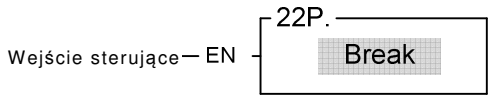
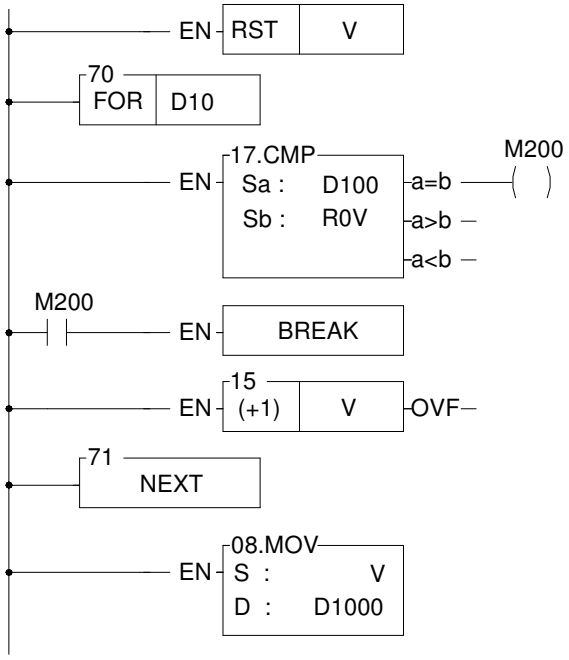
FUN22 P BREAK	PĘTLA BREAK FROM FOR iNEXT （BREAK）	FUN22 P BREAK

SYMBOL DRABINKOWY



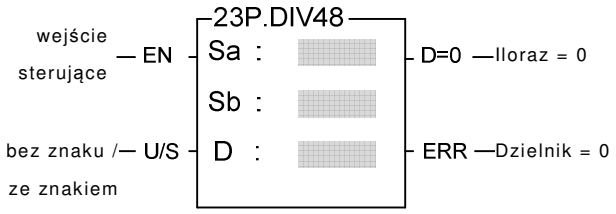
- Jeżeli „EN”=1 lub zmieni się z 0→1（instrukcja P), nastąpi zakończenie pętli programu FOR iNEXT
- Program w pętli FOR-NEXT jest wykonywany N-krotnie (wartość N określana jest argumentem instrukcji FOR). W razie potrzeby zakończenia pętli w cyklu wcześniejszym niż N-ty, należy zastosować instrukcję BREAK
- Instrukcja BREAK musi znajdować się wewnątrz pętli FOR-NEXT



Opis：Operacja zliczania pętli wykorzystywana do realizowania pętli programu FOR-NEXT wyznaczana jest przez rejestr D10. Program w pętli FOR-NEXT, przeznaczony jest do wyszukiwania identycznych danych zapisanych w D100, jak dane w tabeli rejestrów rozpoczynającej się od R0. W przypadku znalezienia takich danych, pętla wyszukiwania zostanie zakończona, a wykonywanie programu będzie kontynuowane za instrukcją NEXT. Jeżeli dane nie zostaną znalezione, pętla zostanie zrealizowana N-krotnie (wartość N zapisana jest w D10) i dopiero wówczas, wykonywanie programu będzie kontynuowane za instrukcją NEXT.

FUN 23 P DIV48	DZIELENIE 48-BITOWE	FUN 23 P DIV48

SYMBOL DRABINKOWY



Sa : Rejestr dzielnej
Sb : Rejestr dzielnika
D : Rejestr do zapisywania wyniku dzielenia (ilorazu)
Sa、Sb、mogą łączyćsię z V, Z, P0~P9 w celu realizacji adresowania pośredniego.

Zakres / Argument	HR	OR	SR	ROR	DR	XR
	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
Sa	○	○	○	○	○	○
Sb	○	○	○	○	○	○
D	○	○	○*	○*	○	○

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to nastąpi 48-bitowa operacja dzielenia. Dzielna i dzielnik tworzone są przez trzy kolejne rejestry, poczynając odpowiednio od Sa i Sb. Jeżeli wynikiem będzie zero, to wyjście 'D=0' zostanie ustawione na 1. Jeżeli dzielnik będzie zerem, to 'ERR' zostanie ustawiony na 1, a wynikowy rejestr pozostanie niezmieniony.
- Wszystkie argumenty związane z tą funkcją są 48-bitowe, dlatego też Sa, Sb i D składają się z 3 kolejnych rejestrów.

Przykład: dzielenie 48-bitowe

W przykładzie tym, dzielna tworzona przez rejestry R2, R1, R0 zostanie podzielona przez dzielnik tworzony przez rejestry R5, R4, R3. Iloraz zostanie zapisany w rejestrze R8, R7 i R6.

Sa

R2	R1	R0
2147483647		

÷ Sb

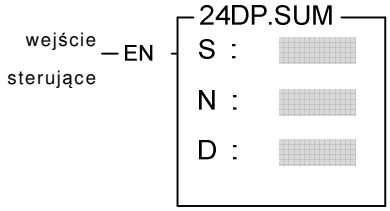
R5	R4	R3
1234567		

R8	R7	R6
1739		

Iloraz

FUN 24DP SUM	SUM (Sumowanie danych)	FUN 24DP SUM

SYMBOL DRABINKOWY



S : Numer początkowy rejestru źródłowego

N : Liczba rejestrów do zsumowania
(kolejne N danych rozpoczynających się od S)

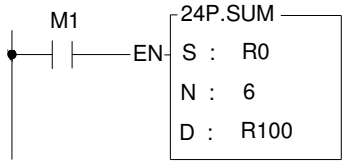
D : Rejestr, w którym zostanie zapisany wynik (suma)

S, N, D mogą łączyć się ze wskaźnikamiV, Z, P0~P9 w celu realizacji adresowania pośredniego.

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	1 □ 511	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to nastąpi dodanie kolejnych N jednostek rejestrów 16-bitowych lub 32-bitowych (instrukcja D) i zapisanie wyników w rejestrze opisanym literą D.
- Jeżeli wartość N jest równa 0 lub jest większa od 511, to operacja nie zostanie przeprowadzona.
- Port komunikacyjny 1~4 może być wykorzystywany jako interfejs komunikacyjny ASCII ogólnego zastosowania. Jeżeli metodą wykrywania błędów danych jest suma kontrolna, to instrukcja ta może być wykorzystana do wygenerowania sumy dla wysyłanych danych lub do sprawdzenia, czy otrzymane dane są błędne.

〈Przykład1〉 Kiedy M1 zmieni stan zWYŁ→WŁ, instrukcja wykona obliczenie sumydanych 16-bitowych.

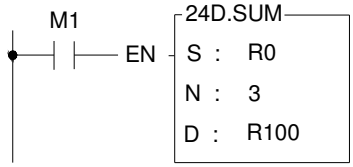


- Na przykładzie po lewej widać, że sumowanych jest 6 rejestrów 16-bitowych rozpoczynających się od R0, a wynik zapisywany jest w rejestrze R100.

R0=0030H
R1=0031H
R2=0032H
R3=0033H
R4=0034H
R5=0035H

➔ R100=012FH

〈Przykład 2〉 Kiedy M1 jest w stanieWŁ, następuje sumowanie danych 32-bitowych.



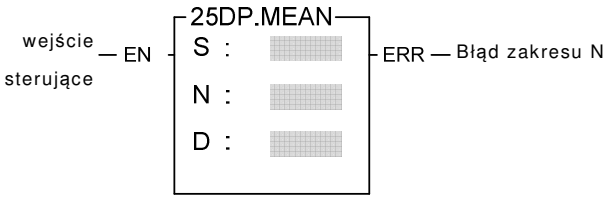
- Na przykładzie po lewej widać, że zsumowane są trzy rejestry 32-bitowe rozpoczynające się od DR0, a wynik zapisywany jest w rejestrze DR100.

R1、R0=00310030H
R3、R2=00330032H
R5、R4=00410039H

➔ R101、R100=00A5009BH

FUN 25DP MEAN	ŚREDNIA (Obliczenie średniej)	FUN 25DP MEAN

SYMBOL DRABINKOWY



S : Numer rejestru źródłowego

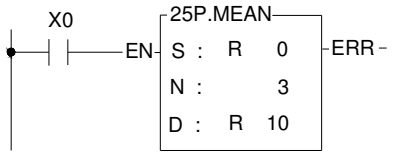
N : Liczba rejestrów do uśrednienia
(N jednostek kolejnych rejestrów zaczunających się od S)

D : Numer rejestru do zapisywaniawyniku (średnia)

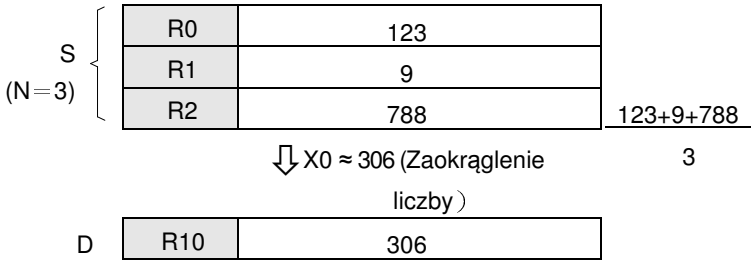
S, N, D mogą łączyć się z V, Z, P0~P9 w celu realizacji adresowania pośredniego.

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to nastąpi dodanie kolejnych N wartości numerycznych 16-bitowych lub 32-bitowych (instrukcja D) rozpoczynających się od S i podzielenie ich przez wartość N. Następnie, wartość średnia (zaokrąglona) zostanie zapisana w rejestrze opisanym literą D.
- Jeżeli wartość N nie mieści się w zakresie od 2 do 256, stan błędu zakresu N "ERR" zostanie ustawiony na 1. W takim przypadku funkcja nie zostanie zrealizowana.

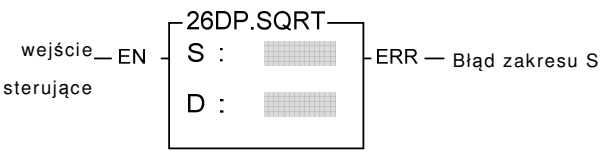


- Na przykładzie po lewej widać, program oblicza średnią wartość z 3 kolejnych rejestrów 16-bitowych rozpoczynających się od R0, a wynik zapisywany jest w rejestrze 16-bitowym R10



FUN 26DP SQRT	Pierwiastek kwadratowy	FUN 26DP SQRT

SYMBOL DRABINKOWY



S : Rejestr źródłowy do obliczeń

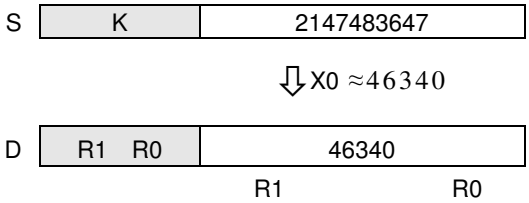
D : Rejestr, w którym zapisywany jest wynik (pierwiastek kwadratowy)

S, D mogą łączyć się z V, Z, P0~P9 w celu realizacji adresowania pośredniego.

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to nastąpi obliczenie pierwiastka kwadratowego(z zaokrągleniem) z danych wymienionych w polu oznaczonym S i zapisanie wyniku w rejestrze D.
- Ponieważ wartość S pochodzi z zawartości rejestru, to jeżeli będzie ona ujemna, stan błędu zakresu S "ERR" zostanie ustawiony na 1. W takim przypadku operacja nie zostanie zrealizowana.

- Instrukcja przedstawiona po lewej stronie oblicza pierwiastek kwadratowy ze stałej wartości 2147483647 i zapisuje wynik w R0.



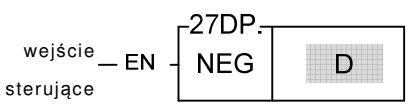
$$\sqrt{2147483647} = 46340.\underline{\underline{95}}$$

↑

zaokrąglenie

FUN 27DP NEG	NEGACJA (obliczenie wartości ujemnej)	FUN 27DP NEG

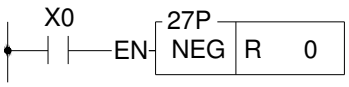
SYMBOL DRABINKOWY



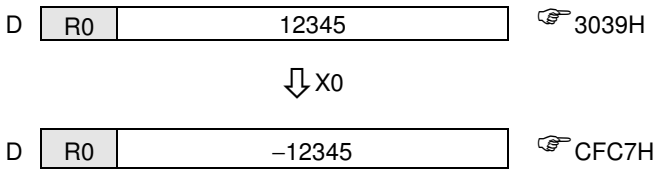
D : Rejestr do obliczeń
D może łączyć się z V, Z, P0~P9 w celu realizacji adresowania pośredniego.

Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to nastąpi zanegowanie (tzn. zostanie obliczona negacja z uzupełnieniem do dwóch) wartości w rejestrze oznaczonym literą D i ponowne zapisanie wyniku w tym rejestrze.
- Jeżeli wartość rejestru D będzie ujemna, to operacja negacji sprawi, że wartość ta stanie się dodatnia.

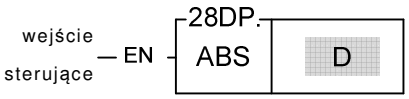


- Instrukcja po lewej neguje wartość rejestru R0 i zapisuje nową wartość w tym rejestrze.



FUN 28DP ABS	WARTOŚĆ BEZWZGLĘDNA (obliczeniewartości bezwzględnej)	FUN 28DP ABS

SYMBOL DRABINKOWY

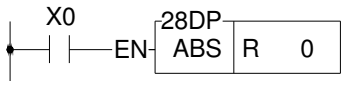


D : Rejestr do obliczenia wartości bezwzględnej

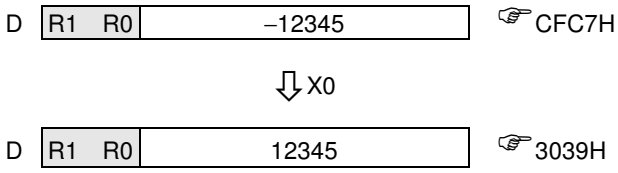
D może łączyć się z V, Z, P0~P9 w celu realizacja adresowania pośredniego

Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to nastąpi obliczenie wartości bezwzględnej zwartości rejestru oznaczonego literą D i ponowne zapisanie wyniku w tym rejestrze.

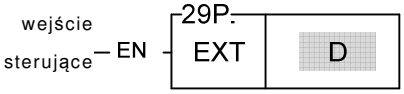


- Instrukcja po lewej oblicza wartość bezwzględną rejestru R0 i zapisuje nową wartość w tym rejestrze



FUN 29DP EXT	ROZSZERZENIE ZNAKOWE	FUN 29DP EXT

SYMBOL DRABINKOWY



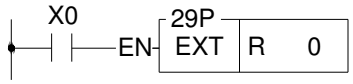
D : Rejestr do przeprowadzenia rozszerzenia znakowego

D może łączyć się z V, Z, P0~P9 w celu realizacji adresowania pośredniego

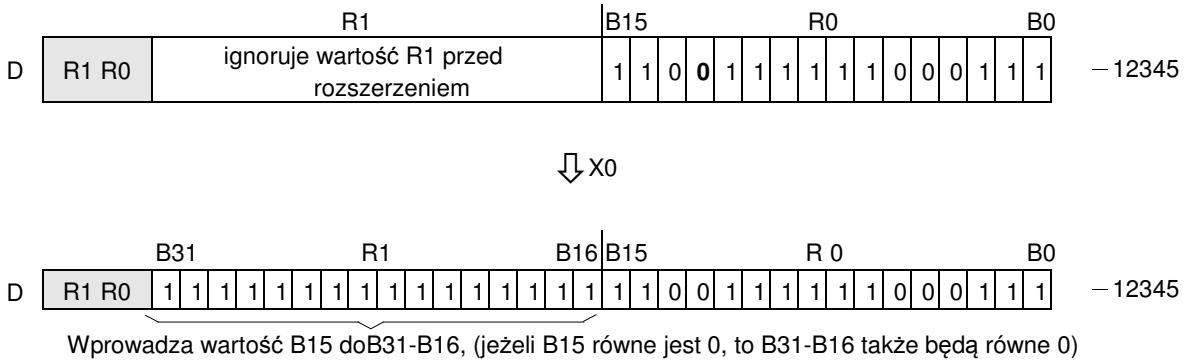
Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to instrukcja ta zrealizuje rozszerzenie znakowe wartości 16-bitowej oznaczonej literą D do wartości 32-bitowej i zapisze nową wartość w rejestrze 32-bitowym składającym się z dwóch kolejnych słów, D + 1 i D.

- Instrukcja ta rozszerza wartość numeryczną rejestru 16-bitowego do odpowiedniej wartości numerycznej rejestru 32-bitowego (na przykład przekształca 33FFH na 000033FFH). Instrukcja wykorzystywana jest głównie w operacjach numerycznych (+,-,*,/,CMP.....), w których można przyjąć 16 i 32-bitowe wartości numeryczne jako argumenty. Przed rozpoczęciem operacji, wszystkie argumenty powinny zostać ustawione na taką samą długość.



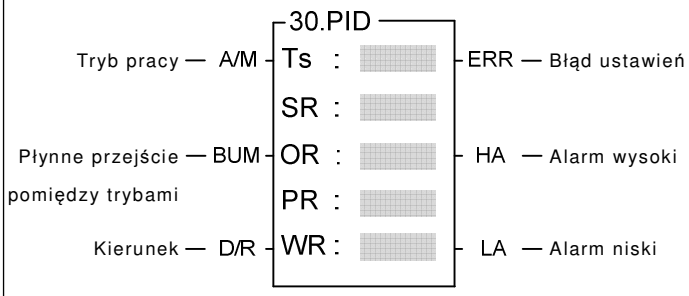
- Instrukcja po lewej rozszerza 16-bitową wartość numeryczną do odpowiedniej wartości 32-bitowej, a następnie zapisuje ją w rejestrze 32-bitowym (DR0=R1R0) zawierającym R0 i R1



Przed rozszerzeniem（16 bitów）R0=CFC7H= -12345
Po rozszerzeniu（32 bitów）R1R0=FFFFCFC7H= -12345
} Obie wartości są w rzeczywistości takie same

FUN 30 PID	OPERACJA PID OGÓLNEGO ZASTOSOWANIA (Krótki opis)	FUN 30 PID

SYMBOL DRABINKOWY



Zakres / Argument	HR	ROR	DR	K
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	
Ts	○	○	○	1～3000
SR	○	○*	○	
OR	○	○*	○	
PR	○	○*	○	
WR	○	○*	○	

Ts : Interwał czasowy dla wykonywania obliczeń PID

SR : Rejestr początkowy tabeli parametrów sterowania składającej się z 8 kolejnych rejestrów.

OR : Rejestr wyjściowy PID

PR : Rejestr początkowy tabeli parametrów procesowychskładającej się z 7 kolejnych rejestrów.

WR : Rejestr początkowy tabeli roboczej dla wewnętrznej operacji PID. Zmienna wymaga (7 rejestrów, które nie mogą być ponownie wykorzystane w innej części programu).

- Funkcja PID (FUN 30) przeprowadza obliczenia oparte na wzorze PID na podstawie aktualnej wartości zmiennej procesowej (PV) wyznaczonej z zewnętrznego sygnału analogowego oraz nastaw (SP) procesu. Wynikiem obliczeń jest wyjściowy sygnał sterujący dla sterowanego procesu, który może zostać podany bezpośrednio do modułu analogowego lub innego interfejsu wyjściowego lub też pozostawiony do realizacji kolejnego procesu. Prawidłowe wykorzystanie sterowania PID w procesie umożliwia otrzymanie szybkiego i dokładnego wyniku PV podążającego za zmianą SP oraz uodparnia proces na zakłócenia.

- Wzór PID w formie cyfrowej:

$$M_n = [(D4005/Pb) \times E_n] + \sum_0^n [(D4005/Pb) \times T_i \times T_s \times E_n] - [(D4005/Pb) \times T_d \times (PV_n - PV_{n-1})/T_s] + Bias$$

M_n : Wyjście sterujące w czasie ”n”

D4005 : Stała wzmocnienia; wartością domyślną jest 1000. Można ją ustawić w zakresie 1~5000.

Pb : Pasmo proporcjonalne (zakres : 2~5000, jednostka 0.1%. Kc (wzmocnienie) =1000/ Pb)

T_i : Całkująca stała czasowa (zakres : 0~9999 odpowiada 0.00~99.99 powtórzeniom/minutę)

T_d : Różniczkująca stała czasowa (zakres : 0~9999 odpowiada 0.00~99.99 minutom)

PV_n : Wartość procesowa w czasie ”n”

PV_{n-1} : Wartość procesowa w czasie ”n-1”

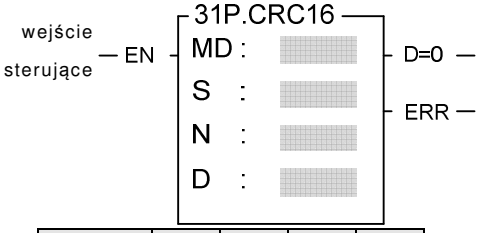
E_n :Błąd w czasie ”n” =nastawa (SP) –wartość procesowa w czasie ”n” (PV_n)

Ts : Przedział czasowy obliczeń PID (zakres: 1~3000, jednostka : 0.01 s)

Bias : Offset wyjścia sterującego (zakres: 0~16380)

FUN31P CRC16	OBLICZANIE CRC16 (CRC16)	FUN31P CRC16

SYMBOL DRABINKOWY



Zakres / Argument	HR	ROR	DR	K
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	
MD				0~1
S	○	○	○	
N	○	○	○	1~256
D	○	○*	○	

MD : 0, Młodszy bit rejestru do obliczenia CRC16
: 1, Zarezerwowane

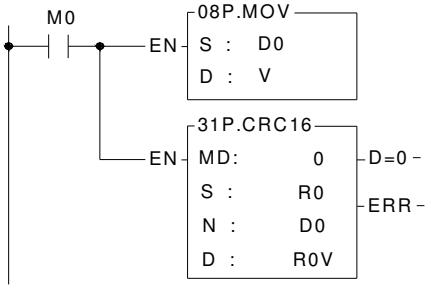
S : Adres początkowydo obliczeń CRC16

N : Długość CRC16 (w bajtach)

D : Rejestr docelowy do zapisu obliczeń CRC16,
W rejestrze D zapisywany jest starszy bajt CRC16
W rejestrze D+1 zapisywany jest młodszy bajtCRC16

S, N, D może łączyć się z rejestrem wskaźnikowym V, Z, P0~P9 w celu realizacji adresowania pośredniego.

- Jeżeli “EN”=1 lub zmieni się z 0→1 (instrukcja P), to instrukcja rozpocznie obliczanie CRC16 od młodszego bajtuS przez długość N. Wynik obliczeń zostanie zapisany w rejestrze D i D+1.
- Wskaźnik wyjściowy "D=0" będzie włączony jeżeli wynik obliczeń będzie równy 0.
- W przypadku nieprawidłowej długości, instrukcja nie przeprowadzi obliczeń i włączy się błąd na wyjściu "ERR".
- Przy komunikacji z inteligentnym urządzeniem peryferyjnym w binarnym formacie danych, bardzo często wykorzystywana jest funkcja CRC16. Metoda ta wykorzystywana jest przez dobrze znany protokół komunikacyjny Modbus RTU.
- CRC16 jest wartością kontrolną przesyłanej ramki danych.
- Po przeprowadzeniu obliczeń CRC16 na podstawie danych otrzymanych w komunikacie oraz wartości kontrolnej, wynik musi wynosić 0. Oznacza to brak błędu w tej ramce wiadomości.



Opis: Jeżeli M0 zmieni się z 0→1, nastąpi obliczenie CRC16 rozpoczynające się od młodszego bajtu R0. Długość oznaczona jest jako D0. Wartość CRC zostanie zapisana w rejestrze R0+V i R0+V+1.
Jeżeli D0=10, to wartość CRC16 zostanie zapisana w rejestrach R10 i R11.

S

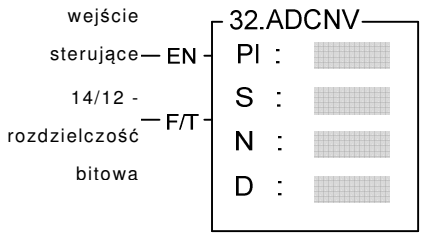
	Starszy bajt	Młodszy bajt
R0	Bez znaczenia	Byte-0
R1	Bez znaczenia	Byte-1
R2	Bez znaczenia	Byte-2
R3	Bez znaczenia	Byte-3
R4	Bez znaczenia	Byte-4
R5	Bez znaczenia	Byte-5
R6	Bez znaczenia	Byte-6
R7	Bez znaczenia	Byte-7
R8	Bez znaczenia	Byte-8
R9	Bez znaczenia	Byte-9

D

	Starszy bajt	Młodszy bajt
R10	00	CRC-Hi
R11	00	CRC-Lo

FUN32 ADCNV	KONWERSJA WARTOŚCI WEJŚCIA ANALOGOWEGO 4～20MA (ADCNV)	FUN32 ADCNV

SYMBOL DRABINKOWY



PI：0, polaryzacja analogowego modułu wejściowego ustawiona jest na unipolarny

：1, polaryzacja analogowego modułu wejściowego ustawiona jest na bipolarny

S：Adres początkowy rejestrów źródłowych

N：Wielkość konwersji (w słowach)

D：Adres początkowy rejestrów docelowych

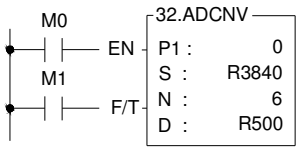
S, N, D może łączyć się z rejestrem współczynnikowym V, Z, P0~P9 w celu realizacji adresowania pośredniego.

Zakres / Argument	HR	IR	ROR	DR	K
	R0 □ R3839	R3840 □ R3903	R5000 □ R8071	D0 □ D4095	
PI					0~1
S	○		○	○	
N	○	○	○	○	1~64
D	○		○*	○	

- Jeżeli jednym z wejść analogowych jest 2～10mA/ 4～20mA/1～5V/2～10V, to sposobem do uzyskania odpowiedniej wartości dla tego sygnału jest analogowy moduł wejściowy. Zakres wejściowy modułu wynosi0 ～10mA/0～5V (Ustawienie przy 5V, polaryzacja unipolarna) lub 0～20mA/0～10V(Ustawienie przy 10V, polaryzacja unipolarna). W takim przypadku wystąpi przesunięcie wartości odczytu. Instrukcja ta stosowana jest w celu eliminacji przesunięcia i konwersji wartości odczytu na wartość w zakresie 0～4095(12-bitów) lub 0～16383(14-bitów).
- Jeżeli "EN"=1, nastąpi konwersja rozpoczynająca się od S, na długości N. Wynik zostanie zapisany w rejestrze D.
- Jeżeli "F/T"=0, wybrana zostanie 12-bitowa rozdzielczość analogowego moduluwejściowego. Jeżeli "F/T"=1, nastąpi przypisanie 14-bitowej rozdzielczości.
- Instrukcja ta nie będzie działać w przypadku nieprawidłowej wartości N.
- Wartość odczytu na wejściu analogowym musi być w formacie -2048~2047 lub -8192~8191, aby konwersja była zgodna. W innym wypadku, jeżeli wartość odczytu będzie w formacie 0~4095 lub 0~16383, konwersja będzie niezgodna.

FUN32 ADCNV	KONWERSJA WARTOŚCI SUROWEJ WEJŚCIA ANALOGOWEGO 4～20MA (ADCNV)	FUN32 ADCNV

Przykład :



Opis : Przy włączonym M0 i wyłączonym M1, przeprowadzona zostanie 6-punktowa konwersja rozpoczynająca się od R3840, podczas której wyeliminowane zostanie przesunięcie wartości odczytu 4～20mA, a odpowiednia wartość 0～4095 zostanie zapisana w R500～R505.

S			D		
R3840	-1229		R500	0	(4 mA)
R3841	409		R501	2047	(12 mA)
R3842	2047	⇨	R502	4095	(20 mA)
R3843	-2048		R503	0	(0 mA)
R3844	-2048		R504	0	(0 mA)
R3845	-2048		R505	0	(0 mA)

Przy włączonych M0 i M1, przeprowadzona zostanie 6-punktowa konwersja rozpoczynająca się od R3840, podczas której wyeliminowane zostanie przesunięcie wartości odczytu 4～20mA, a odpowiednia wartość 0～16383 zostanie zapisana w R500～R505.

S			D		
R3840	-4916		R500	0	(4 mA)
R3841	1637		R501	8191	(12 mA)
R3842	8191	⇨	R502	16383	(20 mA)
R3843	-8192		R503	0	(0 mA)
R3844	-8192		R504	0	(0 mA)
R3845	-8192		R505	0	(0 mA)

FUN33 P LCNV	Konwersja liniowa (LCNV)	FUN33 P LCNV

SYMBOL DRABINKOWY

Wejście sterujące — EN — 33P.LCNV
Md :
S :
Ts :
D :
L :

Md : Tryb pracy , 0～3
S : Początkowy adres danych do przekonwertowania
Ts : Początkowy adres tabeli parametrów konwersji
D : Początkowy adres wyników konwersji
L : Liczba kolejno konwertowanych adresów , 1～64

Argument \ Zakres	HR	IR	ROR	DR	K
	R0 ▯ R3839	R3840 ▯ R3903	R5000 ▯ R8071	D0 ▯ D3999	
Md					0～3
S	○	○	○	○	
Ts	○		○	○	
D	○		○*	○	
L	○		○	○	1～64

- W przypadku, gdy do pomiaru analogowego wykorzystywany jest analogowy moduł wejściowy, wartość odczytana na wejściu analogowym może być przy użyciu tej instrukcji przekonwertowana na zakres jednostek inżynierskich, w celu ich wyświetlenia lub przeprowadzenia operacji sterowania.
- W przypadku kalibracji pomiarowej można zastosować konwersję liniową zmiennej, dzięki czemu zmierzona wartość z PLC może być skorygowana wartością z urządzenia pomiarowego.
- Jeżeli „EN"=1 lub zmieni się z 0→1(instrukcja P), instrukcja przeprowadzi operację konwersji liniowej zgodną z wybranym trybem, gdzie S jest początkowym adresem danych źródłowych, Ts jest początkowym adresem tabeli parametrów konwersji, D jest początkowym adresem do zapisywania przekonwertowanych wyników, a L jest liczbą kolejno konwertowanych adresów źródłowych.
- Istnieją dwa sposoby określania konwersji liniowej:

Sposób 1 : Metoda kalibracji dwupunktowej

Wypełnij tabelę parametrów do konwersji wpisując dolną wartość pomiarową (V_{ML}), górną wartość pomiarową (V_{MH}) oraz odpowiadające im wartości standardowe – dolną (V_{SL}) i górną (V_{SH}). Wynik po konwersji (Dn) wygenerowany zostanie z danych źródłowych (Sn) za pomocą poniższego równania:

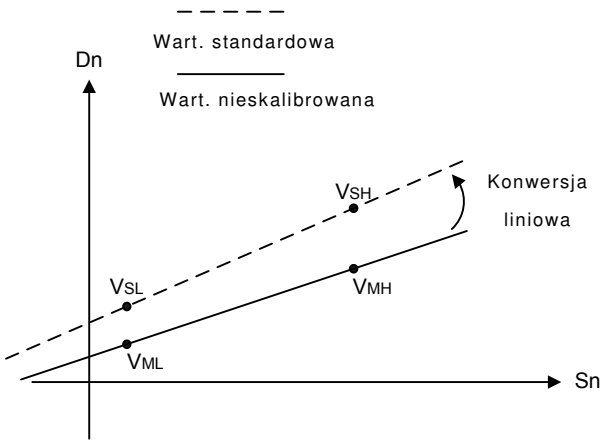
$A = (V_{SL} - V_{SH} / V_{ML} - V_{MH}) \times 10000$

$B = V_{SL} - (V_{ML} \times A / 10000)$

$Dn = (Sn \times A / 10000) + B$

· Zakres argumentów V_{SL},V_{SH}, V_{ML},V_{MH},Sn i Dn wynosi -32768 ~ 32767

· Dla skalowania na wejściu analogowym:
V_{ML}= min. wartość na wejściu analogowym
V_{MH}= maks. wart. na wejściu analogowym
V_{SL}= min. wartość w jednostkach inż.
V_{SH}= maks. wartość w jednostkach inż.



FUN33 P LCNV	Konwersja liniowa (LCNV)	FUN33 P LCNV

Sposób 2 : Mnożnik + metoda przesunięcia

Wypełnij tabelę parametrów do konwersji wpisując wartości mnożnika(A), dzielnika(B) i przesunięcia(C). Wynik po konwersji (Dn) wygenerowany zostanie z danych źródłowych (Sn) za pomocą poniższego równania:

$$Dn = [(Sn \times A) / B] + C$$

Zakres poszczególnych argumentów:

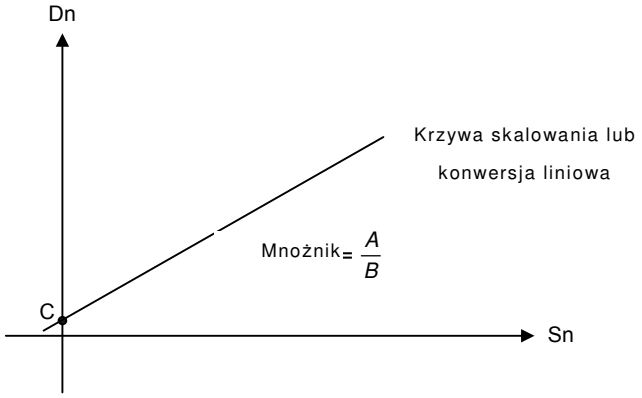
A = 1 ～ 65535

B = 1 ～ 65535

C = -32768 ～ 32767

Sn = 0 ～ 65535

Dn = -32768 ～ 32767



Opis trybu roboczego :

1. Kiedy Md = 0, konwersja liniowa przeprowadzana jest sposobem 1 określania konwersji, a wszystkie dane źródłowe charakteryzują się tymi samymi parametrami do konwersji VML, VMH, VSL i VSH.
2. Kiedy Md = 1, konwersja liniowa przeprowadzana jest sposobem 1 określania konwersji, a dane źródłowe charakteryzują się niezależnymi parametrami do konwersji VML, VMH, VSL, VSH. W przypadku N danych źródłowych, tabela parametrów konwersji powinna zawierać N grup parametrów VML, VMH, VSL, VSH ,tj. N×4 rejestrów.
3. Kiedy Md = 2, konwersja liniowa przeprowadzana jest sposobem 2 określania konwersji, a wszystkie dane źródłowe charakteryzują się tymi samymi parametrami do konwersji A, B i C.
4. Kiedy Md = 3, konwersja liniowa przeprowadzana jest sposobem 2 określania konwersji, a dane źródłowe charakteryzują się niezależnymi parametrami do konwersji A, B, C. W przypadku N danych źródłowych, tabela parametrów do konwersji powinna zawierać N grup parametrów A, B, C, tj. N×3 rejestrów.

FUN33 P LCNV	Konwersja liniowa (LCNV)	FUN33 P LCNV

Przykład programu 1 : Konwersja liniowa w trybie 0

Opis : Kiedy M0=1, konwersja liniowa zostanie zrealizowana w trybie 0, gdzie R100 jest początkowym adresem danych źródłowych, R1000 jest początkowym adresem tabeli parametrów konwersji VML, VMH, VSL, VSH, liczba kolejno konwertowanych wartości wynosi 6, a przekonwertowane wyniki zapisywane będą w R2000～R2005.

Ts

R1000	282	VML
R1001	3530	VMH
R1002	260	VSL
R1003	3650	VSH

	S			D
R100	282		R2000	260
R101	3530		R2001	3650
R102	1906	⇨	R2002	1955
R103	0		R2003	-34
R104	5000		R2004	5184
R105	-115		R2005	-154

FUN33 P LCNV	Konwersja liniowa （LCNV）	FUN33 P LCNV

Przykład programu 2 : Konwersja liniowa w trybie 1

Opis : Kiedy M0=1, konwersja liniowa zostanie zrealizowana w trybie 1, gdzie R100 jest początkowym adresem danych źródłowych, R1000 jest początkowym adresem tabeli parametrów konwersji VML, VMH, VSL, VSH, liczba kolejno konwertowanych wartości wynosi 3, a przekonwertowane wyniki zapisywane będą w R2000～R2002.

Ts

R1000	282	VML_0
R1001	3530	VMH_0
R1002	260	VSL_0
R1003	3650	VSH_0
R1004	-52	VML_1
R1005	1208	VMH_1
R1006	-38	VSL_1
R1007	1101	VSH_1
R1008	235	VML_2
R1009	4563	VMH_2
R1010	264	VSL_2
R1011	4588	VSH_2

S			D	
R100	282	⇨	R2000	260
R101	1208		R2001	1101
R102	2399		R2002	2426

FUN33 P LCNV	Konwersja liniowa (LCNV)	FUN33 P LCNV

Przykład programu 3 : Konwersja liniowa w trybie 2

Opis : Kiedy M0=1, konwersja liniowa zostanie zrealizowana w trybie 2, gdzie R100 jest początkowym adresem danych źródłowych, R1000 jest początkowym adresem tabeli parametrów konwersji A, B, C, liczba kolejno konwertowanych wartości wynosi 6, a przekonwertowane wyniki zapisywane będą w R2000～R2005.

Ts

R1000	985	A
R1001	1000	B
R1002	20	C

S

R100	1000
R101	2345
R102	3560
R103	401
R104	568
R105	2680

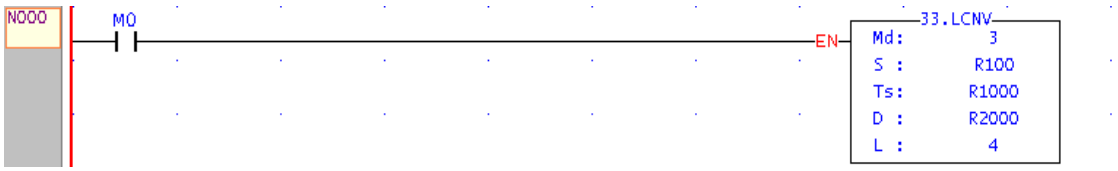
⇨

D

R2000	1005
R2001	2330
R2002	3527
R2003	415
R2004	579
R2005	2660

FUN33 P LCNV	Konwersja liniowa (LCNV)	FUN33 P LCNV

Przykład programu 4 : Konwersja liniowa w trybie 3

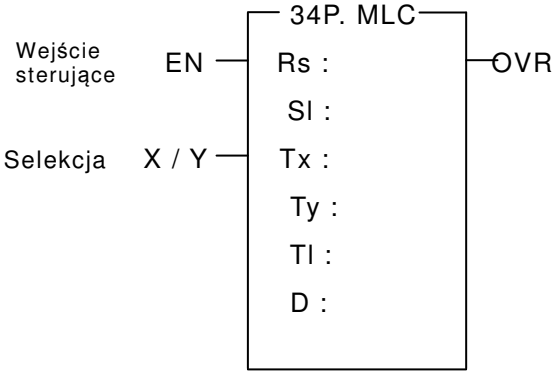


Opis : Kiedy M0=1, konwersja liniowa zostanie zrealizowana w trybie 3, gdzie R100 jest początkowym adresem danych źródłowych, R1000 jest początkowym adresem tabeli parametrów konwersji A, B, C, liczba kolejno konwertowanych wartości wynosi 4, a przekonwertowane wyniki zapisywane będą w R2000～R2003.

	Ts
R1000	5000
R1001	16380
R1002	0
R1003	10000
R1004	16383
R1005	0
R1006	2200
R1007	16380
R1008	-200
R1009	1600
R1010	16383
R1011	-100

	S			D
R100	8192	⇨	R2000	2501
R101	16383		R2001	10000
R102	8190		R2002	900
R103	0		R2003	-100

FUN34 P MLC	Multi-liniowa konwersja (MLC)	FUN34 P MLC



Rs : Początkowy adres danych do przekonwertowania
Sl : Liczba kolejno konwertowanych adresów, 1~64
Tx : Początkowy adres tabeli wartości X
Ty : Początkowy adres tabeli wartości Y
Tl : Rozmiar tabel X,Y, 2~255
D : Początkowy adres wyników konwersji

Argument \ Zakres	HR	IR	ROR	DR	K
	R0 ▯ R3839	R3840 ▯ R3903	R5000 ▯ R8071	D0 ▯ D3999	
Rs	○	○	○	○	
Sl	○		○	○	1~64
x	○		○	○	
Ty	○		○*	○	
Tl	○		○	○	2～255
D	○		○	○	

- W przypadku, gdy do pomiaru analogowego wykorzystywany jest analogowy moduł wejściowy, wartość odczytana na wejściu analogowym może być przy użyciu tej instrukcji przekonwertowana na zakres jednostek inżynierskich, w celu ich wyświetlenia lub przeprowadzenia operacji sterowania.
- W przypadku kalibracji pomiarowej można zastosować multi-liniową konwersję zmiennej, dzięki czemu zmierzona wartość z PLC może być skorygowana wartością z urządzenia pomiarowego.
- Jeżeli „EN"=1 lub zmieni się z 0→1 (instrukcja P), instrukcja przeprowadzi operację multi-liniowej konwersji, w sposób zależny od stanu wejścia X / Y, gdzie Rs jest początkowym adresem danych źródłowych, Sl jest liczbą kolejno konwertowanych adresów źródłowych, Tx jest początkowym adresem tabeli wartości na osi rzędnych X, Ty jest początkowym adresem tabeli wartości na osi odciętych Y, Tl określa rozmiar tabela X / Y, a D jest początkowym adresem do zapisywania wyników konwersji.
- Jeżeli podczas wykonywania instrukcji stan wejścia X / Y = 0, nastąpi porównanie danych źródłowych z wpisami w tabeli Tx w celu odnalezienia odpowiedniej lokalizacji w tej tabeli (wpisy w tabeli Tx muszą być uszeregowane od najmniejszej do największej), a następnie zostanie przeprowadzona konwersja liniowa zgodna ze zlokalizowaną częścią tabel Tx i Ty.
  Jeżeli podczas wykonywania instrukcji stan wejścia X / Y = 1, nastąpi porównanie danych źródłowych z wpisami w tabeli Ty w celu odnalezienia odpowiedniej lokalizacji w tej tabeli (wpisy w tabeli Ty muszą być uszeregowane od najmniejszej do największej, lub odwrotnie), a następnie zostanie przeprowadzona konwersja liniowa zgodna ze zlokalizowaną części tabel Ty i Tx.
- W przypadku, gdy wartości danych źródłowych nie zawierają się w zakresie wartości wpisów w tabeli Tx (gdy X / Y = 0) lub Ty (gdy X / Y = 1), wyjście OVR=1.
- W przypadku nieprawidłowej wartości Sl lub Tl, instrukcja nie zostanie zrealizowana.

FUN34 P MLC	Multi-liniowa konwersja (MLC)	FUN34 P MLC

Wyrażenie:

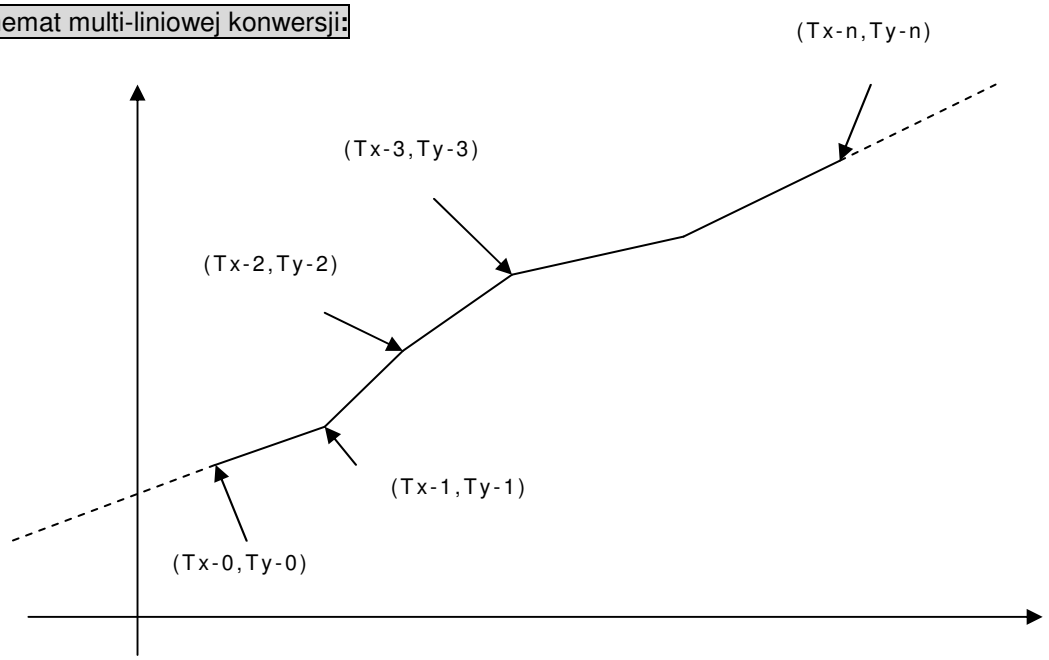
. Aby konwersja liniowa przebiegła prawidłowo, wpisy w tabeli wartości X (Tx) muszą być uporządkowane w kolejności od najmniejszej do największej (gdy X / Y = 0); wpisy w tabeli konwertowanych wartości Y (Ty) mogą być natomiast uporządkowane w kolejności od najmniejszej do największej lub odwrotnie (gdy X / Y = 1). Przy wykonaniu tej instrukcji, nastąpi wyszukanie odpowiedniego fragmentu, poprzez porównanie wpisów tabeli z danymi źródłowymi, a następnie przeprowadzona zostanie konwersja liniowa według następującego równania:

$Vy = (Vx - Tx_n) \times (Ty_n_{+1} - Ty_n / Tx_n_{+1} - Tx_n) + Ty_n$ jeżeli X/Y=0

$Vx = (Vy - Ty_n) \times (Tx_n_{+1} - Tx_n / Ty_n_{+1} - Ty_n) + Tx_n$ jeżeli X/Y=1

.Wartość Vy, V_x, Tx_n, Tx_n_{+1}, Ty_n, Ty_n_{+1} muszą mieścić się w przedziale -32768～32767

Schemat multi-liniowej konwersji:



FUN34 P MLC	Multi-liniowa konwersja (MLC)	FUN34 P MLC

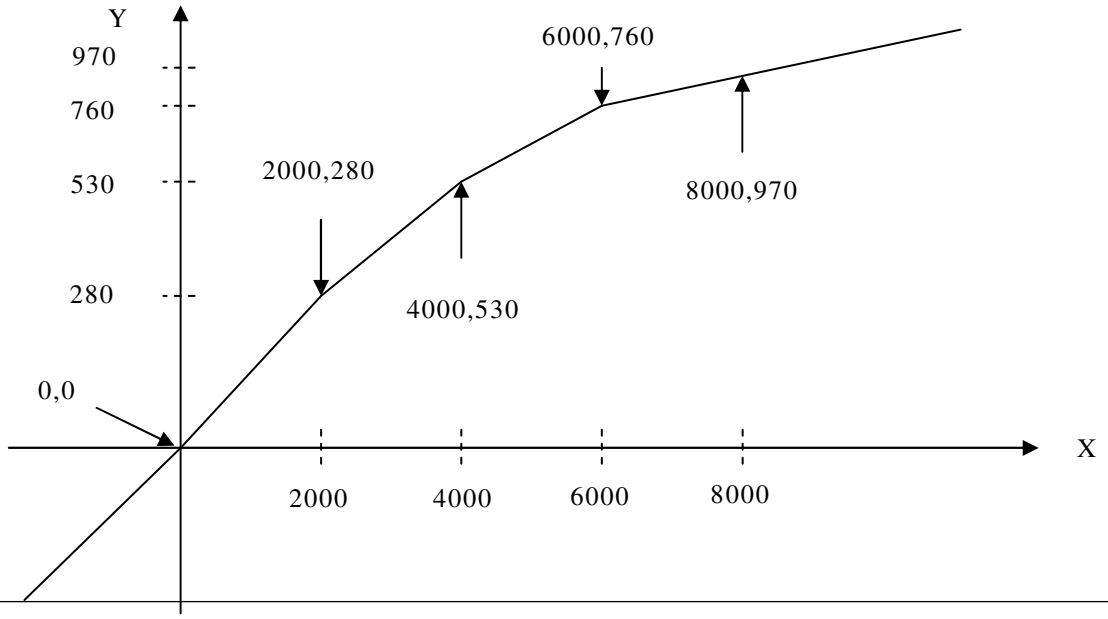
Przykład 1 :

Opis : Kiedy M10=1 i M11=0, a R0 jest początkowym adresem danych do przekonwertowania, R99 jest liczbą kolejnych danych źródłowych do przekonwertowania, R1000 jest początkowym adresem tabeli wartości X (Tx), R2000 jest początkowym adresem tabeli wartości Y (Ty), R199 jest rozmiarem tabel Tx/Ty, dane źródłowe R0~R5 będą poddane konwersji liniowej według tabel Tx i Ty, a wyniki zostaną zapisane w D0～D5.

Status Monitoring

Ref. No.	Status	Data	Ref. No.	Status	Data	Ref. No	Status	Data	Ref. No	Status	Data
R1000	Decimal	0	R2000	Decimal	0	R0	Decimal	1000	D0	Decimal	140
R1001	Decimal	2000	R2001	Decimal	280	R1	Decimal	2500	D1	Decimal	342
R1002	Decimal	4000	R2002	Decimal	530	R2	Decimal	5600	D2	Decimal	714
R1003	Decimal	6000	R2003	Decimal	760	R3	Decimal	7500	D3	Decimal	917
R1004	Decimal	8000	R2004	Decimal	970	R4	Decimal	8000	D4	Decimal	970
R199	Decimal	5				R5	Decimal	10000	D5	Decimal	1180
M10	Enable	ON	M11	Enable	OFF	R99	Decimal	6			

StatusPage0 StatusPage01 StatusPage2



FUN34 P MLC	Multi-liniowa konwersja (MLC)	FUN34 P MLC

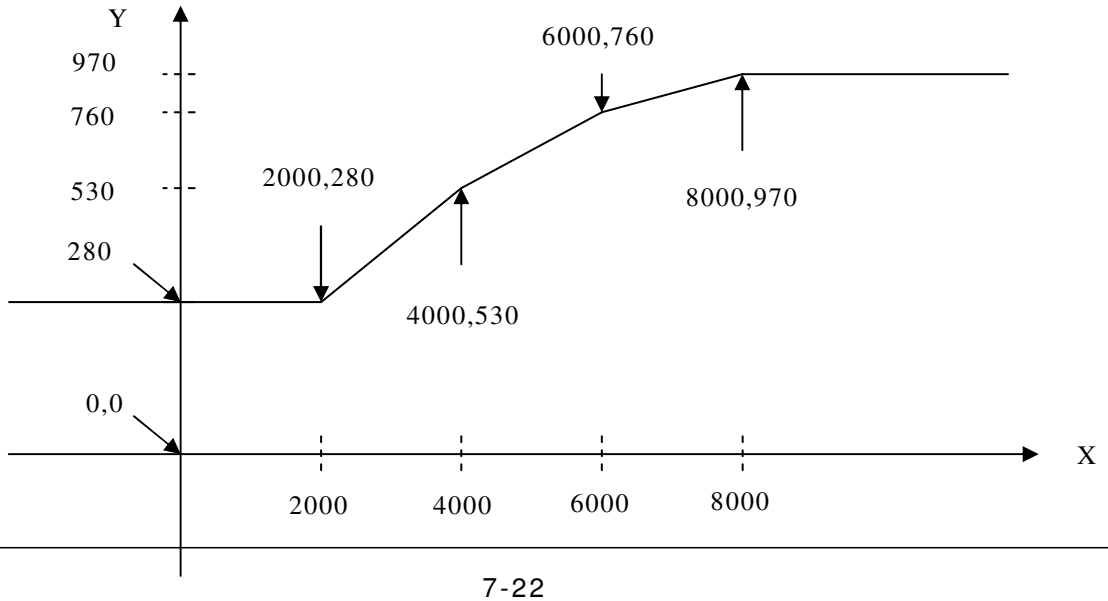
Przykład 2 :

Opis : Kiedy M10=1 i M11=0, a R0 jest początkowym adresem danych do przekonwertowania, R99 jest liczbą kolejnych danych źródłowych do przekonwertowania, R1000 jest początkowym adresem tabeli wartości X (Tx), R2000 jest początkowym adresem tabeli wartości Y (Ty), R199 jest rozmiarem tabel Tx/Ty, dane źródłowe R0~R5 będą poddane konwersji liniowej według tabeli Tx i Ty, a wyniki zostaną zapisane w D0～D5. Jeżeli dane źródłowe ≦ 2000, wartość wynikowa wynosi 280; jeżeli dane źródłowe ≧ 8000, wartość wynikowa wynosi 970.

Status Monitoring

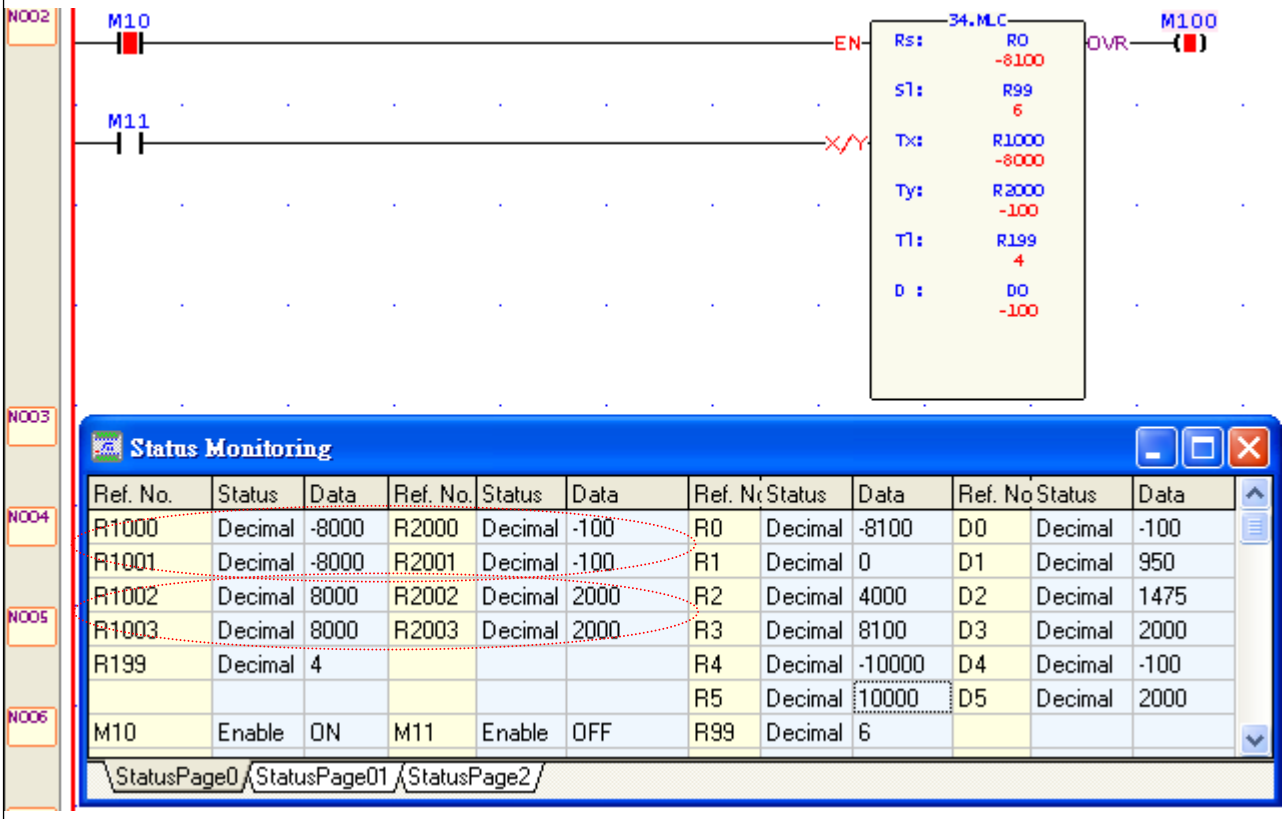
Ref. No.	Status	Data	Ref. No.	Status	Data	Ref. No.	Status	Data	Ref. No.	Status	Data
R1000	Decimal	2000	R2000	Decimal	280	R0	Decimal	1000	D0	Decimal	280
R1001	Decimal	2000	R2001	Decimal	280	R1	Decimal	2000	D1	Decimal	280
R1002	Decimal	4000	R2002	Decimal	530	R2	Decimal	3800	D2	Decimal	505
R1003	Decimal	6000	R2003	Decimal	760	R3	Decimal	7500	D3	Decimal	917
R1004	Decimal	8000	R2004	Decimal	970	R4	Decimal	8000	D4	Decimal	970
R1005	Decimal	8000	R2005	Decimal	970	R5	Decimal	10000	D5	Decimal	970
R199	Decimal	6	R99	Decimal	6	M10	Enable	ON	M11	Enable	OFF

StatusPage0 / StatusPage01 / StatusPage2



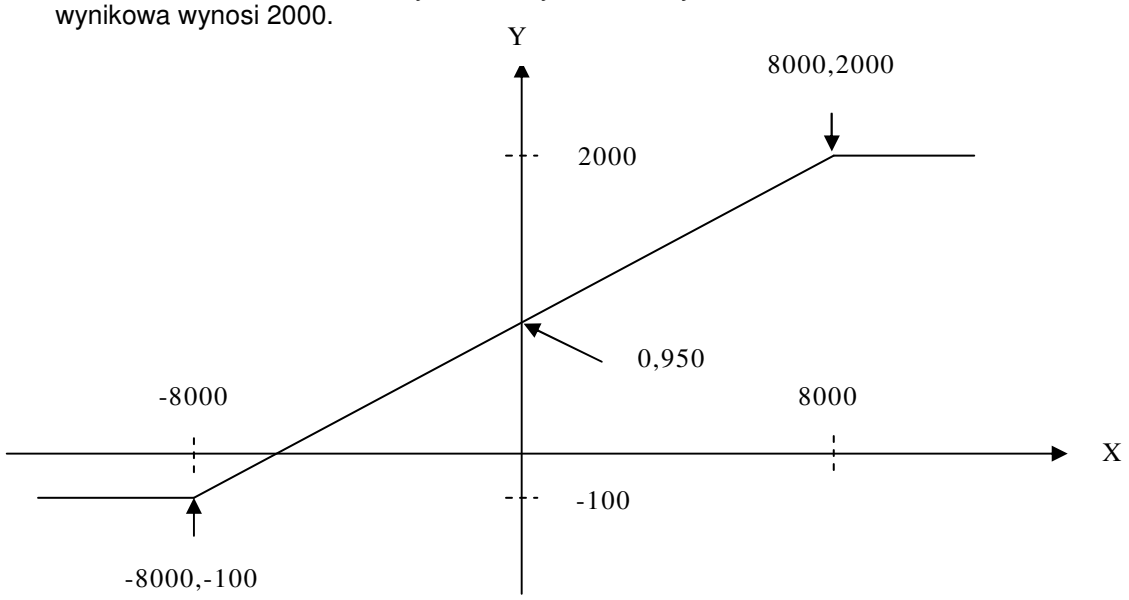
FUN34 P MLC	Multi-liniowa konwersja (MLC)	FUN34 P MLC

Przykład 3 :



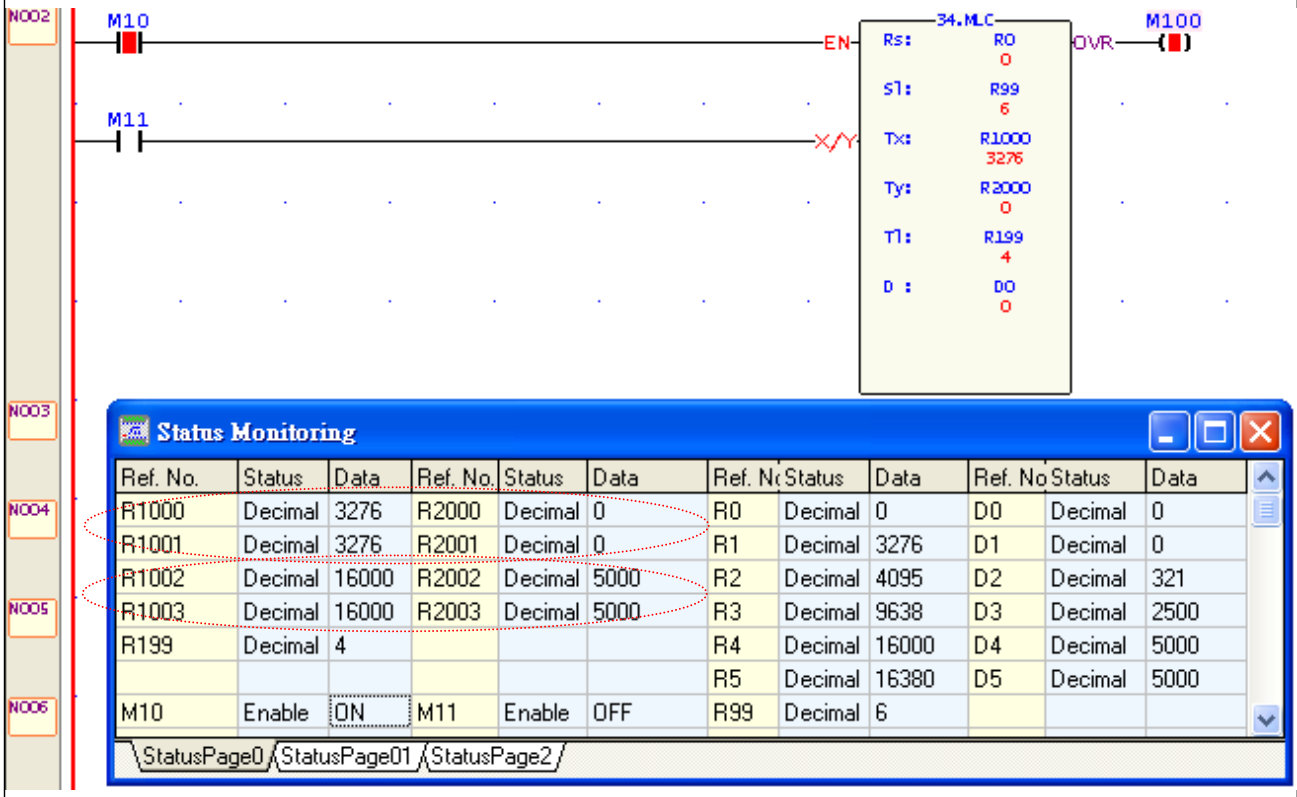
Ref. No.	Status	Data	Ref. No.	Status	Data	Ref. No	Status	Data	Ref. No	Status	Data
R1000	Decimal	-8000	R2000	Decimal	-100	R0	Decimal	-8100	D0	Decimal	-100
R1001	Decimal	-8000	R2001	Decimal	-100	R1	Decimal	0	D1	Decimal	950
R1002	Decimal	8000	R2002	Decimal	2000	R2	Decimal	4000	D2	Decimal	1475
R1003	Decimal	8000	R2003	Decimal	2000	R3	Decimal	8100	D3	Decimal	2000
R199	Decimal	4				R4	Decimal	-10000	D4	Decimal	-100
						R5	Decimal	10000	D5	Decimal	2000
M10	Enable	ON	M11	Enable	OFF	R99	Decimal	6			

Opis : Kiedy M10=1 i M11=0, a R0 jest początkowym adresem danych do przekonwertowania, R99 jest liczbą kolejnych danych źródłowych do przekonwertowania, R1000 jest początkowym adresem tabeli wartości X (Tx), R2000 jest początkowym adresem tabeli wartości Y (Ty), R199 jest rozmiarem tabel Tx/Ty, dane źródłowe R0~R5 będą poddane konwersji liniowej według tabeli Tx i Ty, a wyniki zostaną zapisane w D0～D5. Jeżeli dane źródłowe ≦ -8000, wartość wynikowa wynosi -100; jeżeli dane źródłowe ≧ 8000, wartość wynikowa wynosi 2000.

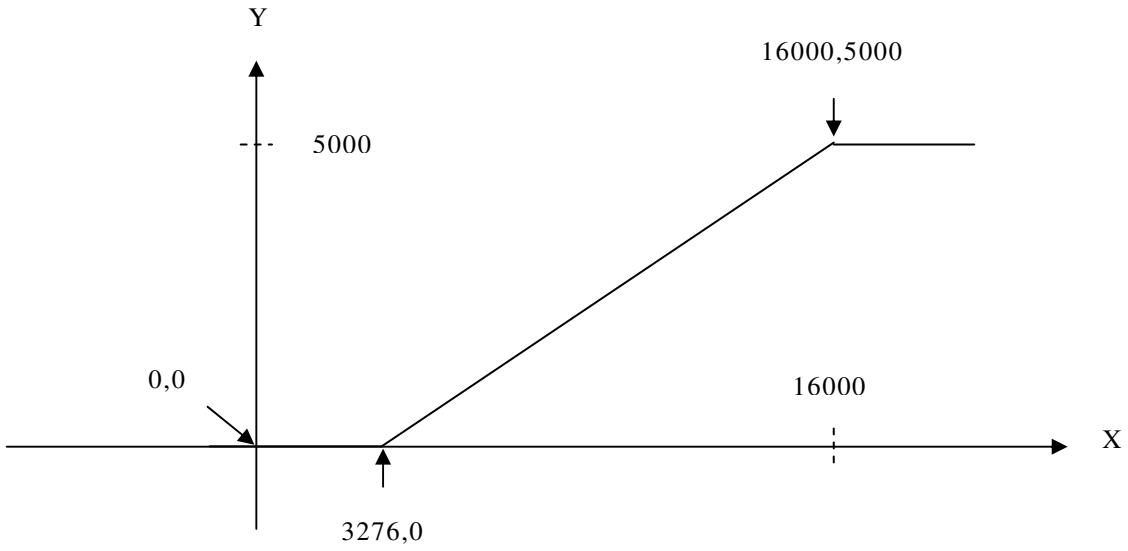


FUN34 P MLC	Multi-liniowa konwersja (MLC)	FUN34 P MLC

Przykład 4 :

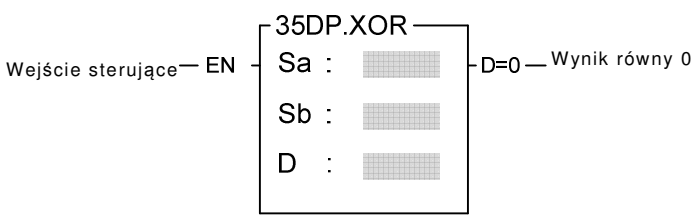


Opis : Kiedy M10=1 i M11=0, a R0 jest początkowym adresem danych do przekonwertowania, R99 jest liczbą kolejnych danych źródłowych do przekonwertowania, R1000 jest początkowym adresem tabeli wartości X (Tx), R2000 jest początkowym adresem tabeli wartości Y (Ty), R199 jest rozmiarem tabel Tx/Ty, dane źródłowe R0~R5 będą poddane konwersji liniowej według tabeli Tx i Ty, a wyniki zostaną zapisane w D0～D5. Jeżeli dane źródłowe ≦ 3276, wartość wynikowa wynosi 0; jeżeli dane źródłowe ≧ 16000, wartość wynikowa wynosi 5000.



FUN 35 D P XOR	XOR	FUN 35 D P XOR

SYMBOL DRABINKOWY



Sa : Dane źródłowe a dla operacji XOR

Sb : Dane źródłowe b dla operacji XOR

D : Rejestr do zapisu wyniku operacji XOR

Sa, Sb, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16/32bit +/- liczba	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli „ EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi realizacja operacji logicznej XOR na danych Sa i Sb. Zadaniem tej operacji jest porównanie odpowiednich bitów Sa i Sb (B0~B15 lub B0~B31) i jeżeli bity na tych samych pozycjach mają różne stany, wtedy bit na tej pozycji w rejestrze wyniku D ustawiany jest w stan 1, w przeciwnym wypadku w stan 0.

- Po zakończeniu operacji, jeżeli wszystkie bity w rejestrze D będą miały stan 0, znacznik „ D = 0” zostanie ustawiony w stan 1.

- Po lewej stronie przedstawiony jest przykład operacji XOR z wykorzystaniem rejestrów R0 i R1, gdzie wynik zapisywany jest w rejestrze R2.

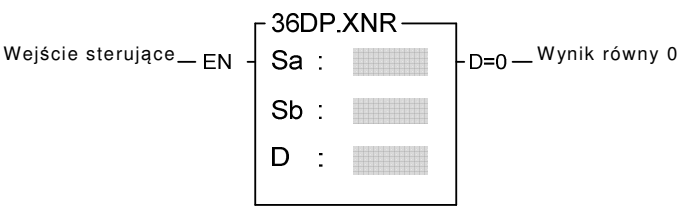
Sa	R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1
Sb	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0

⇩X0= ·

D	R2	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1

FUN 36 D P XNR	XNOR	FUN 36 D P XNR

SYMBOL DRABINKOWY



Sa : Dane źródłowe a dla operacji XNOR

Sb : Dane źródłowe b dla operacji XNOR

D : Rejestr do zapisu wyniku operacji XNOR

Sa, Sb, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresu pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16/32-bit ±liczba	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli „ EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi realizacja operacji logicznej XNR na danych Sa i Sb. Zadaniem tej operacji jest porównanie odpowiednich bitów Sa i Sb (B0~B15 lub B0~B31) i jeżeli bity na tych samych pozycjach mają te same stany, wtedy bit na tej pozycji w rejestrze wyniku D ustawiany jest w stan 1, w przeciwnym wypadku w stan 0.

- Po zakończeniu operacji, jeżeli wszystkie bity w rejestrze D będą miały stan 0, znacznik „ D = 0" zostanie ustawiony w stan 1.

- Po lewej stronie przedstawiony jest przykład operacji XNOR z wykorzystaniem rejestrów R0 i R1, gdzie wynik zapisywany jest w rejestrze R2.

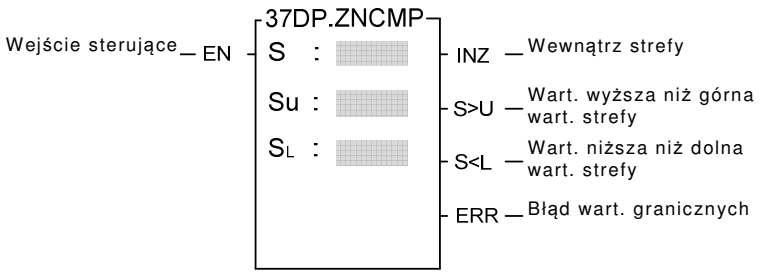
Sa	R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1
Sb	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0

⇩X0 = ·

D	R2	1	0	1	0	1	0	1	0	0	0	1	1	0	1	0	0

FUN 37 D P ZNCMP	Komparator strefy (ZONE COMPARE)	FUN 37 D P ZNCMP

SYMBOL DRABINKOWY



S : Rejestr do porównania: czy w strefie

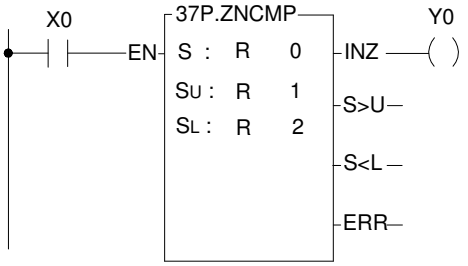
SU : Górna wartość graniczna strefy

SL : Dolna wartość graniczna strefy

S, SU, SL mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
SU	○	○	○	○	○	○	○	○	○	○	○	○	○	○
SL	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Jeżeli „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi porównanie wartości S z górną wartością graniczną S_U i dolną wartością graniczną S_L. W przypadku, gdy wartość S będzie mieścić się pomiędzy górną i dolną wartością graniczną($S_L \leqq S \leqq S_U$), wówczas znacznik „wewnątrz strefy" „INZ" zostanie ustawiony na 1. Jeżeli wartość S będzie wyższa niż górna wartość graniczna S_U, znacznik przekroczenia górnej wartości granicznej „S > U" zostanie ustawiony na 1. Jeżeli wartość S będzie niższa niż dolna wartość graniczna S_L, znacznik przekroczenia dolnej wartości granicznej „S < L" zostanie ustawiony na 1
- Górna wartość graniczna S_U powinna być większa niż dolna wartość graniczna S_L. Jeżeli $S_U < S_L$, znacznik błędu wartości granicznej „ERR" zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.



- Po lewej stronie przedstawiony jest przykład instrukcji, która porównuje wartość R0 z górną i dolną wartością graniczną ustanowioną odpowiednio przez R1 i R2. W przypadku, gdy wartości R0~R2 są takie jak w lewej dolnej tabeli, wynik będzie analogiczny jak po prawej stronie tabeli.
- Aby uzyskać informację o tym, że wartość S jest spoza strefy, można wyjścia „S > U" i „S < L" sprawdzić warunkiem LUB. Alternatywnie na wyjściu „INZ" można zastosować wyjście zanegowane OUT NOT i przypisać je adresowi Y0.

S	R0	200	
SU	R1	300	(Górna wart. graniczna)
SL	R2	100	(Dolna wart. graniczna)

Przed wykonaniem

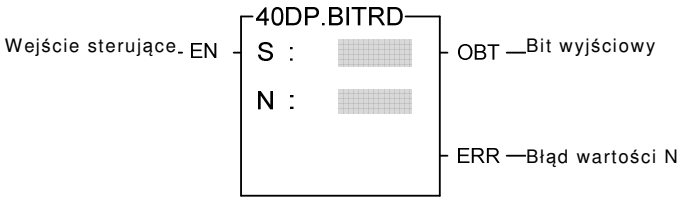
X0 = •

Y0: 1

Wynik wykonania funkcji

FUN 40 D P BITRD	ODCZYTAJ BIT (BIT READ)	FUN 40 D P BITRD

SYMBOL DRABINKOWY



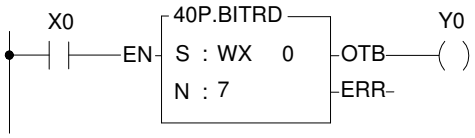
S : Rejestr źródłowy do odczytu

N : Numer bitu do odczytu z S.

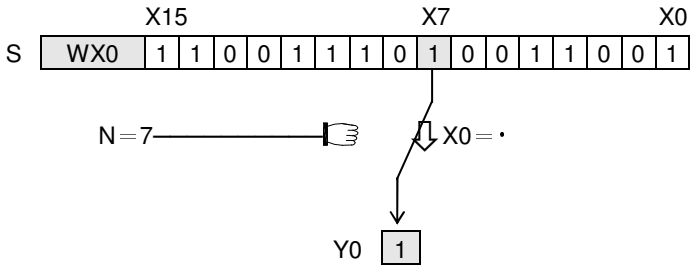
S, N mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi odczytanie stanu N-tego bitu z rejestru danych S i przypisanie go stanowi wyjścia OTB.
- Jeżeli wejście sterujące „EN" = 0, stan wyjścia OTB może: zachowywać stan poprzedniego wywołania (jeżeli M1919=0) lub być ustawiane w stan niski (jeżeli M1919=1).
- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla N jest 0~15. Dla argumentu 32-bitowego (instrukcja D), prawidłowym zakresem dla N jest 0~31. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

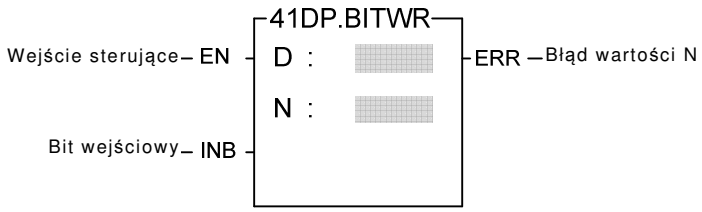


- Po lewej stronie przedstawiony jest przykład instrukcji odczytującej status 7-go bitu (X7) z WX0 (X0~X15) i przenoszącej stan tego bitu na Y0. Wyniki są następujące:



FUN 41 D P BITWR	ZAPISZ BIT (BIT WRITE)	FUN 41 D P BITWR

SYMBOL DRABINKOWY



D : Rejestr do zapisu bitu

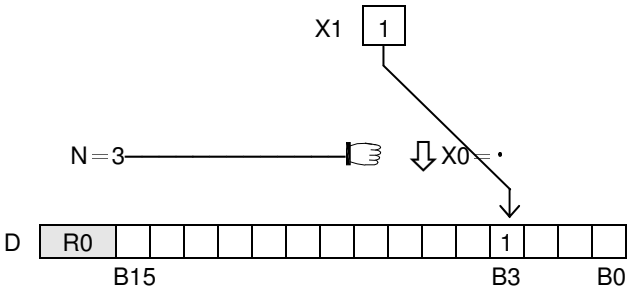
N : Numer bitu rejestru D do zapisu.

D, N mogą łączyć się z V, Z , P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	0 0 □ lub □ 15 31	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi ustawienie N-tego bitu w rejestrze D na stan bitu wejściowego (INB).
- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla N jest 0~15. Dla argumentu 32-bitowego (instrukcja D), prawidłowym zakresem dla N jest 0~31. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

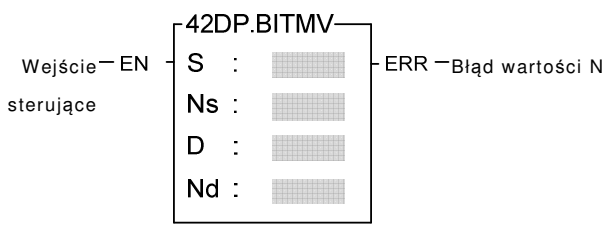
- Po lewej stronie przedstawiony jest przykład instrukcji zapisującej status bitu wejściowego INB do B3 w R0. Zakładając, że X1 = 1, wynik będzie następujący:



Wszystkie bity poza B3 pozostają niezmienione

FUN 42 D P BITMV	PRZENIEŚ BIT (BIT MOVE)	FUN 42 D P BITMV

SYMBOL DRABINKOWY



S : Rejestr źródłowy bitu do przeniesienia

Ns : Numer bitu w rejestrze S do przeniesienia

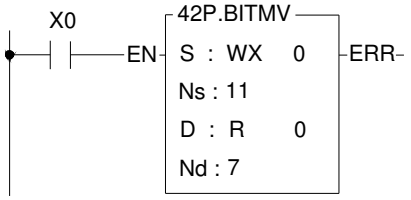
D : Rejestr docelowy do przeniesienia bitu

Nd : Numer bitu docelowego w rejestrze D

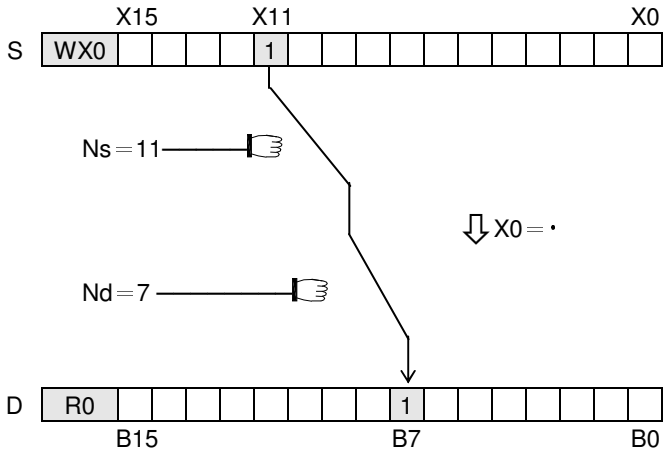
S, Ns, D, Nd mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi przeniesienie stanu Ns-tego bitu z rejestru S na stan Nd-tego bitu w rejestrze D.
- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla N jest 0~15. Dla argumentu 32-bitowego (instrukcja D), prawidłowym zakresem dla N jest 0~31. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

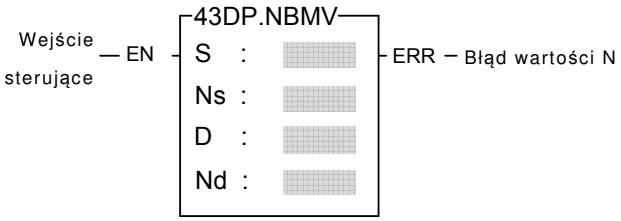


- Po lewej stronie przedstawiony jest przykład instrukcji przenoszącej status bitu B11 (X11) z rejestru S na pozycję B7 w rejestrze D. Wszystkie bity w rejestrze D, poza B7, pozostają niezmienione.



FUN 43 D P NBMV	PRZENIEŚ PÓŁBAJT (NIBBLE MOVE)	FUN 43 D P NBMV

SYMBOL DRABINKOWY



S : Rejestr źródłowy do przeniesienia półbajtu

Ns: Numer półbajtu w rejestrze S do przeniesienia

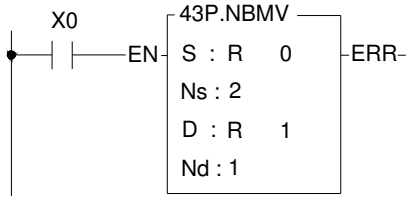
D : Rejestr docelowy do przeniesienia półbajtu

Nd: Numer półbajtu docelowego w rejestrze D

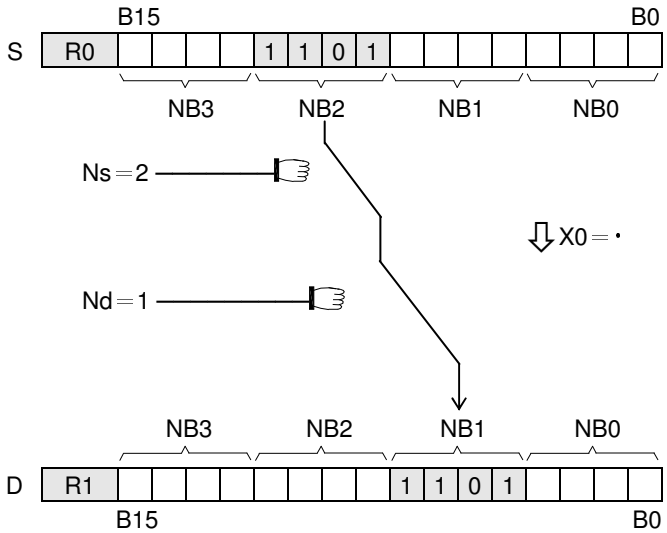
S, Ns, D, Nd mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～7	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～7	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi przeniesienie Ns-tego półbajtu z rejestru S na Nd-ty półbajt w rejestrze D. (Półbajt skalda się z 4 bitów. Rozpoczynając od najniższego bitu w rejestrze B0, każde kolejne 4 bity tworzą półbajt, czyli B0~B3 tworzą półbajt 0, B4~B7 tworzą półbajt 1, itd.)
- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla Ns i Nd jest 0~3. Dla argumentu 32-bitowego (instrukcja D), prawidłowym zakresem dla Ns i Nd jest 0~7. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

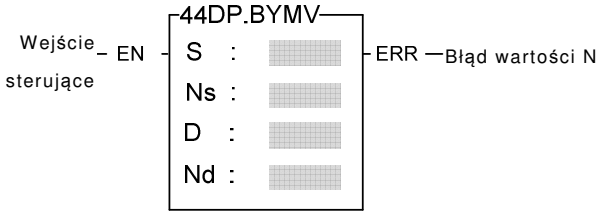


- Po lewej stronie przedstawiony jest przykład instrukcji przenoszącej trzeci półbajt NB2 (B8~B11) z rejestru S na pozycję pierwszego półbajtu NB1 (B4~B7) w rejestrze D. Pozostałe półbajty w rejestrze D pozostają niezmienione.



FUN 44 D P BYMV	PRZENIEŚ BAJT (BYTE MOVE)	FUN 44 D P BYMV

SYMBOL DRABINKOWY



S : Rejestr źródłowy do przeniesienia bajtu

Ns : Numer bajtu w rejestrze S do przeniesienia

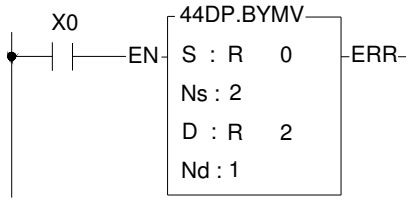
D : Rejestr docelowy do przeniesienia bajtu

Nd : Numer bajtu docelowego w rejestrze D

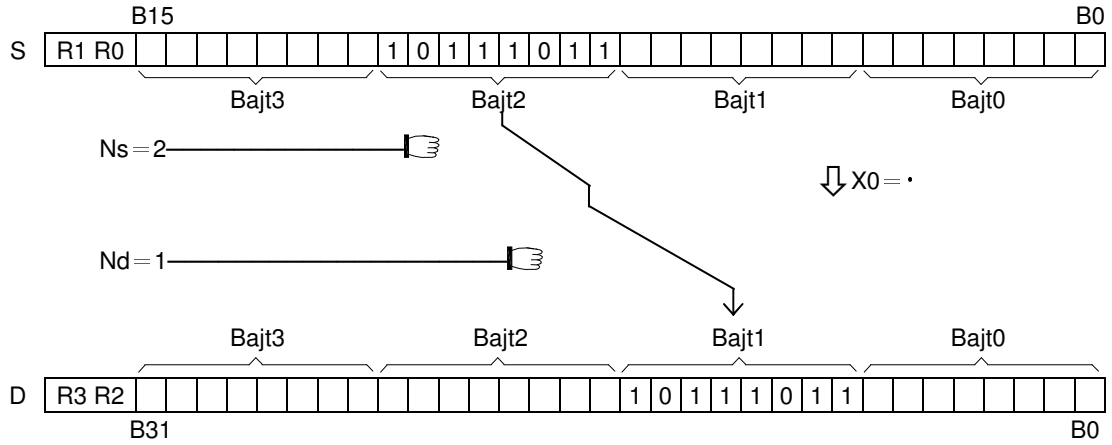
S, Ns, D, Nd mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- numer	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi przeniesienie Ns-tego bajtu z rejestru S na Nd-ty bajt w rejestrze D. (Bajt skalda się z 8 bitów. Rozpoczynając od najniższego bitu w rejestrze B0, każde kolejne 8 bitów tworzy bajt, czyli bity B0~B7 tworzą bajt 0, B8~B15 tworzą bajt 1, itd.)
- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla Ns i Nd jest 0~1. Dla argumentu 32-bitowego (instrukcja D), prawidłowym zakresem dla Ns i Nd jest 0~3. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

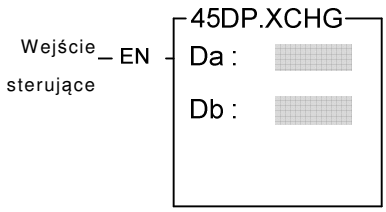


- Po lewej stronie przedstawiony jest przykład instrukcji przenoszącej trzeci bajt (B16~B23) z S (rejestr 32-bitowy złożony z R1 i R0) na pozycję pierwszego bajtu w D (rejestr 32-bitowy złożony z R3 i R2). Pozostałe bajty w rejestrze D pozostają niezmienione



FUN 45 D P XCHG	ZAMIEŃ (EXCHANGE)	FUN 45 D P XCHG

SYMBOL DRABINKOWY



Da : Rejestr a do zamiany

Db : Rejestr b do zamiany

Da, Db mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
Da	○	○	○	○	○	○	○	○*	○*	○	○
Db	○	○	○	○	○	○	○	○*	○*	○	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi zamiana wartości rejestru Da z wartością rejestru Db w formacie 16 lub 32-bitowym (instrukcja D).

- Po lewej stronie przedstawiony jest przykład instrukcji realizującej zamianę wartości w 16-bitowych rejestrach R0 i R1.

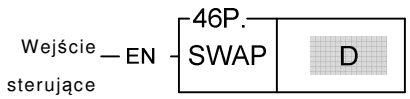
		B15															B0
Da	R0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Db	R1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

⇩ X0 = •

		B15															B0
Da	R0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Db	R1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FUN 46 P SWAP	ZAMIEŃ BAJTY (BYTE SWAP)	FUN 46 P SWAP

SYMBOL DRABINKOWY

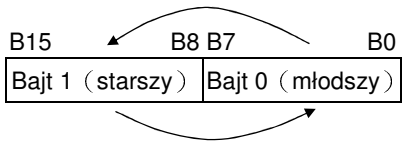


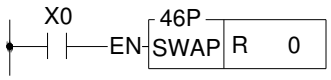
D : Rejestr do zamiany bajtów

D może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

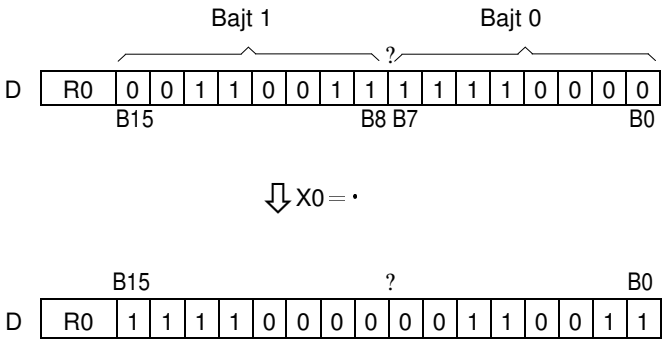
Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi zamiana danych młodszego bajtu 0 (B0~B7) ze starszym bajtem 1 (B8~B15) w 16-bitowym rejestrze D.



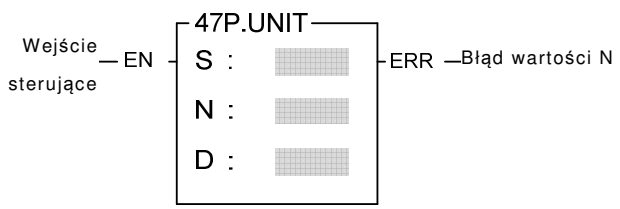


- Po lewej stronie przedstawiony jest przykład instrukcji zamieniającej dane młodszego bajtu (B0~B7) ze starszym bajtem (B8~B15) w R0. Wyniki są następujące:



FUN 47 P UNIT	POŁĄCZ PÓŁBAJTY (NIBBLE UNITE)	FUN 47 P UNIT

SYMBOL DRABINKOWY



S : Początkowy adres rejestrów źródłowych do połączenia bółbajtów

N : Liczba półbajtów do połączenia

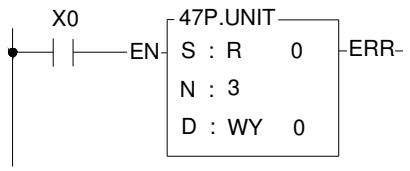
D : Rejestr przechowujący połączone półbajty

S, N, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

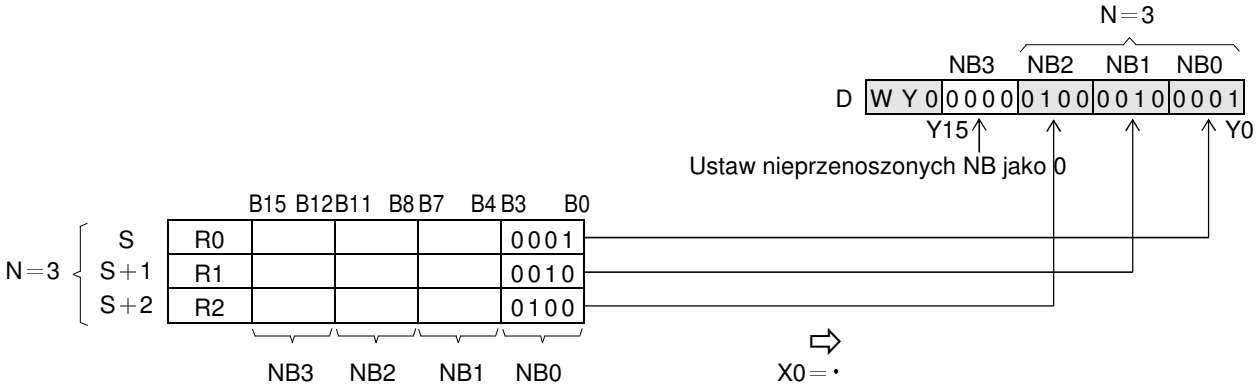
Zakres / Arg.	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	1 □ 4	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), najmłodsze półbajty (NB0) z N kolejnych adresów rejestrów rozpoczynając od S, zostaną przeniesione do NB0, NB1, ...NBn-1 w rejestrze D, w kolejności od najmłodszego do najstarszego. Półbajty nieprzenoszone do rejestru D (w przypadku N<4), uzupełniane są stanami 0. (Półbajt skalda się z 4 bitów. Rozpoczynając od najniższego bitu w rejestrze B0, każde kolejne 4 bity tworzą półbajt, czyli B0~B3 tworzą półbajt 0, B4~B7 tworzą półbajt 1, itd.)

- Instrukcja ta obsługuje tylko argumenty typu WORD (16-bitowe). Dlatego też, w operacji uczestniczą maksymalnie 4 półbajty, a co za tym idzie, prawidłowym zakresem dla N jest 1~4. Liczba N wychodząca poza ten zakres spowoduje ustawienie znacznika błędu wartości N („ERR") na 1, a instrukcja nie zostanie wykonana.



- Po lewej stronie przedstawiony jest przykład instrukcji przenoszącej NB0 z 3 kolejnych rejestrów R0, R1 i R2 do NB0~NB2 w rejestrze WY0.



FUN 48 P DIST	ROZPORWADŹ PÓŁBAJTY (NIBBLE DISTRIBUTE)	FUN 48 P DIST

SYMBOL DRABINKOWY

S : Rejestr źródłowy do rozprowadzenia półbajtów

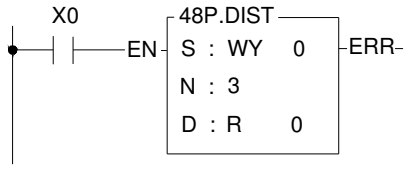
N : Liczba półbajtów do rozprowadzenia

D : Początkowy adres rejestrów przechowujących rozprowadzone półbajty

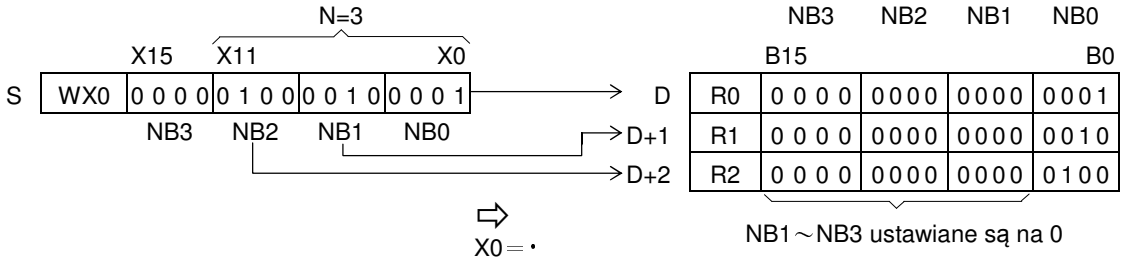
S, N, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Arg.	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	1~4	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), N kolejnych półbajtów rozpoczynając od najmłodszego półbajtu NB0 w rejestrze S, zostanie rozprowadzone, w kolejności od najmłodszego do najstarszego, na 0-owe półbajty (NB0) w N kolejnych rejestrach rozpoczynając od D. Pozostałe pólbajty (inne niż NB0), w każdym z rejestrów D, ustawiane są na zera. (Półbajt skalda się z 4 bitów. Rozpoczynając od najniższego bitu w rejestrze B0, każde kolejne 4 bity tworzą półbajt, czyli B0~B3 tworzą półbajt 0, B4~B7 tworzą półbajt 1, itd.).
- Instrukcja ta obsługuje tylko argumenty typu WORD (16-bitowe). Dlatego też, w operacji uczestniczą maksymalnie 4 półbajty, a co za tym idzie, prawidłowym zakresem dla N jest 1~4. Liczba N wychodząca poza ten zakres spowoduje ustawienie znacznika błędu wartości N („ERR") na 1, a instrukcja nie zostanie wykonana.

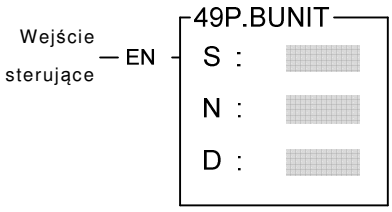


- Po lewej stronie przedstawiony jest przykład instrukcji przenoszącej NB0~NB2 z rejestru WX0 na miejsce NB0 w 3 kolejnych rejestrach R0~R2.



FUN49 P BUNIT	POŁĄCZ BAJTY (BYTE UNITE)	FUN49 P BUNIT

SYMBOL DRABINKOWY



S : Adres początkowy rejestrów źródłowych do połączenia bajtów

N : Liczba kolejnych adresów S do połączenia bajtów

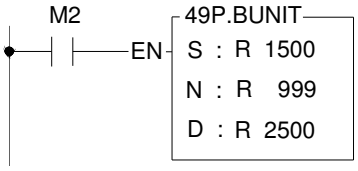
D : Adres początkowy rejestrów do przechowywania połączonych bajtów

S, N, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	
S	○	○	○	
N	○	○	○	1~256
D	○	○*	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi połączenie młodszych bajtów, pochodzących z N kolejnych rejestrów rozpoczynających się od S. Wyniki połączenia zostaną zapisane w rejestrach D.
- Jeżeli liczba łączonych bajtów będzie nieprawidłowa, instrukcja nie zostanie wykonana.
- Przy komunikacji z inteligentnymi urządzeniami peryferyjnymi, z użyciem binarnego formatu danych, instrukcja ta może być wykorzystana do połączenia bajtów pochodzących z kolejnych przesłanych słów.

Przykład :



Opis : Gdy status M2 zmienia się z 0→1, nastąpi połączenie młodszych bajtów rejestrów rozpoczynających się od R1500; liczba łączonych bajtów określona jest przez R999. Wyniki zapisane zostaną w rejestrach rozpoczynających się od R2500.
Jeżeli R999=10, wyniki zostaną zapisane w rejestrach R2500～R2504.

S

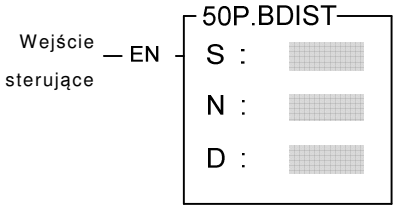
	Starszy bajt	Młodszy bajt
R1500	Nieistotne	Bajt-0
R1501	Nieistotne	Bajt-1
R1502	Nieistotne	Bajt-2
R1503	Nieistotne	Bajt-3
R1504	Nieistotne	Bajt-4
R1505	Nieistotne	Bajt-5
R1506	Nieistotne	Bajt-6
R1507	Nieistotne	Bajt-7
R1508	Nieistotne	Bajt-8
R1509	Nieistotne	Bajt-9

D

	Starszy bajt	Młodszy bajt
R2500	Bajt-0	Bajt-1
R2501	Bajt-2	Bajt-3
R2502	Bajt-4	Bajt-5
R2503	Bajt-6	Bajt-7
R2504	Bajt-8	Bajt-9

FUN50 P BDIST	ROZPROWADŹ BAJTY (BYTE DISTRIBUTE)	FUN50 P BDIST

SYMBOL DRABINKOWY



S : Adres początkowy rejestru źródłowego do rozprowadzenia bajtów

N : Liczba kolejnych bajtów do rozprowadzenia

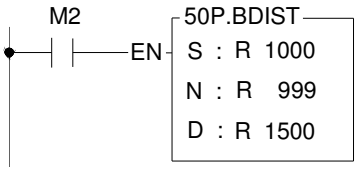
D : Adres początkowy rejestrów do przechowywania rozprowadzonych bajtów

S, N, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres Arg.	HR	ROR	DR	K
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	
S	○	○	○	
N	○	○	○	1~256
D	○	○*	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi rozprowadzenie N bajtów , pochodzących z kolejnych rejestrów rozpoczynających się od S. Wyniki zostaną zapisane na młodszych bajtach rejestrów D (starsze bajty zostaną uzupełnione zerami).
- Jeżeli liczba rozprowadzanych bajtów będzie nieprawidłowa, instrukcja nie zostanie wykonana.
- Przy komunikacji z inteligentnymi urządzeniami peryferyjnymi, z użyciem binarnego formatu danych, instrukcja ta może być zastosowana do rozdziału słów na bajty przed transmisją danych.

Przykład :



Opis : Gdy status M2 zmienia się z 0→1, nastąpi rozdział bajtów rozpoczynając od R1000; liczba bajtów określona jest przez R999. Wyniki zapisane zostaną w rejestrach rozpoczynając od R1500.
Jeżeli R999=9, wyniki zostaną zapisane w rejestrach R1500～R1508.

S

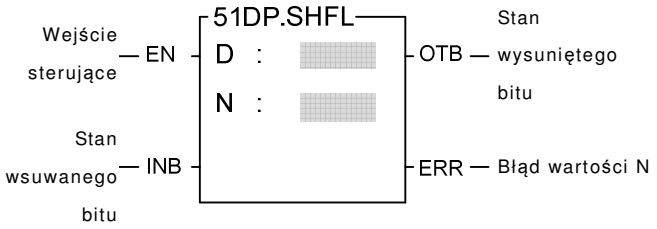
	Starszy bajt	Młodszy bajt
R1000	Bajt-0	Bajt-1
R1001	Bajt-2	Bajt-3
R1002	Bajt-4	Bajt-5
R1003	Bajt-6	Bajt-7
R1004	Bajt-8	Nieistotne

D

	Starszy bajt	Młodszy bajt
R1500	00	Bajt-0
R1501	00	Bajt-1
R1502	00	Bajt-2
R1503	00	Bajt-3
R1504	00	Bajt-4
R1505	00	Bajt-5
R1506	00	Bajt-6
R1507	00	Bajt-7
R1508	00	Bajt-8

FUN 51 D P SHFL	PRZESUŃ BITY W LEWO (SHIFT LEFT)	FUN 51 D P SHFL

SYMBOL DRABINKOWY



D : Rejestr z danymi do przesunięcia

N : Liczba bitów do przesunięcia

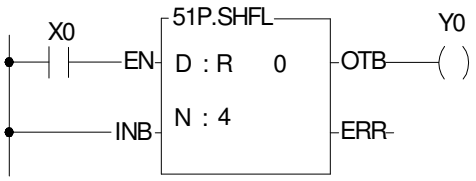
N, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

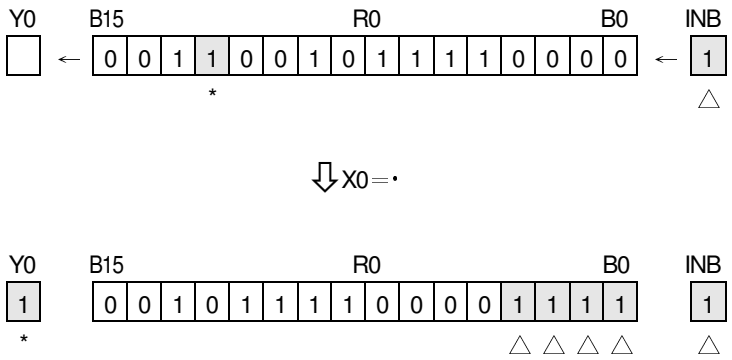
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 1 | lub | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Jeżeli wejście sterujące „ EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi przesunięcie bitów w rejestrze D w lewo o N kolejnych pozycji (w kolejności od najmłodszego do najstarszego). Po przesunięciu w lewo najmłodszych bitów, na ich pozycje zostaną wsunięte stany wejścia INB, a stan wysuniętego najstarszego bitu 15tego lub 31ego (instrukcja D), pojawi się na wyjściu OTB.

- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla N jest 1~16. Dla argumentu 32-bitowego (instrukcja D), poprawnym zakresem dla N jest 1~32. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

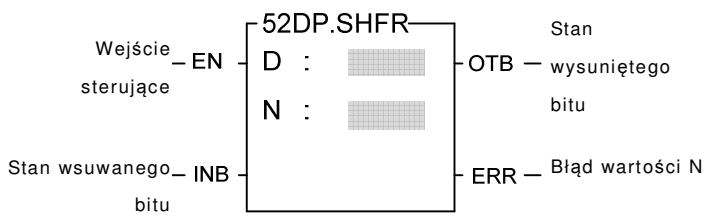
- Po lewej stronie przedstawiony jest przykład instrukcji przesuwającej bity w rejestrze R0 w lewo o 4 kolejne pozycje. Wyniki przedstawione są poniżej.





FUN 52 D P SHFR	PRZESUŃ BITY W PRAWO (SHIFT RIGHT)	FUN 52 D P SHFR

SYMBOL DRABINKOWY



D : Rejestr z danymi do przesunięcia

N : Liczba bitów do przesunięcia

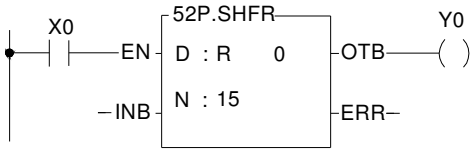
N, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

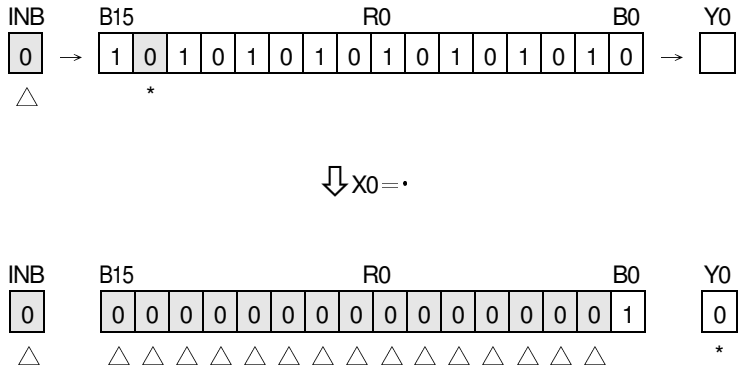
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 1 | lub | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Jeżeli wejście sterujące „ EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi przesunięcie bitów w rejestrze D w prawo o N kolejnych pozycji (w kolejności od najstarszego do najmłodszego). Po przesunięciu w prawo najstarszych bitów, na ich pozycje zostaną wsunięte stany wejścia INB, a stan wysuniętego najmłodszego bitu pojawi się na wyjściu OTB.

- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla N jest 1~16. Dla argumentu 32-bitowego (instrukcja D), poprawnym zakresem dla N jest 1~32. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

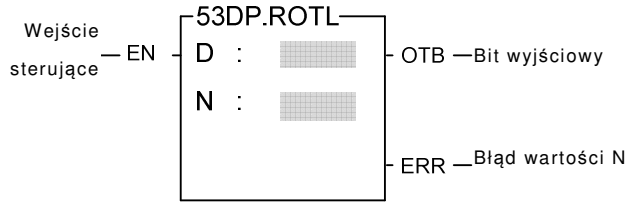
- Po lewej stronie przedstawiony jest przykład instrukcji przesuwającej bity w rejestrze R0 w prawo o 15 kolejnych pozycji. Wyniki przedstawione są poniżej.





FUN 53 D P ROTL	OBRÓĆ BITY W LEWO (ROTATE LEFT)	FUN 53 D P ROTL

SYMBOL DRABINKOWY



D : Rejestr z danymi do obrotu

N : Liczba bitów do obrotu

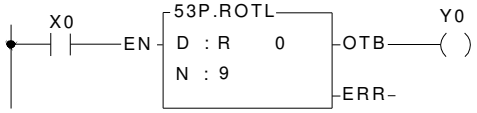
D, N mogą łączyć się z V, Z , P0~P9 w celu zastosowania adresowania pośredniego

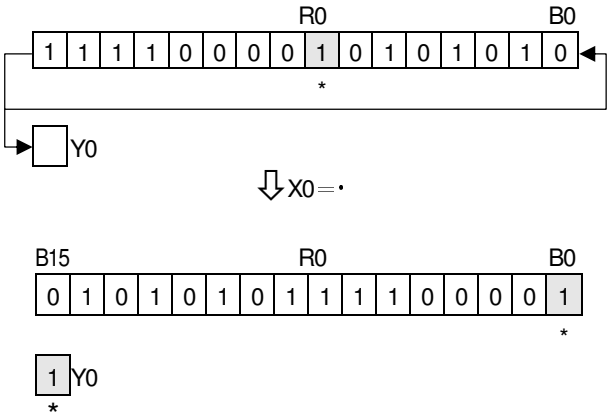
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 1 | lub | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Jeżeli wejście sterujące „ EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi obrót bitów w rejestrze D w lewo o N kolejnych pozycji (w kolejności od najmłodszego do najstarszego, tj. w przypadku 16-bitowej instrukcji, B0→B1, B1→B2, ... , B14→B15, B15→B0. W przypadku 32-bitowej instrukcji, B0→B1, B1→B2, ... , B30→B31, B31→B0). Stan, jako ostatni wsuniętego bitu na B0, pojawia się wyjściu OTB.

- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla N jest 1~16. Dla argumentu 32-bitowego (instrukcja D), poprawnym zakresem dla N jest 1~32. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

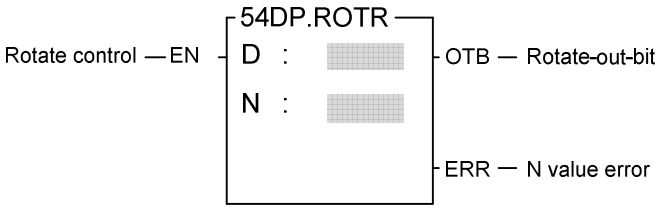
- Po lewej stronie przedstawiony jest przykład instrukcji realizującej obrót bitów w rejestrze R0 w lewo o 9 kolejnych pozycji. Wyniki przedstawione są poniżej.





FUN 54 D P ROTR	OBRÓĆ BITY W PRAWO (ROTATE RIGHT)	FUN 54 D P ROTR

Ladder symbol



D : Rejestr z danymi do obrotu

N : Liczba bitów do obrotu

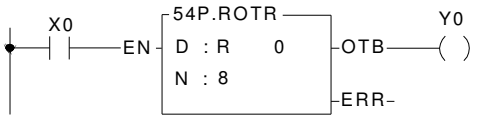
D, N mogą łączyć się z V, Z , P0~P9 w celu zastosowania adresowania pośredniego

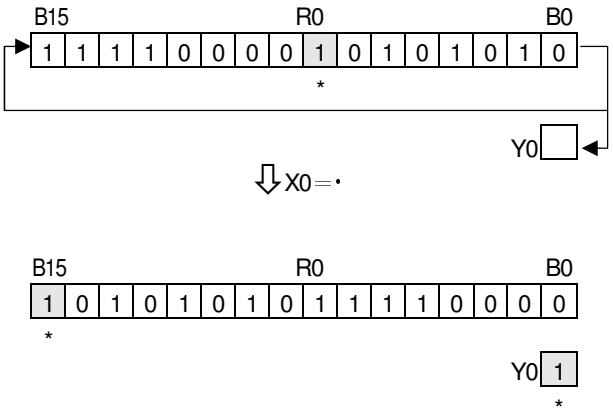
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 1 | lub | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Jeżeli „ EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi obrót bitów w rejestrze D w prawo o N kolejnych pozycji (w kolejności od najstarszego do najmłodszego, tj. w przypadku 16-bitowej instrukcji, B15→B14, B14→B13, ... , B1→B0, B0→B15. W przypadku 32-bitowej instrukcji, B31→B30, B30→B29, ... , B1→B0, B0→B31). Stan, jako ostatni wsuniętego na najstarszą pozycję bitu, pojawia się wyjściu OTB.

- Jeżeli argument jest 16-bitowy, prawidłowym zakresem dla N jest 1~16. Dla argumentu 32-bitowego (instrukcja D), poprawnym zakresem dla N jest 1~32. Jeżeli N znajdzie się poza zakresem, to znacznik błędu wartości N („ ERR") zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.

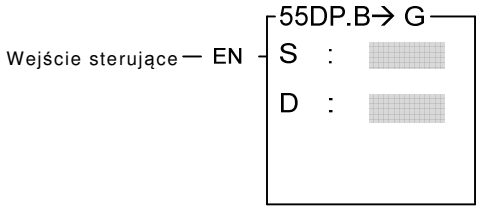
- Po lewej stronie przedstawiony jest przykład instrukcji realizującej obrót bitów w rejestrze R0 w prawo o 8 kolejnych pozycji. Wyniki przedstawione są poniżej.





FUN55 D P B→G	KONWERSJA KODU BINARNEGO NA KOD GRAYA (BINARY-CODE TO GRAY-CODE CONVERSION)	FUN55 D P B→G

SYMBOL DRABINKOWY



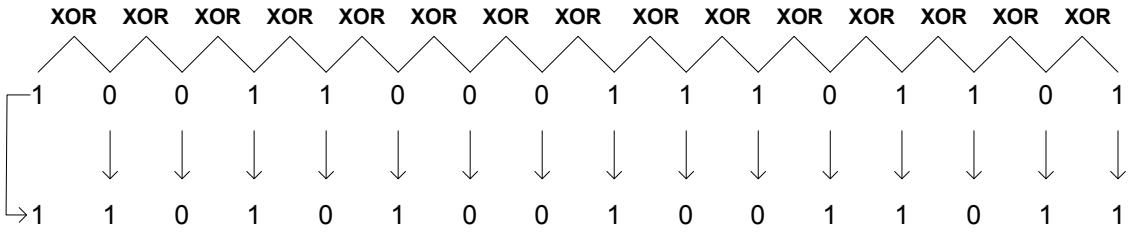
S : Adres rejestru do przekonwertowania

D : Adres rejestru przechowującego wynik konwersji

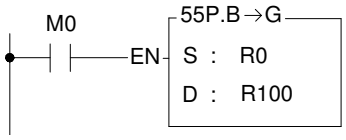
S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja kodu binarnego na kod Gray'a; gdzie S jest rejestrem źródłowym (kod binarny), a D jest rejestrem docelowym (kod Graya) do zapisu wyniku konwersji.
- Metoda konwersji przedstawiona została poniżej



Przykład 1: Jeżeli status M0 zmieni się z 0→1, przeprowadzona zostanie 16-bitowa konwersja kodu

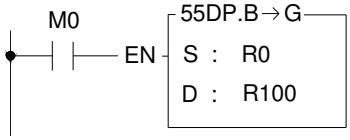


Konwersja 16-bitowego kodu binarnego w R0 na kod Graya i zapis wyniku konwersji w R100.

R0 = 1001010101010011B ➔ R100 = 1101111111111010B

FUN55 D P B→G	KONWERSJA KODU BINARNEGO NA KOD GRAYA (BINARY-CODE TO GRAY-CODE CONVERSION)	FUN55 D P B→G

Przykład 2: Jeżeli M0 =1, przeprowadzona zostanie 32-bitowa konwersja kodu

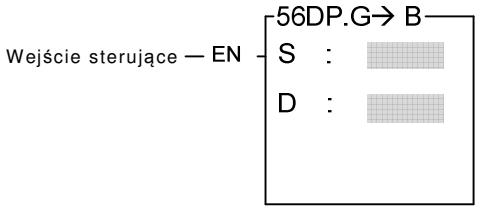


Konwersja 32-bitowego kodu binarnego w DR0 na kod Graya i zapis wyniku konwersji w DR100.

DR0 = 00110111001001000010111100010100B ➔ DR100 = 00101100101101100011100010011110B

FUN56 D P G→B	KONWERSJA KODU GRAYA NA KOD BINARNY (GRAY-CODE TO BINARY-CODE CONVERSION)	FUN56 D P G→B

SYMBOL DRABINKOWY



S : Adres rejestru do przekonwertowania

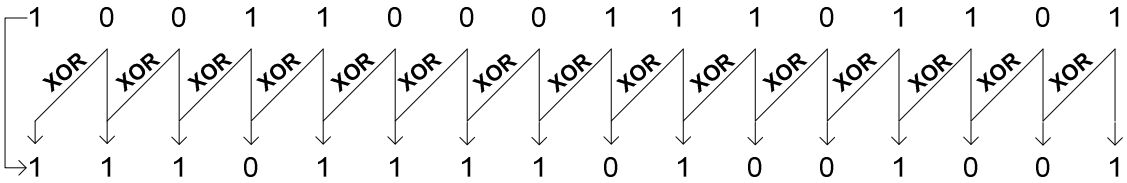
D : Adres rejestru przechowującego wynik konwersji

S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

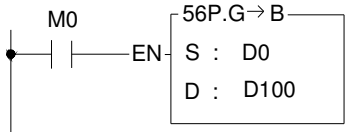
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja kodu Gray'a na kod binarny; gdzie S jest rejestrem źródłowym (kod Graya), a D jest rejestrem docelowym (Kod binarny) do zapisu wyniku konwersji.

- Metoda konwersji przedstawiona została poniżej:



Przykład 1: Jeżeli status M0 zmieni się z 0→1, przeprowadzona zostanie 16-bitowa konwersja kodu

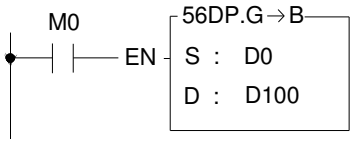


Konwersja 16-bitowego kodu Graya w D0 na kod binarny i zapis wyniku konwersji w D100

D0 = 1001010101010011B ➔ D100 = 1110011001100010B

FUN56 D P G→B	KONWERSJA KODU GRAYA NA KOD BINARNY (GRAY-CODE TO BINARY-CODE CONVERSION)	FUN56 D P G→B

Przykład 2: Jeżeli M0 =1, przeprowadzona zostanie 32-bitowa konwersja kodu

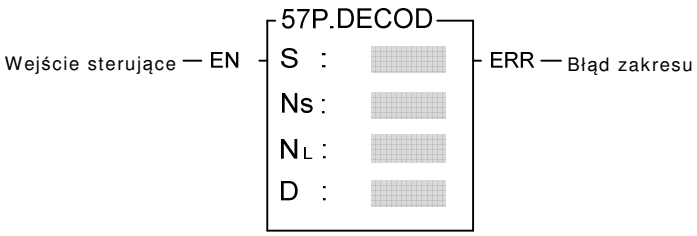


Konwersja 32-bitowego kodu Graya w DD0 na kod binarny i zapis wyniku konwersji w DD100

DD0 = 00110111001001000010111100010100B ➔ DD100 = 00100101110001111100101000011000B

FUN 57 P DECOD	DEKODER KODU 1 z N (DECODE)	FUN 57 P DECOD

SYMBOL DRABINKOWY



S : Rejestr z danymi źródłowymi do dekodowania (16-bitowy)

N_S : Numer bitu początkowego w S, do dekodowania

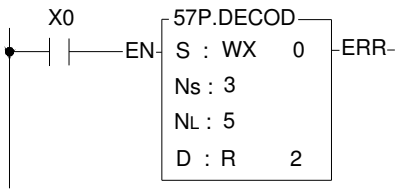
N_L : Liczba bitów w S, do dekodowania (1~8 bitów)

D : Rejestr początkowy do przechowywania dekodowanych wartości (2~256 punktów = 1~16 słów)

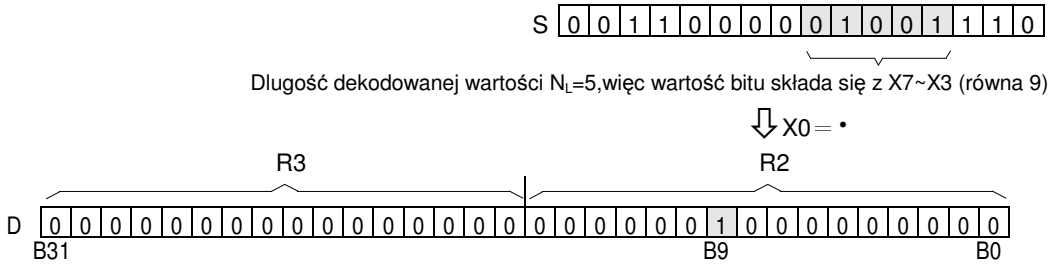
S, N_S, N_L, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N_S	○	○	○	○	○	○	○	○	○	○	○	○	0～15	○
N_L	○	○	○	○	○	○	○	○	○	○	○	○	2～256	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Instrukcja ustawi status pojedynczego bitu spośród 2^{NL} bitów w rejestrach D na 1, a pozostałe bity ustawi na 0. Numer bitu, który zostanie ustawiony na 1, określany jest przez wartość wynikającą ze stanów bitów BN_S～BN_S+N_L-1 w S（wartość ta jest wartością dekodowaną, BN_S jest początkowym bitem wartości dekodowanej a BN_S+N_L-1 jest bitem ostatnim).
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi odczytanie wartości wynikającej ze stanów bitów BN_S～BN_S+N_L-1 w S. Wartość ta określa pozycję bitu w rejestrach D, którego status ustawiony zostanie na 1. Reszta bitów w rejestrach D ustawiona zostanie na 0.
- Instrukcja obsługuje tylko argumenty 16-bitowe, czyli zakres bitów dla S jest B0~B15. W związku z tym, poprawnym zakresem Ns jest 0~15, a długość dekodowanej wartości N_L ograniczona jest do 1~8 bitów. Dlatego ilość bitów dekodowanego wyniku D jest z zakresu $2^{1\sim8}$ pozycji = 2~256 pozycji = 1~16 słów (nawet jeżeli wszystkie pozycje słowa nie są wykorzystane (N_L =1;2 lub 3), wszystkie 16cie bitów D jest kontrolowane przez tą instrukcję).Jeżeli wartość N_S lub N_L będzie poza powyższym zakresem, znacznik błędu zakresu „ERR" zostanie ustawiony na 1, a instrukcja nie zostanie wykonana.
- Jeżeli wartość końcowego bitu przekroczy B15 w S, nastąpi rozszerzenie zakresu w kierunku B0 w S + 1. Jednakże w takim wypadku, S + 1 nie może być rozszerzony na pewne typu argumentów (np. jeżeli S byłoby rejestrem typu D, to S + 1 nie może być rejestrem D3072). W takiej sytuacji, instrukcja zamieni liczbę bitów początkowy na najwyższa graniczną wartość.



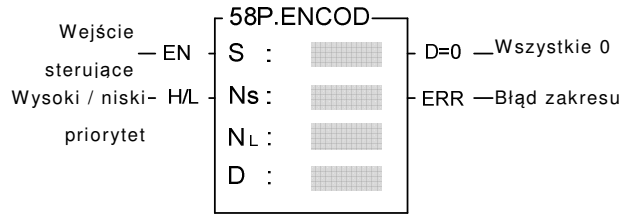
- Po lewej stronie przedstawiony jest przykład instrukcji dekodującej stany pięciu kolejnych bitów od X3 do X7 w rejestrze WX0 i zapisuje wyniki dekodowania w 32-bitowym rejestrze rozpoczynając od R2.



Ponieważ N_L=5, szerokość D wynosi 2^5= 32 punkty = 2 słowa, czyli D składa się z R3 i R2, a dekodowana wartość wynosi 01001=9. W związku z tym, bit B9 (10-ta pozycja) w rejestrze D ustawiany jest na 1, a pozostałe bity na 0.

FUN 58 P ENCOD	KODER KODU 1 z N (ENCODE)	FUN 58 P ENCOD

SYMBOL DRABINKOWY



S : Adres początkowy rejestrów do kodowania

N_S : Numer bitu w S jako punkt początkowy kodowania

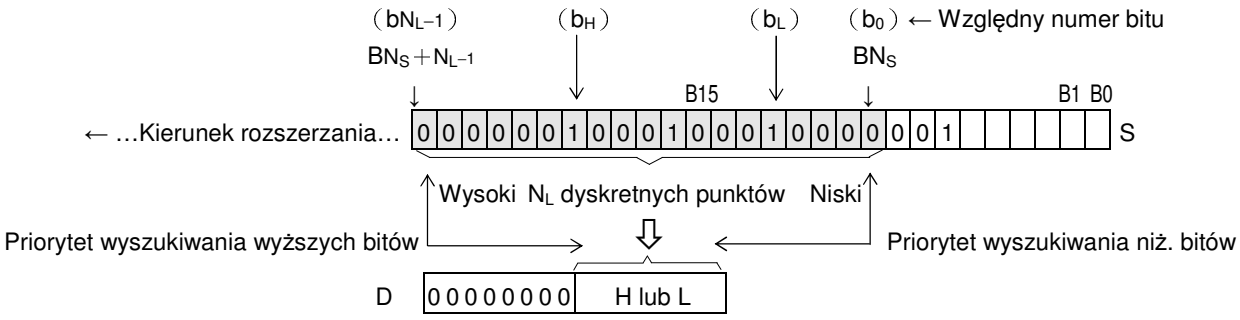
N_L : Liczba bitów w S do kodowania (2~256)

D : Rejestr przechowujący wynik kodowania (1 słowo)

S, N_S, N_L, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Argument	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N_S	○	○	○	○	○	○	○	○	○	○	○	○	0～15	○
N_L	○	○	○	○	○	○	○	○	○	○	○	○	2～256	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście strujace „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to wybranych zostanie N_L kolejnych bitów z zakresu BN_S～BN_S+N_{L-1} (BN_S jest pozycją początkową kodowania, której względnym numerem bitu jest b0 w S; BN_S+N_{L-1} jest bitem końcowym kodowania). Kodowanie realizowane jest od lewej do prawej z priorytetem wyszukania wyższego bitu (gdy H/L=1) lub od prawej do lewej z priorytetem wyszukania niższego bitu (gdy H/L=0). (np. wyszukiwany jest pierwszy bit o stanie 1, a wartość wynikająca z pozycji tego bitu jest zapisywana na młodszym bajcie (B0~B7) rejestru D, natomiast starszy bajt rejestru D ustawiany jest na 0.

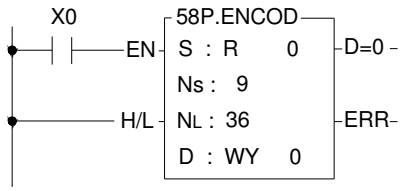


- Jak pokazano na powyższym schemacie, dla kodowania o priorytecie wyszukiwania wyższych bitów, pierwszym wyszukanym bitem o stanie wysokim jest b_H (o względnej pozycji b12), natomiast dla priorytetu wyszukiwania niższych bitów, jest to bit b_L (o względnej pozycji b4). Wśród pozycji N_L musi znaleźć się co najmniej jeden bit o stanie 1. Jeżeli wszystkie bity będą miały stan 0, instrukcja nie zostanie wykonana, a znacznik „D=0" zostanie ustawiony na 1.

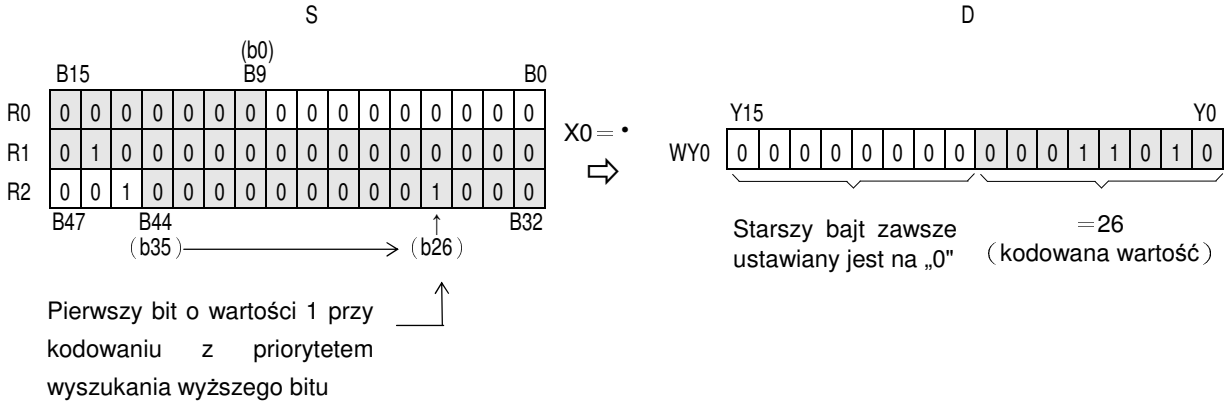
- Ponieważ S jest 16-bitowym rejestrem, wartość Ns może mieścić się w zakresie 0~15. Wartość ta określa początkowy bit kodowania (b0) w zakresie B0~B15 w rejestrze S. Wartość N_L zawiera się w zakresie 2~256 i wykorzystywana jest do identyfikacji bitu końcowego kodowania, tj. wartość ta wyznacza strefę kodowania wybierając N_L kolejnych bitów począwszy od bitu początkowego (b0) w lewo (tj. b0～bN_{L-1}). Jeżeli wartość N_S lub N_L będzie poza powyższym zakresem, znacznik błędu zakresu „ERR" zostanie ustawiony na 1, a instrukcja nie będzie wykonana.

FUN 58 P ENCOD	KODER KODU 1 z N (ENCODE)	FUN 58 P ENCOD

- Jeżeli punkt końcowy kodowania (bN_{L-1}) będzie większy niż B15 w S, nastąpi kontynuacja rozszerzenia zakresu w kierunku S+1, S+2. Jednakże S + 1 nie może być rozszerzony na pewne typu argumentów – w takim wypadku, instrukcja będzie mogła kodować jedynie punkty pomiędzy b0 a największym punktem granicznym.



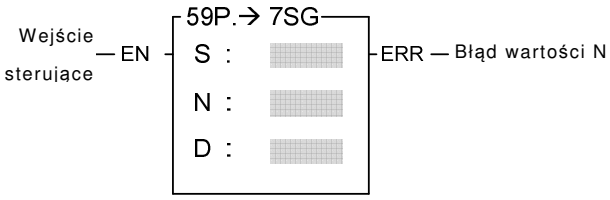
- Po lewej stronie przedstawiony jest przykład kodowania z priorytetem wyszukiwania wyższych bitów. Kiedy stan X0 zmieni się z 0 na 1, instrukcja wybierze 36 kolejnych bitów rozpoczynając od B9 (b0) w lewo oraz przeprowadzi kodowanie z priorytetem wyszukania wyższego bitu (ponieważ H/L=1). Oznacza to, że poczynając od b35 (punkt końcowy kodowania) w prawo, wyszukany zostanie pierwszy bit o stanie 1. W tym przykładzie, wartością wynikową jest b26, czyli wartość D wynosi 001AH=26, jak pokazano na poniższym schemacie.



Pierwszy bit o wartości 1 przy kodowaniu z priorytetem wyszukania wyższego bitu

FUN 59 P →7SG	KONWERSJA NA KOD 7-SEGMENTOWY (7-SEGMENT CONVERSION)	FUN 59 P →7SG

SYMBOL DRABINKOWY



S : Dane źródłowe do konwersji

N : Liczba półbajtów w S do przekonwertowania

D : Adres początkowy rejestrów przechowujących wyniki 7-segmentowe

S, N, D mogą łączyć się z V, Z,P0~ w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16-bit +/- liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja N+1 półbajtów (półbajt składa się z 4 kolejnych bitów, czyli bity B0~B3 w S tworzą półbajt 0, a bity B4~B7 tworzą półbajt 1, itp.) w rejestrze S na kod 7-segmentowy. Wyniki zostaną zapisany na młodszych bajach rejestrów D (najstarszy bajt nie ulega zmianie). Segmenty w rejestrze D ustawiane są kolejno, gdzie segment „a" ustawiany jest na B6, segment „b" ustawiany jest na B5, …, segment „g" ustawiany jest na B0. B7 nie jest wykorzystywany i ustawiany jest na 0. Więcej szczegółów znajduje się w tabeli „Znaki kodu 7-segmentowego".

- Ponieważ instrukcja ograniczona jest do 16 bitów, a rejestr S zawiera tylko 4 półbajty (NB0~NB3), to poprawnym zakres N jest 0~3. Jeżeli wartość N przekroczy ten zakres, znacznik błędu wartości N („ERR") zostanie ustawiony na 1.

- Należy zwrócić uwagę na to, że liczba konwertowanych półbajtów ma wartość N+1. N=0 oznacza jedną cyfrę do konwersji, N=1 oznacza dwie cyfry do konwersji, itd…

- Przy zastosowaniu modułów rozszerzeń FATEK'a, przeznaczonych do obsługi wyświetlaczy 7-segmentowych (FBs-7SGxx) oraz funkcji FUN84 (7SEG), do zastosowań z kodowaniem i bez kodowania, istnieje możliwość łączenia funkcji FUN59 i FUN84 w celu uproszczenia implementacji programu.

FUN 59 P →7SG	KONWERSJA NA KOD 7-SEGMENTOWY (7-SEGMENT CONVERSION)	FUN 59 P →7SG

〈Przykład 1〉 Konwersja kodu szesnastkowego na 7-segmentowy, przy zmianie stanu M1 z 0→1

˙ Rysunek przedstawia instrukcję konwertującą pierwszą cyfrę (półbajt) w rejestrze R0 na kod 7-segmentowy i zapisującą wynik w młodszym bajcie rejestru R100. Starszy bajt w R100 nie ulega zmianie.

Pierwotnie R100＝0000H

R0＝0001H ➔ R100＝0030H（1）

〈Przykład 2〉 Konwersja kodu szesnastkowego na 7-segmentowy przy M1 w stanie 1

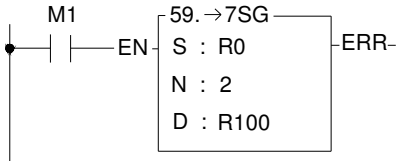
˙ Instrukcja po lewej przekonwertuje pierwszą i drugą cyfrę (półbajty) w rejestrze R0 na kod 7-segmentowy i zapisze wynik w rejestrze R100.

˙ W młodszym bajcie R100 zapisana będzie pierwsza cyfra.

˙ W starszym bajcie R100 zapisana będzie druga cyfra.

R0＝0056H ➔ R100＝5B5FH（56）

〈Przykład 3〉 Konwersja kodu szesnastkowego na 7-segmentowy przy M1 w stanie 1



˙ Instrukcja po lewej przekonwertuje pierwszą, drugą i trzecią cyfrę (półbajty) w rejestrze R0 na kod 7-segmentowy i zapisze wynik w rejestrach R100 i R101.

˙ W młodszym bajcie R100 zapisana będzie pierwsza cyfra

˙ W starszym bajcie R100 zapisana będzie druga cyfra

˙ W młodszym bajcie R101 zapisana będzie trzecia cyfra.

˙ Starszy bajt R101 nie ulega zmianie.

Pierwotnie R101=0000H

R0=0A48H ➔ R100=337FH（48）

R101=0077H（A）

〈Przykład 4〉 Konwersja kodu szesnastkowego na 7-segmentowy przy M1 w stanie 1

˙ Instrukcja po lewej przekonwertuje cyfry (półbajty) 1~4 w rejestrze R0 na kod 7-segmentowy i zapisze wynik w rejestrach R100 i R101

˙ W młodszym bajcie R100 zapisana będzie pierwsza cyfra

˙ W starszym bajcie R100 zapisana będzie druga cyfra

˙ W młodszym bajcie R101 zapisana będzie trzecia cyfra.

˙W starszym bajcie R101 zapisana będzie czwarta cyfra.

R0=2790H ➔ R100=7B7EH（90）

R101=6D72H（27）

FUN 59 P →7SG	KONWERSJA NA KOD 7-SEGMENTOWY (7-SEGMENT CONVERSION)	FUN 59 P →7SG

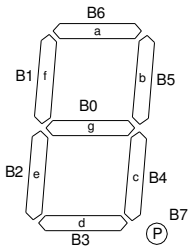
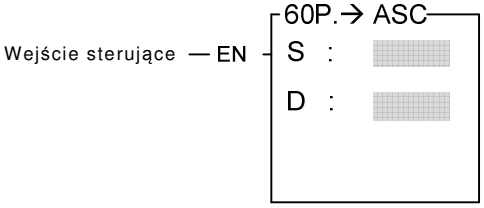
Wartość półbajtu w S		Rozkład wyświetlacza 7-segmentowego	Młodszy bajt w D								Wyświetlany znak
Liczba szesnastkowa	Liczba binarna		B7 ●	B6 a	B5 b	B4 c	B3 d	B2 e	B1 f	B0 g	
0	0000		0	1	1	1	1	1	1	0	0
1	0001		0	0	1	1	0	0	0	0	1
2	0010		0	1	1	0	1	1	0	1	2
3	0011		0	1	1	1	1	0	0	1	3
4	0100		0	0	1	1	0	0	1	1	4
5	0101		0	1	0	1	1	0	1	1	5
6	0110		0	1	0	1	1	1	1	1	6
7	0111		0	1	1	1	0	0	1	0	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	1	1	0	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	0	0	1	1	1	1	1	b
C	1100		0	1	0	0	1	1	1	0	C
D	1101		0	0	1	1	1	1	0	1	d
E	1110		0	1	0	0	1	1	1	1	E
F	1111		0	1	0	0	0	1	1	1	F

Tabela: Znaki kodu 7-segmentowego

FUN 60 P →ASC	KONWERSJA ASCII (ASCII CONVERSION)	FUN 60 P →ASC

SYMBOL DRABINKOWY



S : Znaki alfanumeryczne kodowane na kod ASCII

D : Adres początkowy rejestrów przechowujących wyniki konwersji na kod ASCII

Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	Alfanumer.
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	1～12 alfanumeryczne
S											○
D	○	○	○	○	○	○	○	○*	○*	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja liter i cyfr zapisanych w polu S (pole S może zawierać do 12 znaków alfanumerycznych) na kod ASCII i zapisanie wyników w rejestrach rozpoczynając od D. Każde 2 znaki alfanumeryczne zajmują jeden 16-bitowy rejestr.
- W większości przypadków, instrukcja ta przechowuje alfanumeryczne informacje w programie, odczekuje na pojawienie się określonych okoliczności, konwertuje te informacje na kod ASCII i przenosi je na zewnętrzne urządzenia wyświetlające, które obsługują kodowanie ASCII.

- Po lewej stronie przedstawiony jest przykład instrukcji, która konwertuje 6 liter – ABCDEF na kod ASCII i zapisuje je w 3 kolejnych rejestrach rozpoczynając od R0.

S

Litery ABCDEF

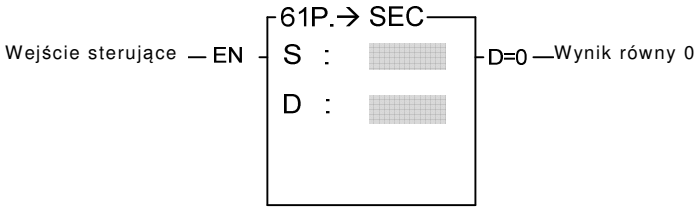
X0 = •

D

	Starszy bajt	Młodszy bajt
R0	42 (B)	41 (A)
R1	44 (D)	43 (C)
R2	46 (F)	45 (E)

FUN 61 P →SEC	KONWERSJA GODZIN:MINUT:SEKUND NA SEKUNDY (HOUR:MINUTE:SECOND TO SECONDS CONVERSION)	FUN 61 P →SEC

SYMBOL DRABINKOWY

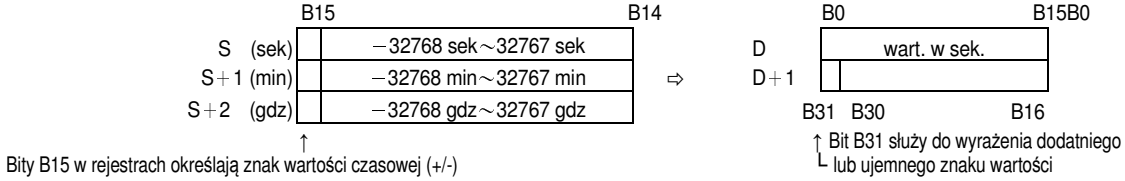


S : Początkowy rejestr danych kalendarzowych do konwersji

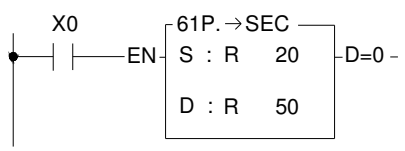
D : Rejestr początkowy przechowujący wynik konwersji

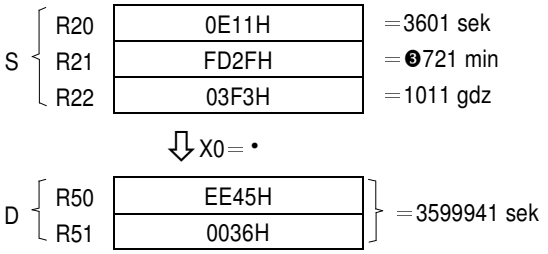
	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
Zakres	WX0 □	WY0 □	WM0 □	WS0 □	T0 □	C0 □	R0 □	R3840 □	R3904 □	R3968 □	R5000 □	D0 □	−117968399 |
Argument	WX240	WY240	WM1896	WS984	T255	C255	R3839	R3903	R3967	R4167	R8071	D4095	117964799
S	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja danych w formacie godzina: minuta: sekunda zawartych w S~S+2, na równoważną im wartość w sekundach. Wartość ta zostanie zapisana w 32-bitowym rejestrze będącym połączeniem D i D+1. Jeżeli wynik = 0, znacznik „D = 0" zostanie ustawiony na 1.
- Spośród wszystkich instrukcji FBs-PLC, instrukcjami związanymi z danymi w formacie godzina: minuta: sekunda są FUN61 i 62. Wykorzystują one 3 słowa (rejestry) do zapisywania danych o czasie (patrz schemat poniżej). Pierwsze słowo stanowi rejestr sekund, drugie minut, a trzecie godzin. Spośród 16 bitów w każdym rejestrze, tylko bity B14~B0 wykorzystywane są do wyrażenia wartości czasowej. Bit B15 wykorzystywany jest do określenia tego, czy wartość jest ujemna czy dodatnia. Jeżeli stan B15 jest 0, to wartość jest dodatnia, natomiast jeżeli stan B15 jest 1, wartość jest ujemna. Wartość czasowa B14~B0 wyrażana jest w kodzie binarnym, natomiast jeżeli wartość ta jest ujemna, B14~B0 wyrażana jest przy użyciu uzupełnienia dwójkowego. Liczba sekund, otrzymana po konwersji tą funkcją, jest wynikiem sumy sekund z trzech rejestrów wyrażających godziny: minuty: sekundy.



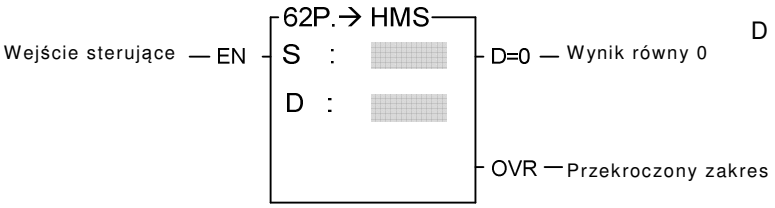
- Poza instrukcjami FUN61 i 62, które traktują rejestry danych w formacie godzina: minuta: sekunda jako zintegrowane dane, pozostałe instrukcje traktują je jako odseparowane wartości rejestrów.
- Przykładowy program, przedstawiony poniżej, konwertuje dane w formacie godzina: minuta: sekunda tworzone przez R20~R22, na odpowiadającą im wartość w sekundach i zapisuje wynik w 32-bitowym rejestrze tworzonym przez R50~R51. Wyniki przedstawione są poniżej.





FUN 62 P →HMS	KONWERSJA SEKUND→GODZINY : MINUTY : SEKUNDY (SECOND→HOUR : MINUTE : SECOND)	FUN 62 P →HMS

SYMBOL DRABINKOWY

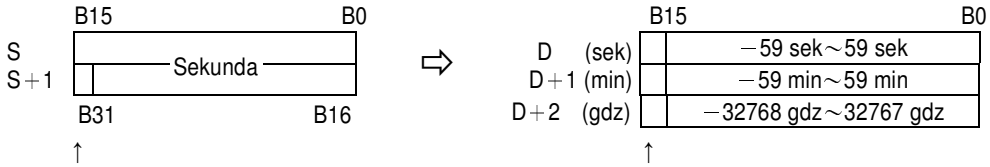


S :Rejestr początkowy liczby sekund do konwersji

D : Adres początkowy rejestrów przechowujących wynik konwersji (godziny: minuty: sekundy)

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	−117968399 | 117964799
S	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja wartości sekund z 32-bitowego rejestru S~S+1, na równoważną wartość w formacie godzina: minuta: sekunda. Wynik zostanie zapisany w trzech kolejnych rejestrach D~D+2. Wszystkie dane w tej instrukcji wyrażane są w kodzie binarnym (w przypadku wartości ujemnych, dane wyrażane są za pomocą uzupełnienia dwójkowego).

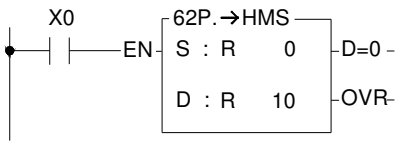


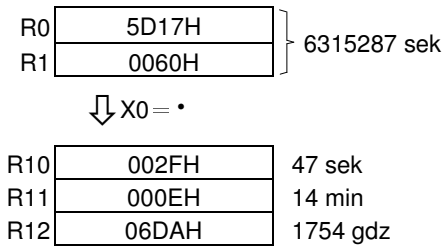
Bit B31 w drugim rejestrze używany jest jako bit znaku drugiej wartości.

Bit B15 w każdym z rejestrów używany jest jako bit znaku godzin: minut: sekund.

- Jak pokazano na powyższym schemacie, po konwersji na godziny: minuty: sekundy, wartość minut i sekund może być tylko w zakresie od -59 do 59, natomiast wartość godzin, w zakresie -32768 do 32767. Z uwagi na to, minimalnymi i maksymalnymi wartościami graniczną w rejestrze D jest: -32768 dla godzin, -59 dla minut, -59 dla sekund oraz 32767 dla godzin, 59 dla minut i 59 dla sekund. Odpowiadające tym wartością granicznym wartości sekund to odpowiednio -117968399 i 117968399. Jeżeli wartość S przekroczy ten zakres, instrukcja nie zostanie wykonana, a znacznik przekroczenia zakresu „OVR" zostanie ustawiony na 1. Jeżeli S = 0, wynik równy będzie 0, a znacznik D = 0" zostanie ustawiony na 1.

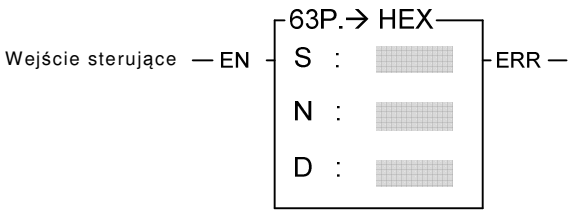
- Przykładem działania opisywanej instrukcji jest program przedstawiony na poniższym schemacie. Zauważ, że zawartość rejestrów przedstawiona jest w systemie heksadecymalnym, natomiast po prawej, odpowiadające im wartość zapisane zostały w systemie dziesiętnym.





FUN 63 P →HEX	KONWERSJA KODU ASCII NA WARTOŚCI HEKSADECYMALNE (CONVERSION OF ASCII CODE TO HEXADECIMAL VALUE)	FUN 63 P →HEX

SYMBOL DRABINKOWY



S : Adres początkowy rejestrów do przekonwertowania

N : Liczba kolejnych rejestrów S do przekonwertowania

D : Adres początkowy rejestrów przechowujących wyniki konwersji (w kodzie heksadecymalnym).

S, N, D, mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16-bit +liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	1～511	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście serujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja N kolejnych heksadecymalnych znaków zapisanych w kodzie ASCII ('0'~'9','A'~'F'), pochodzących z N kolejnych 16-bitowych rejestrów S (młodszy bajt jest wiążący) na wartości szesnastkowe i zapisanie wyników konwersji w rejestrach rozpoczynając od D. Każde 4 kody ASCII zapisywane są w jednym rejestrze. Półbajty w rejestrze, który nie brał udziału w konwersji kodu ASCII, pozostaną niezmienione.

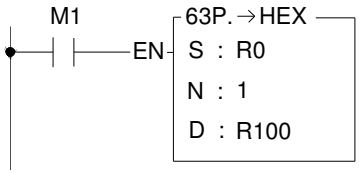
- Jeżeli wartość N będzie równa 0 lub większa niż 511, konwersja nie zostanie wykonana.

- W przypadku wystąpienia błędu kodu ASCII (wartość w rejestrach S spoza zbioru 30H～39H lub 41H～46H), uaktywni się wyjście „ERR".

- Najczęstszym wykorzystaniem tej instrukcji jest konwersja ciągów znaków zapisanych w szesnastkowym kodzie ASCII ('0'~'9','A'~'F'), odbieranych przez porty komunikacyjne (port1 ~ port4) z urządzeń peryferyjnych, na wartości szesnastkowe, które mogą być przetwarzane przez CPU bezpośrednio.

FUN 63 P →HEX	KONWERSJA KODU ASCII NA WARTOŚCI HEKSADECYMALNE (CONVERSION OF ASCII CODE TO HEXADECIMAL VALUE)	FUN 63 P →HEX

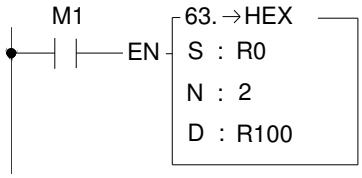
〈Przykład 1〉 Konwersja kodu ASCII na wartość szesnastkową przy zmianie statusu M1 z 0 na 1.



· Instrukcja konwertuje kod ASCII w R0 na wartość heksadecymalną i zapisuje wynik w półbajcie 0 (półbajty 1~3 pozostają niezmienione) w R100.

Pierwotnie R100＝0000H
R0＝0039H（9）➔ R100＝0009H

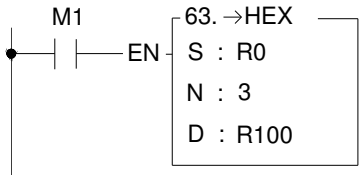
〈Przykład 2〉 Konwersja kodu ASCII na wartość szesnastkową przy M1 w stanie 1.



· Instrukcja konwertuje kody ASCII w R0 i R1 na wartości heksadecymalne i zapisuje wynik w młodszym bajcie (starszy bajt pozostaje niezmieniony) w R100

R0＝0039H（9） Pierwotnie R100＝0000H
R1＝0041H（A）➔ R100＝009AH

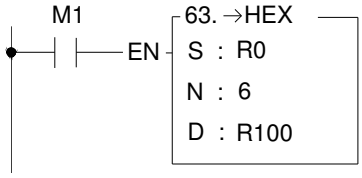
〈Przykład 3〉 Konwersja kodu ASCII na wartość szesnastkową przy M1 w stanie 1.



· Instrukcja konwertuje kody ASCII w R0 i R1 na wartości heksadecymalne i zapisuje wynik w R100 (półbajt 3 pozostaje niezmieniony)

R0＝0039H （9） Oryginalnie R100＝0000H
R1＝0041H （A）
R2＝0045H （E）➔ R100＝09AEH

〈Przykład 4〉 Konwersja kodu ASCII na wartość szesnastkową przy M1 w stanie 1.

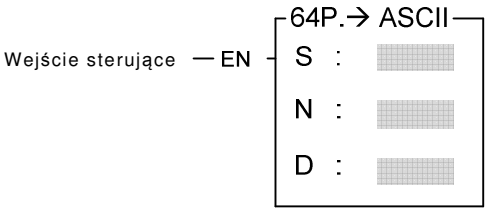


· Instrukcja konwertuje kody ASCII w R0~R5 na wartości heksadecymalne i zapisuje wynik w R100~R101.

R0＝0031H（1） Oryginalnie R100＝0000H
R1＝0032H（2） R101＝0000H
R2＝0033H（3）
R3＝0034H（4）
R4＝0035H（5）➔ R100＝3456H
R5＝0036H（6） R101＝0012H

FUN 64 P →ASCII	KONWERSJA WARTOŚCI HEKSADECYMALNYCH NA KOD ASCII (CONVERSION OF HEXADECIMAL VALUE TO ASCII CODE)	FUN 64 P →ASCII

SYMBOL DRABINKOWY



S : Adres początkowy rejestrów do przekonwertowania
N : Liczba półbajtów w S, do przekonwertowania na heksadecymalny kod ASCII
D : Adres początkowy rejestrów przechowujących wyniki konwersji

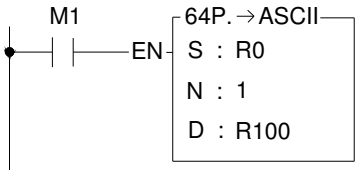
S, N, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16-bit + liczba	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	1~511	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja N kolejnych półbajtów heksadecymalnych wartości zawartych w rejestrach S, na odpowiadające im kody ASCII, a wyniki konwersji zostaną zapisane na młodszych bajtach (starszy bajt pozostanie niezmieniony) rejestrów rozpoczynających się od D.
- Jeżeli wartość N będzie równa 0 lub większa niż 511, konwersja nie zostanie wykonana.
- Najczęstszym wykorzystaniem tej instrukcji jest konwersja danych numerycznych, wypracowanych przez PLC, na ciąg kodów ASCII i przesłanie ich do urządzeń peryferyjnych, obsługujących kod ASCII, przez porty komunikacyjne (port1 ~ port4).

FUN 64 P →ASCII	KONWERSJA WARTOŚCI HEKSADECYMALNYCH NA KOD ASCII (CONVERSION OF HEXADECIMAL VALUE TO ASCII CODE)	FUN 64 P →ASCII

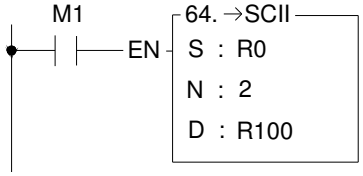
〈Przykład 1〉 Konwersja wartości szesnastkowej na kody ASCII przy zmianie statusu M1 z 0 na 1.



・Instrukcja konwertuje półbajt 0 w R0 na kod ASCII i zapisuje wynik w R100 (starszy bajt pozostaje niezmieniony).

R0＝0009H ➔ R100＝0039H（9）

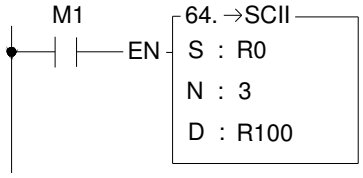
〈Przykład 2〉 Konwersja wartości szesnastkowej na kod ASCII przy M1 w stanie 1.



・Instrukcja konwertuje półbajty NB0～NB1 w R0 na kody ASCII i zapisuje wynik w R100～R101 (starsze bajty pozostają niezmienione).

R0＝009AH ➔ R100＝0039H（9）
R101＝0041H（A）

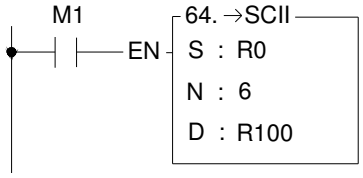
〈Przykład 3〉 Konwersja wartości szesnastkowej na kod ASCII przy M1 w stanie 1.



・Instrukcja konwertuje półbajty NB0～NB2 w R0 na kody ASCII i zapisuje wynik w R100～R102

R0＝0123H ➔ R100＝0031H（1）
R101＝0032H（2）
R102＝0033H（3）

〈Przykład 4〉 Konwersja wartości szesnastkowej na kod ASCII przy M1 w stanie 1.



・Instrukcja konwertuje półbajty NB0～NB5 w R0～R1 na kody ASCII i zapisuje wynik w R100～R105.

R0＝3456H ➔ R100＝0031H（1）
R1＝0012H R101＝0032H（2）
R102＝0033H（3）
R103＝0034H（4）
R104＝0035H（5）
R105＝0036H（6）

END	KONIEC SKANU (PROGRAM END)	END

SYMBOL DRABINKOWY

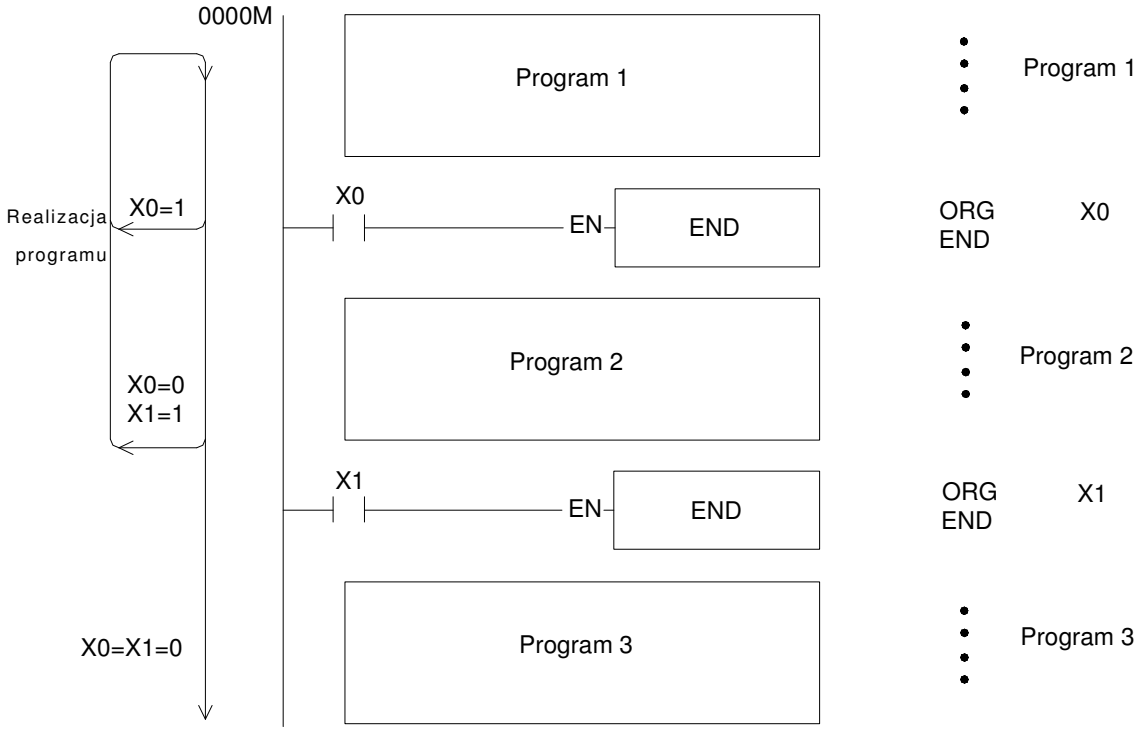
Wejście sterujące — EN — END

Brak argumentu

- Jeżeli wejście sterujące „EN” = 1, instrukcja zostanie wywołana. Po wywołaniu instrukcji END, kiedy „EN” = 1, przebieg programu natychmiast powróci do punktu początkowego programu (pierwszej linii programu głównego – 0000M) w celu rozpoczęcia kolejnego skanu programu – oznacza to, że żadne instrukcje programu, znajdujące się poniżej wywołania funkcji END, nie zostaną wykonane. Jeżeli EN = „0”, funkcja zostanie zignorowana, a linie programu znajdujące się poniżej END, będą dalej wykonywane.

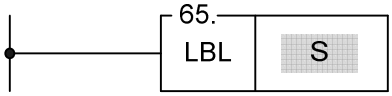
- Funkcja ta może być umieszczona w kilku miejscach programu. Wtedy sygnały „EN” na ich wejściach, sterują punktami końcowymi realizacji programu. Jest to szczególnie przydatne przy debugowaniu (usuwaniu błędów) i testowaniu programu.

- Użycie funkcji END w programie głównym nie jest konieczne. CPU automatycznie powróci do punktu początkowego, po osiągnięciu ostatniej linii programu głównego.



FUN 65 LBL	ETYKIETA (LABEL)	FUN 65 LBL

SYMBOL DRABINKOWY



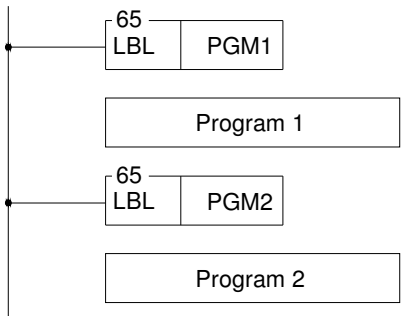
S : Znaki alfanumeryczne, 1~6 znaków

- Instrukcja ta służy do wskazywania miejsc w programie, stanowiących punkt docelowy wykonania funkcji JUMP i CALL, oraz do wskazywania procedur obsługi przerwań sprzętowych. Może być ona także wykorzystana do zwiększenia czytelności i łatwości interpretacji programu.
- Instrukcja ta służy jedynie do oznaczania miejsc w programie w celu sterowania przebiegiem jego wykonania lub zamieszczania adnotacji. Sama instrukcja nie wykonuje żadnych akcji. Obojętnie, czy program będzie zawierać tą funkcję, czy nie - nie wpłynie to na realizację programu.
- Etykieta może zawierać 1 ～ 6 znaków alfanumerycznych i nie może być zdublowana w programie. Następujące etykiety, identyfikujące procedury przerwań sprzętowych, są zarezerwowane. Te „zarezerwowane słowa” nie mogą być zastosowane do normalnych etykiet programu.

Zarezerwowane słowa (etykiety)	Opis
X0+I～X15+I（INT0～INT15） X0–I～X15–I（INT0–～INT15–）	Etykiety procedur obsługi przerwań sprzętowych pochodzących od wejść (X0～X15)
HSC0I～HSC7I	Etykiety procedur obsługi przerwań sprzętowych pochodzących od szybkich liczników HSC0 ～ HSC7
1MSI（1MS）、2MSI（2MS）、3MSI（3MS）、4MSI（4MS）、5MSI（5MS）、10MSI（10MS）、50MSI（50MS）、100MSI（100MS）	Etykiety dla 8miu typów okresowych przerwań sprzętowych, pochodzących od zegara wewnętrznego
HSTAI（ATMRI）、HST0I~HST3I	Etykiety procedur obsługi przerwań sprzętowych pochodzących od stałego szybkiego zegara
PSO0I～PSO3I	Etykiety procedur obsługi przerwań sprzętowych, wywoływanych po zakończeniu wysyłania impulsów, przy pozycjonowaniu NC

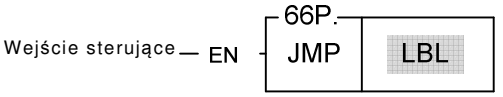
Tylko procedury obsługi przerwań sprzętowych, mogą wykorzystywać etykiety wymienione powyżej. W przypadku pomyłki użycia zarezerwowanej etykiety dla normalnej podprocedury, może to spowodować niepowodzenie działania CPU lub nieoczekiwaną operację.

Etykiety na poniższych schemacie służą jedyne jako przykład (nie są zaimplementowane jako punkty odwołań funkcji CALL ani JUMP). Informacje dotyczące użycia etykiet z instrukcją JUMP, znajdują się w opisie działania instrukcji JMP. Informacje dotyczące użycia etykiet w celu identyfikowania podprocedur, znajdują się w opisie działania instrukcji CALL.



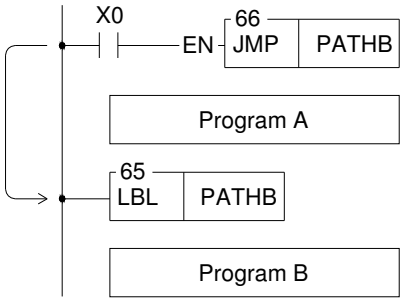
FUN 66 P JMP	SKOK (JUMP)	FUN 66 P JMP

SYMBOL DRABINKOWY



LBL : Etykieta w programie, do której ma nastąpić skok

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), PLC przeskoczy do kolejnej linii, znajdującej się za wskazaną etykietą LBL, po czym będzie kontynuować wykonywanie programu.
- Instrukcja ta jest szczególnie przydatna do zastosowań, w których pewne części programu są wykonywane jedynie przy spełnieniu odpowiednich warunków. Może to skrócić czas skanu, poprzez niewykonywanie całego programu.
- Instrukcja ta umożliwia przeskok w tył (np. kiedy etykieta LBL znajduje się przed wystąpieniem instrukcji JMP). Jednak w takim przypadku, należy zachować ostrożność, ponieważ takie operacje przeskoku mogą spowodować przekroczenia granicznego czasu skanu ustalonego przez zegar „watchdog"(WDT). Wystąpi wtedy błąd timera WDT i zakończenie wykonywania programu.
- Instrukcja JUMP umożliwia jedynie skoki w obszarze programu głównegolub w obszarze podprocedur. Nie ma możliwości realizacji skoków pomiędzy tymi obszarami.



Na schemacie po lewej, jeżeli X0=1, program przeskoczy bezpośrednio pod etykietę LBL oznaczoną jako PATHB i nastąpi kontynuacja realizacji programu B. Realizacja programu A będzie pominięta i nie zostaną zrealizowane żadne instrukcje w ramach tego programu. Statusy rejestrów i cewek powiązanych z programem A pozostaną bez zmian (tak jakby część A programu nie istniała).

FUN 67 P CALL	WYWOŁAJ (CALL)	FUN 67 P CALL

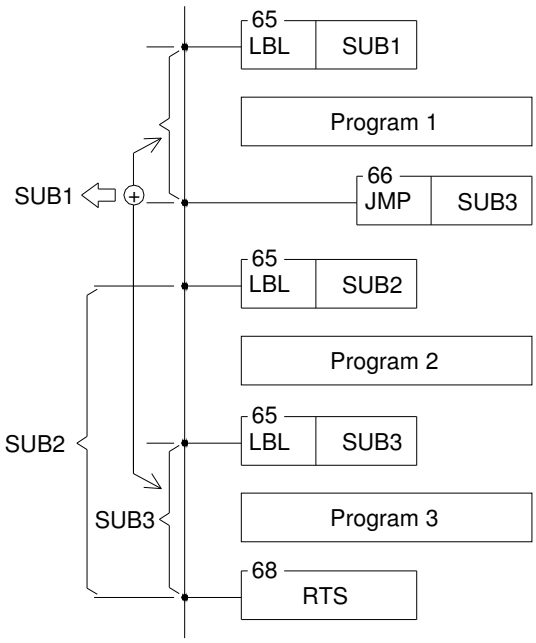
SYMBOL DRABINKOWY

Wejście sterujące— EN — 67P. CALL | LBL

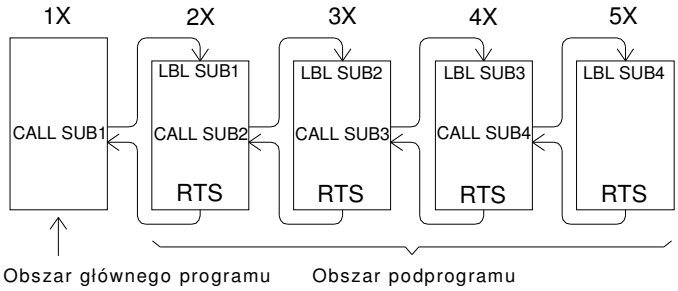
LBL : Etykieta wywoływanej podprocedury

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), PLC wywoła (wykona) podprocedurę o etykiecie wskazanej w polu LBL. Po wywołaniu podprocedury, program będzie wykonywany standardowo (linia po linii), jednakże w przypadku napotkania instrukcji RTS, przebieg podporcedury niezwłocznie zostanie przerwany i wykonywanie programu powróci do linii znajdującej się poniżej wywołania podprocedury (instrukcji CALL).

- Wszystkie podporcedury muszą kończyć się instrukcją „powrót z podproceduty – RTS". W przeciwnym wypadku, może to skutkować błędem wykonania programu lub nastąpi zatrzymanie CPU. Niemniej jednak, jedna instrukcja RTS może być wykorzystywana przez kilka podporcedur (nazywanych wielokrotnie wprowadzanymi podprocedurami: pomimo, że punkty wprowadzania są różne, ścieżka powrotu jest taka sama), tak jak przedstawiono na schemacie podprocedur SUB1～3 po prawej.



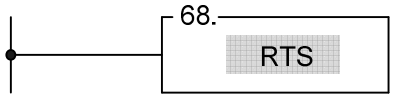
- Jeżeli program główny wywoła podprocedurę, to podporcedura ta, również może wywołać inne podprocedury (tak zwane zagnieżdżone podprocedury) aż do 5 poziomów (łącznie z procedurą obsługi przerwania).



- Procedury obsługi przerwań sprzętowych（HSC0I～HSC7I、PSO0I～PSO3I、X0+I～X15+I／INT0～INT15、X0–I～X15–I／INT0–～INT15–、HSTAI／ATMRI、1MSI／1MS、2MSI／2MS、3MSI／3MS、4MSI／4MS、5MSI／5MS、10MSI／10MS、50MSI／50MS 、100MSI／100MS）są także rodzajem podprocedur. Znajdują się one także w obszarze podprocedur. Jednakże, wywołanie procedury obsługi przerwania sprzętowego, następuje poprzez sygnał wysłany ze sprzętu do CPU (nazywanym zgłoszeniem przerwania sprzętowego). Procedura obsługi przerwania, może także wywołać podprocedurę albo być przerwaną przez inne przerwanie o wyższym priorytecie. Ponieważ jest to także podprocedura (stanowiąca jeden poziom), może ona wywołać lub być przerwana przez 4 poziomy podprocedur/procedur obsługi przerwań. Więcej informacji w opisie instrukcji RTI.

FUN 68 RTS	POWRÓT Z PODPROCEDURY (RETURN FROM SUBROUTINE)	FUN 68 RTS

SYMBOL DRABINKOWY



- Instrukcja ta jest wykorzystywana do wskazywania zakończenia podprocedury. Dlatego może ona występować jedynie w obszarze podprocedur. Instrukcja ta nie ma wejścia sterującego, a więc nie ma możliwości podłączenia do niej żadnych styczników szeregowo - jest podłączona bezpośrednio do linii zasilającej.

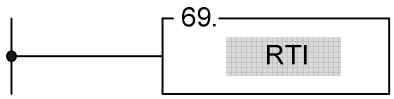
- Jeżeli instrukcja ta zostanie napotkana przez PLC, oznacza to, że realizacja podprocedury została zakończona. Dlatego nastąpi niezwłoczny powrót, do następnej linii, występującej po ostatnim wywołaniu tej podporcedury instrukcją CALL.

- W przypadku, gdy instrukcja ta napotka na którąkolwiek z trzech instrukcji kontroli przepływu MC, SKP lub JMP, to nie zostanie ona zrealizowana (będzie traktowana jako nieistniejąca). Jeżeli powyższe instrukcje zostaną wykorzystane w podprocedurze i będą skutkować brakiem realizacji instrukcji RTS, to PLC zatrzyma wykonywanie programu i ustawi M1933 (znacznik błędu przepływu) na 1. W zawiązku z tym, niezależnie od przepływu programu, musi on zapewnić możliwość wykonanie odpowiedniej instrukcji RTS przez podprocedurę.

- Informacje dotyczące wykorzystania instrukcji RTS znajdują się w opisie instrukcji CALL.

FUN 69 RTI	POWRÓT Z PRZERWANIA (RETURN FROM INTERRUPT)	FUN 69 RTI

SYMBOL DRABINKOWY



- Działanie instrukcji jest zbliżone do działania instrukcji RTS. Niemniej jednak, instrukcja RTS wykorzystywana jest do zakończania wykonywania podporcedur, natomiast RTI do zakończania wykonywania procedur obsługi przerwań sprzętowych. Więcej informacji w opisie instrukcji RTS.

- Instrukcja RTI może być wykorzystywana jednocześnie przez kilka procedur obsługi przerwań sprzętowych. Zasada jest taka sama jak w przypadku wykorzystywania instrukcji RTS przez kilka podporcedur. Więcej informacji w opisie instrukcji CALL.

- Różnica pomiędzy zgłoszeniem przerwania sprzętowego a wykonaniem instrukcji CALL jest taka, że nazwa podprocedury (LBL) i etykieta dla instrukcji CALL określana jest przez użytkownika, a instrukcja ta aktywowana jest w głównym programie lub którymś z podprogramów. W związku z tym, jeżeli PLC napotka instrukcję CALL, a „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to PLC wywoła (wykona) tą podprocedurę. Do wywołania procedury obsługi przerwania, wykorzystywane są sygnały sprzętowe, w celu zgłoszenia przerwania CPU, zatrzymania innych, mniej ważnych zadań, a następnie do wykonania procedury obsługi przerwania odpowiedniej wysłanemu sygnałowi sprzętowemu (określane jest to jako zgłoszenie przerwania). W przeciwieństwie do instrukcji CALL, która wymaga skanowania przed wykonaniem, funkcja przerwania sprzętowego realizowana jest w czasie rzeczywistym, w odpowiedzi do zdarzenia z zewnątrz. Ponadto, procedura obsługi przerwania sprzętowego, nie może być wywołana przez nazwę etykiety. Dlatego też istnieją „zarezerwowane słowa" wykorzystywane w nazwach etykiet odnoszące się do różnych funkcji przerwań oferowanych przez PLC (szczegóły w opisie FUN65). Na przykład, zarezerwowane słowo X0+I przypisane jest do przerwania na wejściu X0. Jeżeli program będzie zawierać etykietę X0+I, a na wejściu X0 pojaw się stan narastający (X0: □), PLC zatrzyma realizację programu o niższym priorytecie i przeskoczy do etykiety podporocedury oznaczonej jako X0+I, w celu natychmiastowej realizacji programu.

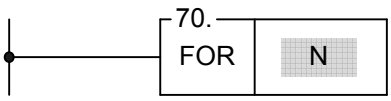
- Jeżeli zgłoszenie przerwania sprzętowego nastąpi w trakcie realizacji przez CPU procedury obsługi przerwania o wyższym priorytecie (np.: przerwania od szybkiego zegara sprzętowego) lub procedury obsługi przerwania o tym samym priorytecie (więcej informacji o poziomach priorytetów znajduje się w Rozdziale 10), to PLC nie wykona procedury obsługi przerwania do momentu, aż zostaną zakończone wszystkie procedury o wyższym priorytecie.

- Jeżeli instrukcja RTI nie może być wykonana w obrębie procedury obsługi przerwania, może to spowodować zatrzymanie CPU. W związku z tym, niezależnie od przebiegu wykonywania procedury obsługi przerwania, należy zapewnić wykonanie w niej instrukcji RTI.

- Szczegółowy opis i przykłady wykorzystania funkcji przerwań znajdują się w Rozdziale 9.

FUN 70 FOR	FOR	FUN 70 FOR

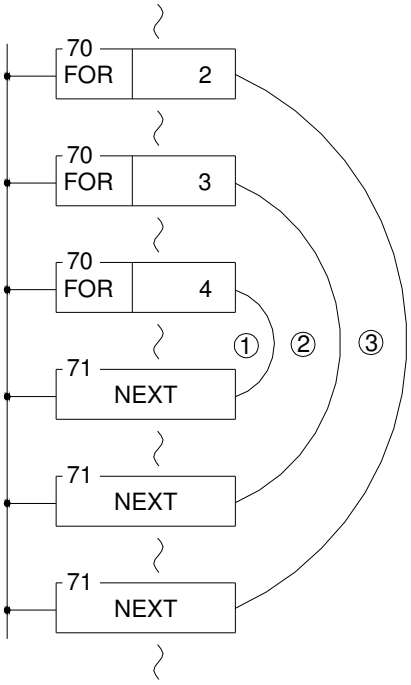
SYMBOL DRABINKOWY



N : Liczba wykonań pętli

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	1 □ 16383
N	○	○	○	○	○	○	○	○	○	○	○	○	○

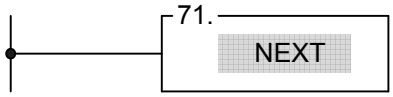
- Instrukcja nie ma wejścia sterującego, a więc nie ma możliwości podłączenia żadnych styków szeregowo i jest podłączona bezpośrednio do linii zasilającej.
- Programy z instrukcjami FOR i NEXT tworzą pętlę programową (początkiem pętli jest pierwsza instrukcja po instrukcji FOR, a końcem jest instrukcja przed NEXT). Kiedy PLC realizuje instrukcję FOR, najpierw zapisuje wartość N występującą przed pierwszym wykonaniem pętli (liczbę wykonań pętli), a następnie realizuje programy zapisany pomiędzy znacznikami FOR-NEXT,w pętli, N razy. Następnie PLC „wyskakuje” z pętli i kontynuuje wykonywanie instrukcji znajdującej się bezpośrednio po instrukcji NEXT.
- Pętle mogą mieć strukturę zagnieżdżoną, tj. jedna pętla może zawierać inne tworząc w ten sposób układ warstwowy. 1 pętla określana jest jako poziom, a maksymalna liczba poziomów zagnieżdżeń wynosi 5. Instrukcje FOR i NEXT muszą być stosowane parami. Pierwsza instrukcja FOR i ostatnia instrukcja NEXT stanowią najbardziej zewnętrzny (pierwszy) poziom pętli o strukturze zagnieżdżonej. Druga instrukcja FOR oraz przedostatnia instrukcja NEXT stanowią drugi poziom, natomiast ostatnia instrukcja FOR i pierwsza instrukcja NEXT tworzą najbardziej wewnętrzny poziom.



- W przykładzie po lewej, pętla ① zostanie zrealizowana 4 ×3 ×2 = 24 razy, pętla ② 3 ×2 = 6 razy, a pętla ③ 2 razy.
- Jeżeli instrukcji FOR nie odpowiada żadna instrukcja NEXT lub gdy instrukcje FOR i NEXT w danej pętli nie zostały użyte w parze lub też kolejność tych instrukcji została odwrócona, pojawi się błąd składni, a program nie zostanie zrealizowany.
- Instrukcja JMP może być wykorzystana do „wyskoczenia” z pętli. Należy jednak pamiętać, że po pierwszym wejściu w pętlę (i realizacji instrukcji FOR), niezależnie od przebiegu program musi zrealizować instrukcję NEXT przed przejściem do instrukcji END. W innym wypadku FBs-PLC zatrzyma operację i wygeneruje komunikat o błędzie.
- poprawnym zakresem N jest 1~16383 powtórzeń. Wartość N będącą poza tym zakresem FBs-PLC potraktowana zostanie jako 1. Należy pamiętać, że w przypadku zbyt dużej wartości N, a tym samym zbyt dużej liczbie pętli ,może pojawić się błąd WDT.

FUN 71 NEXT	KONIEC PĘTLI	FUN 71 NEXT

SYMBOL DRABINKOWY



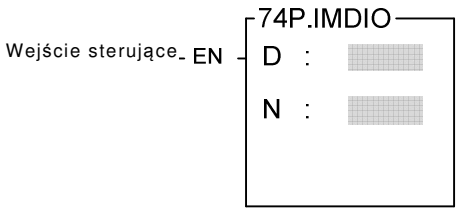
- Instrukcja ta łącznie z instrukcją FOR tworzą pętlę programową. Sama instrukcja nie ma wejścia sterującego, a więc nie ma możliwości podłączenia żadnych styków szeregowo i jest podłączona bezpośrednio do linii zasilającej

- Jeżeli PLC nie wszedł jeszcze w pętlę (nie została zrealizowana instrukcja FOR lub jej realizacja została przerwana), a osiągnięta została instrukcja NEXT, to PLC nie wykona żadnej akcji traktując instrukcję jako nieistniejąca.

- Dodatkowe informacje dotyczące tej instrukcji znajdują się w opisie instrukcji FOR na poprzedniej stronie.

FUN 74 P IMDIO	NATYCHMIASTOWE ODŚWIEŻENIE WE/WY (IMMEDIATE I/O REFRESH)	FUN 74 P IMDIO

SYMBOL DRABINKOWY



D : Adres początkowy We / Wy do odświeżenia
N : Liczba kolejnych We / Wy do odświeżenia

Zakres / Argument	X	Y	K
	Xn jednostki głównej	Yn jednostki głównej	1 □ 36
D	○	○	
N			○

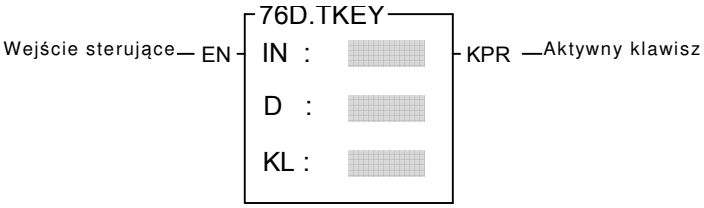
- Przy normalnym cyklu skanu PLC, CPU pobiera stany wszystkich sygnałów wejściowych przed rozpoczęciem wykonywania programu, a następnie na ich podstawie realizuje program. Po zakończeniu wykonywania programu, CPU aktualizuje sygnały wyjściowe bazując na otrzymanych wynikach algorytmu. Dopiero po całkowitym zakończeniu skanu, wszystkie wyniki są jednocześnie przesyłane na wyjście. Dlatego też, pomiędzy pojawieniem się sygnału na wejściu a odpowiedzią na wyjściu, zawsze będzie występowało opóźnienie przynajmniej jednego skanu sterownika (maks. 2 skany). Korzystając z tej instrukcji, możliwe jest natychmiastowe odświeżenie wskazanych wejść lub wyjść, w celu uzyskania szybszej reakcji, bez ograniczenia wynikającego ze sposobu wykonywania skanu przez sterownik.
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 1 na 0 (instrukcja P), to status N kolejnych punktów wejściowych lub wyjściowych (D~D+N-1) zostanie odświeżony.
- Dla sterowników serii FBs, liczba punktów We / Wy, jaka może być natychmiastowo odświeżona, ograniczona jest do liczby punktów We / Wy istniejących na jednostce głównej. W poniższej tabeli przedstawione są dopuszczalne liczby We / Wy dla jednostek głównych o rozmiarach 10,14, 20,24, 32, 40 i 60 punktów:

Typ jednostki głównej / Dopuszczalne liczby We / Wy	10 punktów	14 punktów	20 punktów	24 punktów	32 punktów	40 punktów	60 punktów
Sygnały wejściowe	X0～X5	X0～X7	X0～X11	X0～X13	X0～X19	X0～X23	X0～X35
Sygnały wyjściowe	Y0～Y3	Y0～Y5	Y0～Y7	Y0～Y9	Y0～Y11	Y0～Y15	Y0～Y23

- W przypadku, gdy zakładane sygnały odświeżania We / Wy będą poza zakresem określonym w powyższej tabeli, PLC nie wykona operacji odświeżenia, a znacznik błędu M1931 zostanie ustawiony na 1. Na przykład, jeżeli D=X11, N=10, to wejścia X11 do X20 miałyby być natychmiastowo odświeżone. Zakładając, że jednostką główną jest FBs-32MA, to jej największym punktem wejściowym jest X19, a punkty od X20 wykraczają poza zakres. Przy takim założeniu, przekroczony zostanie zakres i znacznik błędu M1931 zostanie ustawiony na 1.
- Instrukcja ta umożliwia szybkie odświeżenie sygnałów wejściowych / wyjściowych przez PLC. Jednakże opóźnienie filtrów softwarowych i hardwarowych nadal istnieje. Należy zwrócić na to uwagę.

FUN 76 D TKEY	WPROWADZANIE WARTOŚCI DZIESIĘTNYCH Z KLAWISZY (DECIMAL- KEY INPUT)	FUN 76 D TKEY

SYMBOL DRABINKOWY



IN : Początkowy adres wejścia dla klawiszy

D : Rejestr przechowujący wpisane liczby

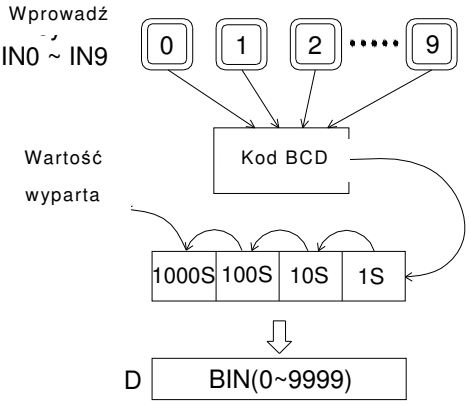
KL: Cewka początkowa odzwierciedlająca stan na wejściu

D może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

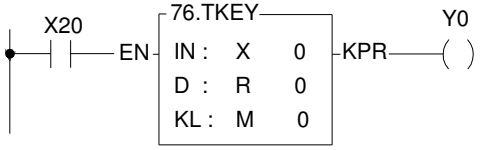
Zakres / Argument	X	Y	M	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 □ X240	Y0 □ Y240	M0 □ M1896	S0 □ S984	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
IN	○														
D					○	○	○	○	○	○	○	○*	○*	○	○
KL		○	○	○											

- Funkcja ta wykorzystuje 10 punktów wejściowych IN~IN+9 (IN0~IN9), gdzie każde przeznaczone jest do wpisywania jednej cyfry (IN->0, IN+1->1…). W zależności od sekwencji wprowadzania tych punktów wejściowych, istnieje możliwość wpisania 4 lub 8 liczb dziesiętnych do rejestrów wskazanych w polu D.

- Jeżeli „EN” = 1, funkcja będzie monitorować 10 punktów wejściowych rozpoczynając on IN i zapisze odpowiednią liczbę w rejestrze D w momencie, gdy któryś z przycisków zostanie wciśnięty. Po zwolnieniu przycisku instrukcja zacznie oczekiwać na następny punkt wejściowy w stanie „ON” i przemieści kolejną liczbę do rejestru D (wcześniej wprowadzona cyfra jest starsza od wprowadzonej później). W przypadku argumentu 16-bitowego, w rejestrze może być zapisane do 4 cyfr ,a 8 cyfr dla argumentu 32-bitowego. Po zapełnieniu rejestru, nowo wprowadzona cyfra wyprze najstarszą cyfrę z rejestru D. Stan 10 punktów wejściowych rozpoczynając od IN zapisywany będzie w 10 odpowiadających im cewkach rozpoczynając od KL. Cewki te będą ustawiane na 1 w czasie gdy odpowiadające im przyciski będą naciśnięte. Przy zwolnionych przyciskach, ich statusy pozostaną niezmienione. Ich status zostanie odświeżony w momencie naciśnięcia kolejnego przycisku. Przez cały czas, gdy którykolwiek przycisk będzie wciśnięty (ON), znacznik wpisywania KPR będzie ustawiony na 1. W jednym momencie może być wciśnięty tylko jeden przycisk spośród IN0~IN9. W przypadku, gdy zostanie wciśniętych kilka przycisków, wprowadzona będzie tylko wartość pierwszego z nich. Poniżej przedstawiony został schemat poglądowy funkcji z argumentem 16-bitowym.



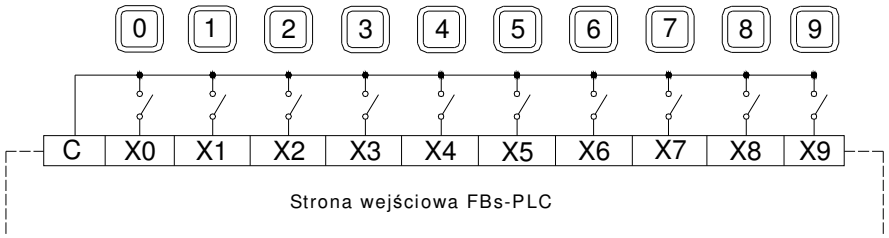
- Jeżeli „EN” = 0, instrukcja ta nie zostanie wykonana. Status wyjścia KPR i cewek KL będzie ustawiony na 0. Jednakże wartości numeryczne w rejestrze D pozostaną niezmienione.



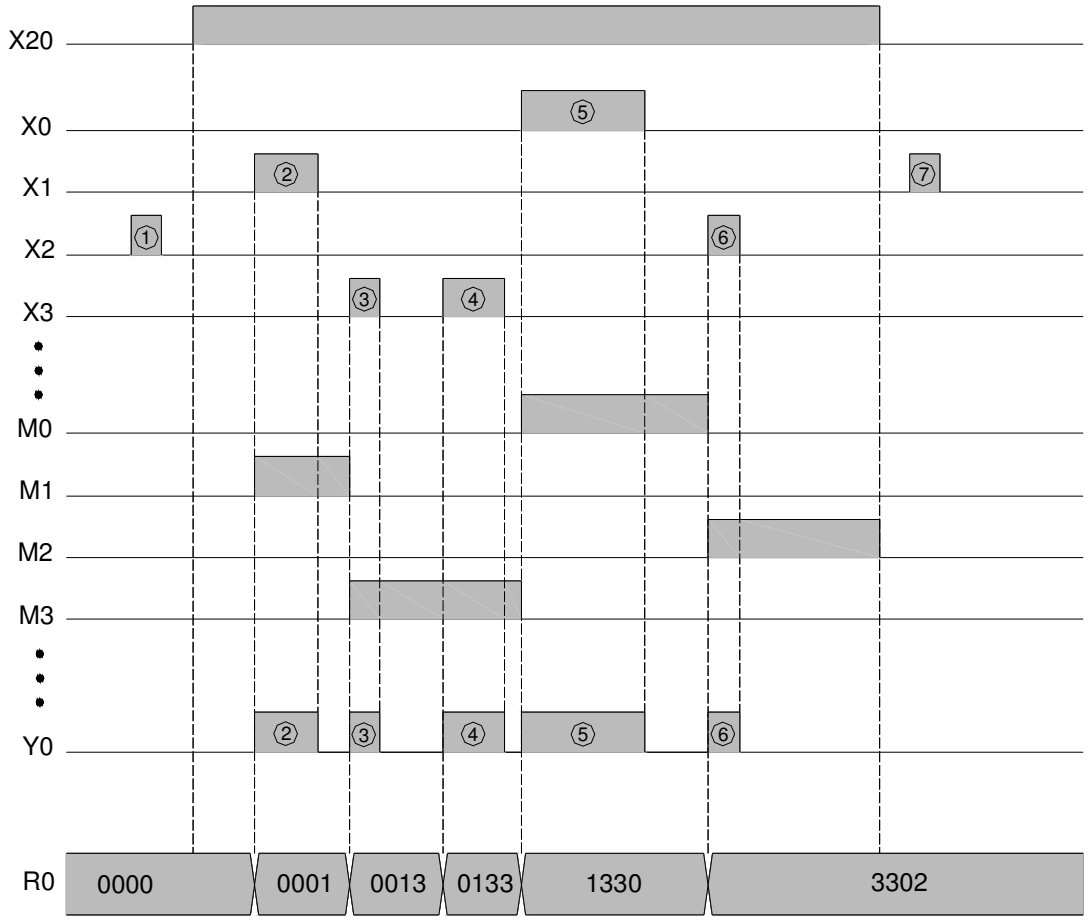
* W instrukcji po lewej, punkt wejściowy X0 reprezentowany jest przez cyfrę „0”, X1 przez cyfrę 1, …, M0 zapisuje czynność X0, M1 zapisuje czynność X1,…, a wejściowe wartości numeryczne zapisywane są w rejestrze R0.

FUN 76 D TKEY	WPROWADZANIE WARTOŚCI DZIESIĘTNYCH Z KLAWISZY (DECIMAL- KEY INPUT)	FUN 76 D TKEY

Poniżej rysunek jest schematem podłączenia wejść dla przykładu:

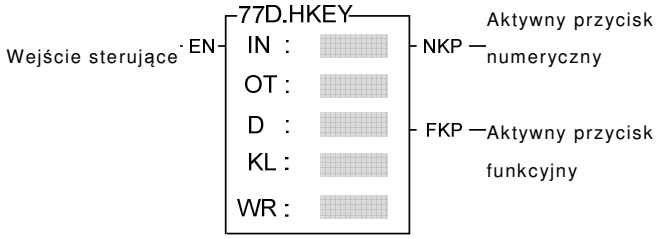


- Jeżeli w poniższym schemacie, klawisze X0~X3 są naciskane w sekwencji ①②③④⑤⑥⑦, a przy kroku ① i ⑦ X20 będzie w stanie 0, to efektywne będą tylko kroki ②③④⑤⑥. Ponieważ w rejestrze mogą być zapisane tylko 4 cyfry, jeden krok spośród pięciu zostanie wyparty. Ostatecznie wartości klawiszy 3302 z kroków ③④⑤⑥ zapisywane są w rejestrze R0.



FUN 77 D HKEY	WEJSCIE Z KLAWIATURY 16TO ZNAKOWEJ (HEX-KEY INPUT)	FUN 77 D HKEY

SYMBOL DRABINKOWY



IN : Początkowy punkt wejściowy do skanowania przycisków

OT: Początkowy punkt wyjściowy do multipleksowego skanowania przycisków (4 punkty)

D : Rejestr do zapisu wprowadzonych liczb

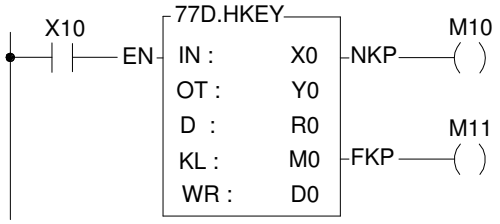
KL : Początkowy przekaźnik stanu przycisku

WR: Rejestr roboczy; nie może być ponownie użyty

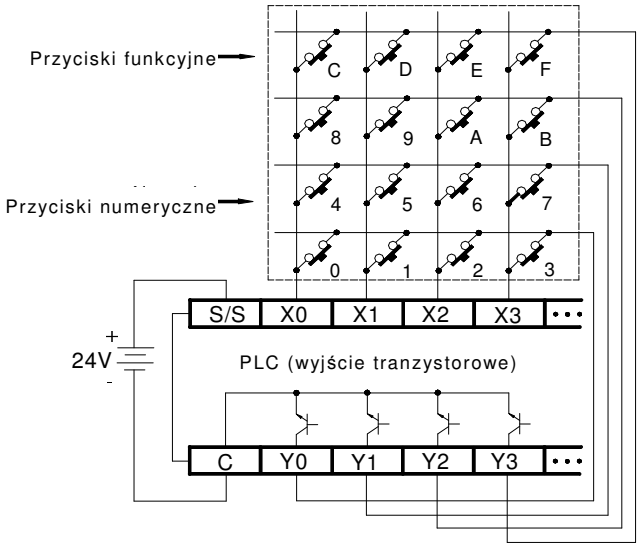
D może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	X	Y	M	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 □ X240	Y0 □ Y240	M0 □ M1896	S0 □ S984	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
IN	○														
OT		○													
D					○	○	○	○	○	○	○	○*	○*	○	○
KL		○	○	○											

- Funkcja przycisków numerycznych (0~9) w tej instrukcji jest zbliżona do instrukcji TKEY. Instrukcje TKEY i HKEY różnią się połączeniem sprzętowym. W przypadku instrukcji TKEY, każdy przycisk charakteryzuje się jednym punktem wejściowym do połączenia, podczas gdy HKEY wykorzystuje 4 punkty wejściowe i 4 punkty wyjściowe tworząc wejście multipleksowe 4x4 z 16 przyciskami. 4x4 oznacza, że możliwość wykorzystania 16 przycisków, a co za tym idzie, oprócz 10 przycisków numerycznych można wykorzystać dodatkowe 6 przycisków funkcyjnych (tak jak w przypadku standardowego wejścia dyskretnego). Czynności przycisków numerycznych i funkcyjnych są niezależne i nie wpływają na siebie wzajemnie.
- Jeżeli „EN" = 1, instrukcja przeskanuje przyciski numeryczne i funkcyjne w macierzy utworzonej przez 4 punkty wejściowe rozpoczynając od IN oraz 4 punkty wyjściowe rozpoczynając od OT. Funkcje przycisków numerycznych i wyjścia „NKP" znajdują się w opisie instrukcji TKEY. Przyciski funkcyjne utrzymają status przycisków A~F w ostatnich 6 przekaźnikach określonych przez KL (w pierwszych 10 zapisywany jest status przycisków numerycznych). Jeżeli którykolwiek z przycisków A~F będzie naciśnięty, status wyjścia FKP (FO1) zostanie ustawiony na 1. Punkty wyjściowe OT dla tej instrukcji muszą być wyjściami tranzystorowymi.
- Największa liczba dla argumentu 16-bitowego składa się z 4 cyfr (9999), natomiast dla argumentu 32-bitowego – z 8 cyfr (99999999). Jednakże niezależnie od tego, czy argument jest 16- czy 32-bitowy, istnieje jedynie 6 przycisków funkcyjnych (A~F).

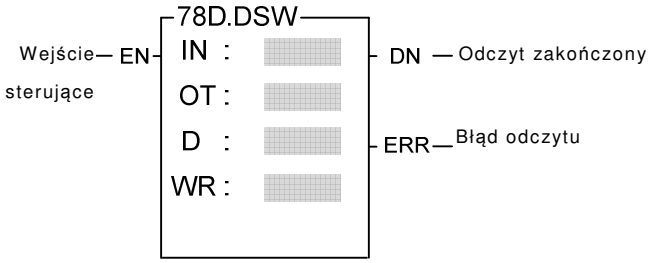


* Powyższa instrukcja wykorzystuje X0~X3 i Y0~Y3 do stworzenia wejścia multipleksowego. Umożliwia ona wprowadzanie 8-cyfrowych wartości numerycznych i zapisywanie wyników w R1R0. Statusy przycisków funkcyjnych zapisywane są w M10(A)~M15(F).



FUN 78 D DSW	WEJSCIE Z ZADAJNIKA TARCZOWEGO (DIGITAL SWITCH INPUT)	FUN 78 D DSW

SYMBOL DRABINKOWY



IN : Początkowy punkt wejściowy dla przełącznika tarczowego

OT : Początkowy punkt wyjściowy dla skanowania multipleksowego (4 punkty)

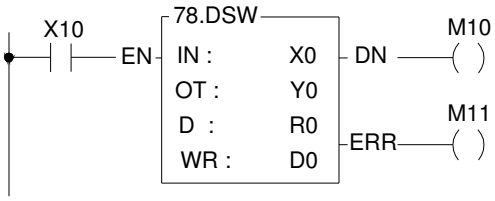
D : Rejestr do zapisu odczytanej wartości

WR: Rejestr roboczy; nie może być ponownie użyty
(WR i WR+1 dla operacji 16-bitowej;
WR, WR+1 i WR+2 dla operacji 32-bitowej)

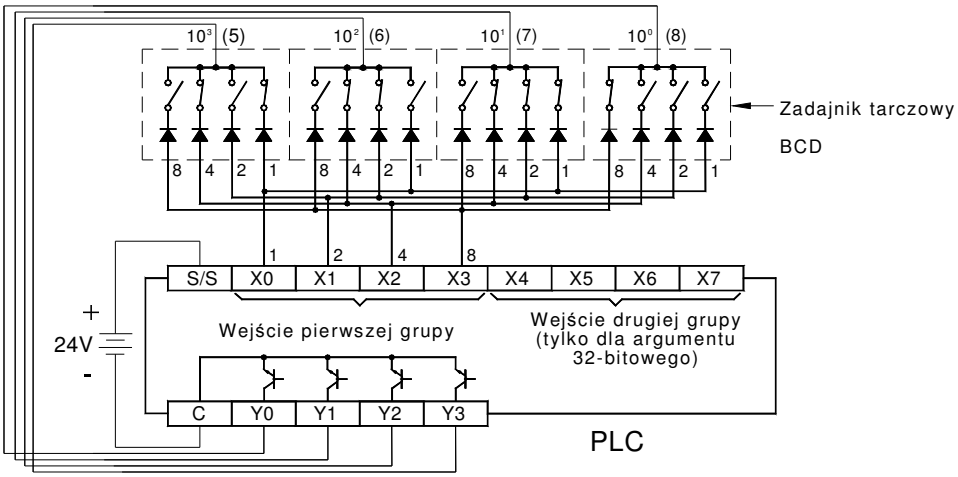
D może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	X	Y	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 □ X240	Y0 □ Y240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	V、Z P0~P9
IN	○												
OT		○											
D			○	○	○	○	○	○	○	○*	○*	○	○

- Jeżeli „EN” = 1, instrukcja odczyta dane z 4 punktów wejściowych rozpoczynając od IN (IN0~IN3). Do odczytania grupy 4-cufrowych wartości BCD (0000~9999) i zapisania ich w rejestrze D potrzebne są 4 operacje skanowania. W przypadku argumentu 32-bitowego, przy każdym skanowaniu odczytywane są 2 cyfry danych poprzez odczyt dodatkowej danej z IN4-IN7 i zapisanie jej w rejestrze D+1. Każdy bit OT0~OT3 będzie ustawiany sekwencyjnie na 1 i dane zapisywane będą odpowiednio do 10^0(jednostek), 10^1(dziesiątek), 10^2(setek) i 10^3(tysięcy). Dopóki EN = 1, PLC będzie skanować i odczytywać w sposób ciągły i cykliczny. Po zakończeniu każdego pełnego cyklu (np.: po odczytaniu 4 cyfr 10^0~10^3), znacznik zakończonego odczytu „DN” zostanie ustawiony na 1. Stan ten będzie jednak utrzymywany tylko na czas jednego skanowania. W przypadku, gdy którakolwiek odczytana wartość będzie poza zakresem 0~9 (BCD), znacznik błędu odczytu „ERR” zostanie ustawiony na 1, a wartość danej grupy cyfr zostanie ustawiona na 0000.
- Punkty wyjściowe muszą być wyjściami tranzystorowymi.

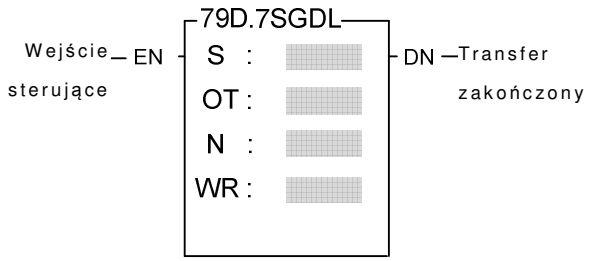


- W przykładzie po lewej, jeżeli X10 ma status 1, to wartość numeryczna przełącznika tarczowego (w tym przykładzie 5678) zostanie odczytana i zapisana w rejestrze R0.
- Bity (8,4,2,1) z tą samą cyfrą powinny być połączone ze sobą szeregowo za pomocą diod (jak pokazano na poniższym schemacie).
- W przypadku argumentu 32-bitowego, zestaw podobnych przełączników tarczowych może być podłączony do X4~X7 (Y0~Y3 są współdzielone z tą drugą grupą).



FUN 79 D 7SGDL	WYJŚCIE 7-SEGMENTOWE Z ZATRZASKIEM (7-SEGMENT OUTPUT WITH LATCH)	FUN 79 D 7SGDL

SYMBOL DRABINKOWY



S : Rejestr przechowujący dane (BCD) do wyświetlenia

OT : Początkowy adres wyjścia skanującego

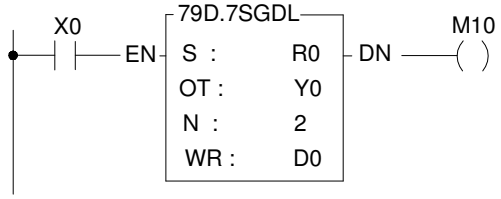
N : Wyjście sygnałowe i polaryzacja zatrzasku

WR : Rejestry robocze. Nie mogą być ponownie użyte

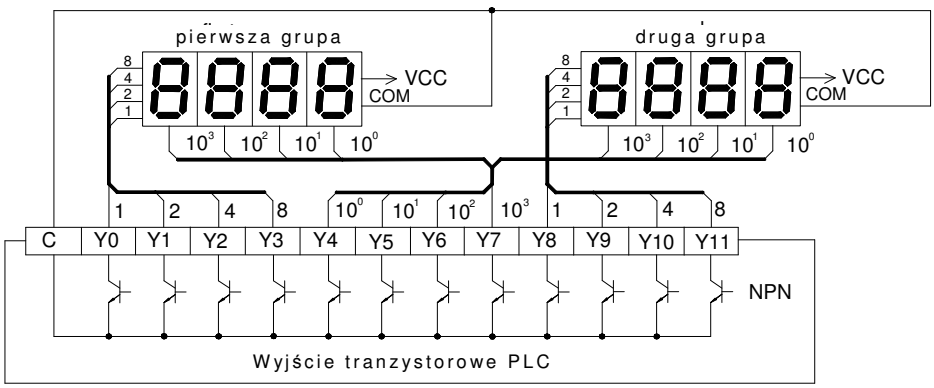
S może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	Y0 □ Y240	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	Liczba 16-bit	V、Z P0~P9
S			○	○	○	○	○	○	○	○	○	○	○	○	○
OT	○														
N														0～3	

- Jeżeli wejście sterujące „EN” = 1, 4 półbajty rejestru S, od cyfry 0 do cyfry 3, są cyklicznie wysyłane do 4 punktów wyjściowych OT0~OT3. Podczas wystawiania danych na wyjścia dla danej cyfry, wysyłany jest także jej sygnał zatrzaskujący (OT4 odpowiada cyfrze 0, OT5 cyfrze 1, itd.). W ten sposób, wartość cyfrowa zostanie załadowana i przesłana do 7-segmentowego wyświetlacza.
- W przypadku zastosowania instrukcji D (32-bitowej), półbajty 0~3 z rejestru S i półbajty 0~3 z rejestru S+1 wysyłane są oddzielnie do OT0~OT3 i OT8~OT11. Ponieważ przekazywane są one w tym samym czasie, mogą wykorzystywać ten sam sygnał zatrzaskujący. 16-bitowe instrukcje nie wykorzystują OT8~OT11.
- Do momentu, aż status „EN” ustawiony będzie na 1, PLC będzie przekazywał półbajty cyklicznie. Po każdym transferze pełnej grupy wartości numerycznych (półbajtów 0~3 lub 0~7), znacznik zakończenia transferu „DN” zostanie ustawiony na 1. Jednakże stan ten zostanie utrzymany tylko na czas jednego skanowania.



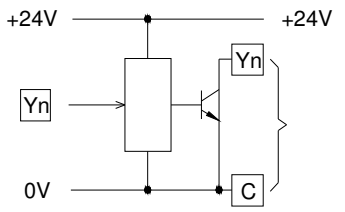
- W przykładzie po lewej, jeżeli X0=1, to 4 półbajty z R0 zostaną przekazane do 7-segmentowego wyświetlacza pierwszej grupy (schemat poniżej). 4 półbajty z R1 zostaną natomiast przekazane do 7-segmentowego wyświetlacza drugiej grupy.



FUN 79 D 7SGDL	WYJŚCIE 7-SEGMENTOWE Z ZATRZASKIEM (7-SEGMENT OUTPUT WITH LATCH)	FUN 79 D 7SGDL

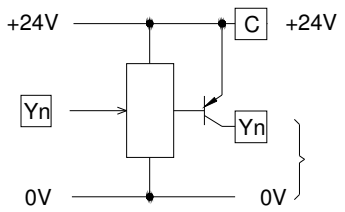
- Wyjście tranzystorowe sterowników PLC FATEK są dwojakiego rodzaju: z negatywną logiką wyjścia tranzystorowego (tranzystor NPN – przy stanie ON, na wyjściu tranzystora pojawia się potencjał masy) i dodatnią logiką wyjścia tranzystorowego (PNP – przy stanie ON, na wyjściu tranzystora pojawia się potencjał dodatni). Budowa wyjść przedstawiona została poniżej:

Wyjście z logiką negatywną FBs-PLC (tranzystor NPN)



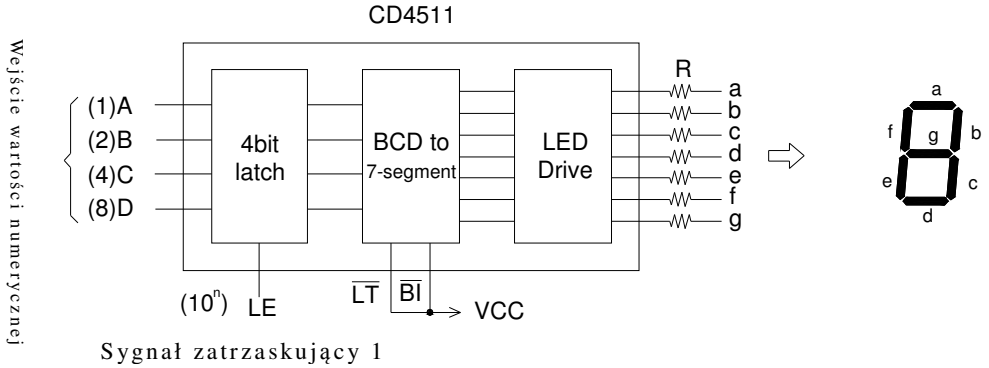
Kiedy Yn ma status ON, na wyjściu jest potencjał masy

Wyjście z logiką dodatnią FBs-PLC (tranzystor PNP)



Kiedy Yn ma status ON, na wyjściu jest potencjał dodatni

- W 7-segmentowych wyświetlaczach z zatrzaskiem, wejścia danych (8, 4, 2, 1) oraz sygnał zatrzaskujący, mogą występować jako przystosowane do współpracy z jednym lub drugim typem polaryzacji. Na przykład, dla wartości numerycznej „8”, i logice dodatniej sterownia, wejścia logiczne powinny mieć stany 1000, natomiast dla logiki negatywnej 0111. Podobnie, jeżeli status sygnału zatrzaskującego jest 0, to dodatni logiczny sygnał zatrzaskujący umożliwi wyświetlenie wartości numerycznych do wprowadzenia przez zatrzask (tj. załadowania). Jeżeli status sygnału zatrzaskującego jest 1, wartości numeryczne są zatrzaskiwane (zachowywane). Odwrotnie jest w przypadku ujemnego sygnału logicznego. Poniższy schemat 7-segmentowego wyświetlacza CD-4511 przedstawia przykład wejścia wartości numerycznych z zatrzaskiem, wykorzystującego logikę dodatnią.



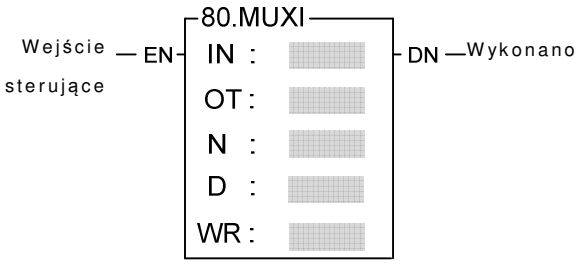
- Z uwagı na to, że polaryzacja wyjścia PLC i wejścia 7-segmentowego wyświetlacza może być dodatnia lub ujemna, to polaryzacje pomiędzy wyjściem a wejściem muszą być koordynowane w celu uzyskania prawidłowego wyniku. Instrukcja ta wykorzystuje N do określenia relacji polaryzacji pomiędzy wyjściem tranzystorowym PLC a wejściem 7-segmentowego wyświetlacza. Poniższa tabela przedstawia wszystkie możliwości takich relacji.

Wejście wartości numerycznej (8~1)	Sygnał zatrzaskujący (10^0-10^3)	Wartość N
Identyczne	Identyczny	0
	Różny	1
Różne	Identyczny	2
	Różny	3

- W powyższym schemacie CD4511 użyto jako przykład. W przypadku wykorzystania wyjścia NPN, polaryzacja wejścia danych jest inna niż PLC, a polaryzacja wejścia zatrzaskującego jest taka sama jak PLC. Wartość N wynosi zatem 2.

FUN 80 MUXI	WEJŚCIE MULTIPLEKSOWE (MULTIPLEX INPUT)	FUN 80 MUXI

SYMBOL DRABINKOWY



IN :Adres początkowy wejść multipleksowanych

OT : Adres początkowy wyjść multipleksujących (wyjście musi być tranzystorowe)

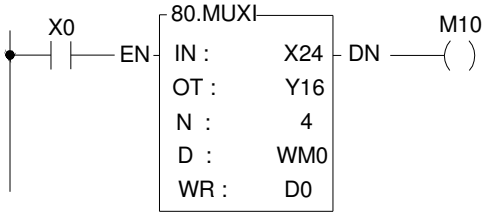
N : Liczba multipleksowanych linii (2~8)

D : Rejestr do zapisu wyników

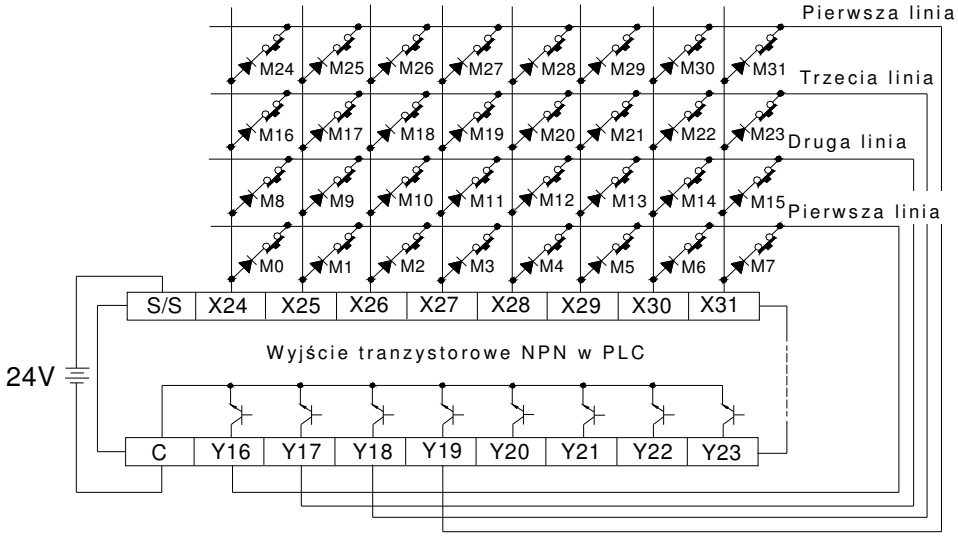
D może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	X	Y	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	X0 □ X240	Y0 □ Y240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 8	V、Z P0~P0
IN	○													
OT		○												
N													○	
D			○	○	○	○	○	○	○	○*	○*	○		○

- Instrukcja ta wykorzystuje metodę multipleksową do odczytywania N linii stanów z 8 kolejnych wejść (IN0~IN7) rozpoczynając od wejścia oznaczonego jako IN. Dzięki tej metodzie można odczytać stany z 8xN wejść przy wykorzystaniu jedynie 8 wejść i N wyjść.
- Metoda skanowania multipleksowego polega na skanowaniu kolejno N wyjść, rozpoczynając od wyjścia OT. Podczas każdego skanowania, jedno z wyjść N ustawione jest na 1 – w ten sposób wybierana jest odpowiednia linia. OT+0 odpowiada pierwszej linii, OT+1 drugiej, itd. Po odczytaniu stanów z wszystkich N linii, stany 8xN wejść, zapisywane są w rejestrze D, a znacznik wykonania funkcji „DN" ustawiany jest na 1 (na czas jednego skanu programu).
- W każdym skanie programu, funkcja odczytuje stany z jednej linii spośród N, zatem dla wykonania funkcji potrzebnych jest N skanów programu.

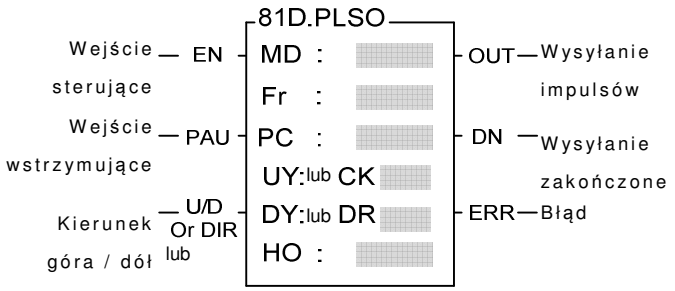


- W przykładzie po lewej odczytywane są 4 linie na 8 wejściach co łącznie daje 32 stany. Wyniki zapisywane są w 32-biitowym rejestrze DWM0 (M0~M31).



FUN 81 D PLSO	WYJŚCIE IMPULSOWE	FUN 81 D PLSO

SYMBOL DRABINKOWY



MD : Wybór trybu wysyłania impulsów
Fr : Zadana częstotliwość impulsów
PC : Ilość impulsów do wysłania
UY : Adres wyjścia wysyłającego impulsy w górę (MD=0).
DY : Adres wyjścia wysyłającego impulsy w dół (MD=0).
HO : Rejestr przechowujący liczbę wysłanych impulsów.
CK : Adres wyjścia impulsującego (MD=1).
DR : Adres wyjścia kierunku (MD=1).

DIR: 1- w górę; 0- w dół.

Zakres / Argument	Y	WX	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K
	Yn Jedn. główn.	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- liczba
MD													0～1
Fr		○	○	○	○	○	○	○	○	○	○	○	8～2000
PC		○	○	○	○	○	○	○	○	○	○	○	○
UY，CK	○												
DY，DR	○												
HO			○	○	○	○	○	○	○	○*	○*	○	

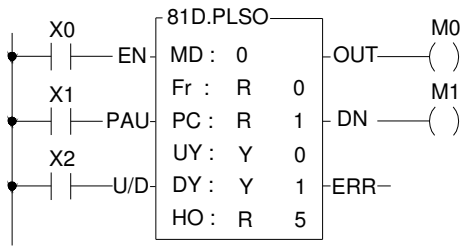
- Jeżeli MD=0, instrukcja ta zrealizuje sterowanie wyjściem impulsowym w następujący sposób:
- Jeżeli „EN” zmieni się z 0→1, nastąpi reset, polegający na ustawieniu znaczników „OUT” i „DN” w stan niski oraz rejestru liczby wysłanych impulsów HO na wartość 0. Funkcja pobierze częstotliwość impulsów i ilość impulsów do wysłania, oraz odczyta stan wejścia określającego kierunek wysyłania umpulsów „U/D” w celu ustalenia kierunku wysyłania: w dół lub w górę. Po zakończeniu resetu instrukcja sprawdzi stan wejścia wstrzymującego wysyłanie impulsów „PAU”. W przypadku, gdy stan wejścia wstrzymującego jest równy 1 (wstrzymaj wysyłanie impulsów), nie zostanie wykonana żadna akcja. Jeżeli PAU = 0, zostaną wysłane impuls WŁ/WYŁ z wypełnieniem 50% i częstotliwością Fr na wyjście UY(U/D=1) lub DY(U/D=0). Wartość w rejestrze HO zostanie powiększona po każdym wysłaniu impulsu. Wysyłanie impulsów zostanie przerwane w momencie, gdy ilość wysłanych impulsów HO będzie równa lub większa od zadanej liczby impulsów do wysłania PC, a znacznik zakończonego wysyłania „DN” zostanie ustawiony na 1. Podczas wysyłania impulsów, znacznik wysyłania „OUT” zostanie ustawiony na 1. W przeciwnym wypadku będzie miał stan 0.
- Po rozpoczęciu wysyłania impulsu, stan na wejściu sterującym „EN” powinien być podtrzymywany w stanie 1. Jeżeli zmieni się na stan 0, wysyłanie impulsu zostanie zatrzymane (wyjście wysyłające impulsy zostanie ustawione w stan 0), a znacznik „OUT” zmieni się ponownie na stan 0. Pozostałe stany i dane pozostaną niezmienione. Jednakże, jeżeli „EN” ponownie zmieni się z 0 na 1, nastąpi reset i cała procedura rozpocznie się od początku.
- W przypadku konieczności wstrzymania wysyłania impulsów, bez jednoczesnego resetowania procedury, należy użyć wejścia wstrzymującego „PAU”. Jeżeli „PAU”=1, instrukcja zatrzyma wysyłanie impulsów (wyjście wysyłające impulsy zostanie ustawione w stan 0, znacznik „OUT” zmieni się na 0, a pozostałe stany i dane nie ulegną zmianie). Po ponownej zmianie wejścia „PAU” z 1 na 0, instrukcja ta powróci do stanu sprzed wstrzymania i będzie kontynuować wysyłanie impulsów.
- Podczas wysyłanie impulsów, instrukcja ta monitoruje wartość zadanej częstotliwości impulsów Fr oraz ilość impulsów do wysłania PC. Dlatego też, dopóki wysyłanie impulsów nie zostanie zakończone, częstotliwość i ilość impulsów do wysłania, mogą być zmieniane. Jednakże, stan wejścia określającego kierunek wysyłania impulsów „U/D”, może być odczytany tylko po resecie (gdy „EN” zmieni się z 0→1). Stan ten będzie utrzymany do momentu zakończenia wysyłania impulsów lub kolejnego resetu. Poza samym momentem resetu, zmiana stanu na wejściu „U/D”, nie wpływa na działanie tej instrukcji.
- Głównym zastosowaniem tej funkcji jest sterowanie silnikiem krokowym za pomocą impulsów UY (w górę) i DY (w dół), w celu kontroli jego obrotu w przód lub wstecz. Niemniej jednak, jeżeli wymagana jest kontrola obrotu tylko w jedną stronę, można wypełnić tylko pole UY lub DY, a drugie pozostawić puste. W takim wypadku instrukcja zignoruje stan wejścia góra/dół „U/D” i wyśle impuls na wyjście wskazane.

FUN 81 D PLSO	WYJŚCIE IMPULSOWE (PULSE OUTPUT)	FUN 81 D PLSO

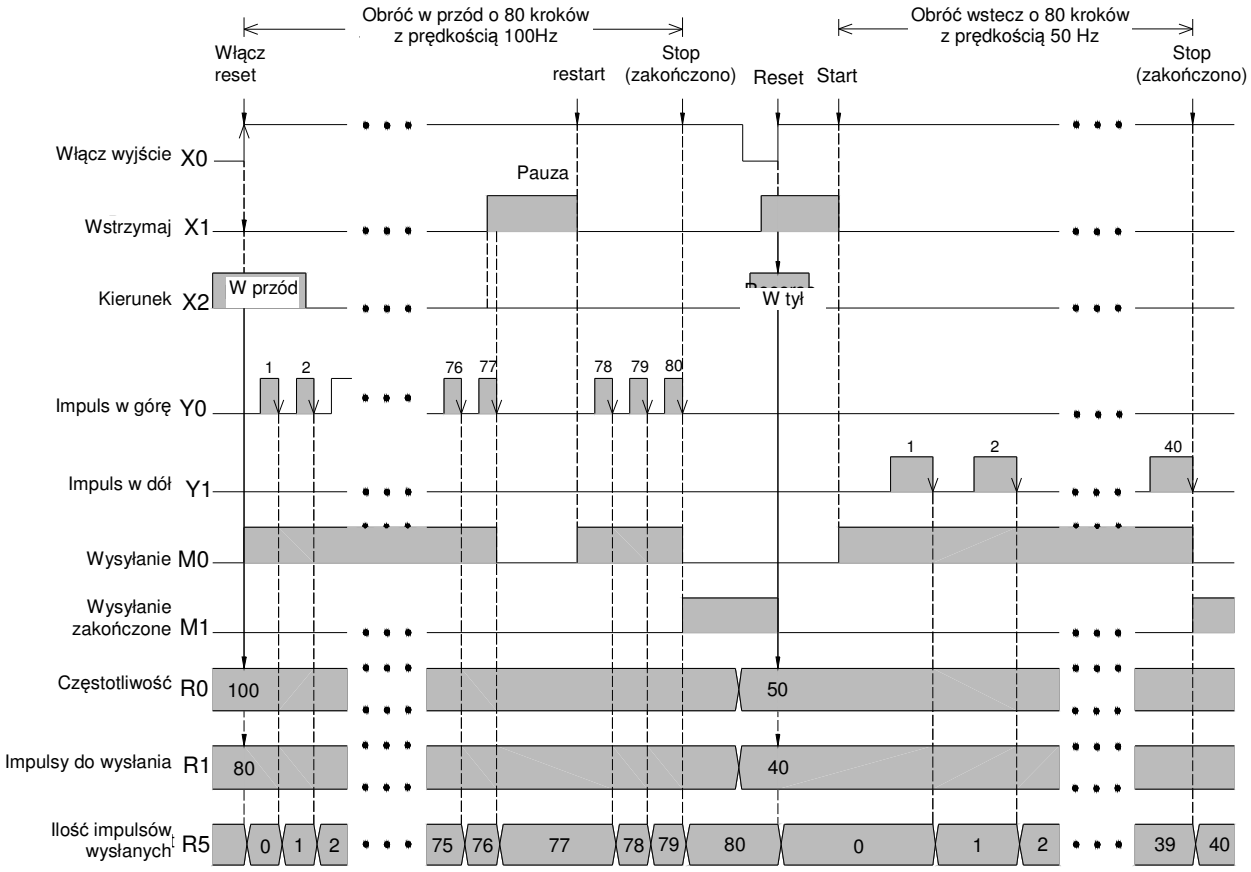
- Jeżeli MD = 1, tryb wysyłania impulsów przez funkcję zmieni się na „krok-kierunek”, gdzie wyjście DR określa kierunek impulsów (DIR = 1 – w górę; DIR = 0 = w dół), a wyjście CK jest wyjściem impulsującym.

- Instrukcja ta może być użyta w programie tylko raz, a UY (CK) i DY (DR) muszą być adresami tranzystorowych wyjść jednostki głównej sterownika PLC.

- Poprawnym zakresem ilości impulsów do wysłania PC, dla argumentu 16-bitowego, jest 0～32767. Dla argumentu 32-bitowego (instrukcja D), zakres ten wynosi 0～2147483647. Jeżeli PC = 0, zadana ilość impulsów do wysłania, traktowana jest jako nieskończona. W takim wypadku funkcja będzie wysyłać impulsy nieskończenie, ich liczba zapisywana będzie w HO a znacznik „DN” ustawiony zostanie na 0. Poprawnym zakresem częstotliwości wysyłanych impulsów (Fr) jest 8～2000. Jeżeli wartość PC lub Fr przekroczy zakres, instrukcja nie zostanie wykonana, a znacznik błędu „ERR” zostanie ustawiony na 1.

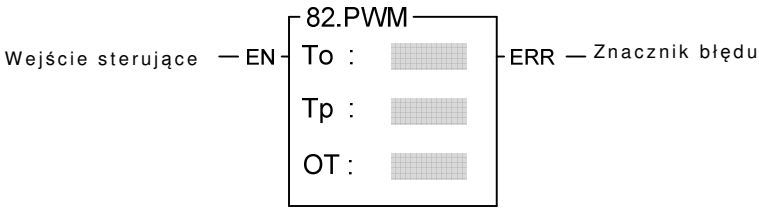


- W przykładzie po lewej, program steruje silnikiem krokowym obracając go w przód o 80 impulsów (kroków) z prędkością 100Hz, a następnie wstecz o 40 impulsów z prędkością 50Hz. Przed resetem (zmiana „EN” z 0→1). należy ustawić kierunek góra / dół, częstotliwość Fr i ilość impulsów do wysłania PC .



FUN 82 PWM	MODULACJA SZEROKOŚCI IMPULSU (PULSE WIDTH MODULATION)	FUN 82 PWM

SYMBOL DRABINKOWY



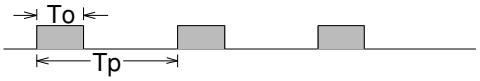
To : Czas załączenia impulsu (0~32767ms)

Tp : Okres (1~32676ms)

OT: Adres wyjścia impulsowego

Zakres / Argument	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	Yn jedn. główn.	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	0 □ 32767
To		○	○	○	○	○	○	○	○	○	○	○	○	○
Tp		○	○	○	○	○	○	○	○	○	○	○	○	○
OT	○													

- Jeżeli wejście sterujące „EN” = 1, funkcja wysyła impulsy na wyjście impulsujące OT, o stanie „ON” trwający To ms i okresie Tp. OT musi być adresem wyjścia tranzystorowego jednostki głównej. Jeżeli „EN” = 0, wyjście impulsujące będzie w stanie 0.

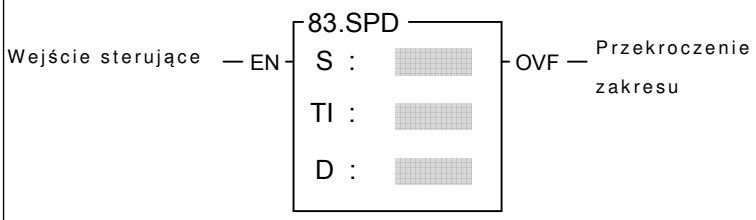


- Jednostkami Tp i To są ms, z dokładnością 1 ms. Minimalną wartością To jest 0 (w takim wypadku wyjście OT będzie zawsze nieaktywne), a maksymalną jest wartość Tp (w takim wypadku wyjście OT będzie zawsze aktywne). Jeżeli To > Tp, funkcja nie zostanie wykonana, a znacznik błędu „ERR” zostanie ustawiony na 1.

- Instrukcja ta może być użyta w programie tylko raz.

FUN 83 SPD	CZĘSTOTLIWOŚĆ NA WEJŚCIU (SPEED DETECTION)	FUN 83 SPD

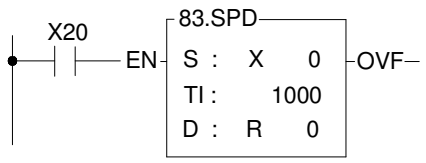
SYMBOL DRABINKOWY

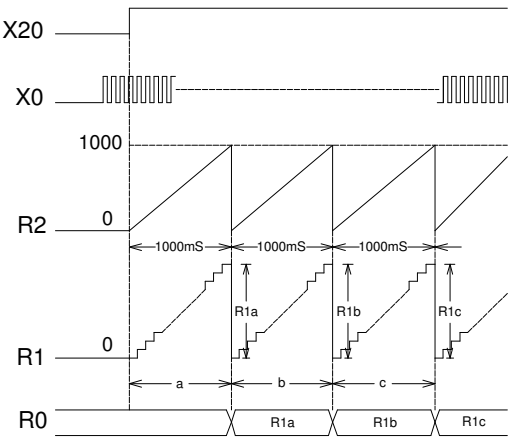


S : Adres wejścia do pomiaru częstotliwości
TI : Czas próbkowania (w ms)
D : Adres rejestru przechowującego wynik pomiaru

Zakres / Argum.	X	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	X0 □ X7	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	1 □ 32767
S	○													
TI		○	○	○	○	○	○	○	○	○	○	○	○	○
D			○	○	○	○	○	○	○	○	○*	○*	○	

- Instrukcja ta wykorzystuje funkcję przerwań sprzętowych 8 szybkich wejść (X0~X7) w jednostce głównej PLC, w celu pomiaru częstotliwości sygnału wejściowego. W określonym czasie próbkowania (TI), instrukcja zlicza ilość impulsów na wejściu S i pośrednio wyznaczy prędkość obrotu elementu (np.: silnika).
- Przy wykorzystaniu tej instrukcji do detekcji prędkości urządzeń, aplikacja powinna mieć możliwość wygenerowania dużej ilości impulsów na obrót w celu osiągnięcia większej rozdzielczości wyniku pomiaru. Jednakże suma częstotliwości wejściowych wszystkich mierzonych sygnałów, powinna być mniejsza niż 5 kHz. W innym wypadku może wystąpić błąd WDT.
- Rejestry D przechowujące wynik pomiaru częstotliwości, wykorzystują trzy kolejne 16-bitowe rejestry (D0~D2). Poza D0, który wynik pomiaru, rejestry D1 i D2 wykorzystywane są do przechowywania liczby aktualnie zliczonych impulsów i czasu pomiaru.
- Jeżeli „EN" = 1, instrukcja rozpocznie zliczanie impulsów na wejściu S, zapisując liczbę zliczonych impulsów w rejestrze D1. W tym samym czasie aktywowany jest timer próbkowania (D2), który zwiększa swoją wartość do momentu, aż wartość D2 osiągnie wartość czasu próbkowania (TI). Ostateczna zliczona ilość impulsów, jaka pojawiła się na wejściu S w czasie Ti, zapisywana jest w rejestrze D0 i rozpoczyna się nowy cykl. Zliczanie będzie powtarzane cyklicznie do momentu aż EN" = 0.
- Ponieważ D0 jest rejestrem 16to bitowym, maksymalna ilość impulsów zliczonych w czasie TI wynosi 32767. Jeżeli czas próbkowania TI jest zbyt długi lub częstotliwość na wejściu S jest za duża, zliczona wartość może przekroczyć 32767. W takim wypadku znacznik przekroczenia zakresu OVF zostanie ustawiony na 1, a zliczanie zostanie zatrzymane.
- Ponieważ czas próbkowania TI jest znany, a każdy obrót urządzenia generuje „n" impulsów, to prędkość obrotową urządzenia można obliczyć za pomocą następującego wzoru : $N = \frac{(D0) \times 60}{n \times TI} \times 10^3$ (rpm)



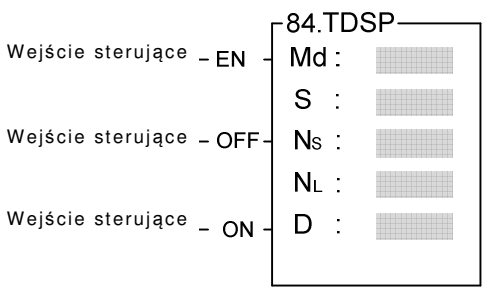


- W przykładzie powyżej, jeżeli każdy obrót urządzenia generuje 20 impulsów (n = 20), a wartość R0 = 200, to prędkość w obrotach na minutę wyniesie

$$N = \frac{(200) \times 60}{60 \times 1000} \times 10^3 = 200\,\text{rpm}$$

FUN 84 TDSP	KONWERSJA UKŁADU PÓL DLA WYŚWIETLACZA 16/7-SEGMENTOWEGO (PATTERN CONVERSION FOR 16/7-SEGMENT DISPLAY)	FUN 84 TDSP

SYMBOL DRABINKOWY



Md : Wybór trybu pracy
S : Adres początkowy konwertowanych znaków
Ns : Numer początkowego znaku w S
Nl : Liczba kolejnych znaków
D : Adres początkowy zapisu przekonwertowanego układu pól
Argument S może łączyć się z V, Z, P0~P9 P9 w celu zastosowania adresowania pośredniego

Zakres / Operand	HR	OR	ROR	DR	K	XR
	R0 □ R3839	R3904 □ R3967	R5000 □ R8071	D0 □ D4095	16/32 bity	V、Z P0~P9
Md					0~1	
S	○	○	○	○	○	○
Ns	○	○	○	○	○	
Nl	○	○	○	○	○	
D	○	○	○*	○		

- Instrukcja ta wykorzystywana jest w aplikacjach stosujących moduły FBs-7SG1/FBs-7SG2. Konwertuje ona źródłowe ciągi znaków alfanumerycznych, na odpowiadające im układy pól świetlnych, wyświetlanych przez wyświetlacze 16-segmentowe lub do generacji układów pól, odpowiadających wartością interpretowanym w kodzie BCD (z wiodącymi zerami) dla wyświetlaczy 7-segmentowych (gdy tryb wyświetlania ustawiony jest na bez dekodowania).

- Jeżeli „EN” = 1, „OFF” = 0, „ON” = 0 oraz Md = 0, instrukcja zrealizuje konwersję na układ pól, gdzie S jest początkowym adresem konwertowanego ciągu znaków, Ns jest wskaźnikiem na pierwszy znak w S, NI jest liczbą kolejnych konwertowanych znaków, a D jest początkowym adresem zapisu wyników konwersji.

  Bajt 0 adresu S jest 1-wszym konwertowanym znakiem, bajt 1 adresu S jest 2-gim konwertowanym znakiem, itd.

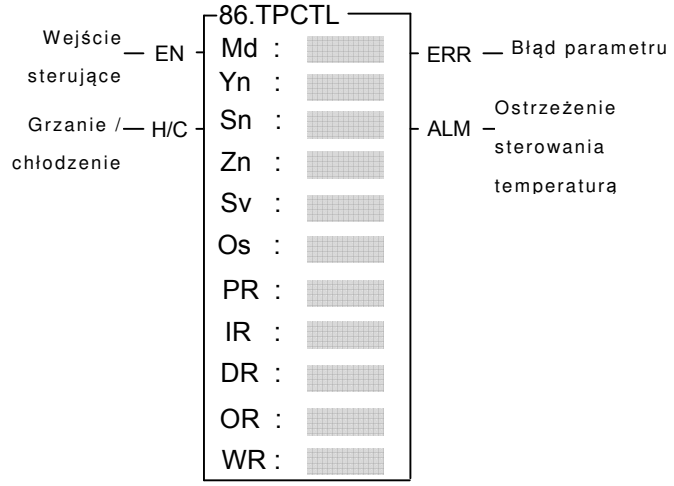
  Ns jest wskaźnikiem określającym, gdzie znajduje się pierwszy konwertowany znak.

  Po wykonaniu funkcji, każdy 8-bitowy znak źródłowy, zostanie przekonwertowany na odpowiadający mu układ pól wyświetlacza 16-segmentowego.

- Jeżeli „OFF” = 1 i Md = 0, wszystkie pola wyświetlanego układu będą wyłączone. Jeżeli Md = 1, wszystkie kody BCD zostaną zastąpione pustym znakiem (0F).
- Jeżeli „ON” = 1 i Md = 0, wszystkie pola wyświetlanego układu będą włączone. Jeżeli Md = 1, wszystkie kody BCD zostaną zastąpione znakiem „8” (włączone wszystkie pola świetlne).
- Więcej szczegółów w Rozdziale 16 „Moduł wyświetlacza FBs-7SG”.

FUN 86 TPCTL	INSTRUKCJA STEROWANIA TEMPERATURĄ PID	FUN 86 TPCTL

SYMBOL DRABINKOWY



Zakres / Argum.	Y	HR	ROR	DR	K
	Y0 | Y255	R0 | R3839	R5000 | R8071	D0 | D3999	
Md					0～1
Yn	○				
Sn					0～31
Zn					1～32
Sv		○	○*	○	
Os		○	○*	○	
PR		○	○*	○	
IR		○	○*	○	
DR		○	○*	○	
OR		○	○*	○	
WR		○	○*	○	

Md: Metoda obliczania regulatora PID
=0, Zmodyfikowana metoda minimalnego przesterowania
=1, Uniwersalna metoda PID

Yn: Początkowy adres wyjść, kontrolujących temperaturę metodą PWM; Zn kolejnych adresów.

Sn: Startowy punkt pomiaru temperatury do kontroli regulatorem PID; Sn = 0～31.

Zn: Liczba kolejnych punktów temperatury do kontroli; 1≤Sn+Zn≤32

Sv: Adres początkowy zadanych wartości temperatur: Zn kolejnych adresów.

Os: Adres początkowy rejestrów określających offset strefy regulacji; Zn kolejnych adresów. .

PR: Adres początkowy rejestrów współczynników proporcjonalnych regulatora (Kc): Zn kolejnych adresów.

IR: Adres początkowy rejestrów stałych całkowania regulatora (Ti); Zn kolejnych adresów.

DR: Adres początkowy rejestrów stałych różniczkowania regulatora (Td); Zn kolejnych adresów.

Opis funkcji i powiadomienia

- Przy zastosowaniu modułów temperaturowych i edycji tabel, uzyskujemy odczytaną wartość bieżącej temperatury, który nazwiemy zmienną procesową (PV). Po obliczeniu software'owego algorytmu regulatora PID, odpowie on na błąd regulacji, wartością wynikającą z ustawionej wartości zadanej, całki błędu i tempa zmiany zmiennej procesowej. Dzięki zastosowaniu regulacji w zamkniętej pętli, spodziewamy się stabilności procesu.
- Wyniki obliczeń regulatora PID można przekonwertować na proporcjonalną wartość wypełnienia sygnału PWM i sterować procesem grzania lub chłodzenia za pomocą wyjścia tranzystorowego, sterującego przekaźnikiem SSR. Jest to bardzo wydajne i ekonomiczne rozwiązanie.
- Wykorzystując wyjście analogowe sterownika (D/A), wynik obliczeń PID, może w sposób ciągły sterować położeniem zaworu proporcjonalnego lub SCR, w celu uzyskania dokładniejszego sterowania temperaturą.
- Cyfrowy algorytm PID jest następujący:

$$Mn = [Kc \times En] + \sum 0n[Kc \times Ti \times Ts \times En] - [Kc \times Td \times (PVn - PVn\text{-}1)/Ts]$$

gdzie,
Mn: Sygnał wyjściowy w chwili „n".
Kc: Wzmocnienie (Zakres: 1～9999；Pb=100(%) / Kc)
Ti: Stała całkowania (Zakres:0～9999, odpowiadający 0.00～99.99 powtórzeń/minutę)
Td: Stała różniczkowania (Zakres:0～9999, odpowiadający 0.00～99.99 minut)

FUN 86 TPCTL	INSTRUKCJA STEROWANIA TEMPERATURĄ PID	FUN 86 TPCTL

PVn : Wartość zmiennej procesowej w chwili „n” (bieżąca wartość)
PVn_1: Wartość zmiennej procesowej w chwili „n-1” (poprzednia wartość)
En: Uchyb w chwili „n” ; E= SP – PVn
Ts: Okres obliczania algorytmu regulatora PID (Dopuszczalne wartości: 10, 20, 40, 80,160, 320; jednostka 0.1sek.)

Instrukcja dopasowywania parametrów regulatora PID

- Im większe współczynnika proporcjonalnego (Kc) tym większy wpływ członu proporcjonalnego na odpowiedź regulatora. Odpowiedź regulatora może być łagodna lub szybka. Jednakże, jeżeli wzmocnienie jest zbyt duże, może to spowodować wystąpienie oscylacji regulowanej temperatury. Należy zatem ustawić możliwe duże „Kc” (ale nie w stopniu powodującym oscylacje), które będzie mogło wzmocnić reakcję regulatora i zmniejszyć uchyb ustalony.
- Człon całkujący może być wykorzystany do wyeliminowania uchybu ustalonego. Im wyższa wartość stałej całkowania Ti, tym większy wpływ członu całkującego na odpowiedź regulatora. W przypadku pojawienia się uchybu ustalonego, należy zwiększyć „Ti”, w celu jego zmniejszenia.
  Jeżeli „Ti” = 0, człon całkujący nie ma wpływu na odpowiedź regulatora.
  Na przykład, jeżeli czas całkowania wynosi 6 minut, Ti=100/6=17; jeżeli czas całkowania wynosi 5 minut, Ti=100/5=20.
- Człon różniczkujący może być wykorzystany aby uczynić proces płynnym i bez przesterowań. Im wyższa wartość stałej różniczkowania Td, tym większy wpływ członu różniczkującego na odpowiedź regulatora. W przypadku zbyt dużego przesterowania, należy zwiększyć „Td”, w celu jego zmniejszenia.
  Jeżeli „Td” = 0, człon różniczkujący nie ma wpływu na odpowiedź regulatora.
  Na przykład, jeżeli czas różniczkowania wynosi 1 minutę, Td = 100; jeżeli czas różniczkowania wynosi 2 minuty, Td = 200.
- Prawidłowe dopasowanie parametrów regulatora PID, umożliwia uzyskanie doskonałych wyników sterowania temperaturą.
- Domyślny okres obliczania algorytmu regulatora PID wynosi 4 sekundy (Ts=40)
- Domyślna wartość współczynnika proporcjonalnego (Kc) wynosi 110, gdzie Pb=1000/110×0.1%≒0.91%; pełny zakres systemu wynosi 1638°, co oznacza1638×0.91≒14.8° dla wprowadzenie proporcjonalnej kontroli zakresu.
- Domyślna stała całkowania wynosi 17, co oznacza, że czas zdwojenia wynosi 6 minut (Ti=100/6=17).
- Domyślna stała różniczkowania wynosi 50, co oznacza, że czas wyprzedzenia wynosi 0.5 minuty(Td=50).
- Po zmianie okresu obliczania algorytmu regulatora PID, koniecznym może się okazać ponowne dopasowanie parametrów Kc, Ti, Td.

Opis funkcji

- FUN86 włączy się po odczytaniu wszystkich kanałów temperaturowych.
- Jeżeli „EN” = 1, w zależności od stanu wejścia H/C, funkcja będzie wykonywała regulację temperatury metodą PID, w trybie grzania (H/C=1) lub chłodzeniem (H/C=0). Aktualne wartości temperatur mierzone są modułami temperaturowymi. Wartości zadane temperatur w rejestrach rozpoczynających się od adresu Sv. Po obliczeniu software’owego algorytmu regulatora PID, odpowie on na błąd regulacji, wartością wynikającą z ustawionej wartości zadanej, całki błędu i tempa zmiany zmiennej procesowej. Odpowiedź ta jest konwertowana na proporcjonalną wartość wypełnienia sygnału PWM, którym odbywa się sterowanie procesem grzania lub chłodzenia za pomocą wyjścia tranzystorowego, sterującego przekaźnikiem SSR. Jest to bardzo wydajne i ekonomiczne rozwiązanie. Równocześnie odpowiedź regulatora zapisywana jest w rejestrach OR, po wcześniejszym przekonwertowaniu na zakres wyjścia analogowego (0~16383). W ten sposób, wykorzystując wyjście analogowe sterownika (D/A), wynik obliczeń PID, może w sposób ciągły sterować położeniem zaworu proporcjonalnego lub SCR, w celu uzyskania dokładniejszego sterowania temperaturą.
- Jeżeli ustawienia Sn i Zn (gdzie: $0 \le Sn \le 31$, $1 \le Zn \le 32$ oraz $1 \le Sn + Zn \le 32$) będą nieprawidłowe, instrukcja ta nie zostanie wykonana a wyjście „ERR” będzie w stanie wysokim.

Instrukcja ta porównuje aktualną wartość temperatury z jej wartością zadaną celu sprawdzenia, czy aktualna temperatura znajduje się w strefie regulacji (dopuszczalnym zakresie odchylenia, określanym przez wartość w rejestrze oznaczonym jako Os - offset). Jeżeli wartość bieżąca temperatury dla danego punktu jej pomiaru będzie w strefie, to status bitu „wewnątrz strefy” dla tego punktu zostanie ustawiony na 1; w przypadku jej wyjścia poza strefę, status bitu zostanie ustawiony na 0 a wyjście „ALM” będzie w stanie wysokim.

FUN 86 TPCTL	PID TEMPERATURE CONTROL INSTRUCTION	FUN 86 TPCTL

- W międzyczasie, funkcja sprawdza także, czy nie wystawić ostrzeżenia o wysokiej temperaturze (rejestrem do zapisu nastawy poziomu ostrzeżenia o wysokiej temperaturze jest R4008). Jeżeli w kolejnych dziesięciu odczytach aktualnej wartości, zmierzona temperatury będzie równa lub większa od nastawy ostrzeżenia o wysokiej temperaturze, bit ostrzeżenia i znacznik „ALM" zostaną ustawione na 1. Można dzięki temu uniknąć problemu wynikającego z utracenia kontroli nad temperaturą, w przypadku zwarcia SSR lub obwodzie grzewczym.
- Instrukcja ta może także wykryć problem z grzaniem wynikający z rozwarcia SSR, obwodu grzewczego lub ze zużytej elementu grzewczego. Kiedy wyjście kontrolujące temperaturę przełączy się na dużą moc (której poziom ustawiany jest w rejestrze R4006) permanentnie w danym okresie czasu (ustawianym w rejestrze R4007), a aktualna temperatura będzie wykraczać poza wymagany zakres, to bit ostrzeżenia i znacznik „ALM" zostaną ustawione na 1.
- WR: Początkowy rejestr roboczy dla tej instrukcji. Zawiera 9 kolejnych rejestrów i nie mogą one być ponownie użyte w programie.

  Zawartość dwóch pierwszych rejestrów WR+0 i WR+1 wskazuje na to, czy aktualna temperatura znajduje się wewnątrz czy poza strefą odchylenia (określaną przez wartość w rejestrach Os). Jeżeli wartość będzie w strefie, stan bitu dla danego punktu pomiaru temperatury zostanie ustawiony na 1; w przypadku wyjścia poza strefę, status bitu zostanie ustawiony na 0 i aktywowany zostanie znacznik „ALM".

  Funkcje bitów w WR+0 są następujące:

  Bit0=1 wskazuje na to, że temperatura w punkcie Sn+0 jest w strefie.

  Bit15=1 wskazuje na to, że temperatura w punkcie Sn+15 jest w strefie.

  Funkcje bitów w WR+1 są następujące:

  Bit0=1 wskazuje na to, że temperatura w punkcie Sn+16 jest w strefie

  Bit15=1 wskazuje na to, że temperatura w punkcie Sn+31 jest w strefie.

  Rejestry WR*+2 i WR+3 są rejestrami zawierającymi bity ostrzeżeń, a ich zawartość wskazuje na ewentualne ostrzeżenie o wysokiej temperaturze lub rozwarcie obwodu grzewczego.

  Funkcje bitów w WR+2 są następujące:

  Bit0=1 wskazuje na obecność ostrzeżenia o wysokiej temperaturze lub rozwarcia obwodu grzewczego w punkcie Sn+0

  Bit15=1 wskazuje na obecność ostrzeżenia o wysokiej temperaturze lub rozwarcia obwodu grzewczego w punkcie Sn+15.

  Funkcje bitów w WR+3 są następujące:

  Bit0=1 wskazuje na obecność ostrzeżenia o wysokiej temperaturze lub rozwarcia obwodu grzewczego w punkcie Sn+16

  Bit15=1 wskazuje na obecność ostrzeżenia o wysokiej temperaturze lub rozwarcia obwodu grzewczego w punkcie Sn+31

  Instrukcja ta wykorzystuje również rejestry WR+4 ～ WR+8.
- Pojedyncza funkcja może realizować sterowanie temperaturą w trybie grzania lub chłodzenia. Sterowanie chłodzeniem i grzaniem, wymaga zastosowania oddzielnych instrukcji.

Specjalne rejestry związane z FUN86

- R4005 : Młodszy bajt określa okres obliczania algorytmu regulatora PID
  =0, obliczanie PID co 1 sekundę.
  =1, obliczanie PID co każde 2 sekundy.
  =2, obliczanie PID co każde 4 sekundy. (wartość domyślna)
  =3, obliczanie PID co każde 8 sekund.
  =4, obliczanie PID co każde 16 sekund.
  ≥5, obliczanie PID co każde 32 sekundy.
  : Starszy bajt określa okres sygnału PWM
  =0 , okres sygnału PWM wynosi 1 sekundę.
  =1 , okres sygnału PWM wynosi 2 sekundy. (wartość domyślna)
  =2 , okres sygnału PWM wynosi 4 sekundy.
  =3 , okres sygnału PWM wynosi 8 sekund.
  =4 , okres sygnału PWM wynosi 16 sekund.
  ≥5 , okres sygnału PWM wynosi 32 sekundy.

FUN 86 TPCTL	PID TEMPERATURE CONTROL INSTRUCTION	FUN 86 TPCTL

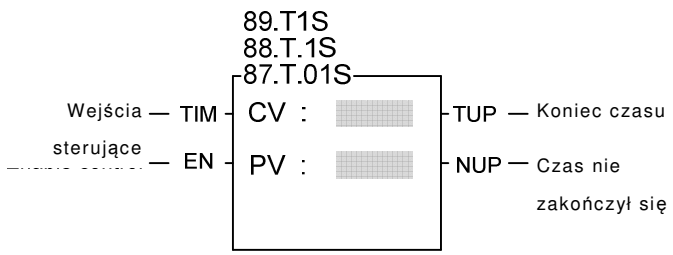
Uwaga 1: Przy zmianie wartości R4005, stan na wejściu sterującym „EN" FUN86 musi być ustawiony na 0. Dopiero po ponownym załączeniu „EN" = 1, funkcja, przy realizacji obliczeń PID, będzie bazować na nowych ustawieniach.

Uwaga 2: Im krótszy czas cyklu tym bardziej równomierne ogrzewanie. Jednakże, uchyb wywołany czasem skanu PLC będzie również większy. W celu osiągnięcia najlepszego sterowania, można bazować na czasie skanu, w celu określenia przedziału pomiędzy rozwiązaniami algorytmu regulatora PID oraz okresu sygnału PWM.

- R4006: Nastawa detekcji dużej mocy na wyjściu przy rozwarciu SSR, obwodu grzewczego lub zużycia elementu grzewczego. Jednostką jest %, a zakres wynosi 80 ~ 100(%); wartość domyślna to 90(%).
- R4007: Czas detekcji występowania dużej mocy na wyjściu przy rozwarciu SSR, obwodu grzewczego lub zużycia elementu grzewczego. Jednostką jest sekunda, a zakres wynosi 60 ~ 65535 (sekund); wartość domyślna to 600 (sekund).
- R4008: Nastawa ostrzeżenia o wysokiej temperaturze przy wystąpieniu zwarcia w SSR lub obwodzie grzewczym. Jednostką jest 0.1 stopnia a zakres wynosi 100 ~ 65535; wartość domyślna to 3500 (jednostka w 0.1℃).
- R4012: Stany bitów R4012 informują o potrzebie sterowania temperatury.
  Bit0=1 oznacza, że pierwszy punkt wymaga sterowania temperatury
  Bit1=1 oznacza, że drugi punkt wymaga sterowania temperatury.
  □
  □
  Bit15=1 oznacza, że szesnasty punkt wymaga sterowania temperatury.
  (Domyślną wartością dla R4012 jest FFFFH – kontrola wszystkich punktów temperatury jest załączona)
- R4013: Stany bitów R4013 informują o potrzebie sterowania temperatury.
  Bit0=1 oznacza, że siedemnasty punkt wymaga sterowania temperatury.
  Bit1=1 oznacza, że osiemnasty punkt wymaga sterowania temperatury.
  □
  □
  Bit15=1 oznacza, że trzydziesty drugi punkt wymaga sterowania temperatury.
  (Domyślną wartością dla R4013 jest FFFFH – kontrola wszystkich punktów temperatury jest załączona)
- Jeżeli „EN" = 1, a bit określający załączenie sterowania dla tego punktu ustawiony jest na 1 (odpowiedni bit w R4012 lub R4013 będzie ustawiony na 1), funkcja FUN86 przeprowadzi sterowanie regulatorem PID i wyśle sygnał wyjściowy jako wynik obliczeń.
- Jeżeli „EN" = 1, a bit określający załączenie sterowania dla tego punktu ustawiony jest na 0 (odpowiedni bit w R4012 lub R4013 będzie ustawiony na 0), funkcja FUN86 nie przeprowadzi sterowania regulatorem PID i nie wyśle żadnego sygnału wyjściowego.
- Program drabinkowy może kontrolować stan odpowiednich bitów w R4012 lub 4013 w celu przekazania funkcji FUN86 informacji, czy dany punkt pomiaru temperatury ma być kontrolowany czy nie, zatem możliwe jest zastosowanie w programie tylko jednej funkcji FUN86.

FUN87 T.01S FUN88 T.1S FUN89 T1S	TIMER KUMULUJĄCY (ACCUMULATIVE TIMER)	FUN87 T.01S FUN88 T.1S FUN89 T1S

SYMBOL DRABINKOWY



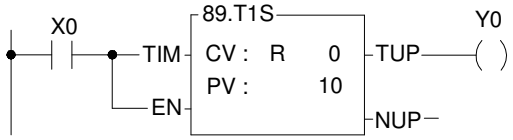
CV : Rejestr do zapisu odmierzonego czasu (aktualna wartość)

PV : Wartość zadana timera

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C199	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	0 □ 32767
CV		○	○	○	○	○	○	○	○	○*	○*	○	
PV	○	○	○	○	○	○	○	○	○	○	○	○	○

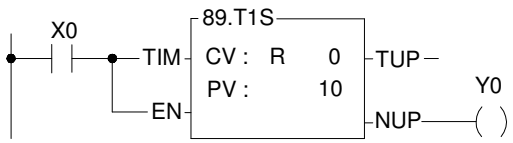
- Działanie tej funkcji, jest takie samo jak normalnych timerów (T0~T255), z tą różnicą, że normalne timery, posiadają jedynie wejście załączenia zliczania czasu EN – kiedy na wejściu jest 1, timer rozpoczyna odliczanie czasu; kiedy na wejściu jest 0, timer jest zerowany. Przy każdej zmianie stanu na wejściu, rozpoczyna się ponowne odliczanie czasu, co uniemożliwia kumulowanie sumarycznego czasu załączenia wejścia zliczania. Zliczanie czasu w normalnych timerach, możliwe jest tylko wtedy, gdy wejście sterujące EN=1. W przypadku tej funkcji, zliczanie również jest wykonywane, gdy „EN" = 1. Gdy na wejściu TIM jest stan 1, funkcja działa identycznie jak normalne timery, tj wartość odmierzonego czasu zeruje się po podaniu stanu niskiego na wejście EN. Natomiast kiedy na wejściu TIM znajduje się stan niski (TIM=0), w chwili pojawienia się na wejściu EN stanu niskiego, wartość bieżąca odmierzonego czasu nie zeruje się, a po ponownym podaniu stanu wysokiego, zliczanie jest kontynuowane od poprzedniej wartości. Dodatkowo, funkcja ta posiada dwa wyjścia: TUP (po doliczeniu do wartości zadanej ma stan 1, w przeciwnym przypadku 0) oraz NUP (po doliczeniu do wartości zadanej ma stan 0, w przeciwnym przypadku 1). Użytkownik może wykorzystać różne kombinacje tych we / wy w celu zbudowania timerów o rożnych funkcjach. Na przykład:

- Timer z opóźnionym załączaniem wyjścia przy włączeniu:



- Wyjście tego timera (Y0), jest normalnie wyłączone. W przypadku, gdy wejście sterujące (X0) zostanie załączone w stan wysoki, to wyjście Y0 zostanie załączone po 10 sekundach.

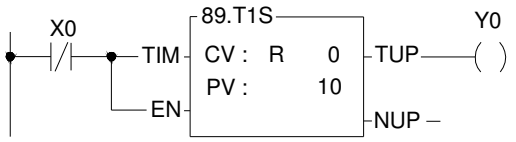
- Timer z opóźnionym wyłączeniem wyjścia przy włączeniu:



- Wyjście tego timera (Y0), jest normalnie załączone. Po przełączeniu w stan wysoki wejścia sterującego X0, wyjście Y0 zostanie wyłączone po 10 sekundach.

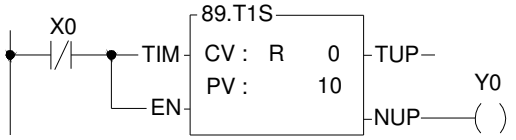
FUN87 T.01S FUN88 T.1S FUN89 T1S	TIMER KUMULUJĄCY	FUN87 T.01S FUN88 T.1S FUN89 T1S

- Timer z opóźnionym załączaniem wyjścia przy wyłączeniu:



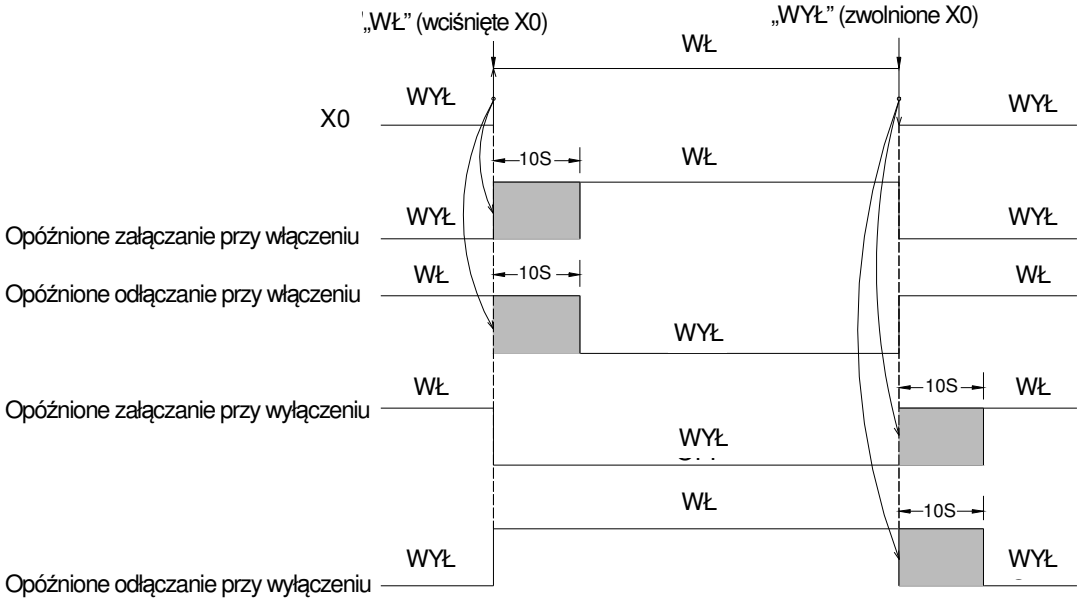
- Gdy wejście sterujące X0 jest w stanie wysokim – wyjście timera Y0 jest wyłączone. Po przejściu X0 w stan niski, wyjście Y0 zostanie wystawione w stan wysoki po 10 sekundach.

- Timer z opóźnionym wyłączeniem wyjścia przy wyłączeniu:



- Gdy wejście sterujące X0 jest w stanie wysokim – wyjście timera Y0 jest włączone. Po przejściu X0 w stan niski, wyjście Y0 zostanie wyłączone po 10 sekundach.

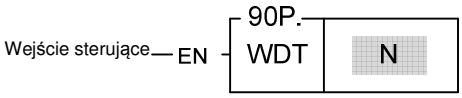
- Poniższe schematy przedstawiają relację we / wy dla 4 rodzajów timerów.



Funkcje timera WDT

FUN 90 P WDT	USTAW CZAS WATCHDOG TIMERA (WATCHDOG TIMER CONTROL)	FUN 90 P WDT

SYMBOL DRABINKOWY



N : Czas watchdog. Zakresem N jest 5~120 , jednostka to 10ms (czyli ustawialnym zakresem czasu jest 50ms~1.2 sec)

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), czas watchdog ustawiony zostanie na Nx10ms. Jeżeli czas skanu przekroczy ustawioną wartość, PLC wyłączy się, a wykonywanie programu zostanie zatrzymane.

- Głównym przeznaczeniem funkcjonalności WDT jest zapewnienie bezpieczeństwa systemu ze strony aplikacji. Na przykład, jeżeli CPU sterownika PLC zostanie nagle uszkodzone i nie będzie możliwości realizacji programu ani odświeżenia We / Wy, to po upływie czasu WDT, WDT automatycznie wyłączy wszystkie We / Wy w celu zapewnienia bezpieczeństwa. W niektórych aplikacjach, jeżeli czas skanu jest zbyt długi, może to spowodować problemy natury bezpieczeństwa lub problemy związane z niespełnianiem wymagań kontroli. Funkcja ta jest wykorzystywana do ustalenia granicznej wartości czasu skanu.

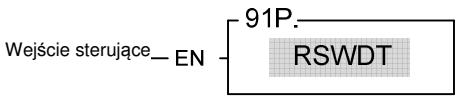
- Po ustawieniu wartości czasu WDT, czas ten jest podtrzymywany i nie ma potrzeby ustawiania go ponownie w każdym skanie programu. Dlatego też, w praktyce, instrukcja ta powinna wykorzystywać instrukcję P(być wywoływana stanem narastającym na wejściu EN).

- Domyślną wartością czasu WDT jest 0.25 sek.

- Zasada działania WDT, szerzej przedstawiona jest w opisie działania funkcji RSWDT(FUN 91).

FUN 91 P RSWDT	RESET TIMERA WATCHDOG (RESET WATCHDOG TIMER)	FUN 91 P RSWDT

SYMBOL DRABINKOWY



Funkcja ta nie posiada argumentu.

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi reset timera WDT (WDT rozpocznie zliczanie czasu od 0).

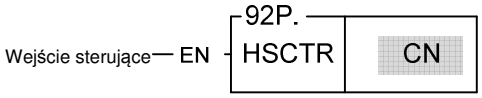
- Przeznaczenie timera watchdog (WDT), zostało wcześniej opisane, w rozdziale opisującym działanie funkcji WDT (FUN90).
  Zasada działania timera watchdog jest następująca:

  Timer watchdog jest zazwyczaj implementowany jako timer sprzętowy z szybkim zerowaniem (nie może być to timer software'owy ponieważ w przypadku usterki CPU, taki timer stałby się bezużyteczny). Timer z „szybkim zerowaniem" oznacza, że po pojedynczym wzbudzeniu timera, zliczona wartość czasu zostanie wyzerowana i rozpocznie się zliczanie czasu od nowa. Jeżeli licznik WDT rozpoczął zliczanie i nigdy potem nie byłby zerowany, wartość odmierzonego czasu w WDT będzie narastać, do momentu osiągnięcia ustawionego maksymalnego czasu skanu (N). W tej chwili, błąd timer WDT zostanie wywołany, a PLC wyłączy się. Natomiast, jeżeli timer WDT będzie zerowany przed upływem ustawionego maksymalnego czasu skanu, błąd timera WTD, nigdy nie zostanie wywołany. Takie działanie timera watchdog, jest wykorzystywane przez PLC, w celu zapewnienia bezpieczeństwa systemu. Po zakończeniu każdego skanu programu i odświeżeniu stanów We / Wy, timer WDT jest zerowany. Zatem, jeżeli system będzie pracował normalnie, a czas skanu nie przekroczy wartości N, błąd timera WDT nigdy nie zostanie wywołany. Jednakże, w przypadku uszkodzenia CPU, co uniemożliwia wyzerowanie czasu timera WDT, lub w przypadku, kiedy czas skanu programu jest zbyt długi, przez co timer WDT nie zostanie wyzerowany przed upływem maksymalnego czasu skanu (N), błąd timera WDT zostanie wywołany, a PLC wyłączy się.

- W pewnych aplikacjach, po ustawienia maksymalnego czasu skanu (FUN90), zdarza się, że czas skanu programu, chwilowo przekracza wartość maksymalną (np. gdy w danym skanie wywołano podprocedurę o długim czasie wykonania). W takich sytuacjach, świadomie wydłużamy czas skanu, nie chcemy przy tym, aby błąd timera WTD był wywoływany. Możemy wtedy użyć tej funkcji w celu wyzerowania timera WDT i uniknięcia wywołania tego błędu. Takie jest główne zastosowanie tej funkcji.

FUN 92 D P HSCTR	ODCZYT AKTUALNEJ WARTOŚCI SZYBKIEGO LICZNIKA SPRZĘTOWEGO (HARDWARE HIGH SPEED COUNTER CURRENT VALUE (CV) ACCESS)	FUN 92 D P HSCTR

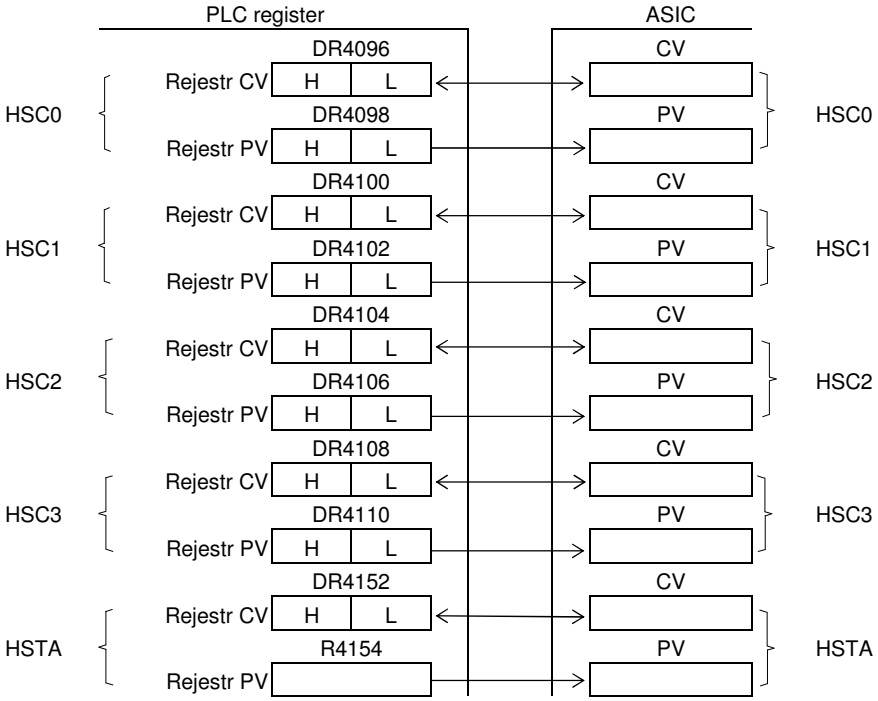
SYMBOL DRABINKOWY



CN : Numer szybkiego licznika sprzętowego

0: HSC0 lub HST0
1: HSC1 lub HST1
2: HSC2 lub HST2
3: HSC3 lub HST3
4: HSTA

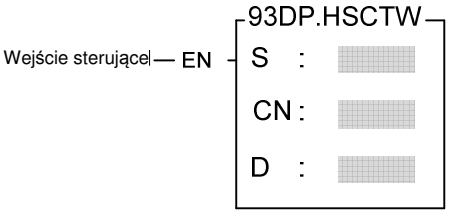
- Liczniki HSC0～HSC3 sterowników PLC serii FBs, stanowią 4 zestawy 32-bitowych szybkich liczników, o różnych trybach zliczania, takich jak impulsy góra/dół, puls-kierunek, zliczanie AB-fazowe. Wszystkie 4 szybkie liczniki wbudowane są w sprzęt ASIC i mogą realizować: zliczanie, porównywanie oraz wywoływanie przerwań sprzętowych, niezależnie i bez udziału CPU. W przeciwieństwie do szybkich liczników software'owych HSC4～HSC7, które wykorzystują metodę przerwań sprzętowych, do wywoływania zliczania przez CPU, co skutkuje tym, że w przypadku zbyt wielu sygnałów zliczania lub ich zbyt dużej częstotliwości, czas skanu sterownika dramatycznie wzrasta. Ponieważ aktualne wartości (CV) liczników HSC0 ～HSC3 są wbudowane w wewnętrzne obwody ASIC, program sterujący użytkownika (program drabinkowy) nie może pozyskiwać ich bezpośrednio z ASIC. Dlatego też, program musi zastosować tą funkcję, w celu odczytania wartości bieżącej (CV) ze sprzętowego HSC i zapisania jej w rejestrze, do którego dostęp możliwy będzie z poziomu programu. Poniżej przedstawiony jest schemat wartości CV i PV w ASIC wraz z odpowiadającymi im rejestrami PLC, dla HSC0~HSC3.



- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0→1 (instrukcja P), funkcja odczyta z ASIC wartość bieżącą (CV) z HSC o numerze określonym przez CN i zapisze ją w odpowiadającym temu HSC rejestrze CV (np. wartość bieżąca (CV) w HSC0 zostanie odczytana i zapisana w DR4096, a wartość bieżąca (CV) w HSC1 zostanie odczytana i zapisana w DR4100).
- Pomimo, że wartość zadana (PV) w ASIC, posiada odpowiadający jej rejestr PV w CPU, nie jest konieczne jej odczytywanie tą funkcją, ponieważ wartość PV w ASIC pochodzi z rejestru PV w CPU.
- HSTA jest timerem o podstawie czasu 0.1ms. Zawartość CV tego timera odzwierciedla odliczony czas w jednostkach 0.1ms.
- Szczegóły dotyczące zastosowań funkcji, zostały opisane w rozdziale 10 „Szybkie liczniki i szybkie timery sterowników serii FBs.

FUN 93 D P HSCTW	ZAPIS WATOŚCI BIEŻĄCEJ I ZADANEJ SZYBKIEGO LICZNIKA SPRZĘTOWEGO	FUN 93 D P HSCTW

SYMBOL DRABINKOWY

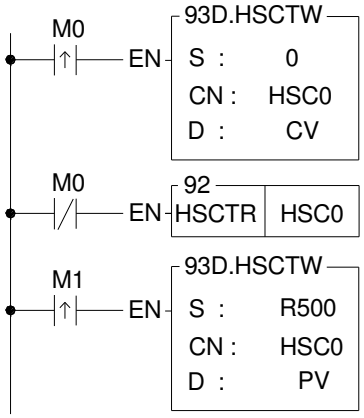


S : Dane źródłowe do zapisu

CN : Numer szybkiego licznika sprzętowego do zapisu
- 0: HSC0 lub HST1
- 1: HSC1 lub HST2
- 2: HSC2 lub HST3
- 3: HSC3 lub HST4
- 4: HSTA

D : Cel zapisu (gdzie: 0 oznacza wartość bieżącą (CV), 1 oznacza wartość zadaną (PV))

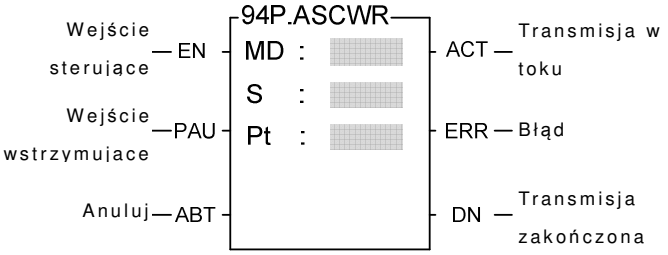
- Relacje pomiędzy wartościami bieżącymi (CV) i zadanymi (PV) liczników HSC0～HSC3 i HSTA w ASIC, oraz odpowiadającymi im adresami rejestrów CV i PV w CPU, zostały opisane w rozdziale omawiającym FUN92.
- Jeżeli wejście serujące „EN”=1 lub zmieni się z 0→1 (instrukcja P), wartość S zostanie zapisana w odpowiednim rejestrze CV lub PV (w zależności od ustawienia w D) HSC w ASIC.
- W przypadku większości aplikacji, konieczne jest ustawianie wartość zadanej (PV) szybkiego licznika sprzętowego. Kiedy zliczona przez licznik sprzętowy wartość bieżąca (CV) zrówna się z jego wartością zadaną (PV), licznik ten natychmiastowo zgłosi przerwanie sprzętowe. Dzięki zastosowaniu procedury obsługi tego przerwania sprzętowego, możliwa jest implementacja różnego rodzaju aplikacji precyzyjnego zliczania lub pozycjonowania.
- Po zaniku zasilania PLC, wartości bieżące (CV) liczników HSC0～HSC3 w ASIC, zostają automatycznie odczytane i zapisane w rejestrach CV HSC0～HSC3 (rejestry te są podtrzymywane) w CPU. Przy ponownym podaniu zasilania, zapisane w chwili zaniku zasilania wartości CV, zostaną przywrócone do ASIC. Natomiast, jeżeli Twoja aplikacja wymaga, aby po uruchomieniu sterownika, wartość bieżąca była zerowana lub zliczanie rozpoczynało się od konkretnej wartości – musisz użyć tej funkcji w celu ustawienia wartości bieżącej (CV) w ASIC.
- Ustawienie niezerowej wartości w rejestrze PV szybkiego timera HSTA, powoduje cykliczne zgłaszanie przerwania sprzętowego HSTAI, co każde PVx0.1ms.
- Szczegóły dotyczące zastosowań funkcji, zostały opisane w rozdziale 10 „Szybkie liczniki i szybkie timery sterowników serii FBs.



- W programie po lewej, gdy M0 zmieni się z 0→1, nastąpi wyzerowanie aktualnej wartości HSC0 i zapisanie w ASIC za pomocą FUN93.
- Jeżeli M0 = 0, program odczyta wartość bieżącą licznika HSC0.
- Jeżeli M1 zmieni się z 0→1, zawartość DR500 zostanie przesunięta do DR4098, a ustawiona tam wartość zapisana w ASIC za pomocą FUN93.
- Kiedy wartość bieżąca będzie równa DR500, zostanie zgłoszone przerwanie sprzętowe HSC0I.

FUN 94 P ASCWR	WYJŚCIE ASCII (ASCII WRITE)	FUN 94 P ASCWR

SYMBOL DRABINKOWY



MD: Tryb wyjścia
=0, wyjście na port komunikacyjny port1.
(inne, dla przyszłych zastosowań)

S : Startowy adres rejestru pliku danych.

Pt : Początkowy adres rejestrów roboczych dla tej funkcji. Zabiera 8 rejestrów i nie może być ponownie użyty w innej części programu.

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3967 □ R4167	R5000 □ R8071	D0 □ D4095	0 □ 1
MD													○
S	○	○	○	○	○	○	○	○	○	○	○	○	
Pt		○	○	○	○	○	○		○	○*	○*	○	

- Jeżeli MD=0, a wejście sterujące „EN” zmieni się z 0→1, nastąpi transmisja danych ASCII na port1 rozpoczynając od rejestru S, a kończąc w chwili napotkania znaku końca pliku.

- Dane w rejestrze S mogą być edytowane za pomocą oprogramowania PROLADDER lub WinProladder (szczegóły w rozdziale 14 „Zastosowanie funkcji ASCII”.). W razie potrzeby użytkownik może też bezpośrednio edytować plik ASCII poprzez zmianę wartości w rejestrach danych. Jednakże, format edytowanych danych musi być zgodny z formatem pliku ASCII (szczegóły w rozdziale 14). W innym przypadku instrukcja zatrzyma transmisję i ustawi wyjście błędu „ERR” na 1. Po prawidłowej i udanej transmisji pliku, stan na wyjściu „DN” zostanie ustawiony na 1.

- Wejście sterujące tej funkcji wzbudzane jest narastającym zboczem. Jeżeli „EN” zmieni się z 0→1, rozpocznie się wykonywanie funkcji, do momentu zakończenia transmisji całego pliku. Podczas transmisji znacznik „ACT” ustawiony będzie na 1. Znacznik ten zmieni się na 0 jedynie w przypadku podania stanu wysokiego na wejście wstrzymania „PAU”, błędu lub anulowania operacji wejściem „ABT”.

- Funkcja ta może występować w programie wielokrotnie, ale w danej chwili czasu, transmisja danych może być wykonywana tylko przez jedną z nich. Rolą programisty jest zapewnienie poprawnej sekwencji wykonywania funkcji.

- Jeżeli podczas wykonywania funkcji, wejście „PAU” = 1, nastąpi wstrzymanie transmisji danych. Transmisja będzie kontynuowana po zmianie stanu wejścia „PAU” na 0.

- Jeżeli podczas wykonywania funkcji, wejście „ABT” = 1, nastąpi anulowanie transmisji pliku. Możliwe jest wówczas wykonywania innych, występujących w programie, funkcji ASCWR.

- Więcej szczegółów w rozdziale 14 „Zastosowanie funkcji wyjściowej ASCII”.

FUN 94 ASCWR	WYJŚCIE ASCII (ASCII WRITE)	FUN 94 ASCWR

- Sygnały interfejsu:

  M1927: Znacznik ten, kontrolowany jest przez CPU i stosowany w ASCWR dla MD:0

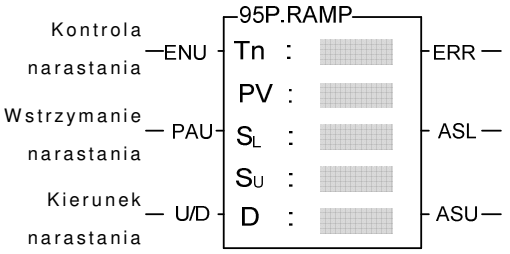
  : WŁ, oznacza, że stan sygnału RTS (podłączonego do CTS w PLC) drukarki jest niski - tzn. drukarka nie jest gotowa lub działa nieprawidłowo.

  : WYŁ, oznacza, że stan sygnału RTS jest wysoki - drukarka jest gotowa.

  Uwaga: Przy zastosowaniu znacznika M1927, we współpracy z timerem, można sprawdzić, czy drukarka działa prawidłowo.

FUN 95 P RAMP	NARASTANIE WARTOŚCI NA WYJŚCIU ANALOGOWYM (RAMP FUNCTION FOR D/A OUTPUT)	FUN 95 P RAMP

SYMBOL DRABINKOWY



Tn : Adres timera odmierzającego takty narastania [o podstawie 0,01s]
PV : Wartość zadana timera (jednostką jest 0.01 sekundy), lub wartość inkrementacji co 0,01s
S_L : Dolna wartość graniczna (dolna wartość funkcji narastającej).
S_U : Górna wartość graniczna (szczytowa wartość funkcji narastającej).
D : Rejestr przechowujący aktualną, narastającą wartość
D+1 : Rejestr roboczy
S_U, S_L może być wartością dodatnią lub ujemną

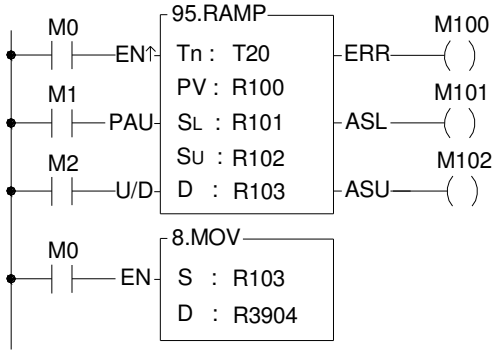
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16-bit +/- liczba
Tn					○								
PV	○	○	○	○	○	○	○	○	○	○	○	○	○
S_L	○	○	○	○	○	○	○	○	○	○	○	○	○
S_U	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○	○*	○	

Opis

- Tn musi być adresem timera o podstawie 0.01s, nie używanym w innej części programu.
- PV jest wartością zadaną timera funkcji. Jednostką jest 10 ms (0.01 sekundy).
- Jeżeli wejście „ENU” zmieni się z 0→1, na początku timer Tn zostanie wyzerowany. Następnie:
  - jeżeli wejście „U/D” = 1, wartość S_L zostanie zapisana w rejestrze D. Jeżeli M1974 = 0, wartość D będzie zwiększana o S_U–S_L / PV co każde 0.01 sek., a jeżeli M1974 = 1, wartość D będzie zwiększana o PV co każde 0.01 sek. Jeżeli wartość D osiągnie wartość S_U, wyjście „ASU” = 1.
  Jeżeli „U/D” = 0 wartość S_U zostanie zapisana w rejestrze D. Jeżeli M1974 = 0, wartość D będzie zmniejszana o S_U–S_L / PV co każde 0.01 sek., a jeżeli M1974 = 1, wartość D będzie zmniejszana o PV co każde 0.01 sek. Jeżeli wartość D osiągnie wartość S_L, wyjście „ASL” = 1.
- Kierunek narastania (U/D) wyznaczany jest w momencie, gdy wejście „ENU” zmienia się z 0→1. Po rozpoczęciu narastania wartości w rejestrze D, zmiany na wejściu U/D będą ignorowane.
- W przypadku potrzeby wstrzymania narastania wartości, należy ustawić wejście „PAU” = 1. Jeżeli „PAU” = 0, a funkcja narastająca nie została zakończona, to wykonywanie funkcji będzie kontynuowane.
- Wartość S_U musi być większa niż S_L. W przeciwnym wypadku, funkcja nie zostanie wykonana, a znacznik błędu „ERR” zostanie ustawiony na 1.
- Instrukcja ta wykorzystuje rejestr D do zapisywania wyjściowej, narastającej wartości. W przypadku, gdy aplikacja wykorzystuje wyjście analogowe do wysyłania komendy prędkości, komendy te mogą być generowaną funkcją RAMP, w celu uzyskania płynniejszego ruchu.
- Oprócz wykorzystania rejestru D do zapisu narastającej wartości funkcji, funkcja ta wykorzystuje również rejestr D+1 jako rejestr roboczy. W związku z tym, rejestr ten nie może być wykorzystywany w innej części programu.

FUN 95 P RAMP	NARASTANIE WARTOŚCI NA WYJŚCIU ANALOGOWYM (RAMP FUNCTION FOR D/A OUTPUT)	FUN 95 P RAMP

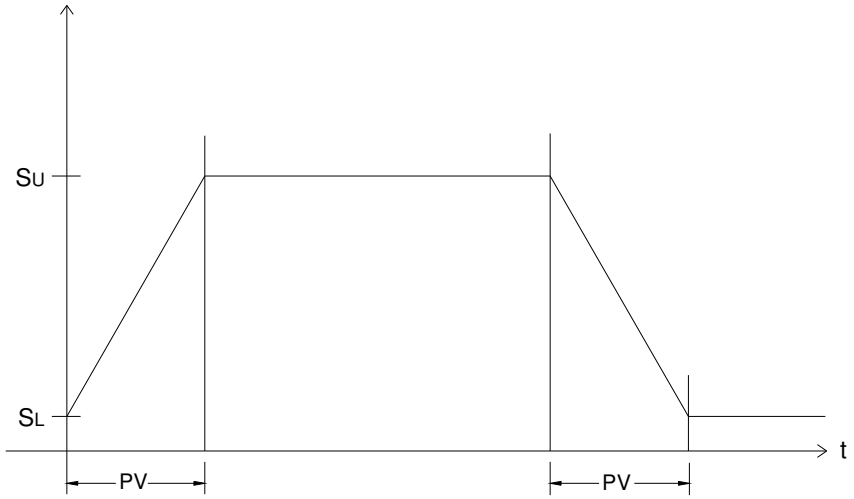
Przykładowy program



Przenoszenie narastającej wartości funkcji do rejestru wyjścia analogowego R3904.

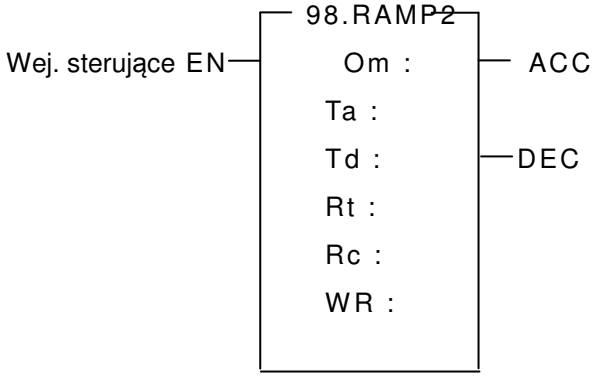
T20: Timer odmierzający takty narastania (timer z podstawą czasową 0.01 sekundy)
R100: wartość zadana timera (jednostką jest 0.01 sekundy).
R101: Dolna wartość graniczna.
R102: Górna wartość graniczna.
R103: Rejestr przechowujący aktualną, narastającą wartość.
R104: Rejestr roboczy

- Jeżeli M19740 i M0 zmieni się z 0→1, nastąpi wyzerowanie timera T20. Jeżeli M2 = 1, dolna wartość graniczna R101 zostanie zapisana w R103, a wartość wyjściowa będzie wzrastać o stałą wartość (R102-R101 / R100) co każde 0.01 sekundy i będzie zapisywana w rejestrze R103. Kiedy wartość w timerze osiągnie wartość w R100, to na wyjściu pojawi się wartość równa R102, a wyjście M102 zostanie ustawione na 1. Jeżeli M2 = 0, górna wartość graniczna zostanie zapisana w R103, a wartość wyjściowa będzie maleć o stałą wartość (R102-R101 / R100) co każde 0.01 sekundy i będzie zapisywana w rejestrze R103. Kiedy wartość w timerze osiągnie wartość w R100, to na wyjściu pojawi się wartość równa R102, a wyjście M101 zostanie ustawione na 1.
- Jeżeli M1 = 1, narastanie wartości przez funkcje zostanie wstrzymane.
- Wartość w rejestrze R102 musi być wyższa niż R101. W przeciwnym wypadku funkcja nie zostanie wykonana, a wyjście M100 zostanie ustawione na 1.



Instrukcja slow up / slow down

FUN98 RAMP2	NADĄŻAJĄCE NARASTANIE NA WYJŚCIU ANALOGOWYM (TRACKING TYPE RAMP FUNCTION FOR D/A OUTPUT)	FUN98 RAMP2



Om: Maks. wartość wyjściowa; zakres 0~65535
Ta: Czas narastania na wyjściu od 0 do maksimum; Zakres: 0~65000, jednostka ms
Td: Czas opadania na wyjściu od maksimum do 0; Zakres: 0~65000, jednostka ms
Rt: Rejestr wartości zadanej wyjścia; Zakres: 0~65535
Rc: Rejestr aktualnej wartości na wyjściu; wykorzystywany dla wyjścia analogowego.
WR: Początkowy adres rejestrów roboczych; wymaga 4 rejestrów

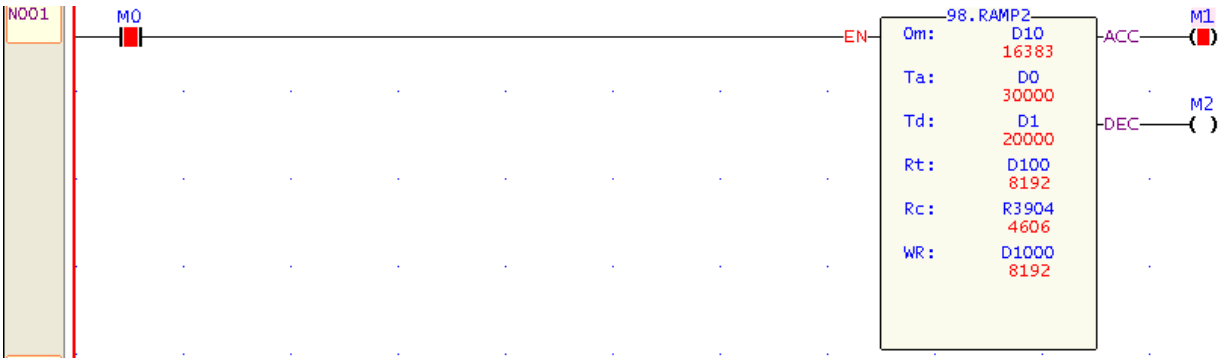
∗ Instrukcja ta może być stosowana w PLC o firmwarze V4.60 lub nowszym.

Argument \ Zakres	HR	OR	ROR	DR	K
	R0 ▯ R3839	R3904 ▯ R3967	R5000 ▯ R8071	D0 ▯ D3999	16bit
Om	○	○	○	○	0~65535
Ta	○	○	○	○	0~65000
Td	○	○	○	○	0~65000
Rt	○	○	○	○	
Rc	○	○	○	○	
WR	○	○	○*	○	

- Jeżeli „EN” = 0, to aktualna wartość na wyjściu (Rc) zostanie bezzwłocznie ustawiona na 0; stan wyjść ACC=0 i DEC=0.
- Jeżeli „EN” = 1, instrukcja ta zostanie wykonana. Na wyjściu pojawi się wartość (Rc) i zostanie ona porównana z wartością zadaną (Rt). W przypadku, gdy wartość zadana będzie większa niż aktualna wartość na wyjściu, aktualna wartość będzie zwiększana zgodnie ze współczynnikiem określanym przez czas przyspieszenia (Ta) i maksymalną wartość wyjściową (Om), do momentu, zrównania się z wartością zadaną (w tym czasie ACC = 1). W przypadku, gdy wartość zadana będzie mniejsza niż aktualna wartość na wyjściu, aktualna wartość będzie zmniejszana zgodnie ze współczynnikiem określanym przez czas zwolnienia (Td) i maksymalną wartość wyjściową (Om) do momentu, zrównania się z wartością zadaną (w tym czasie DEC = 1).
- Jeżeli wyjściowa wartość zadana (Rt) będzie większa niż maksymalna wartość wyjściowa (Om), wartość wyjściowa zostanie zablokowana przez maksymalną wartość.
- Instrukcja ta umożliwia płynne przyspieszanie i zwalnianie poprzez kontrolę aktualnej wartości (Rc) na wyjściu analogowym (R39044~R3967).
- Dla prawidłowego wykonania, zadana wartość wyjściowa (Rt) powinna pozostać niezmieniona przez co najmniej dwa kolejne skany sterownika.
- Instrukcja wymaga 4 rejestrów roboczych, które nie mogą być ponownie użyte w programie
- Instrukcja ta przeznaczona jest do operacji na wartościach dodatnich. Można jednak, przy pomocy prostego i krótkiego programu, zastosować ją do operacji na wartościach ujemnych. Patrz przykład 2.

FUN98 RAMP2	NADĄŻAJĄCE NARASTANIE NA WYJŚCIU ANALOGOWYM (TRACKING TYPE RAMP FUNCTION FOR D/A OUTPUT)	FUN98 RAMP2

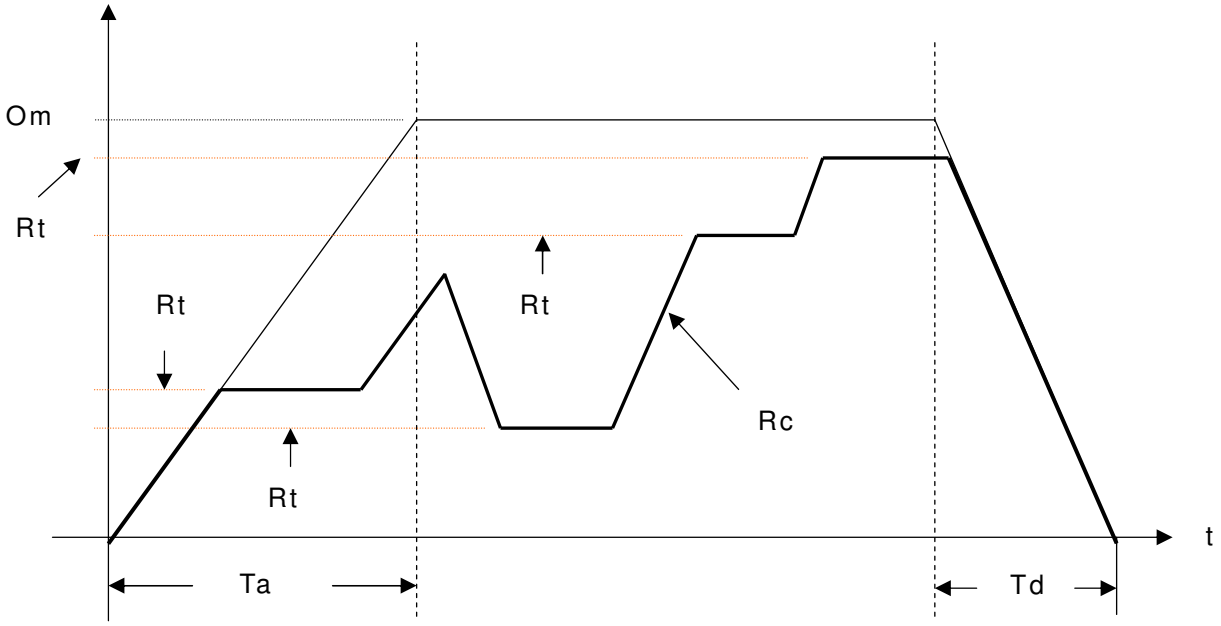
Przykład 1: Kontrola przyspieszenia/opóźnienia



D10 : Maksymalna wartość wyjściowa równa 16383
D0 : Czas narastania wartości na wyjściu od 0 do maksimum równy 30000 ms
D1 : Czas opadania wartości na wyjściu od maksimum do 0 równy 20000 ms
D100: Wyjściowa wartość zadana równa 8192
R3904:Rejestr aktualnej wartości na wyjściu analogowym
D1000~D1003:Rejestry robocze

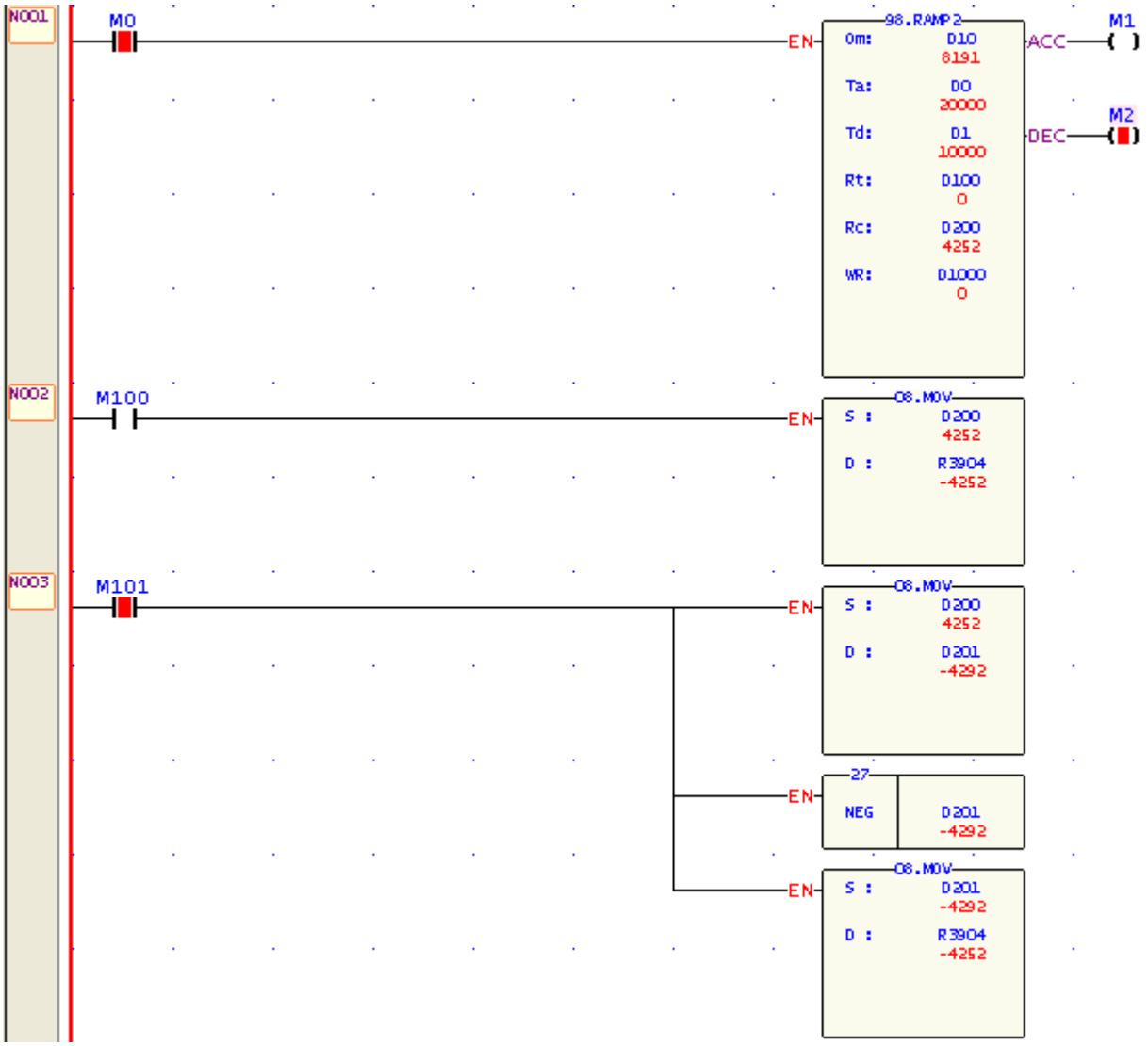
Opis: Jeżeli M0 = 0, aktualna wartość wyjściowa zostanie natychmiast wyzerowana (bez „rampy”).

Jeżeli M0 = 1, na wyjściu pojawi się wartość R3904 i zostanie ona porównana z wartością zadaną (D100). Jeżeli D100＞R3904, to aktualna wartość wyjściowa R3904 będzie zwiększana zgodnie ze współczynnikiem 16383 / 30000 (Om = 16383, Ta = 30000), do momentu, aż R3904 = D100 (w tym czasie ACC = 1). Jeżeli D100＜R3904, to aktualna wartość wyjściowa R3904 będzie zmniejszana zgodnie ze współczynnikiem 16383 / 20000 (Om = 16383, Td = 20000), do momentu, aż R3904 = D100 (w tym czasie DEC= 1).



FUN98 RAMP2	NADĄŻAJĄCE NARASTANIE NA WYJŚCIU ANALOGOWYM (TRACKING TYPE RAMP FUNCTION FOR D/A OUTPUT)	FUN98 RAMP2

Przykład 2: Dodatnie i ujemne wyjś cie sterują ce ACC/DEC



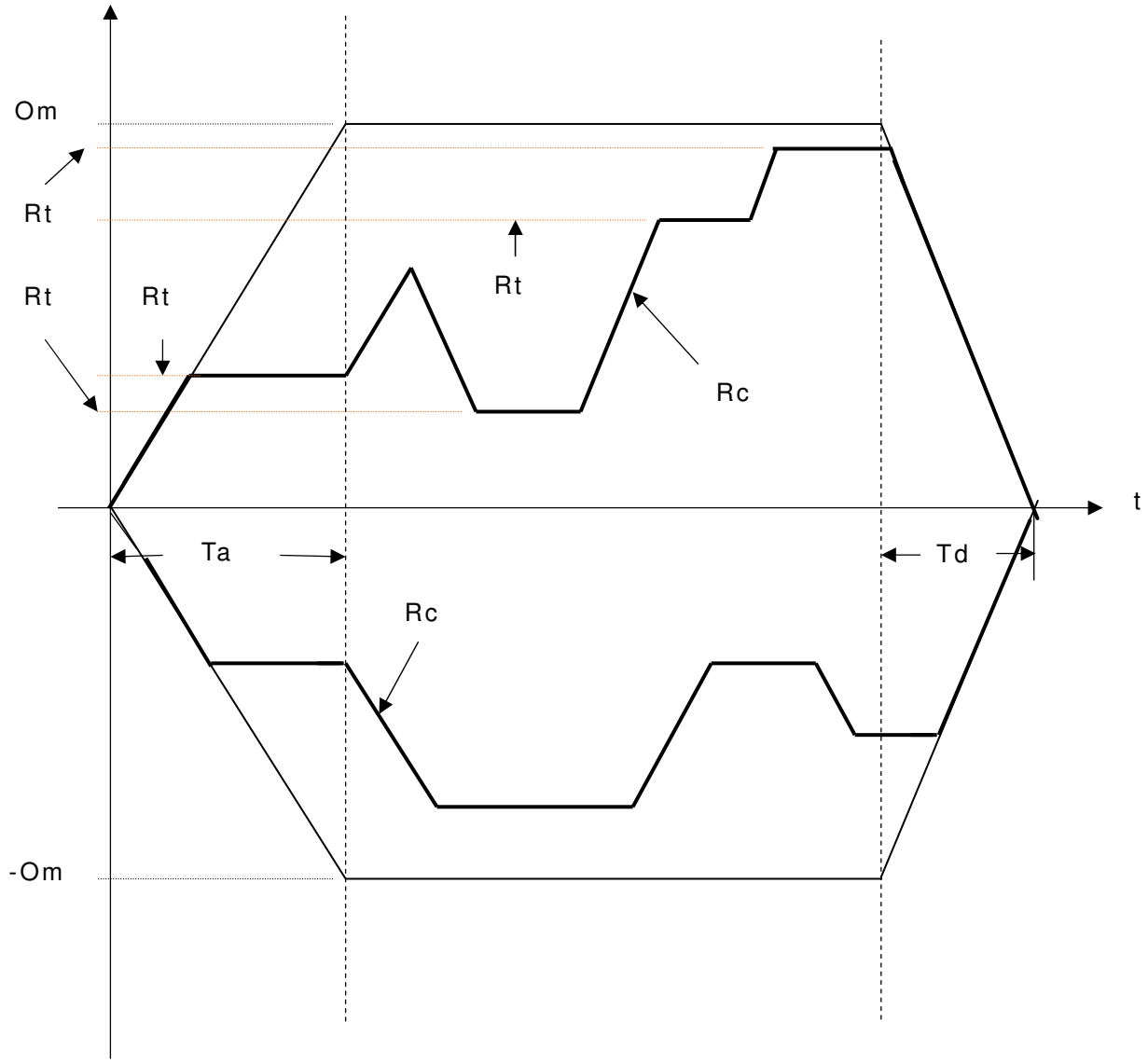
D10 : Maksymalna wartość wyjściowa równa 8191
D0 : Czas narastania wartości na wyjściu od 0 do maksimum równy 20000 ms
D1 : Czas opadania wartości na wyjściu od maksimum do 0 równy 10000 ms
D100: Wyjściowa wartość zadana równa 0
D200: Rejestr aktualnej wartości na wyjściu analogowym
D1000~D1003: Rejestry robocze

Jeżeli M0 = 0, to aktualna wartość wyjściowa zostanie natychmiast wyzerowana (bez „rampy”).

Jeżeli M0 = 1, na wyjściu pojawi się wartość D200 i zostanie ona porównana z wartością zadaną (D100). Jeżeli D100>D200, to aktualna wartość wyjściowa D200 będzie zwiększana zgodnie ze współczynnikiem 8191/20000 (Om=8191, Ta=20000), do momentu, aż D200=D100 (w tym czasie ACC = 1). Jeżeli D100<D200, to aktualna wartość wyjściowa D200 będzie zmniejszana zgodnie ze współczynnikiem 16383 8191/10000 (Om=8191, Td=10000), do momentu, aż D200=D100 (w tym czasie DEC= 1).

M100 = 1, dodatnie wyjście sterujące; M101 = 1, ujemne wyjście sterujące. Zadana wartość wyjściowa (D100) jest zawsze wartością dodatnią w zakresie 0~65535.

FUN98 RAMP2	NADĄŻAJĄCE NARASTANIE NA WYJŚCIU ANALOGOWYM (TRACKING TYPE RAMP FUNCTION FOR D/A OUTPUT)	FUN98 RAMP2



Funkcje operacji na tabelach

Nr fun.	Mnemonik	Funkcjonalność	Nr fun.	Mnemonik	Funkcjonalność
100	R→T	Wpisz rejestr do tabeli	107	T_FIL	Wypełnij tabelę
101	T→R	Wpisz wartość z tabeli do rejestru	108	T_SHF	Przesuń tabelę
102	T→T	Wpisz wartość z tabeli do tabeli	109	T_ROT	Obróć tabelę
103	BT_M	Tabela do tabeli	110	QUEUE	Kolejka
104	T_SWP	Zamień tabele	111	STACK	Stos
105	R-T_S	Wyszukaj wartość w tabeli	112	BKCMP	Porównanie blokowe
106	T-T_C	Porównaj wartości w tabelach	113	SORT	Sortowanie wartości

- Tabelę stanowi 2 lub więcej kolejnych rejestrów (16- lub 32-bitowych). Liczba rejestrów w tabeli nazywa się długością tabeli (L). Obiektem operacyjnym funkcji operujących na tabelach, zawsze jest pojedynczy rejestr (tj. dane 16-to lub 32-bitowe).

- Funkcje operacji na tabelach wykorzystywane są głównie do przetwarzania danych, tj. przesuwania, kopiowania, porównywania, wyszukiwania itp. pomiędzy tabelami i rejestrami lub pomiędzy tabelami. Funkcje te są wygodne w zastosowaniu.

- Większość funkcji operujących na tabelach, wykorzystuje wskaźnik, dla wskazania rejestru docelowego w tabeli, na którym ma odbyć się operacja. Wskaźnikami dla instrukcji tabel 16-to i 32-bitowych będą zawsze rejestry 16-bitowe. Poprawny zakres wskaźnika to 0 do L-1. Zakres ten odnosi się do rejestrów T_0 do T_{L-1} (suma rejestrów L). Poniżej przedstawiony został schemat dla tabel 16- i 32-bitowych.

- W operacjach przesuwania w lewo/prawo oraz obrotu w lewo/prawo, wskazywany jest kierunek operacji. Patrząc w kierunku rosnącym: najstarszy adres w tabeli nazywany jest lewym, najmłodszy nazywany jest prawym (patrz schemat poniżej).

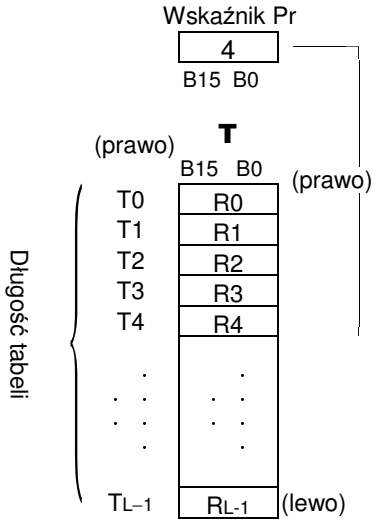


Tabela 16-bitowa

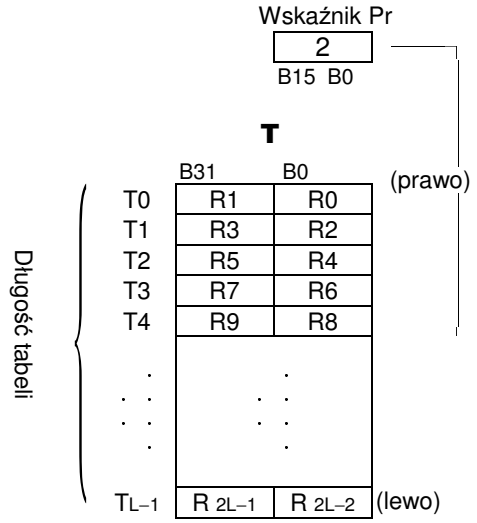
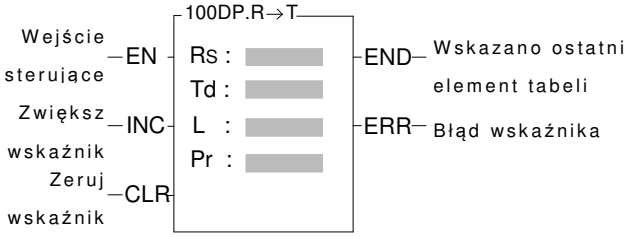


Tabela 32-bitowa

FUN100 D P R→T	WPISZ REJESTR DO TABELI (REGISTER TO TABLE MOVE)	FUN100 D P R→T

SYMBOL DRABINKOWY



Rs : Dane źródłowe, stała lub adres rejestru

Td : Adres początkowy tabeli docelowej

L : Długość tabeli docelowej

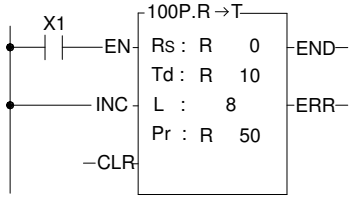
Pr : Rejestr wskaźnika

Rs, Td może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

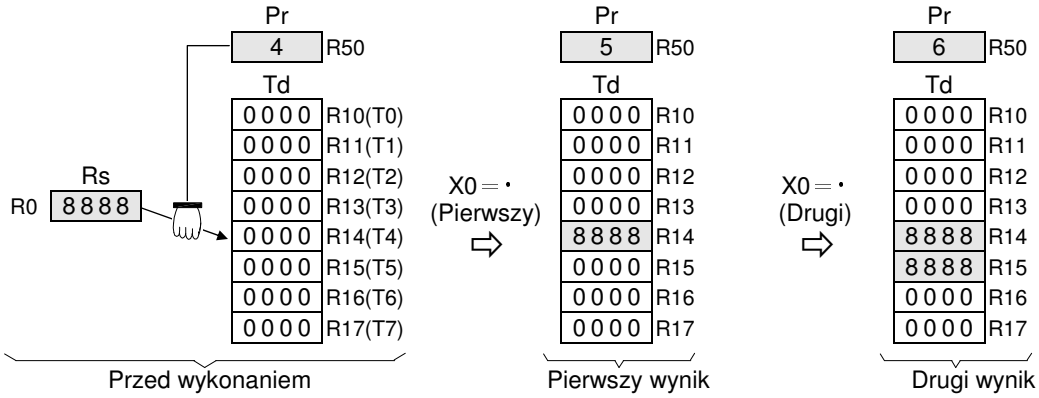
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32bit +/- liczba	V、Z P0~P9
Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~2048	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to zawartość rejestru źródłowego Rs zostanie zapisana w rejestrze Tdpr (określonym przez wartość wskaźnika Pr) w tabeli docelowej Td (o długości L). Przed realizacją tej funkcji, nastąpi sprawdzenie stanu wejścia zerowania wskaźnika „CLR". Jeżeli „CLR" = 1, to najpierw wskaźnik Pr zostanie wyzerowany i nastąpi kontynuacja operacji wpisania wartości Rs do tabeli. Po zakończeniu operacji zostanie sprawdzona wartość Pr. W przypadku, gdy wartość Pr osiągnie L-1 (wskazanie ostatniego rejestru w tabeli), to znacznik „END" zostanie ustawiony na 1 i nastąpi zakończenie wykonywania funkcji. W przypadku, gdy wartość Pr będzie mniejsza niż L-1, to sprawdzany jest stan sygnału na wejściu sterującym „INC". Jeżeli „INC" = 1, to wartość wskaźnika Pr zostaje zwiększona. Wejście sterujące „CLR" może pracować niezależnie, bez wpływu pozostałych wejść.

- Prawidłowym zakresem wskaźnika Pr jest 0 do L-1. W przypadku wykroczenia poza ten zakres, wyjście błędu „ERR" zostanie ustawiony na 1, a funkcja nie zostanie wykonana.



- W przykładzie po lewej, przed wykonaniem funkcji, wskaźnik Pr = 4, wszystkie rejestry tabeli Td mają wartości 0, a wartość Rs = 8888. Poniższy schemat przedstawia wyniki operacji, gdy X1 przejdzie dwukrotnie ze stanu 0 na 1.
- Ponieważ INC = 1, to po każdym wykonaniu funkcji, wartość Pr zostanie zwiększona o 1.



FUN101 D P T→R	WPISZ WARTOŚĆ Z TABELI DO REJESTRU (TABLE TO REGISTER MOVE)	FUN101 D P T→R

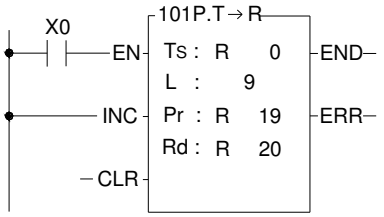
SYMBOL DRABINKOWY

Ts : Adres początkowy tabeli źródłowej
L : Długość tabeli źródłowej
Pr : Rejestr wskaźnika
Rd : Adres rejestru docelowego
Ts, Rd może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

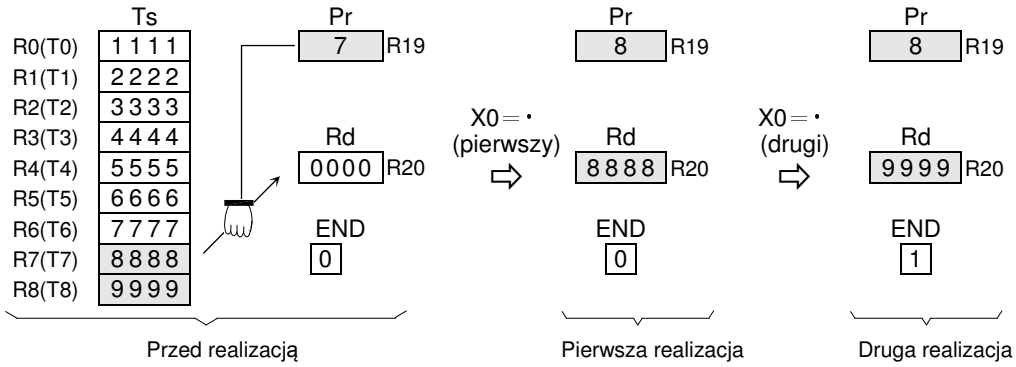
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32bit +/- liczba	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○		
Pr		○	○	○	○	○	○		○	○*	○*	○	2~2048	
Rd		○	○	○	○	○	○		○	○*	○*	○		○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to wartość z rejestru Tspr (określonego przez wartość wskaźnika Pr) w tabeli źródłowej Ts (o długości L), zostanie zapisana w rejestrze docelowym Rd. Przed wykonaniem tej funkcji następuje sprawdzenie stanu wejścia zerowania wskaźnika „CLR". Jeżeli „CLR" = 1, najpierw wartość wskaźnika Pr zostanie wyzerowana i nastąpi kontynuacja operacji pobierania wartości z tabeli. W przypadku, gdy wartość wskaźnika Pr osiągnie L-1 (wskazanie ostatniego rejestru w tabeli), to znacznik „END" zostanie ustawiony na 1 i nastąpi zakończenie wykonywania funkcji. W przypadku, gdy wartość Pr będzie mniejsza niż L-1, to sprawdzany jest stan sygnału na wejściu sterującym „INC". Jeżeli „INC" = 1, to wartość Pr zostanie zwiększona. Wejście sterujące „CLR" może pracować niezależnie, bez wpływu pozostałych wejść.

- Prawidłowym zakresem wskaźnika Pr jest 0 do L-1. W przypadku wykroczenia poza ten zakres, wyjście błędu „ERR" zostanie ustawiony na 1, a funkcja nie zostanie wykonana.

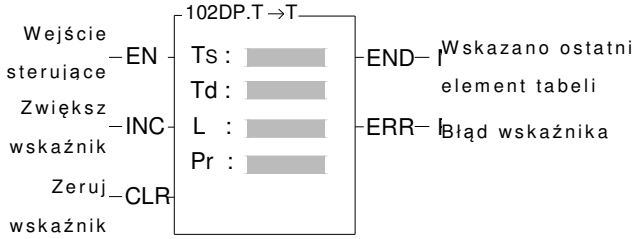


- W przykładzie po lewej, przed wykonaniem funkcji, wartość wskaźnika Pr = 7, a wartości w tabeli Ts i rejestrze Rd są takie, jak przedstawiono po lewej stronie. Schemat poniżej przedstawia wyniki operacji, gdy X0 przejdzie dwukrotnie ze stanu 0 na 1.
- Przy drugim wykonaniu funkcji, wskaźnik Pr osiągnął największą wartość, dlatego też nie został on zwiększony.



FUN102 D P T→T	WPISZ WARTOŚĆ Z TABELI DO TABELI (TABLE TO TABLE MOVE)	FUN102 D P T→T

SYMBOL DRABINKOWY

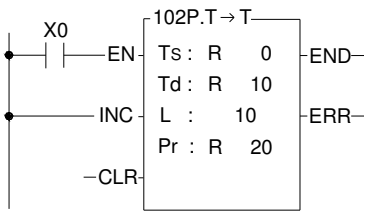


Ts : Adres początkowy tabeli źródłowej
Td : Adres początkowy tabeli docelowej
L : Długość tabel (Ts i Td)
Pr : Rejestr wskaźnika
Ts, Td może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

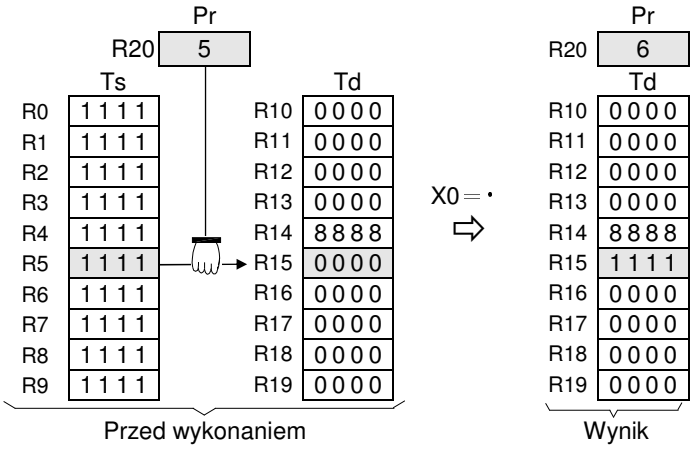
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 2048	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to wartość z rejestru Tspr (określonego przez wartość wskaźnika Pr) w tabeli źródłowej, zostanie wpisana do rejestry Tdpr (który także jest określany przez wartość wskaźnika Pr) w tabeli docelowej. Przed wykonaniem tej funkcji, następuje sprawdzenie stanu wejścia zerowania wskaźnika „CLR". Jeżeli „CLR" = 1, wskaźnik Pr zostaje wyzerowany i następuje wykonanie operacji przesunięcia (w takim przypadku Ts0→Td0). Po wykonaniu przesunięcia, następuje sprawdzenie wartości wskaźnika Pr. W przypadku, gdy wartość Pr osiągnie L-1 (wskazanie ostatniego rejestru w tabeli), to znacznik „END" zostaje ustawiony na 1 i następuje zakończenie wykonania funkcji. W przypadku, gdy wartość Pr będzie mniejsza niż L-1, to sprawdzany jest stan sygnału na wejściu sterującym „INC". Jeżeli „INC" = 1, to wartość wskaźnika Pr zostaje zwiększona o 1. Wejście sterujące „CLR" może pracować niezależnie, bez wpływu pozostałych wejść.

- Zakres efektywny wskaźnika wynosi od 0 do L-1. W przypadku wykroczenia poza zakres, wskaźnik błędu „ERR" zostanie ustawiony na 1, a instrukcja nie będzie zrealizowana

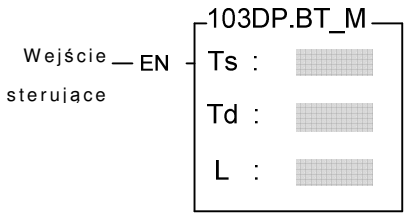


- Schemat poniżej przedstawia stan przed wykonaniem funkcji. Kiedy X0 zmieni się z 0 na 1, zawartość R5 w tabeli Ts zostanie skopiowana do R15 i nastąpi zwiększenie wskaźnika R20 o 1.



FUN103 D P BT_M	TABELA DO TABELI (BLOCK TABLE MOVE)	FUN103 D P BT_M

SYMBOL DRABINKOWY



Ts :Adres początkowy tabeli źródłowej

Td : Adres początkowy tabeli docelowej

L: Długość tabel źródłowej i docelowej

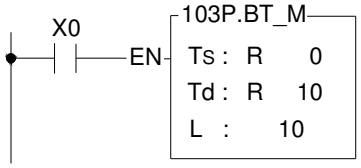
Ts, Td może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

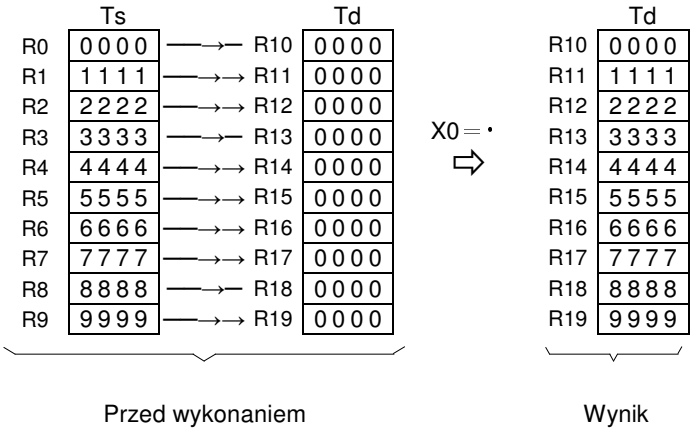
- W funkcji tej, tabele źródłowa i docelowa są jednakowej długości. Przy wykonywaniu tej funkcji, wszystkie dane z tabeli źródłowej Ts, są w całości kopiowane do tabeli docelowej Td. Funkcja nie wykorzystuje wskaźnika.

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to wszystkie dane z tabeli źródłowej Ts (o długości L) zostaną skopiowane do tabeli docelowej Td, o tej samej długości.

- Za każdym razem, kiedy funkcja ta jest wykonywana, następuje kopiowanie całej tabeli. W przypadku dużej długości tabel, wykonanie tej operacji jest czasochłonne W praktyce, należy wykorzystać opcję wykonania P (wykonywanie przy stanie narastającym na wejściu EN), w celu zapobiegnięcia stracie, czasu spowodowanej powtarzaniem tego samego przesunięcia, w każdym skanie programu.

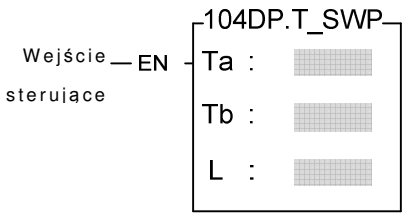


- Dolny schemat po lewej stronie, przedstawia stan przed wykonaniem instrukcji. Kiedy X0 zmieni się z 0 na 1, to zawartość R0~R9 z tabeli Ts zostanie skopiowana do R10~R19.



FUN104 D P T_SWP	ZAMIEŃ TABELE (BLOCK TABLE SWAP)	FUN104 D P T_SWP

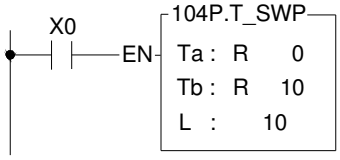
SYMBOL DRABINKOWY



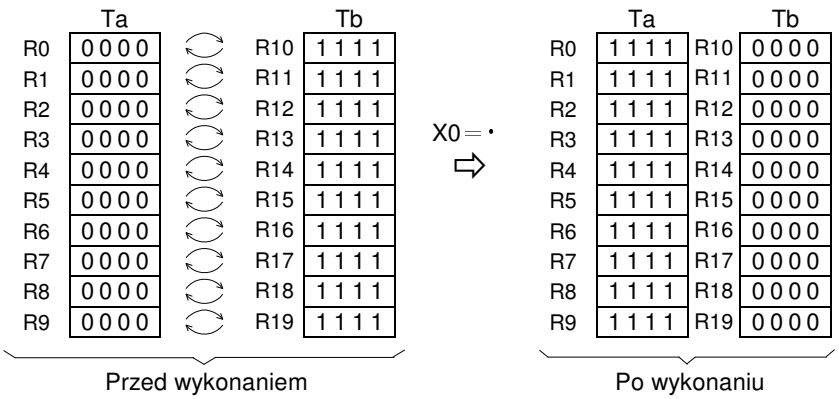
Ta : Adres początkowy tabeli a
Tb : Adres początkowy tabeli b
L : Długość tabel a i b
Ts, Td może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ta	○	○	○	○	○	○	○	○*	○*	○		○
Tb	○	○	○	○	○	○	○	○*	○*	○		○
L						○			○*	○	○	

- Funkcja zamienia zawartość tabel a i b. Dlatego też tabele te muszą być tej samej długości, a rejestry w tabeli muszą mieć możliwość zapisu. Ponieważ podczas wykonywania funkcji, zamieniane są zawartości całych tabel, nie ma potrzeby stosowania wskaźnika.
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to zawartości tabel a i b zostaną zamienione.
- Funkcja zamienia zawartości całych tabel, przy każdym jej wywołaniu. W przypadku dużej długości tabel, wykonanie tej operacji jest czasochłonne. Aby temu zapobiec, należy stosować opcję wykonania P (wykonywanie przy stanie narastającym na wejściu EN).

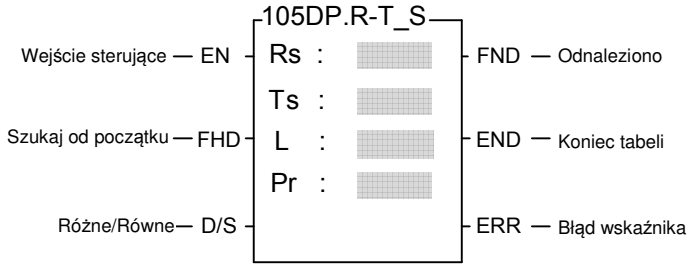


- Dolny schemat po lewej, przedstawia stan przed realizacją funkcji. Kiedy X0 zmieni się z 0 na 1, zawartość R0~R9 w tabeli Ta zostanie zamieniona z zawartością R10~R19 w tabeli Tb.



FUN105 D P R-T_S	WYSZUKAJ WARTOŚĆ W TABELI (REGISTER TO TABLE SEARCH)	FUN105 D P R-T_S

SYMBOL DRABINKOWY

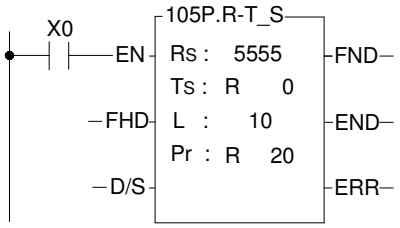


Rs : Dana do wyszukania; wartość stała lub rejestr
Ts : Adres początkowy przeszukiwanej tabeli
L : Długość tabeli
Pr : Rejestr wskaźnika
Rs, Ts mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

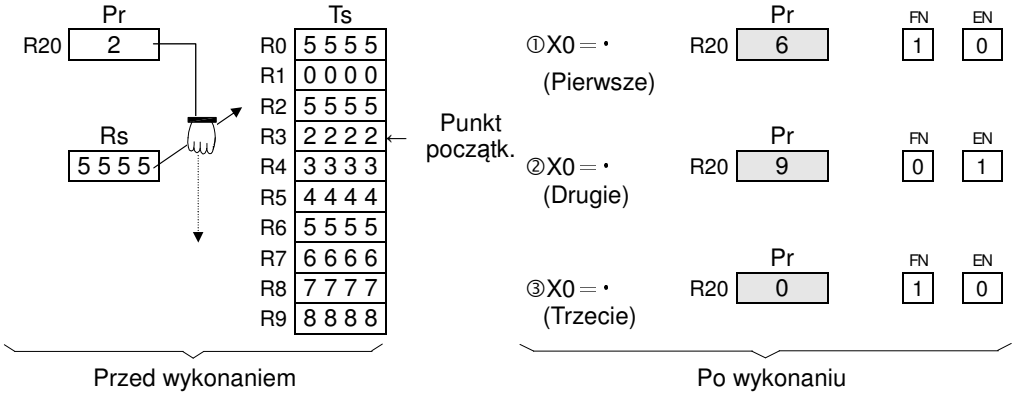
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	Liczba 16/32-bit	V、Z P0~P9
Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja rozpocznie wyszukiwanie od pierwszego rejestru w tabeli Ts (kiedy „FHD" = 1 lub gdy wartość wskaźnika Pr osiągnie wartość L-1), lub od kolejnego rejestru (Tspr + 1) określonego przez wartość wskaźnika (kiedy „FHD" = 0 i Pr < L-1), w celu odszukania pierwszej danej różnej niż Rs (kiedy D/S = 1) lub równej Rs (kiedy D/S = 0). Po odnalezieniu rejestru spełniającego określone warunki, nastąpi automatyczne zatrzymanie wyszukiwania, a wskaźnik Pr wskaże odnaleziony rejestr, jednocześnie wyjście odnaleziono „FND" ustawione zostanie na 1. Po przeszukaniu ostatniego rejestru tabeli, wykonanie funkcji zostanie zatrzymane niezależnie od tego czy rejestr został odnaleziony. W takim wypadku, znacznik „END" zostanie ustawiony na 1, a wartość Pr zatrzyma się na L-1. Przy kolejnej realizacji instrukcji, wartość Pr automatycznie powróci na górną pozycję w tabeli (Pr = 0) w celu ponownego rozpoczęcia wyszukiwania od początku tabeli.

- Prawidłowym zakresem wskaźnika Pr jest 0 do L-1. W przypadku wykroczenia poza ten zakres, wyjście błędu „ERR" zostanie ustawiony na 1, a funkcja nie zostanie wykonana.

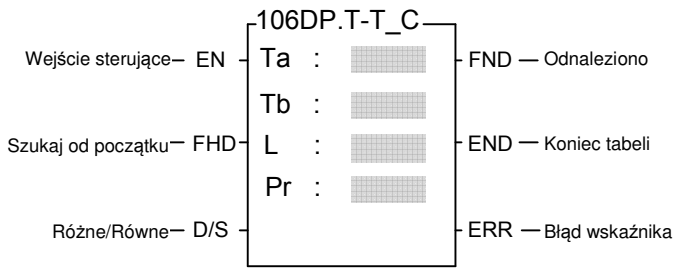


- Funkcja po lewej, realizuje wyszukiwanie rejestru o wartości 5555 w tabeli Ts (ponieważ D/S = 0, wyszukiwana jest identyczna wartość). Przed rozpoczęciem wykonywania funkcji, wskaźnik wskazuje na R2, a początkowym punktem wyszukiwania jest Pr + 1 (tj. wyszukiwanie rozpoczyna się od R3). Po 3-krotnej zmianie X0 z 0→1, wyniki poszczególnych wyszukiwań, będą następujące:



FUN106 D P T-T_C	PORÓWNAJ WARTOŚCI W TABELACH (TABLE TO TABLE COMPARE)	FUN106 D P T-T_C

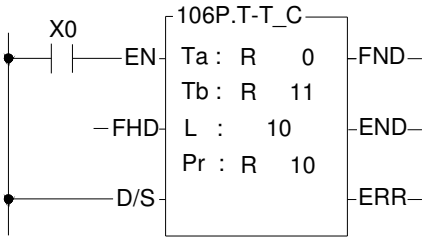
SYMBOL DRABINKOWY



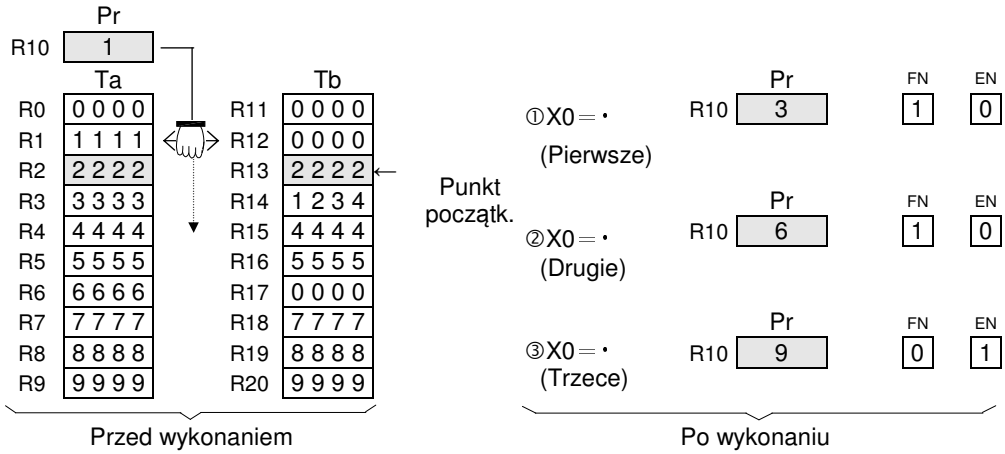
Ta : Adres początkowy tabeli a
Tb : Adres początkowy tabeli b
L : Długości tabel
Pr : Rejestr wskaźnika
Ta, Tb mogą łączyć się z V, Z, P0~P9 P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ta	○	○	○	○	○	○	○	○	○	○	○	○		○
Tb	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja rozpocznie wyszukiwanie od pierwszej pary rejestrów w tabelach Ta i Tb (kiedy „FHD" = 1 lub gdy Pr osiągnie wartość L-1) lub od kolejnej pary rejestrów (Tapr+1 i Tbpr+1) określonej przez wartość wskaźnika Pr (kiedy „FHD" = 0 i Pr < L-1), w celu odszukania par rejestrów o różnych wartościach (kiedy „D/S" = 1) lub o identycznych wartościach (kiedy „D/S" = 0). Po odnalezieniu pary rejestrów (o tej samej lub innej wartości), nastąpi automatyczne zatrzymanie wyszukiwania, a wskaźnik Pr wskaże odnalezioną parę rejestrów, jednocześnie wyjście odnaleziono „FND" ustawione zostanie na 1. Po przeszukaniu ostatniego rejestru tabeli, wykonanie funkcji zostanie zatrzymane niezależnie od tego czy para rejestrów został odnaleziona czy nie. W takim wypadku, znacznik „END" zostanie ustawiony na 1, a wartość Pr zatrzyma się na L-1. Przy kolejnej realizacji instrukcji, wartość Pr automatycznie powróci na górną pozycję w tabeli (Pr = 0) w celu ponownego rozpoczęcia wyszukiwania od początku tabel.
- Prawidłowym zakresem wskaźnika Pr jest 0 do L-1. Wartość wskaźnika, podczas wykonywania wyszukiwania, nie powinna być zmieniana w innych częściach programu, gdyż może to wpływać na wyniki wyszukiwania. W przypadku wykroczenia wartości wskaźnika poza zakres, wyjście błędu „ERR" zostanie ustawiony na 1, a funkcja nie zostanie wykonana.

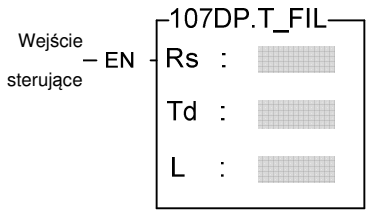


- Funkcja po lewej rozpoczyna wyszukiwanie od rejestru znajdującego się poniżej rejestru określonego przez wskaźnik (ponieważ „FHD" = 0) w celu wyszukania par rejestrów o różnych wartościach (ponieważ „D/S" = 1) w dwóch tabelach. Na początku, Pr wskazuje na Ta1 i Tb1. W pozycjach 1, 3, 6 tabel znajdują się 3 różne wartości. Ale funkcja nie rozpoczyna wyszukiwania od pierwszej pary rejestrów, zatem wyszukiwanie rozpoczyna się od 3 pozycji w dół. Po 3-krotnej zmianie X0 z 0→1, wyniki każdego wyszukiwania będą następujące



FUN107 D P T_FIL	WYPEŁNIJ TABELĘ (TABLE FILL)	FUN107 D P T_FIL

SYMBOL DRABINKOWY



Rs : Dana źródłowa do wypełnienia nią tabeli; wartość stała lub rejestr

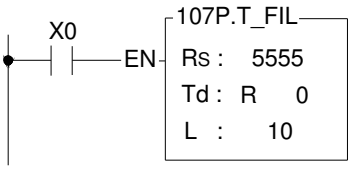
Td : Adres początkowy tabeli docelowej

L :Długość tabeli

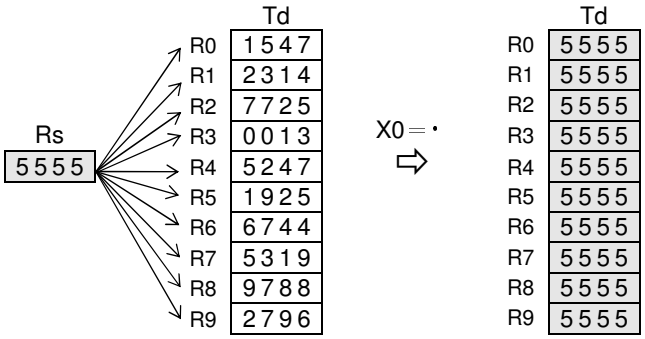
Rs, Td mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Argument	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	Liczba 16/32-bit	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), wartość znajdująca się w Rs, zostanie zapisana we wszystkich rejestrach tabeli Td.
- Funkcja ta wykorzystywana jest głównie do zerowania wartości w tabeli lub wpisywania identycznych wartości do całej tabeli. Funkcję tą należy wywoływać stanem narastającym na wejściu sterującym (z opcją P).



- Funkcja po lewej, wypełni całą tabelę Td, wartościami 5555. Wyniki będą następujące:

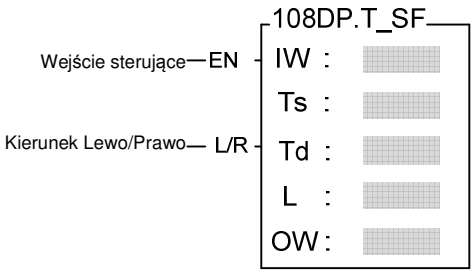


Przed wykonaniem

Po wykonaniu

FUN108 D P T_SHF	PRZESUŃ ATBELĘ (TABLE SHIFT)	FUN108 D P T_SHF

SYMBOL DRABINKOWY



IW : Dana do wsunięcia do tabeli po wykonaniu operacji przesunięcia; wartość stała lub rejestr

Ts : Adres początkowy tabeli źródłowej

Td : Adres początkowy tabeli przechowującej wynik przesunięcia

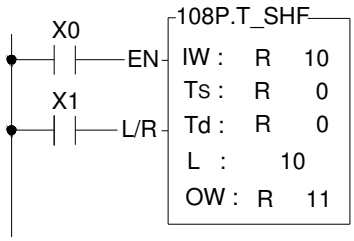
L : Długości tabel Ts i Td

OW : Adres rejestru przechowującego wysuniętą wartość z tabeli

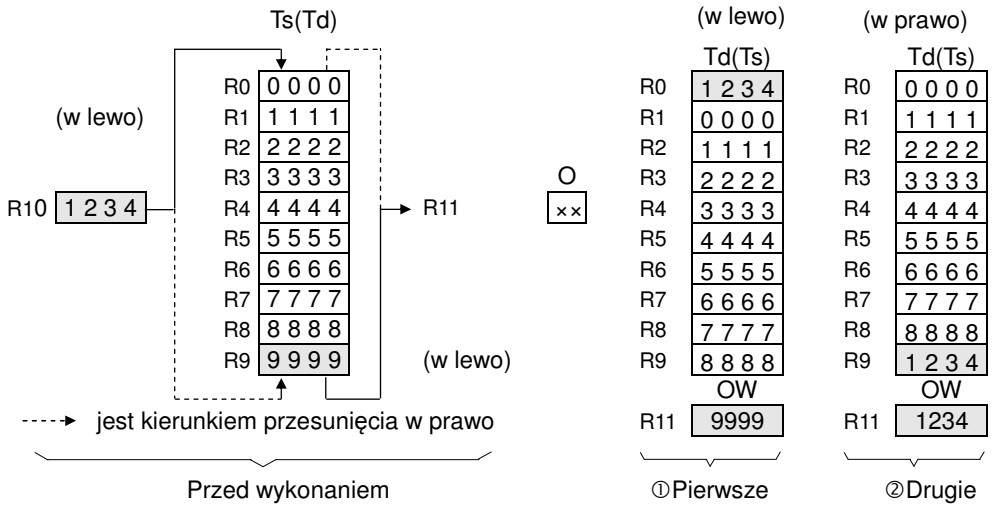
Ts, Td mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	Liczba 16/32-bit	V、Z P0~P0
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	
OW		○	○	○	○	○	○		○	○*	○*	○		

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), wszystkie dane z tabeli Ts zostaną przesunięte o jedną pozycję w lewo (kiedy „L/R" = 1) lub w prawo (kiedy „L/R" = 0). Puste miejsce po operacji przesunięcia zostanie wypełnione wartością IW. Wynik przesunięcia zapisany zostanie w tabeli Td. Wartość wysunięta z tabeli, zostanie zapisane w OW.

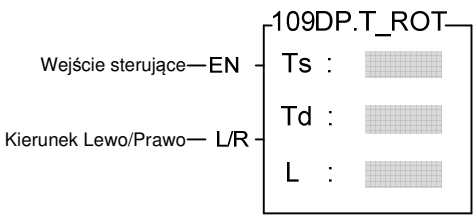


- W programie po lewej, Ts i Td są tą samą tabelą. Dlatego też, tabela sama się przesuwa i zapisuje wynik przesunięcia do siebie samej. Najpierw realizowane jest przesunięcie w lewo (X1 =1, X0 z 0→1), a następnie w prawo (X1 = 0 i X0 z 0→1). Wyniki tych przesunięć przedstawione są poniżej.



FUN109 D P T_ROT	OBRÓĆ TABELĘ (TABLE ROTATE)	FUN109 D P T_ROT

SYMBOL DRABINKOWY



Ts : Adres początkowy tabeli źródłowej

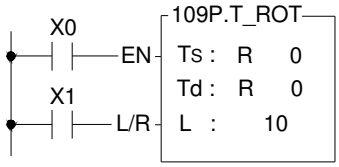
Td : Adres początkowy tabeli przechowującej wynik obrotu

L : Długości tabel

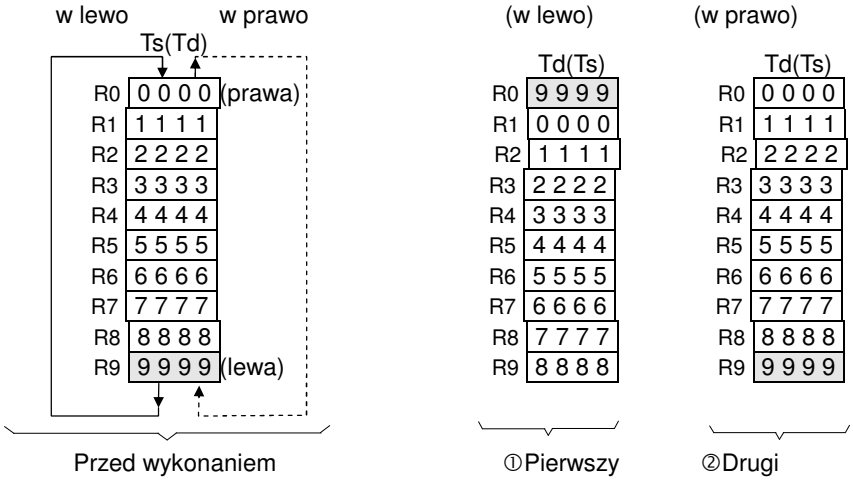
Ts, Td mogą łączyć się z V, Z, P0~P9 celu zastosowania adresowania pośredniego

Zakres	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Argument	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), dane z tabeli Ts zostaną obrócone o 1 pozycję w lewo (kiedy „L/R” = 1) lub w prawo (kiedy „L/R” = 0). Wyniki operacji obrotu zostaną zapisane w tabeli Td.

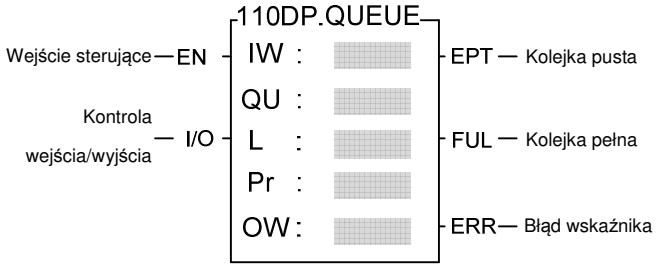


- W programie po lewej, Ts i Td są tą samą tabelą. Po obrocie tabeli, wynik obrotu zostanie zapisany do niej samej. Najpierw nastąpi jeden obrót w lewo (X1 = 1, X0 z 0→1), a następnie w prawo (X1 = 0, X0 z 0→1). Wyniki tych obrotów przedstawione są poniżej.



FUN110 D P QUEUE	KOLEJKA (QUEUE)	FUN110 D P QUEUE

SYMBOL DRABINKOWY



IW : Wartość wprowadzana do kolejki; wartość stała lub rejestr

QU : Adres początkowy kolejki

L : Długość kolejki

Pr : Rejestr wskaźnika

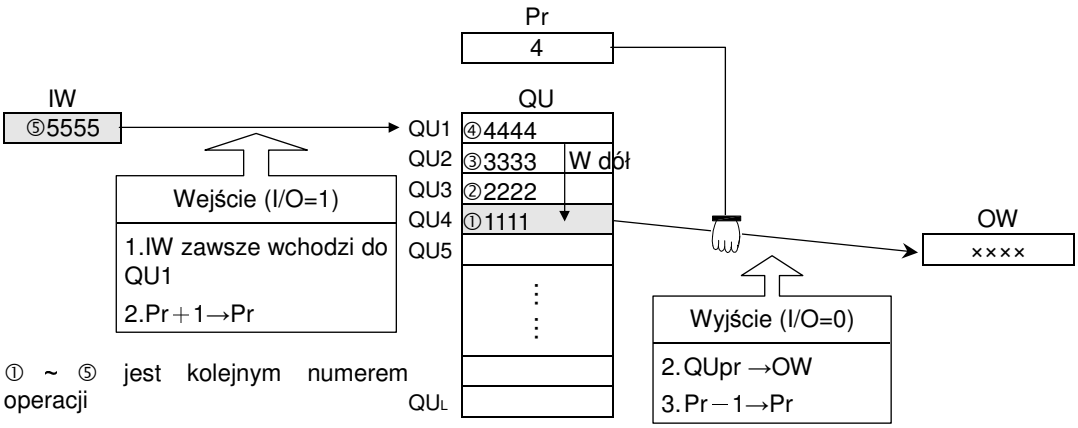
OW : Rejestr do zapisu wartości wychodzących z kolejki

QU może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	Liczba 16/32-bit	V、Z P0~P9
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
QU		○	○	○	○	○	○		○	○	○*	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		
OW		○	○	○	○	○	○		○	○*	○*	○		

- Kolejka jest także rodzajem tabeli. Różni się od zwykłej tabeli tym, że zakres numerów rejestrów w kolejce wynosi od 1 do L, a nie od 0 do L-1. Innymi słowy, wartości QU_1~QU_L odnoszą się odpowiednio do wskaźników Pr = 1 do L, a Pr = 0 oznacza, że kolejka jest pusta.

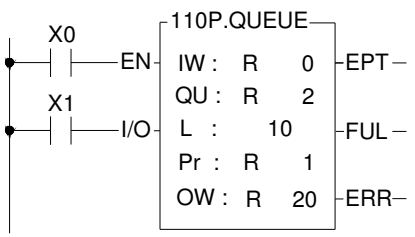
- Kolejka jest instrukcją typu FIFO, tj. dane, które pierwsze wejdą do kolejki będą pierwszymi danymi, które z niej wyjdą. Kolejka składa się z L kolejnych 16- lub 32-bitowych rejestrów (instrukcja D) rozpoczynając od rejestru QU, jak pokazano na poniższym schemacie:



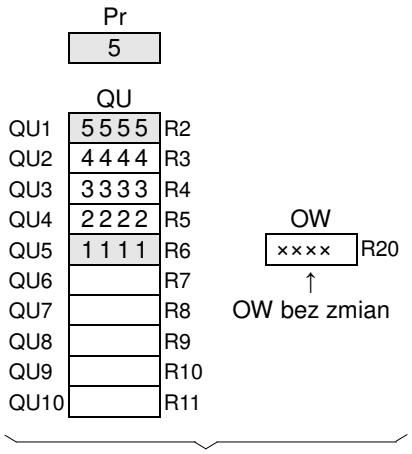
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to stan wejścia „I/O" określa czy dane IW wejdą do kolejki (kiedy „I/O" = 1) czy wyjdą z niej i zostaną przeniesione do OW (kiedy „I/O" = 0). Jak pokazano na schemacie powyżej, wartości IW będą zawsze zapisywane w pierwszym rejestrze (QU1) w kolejce. Po wprowadzeniu wartość do kolejki, wskaźnik Pr zostanie automatycznie zwiększona o 1, aby wskazywał zawsze daną, która jako pierwsza weszła do kolejki. Po wyjściu z kolejki wartości wskazanej przez Pr, zostaje ona przeniesiona bezpośrednio do OW. Wartość Pr zostaje wtedy zmniejszona o 1, aby wskaźnik wskazywał na najstarszą z wartości, które pozostały w kolejce.

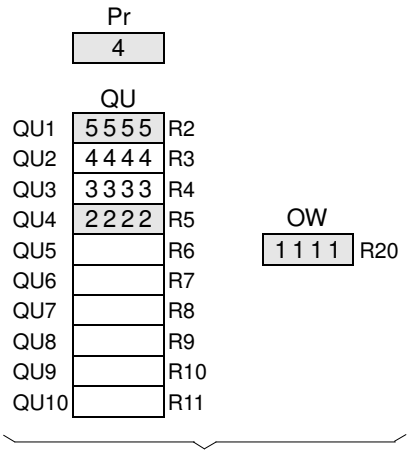
FUN110 D P QUEUE	KOLEJKA (QUEUE)	FUN110 D P QUEUE

- Jeżeli żadne wartości nie weszły jeszcze do kolejki lub wszystkie dane wyszły z kolejki (Pr = 0), to stan znacznika pustej kolejki EPT, zostanie ustawiony na 1. Wówczas, nawet w przypadku kolejnej operacji wyjścia danych, instrukcja ta nie zostanie wykonana. W przypadku, gdy dane będą wprowadzane do kolejki, ale nigdy nie będą z niej wyprowadzane, kolejka po pewnym czasie stanie się pełna (wskaźnik Pr wskaże pozycję QU_L), a stan znacznika pełnej kolejki FUL, zostanie ustawiony na 1. Wówczas, nawet w przypadku kolejnej operacji wejścia danych, instrukcja ta nie zostanie wykonana. Wskaźnik w tej funkcji wykorzystywany jest podczas wprowadzania danych do kolejki, w celu wskazania danej, która weszła do niej najwcześniej. Inne fragmenty programu, nie powinny zmieniać wartości wskaźnika, gdyż funkcja ta może się wykonać nieprawidłowo. W przypadku aplikacji wymagającej ustawienia wartości Pr, dopuszczalny zakres dla tej wartości wynosi od 0 do L (0 oznacza pustą kolejkę, natomiast wartości od 1 do L oznaczają odpowiednio od QU1 do QUL). Jeżeli zakres ten zostanie przekroczony, stan wyjścia błędu wskaźnika „ERR” zostanie ustawiony na 1, a funkcja nie zostanie wykonana.



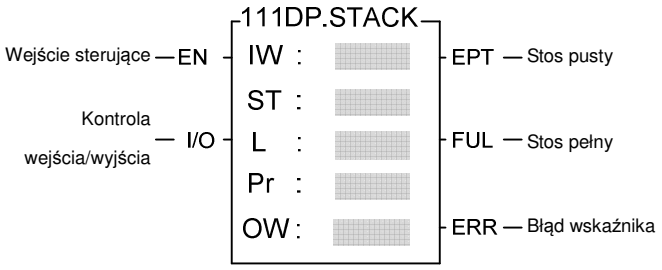
- Program po lewej zakłada, że zawartość kolejki jest identyczna jak w przypadku kolejki na poprzedniej stronie, Na początku, program zrealizuje operację wprowadzenia danych do, a następnie wyprowadzenia ich z kolejki. Wyniki przedstawione zostały poniżej. Pr zawsze wskazuje na pierwszą (najstarszą) daną spośród danych, które pozostały w kolejce.





FUN111 D P STACK	STOS (STACK)	FUN111 D P STACK

SYMBOL DRABINKOWY



IW : Wartość wprowadzana na stos; wartość stała lub rejestr

ST : Adres początkowy stosu

L : Rozmiar stosu

Pr : Rejestr wskaźnika

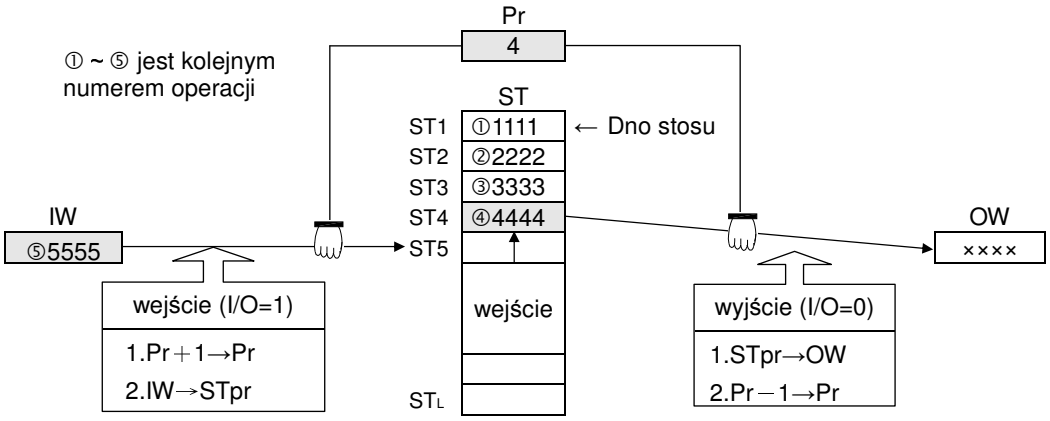
OW : Rejestr do zapisu wartości pobieranej ze stosu

ST może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	Liczba 16/32-bit	V、Z P0~P9
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
ST		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		
OW		○	○	○	○	○	○		○	○*	○*	○		

- Tak jak w przypadku kolejki, stos także jest rodzajem tabeli. Charakter wskaźnika jest dokładnie taki sam jak w przypadku kolejki, tzn. Pr = 1 do L, co odnosi się odpowiednio do ST_1 do ST_L, a Pr = 0 oznacza, że stos jest pusty.

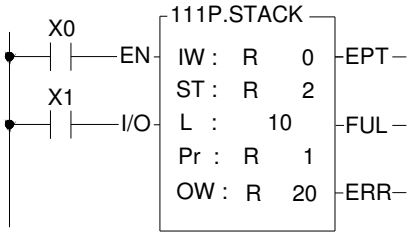
- Stos jest odwrotnością kolejki i jest instrukcją typu LIFO. Oznacza to, że wartość, która została wprowadzona do stosu jako ostatnia, opuści stos jako pierwsza. Stos składa się z L kolejnych 16- lub 32-bitowych rejestrów (instrukcja D) rozpoczynając od ST, jak pokazano na poniższym schemacie:



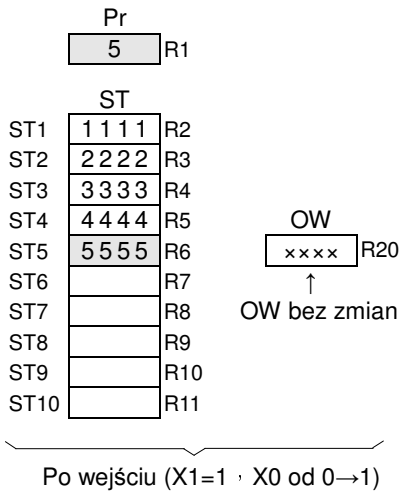
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to stan wejścia „I/O" określa czy wartość IW zostanie wprowadzona do stosu (kiedy „I/O" = 1) czy też wartość w stosie, wskazana przez wartość wskaźnika Pr (czyli dane wprowadzone do stosu najpóźniej) zostanie z niego pobrana i przeniesiona do OW (kiedy „I/O" = 0). Dane wprowadzane są na zasadzie stosu, dlatego też przed wprowadzeniem danych, wartość wskaźnika Pr zostanie zwiększona o 1 w celu wskazania szczytu stosu. Dane wskazane przez wskaźnik Pr (czyli dane wprowadzone do stosu najpóźniej), które opuszczają stos zostaną przeniesione do OW, po czym wartość Pr zmniejszy się o 1. Wskaźnik Pr będzie zawsze wskazywał daną, która została wprowadzona do stosu jako ostatnia.

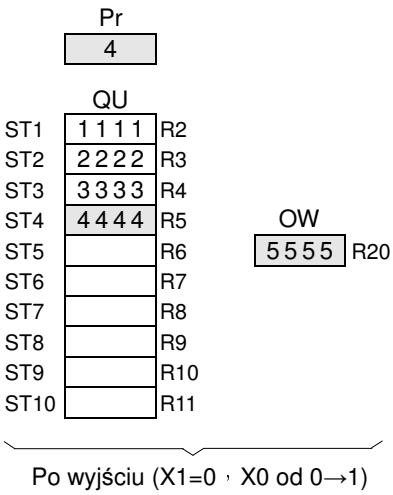
FUN111 D P STACK	STOS (STACK)	FUN111 D P STACK

- Jeżeli żadne dane nie weszły jeszcze do stosu lub wszystkie dane opuściły stos (Pr = 0), to stan wyjścia pustego stosu „EPT” zostanie ustawiony na 1. Wówczas, nawet w przypadku kolejnej operacji wyjścia danych, instrukcja ta nie zostanie wykonana. W przypadku, gdy większa ilość danych będzie wprowadzana, niż wyprowadzana ze stosu, to stos po pewnym czasie stanie się pełny (wskaźnik Pr wskaże pozycję ST_L), a stan wyjścia pełnego stosu „FUL”, zostanie ustawiony na 1. Wówczas, nawet w przypadku kolejnych operacji wprowadzania danych, funkcja nie zostanie wykonana. Wskaźnik stosu, identycznie jak wskaźnik kolejki, nie powinien być zmieniany przez inne funkcje. W przypadku aplikacji wymagającej ustawienia wartości Pr, dopuszczalny zakres dla tej wartości wynosi od 0 do L (0 oznacza pusty stos, natomiast wartości od 1 do L oznaczają odpowiednio od ST_1 do ST_L). Jeżeli zakres ten zostanie przekroczony, stan wyjścia błędu wskaźnika „ERR” zostanie ustawiony na 1, a funkcja nie zostanie wykonana.



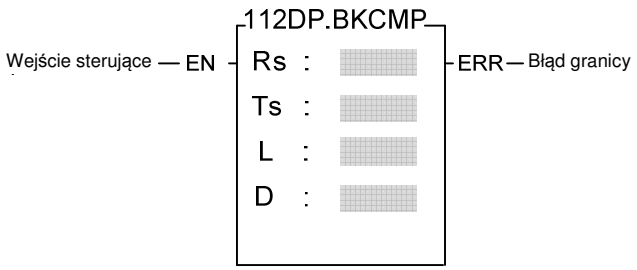
- Program po lewej zakłada, że wstępna zawartość stosu jest identyczna jak w przypadku stosu na poprzedniej stronie, Program wprowadza i wyprowadza dane ze stosu. Wyniki przedstawione zostały poniżej. Pr zawsze wskazuje na daną, która została najpóźniej wprowadzona do stosu.





FUN112 D P BKCMP	PORÓWNANIE BLOKOWE (BLOCK COMPARE（DRUM）)	FUN112 D P BKCMP

SYMBOL DRABINKOWY



Rs : Wartość do porównania; wartość stałą lub rejestr

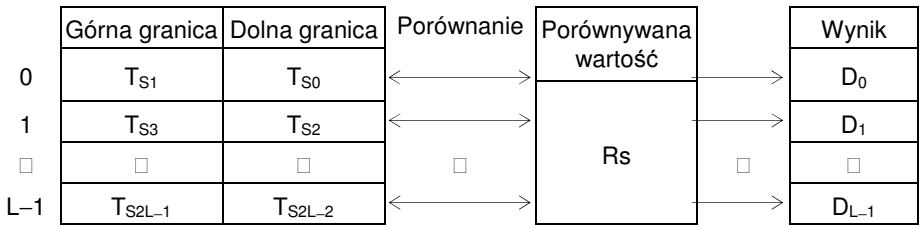
Ts : Adres początkowy tabeli do zapisu dolnych i górnych wartości granicznych

L : Liczba par górnych i dolnych granic

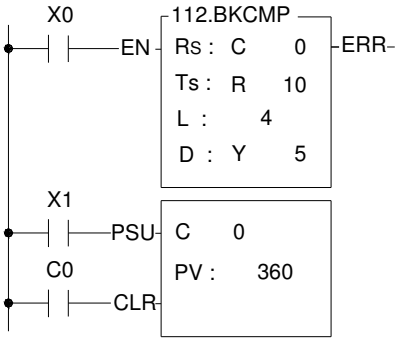
D : Adres początkowy zmiennych dyskretnych do zapisu wyników porównania

Zakres / Argument	Y	M	S	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	Y0 □ Y255	M0 □ M999	S0 □ S999	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	Liczba 16/32-bit
Rs				○	○	○	○	○	○	○	○	○	○	○	○	○
Ts				○	○	○	○	○	○	○	○	○	○	○	○	
L										○				○*	○	1~256
D	○	○	○													

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), zostanie zrealizowane porównanie wartości Rs, z kolejnymi dolnymi i górnymi granicami, tworzonymi przez L kolejnych par 16- lub 32-bitowych (modyfikator D) rejestrów, rozpoczynających się od Ts (każda sąsiadująca para 2 rejestrów od T0 tworzy górną i dolną granicę). Jeżeli wartość Rs mieści się w danym zakresie, pomiędzy górną a dolną granicą, to stan odpowiedniego bitu w D zostanie ustawiony na 1, w przeciwnym wypadku, stan ten zostanie ustawiony na 0.
- Kiedy M1975=0, to w sytuacji, gdy istnieje para, w której górna wartość graniczna jest mniejsza niż dolna wartość graniczna, stan wyjścia błędu „ERR" zostanie ustawiony na 1, a wynikiem porównania dla tej pary będzie stan 0.
- Kiedy M1975=1, to nie istnieją powyższe ograniczenia w relacji pomiędzy górną a dolną granicą. Taki przypadek można stosować w elektronicznych przełącznikach bębnowych z obrotem 360°.



- Aktualnie, funkcja ta jest przełącznikiem bębnowym, która może być wykorzystana w procedurach obsługi przerwań sprzętowych, gdzie w połączeniu z instrukcją natychmiastowego odświeżania wejść/wyjść (IMDIO), może służyć jako precyzyjny przełącznik elektroniczny.

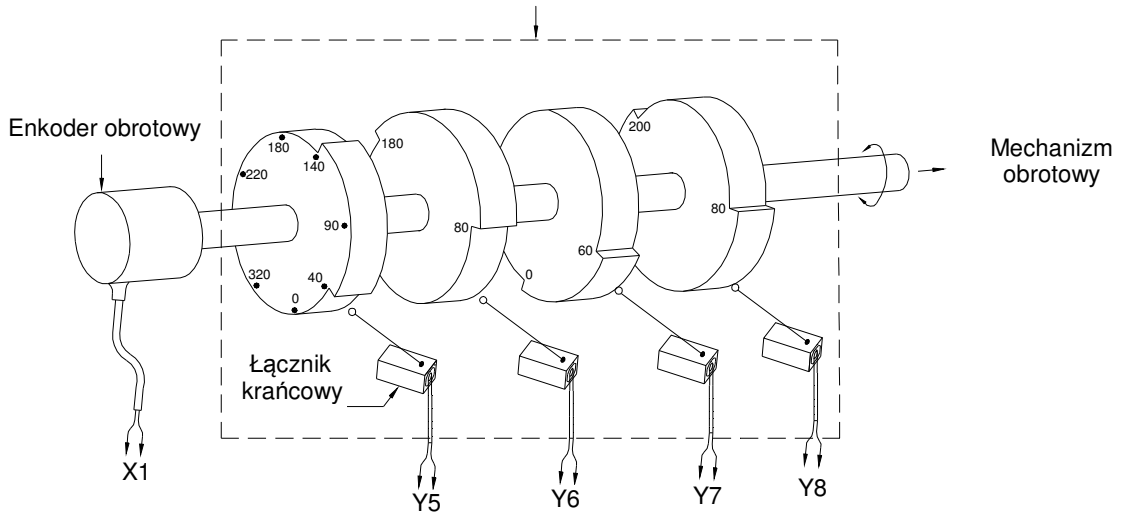


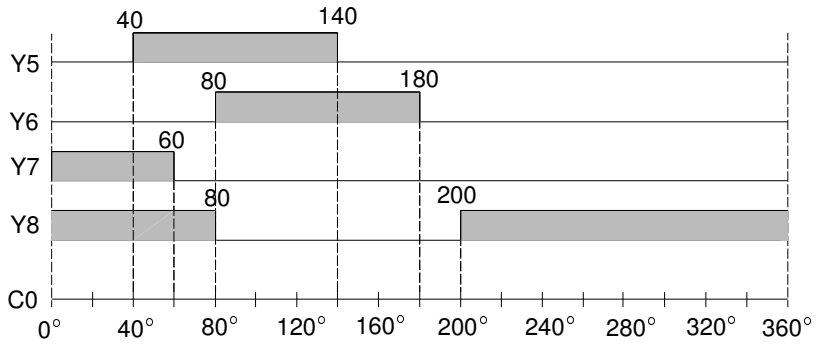
- W programie tym, C0 odpowiada kątowi obrotu (Rs) wałka. Instrukcja porównuje Rs z 4 parami (L = 4) górnych i dolnych wartości granicznych: R10,R11, R12,R13, R14,R15 i R16,R17. Wyniki porównania można uzyskać w czterech punktach wyjściowych od Y5 do Y8.
- Punkt wejściowy X1 jest detektorem kąta obrotu zamontowanym na wałku. Przy każdym pojedynczym stopniu kąta obrotu, X1 generuje impuls. Przy jednym pełnym obrocie X1 generuje 360 impulsów.

FUN112 D P BKCMP	PORÓWNANIE BLOKOWE (BLOCK COMPARE（DRUM）)	FUN112 D P BKCMP

- Program na schemacie powyżej wykorzystuje enkoder obrotowy lub inny detektor kąta obrotu (podłączony bezpośrednio do mechanizmu obrotowego), w celu stworzenia odpowiednika mechanicznego bębna (patrz mechanizm ograniczony przerywaną linia na schemacie poniżej). Poprzez regulację górnych i dolnych wartości granicznych, użytkownik może dowolnie zmieniać zakres aktywacji wyjść, dla danego kąta obrotu. Nie jest to możliwe do wykonania przy użyciu tradycyjnego mechanizmu bębnowego.

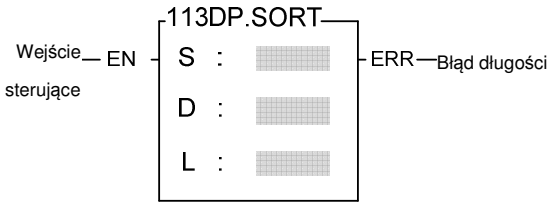
Mechanizm bębnowy imitowany przez powyższy program





FUN113 D P SORT	SORTOWANIE WARTOŚCI (DATA SORTING)	FUN113 D P SORT

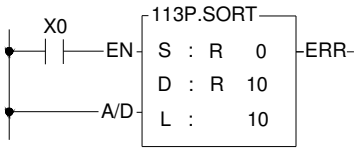
SYMBOL DRABINKOWY



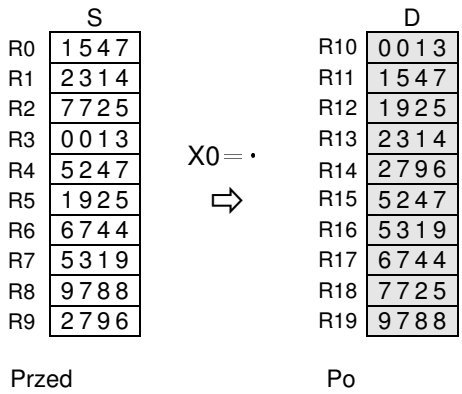
S : Adres początkowy sortowanych wartości
D : Adres początkowy do zapisu wyniku sortowania
L : Liczba kojonych rejestrów do przesortowania

Zakres / Argument	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 127
S	○	○	○	○	○	○	○	○	
D			○				○*	○	
L			○				○	○	○

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja posortuje wartości w rejestrach S w porządku rosnącym (jeżeli A/D = 1) lub malejącym (jeżeli A/D = 0) i zapisze wynik sortowania w rejestrach począwszy od D.
- Dopuszczalny zakres długości danych dla operacji sortowania L, wynosi od 2 do 127. W przypadku innej długości, wyjście „ERR" zostanie ustawiony na 1, a operacja sortowania nie zostanie przeprowadzona.

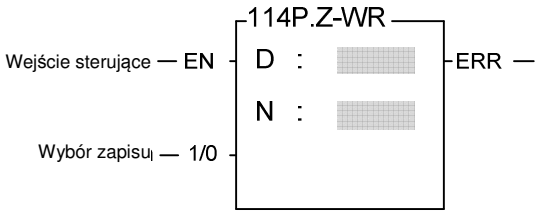


* Przykład po lewej, sortuje tabelę składającą się z rejestrów R0~R9 i zapisuje posortowane dane do R10~R19.



FUN114 D P Z-WR	ZAPIS DO STREFY (ZONE WRITE)	FUN114 D P Z-WR

SYMBOL DRABINKOWY



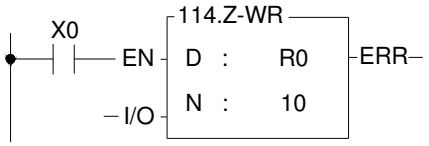
D : Adres początkowy zmiennych do ustawienia lub zresetowania

N : Liczba ustawianych lub resetowanych zmiennych, 1~511

D, N mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego (tyko przy operacjach na rejestrach)

Zakres / Argument	Y	M	S	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	Y0 □ Y255	M0 □ M1911	S0 □ S99	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095		V、Z P0～P9
D	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○
N									○				○	○	1-511	○

● Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja przeprowadzi operację ustalenia wartości w strefie zmiennych, zgodnie ze stanem wejścia 1/0. Wskazana strefa rejestrów lub bitów zostanie wyzerowana (gdy „1/0” = 0) lub ustawiona na 1 (gdy „1/0” = 1).



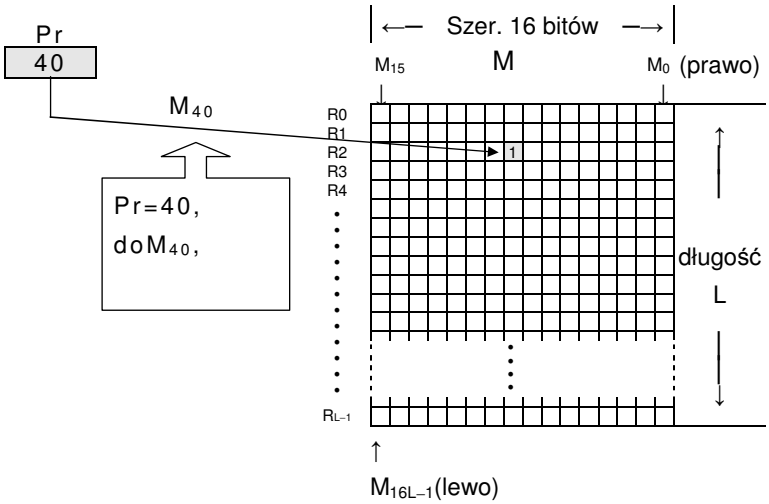
W powyższym przykładzie, gdy X0=1, rejestry R0~R9 zostaną wyzerowane.

˙ W powyższym przykładzie, gdy X0=1, bity M5~M11 zostaną wyzerowane.

Operacje na macierzach

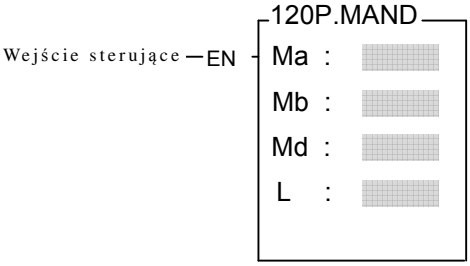
Nr fun.	Mnemonik	Funkcjonalność	Nr fun.	Mnemonik	Funkcjonalność
120	MAND	AND macierzy	126	MBRD	Odczytaj bit z macierzy
121	MOR	OR macierzy	127	MBWR	Wpisz bit do macierzy
122	MXOR	XOR macierzy	128	MBSHF	Przesuń bity macierzy
123	MXNR	XNOR macierzy	129	MBROT	Obróć bity w macierzy
124	MINV	Zaneguj macierz	130	MBCNT	Zlicz stany bitów w macierzy
125	MCMP	Porównaj macierze			

- Macierz składa się z dwóch lub więcej kolejnych rejestrów 16-bitowych. Liczba rejestrów w macierzy nazywa się długością macierzy (L). Jedna macierz zawiera łącznie L×16 bitów (punktów) i to właśnie bit jest podstawową jednostką dla każdej operacji.
- Funkcje macierzowe traktują bity macierzy 16×L jako zestaw kolejnych punktów (oznaczonych od M_0 do M_{16L-1}). Niezależnie od tego czy macierz składa się z rejestrów czy nie, przedmiotem operacji jest bit a nie wartością numeryczna.
- Funkcje macierzowe wykorzystywane są głównie do przetwarzania stanów dyskretnych, takich jak: wpisywanie, kopiowanie, porównywanie, wyszukiwanie, itp. pomiędzy pojedynczym stanem a macierzą lub pomiędzy macierzami. Funkcje te są wygodne w zastosowaniu dla pewnych, szczególnych rodzajów aplikacji.
- Większość funkcji macierzowych, wykorzystuje 16-bitowy rejestr wskaźnika, w celu określenia danego punktu w macierzy. Rejestr ten nazywany jest wskaźnikiem macierzy (Pr). Jego poprawnym zakresem jest 0 do 16L-1, który odnosi się odpowiednio do bitów M_0 do M_{16L-1} w macierzy.
- Wśród operacji macierzowych, znajdują się takie operacje jak: przesuń w prawo/lewo i obróć w prawo/lewo. Przesunięcie w kierunku starszego bitu, nazywamy kierunkiem w lewo, natomiast przesunięcie w kierunku młodszego bitu, nazywamy kierunkiem w prawo (patrz schemat poniżej).



FUN120 P MAND	AND MACIERZY (MATRIX AND)	FUN120 P MAND

SYMBOL DRABINKOWY



Ma : Adres początkowy macierzy źródłowej a

Mb : Adres początkowy macierzy źródłowej b

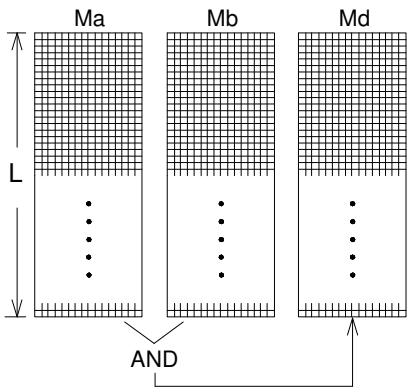
Md : Adres początkowy macierzy docelowej

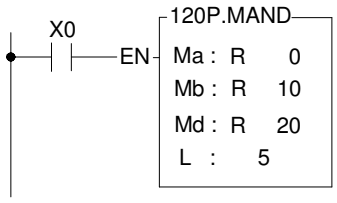
L : Długość macierzy (Ma, Mb i Md)

Ma, Mb, Md mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

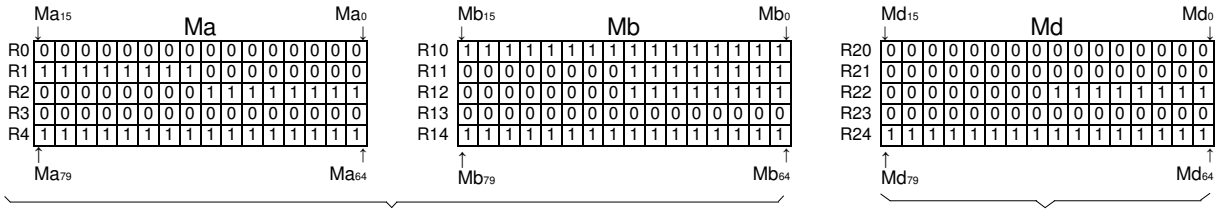
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operacje logicznego AND (jeżeli stan 2 bitów będzie 1, wynikiem będzie 1; w innym wypadku wynikiem będzie 0) pomiędzy bitami dwóch macierzy źródłowych Ma i Mb o długości L. Wynik zostanie zapisany w bitach macierzy docelowej Md, która ma tą samą długością (operacja AND realizowana jest pomiędzy bitami o tych samych pozycjach). Na przykład, jeżeli Ma_0 = 0, Mb_0 = 1, to Md_0 = 0; jeżeli Ma_1 = 1, Mb_1 = 1, to Md_1 = 1; itd., aż do wykonania operacji AND na bitach Ma_{16L-1} i Mb_{16L-1}.



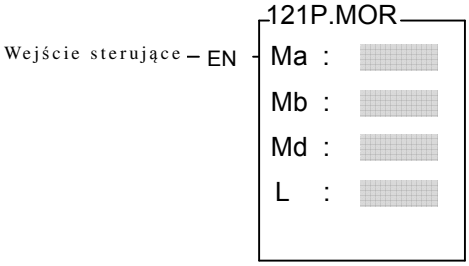


- W programie po lewej, jeżeli stan X0 zmieni się z 0→1, na bitach macierzy: Ma, stworzonej z rejestrów od R0 do R4 i macierzy Mb, stworzonej z rejestrów od R10 do R14, wykonana zostanie operacja logicznego AND. Wyniki zostaną zapisane na bitach macierzy Md ,stworzonej z rejestrów od R20 do R24. Wyniki przedstawione są na schemacie poniżej.



FUN121 P MOR	OR MACIERZY (MATRIX OR)	FUN121 P MOR

SYMBOL DRABINKOWY



Ma : Adres początkowy macierzy źródłowej a

Mb : Adres początkowy macierzy źródłowej b

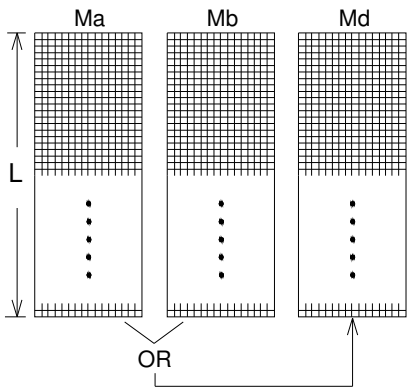
Md : Adres początkowy macierzy docelowej

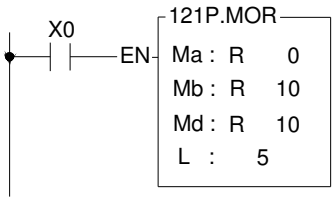
L : Długość macierzy (Ma, Mb i Md)

Ma, Mb, Md mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

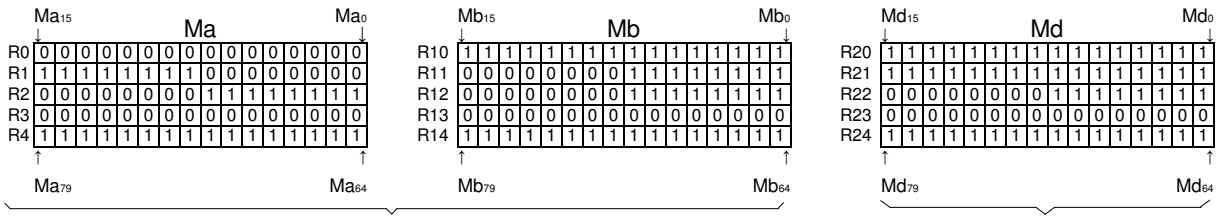
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operacje logicznego OR (jeżeli stan któregokolwiek z 2 bitów będzie 1, wynikiem będzie 1; gdy stan obu z nich będzie 0, wynikiem będzie 0) pomiędzy bitami dwóch macierzy źródłowych Ma i Mb o długości L. Wynik zostanie zapisany w bitach macierzy docelowej Md, która ma tą samą długość (operacja OR realizowana jest pomiędzy bitami o tych samych pozycjach). Na przykład, jeżeli $Ma_0 = 0$, $Mb_0 = 1$, to $Md_0 = 1$; jeżeli $Ma_1 = 0$, $Mb_1 = 0$, to $Md_1 = 0$; itd., aż do wykonania operacji OR na bitach Ma_{16L-1} i Mb_{16L-1}.



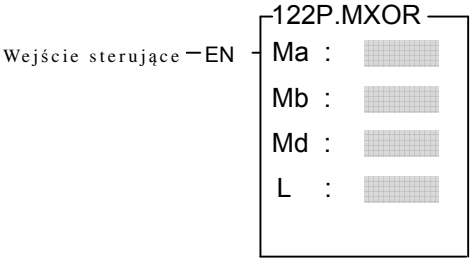


- W programie po lewej, jeżeli stan X0 zmieni się z 0→1, na bitach macierzy: Ma, stworzonej z rejestrów od R0 do R4 i macierzy Mb, stworzonej z rejestrów od R10 do R14, wykonana zostanie operacja logicznego OR. Wyniki zostaną zapisane na bitach macierzy Md ,stworzonej z rejestrów od R10 do R14. W tym przykładzie, macierz docelowa Md i macierz źródłowa Mb, są tą samą macierzą, zatem macierz Mb zostanie nadpisana wynikami operacji OR. Wyniki przedstawione są na schemacie poniżej



FUN122 P MXOR	XOR MACIERZY (MATRIX EXCLUSIVE OR（XOR）)	FUN122 P MXOR

SYMBOL DRABINKOWY



Ma : Adres początkowy macierzy źródłowej a

Mb : Adres początkowy macierzy źródłowej b

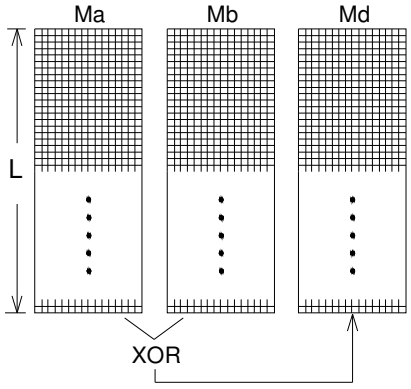
Md : Adres początkowy macierzy docelowej

L : Długość macierzy (Ma, Mb i Md)

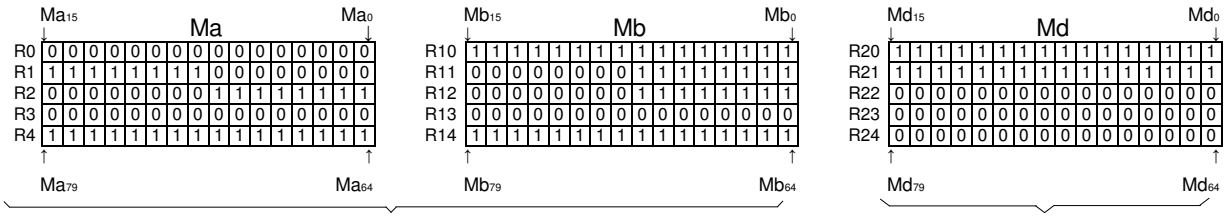
Ma, Mb, Md mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operacje logicznego XOR (jeżeli stan 2 bitów będzie różny, wynikiem będzie 1; w innym wypadku wynikiem będzie 0) pomiędzy bitami dwóch macierzy źródłowych Ma i Mb o długości L. Wynik zostanie zapisany w bitach macierzy docelowej Md, która ma tą samą długość L. Na przykład, w przypadku realizacji operacji XOR na bitach o tych samych numerach, jeżeli $Ma_0 = 0$, $Mb_0 = 1$, to $Md_0 = 1$; jeżeli $Ma_1 = 1$, $Mb_1 = 1$, to $Md_1 = 0$; itd., aż do wykonania operacji XOR na bitach Ma_{16L-1} i Mb_{16L-1}.

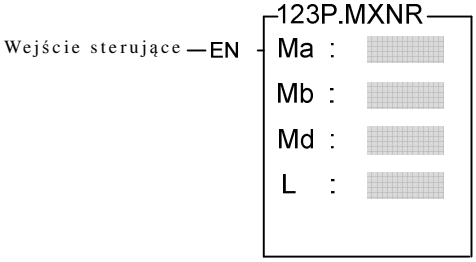


- W programie po lewej, jeżeli stan X0 zmieni się z 0→1, na bitach macierzy: Ma, stworzonej z rejestrów od R0 do R4 i macierzy Mb, stworzonej z rejestrów od R10 do R14, wykonana zostanie operacja logicznego XOR. Wyniki zostaną zapisane na bitach macierzy Md ,stworzonej z rejestrów od R20 do R24. Wyniki przedstawione są na schemacie poniżej.



FUN123 P MXNR	XNOR MACIERZY (MATRIX EXCLUSIVE NOR（XNR）)	FUN123 P MXNR

SYMBOL DRABINKOWY



Ma : Adres początkowy macierzy źródłowej a

Mb : Adres początkowy macierzy źródłowej b

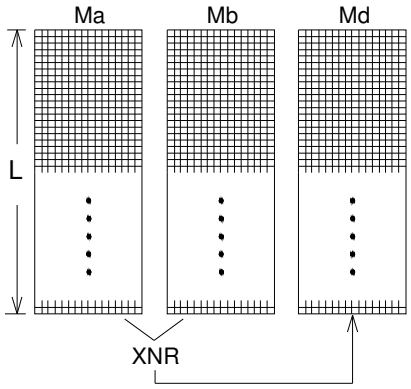
Md : Adres początkowy macierzy docelowej

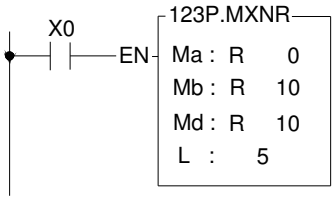
L : Długość macierzy (Ma, Mb i Md)

Ma, Mb, Md mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operacje logicznego XNOR (jeżeli status 2 bitów będzie identyczny, wynikiem będzie 1; w innym wypadku wynikiem będzie 0) pomiędzy bitami dwóch macierzy źródłowych Ma i Mb o długości L. Wynik zostanie zapisany w bitach macierzy docelowej Md, która ma tą samą długość L (operacja XNOR realizowana jest pomiędzy bitami o tych samych pozycjach). Na przykład, jeżeli $Ma_0 = 0$, $Mb_0 = 1$, to $Md_0 = 0$; jeżeli $Ma_1 = 0$, $Mb_1 = 0$, to $Md_1 = 1$; itd., aż do wykonania operacji XNOR na bitach Ma_{16L-1} i Mb_{16L-1}..

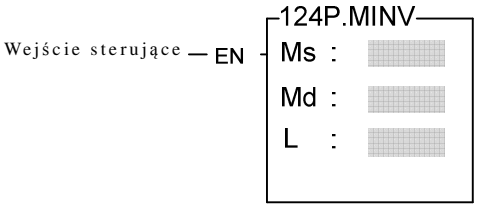




- Jeżeli „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P na bitach macierzy: Ma, stworzonej z rejestrów od R0 do R4 i macierzy Mb, stworzonej z rejestrów od R10 do R14, wykonana zostanie operacja logicznego XNOR.. Wyniki zostaną zapisane na bitach macierzy Md ,stworzonej z rejestrów od R10 do R14. Wyniki przedstawione są na schemacie poniżej.

FUN124 P MINV	ZANEGUJ MACIERZ (MATRIX INVERSE)	FUN124 P MINV

SYMBOL DRABINKOWY



Ms : Adres początkowy macierzy źródłowej

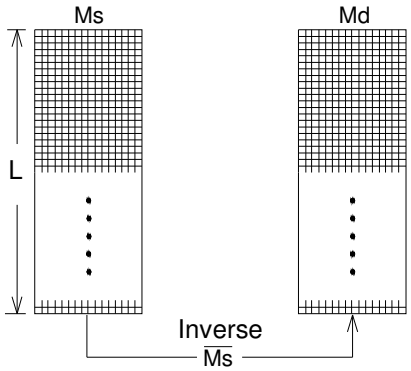
Md : Adres początkowy macierzy docelowej

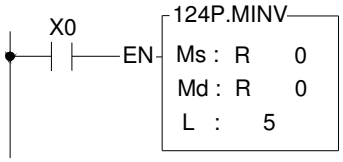
L : Długość macierzy (Ms i Md)

Ma, Md mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

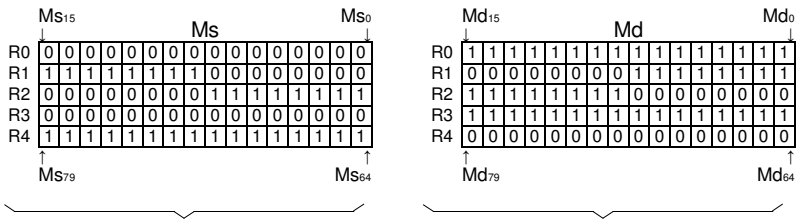
Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to sstan wszystkich bitów macierzy Ms, o długości L, zostanie zanegowany (wszystkie bity o stanie 1, zostaną zmienione na 0, natomiast bity o stanie 0, zostaną zmienione na 1). Wynik zostanie zapisany na bitach macierzy docelowej Md.





- W programie po lewej, jeżeli stan X0 zmieni się z 0→1, bity macierzy stworzonej z rejestrów od R0 do R4, zostaną zanegowane (ponieważ w przykładzie, macierze Ms i Md są tą samą macierzą). Wyniki przedstawione są na schemacie poniżej.

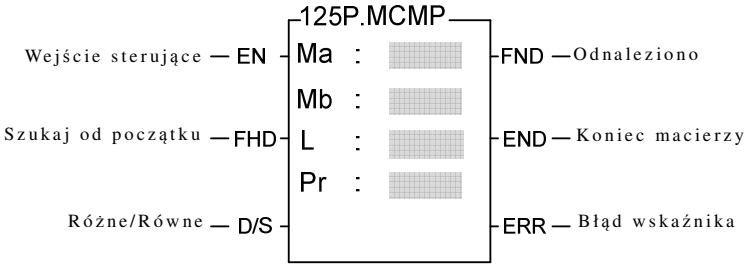


Przed wykonaniem

Po wykonaniu

FUN125 P MCMP	PORÓWNAJ MACIERZE (MATRIX COMPARE)	FUN125 P MCMP

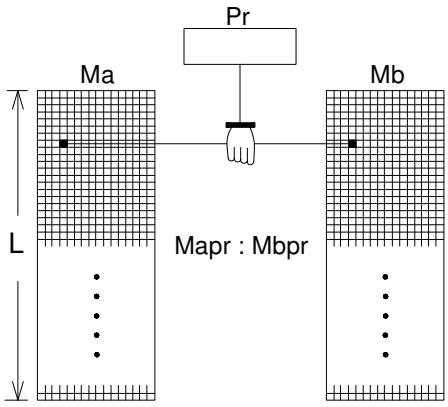
SYMBOL DRABINKOWY



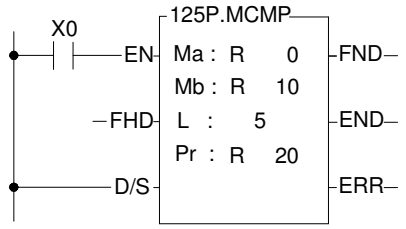
Md: Adres początkowy macierzy a
Mb: Adres początkowy macierzy b
L : Długość macierzy (Ma, Mb)
Pr : Rejestr wskaźnika
Ma, Mb mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

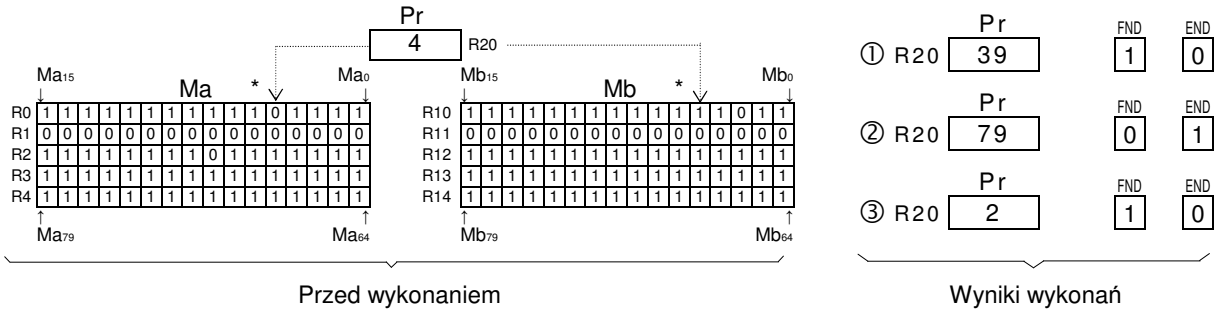
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to rozpoczynając od pierwszej pary bitów (Ma_0 i Mb_0) w macierzach Ma i Mb (kiedy „FHD" = 1 lub Pr = 16L-1) lub rozpoczynając od kolejnej pary bitów (Mapr + 1 i Mbpr + 1) wskazanej przez wskaźnik Pr (kiedy „FHD" = 0 i Pr < L-1), funkcja porówna i wyszuka pary bitów o różnych wartościach (kiedy D/S = 1) lub identycznych wartościach (kiedy D/S = 0). Po znalezieniu pary bitów, wskaźnik Pr wskaże numer bitu w macierzy, który spełnił warunek wyszukiwania. Stan wyjścia „FND" zmieni się na 1. Po przeszukaniu ostatniej pary bitów w macierzy (Ma_{16L-1}, Mb_{16L-1}), wykonywanie funkcji zostanie zakończone, niezależnie od tego, czy para bitów spełniająca warunek została odnaleziona, czy nie. Wtedy stan wyjścia „END" zmieni się na 1, a wartość Pr zostanie ustawiona na 16L-1. Przy kolejnym wywołaniu tej funkcji, wartość Pr automatycznie powróci do początkowego punktu macierzy (Pr = 0) w celu rozpoczęcia wyszukiwania od początku.



- Zakresem wskaźnika jest 0 do 16L-1. Wartość wskaźnika Pr nie powinna być zmieniana przez inne funkcje, gdyż może to wpływać na wynik wyszukiwania. Jeżeli wartość Pr przekroczy dopuszczalny zakres, to stan wyjścia „ERR" zmieni się na 1, a instrukcja nie zostanie wykonana.

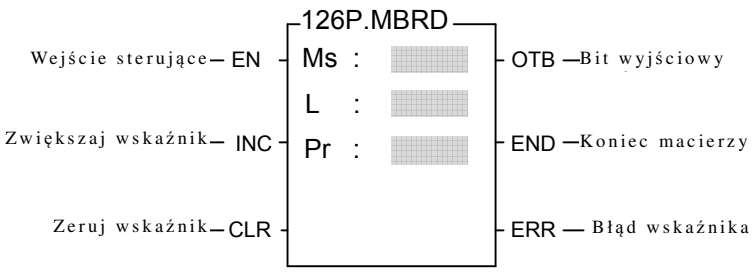


- W programie po lewej, status wejścia „FHD" = 0. Dlatego też, rozpoczynając od pozycji większej o 1 od wartości wskaźnika (oznaczonej *), funkcja przeprowadzi wyszukiwanie bitów o różnych stanach (ponieważ D/S = 1). Poniżej przedstawione są wyniki po trzykrotnej zmianie X0 z 0→1.



FUN126 P MBRD	ODCZYTAJ BIT Z MACIERZY (MATRIX BIT READ)	FUN126 P MBRD

SYMBOL DRABINKOWY



Ms : Adres początkowy macierzy

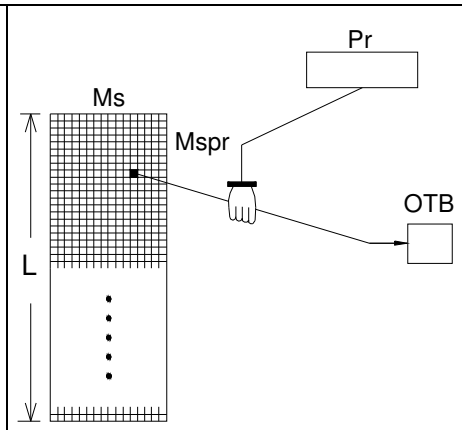
L : Długość macierzy

Pr : Rejestr wskaźnika

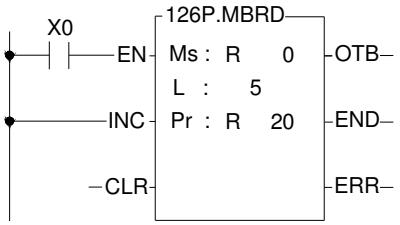
Ms może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C199	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

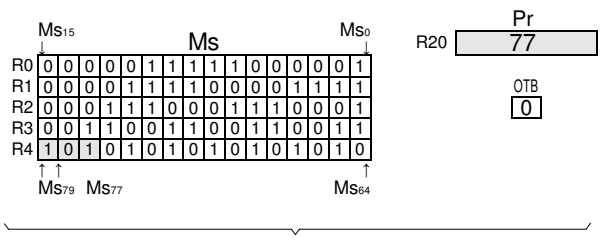
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), stan bitu Ms_{pr} w macierzy Ms, wskazywanego przez wskaźnik Pr, zostanie odczytany i pojawi się na wyjściu „OTB". Przed odczytem, instrukcja sprawdzi stan wejścia zerowania wskaźnika „CLR". Jeżeli stan CLR = 1, to wartość wskaźnika Pr zostanie wyzerowana, przed wykonaniem operacji odczytu. Po dokonaniu odczytu, sprawdzony zostanie stan wejścia „INC" i jeżeli INC = 1, wartość wskaźnika Pr zostanie zwiększona o 1. W przypadku, kiedy wartość wskaźnika osiągnie maksimum (16L-1), wyjście END zostanie ustawione w stan wysoki. Wejście CLR może działać niezależnie od pozostałych wejść.



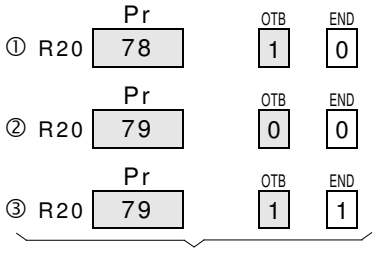
- Prawidłowym zakresem wartości wskaźnika jest 0 do 16L-1. Jeżeli wartość wskaźnika wykroczy poza zakres, stan wyjścia „ERR" zmieni się na 1, a funkcja nie zostanie wykonana.



- W programie po lewej, wejście INC = 1. Dlatego też, po każdym odczycie, wartość wskaźnika będzie zwiększana o 1. Dzięki temu, każdy bit macierzy Ms może być kolejno odczytywany (patrz schemat poniżej). Poniżej przedstawione są wyniki po trzykrotnej zmianie X0 z 0→1



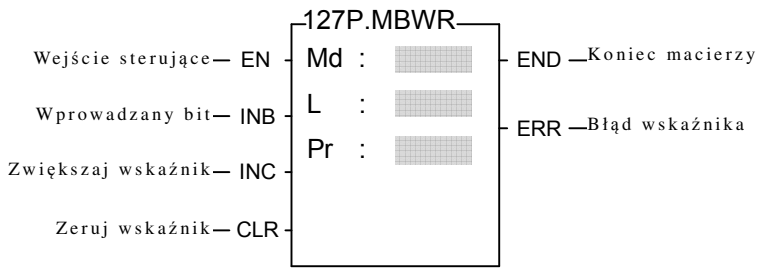
Przed wykonaniem



Wynik wykonań

FUN127 P MBWR	WPISZ BIT DO MACIERZY (MATRIX BIT WRITE)	FUN127 P MBWR

SYMBOL DRABINKOWY



Md : Adres początkowy macierzy

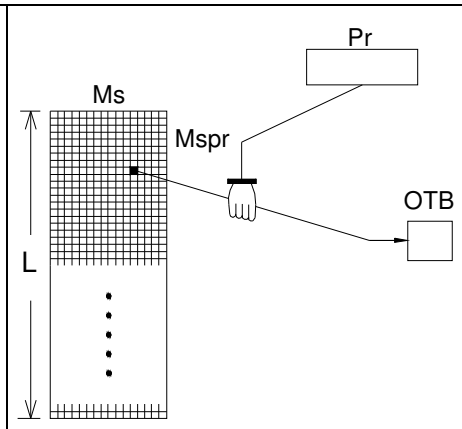
L : Długość macierzy

Pr : Rejestr wskaźnika

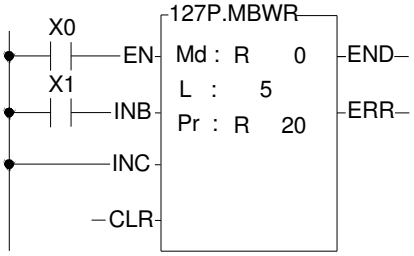
Md może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Md	○	○	○	○	○	○	○	○	○	○		○
L						○			○*	○	○	
Pr	○	○	○	○	○	○	○	○*	○*	○		

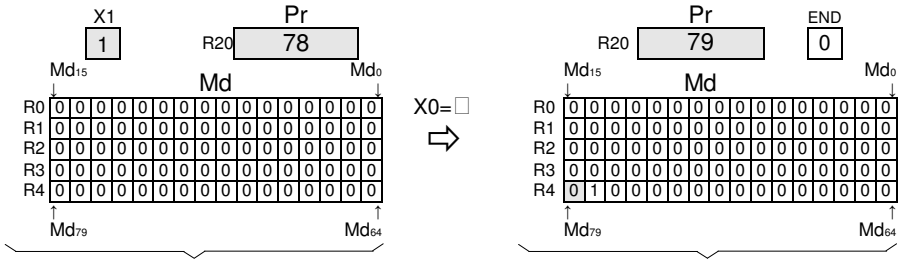
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to stan bitu „INB” zostanie zapisany w bicie Md_{pr} w macierzy Md, wskazywanym przez wartość wskaźnik Pr. Przed wpisaniem funkcja sprawdzi status wejścia zerownia wskaźnika „CLR”. Jeżeli stan „CLR” = 1, wartość wskaźnika Pr zostanie wyzerowana. Dopiero wówczas nastąpi wpis. Po wykonaniu wpisu, wartość Pr zostanie ponownie sprawdzona i jeżeli osiągnie 16L-1 (ostatni bit), to stan wyjścia END zmieni się na 1. Jeżeli wartość wskaźnika Pr będzie mniejsza od 16L-1, a stan wejścia „INC” = 1, to wartość wskaźnika zostanie zwiększona o 1. Wejście sterujące zerowaniem wskaźnika „CLR” może działać niezależnie od pozostałych wejść.



- Prawidłowym zakresem wartości wskaźnika jest 0 do 16L-1. Jeżeli wartość wskaźnika wykroczy poza zakres, stan wyjścia „ERR” zmieni się na 1, a funkcja nie zostanie wykonana.



- W programie po lewej, wartość wskaźnika będzie zwiększana po każdym wykonaniu funkcji (ponieważ „INC” = 1). Zgodnie z poniższym schematem, kiedy X0 zmieni się z 0→1, stan wejścia INB (X1) zostanie wpisany do pozycji Md_{pr} (Md_{78}), a wartość wskaźnika Pr zostanie zwiększona o 1 (zmieni się na 79). W tym wypadku, mimo, że Pr będzie wskazywać na ostatni bit, stan bitu Md_{79} nie był modyfikowany w tym wywołaniu. Dlatego też, stanem wyjścia „END” jest wciąż 0. Stan ten zmieni się na 1, po kolejnej próbie zapisu stanu w Md_{79}.

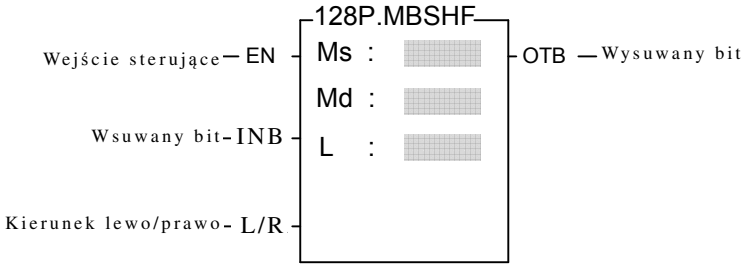


Przed wykonaniem

Wynik wykonania

FUN128 P MBSHF	PRZESUŃ BITY MACIERZY (MATRIX BIT SHIFT)	FUN128 P MBSHF

SYMBOL DRABINKOWY



Ms : Adres początkowy macierzy źródłowej

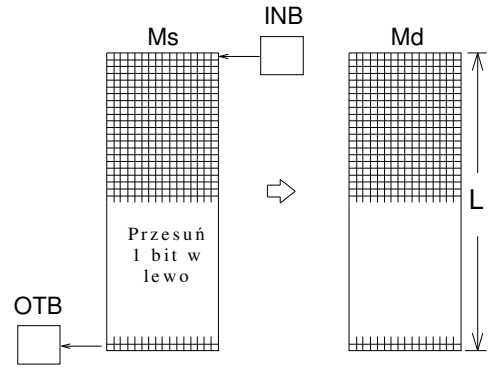
Md Adres początkowy macierzy docelowej

L : Długość macierzy (Ms i Md)

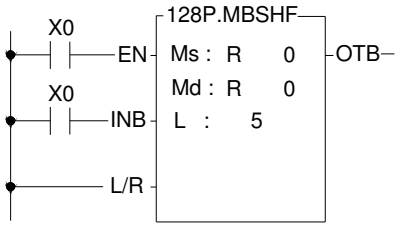
Ms, Md mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

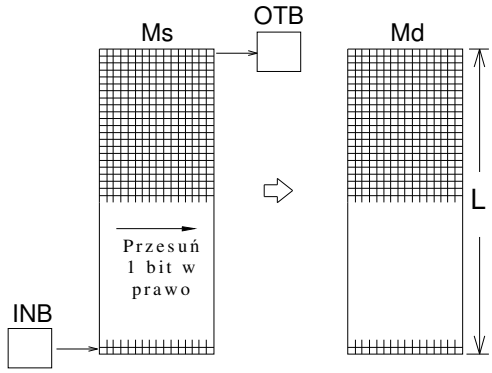
	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Argument	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L											○*	○	○	

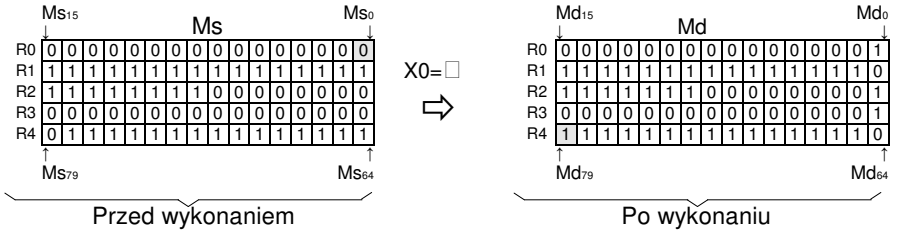
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), wszystkie bity macierz Ms zostaną przesunięte o jedną pozycję w lewo (kiedy L/R = 1) lub w prawo (kiedy L/R = 0). Wolne miejsce (przy przesunięciu w lewo będzie to M_0, a w prawo będzie to M_{16L-1}), uzupełnione zostanie stanem wejścia „INB". Stan bitu wysuniętego z macierzy (w przypadku przesunięcia w lewo - M_{16L-1}, a w prawo - M_0), pojawi się na wyjściu „OTB". Następnie, wynikowa macierz, zostanie zapisana w macierzy docelowej Md.



- Przedstawiony program jest przykładem, gdzie macierze Ms i Md są tymi samymi macierzami. Kiedy X0 zmieni się z 0→1, bity macierzy Ms, zostaną przesunięta w lewo (ponieważ L/R = 1) o 1 bit a wynik przesunięcia zostanie zapisany w Md. Wynik przedstawiony jest po prawej stronie na schemacie poniżej.

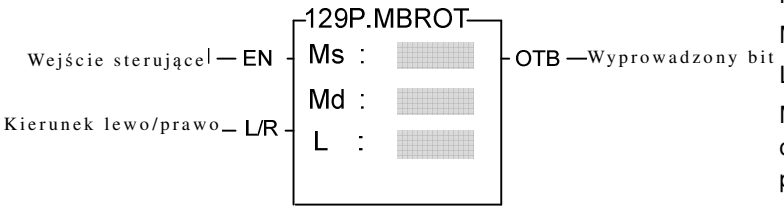






FUN129 P MBROT	OBRÓĆ BITY W MACIERZY (MATRIX BIT ROTATE)	FUN129 P MBROT

SYMBOL DRABINKOWY



Ms : Adres początkowy macierzy źródłowej

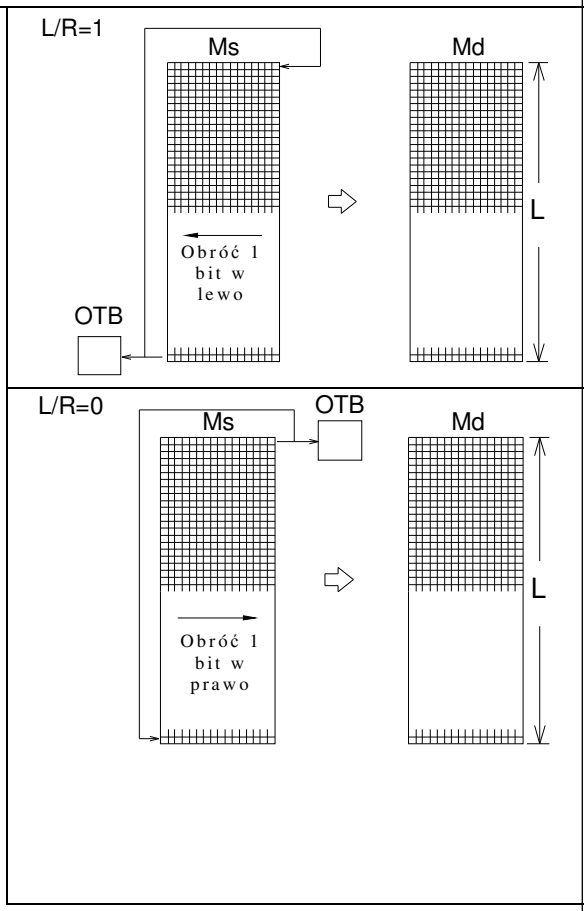
Md : Adres początkowy macierzy docelowej

L : Długość macierzy (Ms i Md)

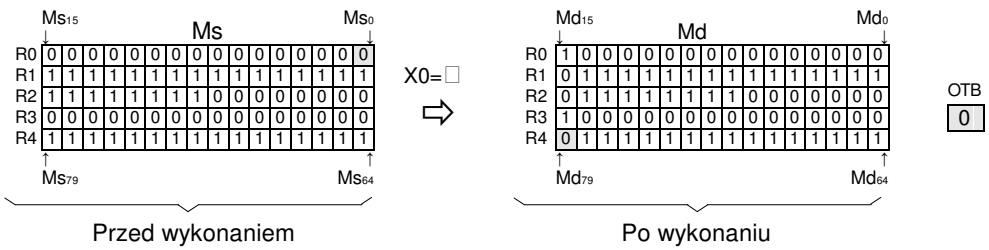
Ms, Md mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L											○*	○	○	

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to macierz Ms zostanie obrócona o jeden bit w lewo (kiedy L/R = 1) lub w prawo (kiedy L/R = 0). Wolne miejsce (przy obrocie w lewo będzie to M_0, a w prawo będzie to M_{16L-1}), wypełnione zostanie stanem wyprowadzanego bitu (w przypadku obrotu w lewo - M_{16L-1}, a w prawo - M_0). Następnie, wynikowa macierz, zostanie zapisana w macierzy docelowej Md. Wyprowadzony bit wykorzystywany jest nie tylko do wypełnienia wolnego miejsca, ale jest również wystawiany na wyjściu „OTB".



- Przedstawiony program jest przykładem, gdzie macierze Ms i Md są tymi samymi macierzami. Kiedy X0 zmieni się z 0→1, macierz Ms zostanie obrócona w prawo (ponieważ L/R = 0) o 1 bit i zapisana ponownie w Ms (ponieważ w tym przykładzie macierze Ms i Md są tymi samymi macierzami). Wynik przedstawiony jest po prawej stronie na schemacie poniżej.

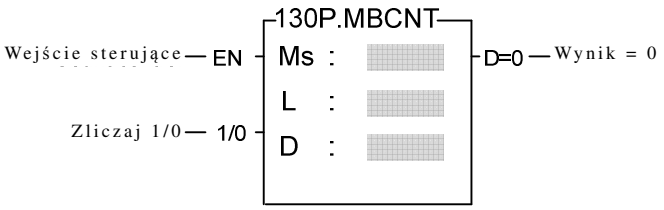


Przed wykonaniem

Po wykonaniu

FUN130 P MBCNT	ZLICZ SATNY BITÓW W MACIERZY (MATRIX BIT STATUS COUNT)	FUN130 P MBCNT

SYMBOL DRABINKOWY



Ms : Adres początkowy macierzy

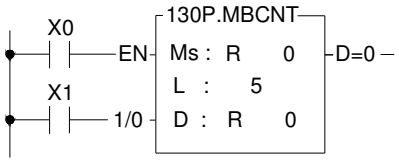
L : Długość macierzy

D : Rejestr do zapisu wyniku zliczenia

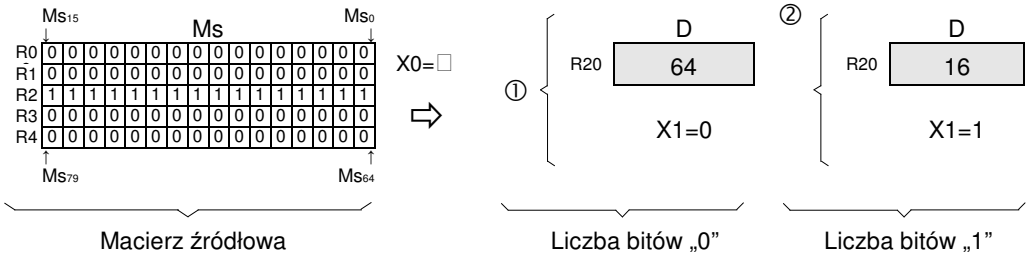
Ms może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	2 □ 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
D		○	○	○	○	○	○		○	○*	○*	○		

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), to spośród 16L bitów w macierzy Ms, funkcja zliczy całkowitą ilość bitów o stanie 1 (kiedy wejście „1/0” = 1) lub o stanie 0 (kiedy wejście „1/0” = 0). Wyniki zliczenia zostanie zapisany w rejestrze określonym adresem D. W przypadku, gdy wynik zliczania będzie równy 0, stan wyjścia „D = 0” zmieni się na 1.

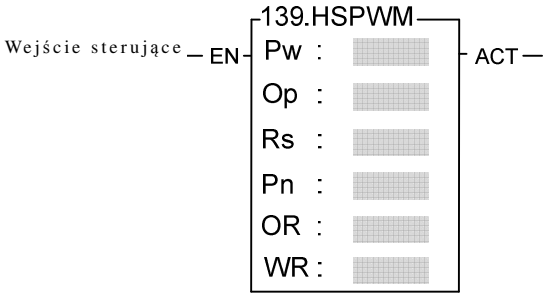


- Program po lewej, ustawia najpierw stan X1 na 0 (w celu zliczenia bitów o stanie 0), a następnie na 1 (w celu zliczenia bitów o stanie 1). W obu tych przypadkach zliczenie następuje po zmianie stanu X0 z 0→1. Wyniki przedstawione są po prawej stronie, na poniższym schemacie.



FUN 139 HSPWM	SZYBKIE WYJŚCIE Z MODULACJĄ SZERKOŚCI IMPULSU (HIGH SPEED PULSE WIDTH MODULATION OUTPUT)	FUN 139 HSPWM

SYMBOL DRABINKOWY



PW : Wyjście PWM (0 = Y0、1 = Y2、2 = Y4、3 = Y6)

Op : Polaryzacja wyjścia ; 0 = Normalna
1 = Odwrócenie wyjścia

Rs : Rozdzielczość ; 0 = 1/100 (1%)
1 = 1/1000 (0.1%)

Pn : Ustawienie częstotliwości wyjściowej (0~255)

OR : Rejestr szerokości impulsu wyjściowego (0~100 lub 0~1000)

WR : Rejestr roboczy

Zakres / Argument	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	Yn jedn. głównej	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	
Pw	○													0～3
Op														0～1
Rs														0～1
Pn		○	○	○	○	○	○	○	○	○	○	○	○	0～255
OR								○				○	○	0～1000
WR			○	○	○	○	○	○		○	○	○	○	

Opis

- Rozdzielczość (RS) i częstotliwość wyjściowa (Pn) muszą być takie same dla wyjść 0 (Y0) i (1) Y2. Oznacza to, że zarówno wyjście 0, jak wyjście 1 muszą mieć tą samą częstotliwość i rozdzielczość jej zadawania. Mogą mieć jedynie różne szerokości impulsu. Tasama zasada dotyczy wyjść 2 (Y4) i 3 (Y6).
- Jeżeli wejście sterujące „EN" = 1, na wskazane w PW wyjście, zostaną wysłane impulsy PWM. Częstotliwość wyjściowa obliczana jest według następujących wzorów:

1. $f_{pwm} = \dfrac{184320}{(P_n + 1)}$ przy RS=0, tj rozdzielczości 1/100

2. $f_{pwm} = \dfrac{18432}{(P_n + 1)}$ przy RS=1, tj rozdzielczości 1/1000

Przykład 1 : Jeżeli Pn (Nastawa częstotliwości wyjściowej) = 50 i Rs = 0(1/100), wtedy

$$f_{pwm} = \frac{184320}{(50 + 1)} = 3614.117 \approx 3.6\text{kHz}$$

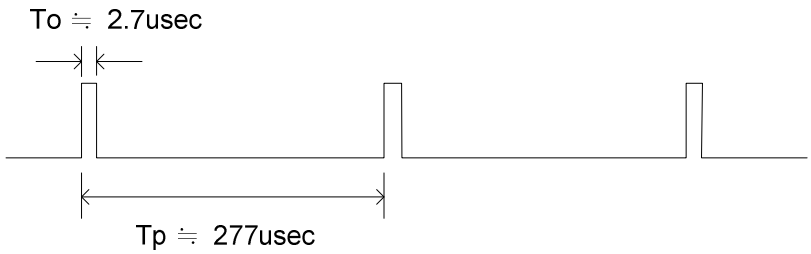
$$T(\text{Okres}) = \frac{1}{f_{pwm}} \approx 277\text{us}$$

Przy Rs = 1/100, jeżeli OR (Nastawa szerokości impulsu wyjściowego) = 1, to T0 ≒ 2.7us; jeżeli OR (Nastawa szerokości impulsu wyjściowego) = 50, to To ≒ 140us.

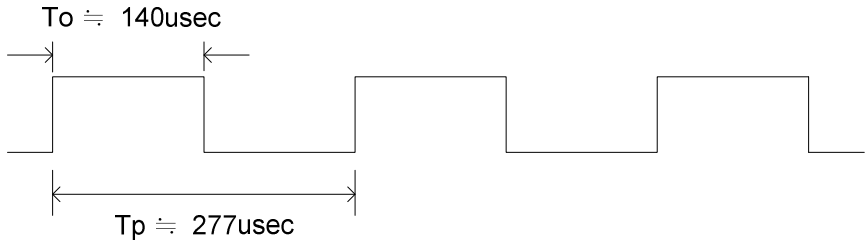
Przebieg wyjściowy :

(1).Pn (Częstotliwość wyjściowa) = 50, Rs = 0 (1/100), OR (Szerokość impulsu wyjściowego) = 1 :

FUN 139 HSPWM	HIGH SPEED PULSE WIDTH MODULATION OUTPUT	FUN 139 HSPWM



(2).Pn (Częstotliwość wyjściowa) = 50, Rs = 0 (1/100), OR (Szerokość impulsu wyjściowego) = 50 :



Przykład 2 : Jeżeli Pn (Nastawa częstotliwości wyjściowej) = 200 i Rs = 1(1/1000), wtedy

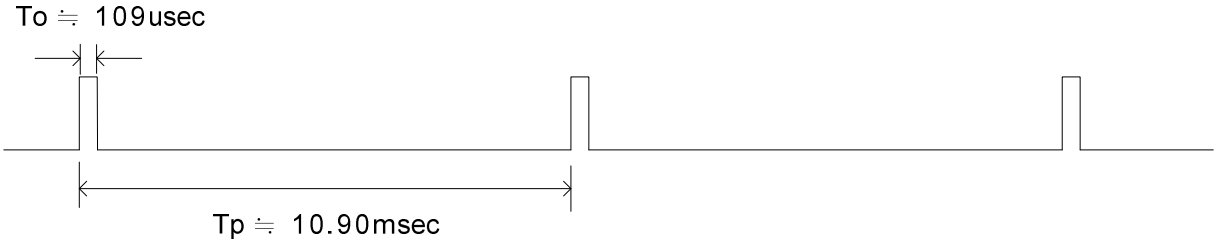
$$f_{pwm} = \frac{18432}{(200+1)} \fallingdotseq 91.7Hz$$

$$T(Okres) = \frac{1}{f_{pwm}} \fallingdotseq 10.9mS$$

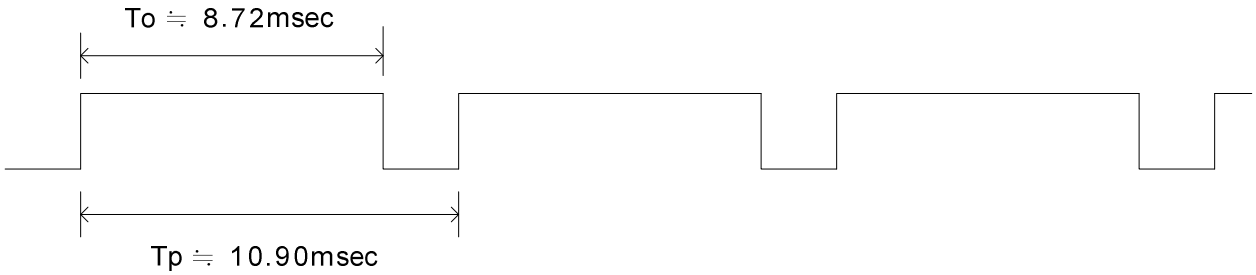
Przy Rs = 1/1000, jeżeli OR (Nastawa szerokości impulsu wyjściowego) = 10, to T0 ≒2.7us; jeżeli OR (Nastawa szerokości impulsu wyjściowego) = 800, to To ≒8.72us.

Przebieg wyjściowy:

(1).Pn (Częstotliwość wyjściowa) = 200, Rs = 1 (1/1000), OR (Szerokość impulsu wyjściowego) = 10 :

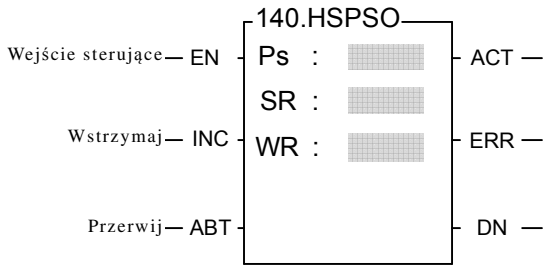


(2).Pn (Częstotliwość wyjściowa) = 200, Rs = 1 (1/1000), OR (Szerokość impulsu wyjściowego) = 800 :



FUN140 HSPSO	SZYBKIE WYJŚCIE IMPULSOWE (HIGH SPEED PULSE OUTPUT) (Skrócony opis funkcji)	FUN140 HSPSO

SYMBOL DRABINKOWY



Ps : Numer wyjścia impulsującego (0～3)
0:Y0 i Y1
1:Y2 i Y3
2:Y4 i Y5
3:Y6 i Y7

SR : Adres początkowy tabeli programu pozycjonującego.

WR : Adres początkowy rejestrów roboczych funkcji; ogółem 7 rejestrów; nie mogą być użyte w innej części programu.

Zakres / Argument	HR	DR	ROR	K
	R0 □ R3839	D0 □ D4095	R5000 □ R8071	2 □ 256
Ps				0～3
SR	○	○	○	
WR	○	○	○*	

Opisy funkcji

- Program pozycjonujący NC instrukcji HSPSO (FUN140) jest programem opisywanym i edytowanym tekstowo. Pojedynczy program pozycjonujący podzielona jest na kroki (zawierające częstotliwość wyjściową, dystans i warunki wysyłania impulsów). Dla jednej funkcji FUN140, można zaprogramować maksymalnie 250 kroków pozycjonowania. Każdy krok programu pozycjonowania wymaga 9 rejestrów. Szczegóły zastosowania opisane są w rozdziale 11 „Pozycjonowanie NC w sterownikach PLC serii FBs”.
- Plusem zapisywania programu pozycjonującego w rejestrach, jest to, że w aplikacjach stosujących interfejs MMI (man machine interface), możliwa jest modyfikacja programu z poziomu MMI. Kiedy wymagana jest zmiana programu pozycjonującego, może być ona łatwo wykonana, przez wysłanie komendy wpisu wartości do ciągu rejestrów .
- Funkcja ta nie realizuje interpolacji liniowej w pozycjonowaniu NC.
- Jeżeli wejście sterujące „EN” = 1, a wartość Ps0～3 nie jest kontrolowana przez żadną inną funkcję FUN140 (stan odpowiednio: Ps0=M1992, Ps1=M1993, Ps2=M1994 i Ps3=M1995 jest WŁ), funkcja wykona kolejny krok programu pozycjonującego (po dojściu do ostatniego kroku, nastąpi ponowny restart od pierwszego kroku); Jeżeli Ps0～3 kontrolowane są inną funkcją FUN140 (stan odpowiednio: Ps0=M1992, Ps1=M1993, Ps2=M1994 i Ps3=M1995 jest WYŁ), to funkcja wstrzyma realizację programu, do chwili zwolnienia wyjścia przez inne funkcje FUN140.
- Jeżeli „EN” =0 wysyłanie impulsów zostanie natychmiast zatrzymane.
- Jeżeli „PAU”=1 a „EN”=1 wysyłanie impulsów zostanie wstrzymane. Jeżeli „PAU” =0, a „EN”=1, nastąpi kontynuacja niezakończonej operacji wysyłania impulsów.
- Jeżeli „ABT”=1, nastąpi natychmiastowe zatrzymanie wysyłania impulsu. (Kiedy następnym razem stan „EN” zmieni się na 1, program pozycjonujący będzie realizowany od pierwszego kroku.)
- Podczas wysyłania impulsów pozycjonujących, stan na wyjściu „ACT” jest WŁ.
- W przypadku błędu w realizacji instrukcji, stan na wyjściu „ERR” zmieni się na WŁ. (Kod błędu zapisywany jest w rejestrze kodów błędów).
- Po wykonaniu wszystkich kroków programu pozycjonującego, status na wyjściu „DN” zmieni się na WŁ.

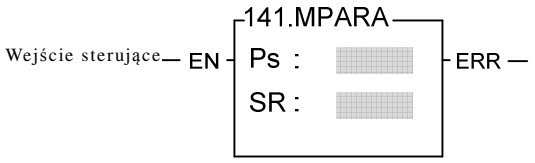
*** Aby funkcja HSPSO mogła być wykorzystywana, tryb pracy wyjścia impulsowego musi zostać skonfigurowany (w przypadku braku ustawień, Y0～Y7 będą traktowane jako normalne wyjścia) na jeden z poniższych trybów.

Tryb U/D: Y0 (Y2, Y4, Y6), jako impuls w górę.
Y1 (Y3, Y5, Y7), jako impuls w dół.
Tryb PLS/DIR: Y0 (Y2, Y4, Y6), jako wyjście impulsowe.
Y1 (Y3, Y5, Y7), jako kierunek.
Tryb A/B: Y0 (Y2, Y4, Y6), jako impuls fazy A.
Y1 (Y3, Y5, Y7), jako impuls fazy B.

- Polaryzację wyjścia impulsowego można ustawić jako normalnie WŁ lub normalnie WYŁ.
- Tryb roboczy wyjścia impulsowego może być konfigurowany za pomocą programu WINPROLADDER na stronie ustawień „Output setup”.

FUN141 MPARA	USTAW PARAMETRY POZYCJONOWANIA NC NC POSITIONING PARAMETER VALUE SETTING (skrócony opis funkcji)	FUN141 MPARA

SYMBOL DRABINKOWY



Ps : Numer wyjścia impulsującego (0～3)

SR : Adres początkowy tabeli parametrów; zawiera 18 parametrów i zajmuje 24 rejestry.

Zakres / Argument	HR	DR	ROR	K
	R0 □ R3839	D0 □ D4095	R5000 □ R8071	2 □ 256
Ps				0～3
SR	○	○	○	

Opis funkcji

- Użycie tej funkcji w programie nie jest konieczne. W przypadku, gdy domyślne wartości parametrów są zgodne z wymaganiami użytkownika, wywołanie tej funkcji nie będzie potrzebne. Jednakże jeżeli zaistnieje potrzeba dynamicznej zmiany parametrów, zmianę taką można zrealizować za pomocą funkcji MPARA.
- Funkcja ta współgra z funkcjami FUN140 lub FUN147, przy realizacji pozycjonowania NC.
- Funkcja ta zostanie wykonana, niezależnie od tego czy wejście „EN” = 0 czy 1.
- W przypadku wystąpienia błędów w wartościach parametrów zapisanych w rejestrach SR, stan na wyjściu „ERR” zmieni się na WŁ. (Kod błędu zapisywany jest w rejestrze kodów błędów).
- Szczegółowe informacje dotyczące funkcji i jej użytkowania, znajdują się w rozdziale 11 „Pozycjonowanie NC w sterownikach PLC serii FBs”.

FUN142 P PSOFF	ZATRZYMAJ SZYBKIE WYJŚCIE IMPULSOWE (STOP THE HSPSO PULSE OUTPUT) (Skrócony opis funkcji)	FUN142 P PSOFF

SYMBOL DRABINKOWY

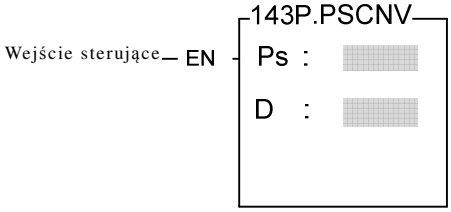
Ps : 0～3
Wskazanie numeru wyjścia impulsującego PSOn (n= Ps) do zatrzymania

Opis funkcji

- Jeżeli wejście sterujące „EN” =1 lub zmieni się z 0→1 (instrukcja P), funkcja ta, spowoduje zatrzymanie wysyłania impulsów pozycjonujących, na wyznaczonym wyjściu HSPSO (szybkim wyjściu impulsowym).
- W aplikacjach, w których poszukiwany jest mechaniczny punkt bazowania, funkcja ta może być wykorzystana do natychmiastowego zatrzymania wysyłania impulsów pozycjonujących, dzięki czemu, punkt bazowania będzie zawsze ten sam.
- Szczegółowe informacje dotyczące funkcji i jej użytkowania, znajdują się w rozdziale 11 „Pozycjonowanie NC w sterownikach PLC serii FBs”.

FUN143 P PSCNV	KONWERSJA ILOŚCI IMPULSÓW NA JEDNOSTKI MECHANICZNE CONVERT THE CURRENT PULSE VALUE TO DISPLAY VALUE (Skrócony opis funkcji)	FUN143 P PSCNV

SYMBOL DRABINKOWY



Ps : 0～3; numer wyjścia pozycjonującego, którego pozycja w impulsach ma być przekonwertowana na jednostki mechaniczne

D : Rejestr do zapisu pozycji po konwersji. Wykorzystuje 2 rejestry, np., jeżeli D=D10, oznacza to, że D10 jest młodszym słowem, a D11 jest starszym słowem.

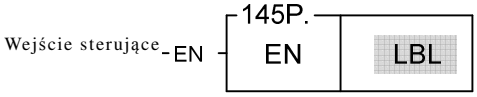
Zakres / Argument	HR	DR	ROR	K
	R0 □ R3839	D0 □ D4095	R5000 □ R8071	2 □ 256
Ps				0 ～3
D	○	○	○	

Opis funkcji

- Jeżeli wejście sterujące „EN" =1 lub zmieni się z 0→1 (instrukcja P), funkcja przekonwertuje aktualną pozycję w impulsach (PS), na odpowiadającą jej wartość w jednostkach mechanicznych (np.: mm, stopnie, cale lub PS), w celu umożliwienia wyświetlania aktualnej pozycji w tych jednostkach.
- Warunkiem uzyskania prawidłowych wyników konwersji, jest wcześniejsze wywołanie funkcji FUN141, z ustawieniem odpowiednich parametrów.
- Szczegółowe informacje dotyczące funkcji i jej użytkowania, znajdują się w rozdziale 11 „Pozycjonowanie NC w sterownikach PLC serii FBs".

FUN145 P EN	ZAŁĄCZ PRZERWANIE SPRZĘTOWE (ENABLE CONTROL OF THE INTERRUPT AND PERIPHERAL)	FUN145 P EN

SYMBOL DRABINKOWY



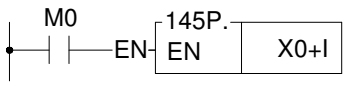
LBL : Etykieta źródła przerwania sprzętowego.

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0→1 (instrukcja P), funkcja załączy przyjmowanie przerwania sprzętowego od źródła wskazanego etykietą przerwania.
- Poniżej znajduje się lista etykiet identyfikujących przerwania sprzętowe, które można załączyć tą funkcją: (Szczegóły w rozdziale 9.3)

Etykieta LBL	Opis	Etykieta LBL	Opis	Etykieta LBL	Opis
HSTAI	Przerwanie od szybkiego licznika HSTA	X4+I	Przerwanie od narastającego zbocza X4	X10+I	Przerwanie od narastającego zbocza X10
HSC0I	Przerwanie od szybkiego licznika HSC0	X4−I	Przerwanie od opadającego zbocza X4	X10−I	Przerwanie od opadającego zbocza X10
HSC1I	Przerwanie od szybkiego licznika HSC1	X5+I	Przerwanie od narastającego zbocza X5	X11+I	Przerwanie od narastającego zbocza X11
HSC2I	Przerwanie od szybkiego licznika HSC2	X5−I	Przerwanie od opadającego zbocza X5	X11−I	Przerwanie od opadającego zbocza X11
HSC3I	Przerwanie od szybkiego licznika HSC3	X6+I	Przerwanie od narastającego zbocza X6	X12+I	Przerwanie od narastającego zbocza X12
X0+I	Przerwanie od narastającego zbocza X0	X6−I	Przerwanie od opadającego zbocza X6	X12−I	Przerwanie od opadającego zbocza X12
X0−I	Przerwanie od opadającego zbocza X0	X7+I	Przerwanie od narastającego zbocza X7	X13+I	Przerwanie od narastającego zbocza X13
X1+I	Przerwanie od narastającego zbocza X1	X7−I	Przerwanie od opadającego zbocza X7	X13−I	Przerwanie od opadającego zbocza X13
X1−I	Przerwanie od opadającego zbocza X1	X8+I	Przerwanie od narastającego zbocza X8	X14+I	Przerwanie od narastającego zbocza X14
X2+I	Przerwanie od narastającego zbocza X2	X8−I	Przerwanie od opadającego zbocza X8	X14−I	Przerwanie od opadającego zbocza X14
X2−I	Przerwanie od opadającego zbocza X2	X9+I	Przerwanie od narastającego zbocza X9	X15+I	Przerwanie od narastającego zbocza X15
X3+I	Przerwanie od narastającego zbocza X3	X9−I	Przerwanie od opadającego zbocza X9	X15−I	Przerwanie od opadającego zbocza X15
X3−I	Przerwanie od opadającego zbocza X3				

- W wielu aplikacjach, nie jest konieczne aby przerwanie sprzętowe od danego źródła, było przyjmowane w każdej chwili. Dzięki funkcją FUN146 (DIS) i FUN145 (EN), można decydować o tym, kiedy dane przerwanie będzie przyjmowane (aktywne).

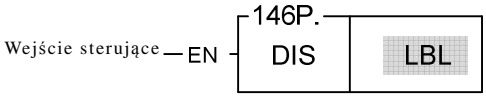
Przykład programu



- Kiedy M0 zmieni się z 0→1, załączone zostanie przyjmowanie przerwania przy zmianie X0 z 0→1. CPU szybko wykona procedurę obsługi przerwania X0+I.

FUN146 P DIS	WYŁĄCZ PRZERWANIE SPRZĘTOWE (DISABLE CONTROL OF THE INTERRUPT AND PERIPHERAL)	FUN146 P DIS

SYMBOL DRABINKOWY



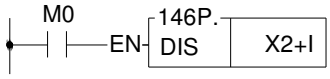
LBL : Etykieta źródła przerwania sprzętowego.

- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0→1 (instrukcja P), funkcja wyłączy przyjmowanie przerwania sprzętowego od źródła wskazanego etykietą przerwania.
- Poniżej znajduje się lista etykiet identyfikujących przerwania sprzętowe, które można wyłączyć tą funkcją: (Szczegóły w rozdziale 9.3):

Etykieta LBL	Opis	Etykieta LBL	Opis	Etykieta LBL	Opis
HSTAI	Przerwanie od szybkiego licznika HSTA	X4+I	Przerwanie od narastającego zbocza X4	X10+I	Przerwanie od narastającego zbocza X10
HSC0I	Przerwanie od szybkiego licznika HSC0	X4−I	Przerwanie od opadającego zbocza X4	X10−I	Przerwanie od opadającego zbocza X10
HSC1I	Przerwanie od szybkiego licznika HSC1	X5+I	Przerwanie od narastającego zbocza X5	X11+I	Przerwanie od narastającego zbocza X11
HSC2I	Przerwanie od szybkiego licznika HSC2	X5−I	Przerwanie od opadającego zbocza X5	X11−I	Przerwanie od opadającego zbocza X11
HSC3I	Przerwanie od szybkiego licznika HSC3	X6+I	Przerwanie od narastającego zbocza X6	X12+I	Przerwanie od narastającego zbocza X12
X0+I	Przerwanie od narastającego zbocza X0	X6−I	Przerwanie od opadającego zbocza X6	X12−I	Przerwanie od opadającego zbocza X12
X0−I	Przerwanie od opadającego zbocza X0	X7+I	Przerwanie od narastającego zbocza X7	X13+I	Przerwanie od narastającego zbocza X13
X1+I	Przerwanie od narastającego zbocza X1	X7−I	Przerwanie od opadającego zbocza X7	X13−I	Przerwanie od opadającego zbocza X13
X1−I	Przerwanie od opadającego zbocza X1	X8+I	Przerwanie od narastającego zbocza X8	X14+I	Przerwanie od narastającego zbocza X14
X2+I	Przerwanie od narastającego zbocza X2	X8−I	Przerwanie od opadającego zbocza X8	X14−I	Przerwanie od opadającego zbocza X14
X2−I	Przerwanie od opadającego zbocza X2	X9+I	Przerwanie od narastającego zbocza X9	X15+I	Przerwanie od narastającego zbocza X15
X3+I	Przerwanie od narastającego zbocza X3	X9−I	Przerwanie od opadającego zbocza X9	X15−I	Przerwanie od opadającego zbocza X15
X3−I	Przerwanie od opadającego zbocza X3				

- W wielu aplikacjach, nie jest konieczne aby przerwanie sprzętowe od danego źródła, było przyjmowane w każdej chwili. Dzięki funkcją FUN146 (DIS) i FUN145 (EN), można decydować o tym, kiedy dane przerwanie będzie przyjmowane (aktywne).

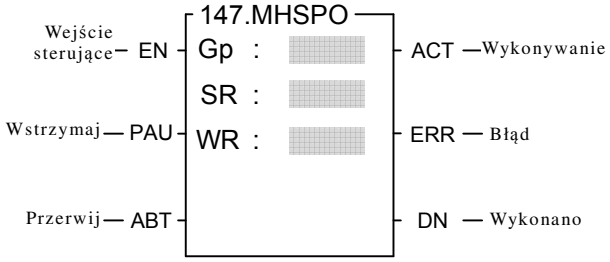
Przykład programu



- Kiedy M0 zmieni się z 0→1, wyłączone zostanie przyjmowanie przerwania przy zmianie X2 z 0→1.

FUN 147 MHSPO	WIELOOSIOWE SZYBKIE WYJŚCIE IMPULSOWE (Multi-Axis High Speed Pulse Output)	FUN 147 MHSPO

SYMBOL DRABINKOWY



Gp : Numer grupy (0～1)

SR : Adres początkowy tabeli programu pozycjonującego

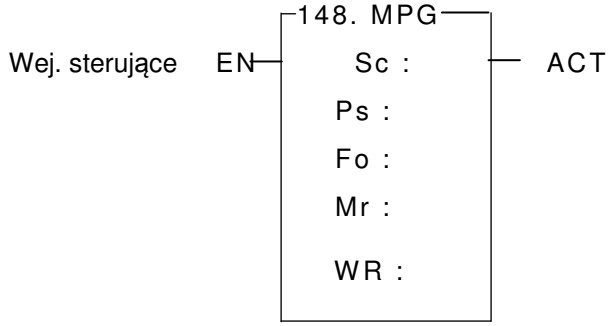
WR : Adres początkowy rejestrów roboczych funkcji; ogółem 9 rejestrów; nie mogą być użyte w innej części programu.

Zakres / Argument	HR	DR	ROR	K
	R0 □ R3839	D0 □ D3999	R5000 □ R8071	
Gp				0～1
SR	○	○	○	
WR	○	○	○*	

Opis funkcji

1. Funkcja FUN147 (MHSPO) wykorzystywana jest do obsługi interpolacji liniowej przy wieloosiowym sterowaniu ruchem. Składa się ona z programu sterowania ruchem napisanego i edytowanego tekstowo. Każdą pozycję określa się jako krok (zawierający częstotliwość wyjściową, dystans i warunki wysyłania impulsów). Każdy krok pozycjonowania zapisany jest przy pomocy 15tu rejestrów.
2. Funkcja FUN147 (MHSPO) może obsługiwać do 4 osi w celu realizacji liniowej interpolacji lub 2 grupy po 2 osie (np.: Gp0 = Osie Ps0 i Ps1 **;** Gp1 = Osie Ps2 i Ps3)
3. Plusem zapisywania programu pozycjonującego w rejestrach, jest to, że w aplikacjach stosujących interfejs MMI (man machine interface), możliwa jest modyfikacja programu z poziomu MMI. Kiedy wymagana jest zmiana programu pozycjonującego, może być ona łatwo wykonana, przez wysłanie komendy wpisu wartości do ciągu rejestrów.
4. Jeżeli wejście sterujące „EN”=1, a żadne inne funkcje FUN147/FUN140 do pozycjonowania Ps0～3 są nieaktywne (stan odpowieni: Ps0=M1992, Ps1=M1993, Ps2=M1994 i Ps3=M1995 będzie WŁ), funkcja wykona kolejny krok programu pozycjonującego (po dojściu do ostatniego kroku, nastąpi ponowny restart od pierwszego kroku); jeżeli Ps0～3 kontrolowane będą przez którąś z innych funkcji FUN147/FUN140 (stan odpowiedni: Ps0=M1992, Ps1=M1993, Ps2=M1994 i Ps3=M1995 będzie WYŁ), to funkcja wstrzyma realizację programu, do chwili zwolnienia wyjść przez inne funkcje.
5. Jeżeli „EN” =0 wysyłanie impulsów zostanie natychmiast zatrzymane.
6. Jeżeli „PAU”=1, a stan „EN” był poprzednio 1, nastąpi wstrzymanie wysyłania impulsów. Jeżeli „PAU”=0, a stanem „EN” nadal będzie 1, niezakończonej operacji wysyłania impulsów.
7. Jeżeli „ABT”=1, nastąpi natychmiastowe przerwanie wysyłania impulsów. Kiedy następnym razem stan „EN” zmieni się na 1, program pozycjonujący będzie realizowany od pierwszego kroku.
8. Podczas wysyłania impulsów pozycjonujących, stan na wyjściu „ACT” jest WŁ.
9. W przypadku zaistnienia błędu realizacji instrukcji, wskaźnik „ERR” będzie WŁ.
   (Kod błędu zapisywany jest w rejestrze kodów błędów.)
10. Po wykonaniu wszystkich kroków programu pozycjonującego, status na wyjściu „DN” zmieni się na WŁ.
11. Więcej szczegółów w rozdziale 11 „Pozycjonowanie NC w sterownikach PLC serii FBs”

FUN148 MPG	POZYCJONOWANIE MANUALNYM ZADAJNIKIEM IMPULSOWYM (MANUAL PULSE GENERATOR FOR POSITIONING)	FUN148 MPG



Sc : Źródłowy szybki licznik; 0~7
Ps : Numer osi do wysyłania impulsów; 0~3
Fo : Częstotliwość wyjściowa (2 rejestry)
Mr : Wartość mnożnika (2 rejestry)
Mr+0 : Licznik (Fa)
Mr+1 : Mianownik (Fb)
WR : Początkowy adres rejestrów roboczych; 4 rejestry.

∗ Funkcja wspierana dla OS w wersji V4.60 lub późniejszych

Argument \ Zakres	HR	ROR	DR	K
	R0 ▯ R3839	R5000 ▯ R8071	D0 ▯ D3999	16 bit
Sc	○	○	○	0~7
Ps	○	○	○	0~3
Fo	○	○	○	
Mr	○	○	○	
WR	○	○*	○	

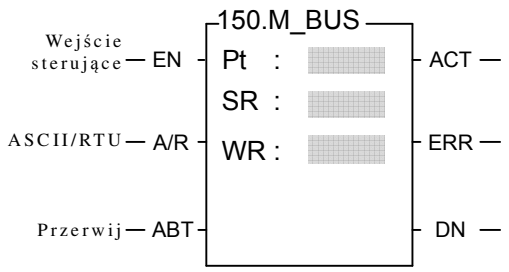
- Funkcja powinna być wywoływana wewnątrz procedury obsługi przerwania sprzętowego 50MSI, zgłaszanego w stałych odstępach czasu, co 50ms lub poprzez zastosowanie szybkiego timera 0,1ms do wygenerowania sekwencji przerwań w stałym odstępie czasu równym 50 ms. Funkcja sprawdza, ile impulsów na wejściu szybkiego licznika pojawiło się w ciągu ostatnich 50ms, następnie na podstawie wartości mnożnika (Mr+0 i Mr+1), oblicza ile impulsów należy wysłać na wskazane wyjście, w kolejnym okresie 50ms. Impulsy te są wysyłane z częstotliwością (Fo).

  Wartość częstotliwości wyjściowej (Fo) musi być wystarczająca, oraz przyspieszenie/opóźnienie (parametr 4 i parametr 8 funkcji FUN141) muszą być na tyle duże, aby w przypadku dużych mnożników (rzędu 100, 200), umożliwić wysłanie na wyjście impulsujące, wszystkich wyliczonych impulsów, w dostępnym przedziale czasowym 50ms.
- Jeżeli „EN”=1, funkcja będzie próbkować liczbę impulsów pojawiającą się na wskazanym szybkim liczniku, w każdym przedziale czasowym. Zaden impuls nie jest wysyłany, jeżeli żaden impuls nie przyszedł. Jeżeli natomiast impulsy wejściowe zostaną wykryte, nastąpi obliczenie ilości impulsów potrzebnych do wysłania, na podstawie wartości mnożnika (Mr+0 i Mr+1). Funkcja wyśle wówczas tą liczbę impulsów z szybkością (Fo), w danym przedziale czasowym.

  Ilość impulsów wyjściowych = (Ilość impulsów wejściowych × Fa) / Fb
- Funkcja jest realizowana sprzętowo. Dlatego też, nie zostanie ona wykonana w przypadku, gdy sprzęt jest zajęty.
- W trakcie wysyłania impulsów, wyjście ACT=1; w przeciwnym wypadku ACT=0.
- Więcej szczegółów w rozdziale 11 „Pozycjonowanie NC w sterownikach PLC serii FBs”.

FUN150 M-BUS	MODBUS MASTER （CZYNI ZE STEROWNIKA MASTERA W PROTOKOLE MODBUS NA PORTACH 1~4）	FUN150 M-BUS

SYMBOL DRABINKOWY



Pt ：1~4, wskazanie numeru portu komunikacyjnego, mającego działać jako ModBus master

SR：Adres początkowy programu komunikacyjnego

WR：Adres początkowy rejestrów roboczych. Funkcja kontroluje 8 rejestrów, które nie mogą być ponownie użyte w programie.

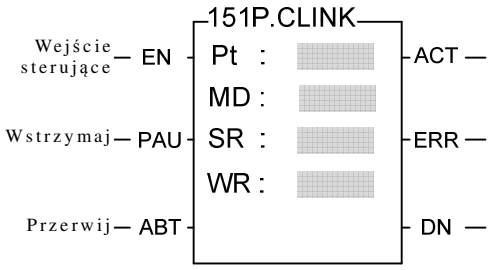
Zakres / Argument	HR	ROR	DR	K
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	
Pt				1～4
SR	○	○	○	
WR	○	○*	○	

Opis

1. Funkcja FUN150 (M-BUS) czyni ze sterownika master'a w protokole ModBus na portach 1～4. Dzięki temu można w łatwy sposób nawiązać komunikację z inteligentnymi urządzeniami peryferyjnymi, wspierającymi protokół ModBus ASCII lub RTU.
2. PLC master, może nawiązać komunikację z 247 stacjami slave, za pośrednictwem interfejsu RS-485.
3. Jedynie sterownik PLC master, wymaga zastosowania tej funkcji.
4. Funkcja wykorzystuje metodę kodowania programowego lub metodę uzupełniania tabeli, do określenia przepływu danych, czyli wskazania jaki rodzaj danych pozyskać, z której stacji slave, aby zapisać je w zmiennych sterownika master lub jaki typ danych zapisać ze sterownika master, w odpowiedniej stacji slave. Do zdefiniowania pojedynczej operacji odczytu/zapisu (pakietu), wykorzystywane jest 7 rejestrów.
5. Jeżeli „EN"=1 zmieni się z 0→1, a stan wejść „PAU" i „ABT" będzie 0, oraz Port 1/2/3/4 nie będzie wykorzystywany przez żadną inną funkcję komunikacyjną [tj. odpowiednio: M1960 (Port1) / M1962 (Port2) / M1936 (Port3) / M1938 (Port4) = 1], funkcja rozpocznie natychmiastowe wysyłanie pakietu ModBus na port 1/2/3/4 i ustawi odpowiedni stan: M1960/M1962/M1936/M1938 na 0 (co oznacza, że dany port jest zajęty). Jeżeli Port 1/2/3/4 był zajęty (M1960/M1962/M1936/M1938 = 0), to funkcja ta przejdzie w stan oczekiwania i pozostanie w nim dopóki inne funkcje komunikacyjne nie zakończą transakcji lub zatrzymają/anulują operację (M1960/M1962/M1936/M1938 =1). Wtedy funkcja ustawi stan odpowiednio: M1960/M1962/M1936/M1938 na 0 (zajmie port) i natychmiast przejdzie do transakcji danych.
6. Jeżeli podczas realizacji transakcji, stan wejścia „ABT" zmieni się na 1, funkcja natychmiast przerwie transakcję i zwolni używany przez nią port (M1960/M1962/M1936/M1938 = 1). Po kolejnym wywołaniu, funkcja rozpocznie transakcję danych od pierwszego pakietu.
7. „A/R"=0, określa protokół ModBus RTU; „A/R"=1, określa protokół ModBus ASCII
8. Podczas transakcji danych, stan na wyjściu „ACT" będzie WŁ.
9. Jeżeli po zakończeniu transakcji pakietu danych, wystąpi błąd, stany na wyjściach „DN" i „ERR" będą WŁ.
10. Jeżeli po zakończeniu transmisji pakietu danych, nie wystąpi błąd, stan na wyjściu „DN" będzie WŁ.

FUN 151 CLINK	COMMUNICATION LINK (INSTRUKCJA SPRAWIA, ŻE PLC PRACUJE JAKO STACJA MASTER W SIECI CPU LINK POPRZEZ PORT 1~4)	FUN 151 CLINK

SYMBOL DRABINKOWY



Pt : Numer portu komunikacyjnego, 1～4

MD : Tryb komunikacyjny, MD0~MD3

SR : Adres początkowy tabeli komunikacyjnej

WR : Adres początkowy tabeli rejestrów roboczych. Wykorzystywany rozmiar tabeli zależy od wybranego trybu komunikacji

Zakres / Argument	HR	ROR	DR	K
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	
Pt				1～4
MD				0～3
SR	○	○	○	
WR	○	○*	○	

Opis

● Funkcja ta działa w jednym z 4 trybów MD0～MD3, z czego trzy tryby MD0～MD2 stanowią „sieć o połączeniu standardowym”, a tryb MD3 stanowi „sieć o szybkim połączeniu”. Poniżej przedstawione są opisy poszczególnych trybów.

- MD0 : Tryb stacji master dla protokołu FATEK CPU LINK.
  Dla każdego PLC, którego program drabinkowy zawiera instrukcję FUN151, MD0 będzie stanowić stację master dla sieci FATEK CPU LINK. Stacja master PLC opierać się będzie na programie komunikacyjnym zapisanym w rejestrach danych, gdzie określane są parametry takie jak stacja docelowa, typ danych, długość danych, itp. w celu odczytu lub zapisu stacji slave za pośrednictwem komendy „protokołu komunikacyjnego FATEK FBs-PLC”. W ten sposób, nawet 254 stacje PLC, mogą wymieniać miedzy sobą dane.

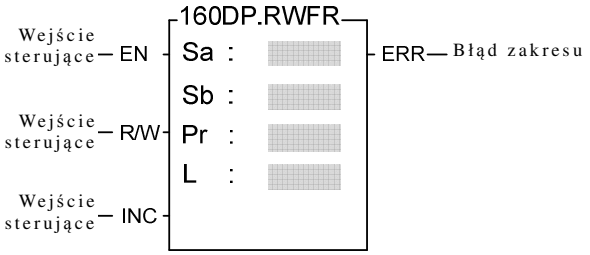
- MD1 : Tryb aktywnej transmisji danych ASCII.
  W trybie tym, funkcja FUN151 przetworzy program komunikacyjny zapisany w rejestrach danych SR i wyśle przetworzone zapytanie do urządzeń peryferyjnych ASCII (takich jak komputer, PLC innej marki, falowniki, i innych urządzeń, które mogą być sterowane za pomocą komend ASCII). Operacja może być ustawiona w trybie realizującym jedynie transmisję (1), w którym odpowiedź z urządzenia peryferyjnego jest ignorowana lub w trybie realizującym transmisje i odbierającym odpowiedź z urządzeń peryferyjnych (2). W przypadku pracy w trybie (2) użytkownik musi opierać się na protokole komunikacyjnym urządzenia peryferyjnego, w celu programowego zinterpretowania odpowiedzi.

- MD2 : Tryb pasywnego odbioru danych ASCII.
  W trybie tym, funkcja FUN151 najpierw czeka na odbiór komunikatów ASCII wysłanych przez zewnętrzne urządzenia peryferyjne ASCII (takie jak komputer, PLC innej marki, czytnik kart, czytnik kodów kreskowych, waga elektroniczna, itp., które mogą wysyłać komunikaty ASCII). Po odebraniu komunikatu, użytkownik musi opierać się na protokole komunikacyjnym urządzenia peryferyjnego, w celu przeprowadzania prawidłowej analizy i odpowiedniego reagowania. Operacja może być realizowana w trybie realizującym jedynie odbiór (1) lub w trybie realizującym odbiór i transmisję odpowiedź do urządzenia peryferyjnego (2). W przypadku trybu (2) użytkownik może skorzystać z metody uzupełniania tabeli, do stworzenia programu komunikacyjnego. Po odebraniu wiadomości, funkcja może w automatycznie przesłać komunikat zwrotny do urządzenia peryferyjnego w oparciu o stworzony program komunikacyjny.

- MD3 : Tryb stacji master w szybkim protokole wymiany danych „High Speed CPU LINK”.
  Największą różnicą pomiędzy tym trybem a MD0, jest to, że odpowiedź zwrotna w MD3 jest o wiele szybsza niż w przypadku MD0. Dzięki zastosowaniu CPU LINK w trybie MD3, FATEK PLC może w łatwy sposób zaimplementować sterowanie rozproszone i monitoring danych w czasie rzeczywistym.

FUN160 D P RWFR	ODCZYT/ZAPIS DO PLIKU REJESTRÓW (READ/WRITE FILE REGISTER)	FUN160 D P RWFR

SYMBOL DRABINKOWY



Sa: Adres początkowy rejestrów danych

Sb: Adres początkowy w pliku danych

Pr : Rejestr wskaźnika

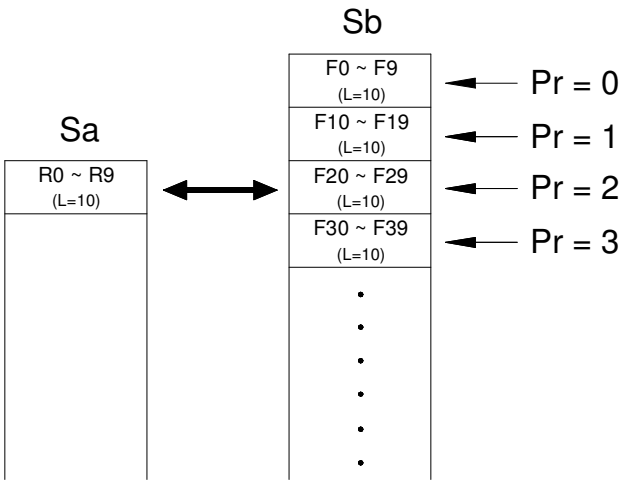
L : Rozmiar rekordu danych, 1~511

Sa może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	FR
Argument	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095		V、Z P0～P9	F0 □ F8191
Sa	○	○	○	○	○	○	○	○	○	○	○	○		○	
Sb															○
Pr		○	○	○	○	○	○		○	○*	○*	○			
L							○				○*	○	1~511		

Opis

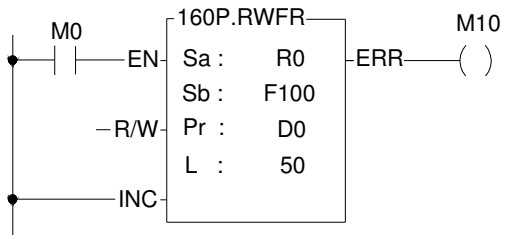
- Jeżeli wejście sterujące „EN"=1 lub zmieni się z 0→1(instrukcja P), funkcja zrealizuje operację odczytu („R/W”=1) lub zapisu („R/W”=0) rejestru plików. Podczas odczytu, zawartość rejestrów plików o adresie początkowym określonym przez Sb i wskaźnik zapisu Pr, zostanie wpisana na zawartości rejestrów danych począwszy od Sa. Podczas zapisu, zawartość rejestrów danych począwszy od Sa zostanie wpisana na zawartości rejestrów o adresie początkowym określonym przez Sa i wskaźnik zapisu Pr. L jest wielkością operacji lub rozmiarem rekordu. Dostęp do pliku rejestrów, bazuje na koncepcji rekordu. Na przykład, dla Sa=R0, Sb=F0, L=10, idea zapisu/odczytu wyglądałaby następująco:



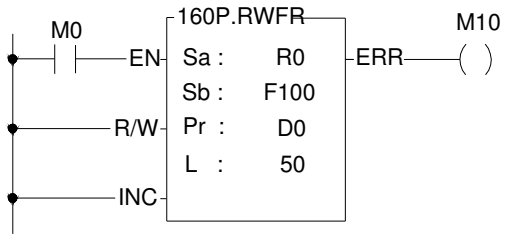
FUN160 D P RWFR	READ/WRITE FILE REGISTER	FUN160 D P RWFR

- Jest to jedyna funkcja, za pomocą której można uzyskać dostęp do rejestru plików, z poziomu programu drabinkowego.
- Jeżeli stan na wejściu „INC”=1, po wykonaniu funkcji, wartość wskaźnika zapisu zostanie zwiększona o 1.
- W przypadku nieprawidłowej wielkości zapisu (L=0 lub > 511) lub gdy przekroczony zostanie zakres rejestru danych (F0～F8191), funkcja ta nie zostanie wykonana.

.



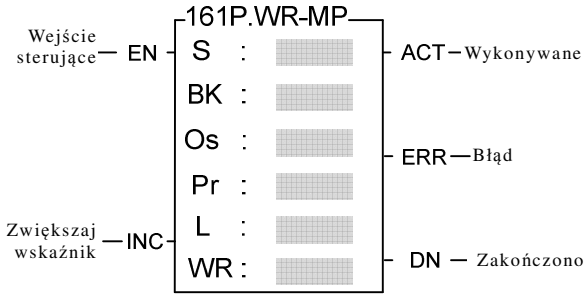
Kiedy stan M0 zmieni się z 0→1, a D0=2, na zawartości rejestrów plików F200~F249 zostanie nadpisana zawartość rejestrów danych R0~R49. Długość zapisu wynosi 50.
Po zakończeniu operacji wartość wskaźnika zapisu zostanie zwiększona o 1.



Kiedy stan M0 zmieni się z 0→1, a D0=1, na zawartości rejestrów danych R0~R49 zostanie nadpisana zawartość rejestrów plików F150~F199.
Po zakończeniu operacji wartość wskaźnika zapisu zostanie zwiększona o 1.

FUN161P WR-MP	Zapis rekordu danych do MEMORY_PACK (Write Data Record into the MEMORY_PACK)	FUN161P WR-MP

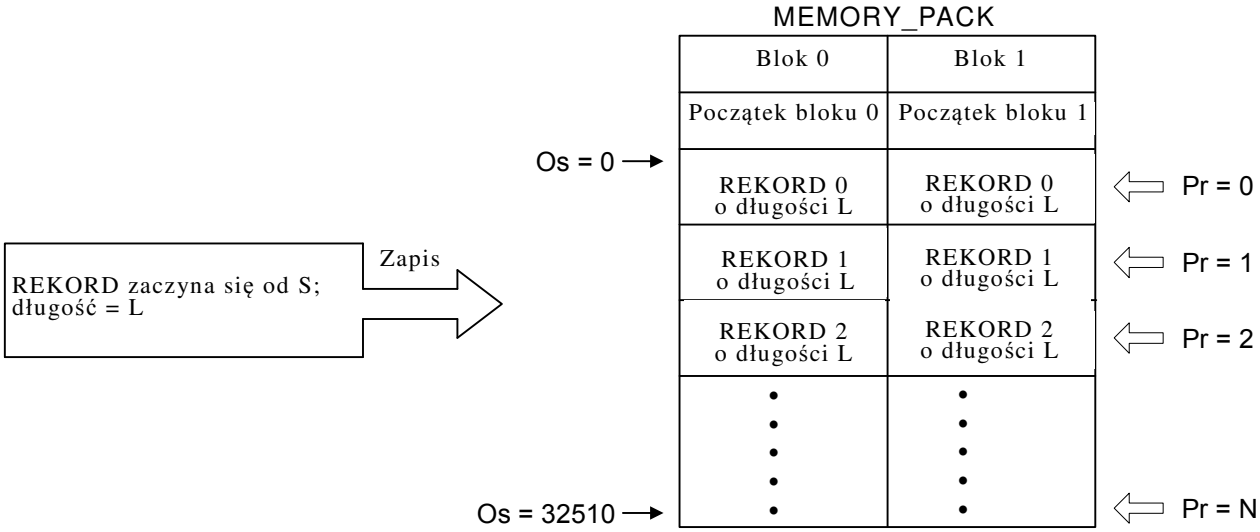
SYMBOL DRABINKOWY



S : Adres początkowy danych źródłowych
BK : Numer bloku w MEMORY_PACK, 0～1
Os : Przesunięcie w bloku
Pr : Rejestr wskaźnika
L : Wielkość zapisu, 1～128
WR : Adres początkowy rejestrów roboczych; 2 rejestry
S może łączyć się z V, Z, P0～P9 w celu zastosowania adresowania pośredniego

Argument \ Zakres	HR	ROR	DR	K	XR
	R0 ▯ R3839	R5000 ▯ R8071	D0 ▯ D4095		V、Z P0～P9
S	○	○	○		○
BK				0～1	
Os	○	○	○	0～32510	
Pr	○	○*	○		
L	○	○*	○	1～128	
WR	○	○*	○		

- Głównym zadaniem MEMORY_PACK serii FBs, jest długoterminowe przechowywanie zapisanego programu drabinkowego użytkownika. Poza tym, MEMORY_PACK może pracować w formie przenośnej, do zapisu i wczytywania parametrów roboczych maszyny, za pomocą funkcji FUN161/FUN162.
  Jeżeli wejście sterujące „EN” zmieni się z 0→1, funkcja zrealizuje zapis danych, gdzie S jest początkowym adresem danych źródłowych, BK jest numerem bloku w MEMORY_PACK do zapisu danych, Os stanowi przesunięcie w bloku, Pr jest wskaźnikiem odpowiedniego obszaru danych, a L określa wielkość zapisu. Dostęp do MEMORY_PACK, bazuje na koncepcji rekordu. Schemat działania przedstawiony został poniżej:

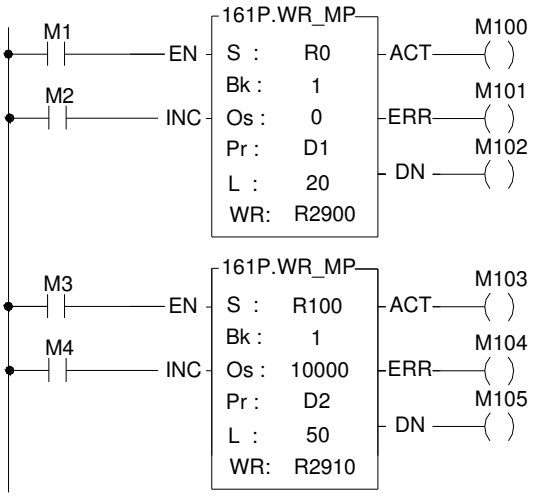


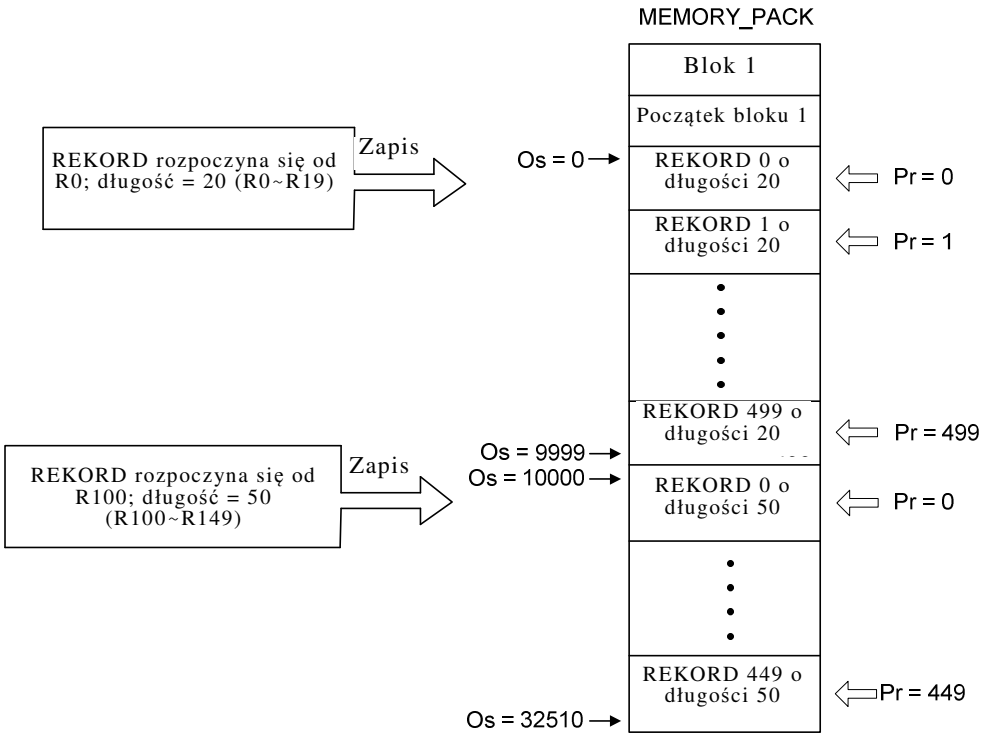
- Jeżeli wejście „INC”=1, po wykonaniu zapisu, wartość wskaźnika zostanie zwiększona o 1 – będzie on wskazywał na następny rekord.

FUN161P WR-MP	Zapis rekordu danych do MEMORY_PACK (Write Data Record into the MEMORY_PACK)	FUN161P WR-MP

- Jeżeli wartość L będzie równa 0 lub większa od 128, albo wskazywany obszar danych będzie poza zakresem, stan na wyjściu „ERR” będzie równy 1, a operacja zapisu nie zostanie wykonana.
- Do zapisu danych i weryfikacji, wymaganych jest kilka skanów PLC. W trakcie realizacji funkcji, stan wyjścia „ACT” będzie równy 1. W przypadku zakończenia realizacji oraz weryfikacji bez błędów, stan na wyjściu „DN” będzie równy 1. Jeżeli pojawią się błędy, stan wyjścia „ERR” będzie równy 1.
  MEMORY_PACK można skonfigurować do zapisu programu drabinkowego użytkownika, parametrów roboczych maszyny lub obydwu. Program drabinkowy może być zapisany tylko w bloku 0, natomiast parametry robocze maszyny mogą być zapisane w bloku 0 lub 1. Całkowita pojemność każdego bloku wykosi 32k słów.

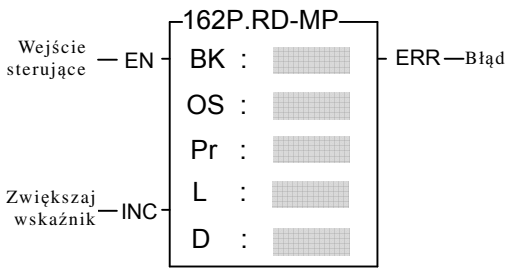
Przykładowy program : Zapisy o różnych długościach do bloku nr 1 w MEMORY_PACK





FUN162 P RD-MP	Odczyt rekordu danych z MEMORY_PACK (Read Data Record from the MEMORY_PACK)	FUN162 P RD-MP

SYMBOL DRABINKOWY

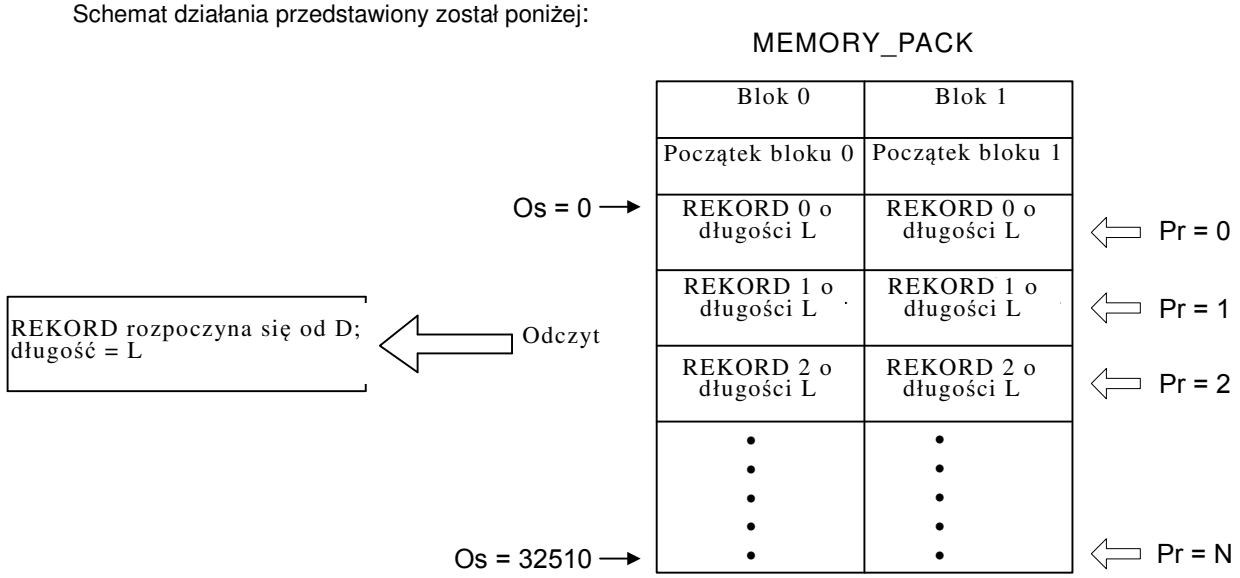


BK : Numer bloku w MEMORY_PACK, 0～1
Os : Przesunięcie w bloku
Pr : Rejestr wskaźnika
L : Wielkość odczytu, 1～128
D : Adres początkowy zapisu odczytanych wartości

Argument \ Zakres	HR	ROR	DR	K
	R0 ▯ R3839	R5000 ▯ R8071	D0 ▯ D3999	
BK				0～1
Os	○	○	○	0～32510
Pr	○	○*	○	
L	○	○*	○	1～128
D	○	○*	○	

- Jeżeli w MEMORY_PACK zapisane zostały dane, przy pomocy funkcji FUN161, to mogą być one odczytane za pomocą funkcji FUN162. Dzięki temu można skrócić czas strojenia maszyny.
- Jeżeli wejście sterujące „EN”=1 lub zmieni się z 0→1(instrukcja P), funkcja wykona odczyt danych, gdzie BK jest numerem bloku w MEMORY_PACK, z którego następuje odczyt, Os stanowi przesunięcie we wskazanym bloku, Pr jest wskaźnikiem odpowiedniego obszaru danych, L określa wielkość odczytu, a D jest początkowym adresem zapisu odczytanych wartości. Dostęp do MEMORY_PACK, bazuje na koncepcji rekordu .

  Schemat działania przedstawiony został poniżej:



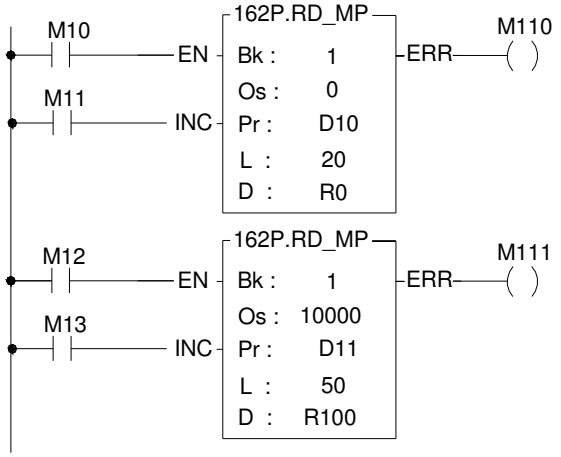
- Jeżeli wejście „INC”=1, po wykonaniu odczytu, wartość wskaźnika zostanie zwiększona o 1 – będzie on wskazywał na następny rekord.

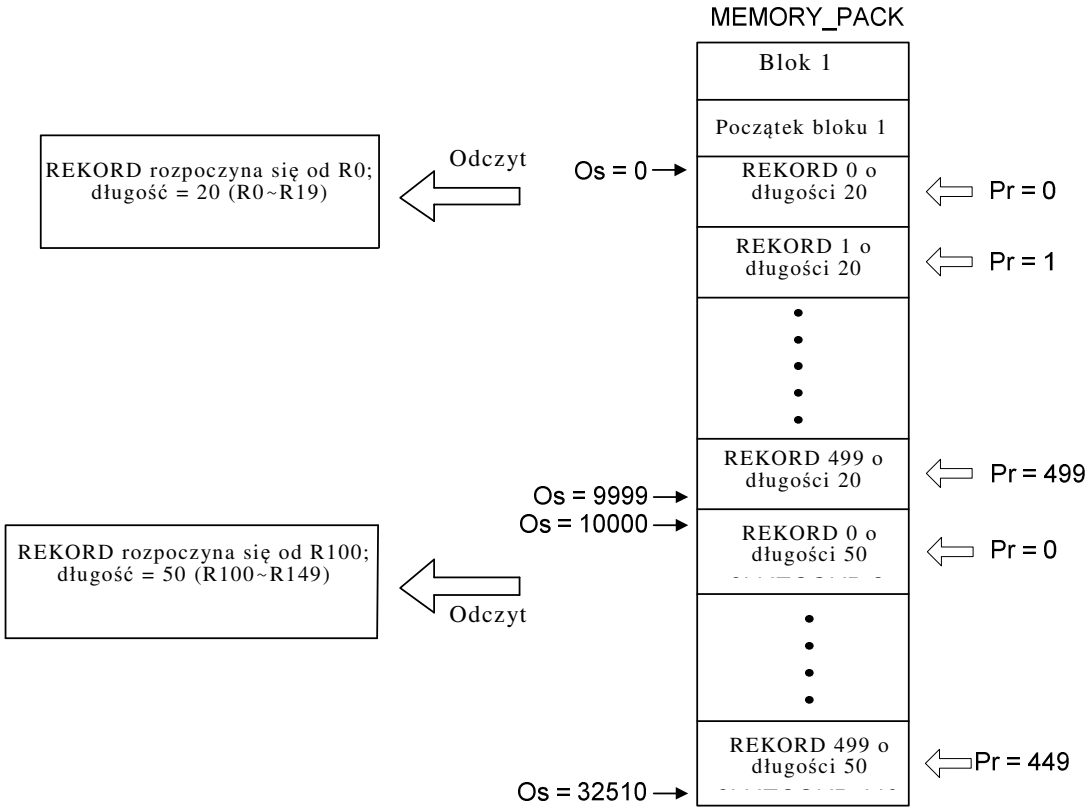
FUN162 P RD-MP	Odczyt rekordu danych z MEMORY_PACK (Read Data Record from the MEMORY_PACK)	FUN162 P RD-MP

- Jeżeli wartość L będzie równa 0 lub większa od 128, albo wskazywany obszar danych będzie poza zakresem, to stan na wyjściu „ERR” będzie równy 1, a operacja odczytu nie zostanie wykonana.
- Jeżeli MEMORY_PACK będzie pusty lub format danych będzie nieprawidłowy, a użytkownik wykorzystał FUN162 do odczytu danych, to stan wyjścia „ERR” będzie równy 1.

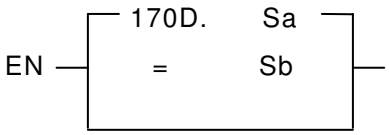
Przykład programu : Odczyty o różnych długościach z bloku nr 1 w MEMORY_PACK

※ Ważne jest, aby dane w MEMORY_PACK miały prawidłowy format. W przeciwnym wypadku, funkcje z poniższego przykładu, nie zostaną wykonane prawidłowo.





FUN170 D =	KOMPARATOR RÓWNOŚCI （Sprawdzenie czy Sa jest równe Sb） (EQUAL TO COMPARE)	FUN170 D =



Sa：Argument A lub początkowy adres Sa
Sb：Argument A lub początkowy adres Sa

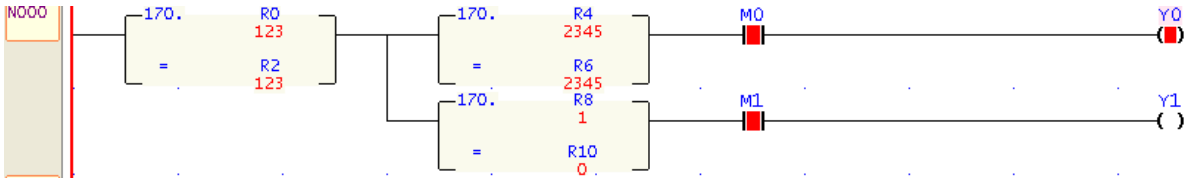
Sa i Sb mogą łączyć się z V, Z, P0～P9 w celu zastosowania adresowania pośredniego

＊Funkcja wspierana dla OS w wersji V4.60 lub późniejszych

Argument \ Zakres	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 ▯ WX240	WY0 ▯ WY240	WM0 ▯ WM1896	WS0 ▯ WS984	T0 ▯ T255	C0 ▯ C255	R0 ▯ R3839	R3804 ▯ R4167	R5000 ▯ R8071	D0 ▯ D3999	16/ 32 bitowa liczba +/-	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

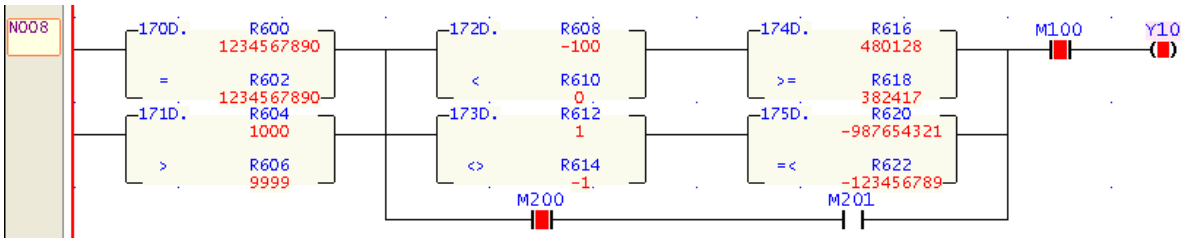
● Jeżeli „EN”=1, funkcja porówna Sa i Sb. Jeżeli Sa = Sb, to stan na wyjściu będzie równy 1; w przeciwnym wypadku 0.

Przykład 1：



Opis: Jeżeli R0=R2, R4=R6 i M0=1, to status na wyjściu Y0 będzie równy 1; w przeciwnym wypadku 0
jeżeli R0=R2, R8=R10 i M1=1, to status na wyjściu Y1 będzie równy 1; w przeciwnym wypadku 0

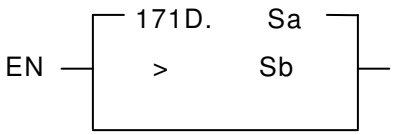
Przykład 2：



Opis: Jeżeli DR600=DR602 lub DR604>DR606, następnie DR608<DR610 i DR616≧DR618, lub DR612≠DR614 i DR620≦DR622, lub M200=1 i M201=1, a następnie M100=1, to status na wyjściu Y10 będzie równy 1; w przeciwnym wypadku 0.

Instrukcje porównywania w linii

FUN171 D >	KOMPARATOR WIĘKSZOŚCI （Sprawdzenie czy Sa jest większe od Sb） (GREATER THAN COMPARE)	FUN171 D >



Sa：Argument A lub początkowy adres Sa
Sb：Argument A lub początkowy adres Sa

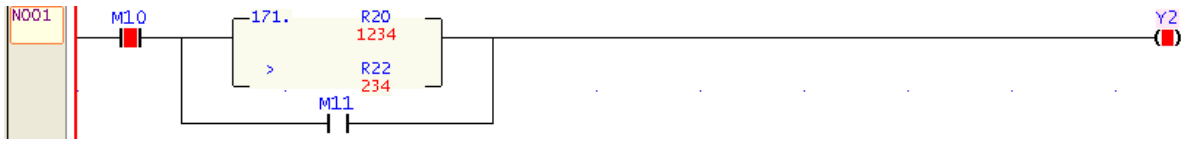
Sa i Sb mogą łączyć się z V, Z, P0～P9 w celu zastosowania adresowania pośredniego

＊Funkcja wspierana dla OS w wersji V4.60 lub późniejszych

Argument \ Zakres	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16/ 32 bitowa liczba +/-	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

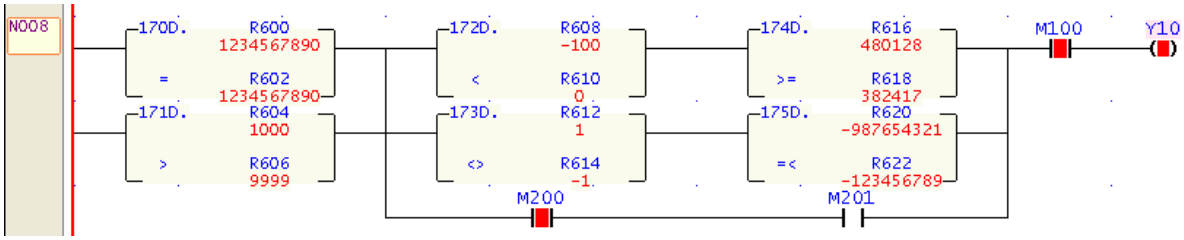
- Jeżeli „EN”=1, funkcja porówna Sa i Sb. Jeżeli Sa > Sb, stan na wyjściu będzie równy 1; w przeciwnym wypadku 0

Przykład 1：



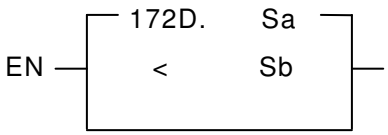
Opis: Jeżeli M10=1, R20 > R22 lub M11=1, to status na wyjściu Y2 będzie równy 1; w przeciwnym wypadku 0.

Przykład 2：



Opis: Jeżeli DR600=DR602 lub DR604>DR606, następnie DR608<DR610 i DR616≧DR618, lub DR612≠DR614 i DR620≦DR622, lub M200=1 i M201=1, a następnie M100=1, to status na wyjściu Y10 będzie równy 1; w przeciwnym wypadku 0.

FUN172 D <	KOMPARATOR MNIEJSZOŚCI (Sprawdzenie czy Sa jest niejsze od Sb) (LESS THAN COMPARE)	FUN172 D <



Sa : Argument A lub początkowy adres Sa
Sb : Argument A lub początkowy adres Sa

Sa i Sb mogą łączyć się z V, Z, P0～P9 w celu zastosowania adresowania pośredniego

＊ Funkcja wspierana dla OS w wersji V4.60 lub późniejszych

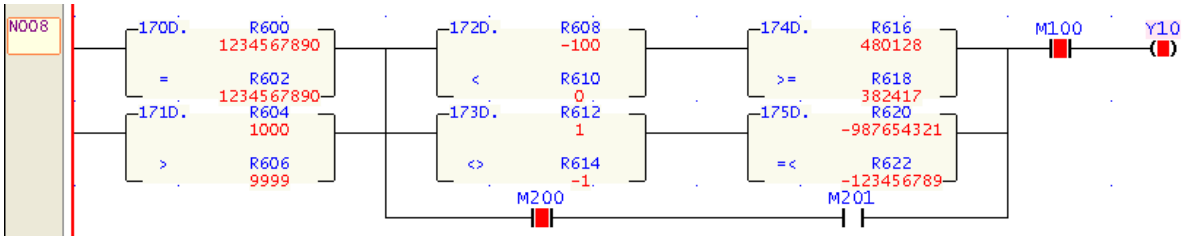
Argument \ Zakres	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 ▯ WX240	WY0 ▯ WY240	WM0 ▯ WM1896	WS0 ▯ WS984	T0 ▯ T255	C0 ▯ C255	R0 ▯ R3839	R3804 ▯ R4167	R5000 ▯ R8071	D0 ▯ D3999	16/ 32 bitowa liczba +/-	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

● Jeżeli „EN”=1, instrukcja porówna Sa i Sb. Jeżeli Sa < Sb, stan na wyjściu będzie równy 1; w przeciwnym wypadku 0

Przykład 1 :

Opis: Jeżeli M10=1, R20 < R22 lub M11=1, to status na wyjściu Y2 będzie równy 1; w przeciwnym wypadku 0

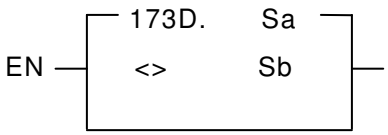
Przykład 2 :



Opis: Jeżeli DR600=DR602 lub DR604>DR606, następnie DR608<DR610 i DR616≧DR618, lub DR612≠DR614 i DR620≦DR622, lub M200=1 i M201=1, a następnie M100=1, to status na wyjściu Y10 będzie równy 1; w przeciwnym wypadku 0.

Instrukcje porównywania w linii

FUN173 D <>	KOMPARATOR NIERÓWNOŚCI (Sprawdzenie czy Sa jest różne od Sb) (NOT EQUAL TO COMPARE	FUN173 D <>



Sa : Argument A lub początkowy adres Sa
Sb : Argument A lub początkowy adres Sa

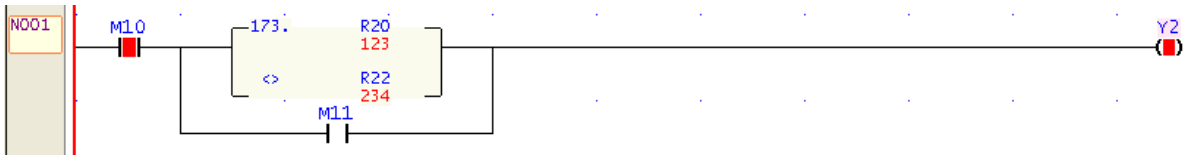
Sa i Sb mogą łączyć się z V, Z, P0～P9 w celu zastosowania adresowania pośredniego

＊ Funkcja wspierana dla OS w wersji V4.60 lub późniejszych

Argument \ Zakres	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16/ 32 bitowa liczba +/-	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

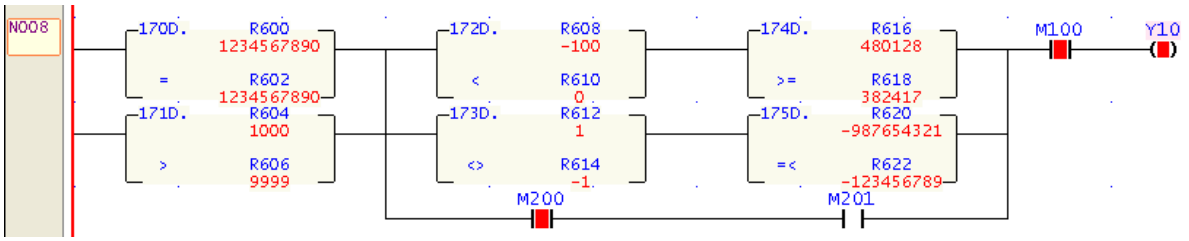
● Jeżeli „EN”=1, instrukcja porówna Sa i Sb. Jeżeli Sa ≠Sb, stan na wyjściu będzie równy 1; w przeciwnym wypadku 0

Przykład 1 :



Opis: Jeżeli M10=1, R20 ≠R22 lub M11=1, to status na wyjściu Y2 będzie równy 1; w przeciwnym wypadku 0

Przykład 2 :



Opis: Jeżeli DR600=DR602 lub DR604>DR606, następnie DR608<DR610 i DR616≧DR618, lub DR612≠DR614 i DR620≦DR622, lub M200=1 i M201=1, a następnie M100=1, to status na wyjściu Y10 będzie równy 1; w przeciwnym wypadku 0.

FUN174 D >=	KOMPARATOR WIĘKSZOŚCI LUB RÓWNOŚCI (Sprawdzenie czy Sa jest większe lub równe Sb) (GREATER THAN OR EQUAL TO COMPARE)	FUN174 D >=

```
EN  ┌ 174D.   Sa ┐
    │   >=    Sb │
    └            ┘
```

Sa : Argument A lub początkowy adres Sa
Sb : Argument A lub początkowy adres Sa

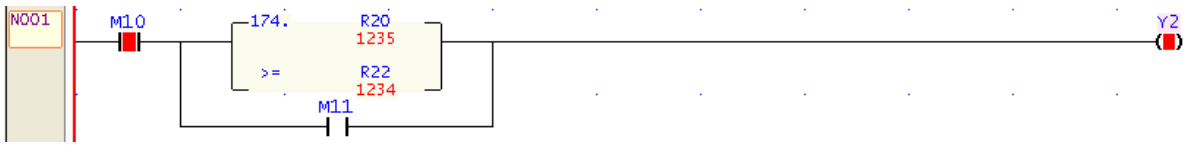
Sa i Sb mogą łączyć się z V, Z, P0～P9 w celu zastosowania adresowania pośredniego

＊ Funkcja wspierana dla OS w wersji V4.60 lub późniejszych

Argument \ Zakres	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 ▯ WX240	WY0 ▯ WY240	WM0 ▯ WM1896	WS0 ▯ WS984	T0 ▯ T255	C0 ▯ C255	R0 ▯ R3839	R3804 ▯ R4167	R5000 ▯ R8071	D0 ▯ D3999	16/ 32 bitowa liczba +/-	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

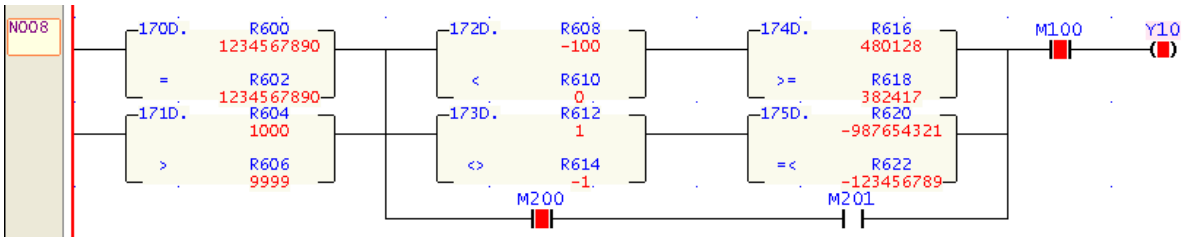
● Jeżeli „EN”=1, instrukcja porówna Sa i Sb. Jeżeli Sa ≧Sb, stan na wyjściu będzie równy 1; w przeciwnym wypadku 0

Przykład 1 :



Opis: Jeżeli M10=1, R20 ≧R22 lub M11=1, to status na wyjściu Y2 będzie równy 1; w przeciwnym wypadku 0

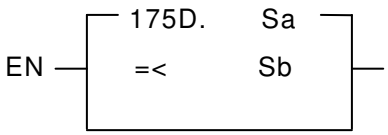
Przykład 2 :



Opis: Jeżeli DR600=DR602 lub DR604>DR606, następnie DR608<DR610 i DR616≧DR618, lub DR612≠DR614 i DR620≦DR622, lub M200=1 i M201=1, a następnie M100=1, to status na wyjściu Y10 będzie równy 1; w przeciwnym wypadku 0.

Instrukcje porównywania w linii

FUN175 D =<	KOMPARATOR MNIEJSZOŚCI LUB RÓWNOŚCI (Sprawdzenie czy Sa jest mniejsze lub równe Sb) (LESS THAN OR EQUAL TO COMPARE)	FUN175 D =<



Sa : Argument A lub początkowy adres Sa
Sb : Argument A lub początkowy adres Sa

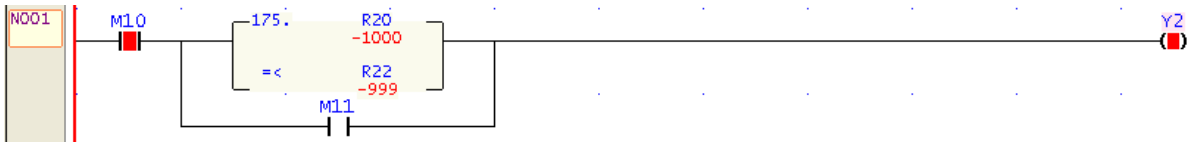
Sa i Sb mogą łączyć się z V, Z, P0～P9 w celu zastosowania adresowania pośredniego

＊ Funkcja wspierana dla OS w wersji V4.60 lub późniejszych

Argument \ Zakres	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16/ 32 bitowa liczba +/-	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

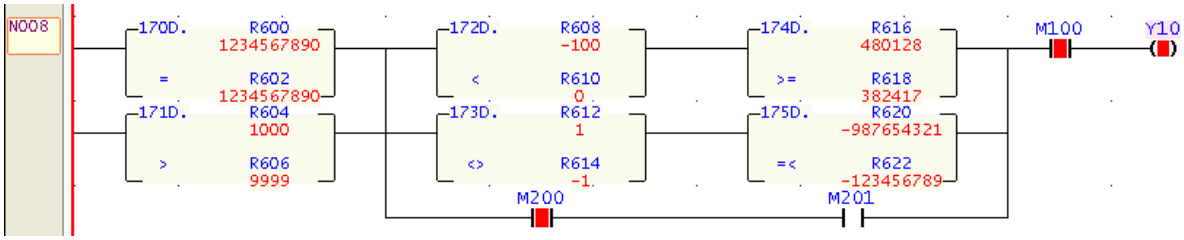
● Jeżeli „EN”=1, instrukcja porówna Sa i Sb. Jeżeli Sa ≦Sb, stan na wyjściu będzie równy 1; w przeciwnym wypadku 0

Przykład 1 :



Opis: Jeżeli M10=1, R20 ≦R22 lub M11=1, to status na wyjściu Y2 będzie równy 1; w przeciwnym wypadku 0

Przykład 2 :



Opis: Jeżeli DR600=DR602 lub DR604>DR606, następnie DR608<DR610 i DR616≧DR618, lub DR612≠DR614 i DR620≦DR622, lub M200=1 i M201=1, a następnie M100=1, to status na wyjściu Y10 będzie równy 1; w przeciwnym wypadku 0.

FUN190 STAT	ODCZYTAJ STAN SYSTEMU (READ SYSTEM STATUS)	FUN190 STAT

We. sterujące EN — 190.STAT
Gp :
D :

Gp : Grupa informacji o stanie
0 : Pobranie informacji o rozszerzeniach we/wy
1~3 : Zarezerwowane dla przyszłych zastosowań
D : Adres początkowy rejestrów do zapisu informacji o stanie systemu
D+0 : Liczba informacji
D+1 : Stan 1
…
D+N: Stan N

Argument \ Zakres	HR	ROR	DR	K
	R0 ▯ R3839	R5000 ▯ R8071	D0 ▯ D3999	
Gp				0~3
D	○	○*	○	

∗ Funkcja wspierana dla OS w wersji V4.62 lub późniejszych

● Jeżeli wejście sterujące „EN”=1, funkcja zostanie wykonana, i jeżeli Gp = 0, co oznacza żądanie uzyskania informacji o podłączonych do sterownika modułach rozszerzeń We / Wy, liczba podłączonych modułów rozszerzeń we / wy zostanie zapisana w rejestrze D, natomiast do rejestrów D+1~D+N zapisane zostaną kody poszczególnych rozszerzeń, w kolejności ich występowania w systemie. Użycie grup Gp=1~3 jest zarezerwowane na przyszłość.

Kod modułu rozszerzeń we / wy	Nazwa modułu rozszerzeń we / wy
1	FBs-8XYR
2	FBs-8X
3	FBs-8YR
4	FBs-16XYR
5	FBs-20X
6	FBs-16YR
7	FBs-24X
8	FBs-24Y
9	FBs-24XYR
10	FBs-40XYR
11	FBs-60XYR
12	FBs-7SG1S (z dekodowaniem)
13	FBs-7SG1H (bez dekodowania)
14	FBs-7SG2S (z dekodowaniem)
15	FBs-7SG2H (bez dekodowania)
16	FBs-6AD
17	FBs-2DA
18	FBs-4DA
19	FBs-4PT
20	FBs-4A2D

Kod modułu rozszerzeń we / wy	Nazwa modułu rozszerzeń we / wy
21	FBs-6TC
22	FBs-6RTD
23	FBs-16TC
24	FBs-16RTD
25	FBs-2TC
26	FBs-2A4TC
27	FBs-2A4RTD
28	FBs-6NTC
29	FBs-16NTC (zarezerwowany)
30	FBs-32DGI
31	FBs-VOM
32	FBs-1LC

FUN190 STAT	ODCZYTAJ STAN SYSTEMU (READ SYSTEM STATUS)	FUN190 STAT

Przykład : W pewnym, systemie zainstalowane są dwa moduły rozszerzeń we / wy FBs-2DA + FBs-6AD

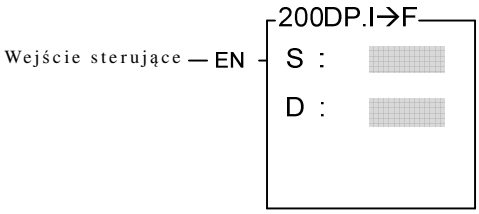
Status Monitoring

Ref. No.	Status	Data	Ref. No.	Status	Data
M500	Enable	ON			
D200	Decimal	2			
D201	Decimal	17			
D202	Decimal	16			

Opis: Jeżeli M500 = 1, funkcja zostanie wykonana. W rejestrze D200 zostanie zapisana całkowita liczba modułów rozszerzeń we/wy, jakie zostały wykryte przez jednostkę główną, w rejestrze D201 zostanie zapisany kod pierwszego modułu rozszerzeń (17=FBs-2DA), natomiast w rejestrze D202 zostanie zapisany kod drugiego modułu rozszerzeń we / wy (16=FBs-6AD).

FUN200 D P I→F	KONWERSJA FORMATU CAŁKOWITEGO NA ZMIENNOPRZECINKOWY (CONVERSION OF INTEGER TO FLOATING POINT NUMBER)	FUN200 D P I→F

SYMBOL DRABINKOWY



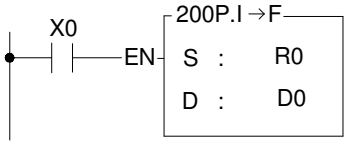
S : Adres początkowy liczby w formacie całkowitym do przekonwertowania

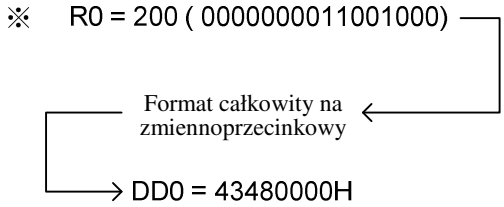
D : Adres początkowy rejestru do zapisu wyniku konwersji

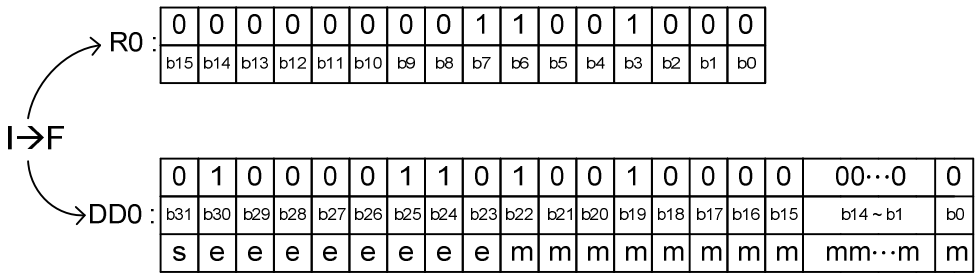
Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	16/32 bitowa liczba całkowita	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja wartości S, zapisanej w formacie całkowitym, na wartość zapisaną w formacie zmiennoprzecinkowym. Wynik konwersji zapisany zostanie w 32-bitowym rejestrze D~D+1.

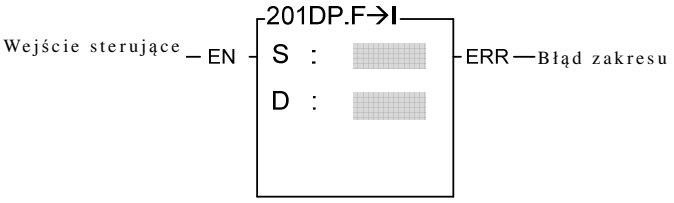






FUN201 D P F→I	KONWERSJA FORMATU ZMIENNOPRZECINKOWEGO NA CAŁKOWITY (CONVERSION OF FLOATING POINT NUMBER TO INTEGER)	FUN201 D P F→I

SYMBOL DRABINKOWY



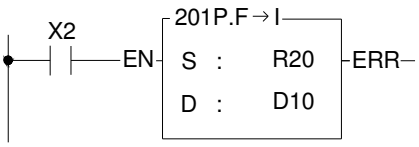
S : Adres początkowy liczby w formacie zmiennoprzecinkowym do przekonwertowania

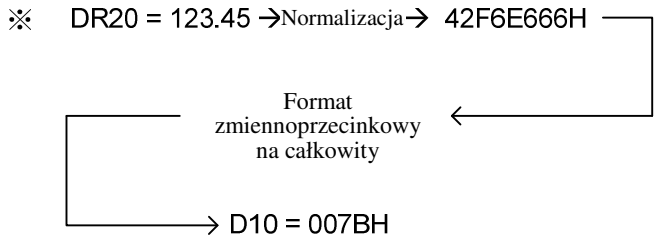
D : Adres początkowy rejestru do zapisu wyniku konwersji

Zakres / Argument	HR	ROR	DR	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	V、Z P0～P9
S	○	○	○	○
D	○	○*	○	○

Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), nastąpi konwersja 32-bitowej wartości S~S+1, zapisanej w formacie zmiennoprzecinkowym, na wartość zapisaną w formacie całkowitym. Wynik konwersji zapisany zostanie rejestrze D.
- Jeżeli wartość wyniku konwersji, przekroczy zakres zmiennej D, funkcja nie zostanie wykonana a stan wyjścia błędu „ERR” zmieni się na 1, natomiast wartość w rejestrze D pozostanie niezmieniona.





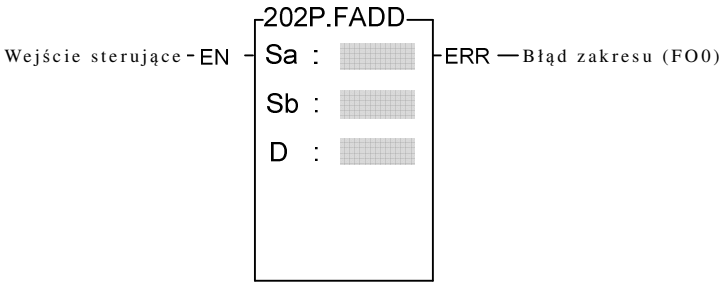
DR20:	0	1	0	0	0	0	1	0	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m

F → I

D10:	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

FUN202 P FADD	DODAWANIE LICZB ZMIENNOPRZECINKOWYCH (FLOATING POINT NUMBER ADDITION)	FUN202 P FADD

SYMBOL DRABINKOWY



Sa: Składnik a

Sb: Składnik b

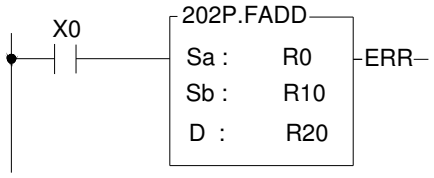
D : Adres rejestru przechowującego wynik dodawania

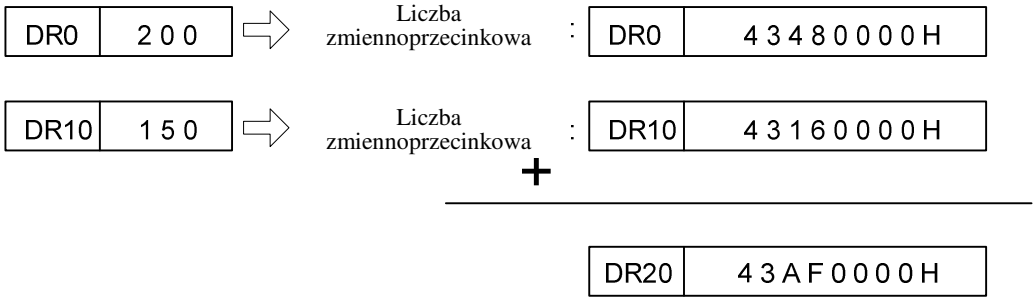
Sa, Sb, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	Liczba zmienno przecinkowa	V、Z P0～P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○		○

Opis

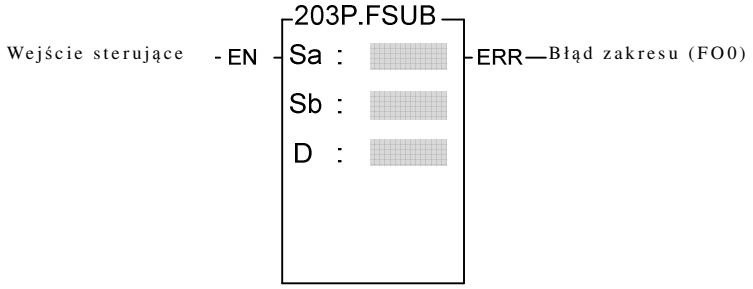
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operację dodawania danych określonych w Sa i Sb i zapisze wyniki w rejestrze D. W przypadku, gdy wynik przekroczy dopuszczalny zakres wartości w jakim wyrażona może być liczba zmiennoprzecinkowa ($\pm 3.4*10^{38}$), stan wyjścia błędu ERR zmieni się na 1, a zawartość rejestru D pozostanie niezmieniona.





FUN 203 P FSUB	ODEJMOWANIE LICZB ZMIENNOPRZECINKOWYCH (FLOATING POINT NUMBER SUBTRACTION)	FUN 203 P FSUB

SYMBOL DRABINKOWY



Sa: Odjemna

Sb: Odjemnik

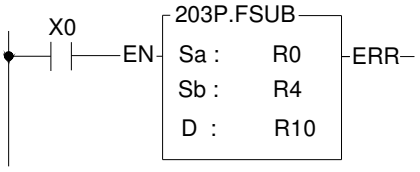
D : Adres rejestru przechowującego wynik odejmowania

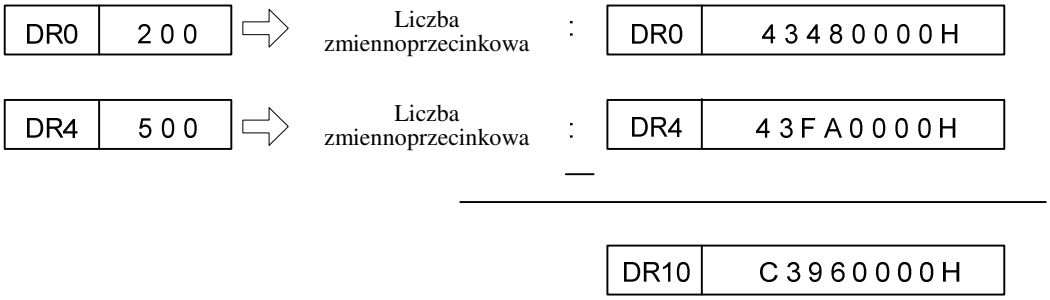
Sa, Sb, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	Liczba zmienno przecinkowa	V、Z P0～P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○		○

Opis

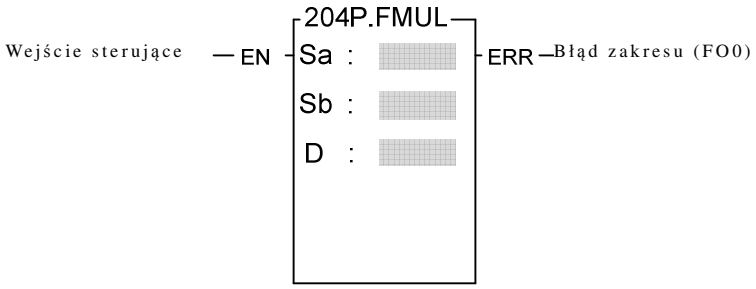
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operację odejmowania danych określonych w Sa i Sb i zapisze wyniki w rejestrze D. W przypadku, gdy wynik przekroczy dopuszczalny zakres wartości w jakim wyrażona może być liczba zmiennoprzecinkowa ($\pm 3.4*10^{38}$), stan wyjścia błędu ERR zmieni się na 1, a zawartość rejestru D pozostanie niezmieniona.





FUN 204 P FMUL	MNOŻENIE LICZB ZMIENNOPRZECINKOWYCH (FLOATING POINT NUMBER MULTIPLICATION)	FUN 204 P FMUL

SYMBOL DRABINKOWY



Sa: Mnożna

Sb: Mnożnik

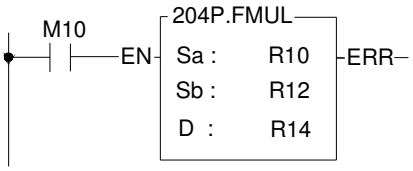
D : Adres rejestru przechowującego wynik przemnożenia

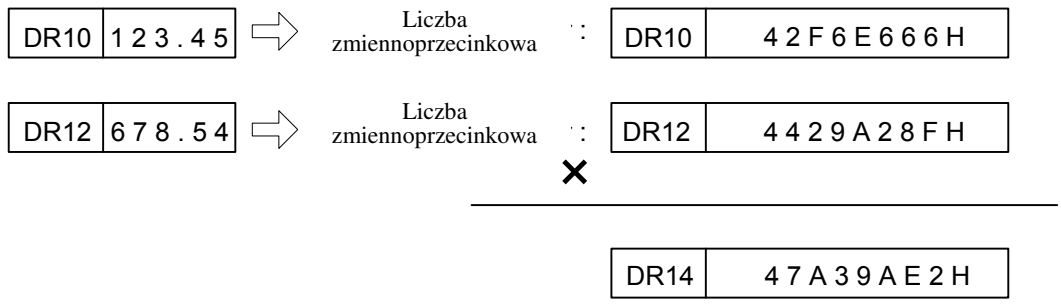
Sa, Sb, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	Liczba zmienno przecinkowa	V、Z P0～P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○		○

Opis

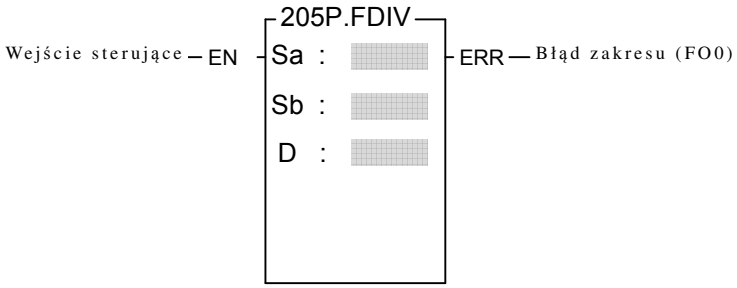
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operację mnożenia danych określonych w Sa i Sb i zapisze wyniki w rejestrze D. W przypadku, gdy wynik przekroczy dopuszczalny zakres wartości w jakim wyrażona może być liczba zmiennoprzecinkowa ($\pm 3.4*10^{38}$), stan wyjścia błędu ERR zmieni się na 1, a zawartość rejestru D pozostanie niezmieniona.





FUN 205 P FDIV	DZIELENIE LICZB ZMIENNOPRZECINKOWYCH (FLOATING POINT NUMBER DIVISION)	FUN 205 P FDIV

SYMBOL DRABINKOWY



Sa: Dzielna

Sb: Dzielnik

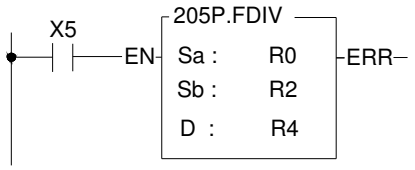
D : Adres rejestru przechowującego wynik dzielenia

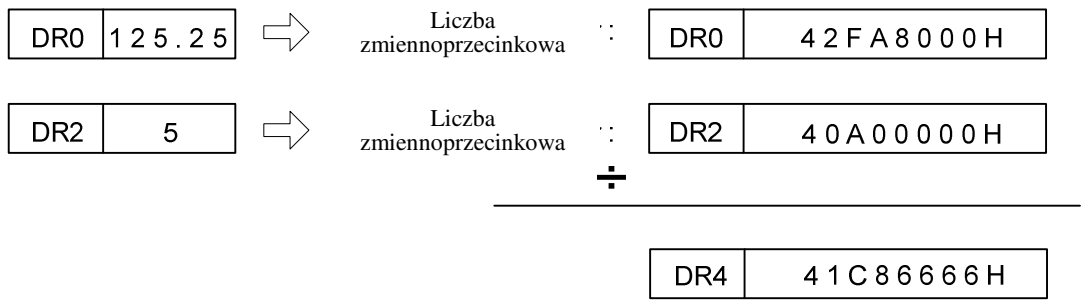
Sa, Sb, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	Liczba zmienno przecinkowa	V、Z P0～P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○		○

Opis

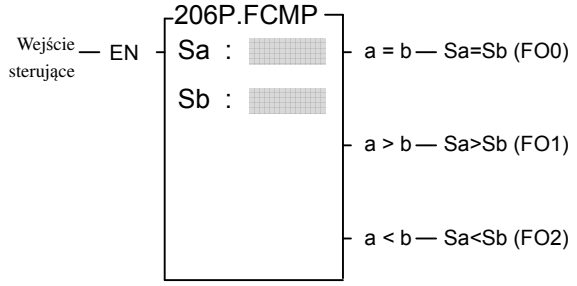
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zrealizuje operację dzielenia danych określonych w Sa i Sb i zapisze wyniki w rejestrze D. W przypadku, gdy wynik przekroczy dopuszczalny zakres wartości w jakim wyrażona może być liczba zmiennoprzecinkowa ($\pm 3.4*10^{38}$), stan wyjścia błędu ERR zmieni się na 1, a zawartość rejestru D pozostanie niezmieniona.





FUN 206 P FCMP	KOMPARATOR LICZB ZMIENNOPRZECINKOWYCH (FLOATING POINT NUMBER COMPARE)	FUN 206 P FCMP

SYMBOL DRABINKOWY



Sa: Adres rejestru do porównania

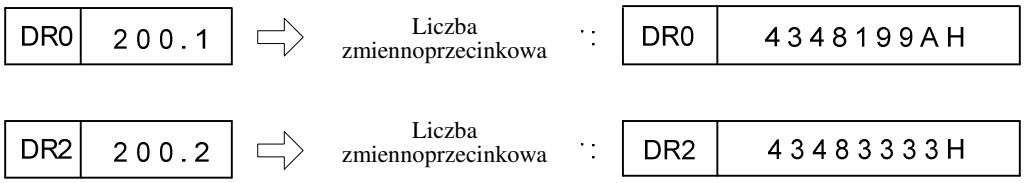
Sb: Adres rejestru do porównania

Sa, Sb mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	Liczba zmienno przecinkowa	V、Z P0～P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○

Opis

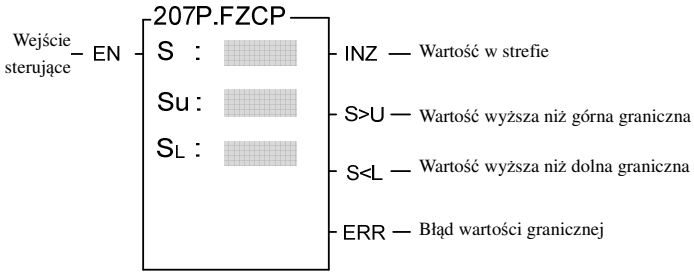
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja porówna wartości w Sa i Sb. Jeżeli wartość Sa będą tożsama z wartością Sb, satn wyjścia FO0 zostanie ustawiony na 1. Jeżeli Sa>Sb, to stan wyjścia FO1 zostanie ustawiony na 1. Jeżeli Sa<Sb, to stan wyjści FO2 zostanie ustawiony na 1



- W przykładzie powyżej, zakłada się, że w DR0 zapisana jest wartość 200.1 a w DR2 200.2. Następnie wartości te są porównywane za pomocą instrukcji FCMP. Stany FO0 i FO1 ustawione są na 0, a FO2 (a<b) ustawiony jest na 1, ponieważ a<b.
- Aby uzyskać złożony wynik, taki jak:. ≧、 ≦, < > itp., należy najpierw wysłać wyniki =, < i > do markerów, a następnie powiązać wyniki z markerów warunkiem logicznym.

FUN 207 P FZCP	ZMIENNOPRZECINKOWY KOMPARATOR STREFY (FLOATING POINT NUMBER ZONE COMPARE)	FUN 207 P FZCP

SYMBOL DRABINKOWY



S : Adres rejestru do sprawdzenia „czy w strefie?”

S_U : Górna wartość graniczna

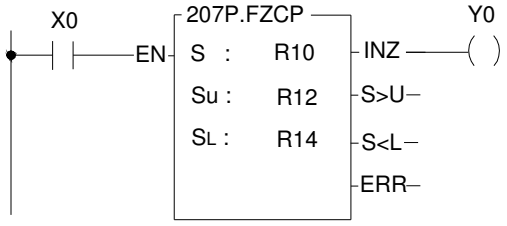
S_L: Dolna wartość graniczna

S, S_U, S_L mogą łączyć się z V, Z, P0~ w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
Su	○	○	○	○	○
SL	○	○	○	○	○

Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja porówna zawartość rejestru S z górną wartością graniczną S_U i dolną wartością graniczną S_L. Jeżeli zawartość S mieści się pomiędzy górną i dolną wartością graniczną (S_L<S<S_U), to stan wyjścia „INZ” zostanie ustawiony na 1. Jeżeli wartość S będzie wyższa od górnej wartości granicznej S_U, to stan wyjścia „S>U” zostanie ustawiony na 1. Jeżeli wartość S będzie niższa od dolnej wartości granicznej S_L, to stan wyjścia „S<L” zostanie ustawiony na 1.
- Górna wartość graniczna S_U, powinna być większa od dolnej wartości granicznej S_L. Jeżeli S_U<S_L, to stan wyjścia „ERR” zostanie ustawiony na 1, a funkcja nie zostanie wykonana.



- Funkcja po lewej, porównuje wartość DR10 z górną i dolną wartością graniczną, określonymi odpowiednio przez DR12 i DR14. Wartości DR10~DR14 przedstawione są po lewej stronie na schemacie poniżej. Po prawej stronie znajdują się wyniki porównania.
- Aby uzyskać informację o tym, że wartość jest poza strefą, należy użyć na wyjściu INZ cewki OUT NOT Y0, lub zastosować warunek OR pomiędzy dwoma wyjściami S>U i S<L, a wynik wystawić na Y0.

FUN 207 P FZCP	ZMIENNOPRZECINKOWY KOMPARATOR STREFY (FLOATING POINT NUMBER ZONE COMPARE)	FUN 207 P FZCP

S	DR10	2000.2	⇨ Liczba zmiennoprzecinkowa	:	DR10	44FA0666H	
Su	DR12	3000.3	⇨ Liczba zmiennoprzecinkowa	:	DR12	453B84CDH	(Górna wartość graniczna)
SL	DR14	1000.1	⇨ Liczba zmiennoprzecinkowa	:	DR14	447A0666H	(Dolna wartość graniczna)

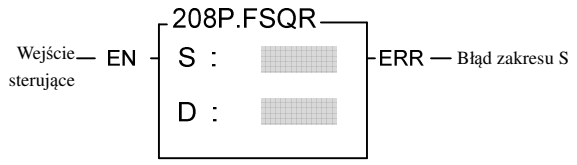
Przed wykonaniem

X0 = • → PORÓWNANIE STREFY LICZB ZMIENNOPRZECINKOWYCH → Y0 = 1

Wyniki wykonania funkcji

FUN 208 P FSQR	PIERWIATSEK KWADRATOWY Z LICZBY ZMIENNOPRZECINKOWEJ (FLOATING POINT NUMBER SQUARE ROOT)	FUN 208 P FSQR

SYMBOL DRABINKOWY



S : Adres źródlowy do wyciągnięcia pierwiastak

D : Adres rejestru do zapisu wyniku (wartości pierwiastka kwadratowego)

S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

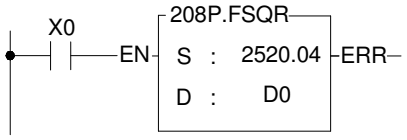
Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

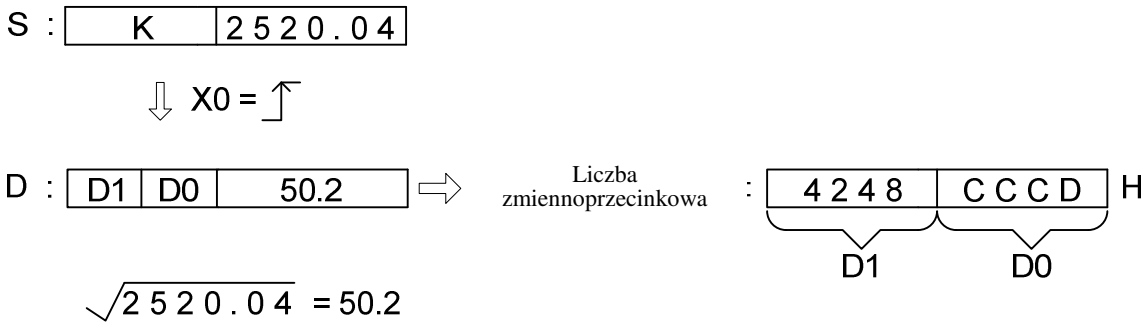
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.

- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja wyciągnie pierwiastek kwadratowy z wartości stałej wpisanej w polu S lub wartości pobranej z rejestrów S~S+1 i zapisze wynik w rejestrze określonym przez D~D+1.

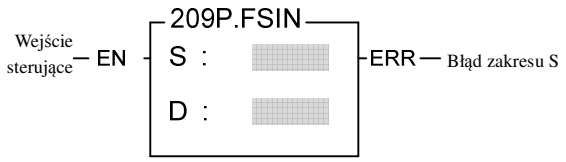
- Jeżeli wartość S będzie ujemna, to stan wyjścia „ERR” zostanie ustawiony na 1, a funkcja nie zostanie wykonana.





FUN 209 P FSIN	SINUS (SIN TRIGONOMETRIC INSTRUCTION)	FUN 209 P FSIN

SYMBOL DRABINKOWY



S : Rejestr źródłowy, dla którego obliczany jest sinus

D : Rejestr do zapisu wyniku (wartości sinusa)

S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

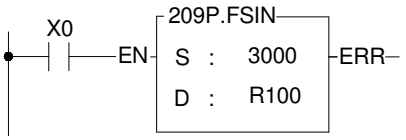
Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	16-bitowa liczba całkowita	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

Opis

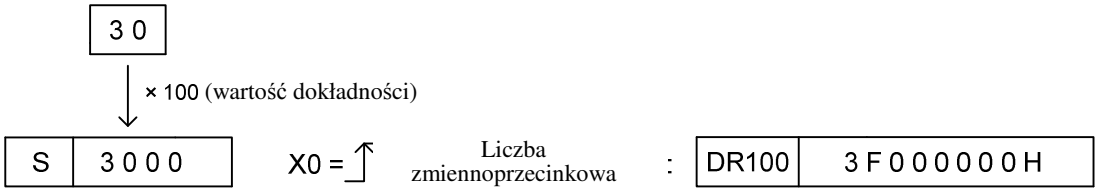
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.

- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy wartość sinusa z kąta, zapisanego w formacie całkowitym, określonego przez rejestr S i zapisze wynik w formcie zmiennoprzecinkowym w rejestrze D~D+1. Dopuszczalny zakres kąta wynosi od -18000 do +18000 (wyrażonego w formacie całkowitym z dokładnością 0.01 stopnia).

- Jeżeli wartość S będzie poza dopuszczalnym zakresem, stan wyjścia błędu „ERR” zostanie ustawiony na 1, a funkcja nie zostanie wykonana.



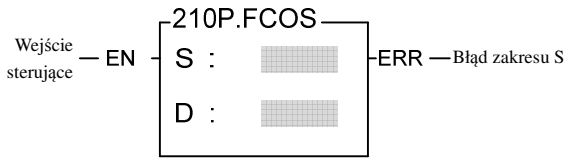
- W przykładzie po lewej, program oblicza sinus z kata 30 stopni i zapisuje wynik w rejestrze DR100.



SIN(30) = 0.5

FUN 210 P FCOS	COSINUS (COS TRIGONOMETRIC INSTRUCTION)	FUN 210 P FCOS

SYMBOL DRABINKOWY



S : Rejestr źródłowy, dla którego obliczany jest cosinus

D : Rejestr do zapisu wyniku (wartości cosinusa)

S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

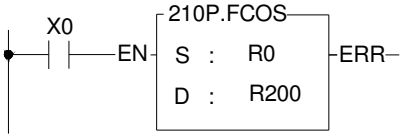
Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	16-bitowa liczba całkowita	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

Opis

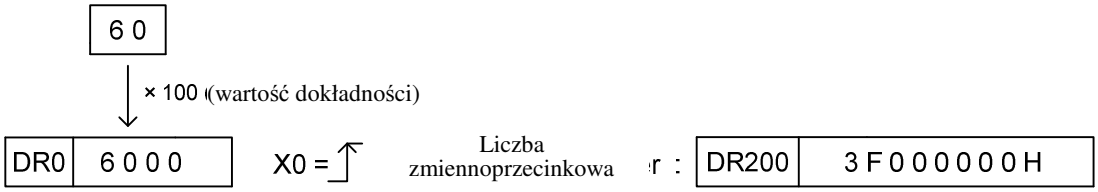
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.

- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy wartość cosinusa z kąta, zapisanego w formacie całkowitym, określonego przez rejestr S i zapisze wynik w formcie zmiennoprzecinkowym w rejestrze D~D+1. Dopuszczalny zakres kąta wynosi od -18000 do +18000 (wyrażonego w formacie całkowitym z dokładnością 0.01 stopnia).

- Jeżeli wartość S będzie poza dopuszczalnym zakresem, stan wyjścia błędu „ERR” zostanie ustawiony na 1, a funkcja nie zostanie wykonana.



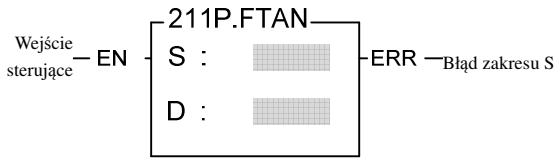
- W przykładzie po lewej, program oblicza cosinus z kąta 60 stopni i zapisuje wynik w rejestrze DR200.



COS(60) = 0.5

FUN 211 P FTAN	TANGENS (TAN TRIGONOMETRIC INSTRUCTION)	FUN 211 P FTAN

SYMBOL DRABINKOWY



S : Rejestr źródłowy, dla którego obliczany jest tangens

D : Rejestr do zapisu wyniku (wartości tangensa)

S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

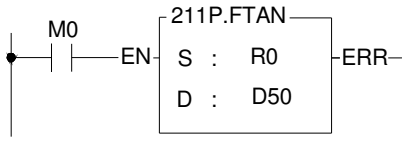
Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	16-bitowa liczba całkowita	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

Opis

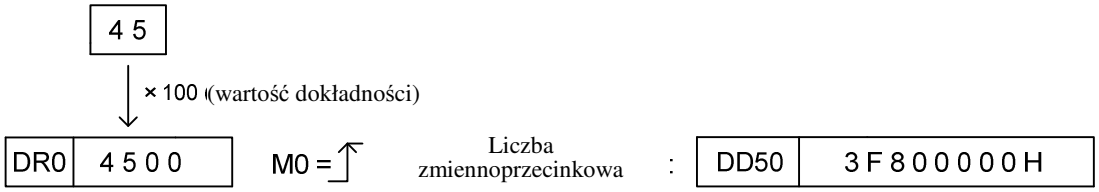
- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.

- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy wartość tangensa z kąta, zapisanego w formacie całkowitym, określonego przez rejestr S i zapisze wynik w formcie zmiennoprzecinkowym w rejestrze D~D+1. Dopuszczalny zakres kąta wynosi od -18000 do +18000 (wyrażonego w formacie całkowitym z dokładnością 0.01 stopnia).

- Jeżeli wartość S będzie poza dopuszczalnym zakresem, stan wyjścia błędu „ERR” zostanie ustawiony na 1, a funkcja nie zostanie wykonana.



- W przykładzie po lewej, program oblicza tangens z kąta 45 stopni i zapisuje wynik w rejestrze DD50.



TAN(45) = 1

FUN 212 P FNEG	ZMIANA ZNAKU LICZBY ZMIENNOPRZECINKOWEJ (CHANGE SIGN OF THE FLOATING POINT NUMBER)	FUN 212 P FNEG

SYMBOL DRABINKOWY

D : Rejestr do zmiany znaku

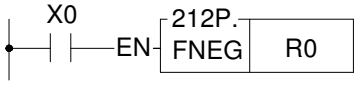
D może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D4095	V、Z P0～P9
D	○	○*	○	○

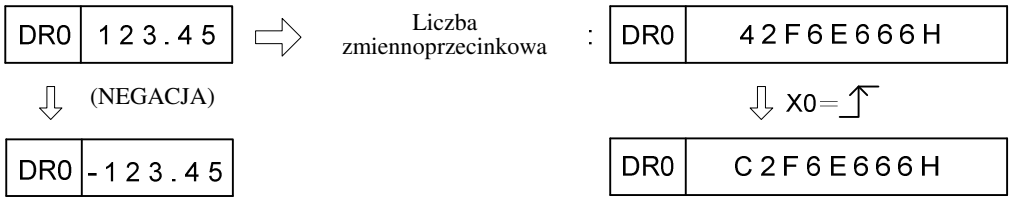
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja zmieni znak liczby zmeinnoprzecinkowej określonej przez rejestr D.

Przykład programowania

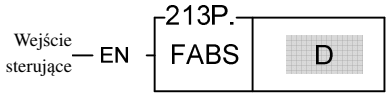


- Funkcja po lewej realizuje negację rejestru DR0 i zapisuje zanegowaną wartość z powrotem w DR0.



FUN 213 P FABS	WARTOŚĆ ABSOLUTNA LICZBY ZMIENNOPRZECINKOWEJ (FLOATING POINT NUMBER ABSOLUTE VALUE)	FUN 213 P FABS

SYMBOL DRABINKOWY



D : Do wyciągnięcia wartości bezwzględnej

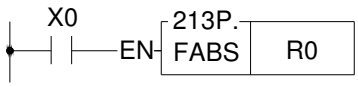
D może łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres	HR	ROR	DR	XR
Argument	R0 □ R3839	R5000 □ R8071	D0 □ D4095	V、Z P0～P9
D	○	○*	○	○

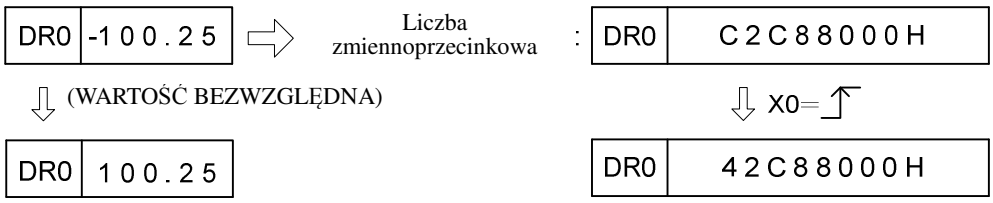
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754. Szczegóły dotyczące formatu znajdują się w rozdziale 5.3 (Systemy liczbowe) … strona 5 – 9.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), instrukcja obliczy wartość bezwzględną liczby zmeinnoprzecinkowej określonej przez rejestr D i zapisze obliczoną wartość ponownie w rejstrze D.

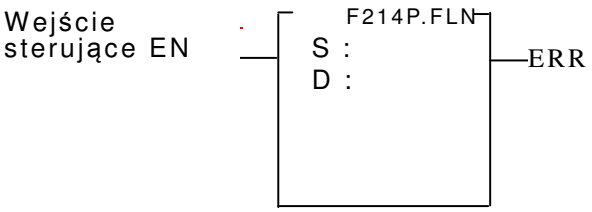
Przykład programowania



- Funkcja po lewej oblicza wartość bezwzględną rejestru DR0 i zapisuje obliczoną wartość z powrotem w DR0.



FUN 214 P FLN	LOGARYTM NATURALNY LICZBY ZMIENNOPRZECINKOWEJ (FLOATING POINT NAPIERIAN LOGARITHM, $\log_e x$ or ln(x))	FUN 214 P FLN



S : Dane źródłowe lub rejestr dla którego obliczana jest wartość logarytmu naturalnego.

D : Rejestr do zapisu wyniku

S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

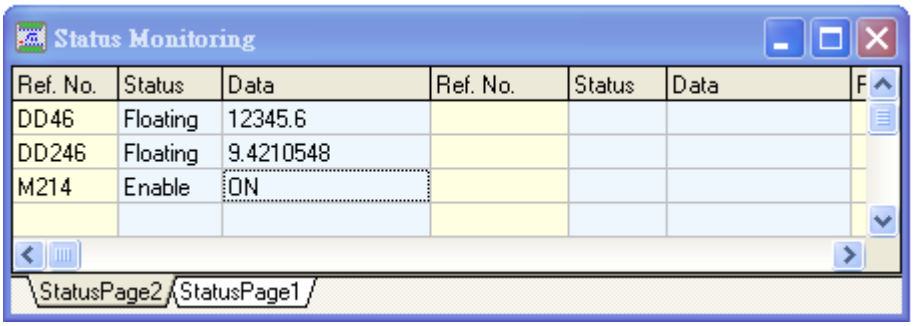
Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D3999	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754.
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy logarytm naturalny wartości stałej wpisanej w polu S lub wartości znajdującej się rejestrze S~S+1, oraz zapisze wynik w rejestrze D~D+1.
- Jeżeli wartość S będzie ujemna lub równa 0, albo wynik logarytmu będzie wykraczać poza dopuszczalny zakres, stan wyjścia błędu „ERR" zostanie ustawiony na 1, a wartość D~D+1 nie zostanie zaktualizowana.
- Funkcje operujące na liczbach zmiennoprzecinkowych, nie mogą być wywoływane wewnątrz procedur obsługi przerwań sprzętowych.

Przykład

• Jeżeli M214=1, oblicz wartość logarytmu naturalnego: DD246 = ln (DD46)



FUN 215 P FEXP	ZMIENNOPRZECINKOWA POTĘGA LICZY EULERA (FLOATING POINT NATURE POWER FUNCTION, e^x)	FUN 215 P FEXP

Wejście sterujące EN

```
EN ──┤ F215P.FEXP ├──ERR
     │ S :        │
     │ D :        │
```

S : Dane źródłowe lub rejestr dla którego obliczana jest wartość potęgi.

D : Rejestr do zapisu wyniku

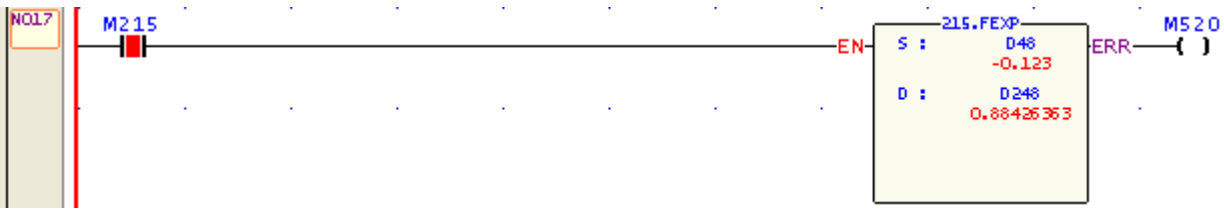
S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D3999	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

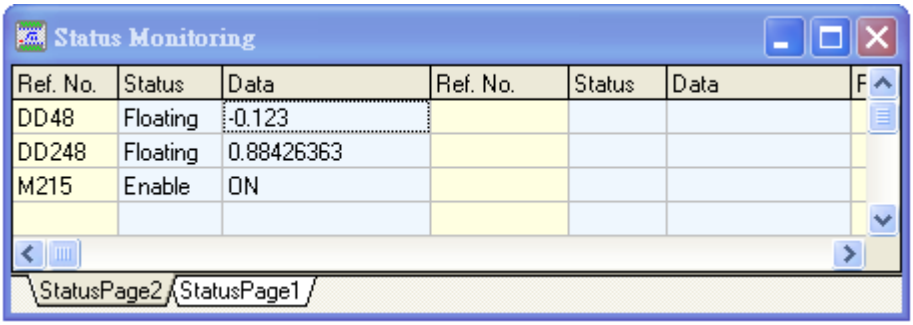
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy wartość potęgi dla wartości stałej wpisanej w polu S, lub wartości znajdującej się rejestrze S~S+1, oraz zapisze wynik w rejestrze D~D+1
- Jeżeli wartość S lub wynik potęgi będzie wykraczać poza dopuszczalny zakres, stan wyjścia błędu „ERR” zostanie ustawiony na 1, a wartość D~D+1 nie zostanie zaktualizowana
- Funkcje operujące na liczbach zmiennoprzecinkowych, nie mogą być wywoływane wewnątrz procedur obsługi przerwań sprzętowych.

Przykład

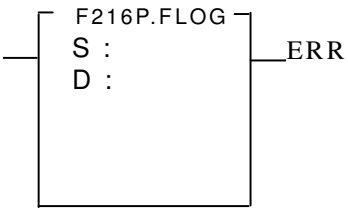


- Jeżeli M215=1, oblicz potęgę: DD248 = e^{DD48}



FUN 216 P FLOG	LOGARYTM LICZBY ZMIENNOPRZECINKOWEJ (FLOATING POINT LOGARITHM, $\log_{10}x$ or log(x))	FUN 216 P FLOG

Wejście sterujące EN



S : Dane źródłowe lub rejestr dla którego obliczana jest wartość logarytmu.

D : Rejestr do zapisu wyniku

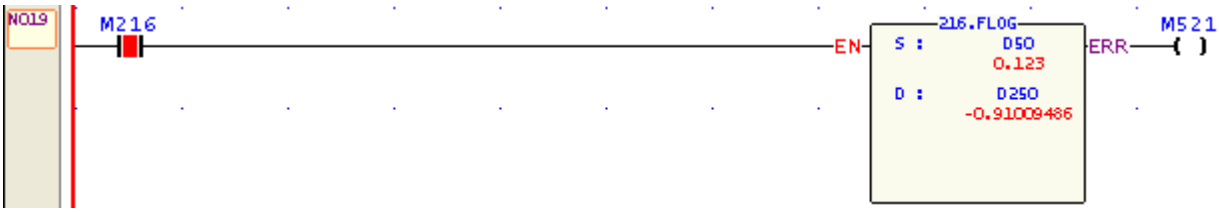
S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D3999	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

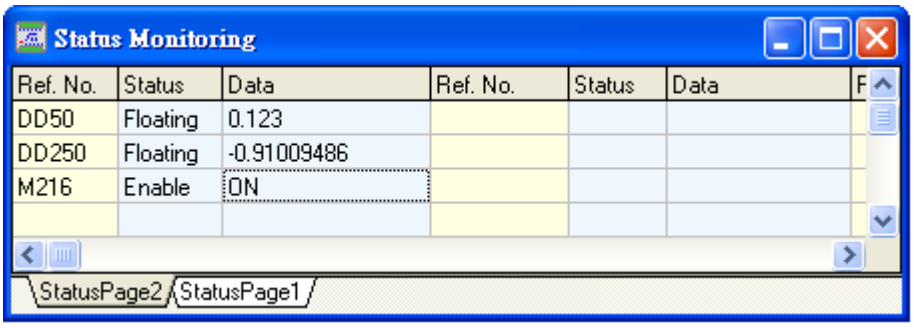
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy logarytm dla wartości stałej wpisanej w polu S, lub wartości znajdującej się rejestrze S~S+1, oraz zapisze wynik w rejestrze D~D+1
- Jeżeli wartość S będzie ujemna lub równa 0, albo wynik logarytmu będzie wykraczać poza dopuszczalny zakres, to status znacznika „ERR” zostanie ustawiony na 1, a wartość D~D+1 nie zostanie zaktualizowana.
- Funkcje operujące na liczbach zmiennoprzecinkowych, nie mogą być wywoływane wewnątrz procedur obsługi przerwań sprzętowych.

Przykład

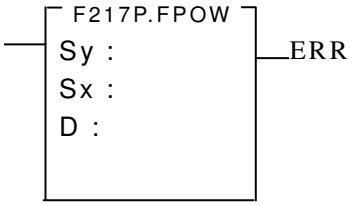


• Jeżeli M216=1, oblicz wartość lagarytmu: DD250 = log (DD50)



FUN 217 P FPOW	X DO POTĘGI Y NA LICZBACH ZMMIENNOPRZECINKOWYCH (FLOATING POINT POWER FUNCTION, x^y)	FUN 217 P FPOW

Wejście sterujące EN



Sy: Dane źródłowe lub rejestr wykładnika

SX: Dane źródłowe lub rejestr podstawy

D : Rejestr do zapisu wyniku

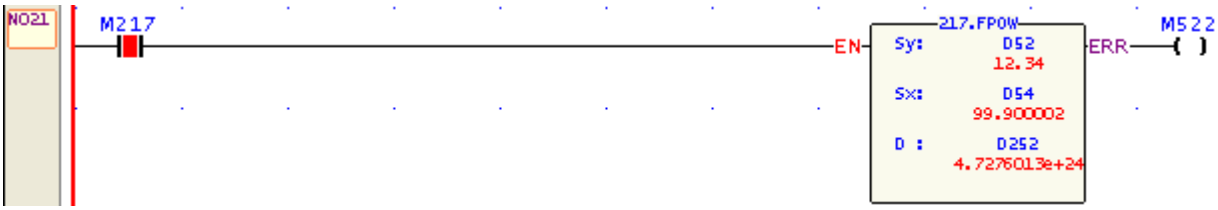
Sy, Sx, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D3999	Liczba zmienno przecinkowa	V、Z P0～P9
Sy	○	○	○	○	○
Sx	○	○*	○	○	○
D	○	○	○		○

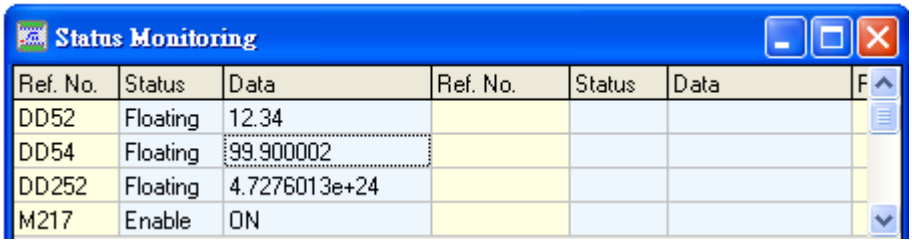
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy wartość potęgi, gdzie wykładnikiem jest wartości w rejestrze Sy, podstawą potęgi jest wartości w rejestrze Sx, a wynik zostanie zapisany w rejestrze D~D+1
- Gdy wynik znajdzie się poza zakresem, stan wyjścia błędu „ERR” zostanie ustawiony na 1, a wartość D~D+1 nie zostanie zaktualizowana.
- Funkcje operujące na liczbach zmiennoprzecinkowych, nie mogą być wywoływane wewnątrz procedur obsługi przerwań sprzętowych.

Przykład



- Jeżeli M217=1, oblicz wartość potęgi: DD252 = $DD54^{DD52}$



Ref. No.	Status	Data	Ref. No.	Status	Data
DD52	Floating	12.34			
DD54	Floating	99.900002			
DD252	Floating	4.7276013e+24			
M217	Enable	ON			

FUN 218 P FASIN	ARCUS SINUS LICZBY ZMIENNOPRZECINKOWEJ (FLOATING POINT ARC SINE FUNCTION, $\sin^{-1}$)	FUN 218 P FASIN

Wejście sterujące EN

```
F218P.FASIN
S :
D :
```

ERR

S : Dane źródłowe lub rejestr dla którego obliczana jest wartość arc sin.

D : Rejestr do zapisu wyniku

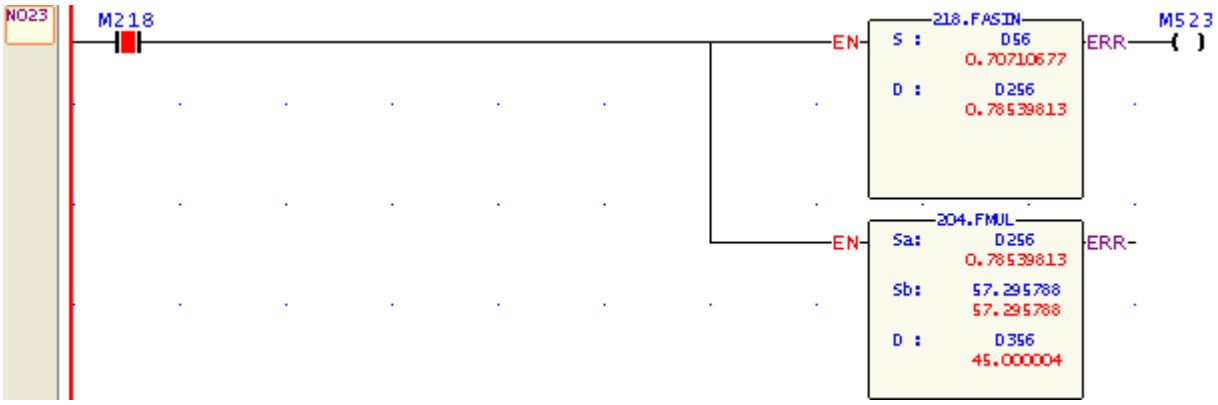
S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D3999	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754.
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy arc sin dla wartości stałej wpisanej w polu S, lub wartości znajdującej się rejestrze S~S+1, oraz zapisze wynik w rejestrze D~D+1.
- Zakres danych S: -1~ +1 ; zakres wartości D : $-\pi/2$ ~ $\pi/2$ (jednostką jest radian)
- Gdy wartość S znajdzie się poza zakresem, stan wyjścia błędu „ERR" zostanie ustawiony na 1, a wartość D~D+1 nie zostanie zaktualizowana.
- Funkcje operujące na liczbach zmiennoprzecinkowych, nie mogą być wywoływane wewnątrz procedur obsługi przerwań sprzętowych.

Przykład



- Jeżeli M218=1, oblicz arc sin: DD256 = $\sin^{-1}$ DD56;
  DD256(jednostka - radian) × 57.295788(180/π) w celu uzyskania wartości w stopniach.

Status Monitoring

Ref. No.	Status	Data	Ref. No.	Status	Data
DD56	Floating	0.70710677			
DD256	Floating	0.78539813			
M218	Enable	ON			
DD356	Floating	45.000004			

FUN 219 P FACOS	ARCUS COSINUS LICZBY ZMIENNOPRZECINKOWEJ (FLOATING POINT ARC COSINE FUNCTION, $\cos^{-1}$)	FUN 219 P FACOS

Wejście sterujące EN

```
F219P.FACOS
S :
D :
```

ERR

S : Dane źródłowe lub rejestr dla którego obliczana jest wartość arc cos.

D : Rejestr do zapisu wyniku

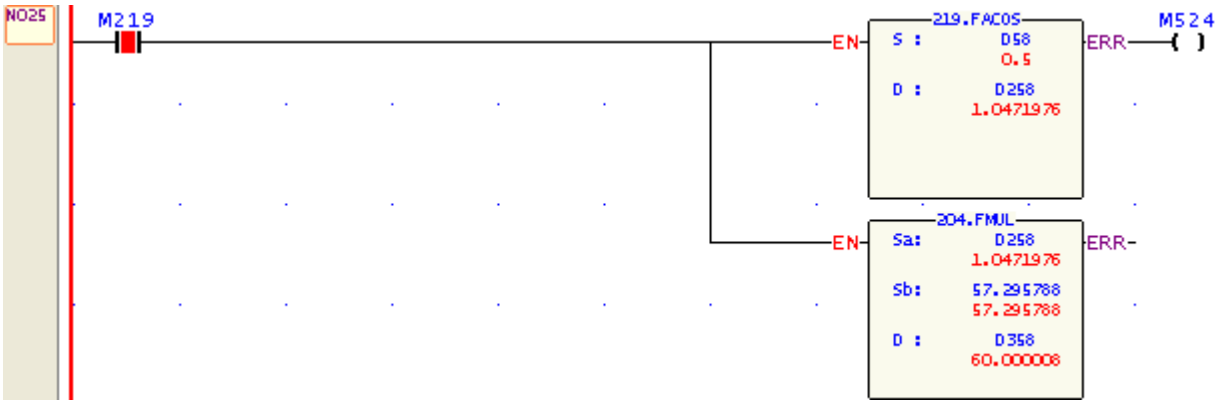
S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D3999	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

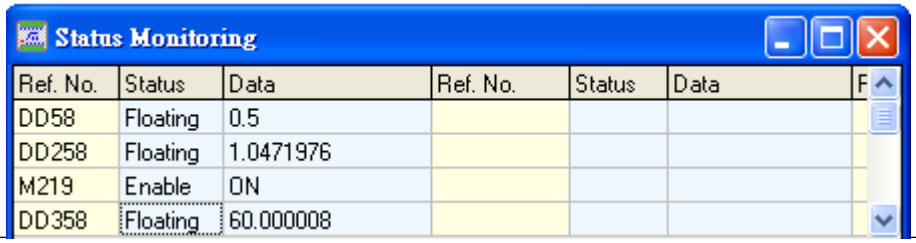
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754.
- Jeżeli wejście sterujące „EN” = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy arc cos dla wartości stałej wpisanej w polu S, lub wartości znajdującej się rejestrze S~S+1, oraz zapisze wynik w rejestrze D~D+1.
- Zakres danych S : -1~ +1 ; zakres wartości D : 0 ~ π (jednostką jest radian)
- Gdy wartość S znajdzie się poza zakresem, stan wyjścia błędu „ERR” zostanie ustawiony na 1, a wartość D~D+1 nie zostanie zaktualizowana.
- Funkcje operujące na liczbach zmiennoprzecinkowych, nie mogą być wywoływane wewnątrz procedur obsługi przerwań sprzętowych.

Przykład



- Jeżeli M219=1, oblicz arc cos: DD258 = $\cos^{-1}$ DD58;
  DD258(jednostka - radian) × 57.295788(180/π) w celu uzyskania wartości w stopniach



Status Monitoring

Ref. No.	Status	Data	Ref. No.	Status	Data
DD58	Floating	0.5			
DD258	Floating	1.0471976			
M219	Enable	ON			
DD358	Floating	60.000008			

FUN 220 P FATAN	ARCUS TANGENS LICZBY ZMIENNOPRZECINKOWEJ (FLOATING POINT ARC TANGENT FUNCTION, $\tan^{-1}$)	FUN 220 P FATAN

Wejście sterujące EN

```
F220P.FATAN
S :
D :
```

ERR

S : Dane źródłowe lub rejestr dla którego obliczana jest wartość arc tg.

D : Rejestr do zapisu wyniku

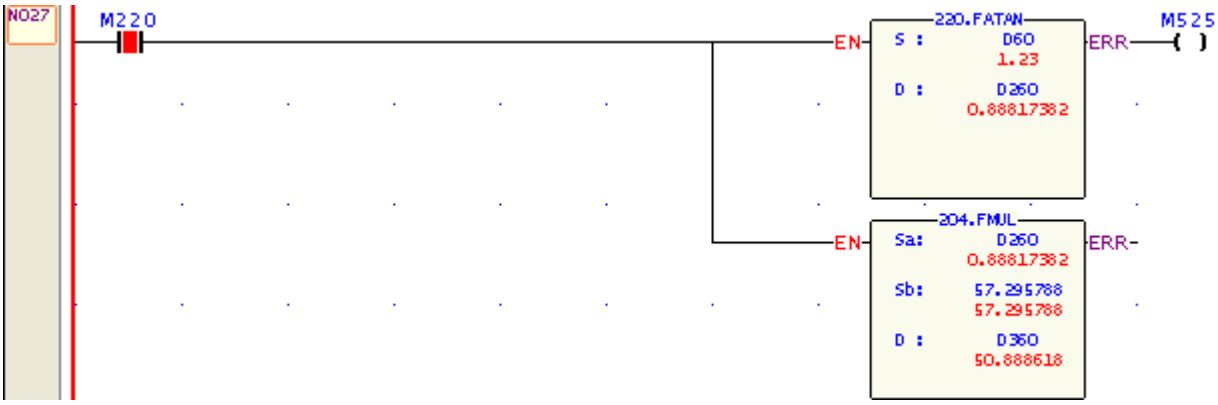
S, D mogą łączyć się z V, Z, P0~P9 w celu zastosowania adresowania pośredniego

Zakres / Argument	HR	ROR	DR	K	XR
	R0 □ R3839	R5000 □ R8071	D0 □ D3999	Liczba zmienno przecinkowa	V、Z P0～P9
S	○	○	○	○	○
D	○	○*	○		○

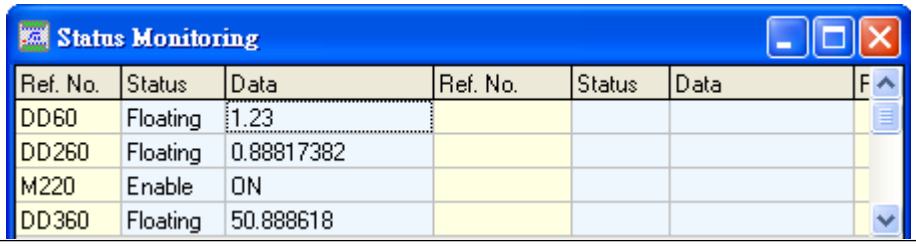
Opis

- Format wartości zmiennoprzecinkowych, stosowany w sterownikach PLC Fatek, jest zgodny z normą IEEE-754.
- Jeżeli wejście sterujące „EN" = 1 lub zmieni się z 0 na 1 (instrukcja P), funkcja obliczy arc tg dla wartości stałej wpisanej w polu S, lub wartości znajdującej się rejestrze S~S+1, oraz zapisze wynik w rejestrze D~D+1
- S może być dowolną wartością ; zakres wartości D : $-\pi/2$ ~ $\pi/2$ (jednostką jest radian)
- W przypadku nieprawidłowego adresowania pośredniego, stan wyjścia błędu „ERR" zostanie ustawiony na 1, a wartość D~D+1 nie zostanie zaktualizowana
- Funkcje operujące na liczbach zmiennoprzecinkowych, nie mogą być wywoływane wewnątrz procedur obsługi przerwań sprzętowych.

Przykład



- Jeżeli M220=1, oblicz wartość arc tg: DD260 = $\tan^{-1}$ DD60;
  DD260 (jednostka - radian) × 57.295788(180/π) w celu uzyskania wartości w stopniach



Status Monitoring

Ref. No.	Status	Data	Ref. No.	Status	Data
DD60	Floating	1.23			
DD260	Floating	0.88817382			
M220	Enable	ON			
DD360	Floating	50.888618			