

W tym rozdziale opisano instrukcje korzystające z rejestrów.

Rejestry są przydatne w kontrolowaniu czasu, zliczeń i wartości przychodzących z urządzeń zewnętrznych.

Użycie rejestrów może spowodować lepszą reakcję programów sterujących na rzeczywiste działanie.

- Wykonaj symulację działań programu, aby zrozumieć działanie głównych instrukcji
- Dzięki tej symulacji możesz zrozumieć rolę instrukcji i procesu przetwarzania wykonywanych w sterowniku programowalnym

3.1 Przenoszenie danych do rejestru

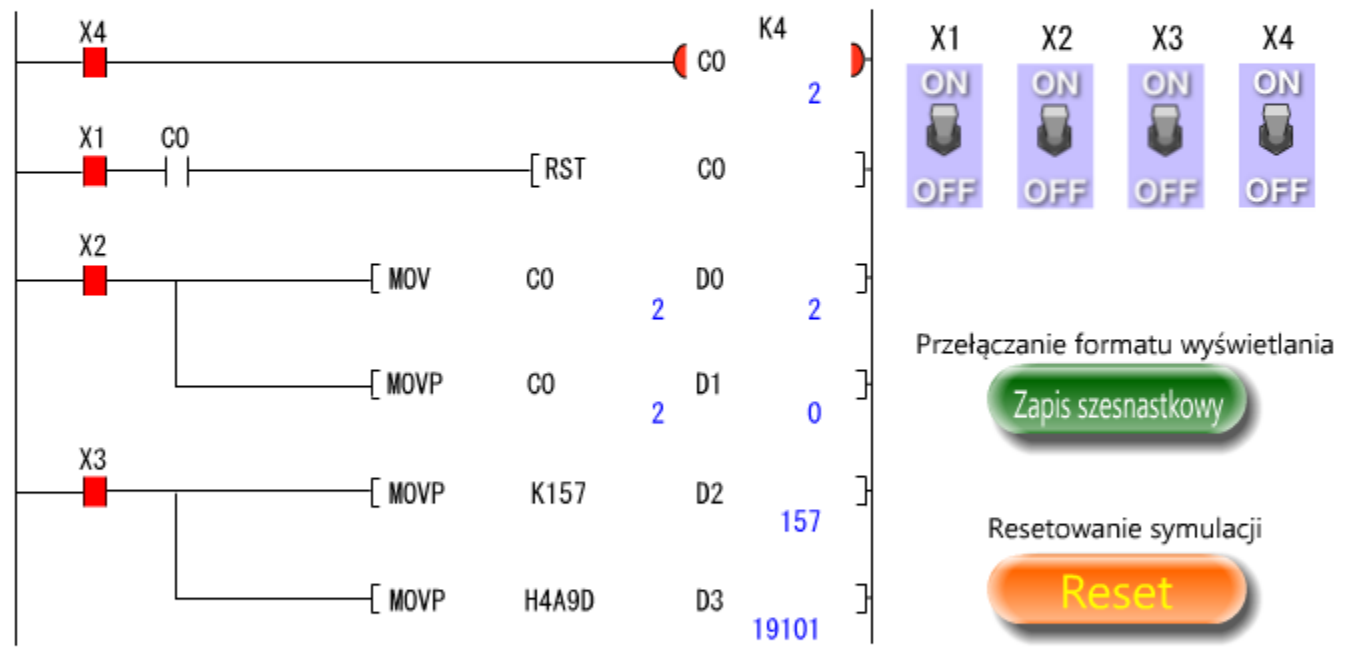
Instrukcja przesyłania danych 16-bitowych (MOV) powoduje przeniesienie (skopiowanie) jednostkowego elementu danych złożonego z 1 słowa (16-bitowego) do określonego rejestru.

Przenoszone dane mogą być wartością w urządzeniu lub mogą zostać zadane. Format przenoszonych danych może być dziesiętny lub szesnastkowy.

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania poniższych instrukcji, klikając przełączniki wejściowe znajdujące się po prawej stronie.

Każda liczba w kolorze niebieskim oznacza wartość (bieżącą wartość) przechowywaną w pamięci.



Gdy wielokrotnie WŁĄCZASZ/WYŁĄCZASZ X4, zwiększa się bieżąca wartość C0 (0, 1...4->0...).

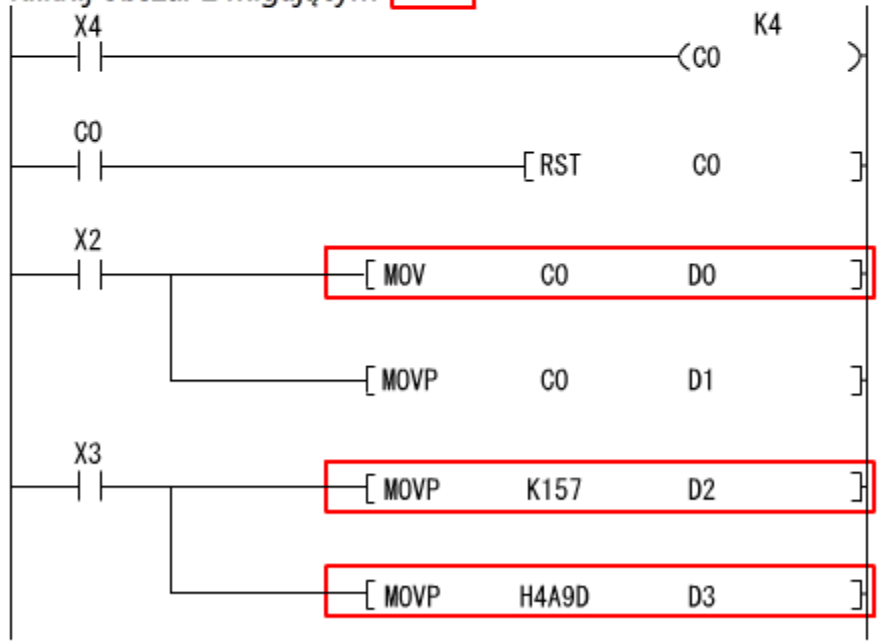
3.1 Przenoszenie danych do rejestru

Kody i funkcje instrukcji

Symbol	Funkcja
	Przesyłanie danych 16-bitowych (MOV) Gdy warunek wejściowy jest spełniony, dane określone w źródle (S) są przesyłanie (kopiowane) do pamięci określonej w miejscu przeznaczenia (D).
	Przesyłanie danych 16-bitowych (impulsowe) (MOVVP) Przy wystąpieniu krawędzi rosnącej warunku (WYŁ. lub WŁ.) dane określone w źródle (S) są przesyłanie (kopiowane) do pamięci określonej w miejscu przeznaczenia (D).

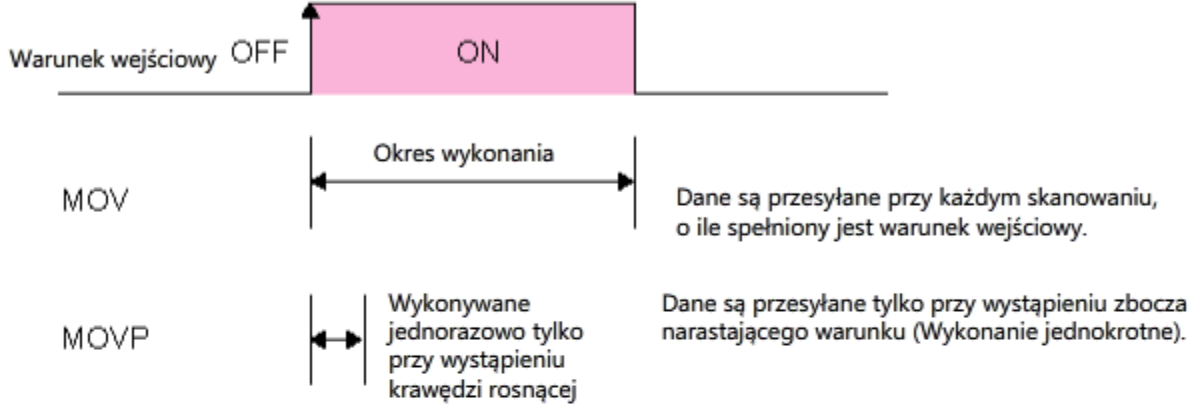
Program drabinkowy

Kliknij obszar z migającym

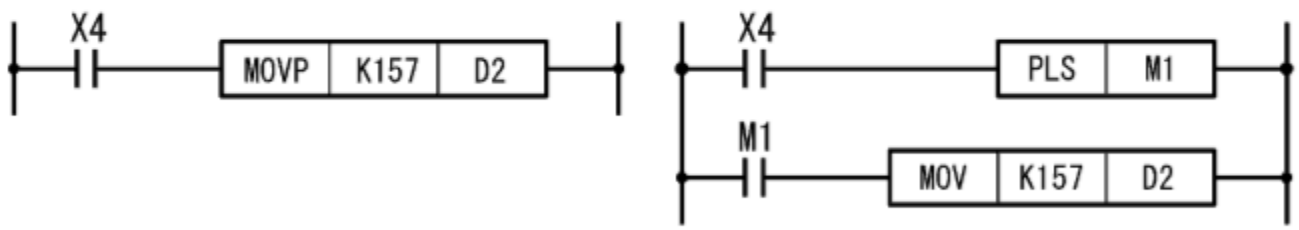


3.1.1 Różnica pomiędzy MOV i MOVP

Instrukcja MOV jest używana do ciągłego odczytywania zmieniających się danych. Jednocześnie instrukcja MOVP jest używana do jednorazowej transmisji danych, takiej jak ustawienie danych lub odczytanie danych w przypadku wystąpienia błędu.



Poniższe rysunki przedstawiają dwa programy, których wynikiem jest taka sama operacja, z instrukcjami MOV i MOVP. Na obu szczeblach drabinki przesyłanie danych jest wykonywane, gdy element X4 zostaje WŁĄCZONY. Dzięki instrukcji MOVP można wykonać działanie bez użycia instrukcji PLS, która określa wykonanie operacji przy wystąpieniu zbocza narastającego.



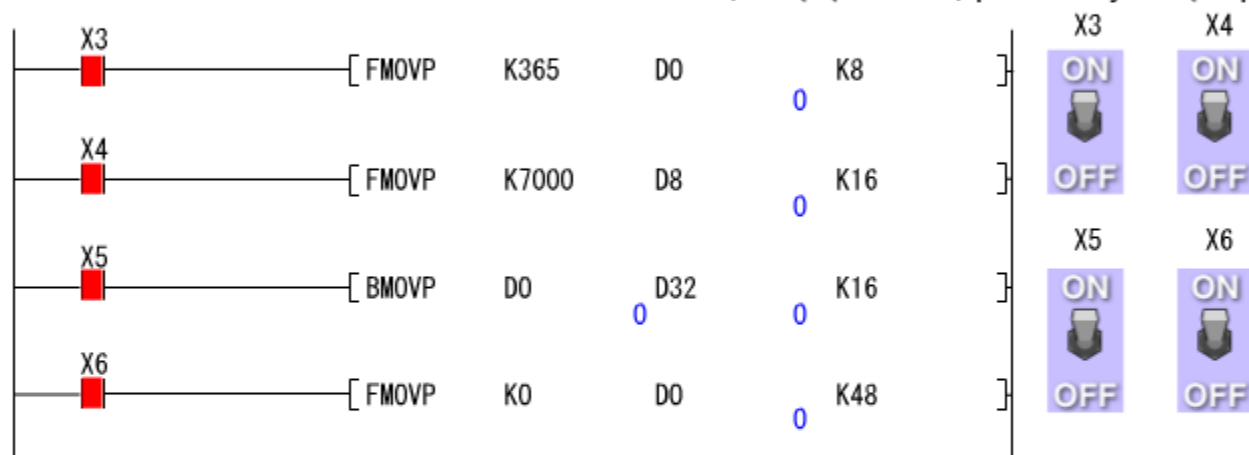
3.1.2

Jednoczesne przenoszenie danych do wielu rejestrów

Instrukcje MOV/MOVP są używane do przesyłania danych do adresu. Aby przesłać dane do wielu adresów pamięci mających numery ciągle, można wykorzystać „instrukcję przesyłania partii identycznych danych” (FMOV) lub „instrukcję przesyłania bloku identycznych danych” (BMOV).

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania poniższych instrukcji, klikając przełączniki wejściowe znajdujące się po prawej stronie. Każda liczba w kolorze niebieskim oznacza wartość (bieżącą wartość) przechowywaną w pamięci.



Monitorowanie pamięci

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

Gdy każdy sygnał wejściowy zostanie WŁĄCZONY, określone dane są przesyłane jednocześnie.

UWAGA: Na trzecim szczeblu drabinki rozpoczynającym się od X5, dane są przesyłane za pomocą instrukcji BMOV.

3.1.2

Jednoczesne przenoszenie danych do wielu rejestrów

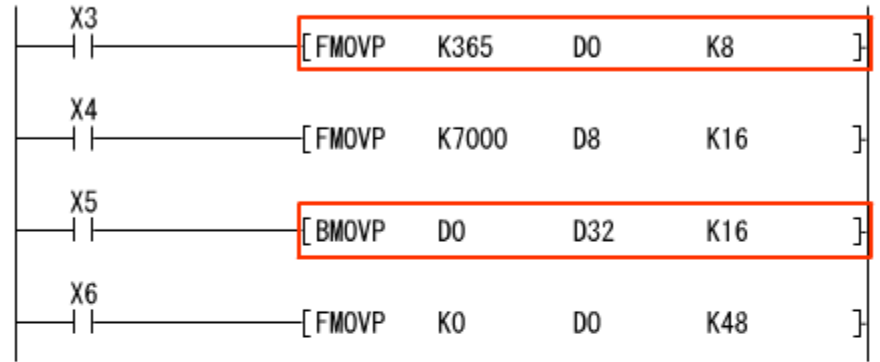
■ Kody i funkcje instrukcji

Symbol	Funkcja
	<p>Przesyłanie partii identycznych danych (FMOV)</p> <p>Jeśli warunek wejściowy jest spełniony, dane określone w źródle (S) są przesyłane (kopiowane) do adresu określonego przez miejsce przeznaczenia (D) i do „n” adresów następujących po D.</p>
	<p>Przesyłanie partii identycznych danych (impulsowe) (FMOVP)</p> <p>Przy wystąpieniu zbocza narastającego warunku, dane określone w źródle (S) są przesyłane (kopiowane) do adresu określonego przez miejsce przeznaczenia (D) i do „n” adresów następujących po D.</p>
	<p>Przesyłanie partii danych blokowych (BMOV)</p> <p>Jeśli warunek wejściowy jest spełniony, dane określone w źródle (S) i kolejnych „n” adresach są przesyłane do adresu określonego przez miejsce przeznaczenia (D) i do kolejnych „n” adresów.</p>
	<p>Przesyłanie partii danych blokowych (impulsowe) (BMOVP)</p> <p>Przy wystąpieniu zbocza narastającego warunku, dane określone w źródle (S) i kolejnych „n” adresach są przesyłane do adresu określonego przez miejsce przeznaczenia (D) i do kolejnych „n” adresów.</p>

3.1.2 Jednoczesne przenoszenie danych do wielu rejestrów

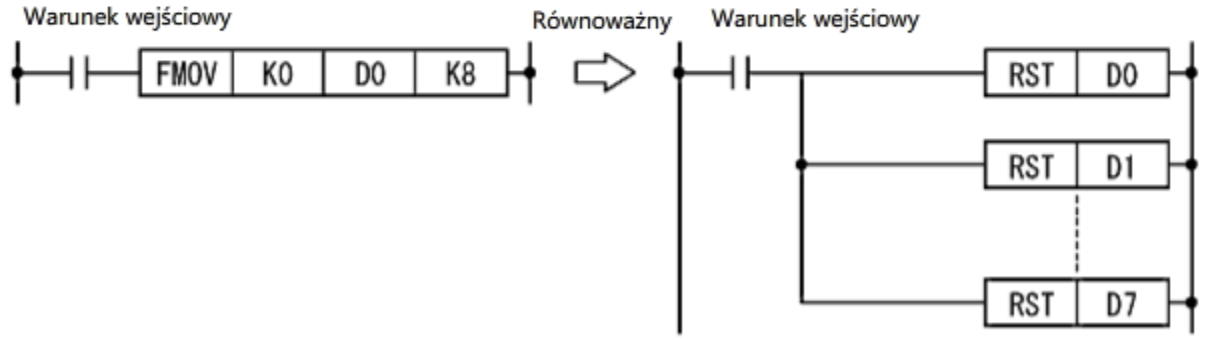
Program drabinkowy i operacja drabinkowa

Kliknij obszar z migającym .



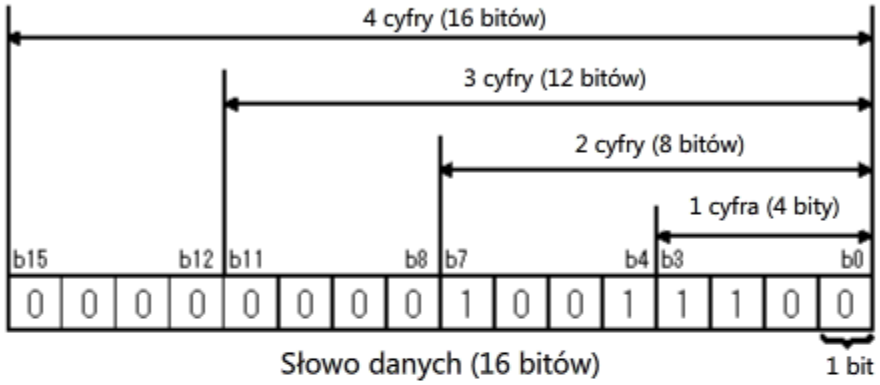
Zastosowanie instrukcji FMOV i BMOV

Instrukcja FMOV jest przydatna do jednorazowego usuwania dużych ilości danych.



3.1.3 Cyfra operandu bitowego

Czterobitowe rejestry są zgrupowane w jednej cyfrze operandu bitowego w celu sterowania informacją bitową w pewnym zakresie (przesyłanie danych itp.).



■ Jak określić cyfrę operandu bitowego

Cyfrę operandu bitowego można wyrazić jako „liczbę cyfr” + „adres początkowy”. Liczba cyfr jest wielokrotnością 4. Poniższa tabela przedstawia kilka przykładów. Poniżej podano przykłady, w których adres początkowy to „M0”.

Zakres bitu	Metoda określania
Dana 16-bitowa	K4M0 (16 bitów, M0 do M15)
Dana 32-bitowa	K8M0 (32 bity, M0 do M31)

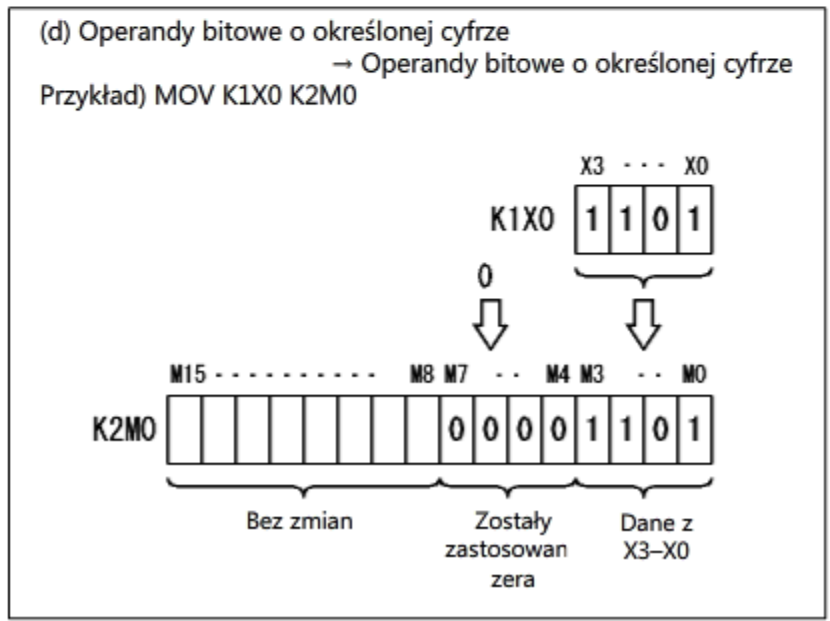
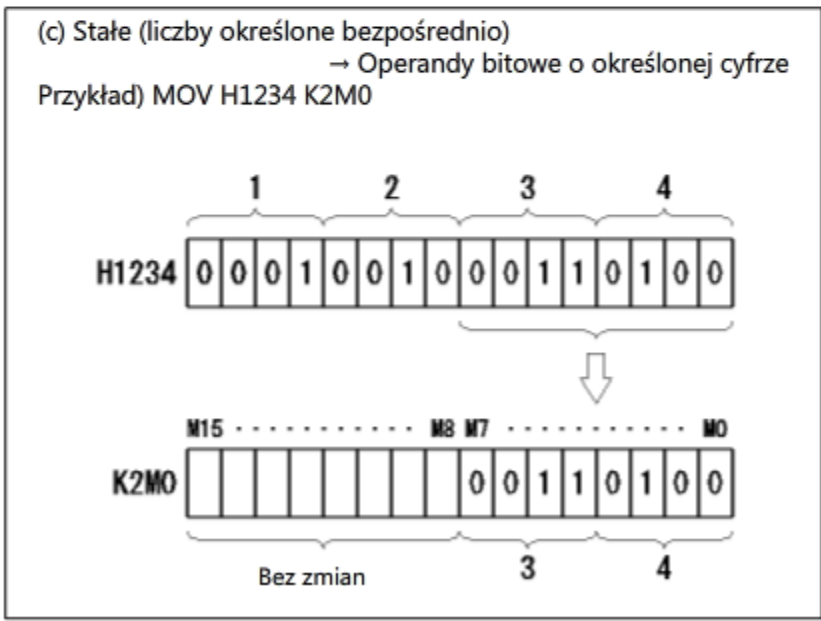
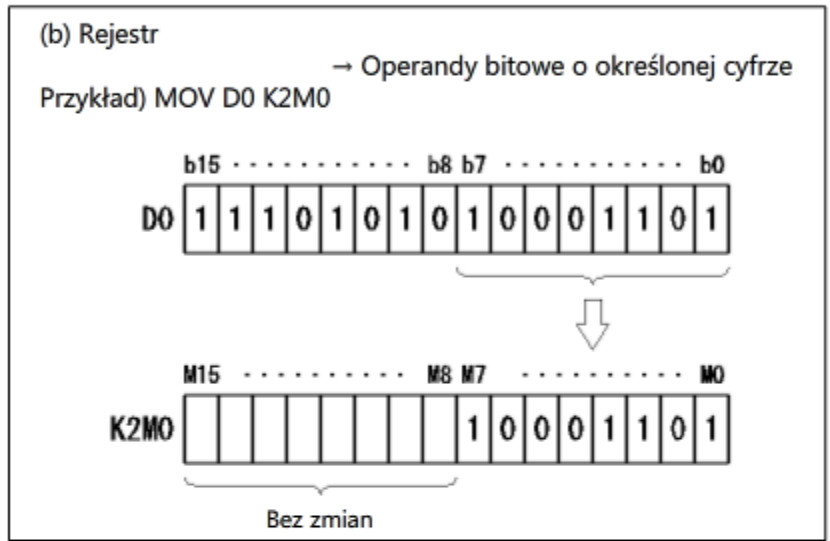
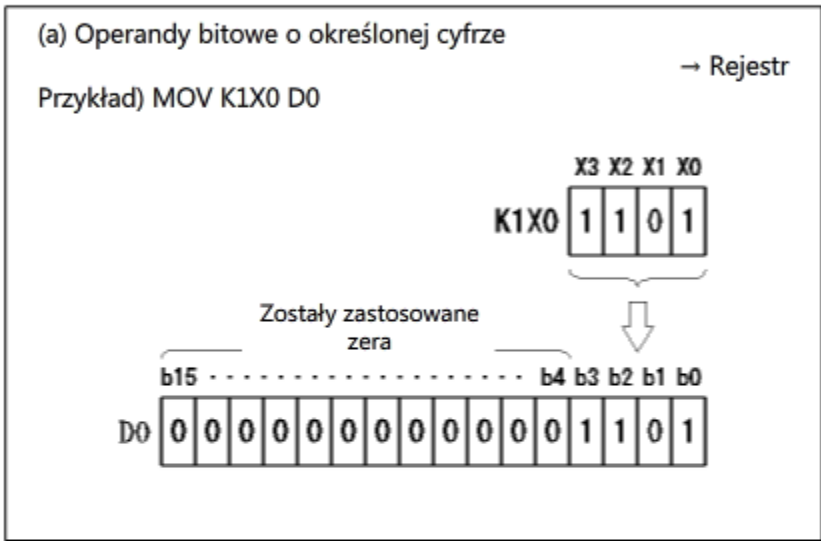
Cyfra operandu bitowego (liczba bitów) określa zakres możliwych do wykorzystania wartości numerycznych.

Cyfra operandu bitowego	Zakres możliwych do wykorzystania wartości numerycznych
K1 (4 bity)	0 do 15
K2 (8 bitów)	0 do 255
K3 (12 bitów)	0 do 4095
K4 (16 bitów)	-32768 do 32767 16 bit może służyć jako znak plus/minus, określający ujemne wartości.

3.1.3 Przykłady przesyłania cyfry operandu bitowego

Instrukcje przesyłania danych są używane do przesyłania (kopiowania) wartości numerycznych ze źródła do adresu miejsca przeznaczenia.

Poniższe przykłady przedstawiają przesyłanie określonych danych.



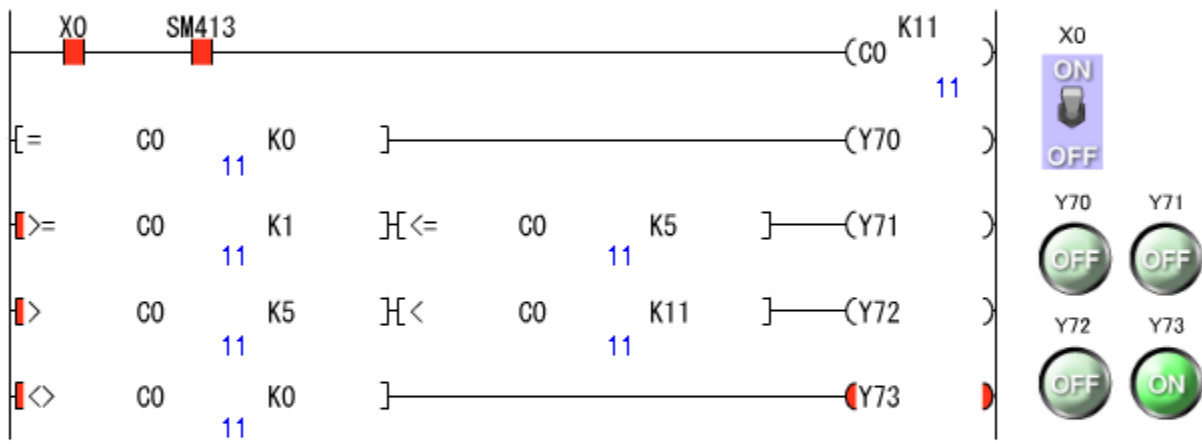
3.2 Porównanie wartości liczbowych

Instrukcje operacji porównania są używane do porównania jednostkowych słów danych i danych przechowywanych w rejestrach.

Jeśli warunek (\neq) jest spełniony, wykonywana jest następna instrukcja.

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania poniższych instrukcji, klikając przełączniki wejściowe znajdujące się po prawej stronie. Każda liczba w kolorze niebieskim oznacza wartość (bieżącą wartość) przechowywaną w pamięci.



Control panel for simulation:

- X0: ON/OFF switch (currently OFF)
- Y70: OFF button
- Y71: OFF button
- Y72: OFF button
- Y73: ON button

Resetowanie symulacji

Reset

Y70 do Y73 zostają WŁĄCZONE/WYŁĄCZONE w zależności od bieżącej wartości C0.

SM413 jest specjalnym przekaźnikiem, który jest WŁĄCZANY lub WYŁĄCZANY w odstępach 1-sekundowych przez moduł CPU (zegar 2-sekundowy).

Gdy X0 jest Wł., C0 dokonuje zliczenia co 2 sekundy.

* SM413 jest specjalnym przekaźnikiem, który jest WŁĄCZANY lub WYŁĄCZANY w odstępach 1-sekundowych (zegar 2-sekundowy). SM403 można stosować w modelach MELSEC iQ-R/Q/L/iQ-F. Model MELSEC-F nie ma przekaźnika zegara 2-sekundowego, ale ma M8011 (zegar 0,01 s), M8012 (zegar 0,1 s), M8013 (zegar 1 s), and M8014 (zegar 1 m).

3.2

Porównanie wartości liczbowych

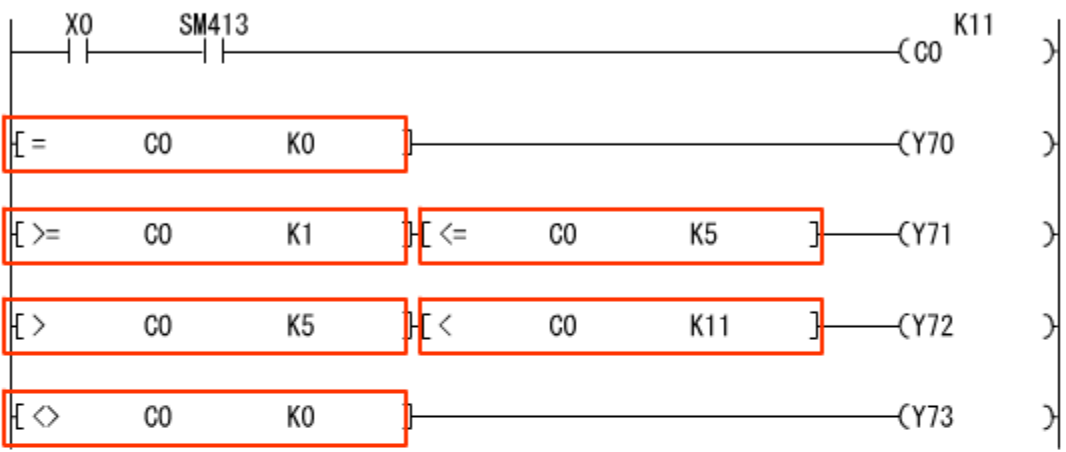
Kody i funkcje instrukcji

Symbol	Funkcja
	Porównuje 16-bitowe dane binarne. (=) Warunek jest spełniony, gdy ŹRÓDŁO 1 jest równe ŹRÓDŁU 2.
	Porównuje 16-bitowe dane binarne. (<) Warunek jest spełniony, gdy ŹRÓDŁO 1 jest mniejsze niż ŹRÓDŁO 2.
	Porównuje 16-bitowe dane binarne. (>) Warunek jest spełniony, gdy ŹRÓDŁO 1 jest większe niż ŹRÓDŁO 2.
	Porównuje 16-bitowe dane binarne. (<=) Warunek jest spełniony, gdy ŹRÓDŁO 1 jest równe ŹRÓDŁU 2 lub mniejsze.
	Porównuje 16-bitowe dane binarne. (>=) Warunek jest spełniony, gdy ŹRÓDŁO 1 jest równe ŹRÓDŁU 2 lub większe.
	Porównuje 16-bitowe dane binarne. (<>) Warunek jest spełniony, gdy ŹRÓDŁO 1 nie jest równe ŹRÓDŁU 2.

3.2 Porównanie wartości liczbowych

■ Program drabinkowy i operacja drabinkowa

Kliknij obszar z migającym



SM413 jest specjalnym przekaźnikiem, który jest WŁĄCZANY lub WYŁĄCZANY w odstępach 1-sekundowych przez moduł CPU (zegar 2-sekundowy). Specjalne przekaźniki (SM) są urządzeniami przekaźnikowymi w module CPU. Każdy specjalny przekaźnik spełnia określoną rolę.

W tym punkcie opisano podstawowe operacje arytmetyczne na rejestrach (numeryczne).

■ Dodawanie i odejmowanie

Operacje arytmetyczne wykorzystujące symbole dodawania (+) i odejmowania (-).

■ Mnożenie i dzielenie

Operacje arytmetyczne wykorzystujące symbole mnożenia (*) i dzielenia (/).

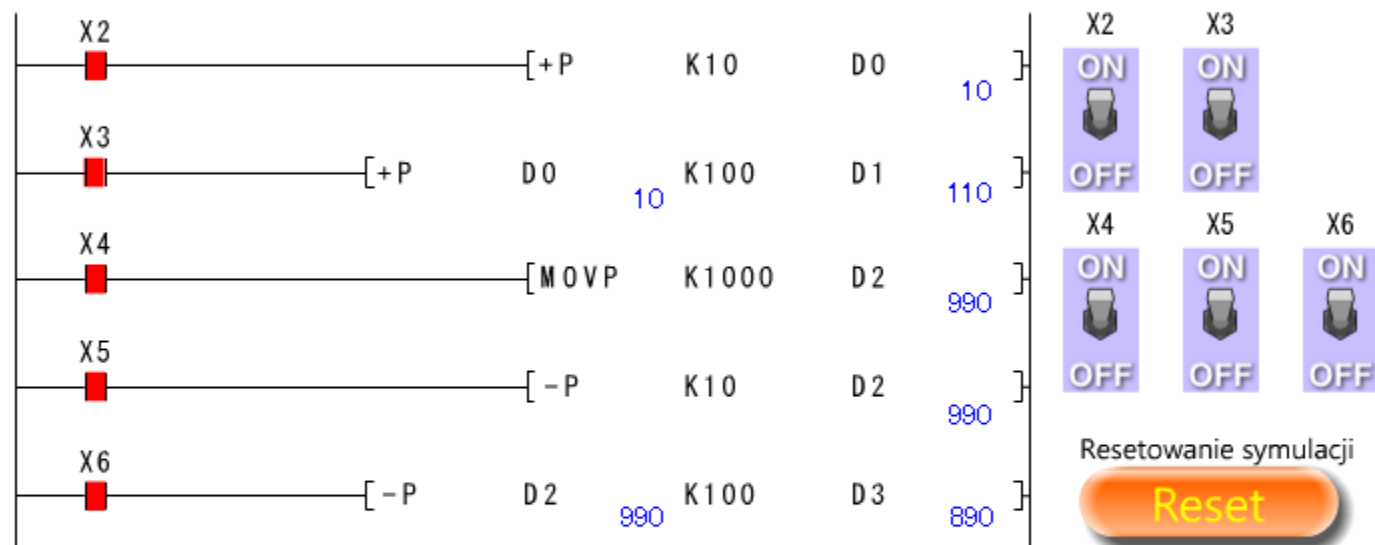
Istnieją różnice pomiędzy instrukcjami dla modelu MELSEC iQ-R/Q/L/iQ-F i MELSEC-F, ale podstawowa koncepcja jest taka sama. Wyjaśnienia w tym punkcie bazują na instrukcjach używanych w modelu MELSEC iQ-R/Q/L /iQ-F.

3.3.1 Dodawanie i odejmowanie

Poniższy diagram przedstawia instrukcje wykonujące dodawanie i odejmowanie i zapisujące otrzymaną wartość w określonych rejestrach.

■ Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania poniższych instrukcji, klikając przełączniki wejściowe znajdujące się po prawej stronie. Każda liczba w kolorze niebieskim oznacza wartość (bieżącą wartość) przechowywaną w urządzeniu.



Gdy każdy sygnał wejściowy zostanie WŁĄCZONY, wykonywana jest operacja arytmetyczna.

- Przykład oparty na modelu MELSEC iQ-R/Q/L/iQ-F.

3.3.1 Dodawanie i odejmowanie

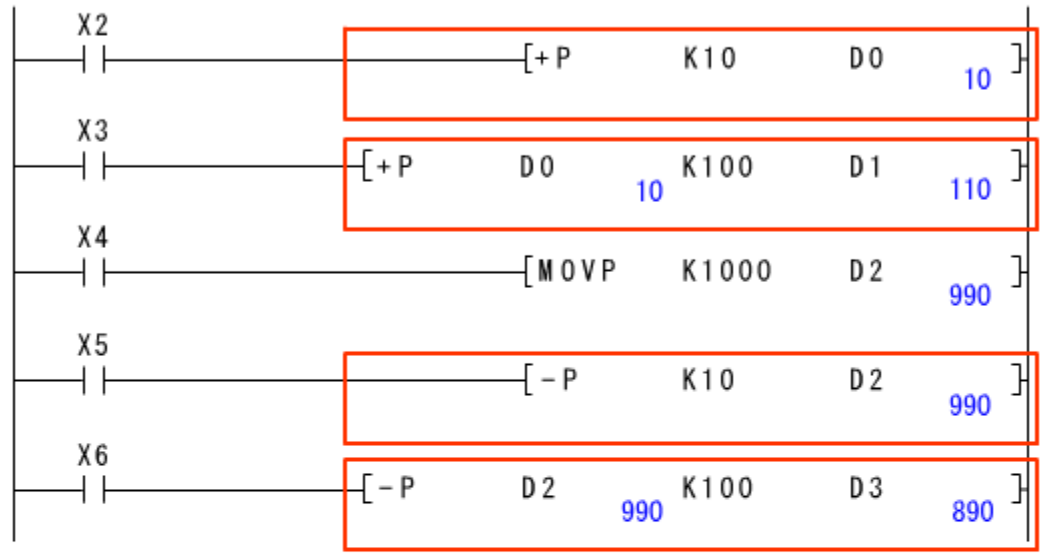
Kody i funkcje instrukcji

Symbol	Funkcja
<p>Instruction symbol: + (P) S D</p>	<p>Dodawanie 16-bitowych danych binarnych</p> <ul style="list-style-type: none"> - + S D : Wykonywana jest operacja „D + S = D”. - + S1 S2 D : Wykonywana jest operacja „S1 + S2 = D”.
<p>Instruction symbol: + (P) S1 S2 D</p>	
<p>Instruction symbol: - (P) S D</p>	<p>Odejmowanie 16-bitowych danych binarnych</p> <ul style="list-style-type: none"> - + S D : Wykonywana jest operacja „D - S = D”. - - S1 S2 D : Wykonywana jest operacja „S1 - S2 = D”.
<p>Instruction symbol: - (P) S1 S2 D</p>	

3.3.1 Dodawanie i odejmowanie

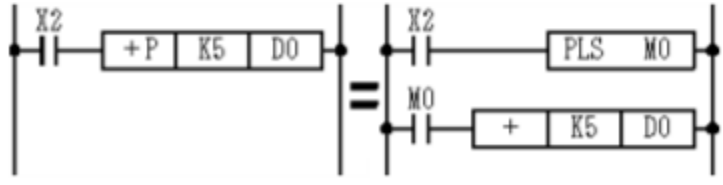
Program drabinkowy i operacja drabinkowa

Kliknij obszar z migającym



Uwaga dotycząca instrukcji dodawania i odejmowania

W normalnych warunkach korzystaj z instrukcji +P/-P w celu wykonania dodawania/odejmowania.
 W przypadku użycia instrukcji +/-, dodawanie/odejmowanie jest wykonywane wielokrotnie, dopóki warunek wejściowy jest spełniony.
 W każdym z poniższych szczebli drabinki dodawanie wykonywane jest tylko jednorazowo, gdy X2 zostaje WŁĄCZONY.



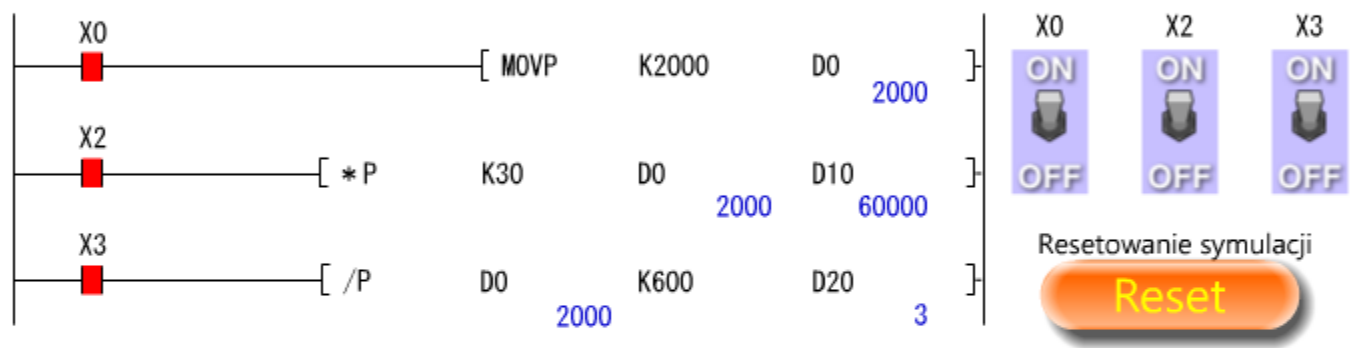
* Przykład oparty na modelu MELSEC iQ-R/Q/L/iQ-F.

3.3.2 Mnożenie i dzielenie

Poniższy diagram przedstawia instrukcje wykonujące mnożenie i dzielenie i zapisujące otrzymaną wartość w określonych rejestrach.

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania poniższych instrukcji, klikając przełączniki wejściowe znajdujące się po prawej stronie. Każda liczba w kolorze niebieskim oznacza wartość (bieżącą wartość) przechowywaną w pamięci.



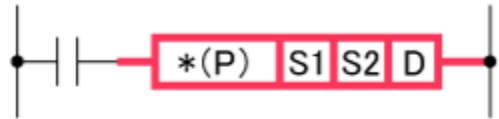
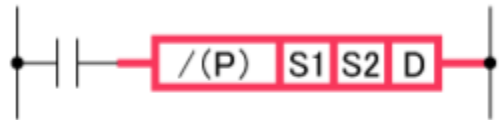
Gdy każdy sygnał wejściowy zostanie WŁĄCZONY, wykonywana jest operacja arytmetyczna.

* Przykład oparty na modelu MELSEC iQ-R/Q/L/iQ-F.

3.3.2

Mnożenie i dzielenie

■ Kody i funkcje instrukcji

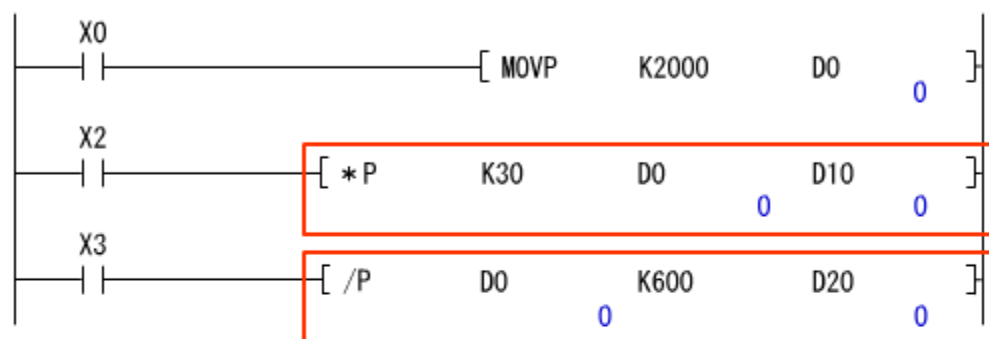
Symbol	Funkcja
	<p>Mnożenie 16-bitowych danych binarnych (*) Wykonywana jest operacja „$S1 \times S2 = (D+1 D)$”. („D+1” jest rejestrem który następuje po D. Jeśli D oznacza D100, „D+1” oznacza D101.) Wynikiem operacji jest dana 32-bitowa, która składa się z 2 jednostek słów („D” i „D+1”).</p>
	<p>Dzielenie 16-bitowych danych binarnych Wykonywana jest operacja „$S1/S2 = (D \text{ [iloraz]}, D + 1 \text{ [reszta]})$”. („D + 1” jest rejestrem, który następuje po D. Jeśli D oznacza D100, „D + 1” oznacza D101). Wynikiem operacji jest liczba całkowita.</p>

3.3.2

Mnożenie i dzielenie

■ Program drabinkowy i operacja drabinkowa

Kliknij obszar z migającym



■ Uwaga dotycząca instrukcji mnożenia i dzielenia

Aby wykonać instrukcję mnożenia i dzielenia, wymagane są dwa kolejne rejestry (D, D+1) dla miejsca przeznaczenia (D).

Mnożenie

$$\begin{array}{|c|} \hline S1 \\ \hline K30 \\ \hline \end{array} * \begin{array}{|c|} \hline S2 \\ \hline D0(2000) \\ \hline \end{array} = \begin{array}{|c|c|} \hline D+1 & D \\ \hline D11 & (60000) & D10 \\ \hline \end{array}$$

Dzielenie

$$\begin{array}{|c|} \hline S1 \\ \hline D0(2000) \\ \hline \end{array} / \begin{array}{|c|} \hline S2 \\ \hline K600 \\ \hline \end{array} = \begin{array}{|c|c|} \hline D & D+1 \\ \hline D20(3) & D21(200) \\ \hline \end{array}$$

Iloraz Reszta

* Przykład oparty na modelu MELSEC iQ-R/Q/L/iQ-F.

3.3.3 Różnice pomiędzy modelami MELSEC iQ-R/Q/L/iQ-F i MELSEC-F

Istnieją różnice pomiędzy symbolami modeli MELSEC iQ-R/Q/L/iQ-F i MELSEC-F. Poniższa tabela przedstawia główne różnice.

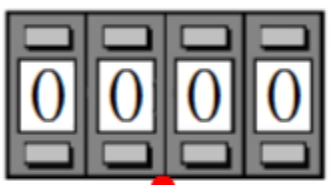
Operacja arytmetyczna	Instrukcja używana w modelu Model MELSEC iQ-R/Q/L/iQ-F	Instrukcja używana w modelu MELSEC-F	Różnice
Dodawanie (+)			Model MELSEC iQ-R/Q/L/iQ-F: +(P) Model MELSEC-F: ADD(P)
Odejmowanie (-)			Model MELSEC iQ-R/Q/L/iQ-F: -(P) Model MELSEC-F: SUB(P)
Mnożenie (*)			Model MELSEC iQ-R/Q/L/iQ-F: *(P) Model MELSEC-F: MUL(P)
Dzielenie (/)			Model MELSEC iQ-R/Q/L/iQ-F: /(P) Model MELSEC-F: DIV(P)

3.4 Przesyłanie/odbieranie danych pomiędzy PLC a urządzeniami we/wy

Przełącznik wejścia cyfrowego jest urządzeniem wejściowym, które przesyła dane do sterownika programowalnego w postaci wartości numerycznych. Wyświetlacz cyfrowy jest urządzeniem wyjściowym, które wyświetla dane otrzymane ze sterownika programowalnego w postaci wartości numerycznych.

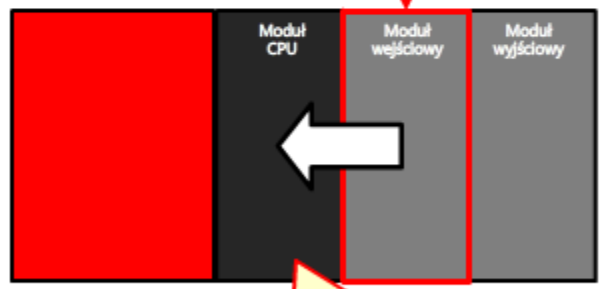
Dane otrzymane z przełącznika wejścia cyfrowego muszą zostać sformatowane, tak aby umożliwić ich przetworzenie przez sterownik programowalny. Podobnie wyjście danych wyświetlacza cyfrowego musi zostać sformatowane w możliwym do odczytania formacie przez wyświetlacz cyfrowy.

Przełącznik wejścia cyfrowego



Przesyła wartości numeryczne

PLC



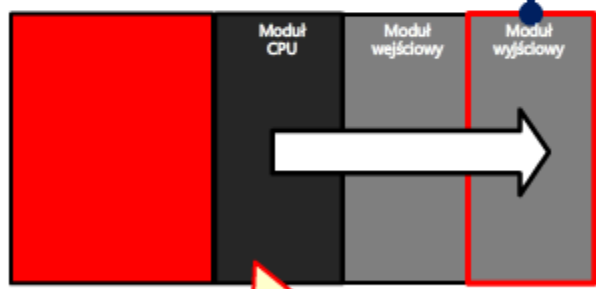
Formatowanie sterownika programowalnego w celu przetworzenia

Wyświetlacz cyfrowy



Wyświetla wartości numeryczne

PLC



Formatowanie wyświetlacza cyfrowego w celu wyświetlania

3.4.1 Otrzymywanie sygnałów wejściowych przełącznika wejścia cyfrowego

W celu otrzymywania sygnałów wejściowych od przełącznika wejścia cyfrowego przez sterownik programowalny stosowane są instrukcje BIN.

Kody i funkcje instrukcji

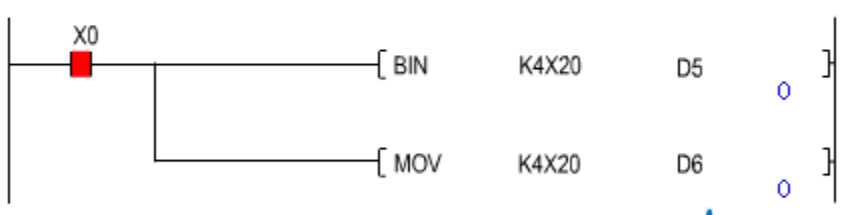
Symbol	Funkcja
	Dane w rejestrze (S) są formatowane na format, który może zostać przetworzony przez sterownik programowalny, a następnie zapisane w rejestrze (D).

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania poniższych instrukcji, klikając przełączniki wejściowe znajdujące się po prawej stronie. Każda liczba w kolorze niebieskim oznacza wartość (bieżącą wartość) przechowywaną w pamięci.

D5 przechowuje dane otrzymane z przełącznika wejścia cyfrowego po ich sformatowaniu przez instrukcję BIN.

D6 przechowuje niesformatowane dane otrzymane z przełącznika wejścia cyfrowego.



X0
ON
OFF

Wejście zadajnika cyfrowego

0 0 0 0

<- Zmniejszenie wartości

<- Zwiększenie wartości

X2F do X20

Resetowanie symulacji

Reset

W przypadku stosowania instrukcji MOV numery nie pasują.

3.4.2 Wyświetlanie danych PLC na wyświetlaczu cyfrowym

Aby wyświetlać dane sterownika programowalnego na wyświetlaczu cyfrowym stosowana jest instrukcja BCD.

Kody i funkcje instrukcji

Symbol	Funkcja
	Dane w rejestrze (S) są formatowane na format, który może zostać wyświetlony na wyświetlaczu cyfrowym, a następnie zapisane w rejestrze (D).

Program drabinkowy i operacja drabinkowa

Wykonaj symulację działania poniższych instrukcji, klikając przełączniki wejściowe znajdujące się po prawej stronie.

Górny wyświetlacz cyfrowy przedstawia dane sformatowane przez instrukcję BCD.

Dolny wyświetlacz cyfrowy przedstawia dane niesformatowane.



W przypadku stosowania instrukcji MOV numery nie pasują.

Resetowanie symulacji



W trakcie tego kursu przekazano następujące informacje:

- Koncepcja sygnałów wejściowych i wyjściowych ze/do sterowników programowalnych
- Główne instrukcje sterujące sterownikami programowalnymi
- Informacje otrzymane przez sterownik programowalny MELSEC są wykonywane w programach drabinkowych w sterowniku programowalnym, a wyniki ich wykonywania są przekazywane na zewnątrz jako sygnały wyjściowe
- Różnice pomiędzy formatami bitowymi i słowa danych
- Podstawy programów sterujących

Uczestnicząc w kursie „Engineering Software MELSOFT GX Works3 (Ladder)” (Oprogramowanie programistyczne MELSOFT GX Works3 (program drabinkowy), dowiesz się, jak edytować i rejestrować programy w module CPU modelu MELSEC iQ-R/iQ-F.

Należy przeprowadzić kurs „GX Works2 Basics” (Podstawy GX Works2), aby dowiedzieć się, jak edytować i rejestrować programy w module CPU modelu MELSEC-Q/L/F.

Po zakończeniu wszystkich etapów kursu **PLC - podstawy programowania**, możesz teraz przystąpić do testu końcowego. W razie niejasności w zakresie któregoś z tematów, wykorzystaj tę możliwość do ponownego zapoznania się z tymi zagadnieniami. **Test końcowy składa się z 11 pytań (54 elementów).**

Możesz zdawać test końcowy dowolną ilość razy.

Jak rozwiązywać test

Po wybraniu odpowiedzi upewnij się, że przycisk **Odpowiedź** został kliknięty. Twoja odpowiedź zostanie utracona, jeśli będziesz kontynuować bez kliknięcia przycisku Odpowiedź. (Zostanie potraktowana jako pytanie, na które nie udzielono odpowiedzi).

Punktacja końcowa

Liczba prawidłowych odpowiedzi, liczba pytań, procent prawidłowych odpowiedzi i wynik zaliczony/niezaliczony pojawiają się na stronie wyniku.

Prawidłowe odpowiedzi: 4

Wszystkie pytania: 4

Procent prawidłowych odpowiedzi: 100%

Aby zaliczyć test, musisz odpowiedzieć poprawnie na **60%** pytań.

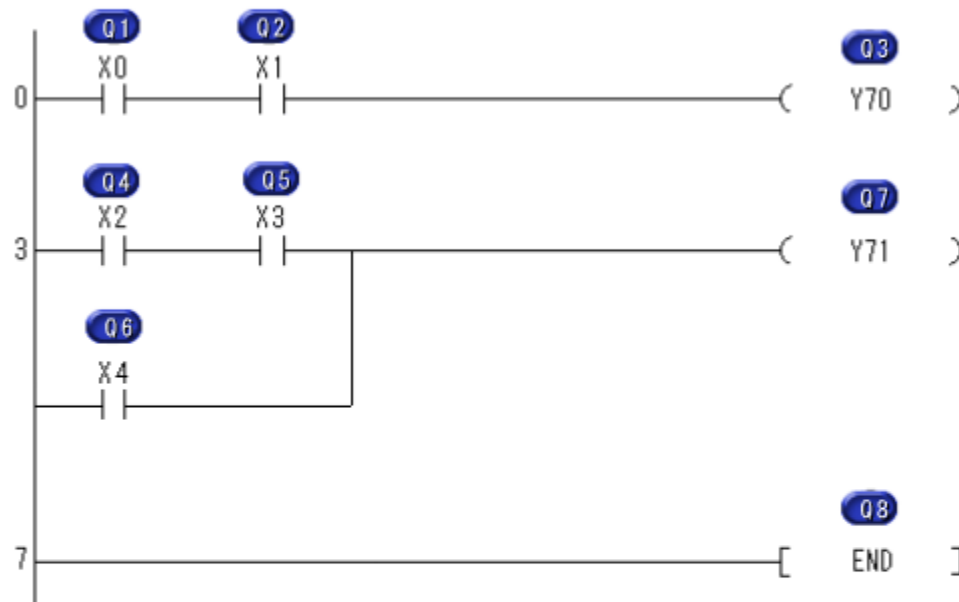
Kontynuuj

Przeglądaj

- Kliknij przycisk **Kontynuuj**, aby zakończyć test.
- Kliknij przycisk **Przeglądaj**, aby przeglądać test. (Sprawdzenie prawidłowych odpowiedzi)
- Kliknij przycisk **Spróbuj ponownie**, aby powtórzyć test.

Test **Test końcowy 1**

Ponumeruj następujące instrukcje w kolejności ich przetwarzania.

Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8

Odpowiedź

Wstecz

Test Test końcowy 2

Poniższe zdania opisują zewnętrzne urządzenia we/wy i sygnały we/wy do/ze sterowników programowalnych. Uzupełnij zdania przy użyciu prawidłowych słów.

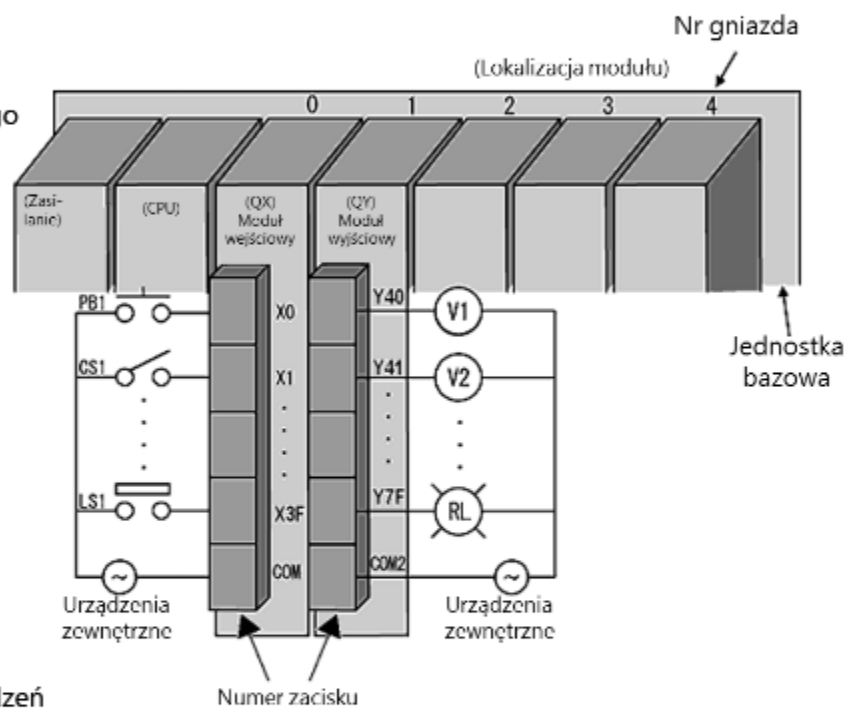
1) Numery wejść i wyjść dla sterowników programowalnych typu Q zaczynają się od (--Select--) i wyrażone są w wartościach (--Select--).

2) Te same numery są używane dla sygnałów wejściowych i wyjściowych. Z tego względu oznaczenia wejść są poprzedzone znakiem (--Select--), a oznaczenia wyjść są poprzedzone znakiem (--Select--).

3) Numery przypisane sygnałom pochodzącym z urządzeń zewnętrznych są określane przez następujące warunki:
- Miejsce, w którym zainstalowany jest (--Select--) na jednostce bazowej
- Numer zacisku

4) Numery przypisane sygnałom wyjściowym (cewkom) przesyłanym do urządzeń zewnętrznych są określane przez następujące warunki:
- Miejsce, w którym zainstalowany jest (--Select--) na jednostce bazowej
- Numer zacisku

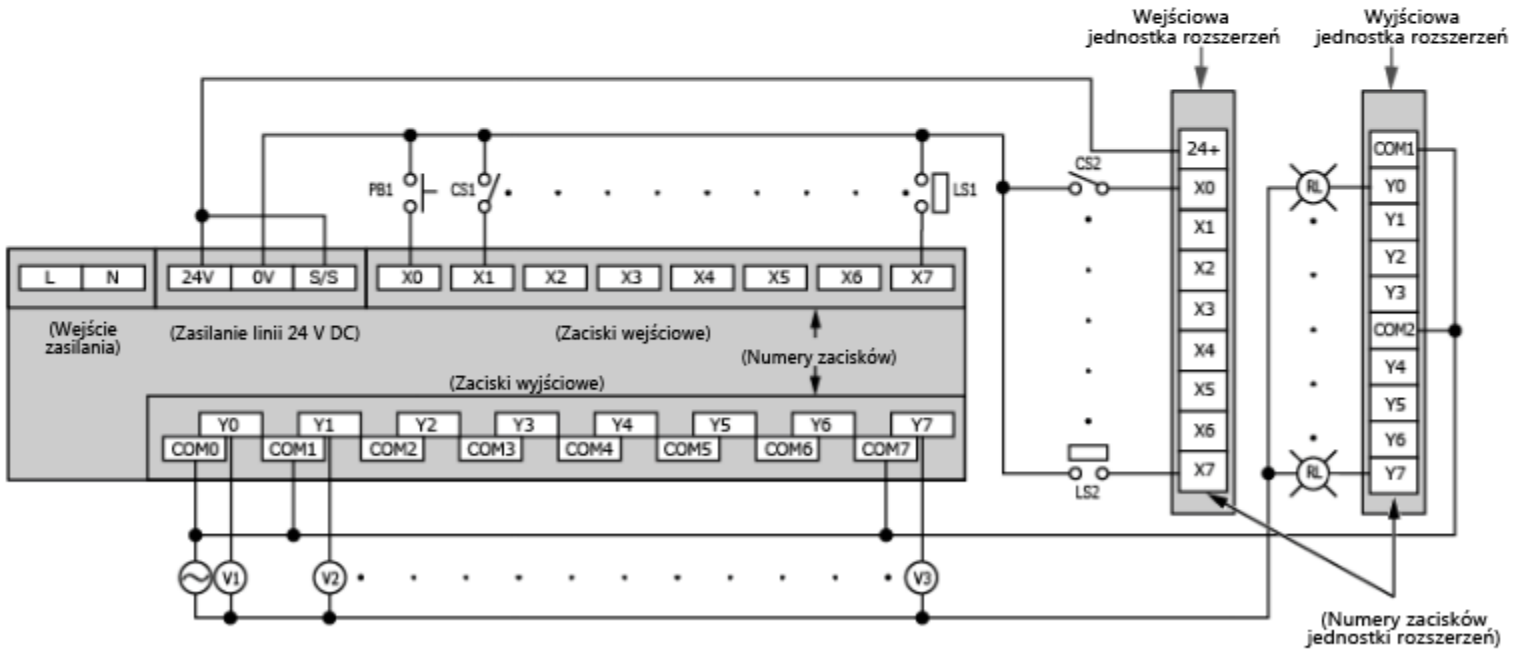
Odpowiedź Wstecz



Test Test końcowy 3

Poniższe zdania opisują zewnętrzne urządzenia we/wy i numer we/wy przypisany do sterowników programowalnych. Uzupełnij zdania przy użyciu prawidłowych słów. (Model MELSEC-F)

- 1) Numery we/wy dla sterowników programowalnych typu MELSEC-F zaczynają się od (--Select--) i wyrażone są w wartościach (--Select--).
- 2) Te same numery są używane dla sygnałów wejściowych i wyjściowych. Z tego względu oznaczenia wejść są poprzedzone znakiem (--Select--), a oznaczenia wyjść są poprzedzone znakiem (--Select--).
- 3) Jeśli dodawana jest jednostka rozszerzeń we/wy, jednostce zostanie przypisany numer, który następuje po numerze przypisanym poprzedniej (--Select--).
- 4) Numer we/wy jednostki zawsze zaczyna się od numeru, którego pierwszą cyfrą jest „0”. Jeśli numer we/wy poprzedniej jednostki kończy się wartością X17, numer we/wy następnej jednostki zaczyna się od (--Select--).

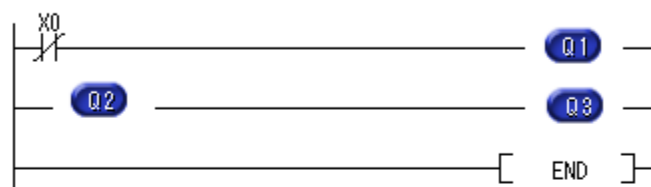


Test **Test końcowy 4**

Przeciągnij i upuść odpowiednie instrukcje, aby dokończyć program wykonujący następujące operacje:

Jeśli przełącznik X0 jest WYŁ., lampka A jest WŁ. (Y70 jest WŁ.)

Jeśli przełącznik jest WŁ., lampka B jest WŁ. (Y71 jest WŁ.)



Q1 --Select-- ▼ Q2 --Select-- ▼ Q3 --Select-- ▼

Odpowiedź

Wstecz

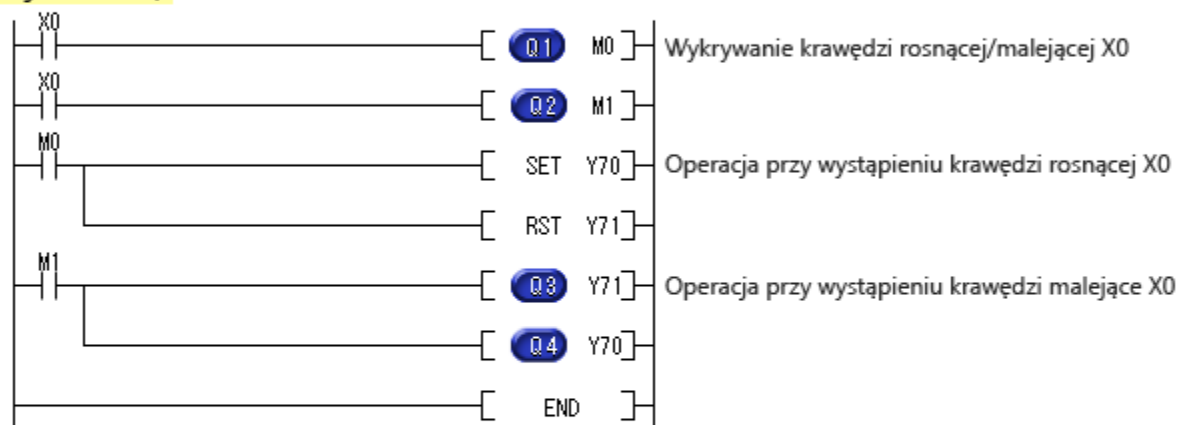
Test Test końcowy 5

Przecignij i upuść odpowiednie instrukcje, aby dokończyć program wykonujący następujące operacje:

Gdy materiały są przetwarzane, „sygnał trwania procesu” (X0) jest Wł.

Przy wystąpieniu krawędzi rosnącej „sygnału trwania procesu” (X0) lampka A jest Wł. (Y70 jest Wł.), a lampka B jest WYł. (Y71 jest WYł.).

Przy wystąpieniu krawędzi malejącej „sygnału trwania procesu” (X0) lampka B jest Wł. (Y70 jest Wł.), a lampka A jest WYł. (Y71 jest WYł.).



Q1 --Select-- ▾

Q2 --Select-- ▾

Q3 --Select-- ▾

Q4 --Select-- ▾

Odpowiedź

Wstecz

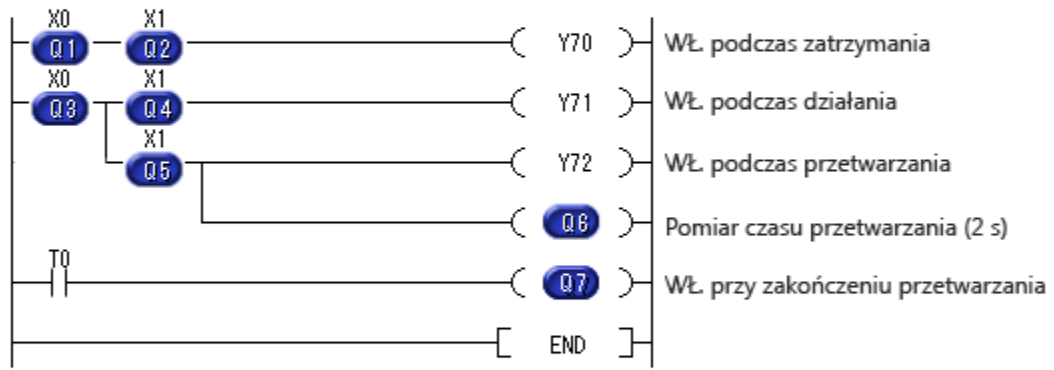
Test Test końcowy 6

Przecignij i upuść odpowiednie instrukcje, aby dokończyć program wykonujący następujące operacje:

Lampka jest WŁĄCZANA poprzez WŁĄCZENIE/WYŁĄCZENIE przełącznika rozpoczęcia działania (X0) i przełącznika rozpoczęcia przetwarzania (X1).

Po 2 sekundach po WŁĄCZENIU obu przełączników lampka D zostaje WŁĄCZONA.

[Rozpoczęcie działania (X0)]	[Przełącznik rozpoczęcia przetwarzania (X1)]	[Lampka]
WYŁ.	WYŁ.	Lampka A (Y70 jest WŁ.)
WŁ.	WYŁ.	Lampka B (Y71 jest WŁ.)
WŁ.	WŁ.	Lampka C (Y72 jest WŁ.), a po 2 s lampka D (Y73 jest WŁ.)

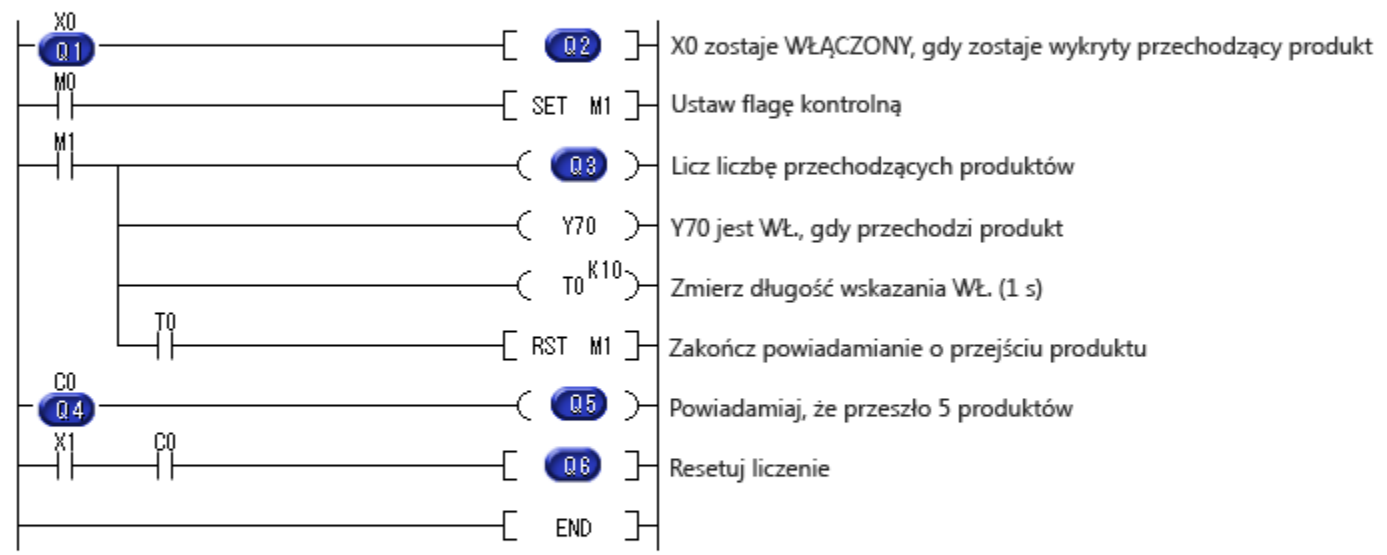


Q1 Q2 Q3 Q4
 Q5 Q6 Q7

Test Test końcowy 7

Przeciągnij i upuść odpowiednie instrukcje, aby dokończyć program wykonujący następujące operacje:

- Gdy produkt przechodzi na przenośniku taśmowym, sygnał X0 jest WŁ.
- Po przejściu produktu (po 3 s) lampka A zostaje WŁĄCZONA. (Y70 jest WŁ. przez 1 s)
- Po przejściu 5 produktów lampka B zostaje WŁĄCZONA. (Y71 jest WŁ.)
- Po WŁĄCZENIU lampki B przełącznik potwierdzenia (X1) zostaje WŁĄCZONY.



Q1 --Select-- Q2 --Select-- Q3 --Select-- Q4 --Select--

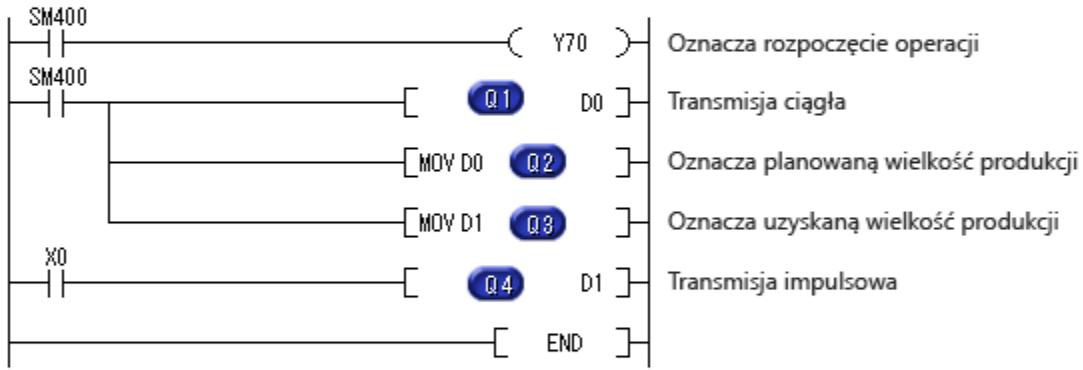
Q5 --Select-- Q6 --Select--

Odpowiedź Wstecz

Test Test końcowy 8

Przeciągnij i upuść odpowiednie instrukcje, aby dokończyć program wykonujący następujące operacje:

- 1) When the operation starts, the lamp A turns ON. (Y70 is ON)
- 2) Zaplanowana wielkość produkcji jest wprowadzana przy użyciu przełączników cyfrowych (X20-X2F). Wielkość jest przekazywana do rejestru danych D0 za każdym razem, gdy jest wprowadzana.
- 3) Wartości zapisane w rejestrze danych (D0, D1) są przesyłane w sposób ciągły i aktualizowane w wyświetlaczu cyfrowego w poniższy sposób.
Y40Y4F: Oznacza planowaną wielkość produkcji (D0) Y50-Y5F: Oznacza uzyskaną wielkość produkcji (D1)
- 4) Przełączniki cyfrowe X30 do X3F są stosowane do wprowadzenia uzyskanej wielkości produkcji. Gdy przełącznik ustawienia zakończenia (X0) zostaje WŁĄCZONY, uzyskana wielkość produkcji jest przekazywana do rejestru danych D1.



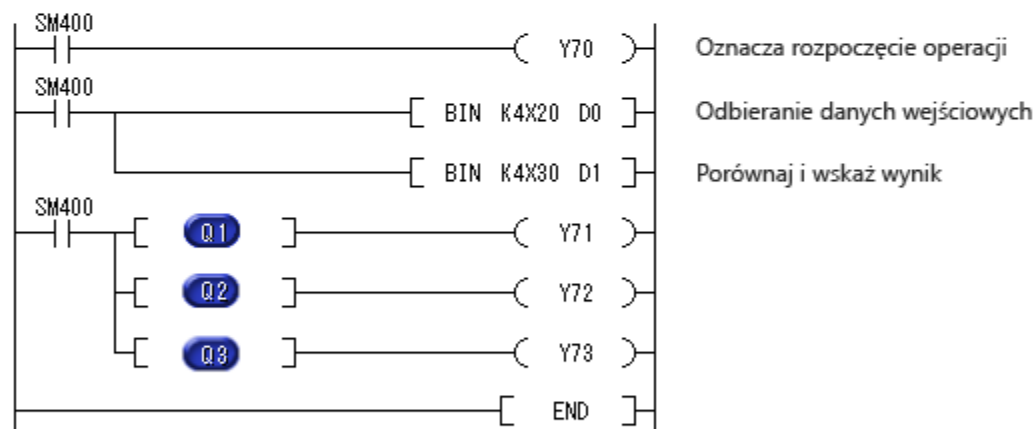
* W tym programie instrukcja MOV jest używana do przesyłania danych.
 * W celu monitorowania D0 i D1 używaj wartości szesnastkowych.

Q1 Q2 Q3 Q4

Test Test końcowy 9

Przeciągnij i upuść odpowiednie instrukcje, aby dokończyć program wykonujący następujące operacje:

- 1) When the operation starts, the lamp A turns ON. (Y70 is ON)
- 2) Przy WŁĄCZENIU przełącznika ustawienia zakończenia (X0) następujące operacje zostaną jednorazowo wykonane.
 - Planowana wielkość produkcji A, która została wprowadzona za pomocą przełączników cyfrowych (X20–X2F), jest formatowana i przekazywana do rejestru danych D0.
 - Planowana wielkość produkcji B, która została wprowadzona za pomocą przełączników cyfrowych (X30–X3F), jest formatowana i przekazywana do rejestru danych D1.
 - Rejestry danych D0 i D1 są wzajemnie porównywane, a wynik jest wskazywany przez lampkę.
 - D0 > D1: Lampka B (Y71 jest WŁ./WYŁ.)
 - D0 = D1: Lampka C (Y72 jest WŁ./WYŁ.)
 - D0 < D1: Lampka D (Y73 jest WŁ./WYŁ.)

Q1 Q2 Q3

Odpowiedź

Wstecz

Przeciągnij i upuść odpowiednie instrukcje, aby dokończyć program wykonujący następujące operacje:

- 1) Gdy rozpoczyna się operacja, lampka A zostaje WŁĄCZONA. (Y70 jest WŁ.)
- 2) Przy rozpoczęciu liczba planowanej produkcji 100 jest zapisywana w rejestrze danych D0.
- 3) Za każdym razem, gdy produkt jest kompletny, następujące informacje są zapisywane w rejestrach danych.

D1: Uzyskana wielkość produkcji (liczona jako krawędź rosnąca X0)

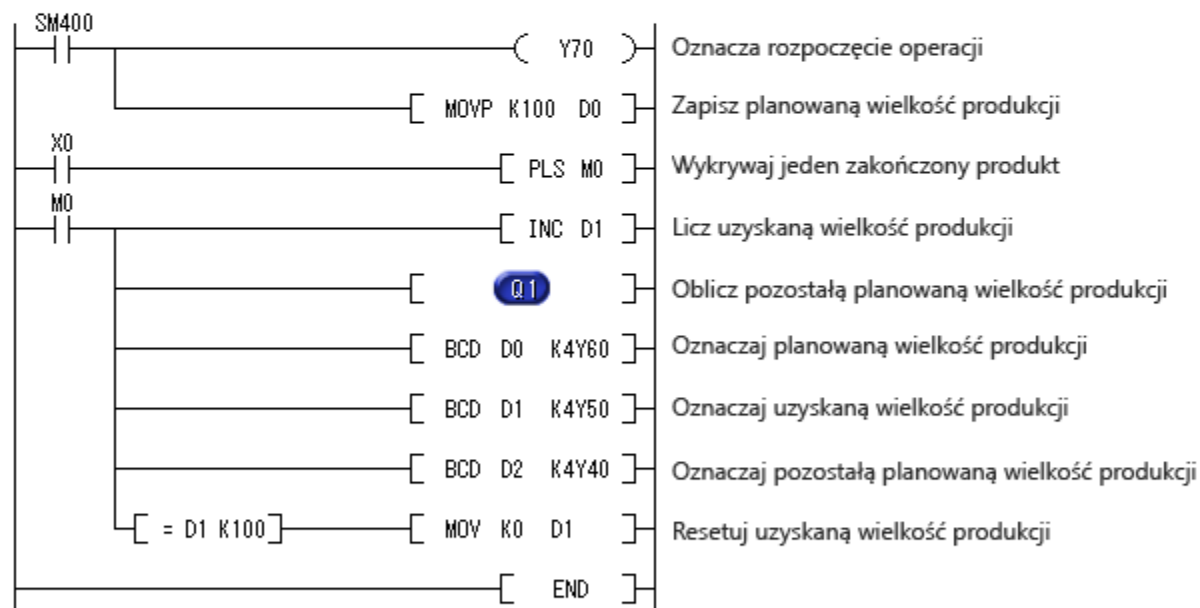
D2: Pozostała planowana wielkość produkcji $D2 = D0 - D1$

Wyświetlacz cyfrowy przedstawia następujące dane:

Y40Y4F: Wartość w D2 (pozostała planowana wielkość produkcji (od 0 do 100))

Y50Y5F: Wartość w D1 (uzyskana wielkość produkcji (od 0 do 100))

Y60Y6F: Wartość D0 (planowana wielkość produkcji (100))



Q1 --Select-- ▾

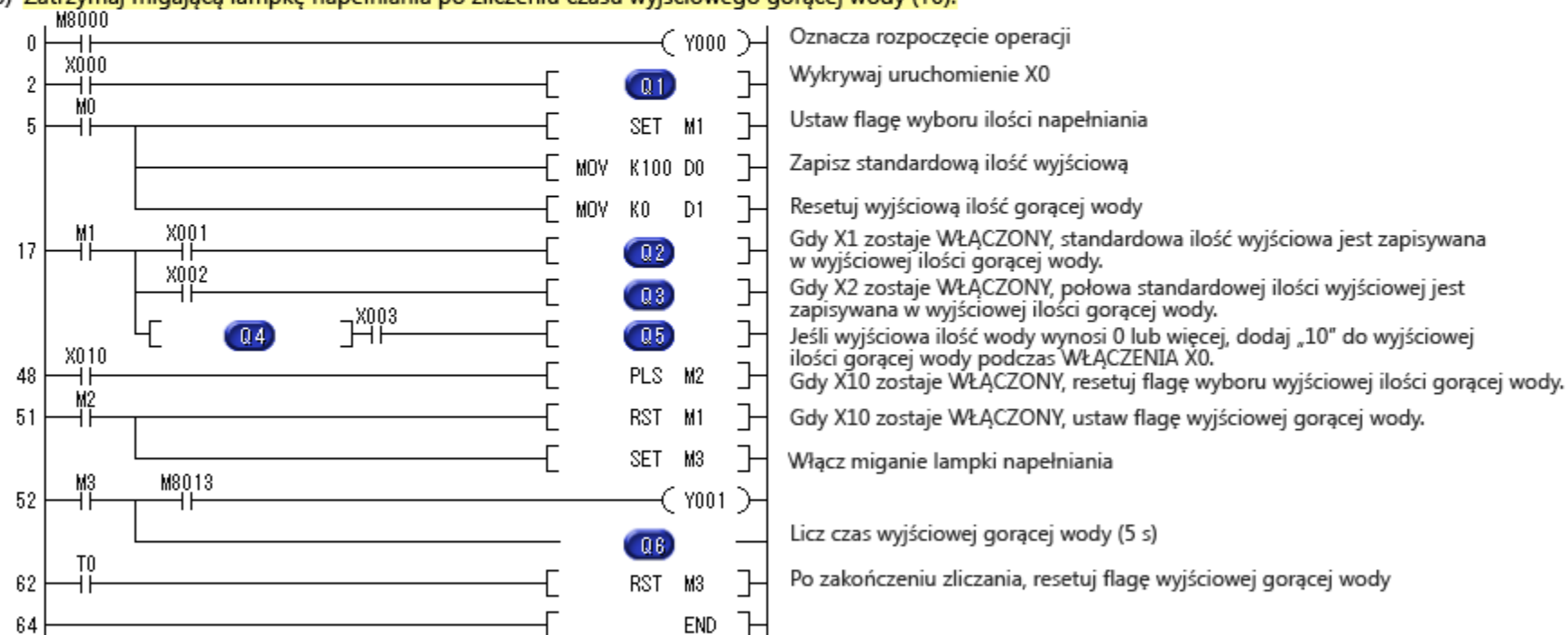
Odpowiedź

Wstecz

Test Test końcowy 11

Program sterujący przedstawiony poniżej jest przeznaczony dla modelu MELSEC-F i zawiera instrukcje oraz specjalne przekaźniki. Przeciągnij i upuść odpowiednie instrukcje, aby dokończyć program, który nalewa gorącą wodę z dystrybutora gorącej wody:

- 1) Gdy rozpoczyna się operacja, lampka zostaje WŁĄCZONA. (Y0 jest WŁ.)
- 2) Przy wystąpieniu krawędzi rosnącej rozpoczęcia działania dystrybutora gorącej wody (X0 zostaje WŁĄCZONY), wartość „100” jest zapisywana w wielkości standardowego dozowania wody D0, a wartość „0” jest zapisywana w wyjściowej ilości gorącej wody D1. (Reset danych)
- 3) Wybierz wyjściową ilość gorącej wody.
Przy wystąpieniu krawędzi rosnącej X1, standardowa ilość wyjściowa D0 jest zapisywana w wyjściowej ilości gorącej wody D1.
Przy wystąpieniu krawędzi rosnącej X2, połowa standardowej ilości wyjściowej D0 jest zapisywana w wyjściowej ilości gorącej wody D1.
- 4) Jeśli jest wybrana wyjściowa ilość gorącej wody D1 i wynosi 0 lub więcej, wartość „+10” jest dodawana do wyjściowej ilości gorącej wody D1 przy wystąpieniu krawędzi rosnącej X3, a następnie wartość dodana jest zapisywana w wyjściowej ilości gorącej wody.
- 5) Przy wystąpieniu krawędzi rosnącej wyjścia gorącej wody (X10), lampka napełniania miga w 1-sekundowych odstępach (naprzemienne WŁ. i WYŁ. Y1) i liczony jest czas wyjściowy gorącej wody (T0).
- 6) Zatrzymaj migającą lampkę napełniania po zliczeniu czasu wyjściowego gorącej wody (T0).



Q1 --Select--

Q2 --Select--

Q3 --Select--

Q4 --Select--

Q5 --Select--

Q6 --Select--

Odpowiedź

Wstecz

Test Wynik testu

Test końcowy został zakończony. Twoje wyniki są przedstawione poniżej.
Aby zakończyć test końcowy, przejdź do następnej strony.

Prawidłowe odpowiedzi: **11**

Wszystkie pytania: **11**

Procent prawidłowych odpowiedzi: **100%**

Kontynuuj

Przełóżaj

Gratulacje. Test został zaliczony.

Kurs **PLC - podstawy programowania** został ukończony.

Dziękujemy za wzięcie udziału w kursie.

Mamy nadzieję, że poruszone tematy były interesujące, a informacje uzyskane w trakcie tego kursu będą przydatne w przyszłości.

Możesz przeglądać kurs dowolną ilość razy.

Przełdaj

Zamknij