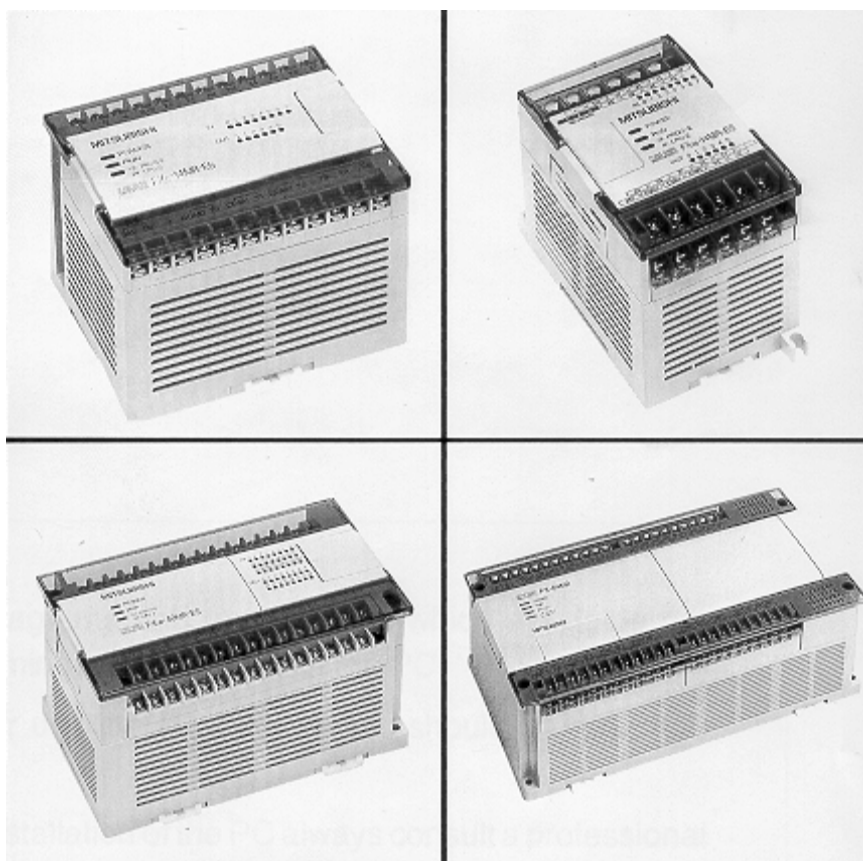


ADVANCED AND EVER ADVANCING **mitsubishi electric**



STEROWNIKI PROGRAMOWALNE
MELSEC-F

PODRECZNIK PROGRAMOWANIA
STEROWNIKÓW PROGRAMOWALNYCH SERII FX
(FX₀, FX_{0S}, FX_{0N}, FX, FX_{2C}, FX_{2N})



Sterowniki programowalne serii FX

Podręcznik programowania

Numer podręcznika: JY992D48301
Wersja podręcznika: G
Data wydania: Wrzesień 1997

Przedmowa

- Ten podręcznik zawiera opis, diagramy i objaśnienia urządzeń i instrukcji sterowników programowalnych PLC i stanowi wprowadzenie do ich poprawnego programowania i obsługi
- Przed przystąpieniem do instalacji PLC podręcznik ten powinien być przeczytany i zrozumiany
- W razie wątpliwości na jakimkolwiek etapie instalacji PLC należy skontaktować się ze specjalistą elektrykiem, wykwalifikowanym i przeszkolonym w zakresie lokalnych i państwowych standardów, które dotyczą miejsca instalacji.
- W razie wątpliwości co do obsługi lub używania PLC należy skontaktować się z najbliższym dystrybutorem Mitsubishi Electric.
- Zawartość podręcznika może ulec zmianie bez powiadomienia zainteresowanych

Przepisy dotyczące bezpieczeństwa użytkownika i ochrony sterownika programowalnego (PLC)

Ten podręcznik dostarcza informacji dotyczących używania sterowników z rodziny FX. Został napisany dla wyszkolonego i kompetentnego personelu. Określenie takiej osoby lub osób jest następujące:

- a) Każdy inżynier, który jest odpowiedzialny za planowanie, projektowanie i konstrukcję wyposażenia automatyki używając produktu stowarzyszonego z tym podręcznikiem, powinien być kompetentny, wyszkolony i wykwalifikowany w zakresie lokalnych i państwowych standardów wymaganych, aby spełniać tę rolę. Inżynierowie ci powinni być w pełni świadomi wszelkich aspektów bezpieczeństwa co do sprzętu automatycznego.
- b) Każdy inżynier eksploatacji i serwisu musi być kompetentny, szkoleny i wykwalifikowany w zakresie lokalnych i państwowych standardów aby wykonywać tę pracę. Inżynierowie ci powinni być także wyszkoleni w zakresie używania i konserwacji kompletnego produktu. Oznacza to znajomość całej załączonej dokumentacji wyżej wymienionego produktu. Cała konserwacja powinna być przeprowadzana zgodnie z założonymi procedurami bezpieczeństwa.
- c) Wszyscy operatorzy kompletnego sprzętu powinni być wyszkoleni w zakresie użytkowania produktu w sposób bezpieczny i skoordynowany stosownie do procedur bezpieczeństwa. Operatorzy powinni być także zaznajomieni z całą dokumentacją, która jest związana z działaniem całego sprzętu.

Uwaga: termin „cały sprzęt” odnosi się do trzyczęściowo skonstruowanego urządzenia, które zawiera lub korzysta z produktu związanego z tym podręcznikiem

Uwagi, o symbolach użytych w tym podręczniku

Wielokrotnie w tym podręczniku zostaną użyte pewne symbole do zaznaczenia tych informacji, które w zamierzeniu mają zapewnić osobiste bezpieczeństwo użytkownika i ochronić integralność wyposażenia. Kiedykolwiek zostanie napotkany któryś z następujących symboli, związana z nim notatka musi być przeczytana i zrozumiana. Każdy z używanych symboli zostanie teraz przedstawiony wraz z krótkim objaśnieniem jego znaczenia:

Ostrzeżenia dotyczące sprzętu



Oznacza, że wskazane zagrożenie spowoduje fizyczne uszkodzenia i utratę właściwości



Oznacza, że wskazane zagrożenie może spowodować fizyczne uszkodzenia i utratę właściwości



Oznacza punkt o szczególnym znaczeniu lub zostanie szczegółowo objaśniony

Ostrzeżenia dotyczące oprogramowania



Oznacza element oprogramowania, który musi być używany ze szczególną ostrożnością



Oznacza informację skojarzona z elementem oprogramowania, której użytkownik powinien być świadom



Oznacza punkt o szczególnym znaczeniu lub zostanie szczegółowo objaśniony

Spis treści

1	Wprowadzenie	1-2
1.1	Przegląd	1-2
1.2	Czym jest sterownik programowalny	1-3
1.3	Co jest potrzebne do programowania PLC?	1-3
1.4	Numery wersji CPU	1-4
1.4.1	Wersje CPU FX _{0N}	1-4
1.1.2	Wersje CPU FX i FX _{2C}	1-4
1.5	Specjalne wymagania dotyczące wyposażenia programowego	1-5
1.5.1	Wersja CPU FX 3.07 lub późniejsza i FX _{2C}	1-5
1.5.2	Wszystkie wersje CPU FX _{2N}	1-5
2	Podstawowe instrukcje programu	2-1
2.1	Czym jest program?	2-1
2.2	Zarys podstawowych urządzeń używanych w programowaniu	2-1
2.3	Jak czytać schemat drabinkowy?	2-2
2.4	Instrukcje LOAD i LOAD Inverse	2-3
2.5	Instrukcja OUT	2-4
2.5.1	Odmiany timerów i liczników	2-4
2.5.2	Dwukrotne użycie cewki tego samego urządzenia (przełącznika)	2-5
2.6	Instrukcje AND i AND INVERSE	2-6
2.7	Instrukcja OR i OR INVERSE	2-7
2.8	Instrukcja LOAD PULSE, LOAD TRAILING PULSE	2-8
2.9	Instrukcja AND PULSE, AND TRAILING PULSE	2-9
2.10	Instrukcja OR PULSE, OR TRAILING PULSE	2-10
2.11	Instrukcja OR BLOCK	2-11
2.12	Instrukcja AND BLOCK	2-12
2.13	Instrukcje MPS, MRD i MPP	2-13
2.14	Instrukcja MASTER CONTROL i MASTER CONTROL RESET	2-15
2.15	Instrukcja SET i RESET	2-17
2.16	Instrukcja TIMER, COUNTER(OUT i RESET)	2-18
2.1.1	Timery podstawowe, timery z pamięcią i liczniki	2-18
2.1.2	Standardowe liczniki 32-bitowe	2-19
2.1.3	Liczniki szybkie	2-19
2.17	Impulsy narastające i opadające	2-20
2.18	Instrukcja INVERSE	2-21
2.19	Instrukcja NO OPERATION	2-22
2.20	Instrukcja END	2-23
3	Programowanie STL	3-1
3.1.1	Czym jest STL, SFC i IEC1131 część 3?	3-1
3.2	Jak działa STL?	3-2
3.2.1	Każdy krok jest programem	3-2
3.3	Jak zacząć i zakończyć program STL?	3-3
3.1.1	Zagnieżdżone programy STL	3-3
3.1.2	Aktywowanie nowych stanów	3-3
3.1.3	Zatrzymywanie programu STL	3-4
3.4	Przechodzenie pomiędzy krokami STL	3-5
3.4.1	Użycie instrukcji SET do sterowania cewką znacznika stanu STL	3-5
3.4.2	Użycie instrukcji OUT do sterowania cewką znacznika stanu STL	3-6
3.5	Reguły i techniki pisania programów STL	3-7
3.5.1	Podstawowe uwagi dotyczące zachowania się programów STL	3-7
3.5.2	Sterowanie kroku pojedynczym sygnałem	3-9
3.6	Ograniczenia niektórych instrukcji używanych z STL	3-10
3.7	Rozgałęzienia wyboru w programie STL	3-11
3.8	Rozgałęzienia równoległe w programie STL	3-12
3.9	Ogólne zasady realizacji poprawnych rozgałęzień STL	3-14
3.10	Ogólne zalecenia dotyczące stosowania oprogramowania FX-PCS/AT-EE	3-15
3.11	Przykłady programowania	3-16
3.11.1	Prosty przepływ STL	3-16

3.11.2	Przykładowy program rozgałęzienia wyboru	3-18
3.12	Zaawansowane użycie STL	3-20
4	Charakterystyka urządzeń	4-1
4.1	Wejścia	4-1
4.2	Wyjścia	4-2
4.3	Znaczniki M	4-3
4.3.1	Ogólne informacje dotyczące przekaźników pomocniczych	4-3
4.3.2	Przekaźniki pomocnicze z podtrzymaniem/zatraskowe	4-4
4.3.3	Znaczniki systemowe	4-5
4.1.4	Znaczniki pojedynczego impulsu	4-5
4.4	Znaczniki stanu S	4-6
4.4.1	Ogólny stan znaczników	4-6
4.4.2	Znaczniki stanu z podtrzymaniem/zatraskowe	4-7
4.4.3	Znaczniki kroku STL	4-8
4.4.4	Znaczniki wskaźnikowo-diagnostyczne	4-9
4.5	Etykiety skoków	4-10
4.6	Etykiety przerwania	4-11
4.6.1	Przerwania od wejść	4-12
4.6.2	Przerwania od timerów	4-12
4.6.3	Blokowanie poszczególnych przerwania	4-13
4.6.4	Przerwania od liczników	4-13
4.7	Stała K	4-14
4.8	Stała H	4-14
4.9	Timery	4-15
4.9.1	Zasada działania timera	4-16
4.9.2	Timery o przełączanej rozdzielczości	4-16
4.9.3	Timery z pamięcią	4-17
4.9.4	Timery wykorzystywane w przerwaniach i podprogramach (CALL)	4-18
4.1.5	Dokładność timera	4-18
4.10	Liczniki	4-19
4.10.1	Liczniki 16-bitowe, dodające, zwykłe i zatraskowe	4-20
4.10.2	Liczniki 32-bitowe, dwukierunkowe, zwykłe i zatraskowe	4-21
4.11	Liczniki szybkie	4-22
4.1.1	Podstawowe operacje na liczniku szybkim	4-23
4.1.2	Dostępność liczników szybkich w programowalnych sterownikach FX ₀ , FX _{0S} i FX _{0N}	4-24
4.1.3	Dostępność liczników szybkich w programowalnych sterownikach FX i FX _{2C}	4-25
4.1.4	Dostępność liczników szybkich w programowalnych sterownikach FX _{2N}	4-28
4.1.5	Liczniki jednofazowe - uruchamiane i zerowane programowo (C235-C240)	4-29
4.1.6	Liczniki jednofazowe - z uruchamianiem i zerowaniem sprzętowym (C246-C250)	4-30
4.1.7	Liczniki dwufazowe, dwukierunkowe (C246-C250)	4-31
4.1.8	Liczniki A/B fazowe (C252-C255)	4-32
4.12	Rejestry danych	4-33
4.1.1	Rejestry ogólnego przeznaczenia	4-34
4.1.2	Rejestry z podtrzymaniem/zatraskowe	4-35
4.1.3	Rejestry diagnostyczne	4-35
4.1.4	Rejestry plikowe	4-36
4.1.5	Rejestry ustawiane zewnętrznie	4-37
4.13	Rejestry indeksowe	4-38
4.1.1	Modyfikowanie stałej	4-39
4.1.2	Niewłaściwe użycie modyfikatorów	4-39
4.1.3	Użycie wielokrotne rejestrów indeksowych	4-39
4.14	Bity, Słowa, zapis BCD i zapis szesnastkowy	4-40
4.1.1	Urządzenia bitowe, pojedyncze i grupowe	4-40
4.1.2	Urządzenia słowowe	4-42
4.1.3	Interpretacja danych słowowych	4-42
4.1.4	Uzupełnienie do 2	4-45
4.15	Zapis zmiennoprzecinkowy i szesnastkowy	4-46
4.15.1	Zapis naukowy	4-47
4.1.2	Format zmiennoprzecinkowy	4-48
4.1.3	Zestawienie notacji naukowej i liczb zmiennoprzecinkowych	4-49

5. Instrukcje specjalne			5-1
5.1 Sterowanie przebiegiem programu - funkcje 00-09			5-4
5.1.1 CJ (FNC 00)	5-5	5.1.2 CALL (FNC 01)	5-7
5.1.4 IRET, EI, DI (FNC 03, 04, 05)	5-9	5.1.5 FEND (FNC 06)	5-11
5.1.7 FOR, NEXT (FNC 08, 09)	5-13	5.1.3 SRET (FNC 02)	5-8
5.2 Instrukcje przesłania i porównania - funkcje 10-19		5.1.6 WDT (FNC 07)	5-12
5.2.1 CMP (FNC 10)	5-17	5.2.3 ZCP (FNC 11)	5-17
5.2.4 SMOV (FNC 13)	1-18	5.2.5 CML (FNC 14)	5-19
5.2.7 FMOV (FNC 16)	5-21	5.2.8 XCH (FNC 17)	5-21
5.2.10 BIN (FNC 19)	5-22	5.2.3 MOV (FNC 12)	5-16
5.3 Operacje arytmetyczne i logiczne - funkcje 30-39		5.2.6 BMOV (FNC 15)	5-20
5.3.1 ADD (FNC 20)	5-25	5.2.9 BCD (FNC 18)	5-22
5.3.4 DIV (FNC 23)	5-28	5.3.2 SUB (FNC 21)	5-26
5.3.7 WAND (FNC 26)	5-30	5.3.5 INC (FNC 24)	5-29
5.3.10 NEG (FNC 29)	5-31	5.3.8 WOR (FNC 27)	5-30
5.4 Rotacje i przesunięcia - funkcje 30-39		5.3.3 MUL (FNC 22)	5-27
5.4.1 ROR (FNC 30)	5-35	5.3.6 DEC (FNC 25)	5-29
5.4.4 RCL (FNC 33)	5-36	5.3.9 WXOR (FNC 28)	5-31
5.4.7 WSFR (FNC 36)	5-38	5.4.3 RCR (FNC 32)	5-36
5.4.10 SFRD (FNC 39)	5-40	5.4.6 SFTL (FNC 35)	5-37
5.5 Operacje na danych - funkcje 40-49		5.4.8 WSFL (FNC 37)	5-38
5.5.1 ZRST (FNC 40)	5-43	5.4.9 SFWR (FNC 38)	5-39
5.5.4 SUM (FNC 43)	5-45	5.5.2 DECO (FNC 41)	5-43
5.5.7 ANS (FNC 46)	5-47	5.5.5 BON (FNC 44)	5-45
5.5.10 FLT (FNC 49)	5-49	5.5.8 ANR (FNC 47)	5-47
5.6 Szybkie przetwarzanie - funkcje 50-59		5.5.3 ENCO (FNC 42)	5-44
5.6.1 REF (FNC 50)	5-53	5.5.6 MEAN (FNC 45)	5-46
5.6.4 HSCS (FNC 53)	5-55	5.5.9 SQR (FNC 48)	5-48
5.6.7 ALT (FNC 56)	5-60	5.6.3 MTR (FNC 52)	5-52
5.6.10 PLSR (FNC 59)	5-63	5.6.6 HSZ (FNC 55)	5-57
5.7 Instrukcje podręczne - funkcje 60-69		5.6.8 PLSY (FNC 57)	5-61
5.7.1 IST (FNC 60)	5-67	5.6.9 PWM (FNC 58)	5-62
5.7.4 INCD (FNC 63)	5-71	5.7.2 SER (FNC 61)	5-69
5.7.7 ALT (FNC 66)	5-73	5.7.5 TTMR (FNC 64)	5-72
5.7.10 SORT (FNC 69)	5-77	5.7.8 RAMP (FNC 67)	5-73
5.8 Zewnętrzne urządzenia wejścia/wyjścia - funkcje 70-79		5.7.3 ABSD (FNC 62)	5-70
5.8.1 TKY (FNC 70)	5-81	5.7.6 STMR (FNC 65)	5-72
5.8.4 SEGD (FNC 73)	5-84	5.7.9 ROTC (FNC 68)	5-75
5.8.7 ASC (FNC 76)	5-88	5.8.2 HKY (FNC 71)	5-82
5.8.10 TO (FNC 79)	5-91	5.8.5 SEGL (FNC 74)	5-85
5.9 Zewnętrzne urządzenia szeregowo - funkcje 80-89		5.8.8 PR (FNC 77)	5-89
5.9.1 RS (FNC 80)	5-95	5.8.3 DSW (FNC 72)	5-83
5.9.4 HEX (FNC 83)	5-102	5.8.6 ARWS (FNC 75)	5-87
5.9.7 VRSC (FNC 86)	5-104	5.8.9 FROM (FNC 78)	5-90
5.10 Zewnętrzne jednostki F2 - funkcje 90-99		5.9.2 PRUN (FNC 81)	5-99
5.10.1 MNET (FNC 90)	5-113	5.9.5 CCD 9FNC 84)	5-103
5.10.4 RMST (FNC 93)	5-114	5.9.8 PID (FNC 88)	5-105
5.10.7 RMMN (FNC 96)	5-116	5.9.3 ASCI (FNC 82)	5-101
5.11 Operacje zmiennoprzecinkowe 1 & 2 - funkcje 110-129		5.9.6 VRRD (FNC 85)	5-104
5.11.1 ECMP (FNC 110)	5-121	5.10.2 ANRD (FNC 91)	5-113
5.11.4 EBIN (FNC 119)	5-122	5.10.5 RMWR (FNC 94)	5-115
5.11.7 EMUL (FNC 122)	5-124	5.10.8 BLK (FNC 97)	5-117
5.11.10 INT (FNC 129)	5-126	5.10.3 ANWR (FNC 92)	5-114
5.12 Operacje trygonometryczne - funkcje 130-139		5.10.6 RMRD (FNC 95)	5-116
5.12.1 SIN (FNC 130)	5-129	5.10.9 MCDE (FNC 98)	5-118
5.12.2 COS (FNC 131)	5-130	5.11.3 EBCD (FNC 118)	5-122
5.12.3 TAN (FNC 132)	5-130	5.11.6 ESUB (FNC 121)	5-124
5.13 Operacje na danych 2 - funkcje 140-149		5.11.8 EDIV (FNC 123)	5-125
5.13.1 SWAP (FNC 147)	5-133	5.11.9 ESQR (FNC 127)	5-125
5.14 Sterowanie zegarem czasu rzeczywistego - funkcje 160-169		5.12.1 SIN (FNC 130)	5-129
5.14.1 TCMP (FNC 160)	5-137	5.12.2 COS (FNC 131)	5-130
5.14.4 TSUB (FNC 163)	5-140	5.12.3 TAN (FNC 132)	5-130
5.14.5 TRD (FNC 166)	5-141	5.13.1 SWAP (FNC 147)	5-133
5.15 Kody Graya - funkcje 170-179		5.14.2 TZCP (FNC 161)	5-138
5.15.1 GRY (FNC 170)	5-145	5.14.5 TRD (FNC 166)	5-141
5.15.2 GBIN (FNC 171)	5-145	5.14.3 TADD (FNC 162)	5-139
5.16 Porównania operacyjne - funkcje 220-249		5.14.6 TWR (FNC 167)	5-142
5.16.1 LD compare (FNC 224-230)	5-149	5.15.1 GRY (FNC 170)	5-145
5.16.2 AND compare (FNC 232-238)	5-150	5.15.2 GBIN (FNC 171)	5-145
5.16.3 OR compare (FNC 240-246)	5-151	5.16.1 LD compare (FNC 224-230)	5-149

6	Urządzenia diagnostyczne	6-1
6.1	Status PLC (M8000-M8009 i D8000-D8009)	6-2
6.2	Urządzenia zegara (M8000- M8009 i D8000-D8009)	6-3
6.3	Znaczniki operacyjne	6-4
6.4	Tryb programowy PLC (M8030-M8039 i D8030-D8039)	6-5
6.5	Znaczniki STL (M8040-M8049 i D8040-D8049)	6-6
6.6	Znaczniki przerwania (M8050-M8059 i D8050-D8059)	6-7
6.7	Urządzenia detekcji błędów (M8060-M8069 i D8060-D8069)	6-8
6.8	Urządzenia operacji łączących (M8070-M8099 i D8070-D8099)	6-9
6.9	Urządzenia pomocnicze (M8100-M8119 i D8100-D8119)	6-10
6.10	Urządzenia adaptera komunikacyjnego, tzn. 232ADP, 485ADP (M8120-M8129 i D8120-D8129)	6-10
6.11	Znaczniki tabeli porównań HSZ (M8130-M8139 i D8130-D8139)	6-11
6.12	Urządzenia pomocnicze (M8160-M8199)	6-12
6.13	Rejestry indeksowe (D8180 do D8199)	6-13
6.14	Sterowanie licznika góra/dół (M8200-M8234 i D8200-D8234)	6-14
6.15	Sterowanie szybkiego licznika (M8235-M8255 i D8235-D8255)	6-14
6.16	Tabela kodów błędów	6-15
7	Czasy wykonania i hierarchia instrukcji	7-1
7.1	Instrukcje podstawowe	7-1
7.2	Instrukcje specjalne	7-3
7.3	Hierarchiczne powiązania instrukcji podstawowych	7-12
7.4	Przetwarzanie wsadowe	7-14
7.5	Podsumowanie przydziału pamięci urządzeń	7-14
7.6	Ograniczenia w użyciu instrukcji	7-16
7.1.1	Instrukcje, które mogą być używane tylko raz w głównym obszarze programu	7-16
7.1.2	Instrukcje, które nie są odpowiednie do stosowania z jednostkami wejściowymi 110V AC	7-16
8	Specyfikacja urządzeń sterownika	8-1
8.1	Specyfikacja urządzeń FX ₀ i FX _{0S}	8-1
8.2	Specyfikacja urządzeń FX _{0N}	8-2
8.3	Specyfikacja urządzeń FX (wersja CPU 2.0 do 3.06)	8-4
8.4	Specyfikacja urządzeń FX (wersja CPU od 3.07 i wszystkie FX _{2C})	8-6
8.5	Specyfikacja urządzeń FX _{2N}	8-8
9	System adresowania urządzeń	9-1
9.1	Adresowanie modułów rozszerzeń	9-1
9.2	Użycie modułu FX2-24EI do obsługi specjalnych bloków funkcjonalnych serii F	9-2
9.3	Użycie modułu FX2-24EI do obsługi F-16NP/NT	9-3
9.1.1	Użycie modułu FX2-24EI do obsługi F2-6A	9-4
9.1.2	Użycie modułu FX2-24EI do obsługi F2-32RM	9-4
9.1.3	Użycie modułu FX2-24EI do obsługi F2-30GM	9-5
9.4	Adaptory połączeń równoległych FX2-40AP/W	9-6
9.5	Funkcje zegara czasu rzeczywistego	9-7
9.1.1	Ustawianie zegara czasu rzeczywistego	9-8
10	Wskazówki techniczne	10-1
10.1	Wskazówki zaawansowanego programowania	10-1
10.2	Użytkownicy jednostek FX zasilanych prądem stałym	10-1
10.3	Wykorzystanie znaczników RUN/STOP	10-2
10.1.1	Konfiguracja przycisku RUN/STOP	10-2
10.1.2	Zdalne sterowanie RUN/STOP	10-3
10.4	Tryb stałego czasu przebiegu programu	10-4
10.5	Zamiana stanów ON/OFF	10-4
10.6	Stosowanie urządzeń z podtrzymaniem w celu osiągnięcia maksymalnej sprawności	10-5
10.7	Indeksowanie przy wyświetlaniu wielu wartości danych	10-5
10.8	Czytanie i obróbka danych z nastawników cyfrowych	10-6
10.9	Pomiary impulsów o wysokiej częstotliwości	10-6

10.9.1	Pomiar impulsu z wykorzystaniem timera 1 ms	10-6
10.1.2	Pomiar impulsu z wykorzystaniem timera 0,1 ms	10-7
10.10	Wykorzystanie znacznika zakończenia wykonywania operacji M8029	10-7
10.11	Tworzenie instrukcji MTR definiowanej przez użytkownika	10-8
10.12	Przykładowa aplikacja systemu, wykorzystująca STL i IST	10-8
10.13	Wykorzystanie instrukcji PWM do sterowania silnikiem	10-15
10.14	Techniki programowania PID	10-18
10.14.1	Ustawienie parametrów początkowych PID	10-18
10.14.2	Utrzymywanie MV w zadanym zakresie	10-19
10.14.3	Zmiana sterowania ręczna/automatyczna	10-19
10.1.4	Wykorzystanie sygnałów alarmowych PID	10-20
10.1.5	Inne wskazówki programowania PID	10-20
10.15	Dodatkowe funkcje PID	10-21
10.15.1	Sterowanie zakresem wartości wyjściowej	10-21
10.15.2	Operacja samodostrajania	10-22

Indeks **11-1**

Tabela znaków ASCII	11-9
Lista instrukcji specjalnych	11-10

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

1	Wprowadzenie	1-2
1.1	Przegląd	1-2
1.2	Czym jest sterownik programowalny	1-3
1.3	Co jest potrzebne do programowania PLC?	1-3
1.4	Numery wersji CPU	1-4
1.4.1	Wersje CPU FX _{0N}	1-4
1.4.2	Wersje CPU FX i FX _{2C}	1-4
1.5	Specjalne wymagania dotyczące wyposażenia programowego	1-5
1.5.1	Wersja CPU FX 3.07 lub późniejsza i FX _{2C}	1-5
1.5.2	Wszystkie wersje CPU FX _{2N}	1-5

1 Wprowadzenie

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1.1 Przegląd

1. Cel podręcznika

Podręcznik ten zawiera szczegóły dotyczące wszystkich aspektów działania i programowania sterowników programowalnych FX, FX2C, FX0N, FX0S i FX0. Informacji dotyczących sprzętu i instalacji sterownika należy szukać we właściwym podręczniku dostarczonym wraz z modułem.

2. Jak korzystać z tego podręcznika?

Podręcznik ten omawia wszystkie funkcje całej rodziny sterowników programowalnych rodziny FX. Z tego powodu do odpowiednich rozdziałów zostały załączone poniższe symbole, wskazujące których sterowników dana sekcja dotyczy.

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Zacieniowane pole wskazuje na omawiany typ sterownika

- "FX_{0(S)}" - wszystkie sterowniki FX₀ i FX_{0S}
- "FX_{0N}" - wszystkie sterowniki FX_{0N}
- "FX" - wszystkie sterowniki FX i FX₂ (wersja CPU 2.30 lub wcześniejsze)
- "FX_(2C)" - wszystkie sterowniki FX i FX₂ (wersja CPU 3.07 lub późniejsze)
wszystkie sterowniki FX_{2C} (patrz strona 1-4)
- "FX_{2N}" - wszystkie sterowniki FX_{2N}

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Jeżeli pole jest zacieniowane w połowie, jak na rys. z lewej, oznacza to, że nie wszystkie funkcje opisane w danym rozdziale dotyczą tego sterownika. Tekst objaśnia dalsze szczegóły albo w tekście pojawiają się niezależne odnośniki.

Jeżeli przy tytule rozdziału brak jest w/w symboli, należy przyjąć, że sekcja ta dotyczy wszystkich typów sterowników, chyba że będzie to wyraźnie napisane.

3. Rodzina FX

Ogólny termin, który jest często używany do opisanie wszystkich sterowników bez specyfikowania indywidualnych typów lub nazw modeli.

4. Numery wersji CPU i wspomaganie programowe

Jako że Mitsubishi unowocześnia każdy model, różne wersje mają różne możliwości.

- szczegóły dotyczące numerów wersji i ich możliwości znajdują się w rozdziale 1.4
- szczegóły dotyczące zewnętrznego programowania każdego modelu znajdują się w rozdziale 1.5

1.2 Czym jest sterownik Programowalny?

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Sterownik programowalny (sterownik o programowalnej logice, ang. programmable logic controller, PLC) to urządzenie, które może być zaprogramowane przez użytkownika do przyjęcia sekwencji zdarzeń. Zdarzenia te są wyzwalane przez zewnętrzne bodźce wprowadzane na wejścia sterownika i odbierane przez PLC. Ich działanie może być opóźnione na skutek wprowadzonych opóźnień czasowych albo zliczania zewnętrznych lub wewnętrznych zdarzeń. Sterownik programowalny będzie wykonywał pętlę wewnątrz - zdefiniowanego przez użytkownika - programu w oczekiwaniu na sygnał wejściowy lub generował sygnał wyjściowy w określonym programowo czasie.

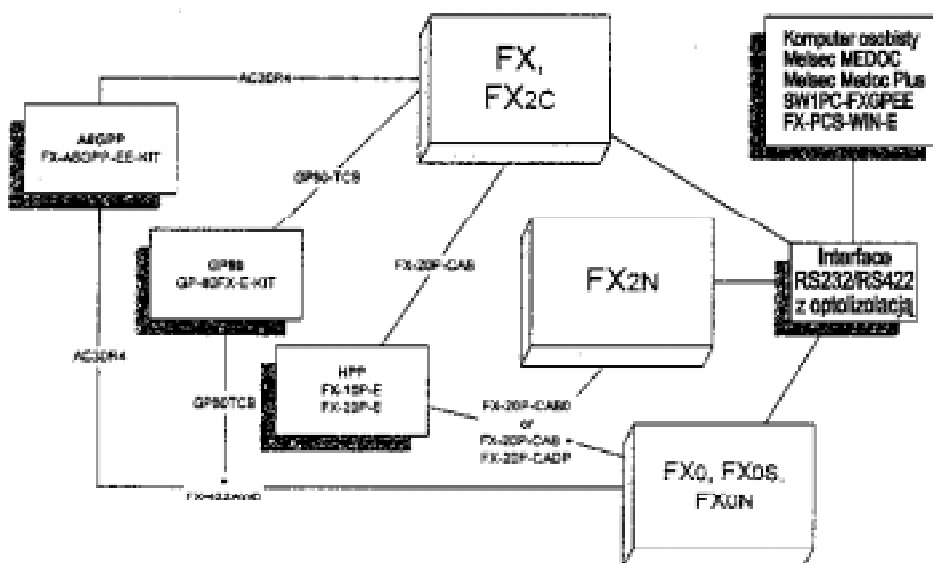
Uwaga dotycząca terminologii

Termin "sterownik programowalny" jest ogólnie używany do określenia wszystkich elementów tworzących system sterujący pod wspólnym mianem. Czasami używa się terminu "sterownik o programowalnej logice" (PLC), na określenie tego samego systemu sterującego. Konstrukcja sterownika może być rozdzielona na poszczególne części. Ten element, gdzie program jest ładowany, przechowywany i wykonywany, jest znany jako Główna Jednostka Przetwarzania (MPU). Inne często używane terminy na określenie tego elementu, to "jednostka główna", "sterownik", "procesor" lub "CPU". Termin CPU jest trochę mylący, ponieważ obecnie bardziej zaawansowane produkty mogą zawierać lokalne CPU. Główne CPU (albo bardziej poprawnie: MPU) zarządza tymi lokalnymi CPU poprzez sieć komunikacyjną albo magistralę.

1.3 Co jest potrzebne do programowania PLC?

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Do zaprogramowania sterowników rodziny Mitsubishi FX dostępne są różnorodne narzędzia. Każde z tych narzędzi może używać instrukcji i ma dostęp do urządzeń wymienionych w tym podręczniku dla określonego sterownika.



1.4 Numery wersji CPU

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------



Mitsubishi przez cały czas unowocześnia i rozszerza możliwości swoich produktów. Ze względu na naturę sterowników programowalnych, które mogą być uważane za "komputery przemysłowe" pojawiają się czasami zmiany wewnątrz głównych CPU (ang. Central Processor Unit). Zmiany te są takie, jakich doświadczyli użytkownicy biurowych i domowych komputerów po wymianie procesora na lepszy. Poniższe listy zawierają wersje CPU, które uległy unowocześnieniu, lub do których zostały dodane nowe funkcje i możliwości.

1.4.1 Wersje CPU FX_{0N}

CPU wersja 1.20 Zostały dodane następujące możliwości:
Programowe sterowanie dla protokołu komunikacji nr 1 i nr 4 poprzez FX_{0N}-485ADP, sieć 1:N

CPU wersja 1.40 Zostały dodane następujące możliwości:
Programowe sterowanie dla komunikacji, wykorzystującej FX_{0N}-485ADP, sieć peer-to-peer (N:N)

1.4.2 Wersje CPU FX i FX_{2C}

CPU wersja 3.07 *Zostały dodane następujące instrukcje:* ASCI (FNC 82), CCD (FNC 84), FTL (FNC 49), HEX (FNC 83), RS (FNC 80), SER (FNC 61), SORT (FNC 69), SQR (FNC 48)

Następujące instrukcje zostały unowocześnione:

EI (FNC 04), BMOV (FNC 15), HSCS (FNC 53), PLSY (FNC 57), FMOV (FNC 16), MEAN (FNC 45), ABSD (FNC 62), DSW (FNC 72), SEGL (FNC 74), PR (FNC 77)

Zostały dodane następujące zakresy urządzeń:

Urządzenia wejścia i wyjścia są adresowalne niezależnie do liczby 256. Ogólna liczba wejść i wyjść (sprzętowych lub programowych) wynosi 256.

Liczbę znaczników zwiększono do 1536 (M0-M1535).

Liczbę rejestrów danych zwiększono do 1000 (D0-D999).

Dodano opcjonalną pamięć 2000 rejestrów plikowych (D6000-D7999).

Zakres etykiet zwiększono do 128 (P0-P127).

CPU wersja 3.11 *Zostały dodane następujące instrukcje:*
PID (FNC 88)

CPU wersja 3.2 Zostały dodane następujące możliwości:
Programowe sterowanie dla protokołu komunikacji nr 4 poprzez FX-485ADP, sieć 1:N

CPU wersja 3.30 Zostały dodane następujące możliwości:
Programowe sterowanie dla protokołu komunikacji nr 1 poprzez FX-485ADP, sieć 1:N

Następujące instrukcje zostały usunięte: ANRD (FNC 91), ADWR (FNC 92), BLK (FNC 97), MCDE (FNC 98), MNET (FNC 90)

1.5 Specjalne wymogi dotyczące wyposażenia programowego

1.5.1 Wersja CPU FX 3.07 lub późniejsza i FX_{2C}

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

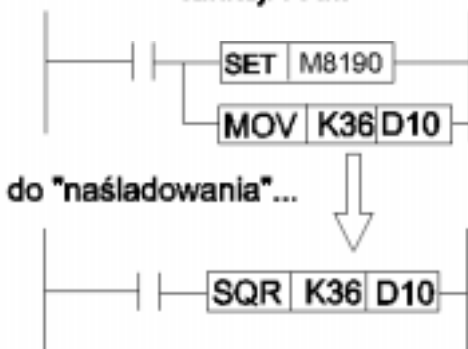


Narzędzia programujące pracujące w starych systemach programowania nie są w stanie wykorzystać nowych możliwości dodanych do CPU od wersji 3.07 (i dostępnych we wszystkich jednostkach FX 2C). Pomimo tego, programowanie niektórych standardowych instrukcji specjalnych w połączeniu ze specjalnymi znacznikami (M) może być tak samo efektywne jak nowe instrukcje. Poniższe tabele pokazują, która wersja oprogramowania peryferyjnego będzie pracować bezpośrednio ze wszystkimi nowymi możliwościami, a które wersje oprogramowania peryferyjnego wymagają użycia zmodyfikowanych instrukcji.

Tabela urządzeń zewnętrznych			
Opis	Numer modelu	Wersja oprogramowania, które...	
		...wymaga użycia znacznika M	...programuje wszystkie instrukcje bezpośrednio
Programator podręczny (HHP)	FX-10P-E	V 1.10	od V 2.00
Kaseta HHP	FX-20P-MFXA-E	V 1.20	od V 2.00
Narzędzia programujące	FX-PCS/AT-E-KIT	V 1.01	od V 2.00
	FX-A6GPP-E-KIT	V 1.00	od V 2.00
Jednostki dostępu do danych	FX-10DU-E	V 1.10	od V 2.00
	FX-20DU-E	V 1.10	od V 2.00
	Inne jednostki DU		od V 1.00

Kombinacje istniejących instrukcji i znaczników M do „naśladowania” określonych instrukcji					
Istniejące instrukcje FX			używane do naśladowania instrukcji		
Mnemonik	Numer FNC	modyfikuje	Instrukcja „naśladowana”	Mnemonik	Numer FNC
MOV	12	M8190	pierwiastek kw.	SQR	48
MOV	12	M8191	zm. przecinek	FLT	49
RAMP	67	M8193	wyszukiw. danej	SER	61
RAMP	67	M8194	instrukcja RS232	RS	80
FMOV	16	M8196	Hex na ASCII	ASCI	82
FMOV	16	M8197	ASCII na Hex	HEX	83
FMOV	16	M8195	Kontrola sum	CCD	84

Wykorzystanie istniejących funkcji FX...



Przykład użycia

Ten format jest wymagany dla poprawnego działania instrukcji. Bezpośrednio przed zmodyfikowaniem instrukcji użytkownik musi zaprogramować instrukcję "naśladowującą" ze zmodyfikowanym znacznikiem M instrukcję SET.



Użycie nowych etykiet przerwań:

Aby programować nowe etykiety przerwań I010 przez I060 dla instrukcji HSCS (FNC 53), posługując się starszym wyposażeniem programowym, należy zamienić następujące specjalne kody M na odpowiedni wskaźnik przerwań. Patrz tabela po prawej stronie

Kombinacje istniejących instrukcji i znaczników M do „naśladowania” operacji etykiet przerwań	
Istniejący znacznik, używany do zamiany danej etykiety przerwania	Etykieta przerwania
M8181	I010
M8182	I020
M8183	I030
M8184	I040
M8185	I050
M8186	I060



Użycie M8198 z instrukcją BMOV:

Ze starym oprogramowaniem i urządzeniami peryferyjnymi rejestry plikowe nie mogą być użyte jako urządzenie docelowe w instrukcji BMOV (FNC 15). Aby przesłać dane instrukcją BMOV do rejestrów plikowych używając starego wyposażenia należy ustawić znacznik M8198. To przełącza parametry źródła i przeznaczenia tzn., że źródło jest traktowane jako przeznaczenie, a przeznaczenie staje się źródłem.



Uwaga ogólna:

Jeżeli używane jest zaktualizowane oprogramowanie lub urządzenia peryferyjne, to nie należy korzystać z technik omówionych w tym dziale, ponieważ normalne operacje zostaną omówione w kolejnych sekcjach.

1.5.2 Wszystkie wersje CPU FX_{2N}

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------



Wprowadzenie nowej wersji CPU zapewnia użytkownikowi FX wiele nowych urządzeń i instrukcji. Aby wykorzystać pełne możliwości jednostek FX_{2N}, użytkownik musi zaktualizować posiadane starsze narzędzia sprzętowe i programowe.

Pomimo to, w wyniku kompatybilności w dół FX_{2N}, nie ma potrzeby aktualizowania istniejących narzędzi programowych do wykorzystania z jednostkami FX_{2N} wobec funkcjonalnej równoważności z jednostką CPU w wersji 3.30.

Tabela urządzeń zewnętrznych		
Opis	Numer modelu	Wersja oprogramowania wspomagającego FX _{2N}
Programator podręczny (HHP)	FX-10P-E	od V 3.00
Kaseta HHP	FX-20P-MFXA-E	od V 3.00
Jednostki dostępu do danych	FX-10-DU-E	od V 4.00
	FX-20-DU-E	tylko dla urządzeń FX
	FX-25-DU-E	od V 2.00
	FX-30-DU-E	od V 3.00
	FX-40-DU-E(S)	tylko dla urządzeń FX
	FX-40-DU-TK-ES	od V 3.00
	FX-50-DU-TK(S)-E	od V 2.10

Notatki

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

2	Podstawowe instrukcje programu	2-1
2.1	Czym jest program?	2-1
2.2	Zarys podstawowych urządzeń używanych w programowaniu	2-1
2.3	Jak czytać schemat drabinkowy?	2-2
2.4	Instrukcje LOAD i LOAD Inverse	2-3
2.5	Instrukcja OUT	2-4
2.5.1	Odmiany timerów i liczników	2-4
2.5.2	Dwukrotne użycie cewki tego samego urządzenia (przełącznika)	2-5
2.6	Instrukcje AND i AND INVERSE	2-6
2.7	Instrukcja OR i OR INVERSE	2-7
2.8	Instrukcja LOAD PULSE, LOAD TRAILING PULSE	2-8
2.9	Instrukcja AND PULSE, AND TRAILING PULSE	2-9
2.10	Instrukcja OR PULSE, OR TRAILING PULSE	2-10
2.11	Instrukcja OR BLOCK	2-11
2.12	Instrukcja AND BLOCK	2-12
2.13	Instrukcje MPS, MRD i MPP	2-13
2.14	Instrukcja MASTER CONTROL i MASTER CONTROL RESET	2-15
2.15	Instrukcja SET i RESET	2-17
2.16	Instrukcja TIMER, COUNTER(OUT i RESET)	2-18
2.16.1	Timery podstawowe, timery z pamięcią i liczniki	2-18
2.16.2	Standardowe liczniki 32-bitowe	2-19
2.16.3	Liczniki szybkie	2-19
2.17	Impulsy narastające i opadające	2-20
2.18	Instrukcja INVERSE	2-21
2.19	Instrukcja NO OPERATION	2-22
2.20	Instrukcja END	2-23

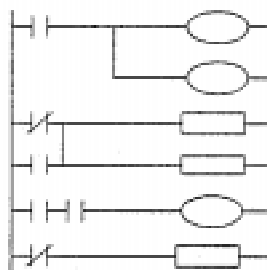
2. Podstawowe instrukcje programu

2.1 Czym jest program?

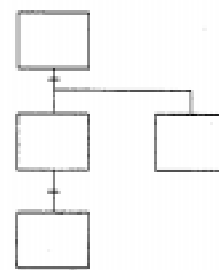
Program jest serią połączonych instrukcji napisanych w języku, który jest zrozumiały przez PLC. Istnieją trzy sposoby przedstawienia programu: lista instrukcji, schemat drabinkowy i schemat blokowy SFC/STL. Nie wszystkie narzędzia programowania mogą pracować we wszystkich formatach programowania. Ogólnie, ręczne panele programatory pracują tylko w formacie instrukcji, natomiast większość graficznych programów narzędziowych akceptuje zarówno instrukcje jak i schematy drabinkowe. Specjalistyczne oprogramowanie pozwala również na skorzystanie z SFC/STL.

```
LD    X10
OUT   Y7
AND   M38
SET   S5
LD    X21
OUT   T01
      K40
```

Format instrukcji



Format drabinkowy



Format SFC

2.2 Zarys podstawowych urządzeń używanych w programowaniu

Istnieje sześć podstawowych programowalnych urządzeń. Każde z nich charakteryzuje się odmiennym sposobem używania. W celu szybkiej i łatwej identyfikacji każdemu urządzeniu przyporządkowano jednoliterowe oznaczenie:

- X: identyfikuje wszystkie bezpośrednie, fizyczne wejścia do PLC.
- Y: identyfikuje wszystkie bezpośrednie, fizyczne wyjścia z PLC.
- T: identyfikuje przekaźniki czasowe - timery, zawarte w PLC.
- C: identyfikuje liczniki, zawarte w PLC.
- M i S: identyfikują wewnętrzne znaczniki binarne PLC.

Są one urządzeniami dwustanowymi, co oznacza, że mogą znajdować się tylko w dwóch stanach: ON lub OFF, 1 lub 0.



Szczegółowe informacje o urządzeniach

Informacje te zawarte są w rozdziale 4. Dla zrozumienia treści bieżącego rozdziału wystarczy powyższe omówienie.

2.3 Jak czytać schemat drabinkowy?

Schemat drabinkowy jest ściśle powiązany ze schematem obwodów sterowania na przekaźnikach. Występują w nim zarówno styki jak i cewki, które mogą być powiązane ze sobą w różnych konfiguracjach. Jednak podstawowe zasady pozostają takie same.

Cewka przekaźnika (przekaźnik) steruje bezpośrednio wyjściem PLC (gdy jest urządzeniem Y) lub steruje wewnętrznym timerem, licznikiem lub znacznikiem (gdy jest urządzeniem T, C, M lub S). Każda cewka ma przyporządkowane styki. Styki są dostępne w konfiguracji "normalnie rozwarne" (NO ang. Normal Open) jak i "normalnie zwarte" (NC ang. Normal Closed).

Określenie "normalnie" odnosi się do stanu styku przekaźnika (nieaktywnego). W przypadku zwykłego przekaźnika (np. Y, M lub S), kiedy cewka jest w stanie OFF (nieaktywna), przez styk "normalnie rozwarne" nie będzie płynął prąd, tzn. obciążenie przezeń podłączone nie będzie aktywne. Natomiast, przez styk "normalnie zwarte" prąd będzie płynął, tak więc podłączone doń obciążenie będzie aktywne.

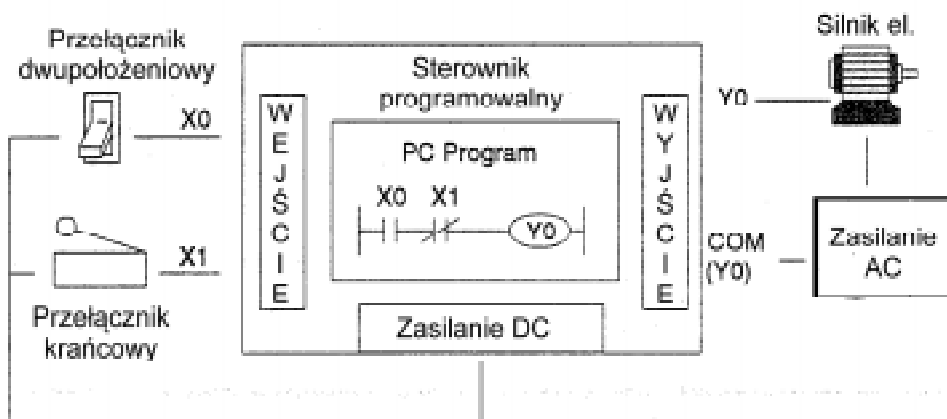
Aktywacja cewki odwraca stan styku, tj. prąd popłynie przez styk "normalnie rozwarne", zaś styk "normalnie zwarte" przerwie przepływ prądu.

Fizyczne wejścia X PLC nie posiadają sterowanych cewek. Urządzenia te mogą być użyte jedynie w formie styku (NO i NC).

Przykład:

Z uwagi na bliską analogię ze schematem przekaźnikowym, programy w formacie drabinkowym mogą być czytane jako przepływ prądu od lewej do prawej pionowej linii zasilającej. Prąd musi przepłynąć przez szereg styków oznaczonych jako X0 i X1 by włączyć cewkę wyjściową Y0.

Jak pokazano na poniższym przykładzie, ustawienie styku X0 w stan ON spowoduje ustawienie wyjścia Y0 także w stan ON. Jeżeli jednak przełącznik krańcowy X1 jest aktywny to wyjście Y0 przechodzi w stan OFF. Tak się dzieje, ponieważ połączenie pomiędzy prawą i lewą linią pionową jest przerwane i prąd pomiędzy nimi nie płynie.

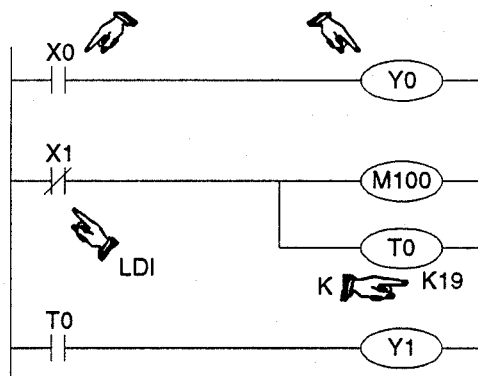


2.4 Instrukcje LOAD i LOAD Inverse

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
LD (LoaD)	Inicjująca operacja logiczna – typ styku NO (normalnie rozwarty)		X, Y, M, S, T, C	1
LDI (LoaD Inverse)	Inicjująca operacja logiczna – typ styku NC (normalnie zwarty)		X, Y, M, S, T, C	1

Przykład programu:



```

0 LD X 0
1 OUT Y 0
2 LDI X 1
3 OUT M 100
4 OUT T 0
   SP K 19
7 LD T 0
8 OUT Y 1

```

Podczas używania programatora ręcznego klawisz spacji musi być wciśnięty, aby umożliwić wprowadzanie stałej.

Najważniejsze informacje do zapamiętania:

- instrukcje LD i LDI należy łączyć bezpośrednio z lewą linią zasilania
- instrukcje LD i LDI służą do zdefiniowania nowego bloku programu gdy użyto instrukcji ORB i ANB (zobacz w następnych rozdziałach)




Instrukcja OUT

- Szczegóły związane z instrukcją OUT (włączając podstawowe odmiany timerów i liczników) są podane na kolejnych stronach.

2.5 Instrukcja OUT

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
OUT (Out)	Końcowa operacja logiczna – sterowanie cewką		Y, M, S, T, C	Y, M : 1 S, M spec: 2 T : 3 C (16 bit) : 3 C (32 bit) : 5

Najważniejsze informacje do zapamiętania:

- instrukcję OUT należy łączyć bezpośrednio z prawą linią zasilania
- instrukcja OUT nie może być użyta do sterowania urządzeń wejściowych typu "X"
- instrukcje OUT mogą być łączone równolegle (patrz na poprzednią stronę; konfiguracja M100/T0)

2.5.1 Odmiany timerów i liczników

Podczas konfigurowania instrukcji OUT do pracy w charakterze timera T albo licznika C musi być wprowadzona stała. Stała ta jest zdefiniowana literą K (zobacz przykład z poprzedniej strony; T0 K19).

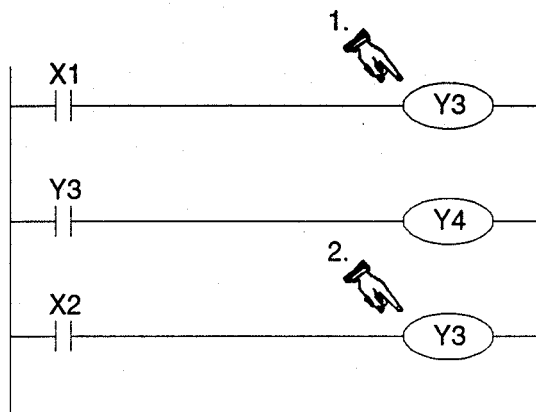
W przypadku timera stała K zawiera daną określającą czas opóźnienia pomiędzy momentem pobudzenia a momentem uaktywnienia jego styku. Jeśli timer z krokiem 100 ms zawiera stałą "K100" to upłynie (100 X 100 ms) 10 s zanim uaktywniony zostanie styk timera.

W przypadku licznika stała definiuje ile impulsów musi zliczyć licznik, aby został uaktywniony styk licznika. Np. licznik ze stałą 8 musi być pobudzony 8 razy zanim styk licznika zostanie w końcu uaktywniony.

Następująca tabela określa niektóre podstawowe parametry dla różnych timerów i liczników.

Zegar/Licznik	Ustawienie stałej K	Ustawienie bieżące	Kroki programu
Timer 1 ms	1 do 32767	0,001 do 32,767 s	3
Timer 10 ms		0,01 do 327,67 s	
Timer 100 ms		0,1 do 3276,7 s	
Licznik 16-bitowy	1 do 32.767	1 do 32.767	5
Licznik 32-bitowy	-2.147.483.648 do 2.147.483.647	-2.147.483.648 do 2.147.483.647	

2.5.2 Dwukrotne użycie cewki tego samego urządzenia (przełącznika)



Na ogół błędem jest dwukrotne użycie cewki tego samego urządzenia. Wielokrotne użycie cewek wyjściowych tego samego urządzenia może spowodować, że wynik operacji programu stanie się nieprzewidywalny. W przykładzie obok pokazana została sytuacja dwukrotnego użycia cewki; są tu dwa wyjścia Y3. Pojawi się następująca sekwencja zdarzeń, kiedy wejścia X1=ON i X2=OFF:

1. Pierwsze Y3 przechodzi w stan ON, ponieważ X1 jest w stanie ON. Styki połączone z Y3 zostają zasilone, kiedy cewka wyjścia Y3 zostaje zasilona. Stąd wyjście Y4 przechodzi w stan ON

2. Ostatnia i najważniejsza linia w tym programie

sprawdza stan wejścia X2. Jeśli nie jest ono w stanie ON to druga cewka Y3 nie uaktywnia się. Tak więc stan cewki Y3 aktualizuje się dla odzwierciedlenia nowej sytuacji, tzn. ustawia się w pozycji OFF. Ostatecznie stany wyjść są następujące: Y3=OFF, Y4=ON



Dwukrotne użycie cewki:

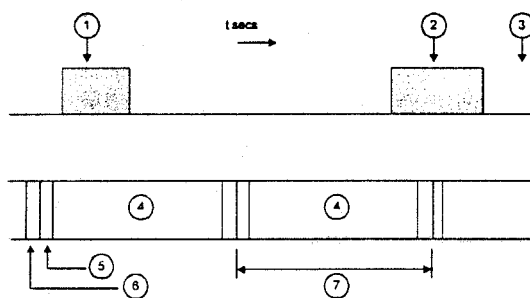
- Zawsze należy sprawdzić program aby wyeliminować występowanie dwukrotnego użycia tej samej cewki. Jeżeli ono wystąpi, program nie będzie pracował zgodnie z oczekiwaniem - może doprowadzić do nieprzewidywalnego fizycznego zagrożenia ze względu na złe sterowanie obiektu.



Efekt ostatniego pobudzenia wyjścia:

- W przypadku dwukrotnego użycia tej samej cewki efektywna jest ostatnia operacja, tj. wynik ostatniej operacji w programie dotyczącej programowania danej cewki będzie określał stan danego urządzenia.

Czas trwania sygnału wejściowego:



Czas trwania stanu ON lub OFF na wejściu PLC musi być dłuższy niż czas cyklu programu PLC. Biorąc pod uwagę 10 ms standardowy filtr wejściowy jako opóźnienie odpowiedzi wejścia, czas trwania stanu ON/OFF musi być dłuższy niż 20 ms, jeśli cykl programu wynosi 10 ms. Wobec tego w przykładzie obok impulsy wejściowe o częstotliwości 25 Hz ($1 \text{ s} / (20 \text{ ms ON} + 20 \text{ ms OFF})$) nie są rozróżniane. Istnieją instrukcje specjalne, umożliwiające obsługę tak szybkich sygnałów wejściowych.

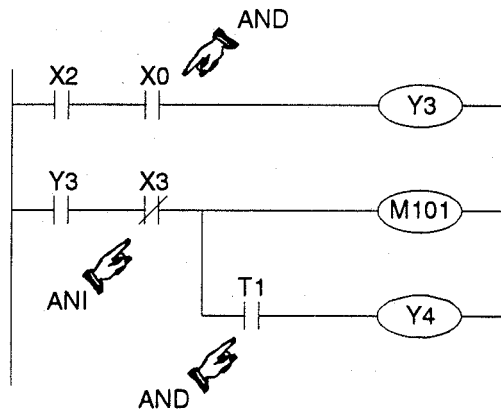
- ①: stan ON wejścia nierozpoznany
- ②: stan ON wejścia rozpoznany
- ③: stan OFF wejścia nierozpoznany
- ④: 1. przebieg programu
- ⑤: przetwarzanie wejścia
- ⑥: przetwarzanie wyjścia
- ⑦: pełny cykl przebiegu (skanu) programu

2.6 Instrukcje AND i AND INVERSE

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
AND (AND)	połączenie szeregowo styków NO (normalnie otwartych)		X, Y, M, S, T, C	1
ANI (ANd Inverse)	połączenie szeregowo styków NC (normalnie zwartych)		X, Y, M, S, T, C	1

Przykład programu:



	0	LD	X	2
	1	AND	X	0
	2	OUT	Y	3
	3	LD	Y	3
	4	ANI	X	3
	5	OUT	M	101
	6	AND	T	1
	7	OUT	Y	4

Najważniejsze informacje do zapamiętania:

- Instrukcje AND i ANI należy używać do szeregowego łączenia styków. Szeregowo można łączyć dowolną liczbę styków (zobacz następny punkt "Ograniczenia zewnętrzne")
- Cewka sterowana poprzez styk podłączony do początkowej instrukcji OUT nazywa się wyjściem warunkowym (zobacz program powyżej OUT Y4; patrz również p. 2.13). Wyjścia warunkowe można stosować wielokrotnie, tak długo, jak zachowany jest właściwy porządek połączeń.



Ograniczenia zewnętrzne:



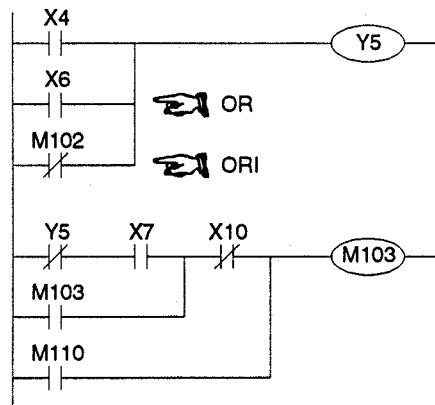
- Sterownik programowalny nie ma ograniczeń programowych jeśli chodzi o liczbę styków połączonych szeregowo bądź równolegle. Istnieją jednak ograniczenia sprzętowe, dotyczące paneli programatorów ręcznych, ekranu prezentacji i jego wydruku. Zaleca się aby każda linia lub szczebel schematu drabinkowego zawierał nie więcej niż 10 styków i 1 cewkę. Liczba wyjść warunkowych nie powinna przekraczać 24.

2.7 Instrukcja OR i OR INVERSE

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
OR (OR)	połączenie równoległe styków NO (normalnie otwartych)		X, Y, M, S, T, C	1
ORI (OT Inverse)	połączenie równoległe styków NC (normalnie zwartych)		X, Y, M, S, T, C	1

Przykład programu:



```

0 LD X 4
1 OR X 6
2 ORI M 102
3 OUT Y 5
4 LDI Y 5
5 AND X 7
6 OR M 103
7 ANI X 10
8 OR M 110
9 OUT M 103

```

Najważniejsze informacje do zapamiętania:

- instrukcji OR i ORI należy używać do równoległego łączenia styków. Równoległe łączenie bloku styków, zawierającego co najmniej dwa szeregowo połączone styki do innego bloku drabinki, wymaga użycia instrukcji ORB.
- instrukcje OR i ORI należy łączyć bezpośrednio z lewą linią zasilania

Ograniczenia zewnętrzne:



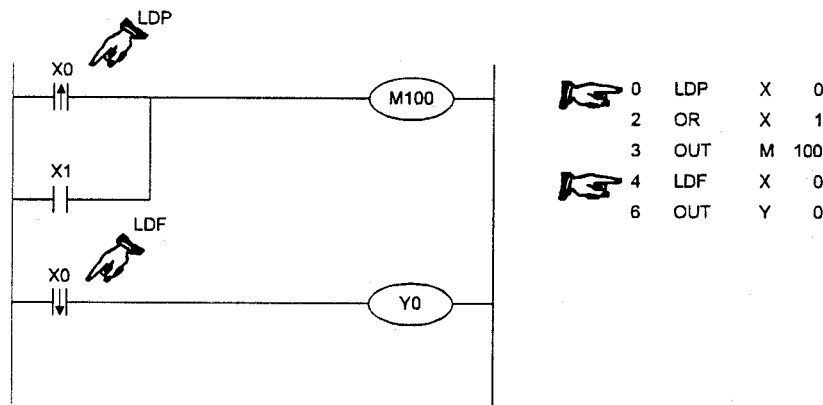
- Sterownik programowalny nie ma ograniczeń programowych jeśli chodzi o liczbę styków połączonych szeregowo bądź równoległe. Istnieją jednak ograniczenia sprzętowe, dotyczące paneli programatorów ręcznych, ekranu prezentacji i jego wydruku. Zaleca się aby każda linia lub szczebel schematu drabinkowego zawierał nie więcej niż 10 styków i 1 cewkę. Liczba wyjść warunkowych nie powinna przekraczać 24.

2.8 Instrukcja LOAD PULSE, LOAD TRAILING PULSE

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
LDP (LoaD Pulse)	Inicjująca operacja logiczna – narastające zbocze impulsu		X, Y, M, S, T, C	2
LDF (LoaD Falling pulse)	Inicjująca operacja logiczna – opadające zbocze impulsu		X, Y, M, S, T, C	1

Przykład programu:



Najważniejsze informacje do zapamiętania:

- instrukcje LDP i LDF należy łączyć bezpośrednio z lewą linią zasilania
- instrukcji LDP i LDF należy użyć do zdefiniowania nowego bloku programu podczas używania instrukcji ORB i ANB (zobacz następne rozdziały)
- instrukcja LDP działa w trakcie jednego przebiegu programu, po tym jak stan skojarzonego urządzenia zmieni się z OFF na ON
- instrukcja LDF działa w trakcie jednego przebiegu programu, po tym jak stan skojarzonego urządzenia zmieni się z ON na OFF



Znaczniki pojedynczej operacji M2800-M3071:



- Gdy instrukcje operacji impulsowej są używane ze znacznikami M2800-M3071, aktywują tylko pierwszą instrukcję programu, poza jego miejscem, w którym nastąpiła zmiana stanu urządzenia. Wszystkie inne instrukcje operacji impulsowych pozostaną nieaktywne.
- Jest to przydatne w programach STL (zobacz rozdział 3) do wykonania operacji krokowych przy użyciu pojedynczego urządzenia.
- Wszystkie inne instrukcje (LD, AND, OR itd.) będą działały normalnie.



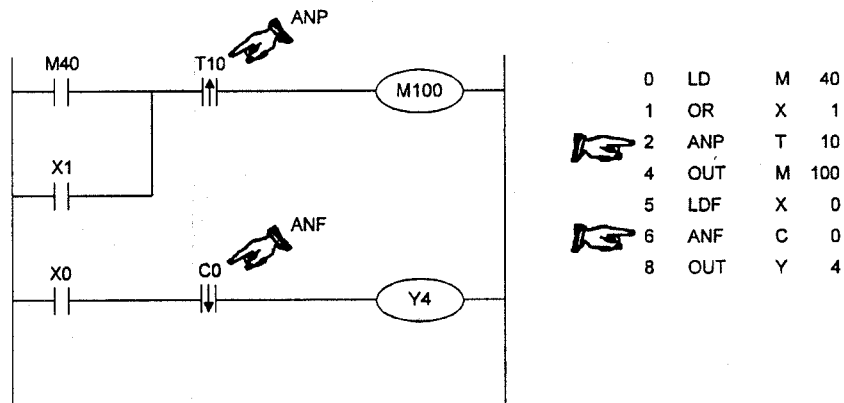
Więcej szczegółów na ten temat na stronie 4-5.

2.9 Instrukcja AND PULSE, AND TRAILING PULSE

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
ANP (ANd Pulse)	połączenie szeregowe narastających zboczy impulsów		X, Y, M, S, T, C	2
ANF (ANd Falling pulse)	połączenie szeregowe opadających zboczy impulsów		X, Y, M, S, T, C	2

Przykład programu:



Najważniejsze informacje do zapamiętania:

- Instrukcji ANDP i ANDF należy używać do szeregowego łączenia styków impulsowych
- Ich użycie jest identyczne jak dla instrukcji AND i ANI; zobacz wcześniej.
- Instrukcja ANP działa w trakcie jednego przebiegu programu, po tym jak stan skojarzonego urządzenia zmieni się z OFF na ON
- instrukcja ANF działa w trakcie jednego przebiegu programu, po tym jak stan skojarzonego urządzenia zmieni się z ON na OFF



Znaczniki pojedynczej operacji M2800-M3071:

- Gdy instrukcje są używane ze znacznikami M2800-M3071, aktywują tylko pierwszą kolejną instrukcję programu.

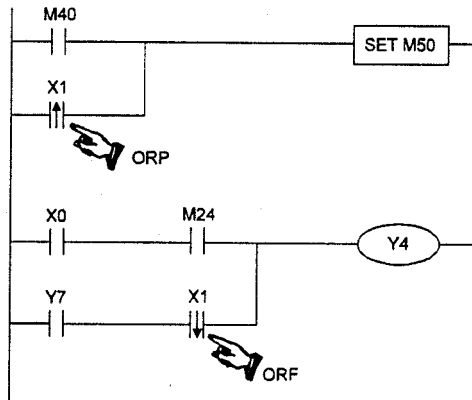
Więcej szczegółów na ten temat na stronie 2-8.

2.10 Instrukcja OR PULSE, OR TRAILING PULSE

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
ORP (OR Pulse)	połączenie równoległe narastających zboczy impulsów		X, Y, M, S, T, C	2
ORF (OR Falling Pulse)	połączenie równoległe opadających zboczy impulsów		X, Y, M, S, T, C	2

Przykład programu:



```

0 LD M 40
1 ORP X 1
3 SET M 50
4 LD X 0
5 AND M 24
6 LD Y 7
7 ORF X 1
9 ORB
10 OUT Y 4

```

Najważniejsze informacje do zapamiętania:

- Instrukcji ORP i ORF należy używać do równoległego łączenia styków impulsowych
- Ich użycie jest identyczne jak dla instrukcji OR i ORI; zobacz wcześniej.
- instrukcja ORP działa w trakcie jednego przebiegu programu, po tym jak stan skojarzonego urządzenia zmieni się z OFF na ON
- instrukcja ORF działa w trakcie jednego przebiegu programu, po tym jak stan skojarzonego urządzenia zmieni się z ON na OFF



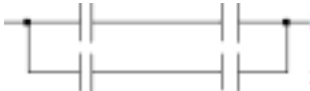
Znaczniki pojedynczej operacji M2800-M3071:

- Gdy instrukcje są używane ze znacznikami M2800-M3071, aktywują tylko pierwszą kolejną instrukcję programu.

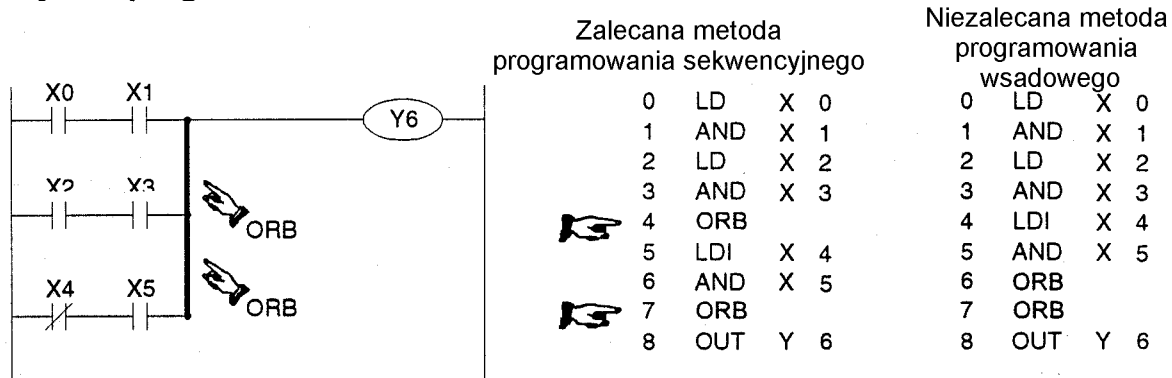
Więcej szczegółów na ten temat na stronie 2-8.

2.11 Instrukcja OR BLOCK

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
ORB (OR Block)	połączenie równoległe obwodów wielostykowych		brak	1

Przykład programu:



Najważniejsze informacje do zapamiętania:

- Instrukcja ORB jest instrukcją niezależną i nie jest kojarzona z żadnym urządzeniem
- Należy używać instrukcji ORB do równoległego łączenia obwodów wielostykowych - zwykle bloki obwodów szeregowych, równoległe z poprzedzającym układem. Blok obwodu szeregowego to taki, w którym połączone są szeregowo co najmniej dwa styki, ale w którym użyta jest instrukcja ANB.
- Do zadeklarowania styku początkowego bloku (obwodu) używa się instrukcji LD lub LDI. Po skompletowaniu bloku należy połączyć go równoległe z poprzednim blokiem, używając instrukcji ORB.



Ograniczenia przetwarzania wsadowego

- Używając instrukcji ORB w przetwarzaniu wsadowym (sumowanie wszystkich bloków odbywa się po ich zdefiniowaniu) należy upewnić się, że nie więcej niż 8 instrukcji LD i LDI jest użytych do definicji bloków programu (połączonych równoległe)

Nieprzestrzeganie tego prowadzi do błędu programu (zobacz prawy wydruk programu)

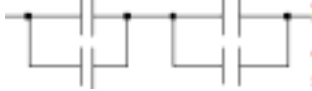


Ograniczenia przetwarzania sekwencyjnego

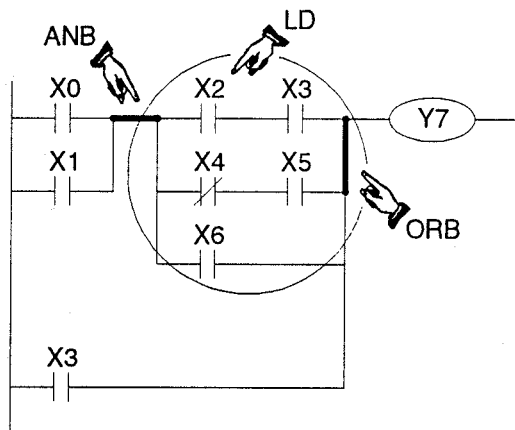
- Nie ma ograniczeń w ilości równoległych obwodów przy użyciu instrukcji ORB w procesie przetwarzania sekwencyjnego. (zobacz lewy wydruk programu)

2.12 Instrukcja AND BLOCK

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
ANB (ANd Block)	połączenie szeregowo obwodów wielostykowych		brak	1

Przykład programu:



Zalecana metoda programowania sekwencyjnego

0	LD	X	0
1	OR	X	1
2	LD	X	2
3	AND	X	3
4	LDI	X	4
5	AND	X	5
6	ORB		
7	OR	X	6
8	ANB		
9	OR	X	3
10	OUT	Y	7

Najważniejsze informacje do zapamiętania:

- Instrukcja ANB jest instrukcją niezależną i nie jest kojarzona z żadnym urządzeniem.
- Instrukcji ANB należy używać do połączenia obwodów wielostykowych - zwykle bloków obwodów równoległych. Bloki obwodów równoległych to takie, w których jeden z nich zawiera co najmniej dwa szeregowo połączone styki lub te, w których użyta jest instrukcja ORB.
- Do zadeklarowania styku początkowego bloku (obwodu) używa się instrukcji LD lub LDI. Po skompletowaniu bloku należy połączyć go szeregowo z poprzednim blokiem, używając instrukcji ANB.



Ograniczenia przetwarzania wsadowego

- Używając instrukcji ANB w przetwarzaniu wsadowym należy upewnić się, że nie więcej niż 8 instrukcji LD i LDI jest użytych do definicji bloków programu (połączonych równolegle). Nieprzestrzeganie tego prowadzi do błędu programu (zobacz wyjaśnienie instrukcji ORB w programie)

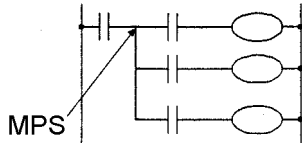
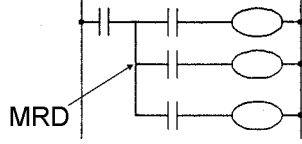
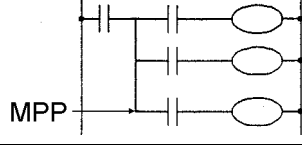


Ograniczenia przetwarzania sekwencyjnego

- Instrukcja ANB może być użyta zawsze, gdy konieczne jest szeregowo połączenie kilku obwodów równoległych do poprzedzających je bloków. (zobacz wydruk programu)

2.13 Instrukcje MPS, MRD i MPP

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
MPS (Point Store)	zapamiętuje bieżący wynik wewnętrznych operacji PLC		brak	1
MRD (Read)	odczytuje bieżący wynik wewnętrznych operacji PLC		brak	1
MPP (PoP)	odtworza i usuwa bieżący wynik wewnętrznych operacji PLC		brak	1

Najważniejsze informacje do zapamiętania:

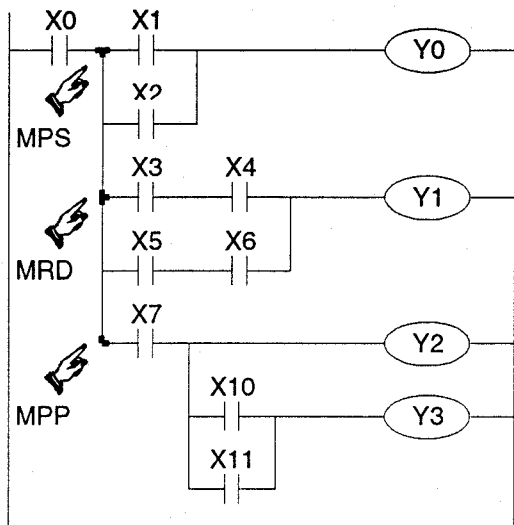
- Instrukcje te są używane do łączenia cewki wyjściowej z lewą stroną styku. Bez tych instrukcji połączenie może być wykonane tylko z prawą stroną ostatniego styku
- Instrukcja MPS przechowuje wartość logiczną (stan) punktu połączenia obwodu drabinki i dlatego gałąź cewki może przywołać tę wartość później
- Instrukcja MRD odczytuje poprzednio zapamiętaną wartość logiczną punktu przyłączenia w celu dołączenia nowej gałęzi
- Instrukcja MPP odtwarza i usuwa zapamiętaną wartość logiczną punktu przyłączenia. Po pierwsze dołącza następny styk, a następnie usuwa punkt z pamięci tymczasowej.
- Każdej instrukcji MPS musi odpowiadać instrukcja MPP
- Ostatni styk lub obwód cewki musi łączyć się z instrukcją MPP
- Na każdym etapie programu liczba aktywnych par MPS-MPP nie może być większa niż 11.



Użycie MPS, MRD i MPP

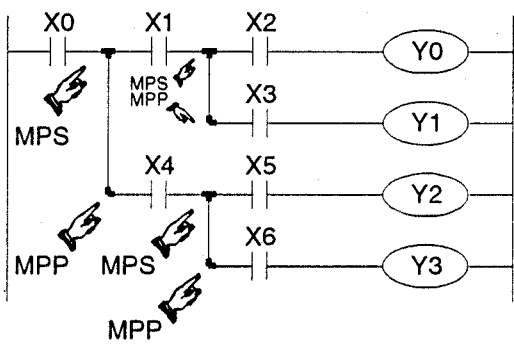
- Podczas pisania programu w formacie drabinkowym, narzędzia programowe automatycznie dodają wszystkie instrukcje MPS, MRD i MPP w czasie konwersji programu. Instrukcje te są widoczne podczas przeglądania wygenerowanego programu.
- Podczas pisania programu w formacie instrukcji, użytkownik musi sam zadbać o wprowadzenie w odpowiednie miejsca instrukcji MPS, MRD i MPP.

Przykłady programów:



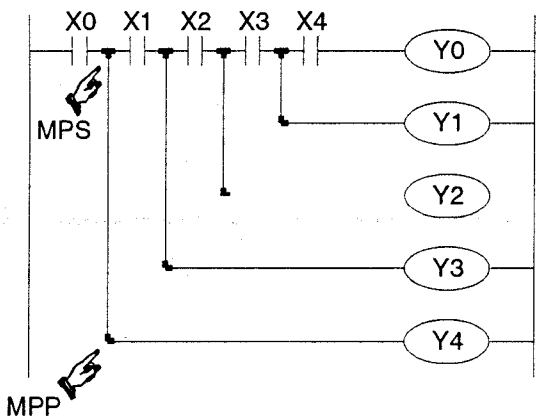
```

0 LD X 0
1 MPS
2 LD X 1
3 OR X 2
4 ANB
5 OUT Y 0
6 MRD
7 LD X 3
8 AND X 4
9 LD X 5
10 AND X 6
11 ORB
12 ANB
13 OUT Y 1
14 MPP
15 AND X 7
16 OUT Y 2
17 LD X 10
18 OR X 11
19 ANB
20 OUT Y 3
    
```



```

0 LD X 0
1 MPS
2 AND X 1
3 MPS
4 AND X 2
5 OUT Y 0
6 MPP
7 AND X 3
8 OUT Y 1
9 MPP
10 AND X 4
11 MPS
12 AND X 5
13 OUT Y 2
14 MPP
15 AND X 6
16 OUT Y 3
    
```





```

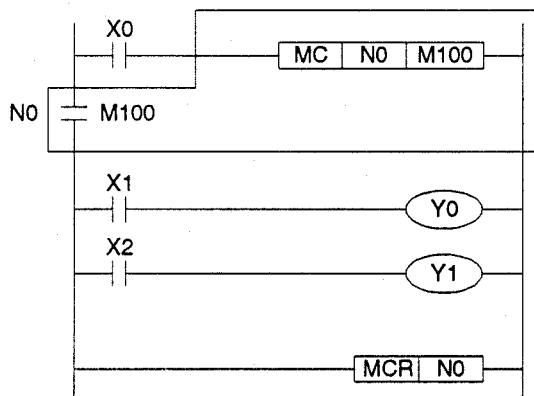
0 LD X 0
1 MPS
2 AND X 1
3 MPS
4 AND X 2
5 MPS
6 AND X 3
7 MPS
8 AND X 4
9 OUT Y 0
10 MPP
11 OUT Y 1
12 MPP
13 OUT Y 2
14 MPP
15 OUT Y 3
16 MPP
17 OUT Y 4
    
```

2.14 Instrukcja MASTER CONTROL i MASTER CONTROL RESET

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
MC (Master Control)	zaznacza początek bloku sterującego master		Y, M (cewki specjalne M niedozwolone), N oznacza poziom zagnieżdżenia (N0 do N7)	3
MCR (Master Control Reset)	zaznacza koniec bloku sterującego master		N oznacza poziom zagnieżdżenia (N0 do N7) do wyzerowania	2

Przykład programu:



```

0 LD X 0
1 MC N 0
  SP M 100
4 LD X 1
5 OUT Y 0
6 LD X 2
7 OUT Y 1
8 MCR N 0

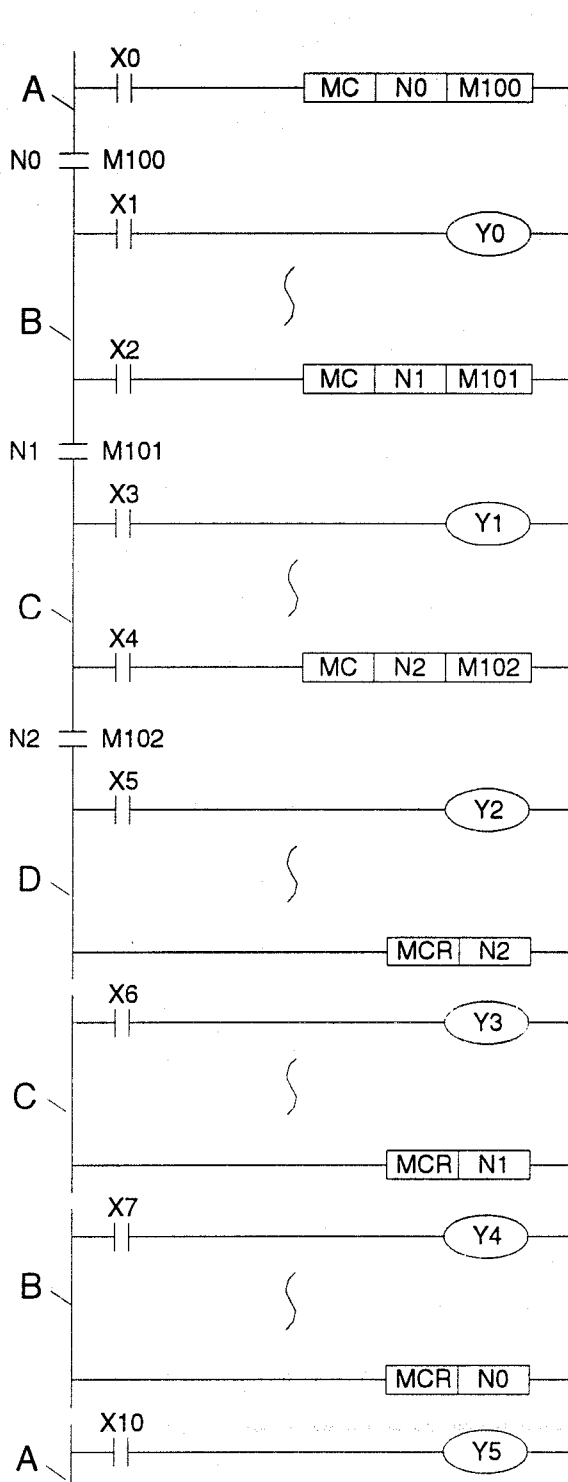
```

Uwaga: SP - klawisz spacji

N - Poziom zagnieżdżenia MC (N0 do N7)

Najważniejsze informacje do zapamiętania:

- Po wykonaniu instrukcji MC lewa linia zasilania przesuwana jest do punktu programu za instrukcją MC. Instrukcja MCR przywraca poprzednie położenie linii.
- Instrukcja MC zawiera również wskaźnik poziomu zagnieżdżenia N. Poziomy zagnieżdżenia mieszczą się w zakresie N0 do N7 (8 punktów). Najwyższy poziom to "0", najniższy - "7"
- Instrukcja MCR zeruje dany poziom zagnieżdżenia. Wyzerowanie danego poziomu zagnieżdżenia powoduje również zerowanie wszystkich niższych poziomów. Np. MCR N5 zeruje poziomy 5 do 7
- Gdy wejście X0 jest w stanie ON, wszystkie instrukcje zawarte pomiędzy instrukcjami MC i MCR wykonują się
- Gdy wejście X0 jest w stanie OFF, żadna z instrukcji zawartych pomiędzy instrukcjami MC i MCR nie wykonuje się; zerowane są wówczas wszystkie urządzenia z wyjątkiem timerów z pamięcią, liczników i urządzeń sterowanych przez instrukcje SET/RST
- Instrukcje MC mogą być używane dowolną ilość razy, zmieniając numer urządzenia Y i M. Użycie tego samego numeru urządzenia dwa razy jest traktowane jako przypadek dwukrotnego użycia tej samej cewki (zobacz rozdział 2.5.2). Poziomy zagnieżdżenia mogą być zdublowane, ale kiedy dany poziom zagnieżdżenia jest wyzerowany, wszystkie wystąpienia tego poziomu są zerowane, a nie tylko ten występujący w lokalnej instrukcji MC.



Przykład programu z zagnieżdżoną instrukcją MC:

Poziom N0: linia B aktywuje się, gdy X0 jest w stanie ON

Poziom N1: linia C aktywuje się, gdy X0 i X2 są w stanie ON

Poziom N2: linia D aktywuje się, gdy X0, X2 i X4 są w stanie ON

Poziom N1: instrukcja MCR N2 otwiera linię C. Jeśli MCR wyzeruje poziom N0, wtedy aktywna będzie linia A, ponieważ wszystkie poziomy poniżej poziomu 0 będą wyzerowane

Poziom N0: MCR N1 otwiera linię B

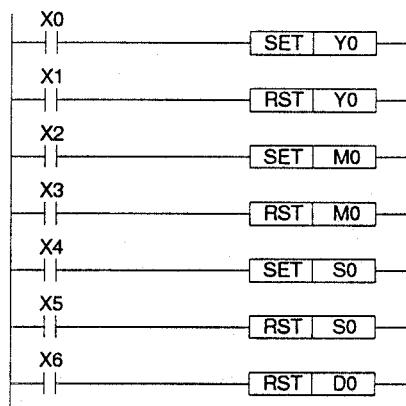
Stan inicjujący: MCR N0 otwiera linię inicjującą A. Wyjście Y5 przełącza stany ON/OFF zgodnie ze stanem ON/OFF X10, niezależnie od stanu ON/OFF wejść X0, X2 lub X4

2.15 Instrukcja SET i RESET

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
SET (SET)	ustawia bit urządzenia w stan ON		Y, M, S	Y, M : 1 S, specjalny M: 2
RST (ReSeT)	ustawia bit urządzenia w stan OFF		Y, M, S, D, V, Z (patrz sekcja 2.16 – timery i liczniki T, C)	D, rejestry specjalne D, V i Z : 3

Przykład programu:



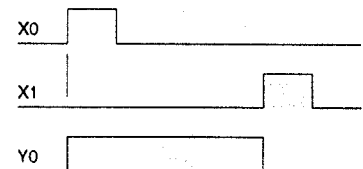
```

0 LD X 0
1 SET Y 0
2 LD X 1
3 RST Y 0
4 LD X 2
5 SET M 0
6 LD X 3
7 RST M 0
8 LD X 4
9 SET S 0
10 LD X 5
11 RST S 0
12 LD X 6
13 RST D 0

```

Najważniejsze informacje do zapamiętania:

- Ustawienie X0 w stan ON powoduje ustawienie Y0 w stan ON; Y0 pozostaje w stanie ON nawet wtedy, gdy X0 przejdzie w stan OFF
- Ustawienie X1 w stan ON powoduje przejście Y0 w stan OFF; Y0 pozostaje w stanie OFF nawet wtedy, gdy X1 przejdzie w stan OFF
- Instrukcje SET i RST mogą być użyte dla tego samego urządzenia tyle razy ile jest to konieczne, z tym, że ostatnio aktywowana instrukcja określa bieżący jego stan.
- Można również używać instrukcji RST do wyzerowania zawartości urządzeń danych, takich jak rejestry danych, rejestry indeksowe itd. Efekt przypomina przesłanie stałej "K0: do urządzenia danych.





Zerowanie timerów i liczników

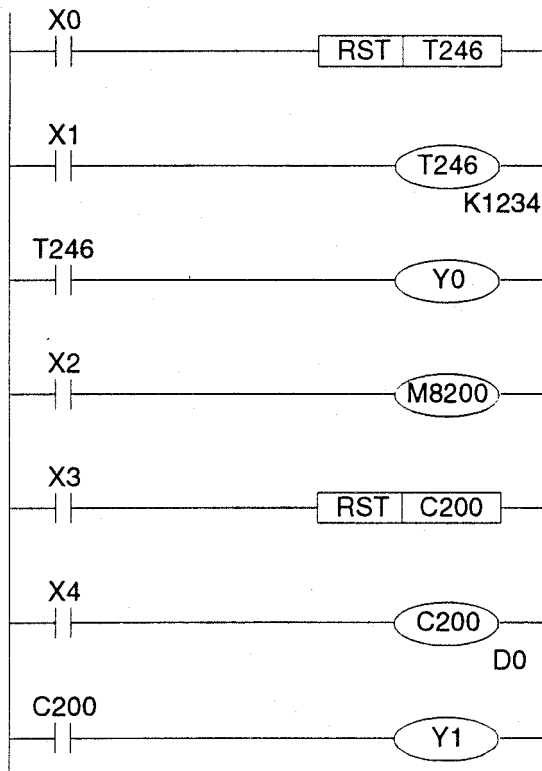
- Zobacz stronę 2-18

2.16 Instrukcja TIMER, COUNTER (OUT i RESET)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
OUT (OUT)	sterowanie cewką timera lub licznika		T, C	liczniki 32-bitowe: 5 inne: 3
RST (ReSeT)	zeruje styki i bieżące wartości timera i licznika		X, Y, M, S, T, C	1

Przykład programu:



2.16.1 Timery podstawowe, timery z pamięcią i liczniki

Urządzenia te mogą być zerowane w każdej chwili przy użyciu instrukcji RST (z numerem zerowanego urządzenia).

W czasie zerowania wszystkie czynne styki cewki i rejestry bieżącej wartości są zerowane dla wybranego urządzenia. W przykładzie T246 (1ms timer z pamięcią) jest aktywowany, gdy X1 jest w stanie ON. Kiedy bieżąca wartość T246 osiąga ustaloną wartość "K", tj. 1234, cewka timera T246 zostanie aktywowana. To ustawia styk "normalnie rozarty" w stan ON. Stąd Y0 przechodzi w stan ON. Ustawienie X0 w stan ON zeruje timer T246 w sposób opisany powyżej. Ponieważ styki T246 są wyzerowane, wyjście Y0 przejdzie w stan OFF.



Timery z pamięcią:
Więcej szczegółów na ten temat na stronie 4-17

2.16.2 Standardowe liczniki 32-bitowe

Licznik 32-bitowy C200 liczy (w górę lub w dół) zgodnie z ustawieniem M8200 w stan ON/OFF. W przykładzie programu pokazanym na poprzedniej stronie, C200 jest użyty do zliczania liczby cykli OFF->ON na wejściu X4

Styk wyjścia jest ustawiany (lub zerowany w zależności od kierunku liczenia), do momentu osiągnięcia wartości równej (w tym przykładzie) zawartości rejestrów danych D1, D0 (do licznika 32-bitowego potrzebna jest 32-bitowa dana)

Styk wyjścia jest zerowany i bieżąca wartość licznika jest wyzerowana do "0", kiedy wejście X3 znajduje się w stanie ON.



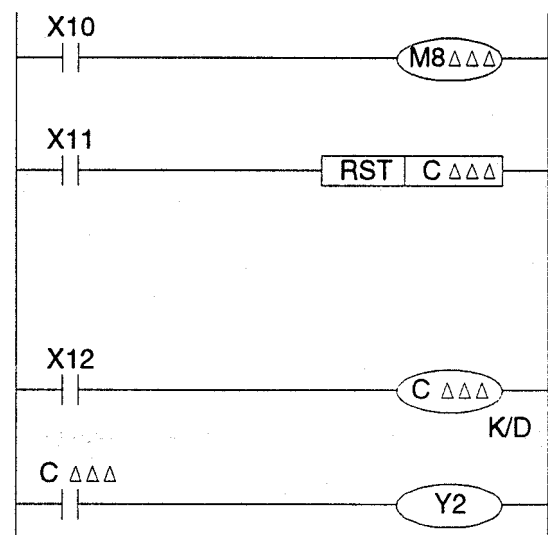
Liczniki 32-bitowe:

- Więcej szczegółów na ten temat na stronie 4-21

2.16.3 Liczniki szybkie

Liczniki szybkie mają wybierany kierunek zliczania. Kierunki są wybierane przez sterowanie odpowiednimi specjalnymi znacznikami M. W przykładzie pokazanym obok licznik działa w następujący sposób; kiedy X10 jest w stanie ON, rozpoczyna się liczenie w dół. Kiedy X10 jest w stanie OFF, rozpoczyna się liczenie w górę.

W przykładzie styki licznika C $\Delta\Delta\Delta$ i jego bieżąca wartość są wyzerowane, kiedy X11 jest w stanie ON. Kiedy X12 jest w stanie ON, sterowany licznik jest włączony. To znaczy, że będzie on w stanie rozpocząć zliczanie związanych z nim sygnałów wejściowych (nie będzie to X12 - liczniki szybkie mają przypisane specjalne sygnały wejściowe, zobacz stronę 4-22)



Dostępność urządzeń:

Nie wszystkie opisane tutaj urządzenia są dostępne we wszystkich sterownikach. Zakresy aktywnych urządzeń mogą się różnić na różnych PLC. Przed użyciem należy sprawdzić specyfikację tych urządzeń na wybranym PLC.

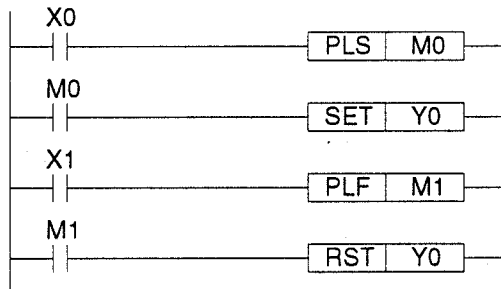
Więcej informacji na temat szybkich liczników zawiera strona 4-22. Zobacz też rozdział 8 - zakresy urządzeń PLC.

2.17 Impulsy narastające i opadające

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
PLS (PuLSe)	narastające zbocze impulsu		Y, M (specjalne M niedozwolone)	2
PLF (PuLse Falling)	opadające zbocze impulsu		Y, M (specjalne M niedozwolone)	2

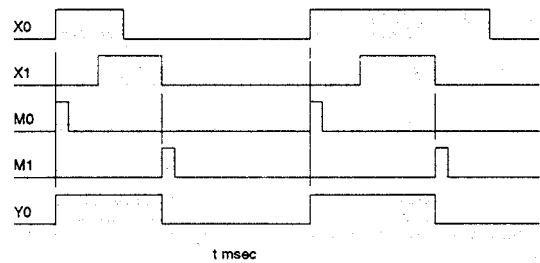
Przykład programu:



	0	LD	X	0
	1	PLS	M	0
	3	LD	M	0
	4	SET	Y	0
	5	LD	X	1
	6	PLF	M	1
	8	LD	M	1
	9	RST	Y	0

Najważniejsze informacje do zapamiętania:

- Gdy wykonywana jest instrukcja PLS, urządzenia obiektowe Y i M zostają uaktywnione na czas jednego cyklu operacyjnego po tym, jak sterujący sygnał wejściowy przyjął stan ON.
- Gdy wykonywana jest instrukcja PLF, urządzenia obiektowe Y i M zostają uaktywnione na czas jednego cyklu operacyjnego po tym, jak sterujący sygnał wejściowy przyjął stan OFF.
- Gdy status PLC zmienił się z RUN na STOP i znów na RUN, przy stałym sygnale wejściowym ON, instrukcja PLS M0 jest wykonana ponownie. Jeżeli jednak użyto znacznika z podtrzymaniem M (zatraskowego) zamiast M0, instrukcja nie wykona się. Aby urządzenie z podtrzymaniem mogło być ponownie wyzwolone, jego wejściowy sygnał sterujący (np. X0) musi zmienić stan na OFF w czasie trwania sekwencji RUN/STOP/RUN.

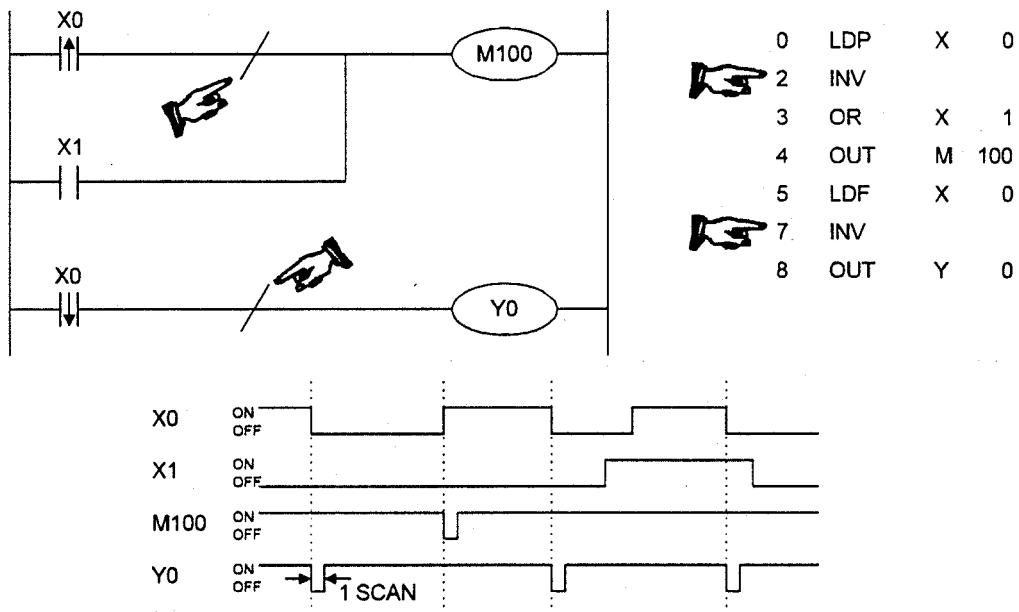


2.18 Instrukcja INVERSE

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
INV (Inverse)	odwraca bieżący wynik wewnętrznych operacji PLC		brak	1

Przykład programu:



Najważniejsze informacje do zapamiętania:

- Instrukcja INV używana jest do zmiany (odwrócenia) stanu logicznego określonego punktu sieci drabinkowej.
- Użycie jest identyczne jak dla instrukcji AND i ANI; zobacz wcześniej.



Zastosowania dla INV:

Tej instrukcji należy używać w celu szybkiej zmiany wartości logicznej złożonego obwodu. Instrukcja ta jest także przydatna razem z instrukcjami impulsowymi LDP, LDF, ANP, itd.

2.19 Instrukcja NO OPERATION

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------


Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
NOP (No OPeration)	brak operacji lub krok zerowy	brak	brak	1

Najważniejsze informacje do zapamiętania:

- Umieszczanie instrukcji NOP wewnątrz programu minimalizuje zakres prac związanych ze zmianą lub edycją programu.
- Istnieje możliwość zmiany działania obwodu poprzez zamianę zastosowanych instrukcji na instrukcje NOP.
- Zamiana instrukcji LD, LDI, ANB lub ORB na instrukcję NOP zmieni znacząco obwód; z dużym prawdopodobieństwem pojawią się błędy programu
- Po wyzerowaniu programu wszystkie jego instrukcje zostaną zastąpione instrukcjami NOP

2.20 Instrukcja END

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Format	Urządzenia	Kroki programu
END (END)	wymusza zakończenie bieżącego przebiegu programu		brak	1

Najważniejsze informacje do zapamiętania:

- Umieszczenie instrukcji END w programie wymusza zakończenie bieżącego przebiegu i aktualizację procesów na wyjściach i wejściach.
- Umieszczenie instrukcji END wewnątrz programu wspomaga wyszukiwanie błędów w programie, ponieważ sekcja po instrukcji END jest wyłączona i izolowana z obszaru, który jest wykonywany. Należy pamiętać o usunięciu instrukcji END z bloków, które już zostały sprawdzone.
- Kiedy wykonywana jest instrukcja END timer watchdoga sterownika jest automatycznie odświeżany.



Przebieg programu:

- Przebieg programu jest jednokrotnym wykonaniem załadowanego programu od jego początku do końca. Dotyczy to aktualizacji wszystkich wejść, wyjść i timera watchdoga. Czas trwania takiego procesu zwany jest czasem przebiegu (czasem skanowania). Zależy on od długości i złożoności programu. Natychmiast po ukończeniu jednego przebiegu programu uruchamiany jest następny. Cały proces odbywa się cyklicznie. Aktualizacja wejść odbywa się na początku każdego przebiegu, wyjścia natomiast aktualizowane są na końcu przebiegu.

Notatki

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

3	Programowanie STL	3-1
3.1	Czym jest STL, SFC i IEC1131 część 3?	3-1
3.2	Jak działa STL?	3-2
3.1.1	Każdy krok jest programem	3-2
3.3	Jak zacząć i zakończyć program STL?	3-3
3.3.1	Zagnieżdżone programy STL	3-3
3.3.2	Aktywowanie nowych stanów	3-3
3.3.3	Zatrzymywanie programu STL	3-4
3.4	Przechodzenie pomiędzy krokami STL	3-5
3.4.1	Użycie instrukcji SET do sterowania cewką znacznika stanu STL	3-5
3.4.2	Użycie instrukcji OUT do sterowania cewką znacznika stanu STL	3-6
3.5	Reguły i techniki pisania programów STL	3-7
3.5.1	Podstawowe uwagi dotyczące zachowania się programów STL	3-7
3.5.2	Sterowanie kroku pojedynczym sygnałem	3-9
3.6	Ograniczenia niektórych instrukcji używanych z STL	3-10
3.7	Rozgałęzienia wyboru w programie STL	3-11
3.8	Rozgałęzienia równoległe w programie STL	3-12
3.9	Ogólne zasady realizacji poprawnych rozgałęzień STL	3-14
3.10	Ogólne zalecenia dotyczące stosowania oprogramowania FX-PCS/AT-EE	3-15
3.11	Przykłady programowania	3-16
3.11.1	Prosty przepływ STL	3-16
3.11.2	Przykładowy program rozgałęzienia wyboru	3-18
3.12	Zaawansowane użycie STL	3-20

1 Programowanie STL

FX_{0(S)}FX_{0N}

FX

FX_(2C)FX_{2N}

Rozdział ten różni się od reszty tego podręcznika, ponieważ został opracowany w formie wykładu. Programowanie STL/SFC, pomimo tego, że stosowane od wielu lat, wciąż jest źle przedstawiane i źle rozumiane. My, w Mitsubishi, chcielibyśmy wykorzystać okazję i spróbować naprawić to niedopatrzenie, ponieważ uważamy, że programowanie STL/SFC staje się tak ważne jak programowanie drabinkowe.

3.1 Czym jest STL, SFC i IEC1131 część 3?

Poniższe wyjaśnienie jest bardzo skrócone, lecz zostało opracowane w celu szybkiego zestawienia różnic i podobieństw pomiędzy STL, SFC i IEC1131 cz.3.

W ostatnich latach programowanie SFC (Sequential Function Chart) (włączając w to podobne style, jak Grafcet i Funktionplan) zdobyło wielką popularność w Europie i zapoczątkowało tworzenie normy IEC1131 cz.3.

Standard IEC1131 został opracowany w celu otrzymania przenośnego języka programowania. Chodziło o to, żeby program napisany w standardach IEC1131 dla programowalnego sterownika jednego producenta mógł być łatwo przeniesiony (przetworzony) na PLC innego producenta.

Programowanie STL wykorzystuje jedną z podstawowych instrukcji programowania, o takiej samej nazwie - STL dostępnych dla wszystkich sterowników rodziny FX. Skrót STL oznacza SStep Ladder programming. Programowanie STL jest łatwe do zrozumienia i stanowi jedną z najpotężniejszych możliwych technik programowania sterowników. Klucz do STL leży w jego możliwościach, pozwalających programiście na tworzenie programu, który "przepływa" i działa prawie tak samo jak SFC. Nie jest to dziełem przypadku, ponieważ ta technika programowania została specjalnie rozwinięta w celu uzyskania systemu łatwego do monitorowania i programowania.

Jedną z kluczowych różnic systemu programowania STL, propagowanego przez Mitsubishi, jest to, że może on wprowadzony do sterownika programowalnego w 3 formatach. Są nimi:

- I) Lista instrukcji - system wprowadzania słowny/mnemotechniczny
- II) Schemat drabinkowy - graficzna metoda konstrukcji programu, używająca symboliki logiki przekaźnikowej
- III) Schemat SFC - system wprowadzania oparty na sieci działań STL (podobny do SFC)

Przykłady tych metod programowania można zobaczyć na stronie 2-1.



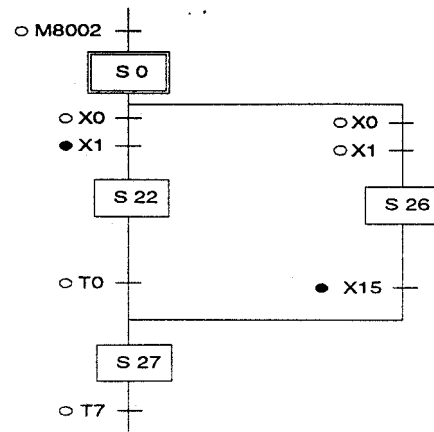
Uwaga ogólna:

IEC1131-3:03.1993 Sterowniki programowalne; część 3; język programowania.

Powyższy standard jest technicznie identyczny z Euro-Normą EN61131-3:07.1993

3.2 Jak działa STL?

Jak wspomniano poprzednio, STL jest systemem, który pozwala użytkownikowi napisać program, funkcjonujący w sposób prawie identyczny jak sieć działań, można to zobaczyć na diagramie obok. Mocną stroną STL jest to, że pozwala organizować duże programy w mniejsze części, którymi łatwiej zarządzać. Każdą z tych części można traktować jako stan lub krok. Żeby odróżnić stany, każdy z nich opatrzony jest unikatowym numerem identyfikacyjnym. Numery te wzięte są ze znaczników stanu (zobacz stronę 4-6).



3.2.1 Każdy krok jest programem

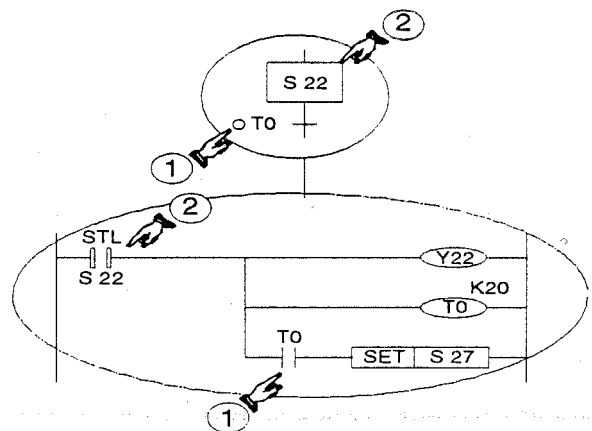
Każdy stan jest całkowicie izolowany od innych stanów wewnątrz całego programu. Każdy stan jest oddzielnym programem, a użytkownik składa je po to by mogły wykonać swoje zadanie. Oznacza to, że stany mogą być ponownie używane wiele razy i w różnych układach. Skraca to czas programowania oraz zmniejsza liczbę popełnionych błędów.

Spojrzenie do wewnątrz STL

Na pierwszy rzut oka program STL wygląda tak, jakby był podstawową siecią działań. Ale aby dowiedzieć się, co się naprawdę dzieje, stan STL musi być "umieszczony pod mikroskopem". Podczas dokładnego badania pojedynczego stanu można podglądać podprogram.

Z wyjątkiem instrukcji STL można natychmiast spoznać, że podprogram STL wygląda dokładnie jak zwykłe programowanie.

- ① Instrukcja STL jest pokazana jako pogrubiony symbol styku "normalnie rozwartego". Programowanie za instrukcją STL jest tylko wtedy aktywne, gdy skojarzona cewka jest aktywna (odpowiedni stan jest aktywny).
- ② Warunek przejścia jest również zapisany przy użyciu standardowego programowania.



Omówienie to pokazuje, że STL jest rzeczywiście metodą sekwencjonowania serii zdarzeń, albo, jak wspomniano wcześniej, łączenia razem wielu małych programów.

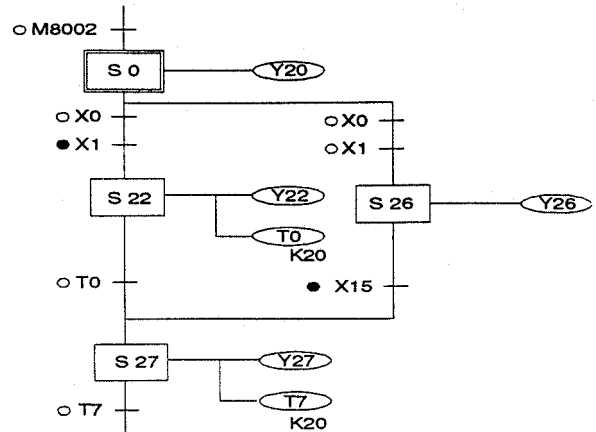
Kombinacja programu SFC ze schematem drabinkowym

Czasami programy STL będą pokazane na wydruku jako kombinacja sieci działań jak i wewnętrznego podprogramu (na przykładzie obok).



Identyfikacja stanów styków
Używana jest następująca konwencja:
○ styk normalnie rozwarty
● styk normalnie zwarty

Zamiennie używa się identyfikatorów 'a' i 'b' dla oznaczenia stanów, odpowiednio, normalnie rozwartych i normalnie zwartych lub kreski nad oznaczeniem styku normalnie zwartego np. X000

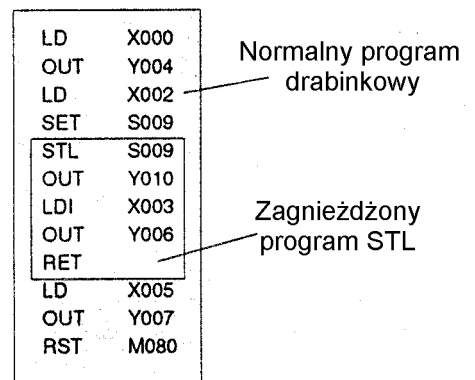


3.3 Jak zacząć i zakończyć program STL?

Przed rozpoczęciem jakiegokolwiek złożonego programowania należy wyjaśnić podstawy, jak zacząć i co ważniejsze, jak zakończyć program STL.

3.3.1 Zagnieżdżone programy STL

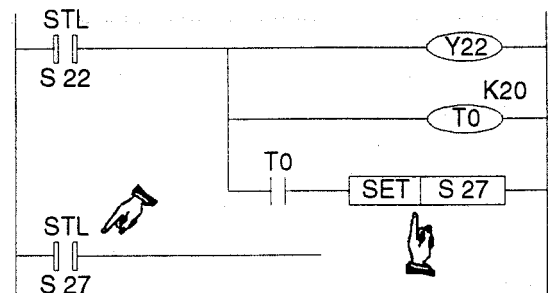
Program STL nie musi zupełnie zastępować programu w standardowej logice drabinkowej. W gruncie rzeczy mogłoby być to bardzo trudne. Zamiast tego małe albo nawet duże sekcje programu STL mogą być wprowadzone w dowolnym punkcie programu. Po ukończeniu zadania STL sterowanie musi wrócić do przetwarzania instrukcji standardowego programu, aż do wystąpienia następnego bloku programu STL. Dlatego tak ważna jest identyfikacja początku i końca programu STL.



3.3.2 Aktywowanie nowych stanów

Jak jest używany i sterowany program po wybraniu kroku STL? Aby krok STL był aktywny, skojarzona cewka stanu musi być w stanie ON. Stąd więc, aby uruchomić sekwencję STL, wystarczy ustawić odpowiedni stan w ON.

Istnieje wiele różnych metod ustawiania stanu, np. cewki stanów inicjujących mogą być impulsowane, ustawiane instrukcją SET lub instrukcją OUT. Jednakże, w języku programowania STL, proponowanego przez Mitsubishi, cewka STL, ustawiana instrukcją SET ma inne znaczenie niż ta, która jest zawarta w instrukcji OUT.

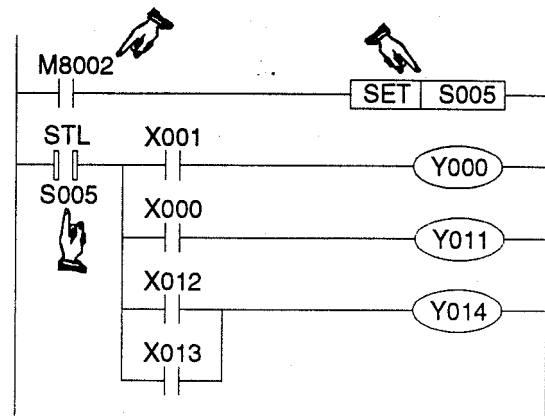


Uwaga:
Dla normalnych operacji STL zaleca się wybieranie stanów za pomocą instrukcji SET. Do uaktywnienia kroku STL stan jego cewki jest ustawiany instrukcją SET w stan ON.

Kroki inicjujące

Dla programu STL, który ma być aktywowany po włączeniu zasilania sterownika, można użyć układu wyzwalającego pokazanego na rysunku obok, tzn. wykorzystującego znacznik M8002 do sterowania ustawienia stanu początkowego.

Krok STL uruchamiany w ten sposób często jest nazywany krokiem inicjującym. Podobnie, krok aktywowany przed jakąkolwiek sekwencją STL jest także nazywany krokiem inicjującym.



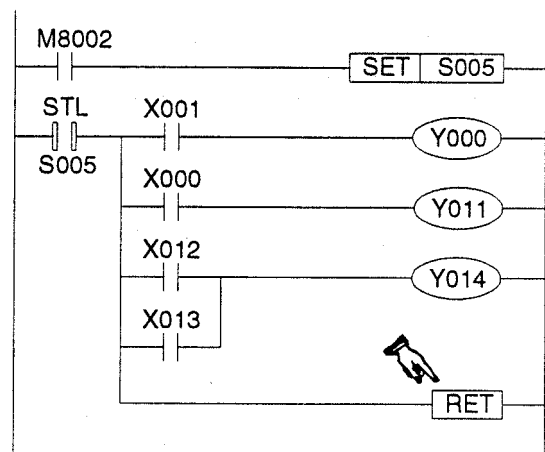
3.3.3 Zatrzymywanie programu STL

Po uruchomieniu programu STL, CPU sterownika będzie przetwarzać wszystkie kolejne instrukcje, jako części tego programu. Oznacza to, że gdy rozpocznie się drugi przebieg programu, normalne instrukcje na jego początku są traktowane, jakby były częścią programu STL. Jest to błędne i spowoduje błąd programu oraz wyłączenie operacji sterownika.

Program STL musi zwrócić sterowanie do drabinki po zakończeniu operacji STL. To znaczy, że ostatni krok w programie STL musi być w jakiś sposób zidentyfikowany.

Powrót do drabinki

Osiąga się to poprzez umieszczenie instrukcji RET albo RETURN jako ostatniej instrukcji w ostatnim kroku bloku programu STL. Instrukcja ta zwraca sterowanie do schematu drabinkowego.



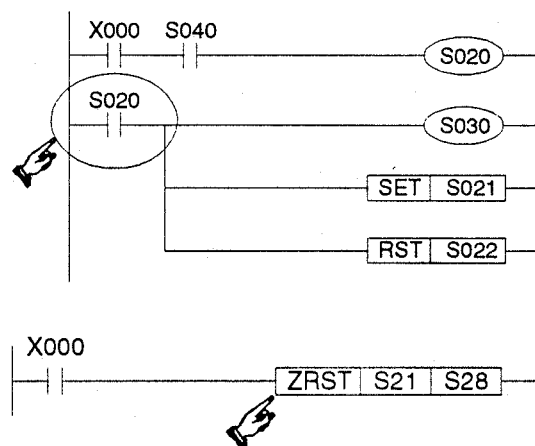
Uwaga: Instrukcja RET może być użyta do oddzielenia programów STL od standardowych sekcji schematu drabinkowego. Instrukcja RET jest używana do zaznaczenia końca kompletnego programu STL.

3.4 Przechodzenie pomiędzy krokami STL

Aby uaktywnić krok STL, użytkownik musi najpierw wysteroować cewkę znacznika stanu. Wskazano już, że sposobem uruchomienia programu STL jest ustawienie cewki, tzn. wysteroowanie znacznika stanu inicjującego. Wspomniano też, że używanie instrukcji OUT do wysteroowania cewki znacznika stanu ma inne znaczenie od instrukcji SET. Różnice te będą teraz wyjaśnione:

3.4.1 Użycie instrukcji SET do sterowania cewką znacznika stanu STL

- Instrukcja SET jest użyta do sterowania cewką stanu STL aby uaktywnić krok. Gdy bieżący krok STL uaktywnia następujący po nim drugi krok, cewka źródłowego znacznika stanu jest zerowana. Stąd, pomimo tego, że instrukcja SET używana jest tylko do aktywowania bieżącego stanu, automatycznie następuje zerowanie stanu poprzedniego. Jednak, jeżeli stan STL jest sterowany przez serię instrukcji schematu drabinkowego, tzn. nie jest poprzedzony stanem STL, wtedy obowiązują standardowe reguły programowania. W przykładzie obok S20 nie jest zerowane, nawet wtedy, gdy S30 lub S21 zostały wysteroowane. Co więcej, jeżeli S20 jest w stanie OFF, S30 również przestanie działać. Tak się dzieje, ponieważ S20 nie zostało użyte jako znacznik stanu STL. Pierwsza instrukcja wywołująca znacznik S20 jest standardową instrukcją LD i nie jest instrukcją STL.



Uwaga: Jeżeli użytkownik chce wymusić zerowanie kroku STL, może to zrobić, używając RST albo ZRST (FNC 40).

- Instrukcja SET jest używana do wysteroowania występującego po niej kroku STL, który zwykle ma większy numer znacznika stanu niż krok bieżący.
- Instrukcja SET jest używana do wysteroowania stanów STL, które pojawiają się wewnątrz programu STL, tzn. instrukcja SET nie jest używana do aktywowania stanu, który pojawia się w drugiej, oddzielnej sieci działań.

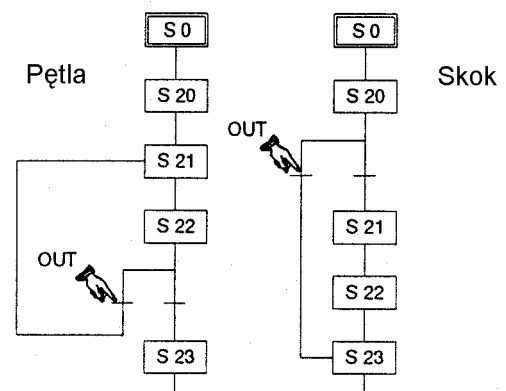
3.4.2 Użycie instrukcji OUT do sterowania cewką znacznika stanu STL

Użycie tej instrukcji ma te same cechy operacyjne jak użycie instrukcji SET. Jednak istnieje jedna, bardzo ważna funkcja, w której SET nie jest używana. Dotyczy to "odległych skoków".

Instrukcja OUT służy do tworzenia pętli i skoków.

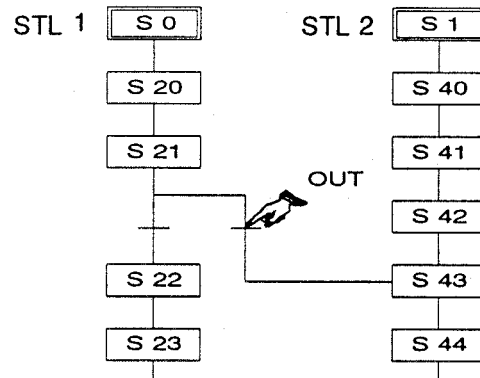
Jeżeli użytkownik chce wykonać skok w tył programu, tzn. wrócić do stanu, który już został przetworzony, musi użyć instrukcji OUT z odpowiednim numerem stanu STL.

Użytkownik może też wykonać duży "skok" w przód, przeskakując całą sekcję STL.



Instrukcja OUT służy do wykonywania odległych skoków.

Instrukcja OUT powinna być również użyta, gdy wymaga się, aby stan z jednego programu przełączył stan w innym, oddzielnym programie STL.



Uwaga: Pomimo tego, że jest możliwa realizacja pętli i skoków z wykorzystaniem instrukcji SET, do prezentacji programu STL w formie SFC wymaga się użycia instrukcji OUT.

3.5 Reguły i techniki pisania programów STL

Można łatwo zauważyć, że jest wiele zalet używania stylu programowania STL, ale istnieje kilka punktów, z których użytkownik musi zdawać sobie sprawę, pisząc podprogramy STL. Są one wyszczególnione w tej sekcji.

3.5.1 Podstawowe uwagi dotyczące zachowania się programów STL

- Kiedy stan STL zostanie uaktywniony, jego program jest przetwarzany do momentu, gdy wyzwolony zostanie następny krok. Program może zawierać wszystkie elementy i cechy standardowego programowania drabinkowego, jak instrukcje LD, AND, OR, OUT, RST itd. oraz instrukcje specjalne.
- Pisząc podprogram stanu STL, pierwsza pionowa linia bazowa po instrukcji STL może być traktowana w podobny sposób, jak lewa linia bazowa w programie drabinkowym.

Każdy krok STL tworzy swoją własną linię bazową, tzn. że użytkownik nie może użyć instrukcji MPS bezpośrednio za instrukcją STL. (zobacz 1), czyli przed instrukcją MPS musi wystąpić co najmniej jeden pojedynczy styk.

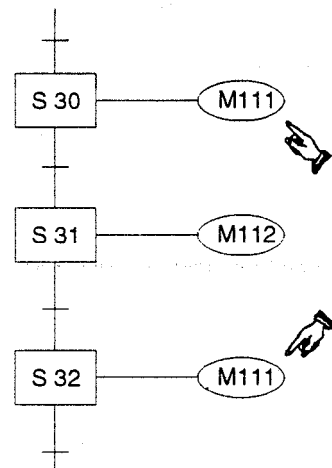
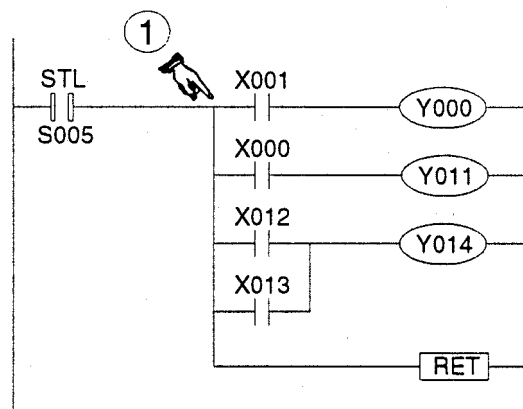


Uwaga: bezpośrednio po instrukcji STL zezwala się na użycie cewki wyjściowej a nawet instrukcji specjalnej.

- W normalnym programowaniu użycie podwójnej cewki nie jest akceptowalną techniką. Jednak w oddzielnych blokach programu STL zezwala się na powtórne użycie cewki. Tak się dzieje, ponieważ użytkownik może wykorzystać unikatową cechę STL izolującą wszystkie kroki STL z wyjątkiem aktywnych kroków STL. Oznacza to, że w praktyce nie wystąpi konflikt podwójnej cewki. Przykład obok pokazuje znacznik M111 użyty dwa razy w jednym programie STL.



Ostrzeżenie: Ta sama cewka nie powinna być programowana w krokach, które będą aktywne w tym samym czasie, ponieważ prowadzi to do takich samych rezultatów, jak w przypadku podwójnej cewki.



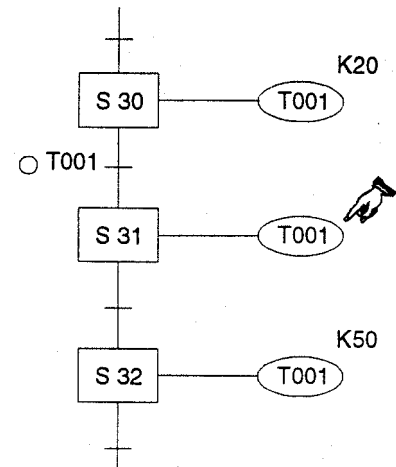


Kiedy krok STL przekazuje sterowanie do następnego kroku, występuje przedział czasu (jeden przebieg - skan), podczas którego oba kroki są aktywne. Może wystąpić problem podwójnych cewek, szczególnie dotyczy to timerów.

W takim przypadku należy zwrócić uwagę na to, by operacja timera została ukończona podczas aktywności kroku STL.

Jeśli timer jest nieprzerwanie używany w następujących po sobie krokach, wtedy możliwe jest, że cewka timera nigdy nie jest deaktywowana i styki timera nie będą wyzerowane, co prowadzi do wadliwego działania timera.

Przykład obok pokazuje złe użycie timera T001. Kiedy sterowanie przechodzi z S30 do S31, T001 nie jest wyzerowany, ponieważ jego cewka jest w stanie ON również w nowym kroku.



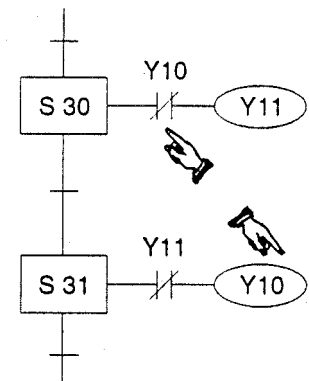
Uwaga: Aby zapewnić poprawne działanie tych samych timerów, nie powinny być one używane w następujących po sobie krokach STL.

Pamiętając o tej prostej zasadzie uzyska się pewność, że każdy timer będzie poprawnie wyzerowany przed jego następną operacją.



Jak już wspomniano, podczas transferu pomiędzy krokami, te dwa kolejne kroki będą jednocześnie aktywne dla jednego przebiegu programu (skanu).

Oznacza to, że jeżeli użytkownik ma dwa wyjścia zawarte w następujących po sobie krokach, które nie mogą być jednocześnie aktywne, muszą one być blokowane. Dobrym tego przykładem będzie sterowanie sygnałów wybierających kierunek obrotów silników. W przykładzie obok pokazane jest jak wyjścia Y11 i Y10 wzajemnie się blokują.



3.5.2 Sterowanie kroku pojedynczym sygnałem

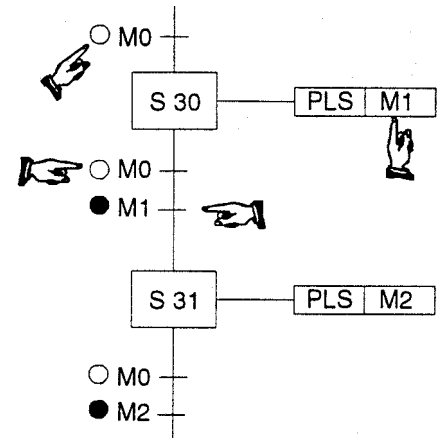
Przejście pomiędzy aktywnymi krokami STL może być sterowane pojedynczym sygnałem. Można tego dokonać przy pomocy dwóch metod.

Metoda 1 - użycie znaczników blokady

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

W przykładzie tym należy zaprogramować oddzielne urządzenia blokujące, a sygnał sterujący musi tylko wytworzyć impuls ON. Zabezpiecza to przed ciągłym przełączaniem stanów STL. Program wykorzystujący tę metodę pokazuje poniższy przykład.

- Po podaniu impulsu na M0, S30 jest aktywne
- Działanie znacznika M1 zabezpiecza przed ciągłym działaniem sekwencji, ponieważ mimo, że M0 jest w stanie ON, transfer wymaga, żeby M0 było w stanie ON, a M1 w stanie OFF.
- Po jednym przebiegu znacznik impulsowany M0 i znacznik blokady M1 są zerowane.
- Po następnym impulsie na M0, krok STL przeniesie w podobny sposób sterowanie programu z S31 do następnego kroku. Tym razem, używając M2 jako znacznika blokady, ponieważ użycie podwójnej cewki w następujących po sobie krokach nie jest dozwolone.
- Powodem użycia znaczników blokady jest okres potwierdzenia, w czasie którego oba stany użyte w przeniesieniu sterowania programu są w stanie ON dla jednego przebiegu programu. Bez tej blokady byłoby możliwe natychmiastowe przeskoczenie wszystkich stanów STL za jednym zamachem!



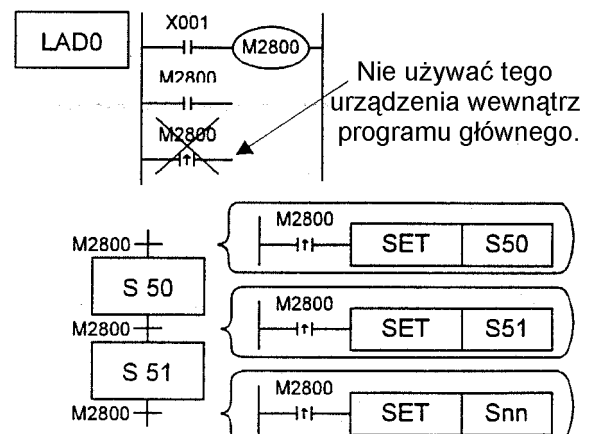
Metoda 2 - specjalne znaczniki pojedynczego impulsu

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Przy użyciu styków impulsowych (LDP, LDF, ANP itp) i określonego zakresu znaczników (M2800-M3071) sterownik programowalny FX2N osiąga taki sam rezultat jak metoda 1. Ta specjalna cecha tych urządzeń zabezpiecza przed swobodnym przebiegiem stanów, ponieważ tylko pierwsze pojawienie się instrukcji LDP będzie aktywne.

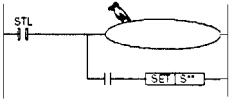
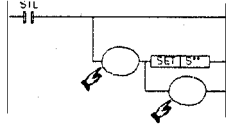
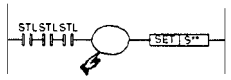
Przykład poniżej pokazuje potrzebne instrukcje

- Załóżmy, że S50 jest już aktywne
- Kiedy X01 uaktywnia M2800, to z kolei uaktywnia instrukcję LDP M2800 w stanie S50 i przebieg przechodzi do S51.
- Instrukcja LDP M2800, zawarta w części przejścia S51 nie wykonuje się, ponieważ jest to drugie pojawienie się M2800 w styku impulsowym.
- Kiedy X01 ponownie aktywuje M2800, instrukcja LDP w S51 jest pierwszym pojawieniem się, ponieważ S50 jest teraz nieaktywne. Sterowanie przechodzi do następnego kroku w ten sam sposób.



3.6 Ograniczenia niektórych instrukcji używanych z STL

Chociaż STL może pracować z większością instrukcji podstawowych i specjalnych, jest kilka wyjątków. Podstawową regułą jest to, że STL i format MC-MCR nie mogą być łączone. Inne ograniczenia instrukcji podano w tabeli poniżej.

Stan operacyjny		Instrukcje podstawowe		
		LD, LDI, AND, ANI, OR, ORI, NOP, OUT, SET, RST, PLS, PLF	ANB, ORB, MPS, MRD, MPP	MC, MCR
stany inicjujące i ogólne		✓	✓	x
stany rozgałęziające i łączące	przetwarzanie wyjścia 	✓	✓	x
	przeniesienie 	✓	x	x

Ograniczenia dotyczące użycia instrukcji specjalnych.

- Większość instrukcji specjalnych może być użyta w programach STL. Trzeba tylko zwrócić uwagę, w jaki sposób STL izoluje każdy nieaktywny krok. Zaleca się by operacje instrukcji specjalnych były zakończone zanim aktywny krok STL przeniesie sterowanie do następnego kroku.

Inne ograniczenia:

- Struktury FOR-NEXT nie mogą zawierać bloków programu STL.
- Podprogramy i przerwania nie mogą zawierać bloków programu STL.
- Bloki programów STL nie mogą być umieszczane po instrukcji FEND
- Instrukcje FOR-NEXT są dozwolone w programie STL z zagnieżdżeniem do 4 poziomów.



Więcej szczegółów w tabelach na stronach 7-12, 7-13



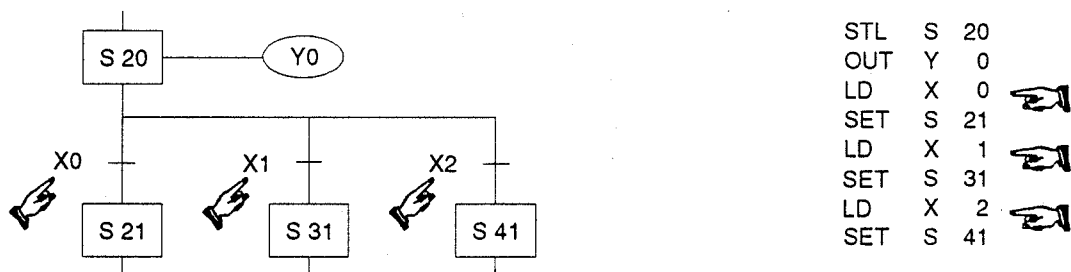
Używanie operacji skoków z STL.

Chociaż możliwe jest używanie operacji skoku (instrukcja CJ) wewnątrz programu STL, wprowadza to dodatkowe i często niepotrzebne komplikacje przebiegu programu. Dla zapewnienia łatwej obsługi i szybkiego wykrywania błędów zaleca się unikanie umieszczania instrukcji skoku w programach STL.

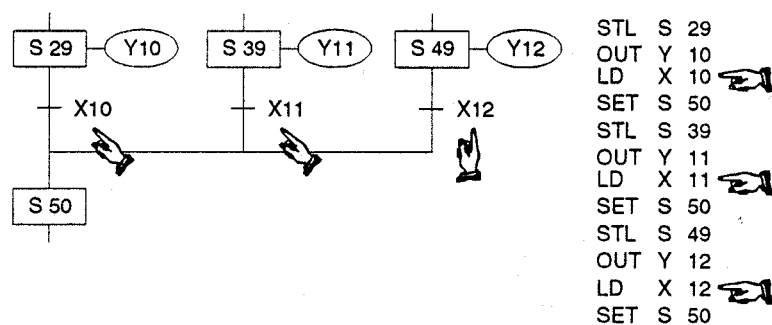
3.7 Rozgałęzienia wyboru w programie STL

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Do tej pory można było sądzić, że STL jest tylko prostym językiem sieci działań. Jedną z unikatowych cech STL jest możliwość tworzenia programów, które mogą pracować w kilku trybach operacyjnych. Na przykład niektóre maszyny wymagają wyboru trybu ręcznego lub automatycznego, inne zaś mogą potrzebować zdolności do wyboru operacji lub procesów wytwarzania "A", "B", "C" lub "D". STL umożliwia to poprzez wielokrotne rozgałęzienia programu rozpoczynające się w jednym stanie STL. Każde rozgałęzienie jest wtedy programowane jako indywidualny tryb pracy i ponieważ nie powinno być innych aktywnych trybów wybór gałęzi programu musi się wzajemnie wykluczać. Ten typ konstrukcji programu zwany jest "selektywnym programowaniem gałęziowym". Poniżej pokazano przykład programu (jest to tylko podprogram STL dla stanu S20) należy zauważyć, że każda gałąź jest ustawiana przez inny styk.



Konstrukcja rozgałęzienia wyboru stwarzająca możliwość zmiany przepływu programu jest bardzo przydatna, ale jej użyteczność jeszcze bardziej wzrasta, dzięki możliwości ponownego łączenia zbioru indywidualnych gałęzi.



Ten typ konstrukcji programu STL nazywa się "łączeniem pierwszego stanu" ponieważ pierwszy stan (na przykładzie S29, S39 lub S49) który zakończy swoją operację spowoduje, że stan połączenia S50 zaktywizuje się. Należy zauważyć, jak każdy z końcowych stanów STL różnych gałęzi programowych wywołuje ten sam łączący stan STL.



Ograniczenie liczby gałęzi

Ogólne uwagi na temat programowania rozgałęzień wyboru STL znajdują się na stronie 3-14

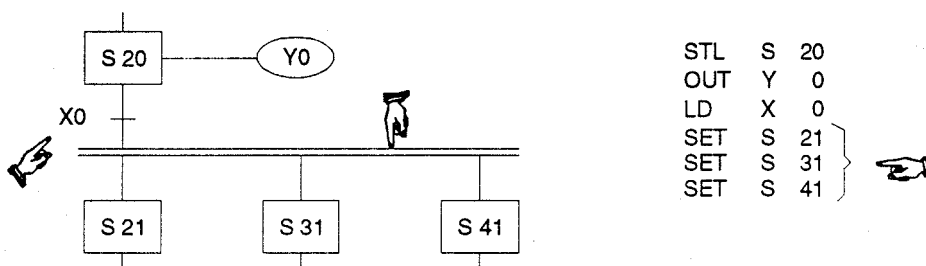
Uwagi na temat oprogramowania FX-PCS/AT-EE

- Ogólne uwagi na temat środków ostrożności przy wykorzystywaniu oprogramowania FX-PCS/AT-EE znajdują się na stronie 3-15

3.8 Rozgałęzienia równoległe w programie STL

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

W poprzedniej technice rozgałęzień można było zobaczyć, jak pojedynczy przepływ mógł być wybrany z grupy. Poniżej przedstawiono metodę opisującą, jak grupa indywidualnych przepływów może być jednocześnie (równoległe) aktywizowana. Aplikacje mogą obsługiwać automaty sprzedające, które muszą wykonywać kilka zadań jednocześnie, np. gotować wodę, dodawać różne składniki smakowe (kawa, herbata, mleko, cukier, itp). W przykładzie poniżej, kiedy stan S20 jest aktywny i X0 następnie zostanie przełączony w stan ON, stan S21, S31 i S41 przejdą w stan ON jako następny stan. Stąd też trzy oddzielne, indywidualne rozgałęzienia zostaną uruchomione z pojedynczego punktu. Ta technika programowania często jest zwana "techniką gałęzi równoległych". Żeby było to wyraźnie widać, rozgałęzienie równoległe jest zaznaczone dwiema poziomymi równoległymi liniami.



Kiedy zostanie uaktywniona grupa rozgałęzionych przepływów, użytkownik ma do wyboru dwie metody:

a) Ten przepływ, który zakończy się najszybciej, uaktywni funkcję łączącą ("łączenie pierwszego stanu", opisaną w poprzedniej sekcji)

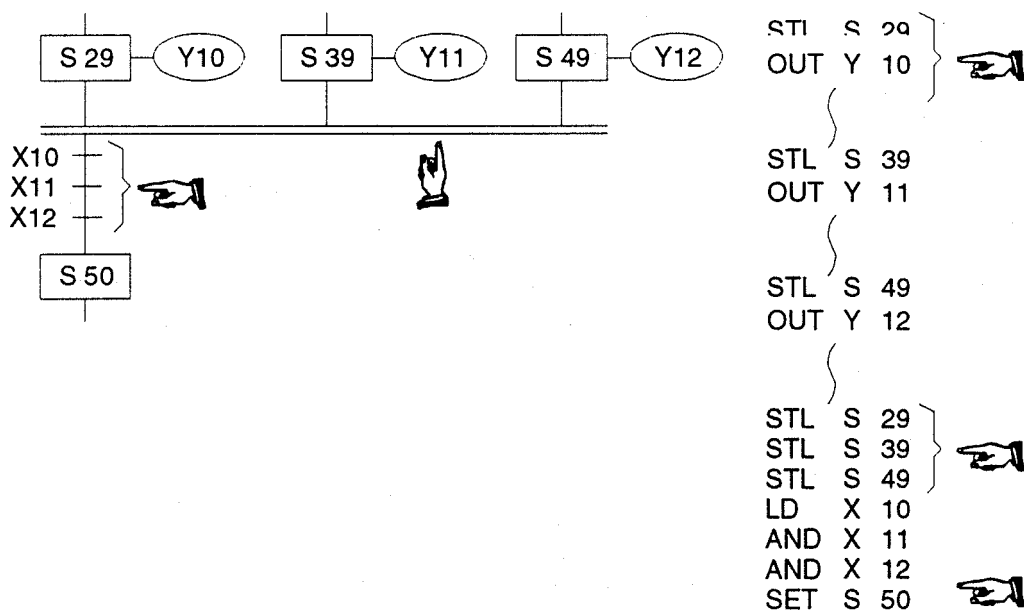
lub

b) Przepływ STL nie będzie kontynuowany, dopóki wszystkie gałęzie przepływu nie ukończą swojego działania.

To nosi nazwę "łączenia wielokrotnych stanów" - łączenie równoległe.

Poniżej wyjaśnienie "łączenia wielokrotnych stanów".

W poniższym przykładzie stany S29, S39 i S49 muszą być aktywne. Przeglądając listę instrukcji można zauważyć, że każdy ze stanów (ostatnich stanów przed łączeniem) ma swoje własne instrukcje operacyjno/przetwarzające, ale zostały również dołączone dodatkowe instrukcje STL (w podobny sposób jak podstawowe instrukcje AND). Zanim zostanie uaktywniony stan S50, muszą być aktywne warunki przejścia tranzycji. W tym przykładzie są to X10, X11, X12. Gdy są już spełnione wszystkie warunki dotyczące stanów i wejść, stan łączący (tu S50) może być ustawiony w stan ON. Tak jak w ogólnym przypadku, wszystkie stany używane w procedurze ustawiania są zerowane automatycznie.



Ponieważ więcej niż jeden stan jest równocześnie łączony z następnymi stanami (czasami jest to opisywane jako łączenie równoległe), zaznacza się to poziomymi, równoległymi liniami.



Ograniczenia, dotyczące liczby gałęzi
zobacz strona 3-14

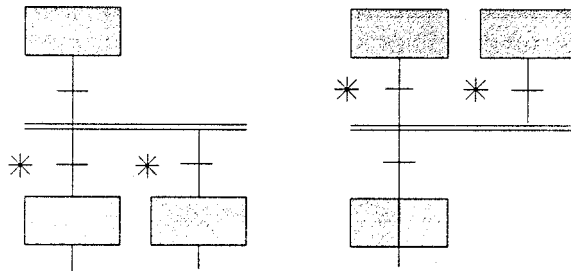
Uwagi dotyczące używania oprogramowania FX-PCS/AT-EE

- Ogólne uwagi na temat środków ostrożności przy wykorzystywaniu oprogramowania FX-PCS/AT-EE znajdują się na stronie 3-15

3.9 Ogólne zasady realizacji poprawnych rozgałęzień STL

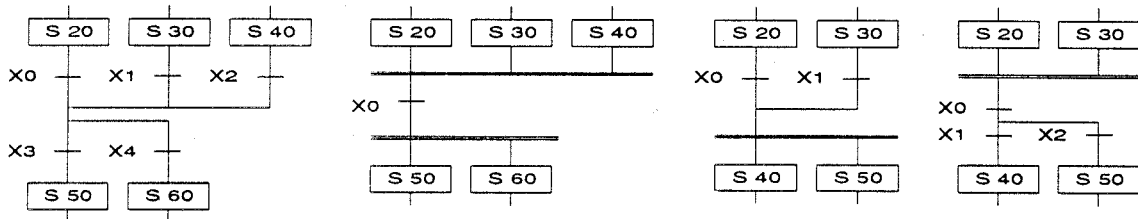
Dla każdego punktu rozgałęzienia można zaprogramować do 8 następných rozgałęzień. Nie ma żadnych ograniczeń, dotyczących liczby stanów zawartych w pojedynczym przepływie STL. Stąd istnieje możliwość rozgałęzienia pojedynczego stanu inicjującego na 8 przepływów, które z kolei można rozgałęzić na następnych 8 przepływach itd. Jeżeli program sterownika jest czytany/pisany przy użyciu listy instrukcji lub schematu drabinkowego, powyższe reguły są dozwolone. Jednakże użytkownicy pakietu FX-PCS/AT-EE, którzy korzystają z cech programowania STL, są ograniczeni następnymi restrykcjami, uniemożliwiającymi automatyczną konwersję programu STL (więcej szczegółów na stronie 3-15).

Używając gałęzi, różne typy rozgałęzienia/łączenia nie mogą być mieszane w tym samym punkcie rozgałęzienia. Elementy oznaczone '*' są warunkami transferu, które nie są dopuszczalne.



Zaleca się następujące gałęzi.

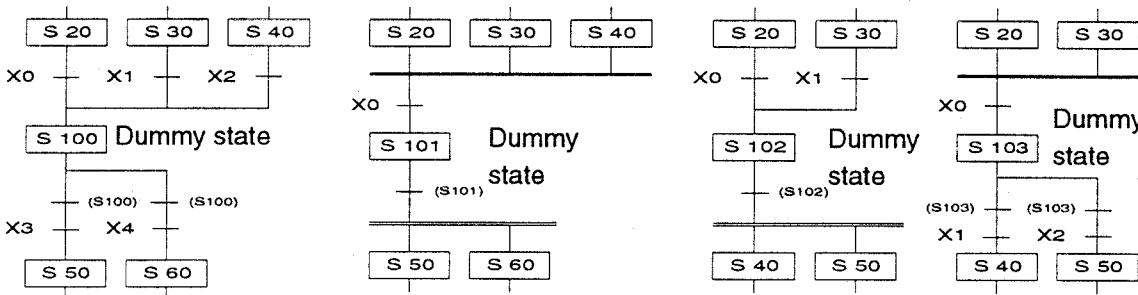
konfiguracje/modyfikacje



Przepisz jako



Przepisz jako



W formacie instrukcji...



W formacie instrukcji...



```

STL S 20
LD X 0
SET S 100
STL S 30
LD X 1
SET S 100
STL S 40
LD X 2
SET S 100
STL S 100
LD S 100
AND X 3
SET S 50
LD S 100
AND X 4
SET S 60
    
```

```

STL S 20
STL S 30
STL S 40
LD X 0
SET S 101
STL S 101
LD S 101
SET S 50
SET S 60
    
```

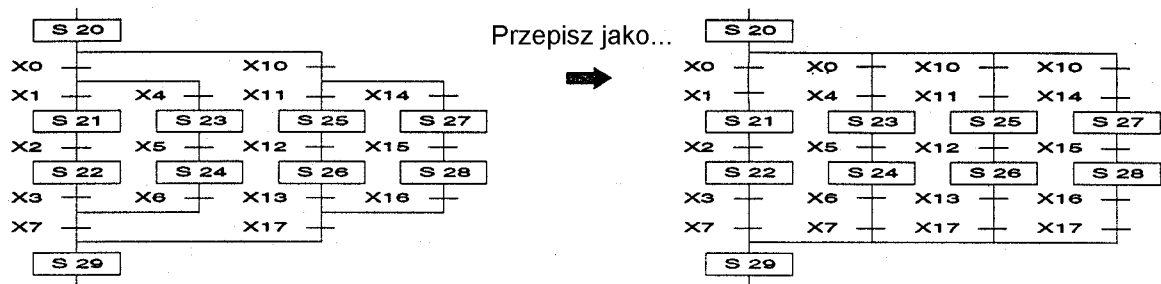
```

STL S 20
LD X 0
SET S 102
STL S 30
LD X 1
SET S 102
STL S 102
LD S 102
SET S 40
SET S 50
    
```

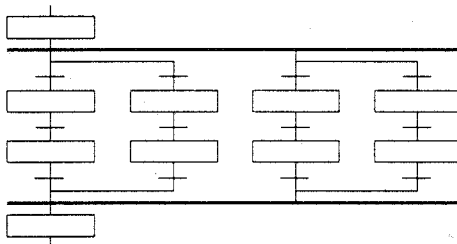
```

STL S 20
STL S 30
LD X 0
SET S 103
STL S 103
LD S 103
AND X 1
SET S 40
LD S 103
AND X 2
SET S 50
    
```

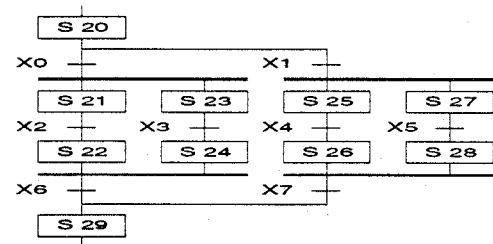
Dalsze polecane zmiany programu



Naruszenie programu!



Przepisz jako...



STL	\$	20	STL	\$	22
LD	X	0	STL	\$	24
SET	\$	21	LD	X	6
SET	\$	23	SET	\$	20
LD	X	1	STL	\$	26
SET	\$	25	STL	\$	28
SET	\$	27	LD	X	7
			SET	\$	29

3.10 Ogólne zalecenia dotyczące stosowania oprogramowania FX-PCS/AT-EE

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

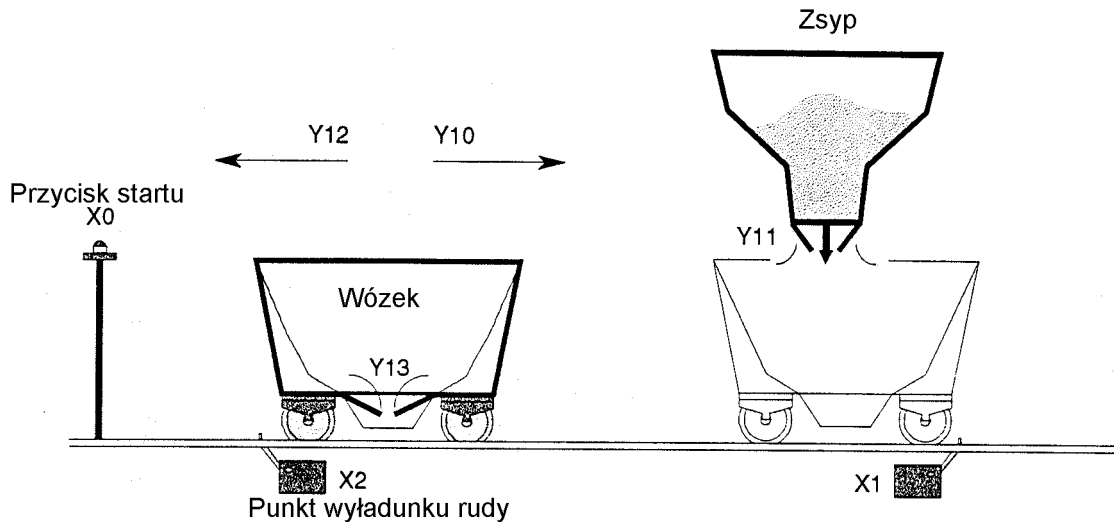
Oprogramowanie to umożliwia programowanie sieci działań SFC. Jedną z cech tej możliwości jest to, że może ono czytać i konwertować istniejące programy STL z powrotem do przepływów SFC nawet jeżeli one nie były nigdy pierwotnie programowane przy użyciu pakietu FX-PCS/AT-EE. Przy automatycznym generowaniu przepływu SFC powinny być wzięte pod uwagę następujące reguły i punkty:

- 1) Przepływ STL powinien być inicjowany jednym ze znaczników stanu z zakresu S0-S9.
- 2) Wybór lub dołączenie gałęzi powinno być pisane sekwencyjnie zaczynając od lewej do prawej. Było to pokazane na stronie 3-11, tzn. dla rozgałęzienia wyboru S21 było wyspecyfikowane przed S31, które było specyfikowane przed S41. Stany dołączone były zaprogramowane w podobny sposób, S29 poprzedzało S39, które poprzedzało S49.
- 3) Całkowita liczba rozgałęzień, które mogą być zaprogramowane w trybie programowania STL, jest ograniczona do 16 obwodów na jeden przepływ STL. Każdy punkt gałęzi jest ograniczony do 8 rozgałęzień. To oznacza, że 2 punkty po 8 rozgałęzień spełniają wcześniejsze ograniczenie. Te ograniczenia wprowadzono po to, by użytkownik zawsze mógł widzieć diagram przepływów STL na komputerze z oprogramowaniem FX-PCS/AT-EE i kiedy zaistnieje potrzeba, mógł go przejrzeć wydrukować.

3.11 Przykłady programowania

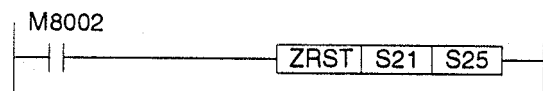
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

3.11.1 Prosty przepływ STL



Ten prosty przykład jest częścią programu półautomatycznego ładowania/rozładowania transporterów rudy. Przykładowy program zawiera wbudowaną procedurę inicjującą, wykonywaną tylko podczas włączania zasilania PC. Osiągnięto to wykorzystując specjalny znacznik M8002.

Aktywuje on instrukcję Zone ReSeT (instrukcja specjalna ZRST, FNC 40), która zeruje wszystkie operacyjne stany STL. Przykład obok pokazuje użycie instrukcji M8002/ZRST.



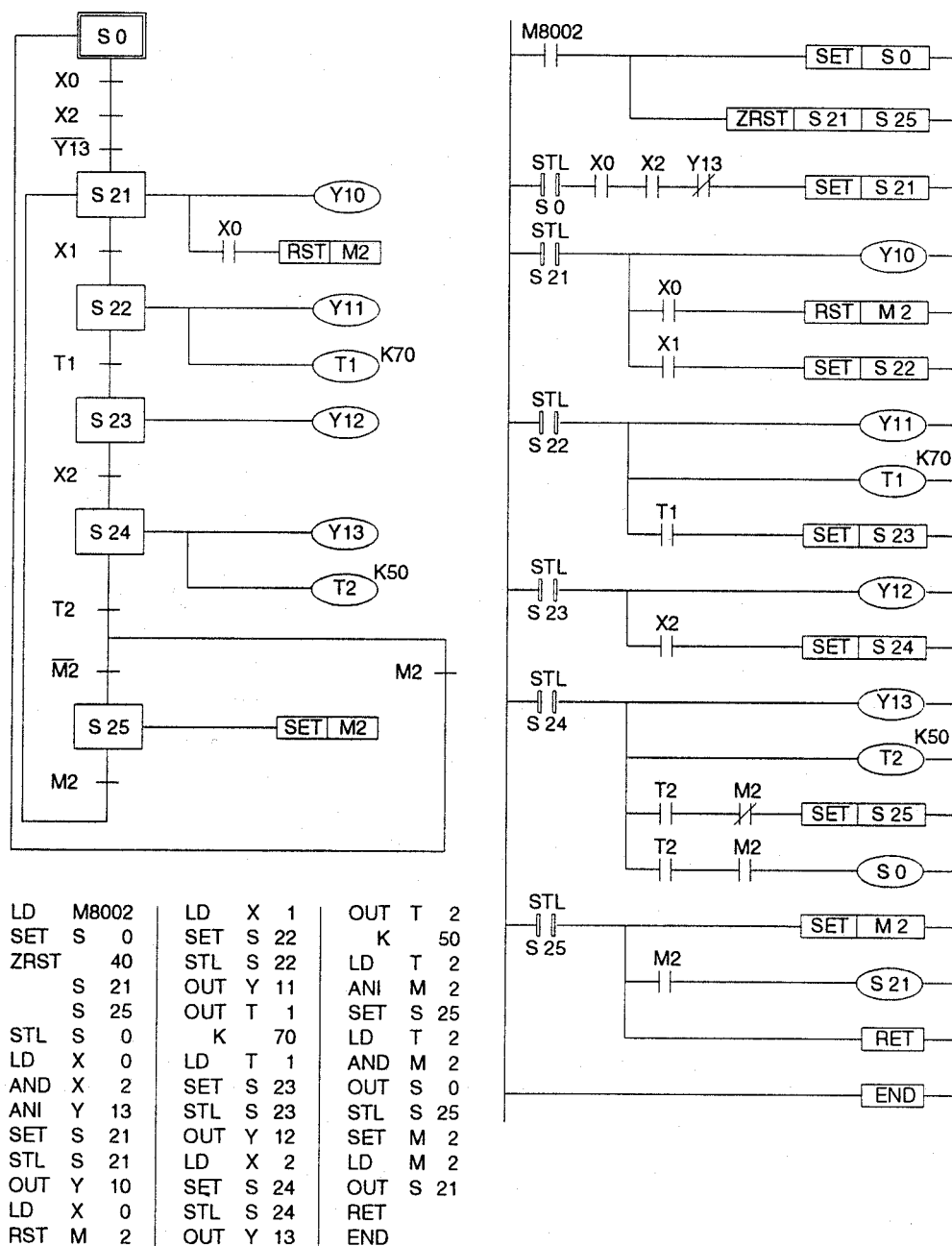
Przycisk X0 pracuje jako przycisk startowy i jako przycisk wybierający tryb pracy. Stan S0 jest inicjowany instrukcją ZRST. System czeka na podanie sygnałów na wejścia X0 i X2 i nieaktywność wyjścia Y13. Oznacza to, że transporter rudy jest ustawiony w punkcie rozładunku, tzn. nad czujnikiem pozycji X2. Transporter nie jest w tym momencie rozładowywany, tzn. nie jest aktywny sygnał otwarcia drzwi transportera (Y13) oraz nie otrzymano sygnału z przycisku X0. Kiedy wszystkie punkty zostały wykonane, program przechodzi do stanu S21.

W tym stanie transporter porusza się w prawo (Y10) i zatrzymuje się, gdy zadziała wyłącznik krańcowy (X1) pod lejem ładującym. Jeśli był naciśnięty przycisk X0, transporter będzie realizował powtarzający się cykl (M2 jest wyzerowany), w którym transporter po rozładowaniu jest natychmiast zawracany pod lej ładujący. Ten tryb musi być wybrany po każdym powrocie do stacji ładowania.

W punkcie ładowania program przechodzi do stanu S22. Stan ten otwiera klapę leja i wypełnia transporter rudą. Po określonym czasie S23 jest aktywowany i transporter wraca Y12 do punktu rozładunku (X2).

Tam otwiera swoje dolne klapy (Y13). Po określonym czasie, podczas którego transporter jest opróżniany, program sprawdza, czy w ostatnim cyklu został wybrany tryb powtórzenia. tzn M2 jest zerowany. Jeżeli M2 został wyzerowany (w stanie S21) program przeskakuje do kroku S21 i transporter wraca do natychmiastowego napełniania. Jeżeli M2 nie jest wyzerowany, tzn. jest aktywny, program wraca do stanu S0, gdzie transporter będzie czekał na naciśnięcie przycisku startowego.

Jest to kompletny, prosty program, ale pokazuje sposób wykorzystywania serii zadań, wykonywanych za pomocą programowania STL.

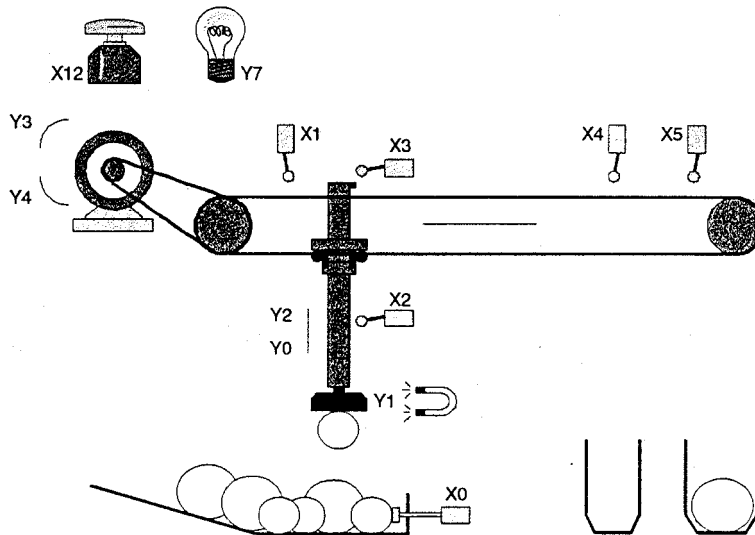


Identyfikacja styków normalnie zwartych

W przykładzie wykorzystano konwencję oznaczania styków normalnie zwartych kreską nad oznaczeniem styku. Na stronie 3-3 pokazano dalsze odmiany i inne metody użyte do tej identyfikacji.

3.11.2 Przykładowy program rozgałęzienia wyboru

Następujący przykład opisuje automat sortujący. Automat wybiera kulki łożyskowe z pojemnika źródłowego i sortuje je w dwóch oddzielnych pojemnikach, uwzględniając ich rozmiar.



Sekwencja zdarzeń (od włączenia zasilania) jest następująca:

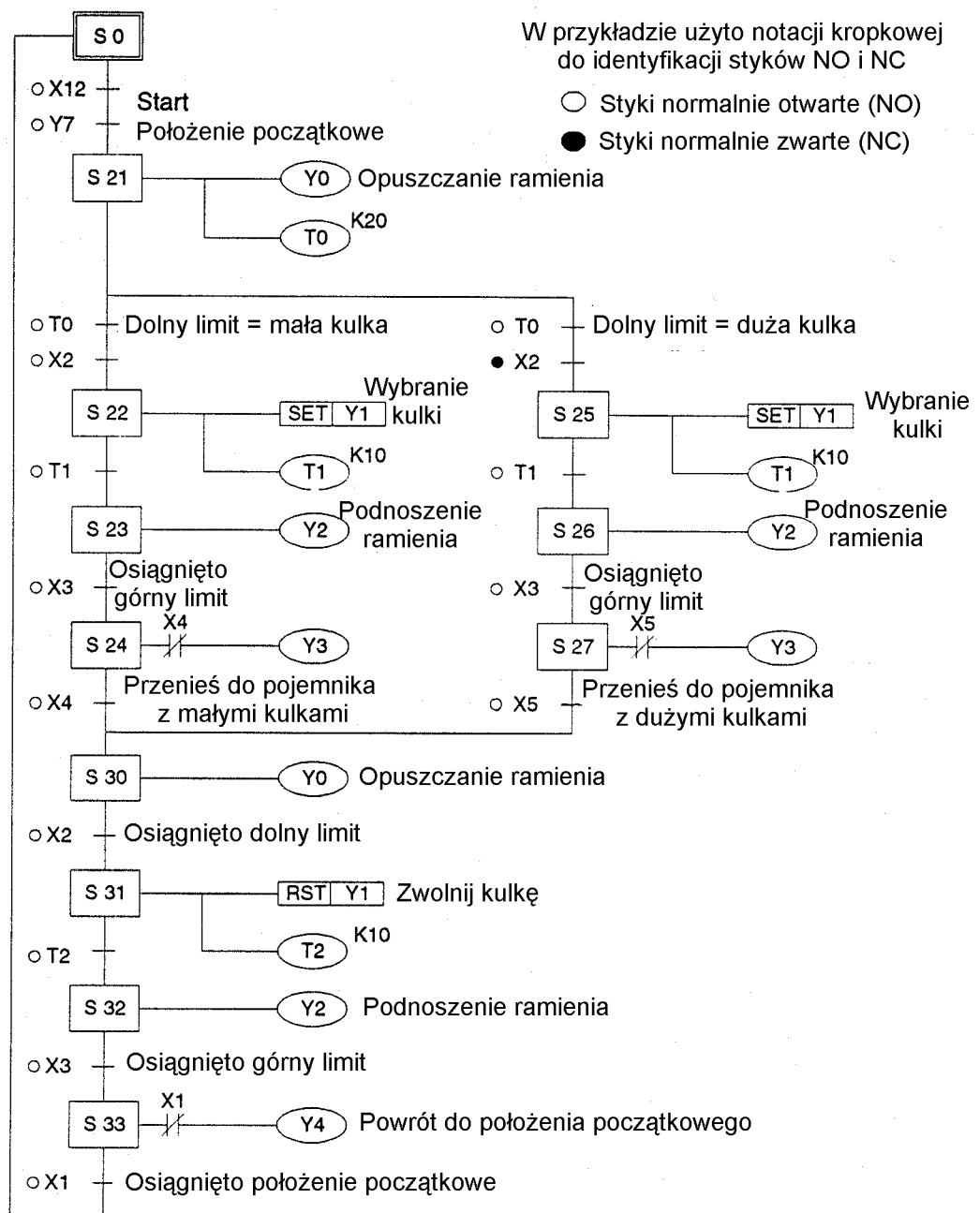
- 1) ramię wybierające jest przesuwane do swojego punktu zerowego, po naciśnięciu przycisku startu (X12). Po osiągnięciu punktu zerowego zapala się lampka (Y7).
- 2) ramię wybierające jest opuszczane (Y0) do momentu uchwycenia kulki łożyska (Y1); jeśli nastąpi przełączenie dolnego wyłącznika krańcowego (X2), wybierane jest mała kulka, konsekwentnie brak przełączenia dolnego wyłącznika krańcowego oznacza wybranie dużej kulki. Dodatkowo, przełącznik zbliżeniowy X0 wewnątrz pojemnika źródłowego informuje o obecności kulek.
- 3) w zależności, która kulka została wybrana, ramię wycofuje się (wyjście Y2 jest aktywne do momentu odebrania sygnału z wejścia X3) i przesuwane jest w prawo (Y3), gdzie zatrzyma się na wyłączniku krańcowym (X4 lub X5), wskazującym położenie odpowiedniego pojemnika.
- 4) program kontynuuje działanie opuszczając ramię (Y0) do momentu otrzymania sygnału z dolnego wyłącznika krańcowego (X2)
- 5) wybrana kulka zostaje uwolniona (Y1 jest wyzerowane)
- 6) ramię (Y2) jest ponownie wycofane
- 7) ramię powraca (Y4) do punktu zerowego (X1)



Ważne punkty

- Rozgałęzienie wyboru jest użyte do selekcji programu dostarczającego małe lub duże kulki łożyskowe. Po osiągnięciu miejsca przeznaczenia (tzn. został wykonany krok S24 lub S27) dwa niezależne przepływy programu są ponownie złączane w kroku S30.
- Przykładowy program pracuje w pojedynczym cyklu, tzn. musi być naciśnięty przycisk startu, by zapoczątkować wybór kulek.

Pełny diagram przepływu STL.



3.12 Zaawansowane użycie STL

Programowanie STL może być rozszerzone przy użyciu instrukcji specjalnej inicjującej stan - IST FNC 60. Kiedy używana jest instrukcja IST, przyporządkowanie znaczników następuje automatycznie. Instrukcja IST dostarcza programiście metod do tworzenia programów o wielu trybach. Dostępne tryby to:

- a) automatyczny
 - pojedynczy krok
 - pojedynczy cykl
 - praca ciągła
- b) ręczny
 - sterowanie przez operatora
 - powrót do stanu początkowego

Więcej szczegółów na temat tej instrukcji na stronie 5-68

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

4	Charakterystyka urządzeń	4-1
4.1	Wejścia	4-1
4.2	Wyjścia	4-2
4.3	Znaczniki M	4-3
4.3.1	Ogólne informacje dotyczące przekaźników pomocniczych	4-3
4.3.2	Przekaźniki pomocnicze z podtrzymaniem/zatraskowe	4-4
4.3.3	Znaczniki systemowe	4-5
4.3.4	Znaczniki pojedynczego impulsu	4-5
4.4	Znaczniki stanu S	4-6
4.4.1	Ogólny stan znaczników	4-6
4.4.2	Znaczniki stanu z podtrzymaniem/zatraskowe	4-7
4.4.3	Znaczniki kroku STL	4-8
4.4.4	Znaczniki wskaźnikowo-diagnostyczne	4-9
4.5	Etykiety skoków	4-10
4.6	Etykiety przerw	4-11
4.6.1	Przerwania od wejść	4-12
4.6.2	Przerwania od timerów	4-12
4.6.3	Blokowanie poszczególnych przerw	4-13
4.6.4	Przerwania od liczników	4-13
4.7	Stała K	4-14
4.8	Stała H	4-14
4.9	Timery	4-15
4.9.1	Zasada działania timera	4-16
4.9.2	Timery o przełączanej rozdzielczości	4-16
4.9.3	Timery z pamięcią	4-17
4.9.4	Timery wykorzystywane w przerwanach i podprogramach (CALL)	4-18
4.9.5	Dokładność timera	4-18
4.10	Liczniki	4-19
4.10.1	Liczniki 16-bitowe, dodające, zwykłe i zatraskowe	4-20
4.10.2	Liczniki 32-bitowe, dwukierunkowe, zwykłe i zatraskowe	4-21
4.11	Liczniki szybkie	4-22
4.11.1	Podstawowe operacje na liczniku szybkim	4-23
4.11.2	Dostępność liczników szybkich w programowalnych sterownikach FX ₀ , FX _{0S} i FX _{0N}	4-24
4.11.3	Dostępność liczników szybkich w programowalnych sterownikach FX i FX _{2C}	4-25
4.11.4	Dostępność liczników szybkich w programowalnych sterownikach FX _{2N}	4-28
4.11.5	Liczniki jednofazowe - uruchamiane i zerowane przez program użytkownika (C235-C240)	4-29
4.11.6	Liczniki jednofazowe - z uruchamianiem i zerowaniem sprzętowym (C246-C250)	4-30
4.11.7	Liczniki dwufazowe, dwukierunkowe (C246-C250)	4-31
4.11.8	Liczniki A/B fazowe (C252-C255)	4-32
4.12	Rejestry danych	4-33
4.12.1	Rejestry ogólnego przeznaczenia	4-34
4.12.2	Rejestry z podtrzymaniem/zatraskowe	4-35
4.12.3	Rejestry diagnostyczne	4-35
4.12.4	Rejestry plikowe	4-36
4.12.5	Rejestry ustawiane zewnętrznie	4-37
4.13	Rejestry indeksowe	4-38
4.13.1	Modyfikowanie stałej	4-39
4.13.2	Niewłaściwe użycie modyfikatorów	4-39
4.13.3	Użycie wielokrotne rejestrów indeksowych	4-39
4.14	Bity, Słowa, zapis BCD i zapis szesnastkowy	4-40
4.14.1	Urządzenia bitowe, pojedyncze i grupowe	4-40
4.14.2	Urządzenia słowowe	4-42
4.14.3	Interpretacja danych słowowych	4-42
4.14.4	Uzupełnienie do 2	4-45
4.15	Zapis zmiennoprzecinkowy i szesnastkowy	4-46
4.15.1	Zapis naukowy	4-47
4.15.2	Format zmiennoprzecinkowy	4-48
4.15.3	Zestawienie notacji naukowej i liczb zmiennoprzecinkowych	4-49

4. Charakterystyka urządzeń

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

4.1 Wejścia

Mnemonik urządzenia: X (Input)

Zastosowanie: reprezentuje fizyczne wejścia sterownika programowalnego

Oznaczenie: I/P

Inp

(X) wejście

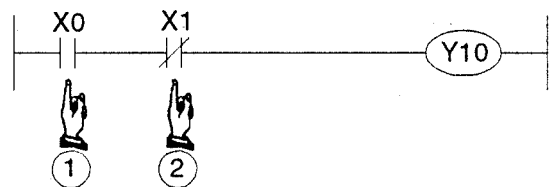
Styk wejściowy

Dostępne formy: tylko styki normalnie rozwarte (①) i normalnie zwarte (②)
zobacz przykład zastosowania

Numerowanie urządzeń: ósemkowe (tzn. X0 do X7, X10 do X17, itd.)

Dalsze zastosowania: brak

Przykład użycia urządzenia



Dostępne urządzenia:

- zobacz punkt informacyjny na stronie 4-2, Wyjścia. lub odpowiednie tabele wybranych sterowników w rozdziale 8.

Szczegóły dotyczące konfiguracji

- zobacz rozdział 9

4.2 Wyjścia

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik urządzenia: Y (Output)

Zastosowanie: reprezentuje fizyczne wyjścia sterownika programowalnego

Oznaczenie: O/P

Otp

Out (Y)

Wyjście (Y)

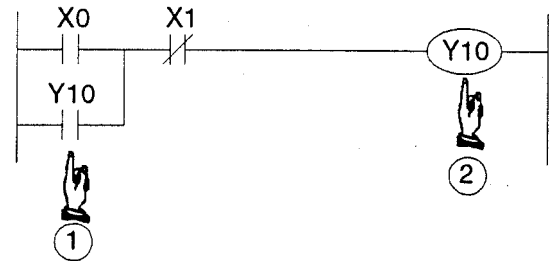
Wyjścia (cewka, przekaźnik, znacznik, styk)

Dostępne formy: tylko styki normalnie rozwarte (①) i normalnie zwarte oraz cewki wyjściowe (②) zobacz przykład zastosowania

Numerowanie urządzeń: ósemkowe (tzn. Y0 do Y7, Y10 do Y17, itd.)

Dalsze zastosowania: brak

Przykład użycia urządzenia



Dostępne urządzenia:

Sterownik (PLC)	Maksymalna liczba wejść	Maksymalna liczba wyjść	Liczba dostępnych wejść/wyjść
FX _{0(S)}	ustawiane przez wybraną jednostkę bazową		30
FX _{0N}	84	(40)	128
	(60)	64	
FX	128	128	256
FX _(2C)	256 (adresowane programowo)	256 (adresowane programowo)	256 (adresowane programowo i sprzętowo)
FX _{2N}			

- Powyższa tabela określa wartości maksymalne. Konkretnie wartości zależą od indywidualnych ustawień. Szczegóły w rozdziale 9.
- Więcej informacji o dostępności urządzenia dla poszczególnych sterowników można znaleźć w rozdziale 8.

4.3 Znaczniki M

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic urządzenia: M (Marker)

Zastosowanie: wewnętrzny znacznik stanu, przekaźnik pomocniczy sterownika programowalnego

Oznaczenie: znacznik (cewka, przekaźnik, styk), flaga

Dostępne formy: styki normalnie rozwarte (①) i normalnie zwarte oraz cewki wyjściowe (②) zobacz przykład zastosowania

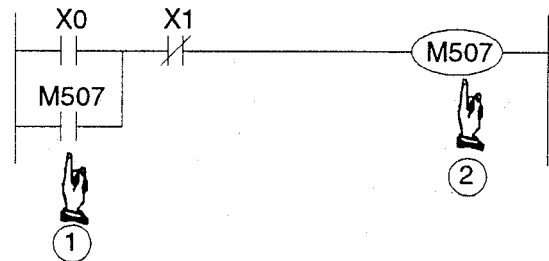
Numerowanie urządzeń: dziesiętne (tzn. M0 do M9, M10 do M19)

Dalsze zastosowania:

znaczniki z podtrzymaniem/zatrzaskowe - strona 4-4

znaczniki diagnostyczne - strona 4-5

Przykład użycia urządzenia:



4.3.1 Ogólne informacje dotyczące przekaźników pomocniczych

W sterowniku programowalnym dostępna jest pewna liczba przekaźników pomocniczych. Cewki tych przekaźników są sterowane stykami urządzeń w PLC, definiowanymi przez program, w taki sam sposób, jak przekaźniki wyjściowe.

Wszystkie przekaźniki pomocnicze mają nieograniczoną liczbę programowych styków (normalnie rozwartych i normalnie zwartych), które mogą być używane w programie sterownika. Należy zauważyć, że styki te nie mogą bezpośrednio sterować zewnętrznym obciążeniem. Do tego celu służą przekaźniki wyjściowe.



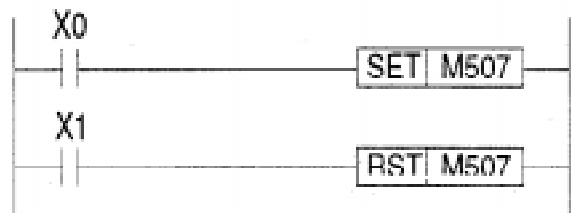
Dostępne urządzenia:

- Przedstawione są one w punkcie informacyjnym na stronie 4-4: "Przekaźniki pomocnicze z podtrzymaniem/zatrzaskowe", oraz w tabelach dla wybranych PLC w rozdziale 8.

4.3.2 Przełączniki pomocnicze z podtrzymaniem/zatrzaskowe

Jeśli nastąpi zanik zasilania podczas pracy PLC, wszystkie przełączniki wyjściowe i przełączniki ogólnego stosowania przejdą w stan OFF (stan wyłączenia). Wszystkie te przełączniki pozostaną w stanie OFF, z wyjątkiem tych, które są reaktywowane po wznowieniu działania sterownika. Dlatego ich stan jest taki sam, jaki był tuż przed zanikiem zasilania. Urządzenia te określa się jako zatrzaskowe lub z podtrzymaniem. Stan jest podtrzymywany przez baterię lub pamięć EEPROM wbudowaną w PLC.

Układ pokazany na stronie 4-3 jest przykładem układu samozachowawczego (z pamięcią). Przełącznik M507 jest aktywowany, kiedy X0 jest w stanie ON. Jeżeli X0 jest w stanie OFF po aktywacji M507, stan ON przełącznika M507 jest zachowany, tzn. normalnie rozarty styk M507 jest zwarty. Gdy wejście X0 jest w stanie OFF, M507 jest zerowany (przełączony w stan OFF), tzn. jego styk normalnie rozarty zostaje otwarty. Instrukcje SET i RST mogą być użyte do podtrzymania stanu przełącznika chwilowo aktywowanego.



Dostępne urządzenia:

Sterownik (PLC)	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
Przełączniki pomocnicze	496 (M0-495)	384 (M0-383)	500 (M0-499)	500 (M0-499)	500 (M0-499)
Przełączniki pomocnicze z podtrzymaniem/zatrzaskowe	16 (M496-511)	128 (M384-511)	524 (M500-1023)	1036 (M500-1535)	2572 (M500-3071)
Razem dostępnych	512	512	1024	1536	3072

- Więcej informacji dotyczących dostępności urządzeń w konkretnych typach sterownika w rozdziale 8. Informacje na temat dostępności urządzeń przy użyciu sterownika FX, wyposażonego w FX2-40AW/AP, na stronie 9-6.

Obciążenia zewnętrzne:



- Przełączniki są wyposażone w nieograniczoną liczbę styków normalnie rozartych i normalnie zwartych. Są one dostępne w całym programie PLC. Styki te nie mogą bezpośrednio sterować obciążeniem zewnętrznym. Wszystkie obciążenia zewnętrzne powinny być sterowane przy użyciu bezpośrednich wyjść (Y).

4.3.3 Znaczniki systemowe

Sterownik programowalny ma pewną liczbę specjalnych przekaźników pomocniczych, których stan definiowany jest przez oprogramowanie systemowe sterownika. Przekaźniki te, zwane znacznikami systemowymi, mają numery od M8000. Pełnią one specyficzne funkcje i są podzielone na dwie następujące grupy:

a) znaczniki tylko do odczytu:

- Cewki tych przekaźników są automatycznie sterowane przez program systemowy PLC. Tylko styki tych przekaźników mogą być używane w programie użytkownika.

Przykłady: M8000: monitor stanu RUN (jest w stanie ON podczas działania programu)
M8002: impuls początkowy (generowany natychmiast po starcie PLC)
M8012: impulsy zegarowe 100 ms

b) znaczniki do zapisu i odczytu, sterujące cewkami specjalnych przekaźników pomocniczych

- PLC wykonuje określone z góry specjalne operacje podczas sterowania tych przekaźników w programie użytkownika

Przykłady: M8033: wszystkie stany wyjściowe są zachowane kiedy operacje PLC są zatrzymane
M8034: wszystkie wyjścia są zablokowane
M8039: PLC pracuje w stałym przebiegu programu

Dostępne urządzenia:

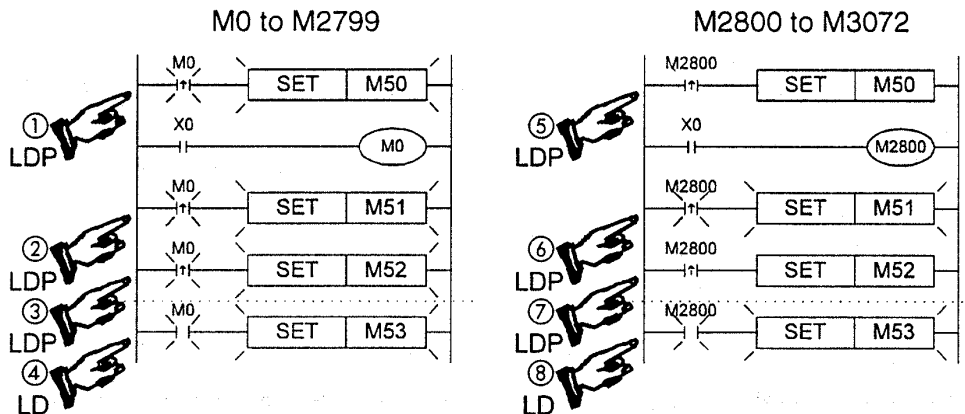


- Nie wszystkie PLC używają ten sam zakresu, liczbę i znaczenie operacyjne systemowych przekaźników pomocniczych. Przed użyciem urządzenia należy sprawdzić jego dostępność i funkcję. Specyfikacja zakresów i znaczeń w PLC są omówione w rozdziale 6.

4.3.4 Znaczniki pojedynczego impulsu

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Znaczniki M, z zakresu M2800-M3072, mają specjalne znaczenie, kiedy są używane ze stykami impulsowymi LDP, LDF itd. Takie użycie tych znaczników powoduje, że tylko pierwsza instrukcja styku impulsowego, znajdująca się za cewką tego przekaźnika, jest aktywowana.



Ustawienie X0 w stan ON powoduje ustawienie M0 w stan ON.

- Styki ①, ②, ③ są stykami impulsowymi i aktywują się na czas jednego przebiegu programu.
- Styk ④ jest normalnym stykiem LD i aktywuje się podczas gdy M0 jest w stanie ON.

Ustawienie X0 w stan ON powoduje ustawienie M2800 w stan ON.

- Styk ⑥ jest stykiem impulsowym i aktywuje się na czas jednego przebiegu programu.
- Styki ⑤ i ⑦ są stykami impulsowymi tego samego urządzenia M co styk ⑥. Styk ⑥ już działał, tak więc styki ⑤ i ⑦ nie działają.
- Styk ⑧ jest normalnym stykiem LD i aktywuje się podczas gdy M0 jest w stanie ON.

4.4 Znaczniki stanu S

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik urządzenia: S (Status Marker)

Zastosowanie: wewnętrzny znacznik stanu sterownika programowalnego

Oznaczenie: stan (cewka/przełącznik/styk/flaga)

S (cewka/przełącznik/styk/flaga)

krok STL (cewka/przełącznik/styk/flaga)

znacznik wskaźnikowy

Dostępne formy: styki normalnie rozwarne (①) i normalnie zwarte oraz cewki wyjściowe (②)
zobacz przykład zastosowania

Numerowanie urządzeń: dziesiętne (tzn. S0 do S9, S10 do S19)

Dalsze zastosowania:

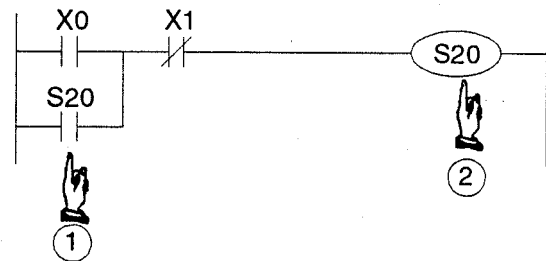
znaczniki stanu - zobacz strona 4-6

znaczniki stanu z podtrzymaniem/zatrzaskowe - strona 4-7

znaczniki kroku STL - zobacz strona 4-8

znaczniki wskaźnikowe - zobacz strona 4-9

Przykład użycia urządzenia:



4.4.1 Ogólny stan znaczników

W sterowniku programowalnym używana jest pewna liczba przełączników pomocniczych. Cewki tych przełączników są sterowane stykami urządzeń w PLC w taki sam sposób, jak przełączniki wyjściowe są sterowane w programie.

Wszystkie przełączniki pomocnicze mają dowolną liczbę programowych styków (normalnie rozwartych i normalnie zwartych), które mogą być używane w programie użytkowym. Należy zauważyć, że te styki nie mogą bezpośrednio sterować zewnętrznym obciążeniem. Do tego celu służą przełączniki wyjściowe.



Dostępne urządzenia:

- Przedstawione są one w punkcie informacyjnym na stronie 4-7: "Przełączniki stanu z podtrzymaniem/zatrzaskowe", oraz w tabelach dla wybranych PLC w rozdziale 8.

4.4.2 Znaczniki stanu z podtrzymaniem/zatrzaskowe

Jeśli nastąpi zanik zasilania podczas pracy PLC, wszystkie przekaźniki wyjściowe i przekaźniki ogólnego stosowania przejdą w stan OFF (stan wyłączenia). Wszystkie te przekaźniki pozostaną w stanie OFF, z wyjątkiem tych, które są reaktywowane po wznowieniu działania sterownika. Urządzenia te określa się jako zatrzaskowe lub z podtrzymaniem. Stan jest podtrzymywany przez baterię lub pamięć EEPROM wbudowaną w PLC. Przekaźniki stanu z podtrzymaniem/zatrzaskowe działają w podobny sposób jak pomocnicze przekaźniki z podtrzymaniem/zatrzaskowe, opisane na stronie 4-4



Dostępne urządzenia:

Sterownik (PLC)	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
Przekaźniki pomocnicze	64 (S0-63)	brak		500 (S0-499)	
Przekaźniki pomocnicze z podtrzymaniem/zatrzaskowe	brak	128 (S0-127)		500 (S500-999)	
Razem dostępnych	64	128		1000	

- Dalsze informacje dotyczące dostępności urządzeń dla poszczególnych PLC znajdują się w rozdziale 8.

Obciążenia zewnętrzne:

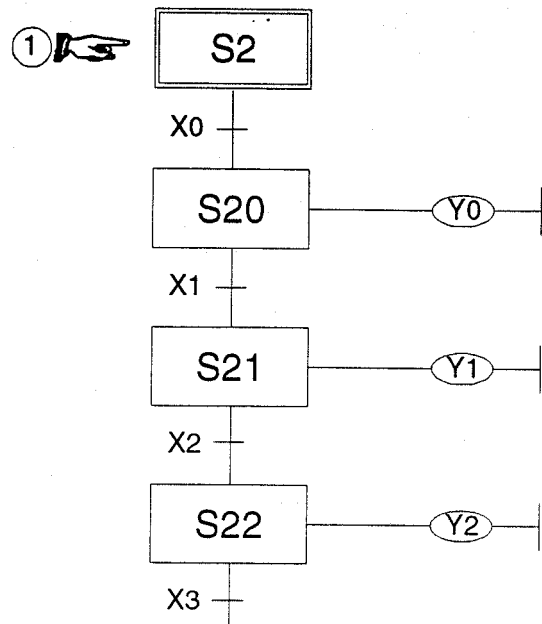


- Przekaźniki stanu są wyposażone w nieograniczoną liczbę styków normalnie rozwartych i normalnie zwartych. Są one łatwo dostępne w całym programie PLC. Styki te nie mogą bezpośrednio sterować obciążeniem zewnętrznym. Wszystkie obciążenia zewnętrzne powinny być sterowane przy użyciu bezpośrednich wyjść (Y).

4.4.3 Znaczniki kroku STL

Znaczniki (S) są bardzo ważnymi urządzeniami podczas programowania sterowania procesem krok po kroku. Są one używane w kombinacji z podstawową instrukcją STL.

Kiedy używane są wszystkie sposoby programowania STL, niektóre stany mają predefiniowaną operację. Krok oznaczony (1) na schemacie obok nazywa się "stanem inicjującym". Wszystkie inne kroki stanu są następnie używane do budowania pełnego schematu funkcji STL. Należy pamiętać, że chociaż pozostałe kroki stanu są używane w formacie STL, wciąż zachowują swój ogólny lub zatraskowy stan. Zakres dostępności urządzeń jest podany w tabeli punktu poprzedniego.



Przyporządkowane stany:

- Gdy wykonywana jest instrukcja specjalna IST (Initial State FNC 60) następujące urządzenia stanu są automatycznie przypisywanymi operacjami, które nie mogą być zmieniane przez program użytkownika:

S0	: stan inicjujący ręcznego sterowania
S1	: stan inicjujący powrotu do zera
S2	: stan inicjujący sterowania automatycznego
S10-S19	: przydzielone dla tworzenia sekwencji programu powrotu do zera.

Monitorowanie programów STL:

- Aby monitorować dynamicznie aktywne stany w programie STL, specjalny przekaźnik pomocniczy M8047 musi być w stanie ON.

Programowanie STL/SFC

- Więcej informacji na temat sposobu programowania STL/SFC znajduje się w rozdziale 3.

Instrukcja IST:

- Opis instrukcji IST znajduje się na stronie 5-67

4.4.4 Znaczniki wskaźnikowo-diagnostyczne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Niektóre znaczniki stanu mogą być użyte jako wyjścia dla diagnostyki zewnętrznej działania programu sterownika (zwane annunciator flags) jeśli użyto niektórych instrukcji specjalnych. Te instrukcje to:

instrukcja ANS FNC 46: ANunciator Set - zobacz strona 5-47

instrukcja ANR FNC 47: ANunciator Reset - zobacz strona 5-47

Podczas używania funkcji annunciatora, znaczniki kontrolowanego stanu są w zakresie S900-S999 (100 punktów). Przy zaprogramowaniu zewnętrznego układu diagnostyki, jak pokazano poniżej i monitorowaniu specjalnego rejestru danych D8049 wskazywany będzie w nich najniższy aktywowany stan z zakresu annunciator.

Każdemu z tych stanów może być przypisany błąd lub warunek błędu. Kiedy pojawi się błąd, przypisany stan jest ustawiony na ON. Jeśli pojawi się jednocześnie więcej błędów, wyświetlany jest najniższy numer błędu. Po usunięciu przyczyny wskazanego błędu wskazywany jest następny błąd o najniższym numerze. To oznacza, że przyczyny błędów o najwyższych priorytetach wskazywane w systemie diagnostycznym powinny mieć przydzielone stany na najniższych numerach zakresu annunciatora.

Wszystkie znaczniki stanu, używane w funkcji annunciatora wchodzą w zakres rejestrów stanu z podtrzymaniem/zatrząskowych.

Monitorowanie jest umożliwiające przez wysterowanie specjalnego przekaźnika pomocniczego M8049 w stan ON.

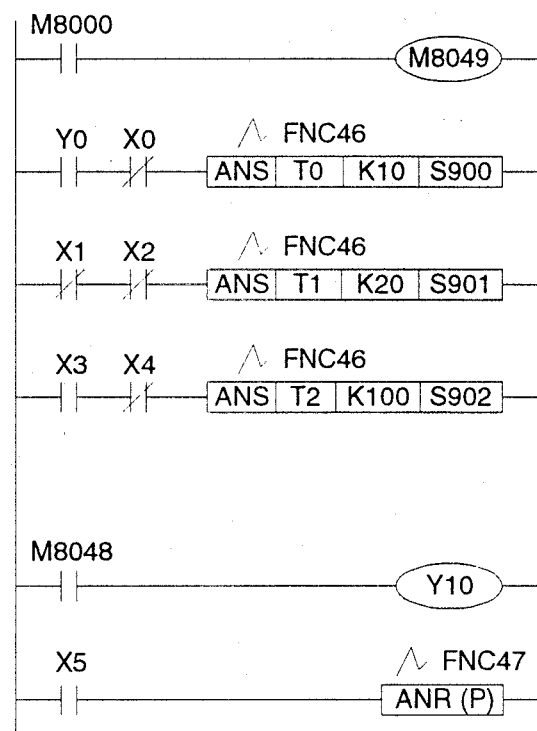
Znacznik S900 jest aktywowany, jeżeli wejście X0 nie jest wysterowane w przeciągu 1 s po tym, jak wyjście Y0 zostało ustawione w stan ON.

Znacznik S901 jest aktywowany, jeżeli oba wejścia X1 i X2 są w stanie OFF przez dłużej niż 2 s.

Jeżeli cykl sterowanej maszyny jest krótszy niż 10 s, a wejście X3 pozostaje w stanie ON, znacznik S902 będzie ustawiony w stan ON, jeżeli X4 nie zostanie zaktywowane w czasie cyklu maszyny.

Jeśli jakikolwiek znacznik z zakresu S900-S999 jest aktywowany, tzn. jest w stanie ON, specjalny przekaźnik pomocniczy M8048 jest aktywowany w celu włączenia wyjścia wskaźnika błędu Y10.

Znaczniki aktywowane przez program użytkownika, wykrywający błędy, są ustawiane w stan OFF przez aktywowanie wejścia X5. Za każdym razem, kiedy X5 jest aktywowany, aktywne znaczniki annunciatora są zerowane w narastającym porządku numerów stanów.



4.5 Etykiety skoków

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic: P (Pointer)

Zastosowanie: sterowanie przebiegiem programu

Oznaczenie: Etykieta

Etykieta programu

P

Dostępne formy: etykieta: pojawia się po lewej stronie linii zasilającej (bazowej), gdy program jest przeglądany w trybie drabinkowym

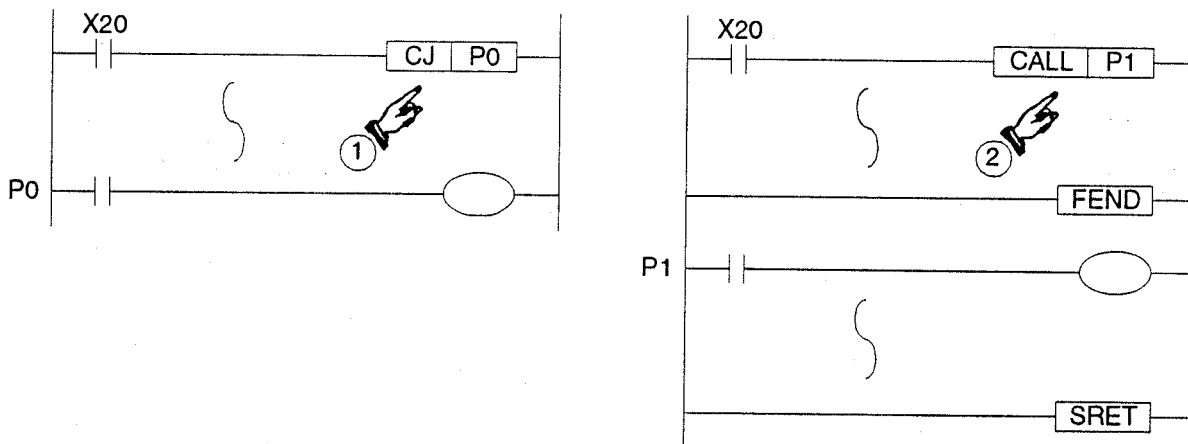
Numeracja etykiet: dziesiętna, tzn. P0-P9, P10-P19

Zastosowania: może być używana z instrukcjami skoku warunkowego (CJ FNC 00) -

zobacz strona 5-5 i element ① na przykładzie

z instrukcjami wywołania podprogramów (CALL FNC 01) - niedostępna w sterownikach FX0 i FX0N - zobacz strona 5-7 i element ② na przykładzie

Przykład użycia urządzenia:



Dostępne urządzenia:



- sterowniki programowalne FX0(S), FX0N i FX mają 64 etykiety; dostępne w zakresie P0-P63

- sterowniki programowalne FX(2C) i FX2N mają 128 etykiet; dostępnych w zakresie P0-P127

Skok na koniec programu

- W czasie używania instrukcji skoku warunkowego (CJ FNC 00) można przeskoczyć automatycznie na koniec programu, używając etykiety P63 w instrukcji CJ. Nie jest wymagane przypisanie etykiety P63 do instrukcji END.



Dostępność etykiet:

- Szczegółowe informacje o dostępności etykiet dla poszczególnych sterowników można znaleźć w rozdziale 8.

4.6 Etykiety przerwań

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik urządzenia: I (Interrupt)

Zastosowanie: miejsce początku programu obsługi przerwania

Oznaczenie: Przerwanie

Przerwanie szybkie

I

Dostępne formy: etykieta: pojawia się po lewej stronie lewej linii zasilającej (bazowej), gdy program jest przeglądany w trybie drabinkowym (zobacz ① na przykładzie)

Numerowanie etykiet: specjalny system numerowania, oparty na użytym wejściu przerwania i metodzie wyzwalania przerwania.

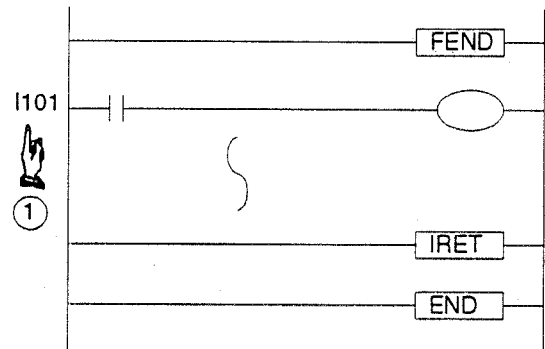
Dalsze zastosowania: przerwania od wejścia - zobacz strona 4-12

przerwania od timera - zobacz strona 4-12

maskowanie przerwania - zobacz strona 4-13

przerwania od licznika - zobacz strona 4-13

Przykład użycia urządzenia:



Dodatkowe instrukcje specjalne:

- przerwania są generowane przez urządzenia przerywające, wskaźnik przerwania i różne użycie trzech, dedykowanych specjalnych instrukcji przerwania.

- IRET FNC 03: powrót z przerwania - zobacz strona 5-9

- EI FNC 04: zezwolenie na przerwanie - zobacz strona 5-9

- DI FNC 05: zabronienie przerwania - zobacz strona 5-9

Poziomy zagnieżdżenia:

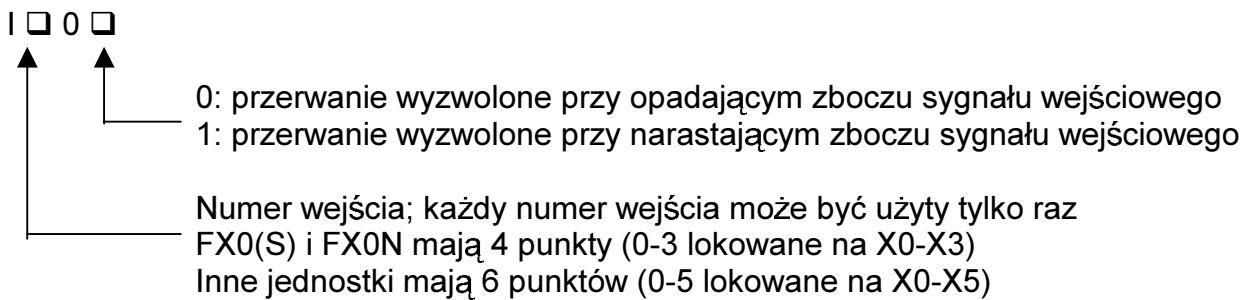
- Podczas obsługi przerwania wszystkie inne przerwania są zabronione. Aby umożliwić zagnieżdżone przerwania, w programie obsługi przerwania należy użyć instrukcji EI-DI. Przerwania mogą być zagnieżdżone na dwóch poziomach.

Pozycja etykiety:

- Etykiety przerwania mogą być użyte tylko po instrukcji FEND (FNC 06)

4.6.1 Przerwania od wejść

Identyfikacja numeru wskaźnika przerwania:



Przykład: I001

Sekwencja instrukcji umieszczona po etykiecie (oznaczonej przez wskaźnik I001) jest wykonywana po narastającym zboczach sygnału na wejściu X0. Program wraca z obsługi przerwania po napotkaniu instrukcji IRET.



Reguły użycia:

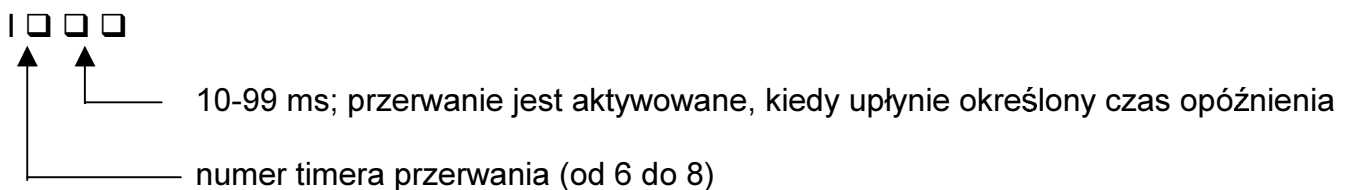
Dla działania przerw muszą być spełnione następujące warunki:

- etykiety przerw nie mogą mieć tych samych numerów na pozycji setek, tzn. I100 i I101 nie są dozwolone
- wejścia, używane do przerw, nie mogą pokrywać się z wejściami już przypisanymi do innych szybkich instrukcji w programie użytkownika

4.6.2 Przerwania od timerów

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Identyfikacja numeru wskaźnika przerwania:



Przykład: I610

Sekwencja instrukcji umieszczona po etykiecie (oznaczonej przez wskaźnik I610) jest wykonywana co 10 ms. Program wraca z obsługi przerwania po napotkaniu instrukcji IRET.



Reguły użycia:

- Dla działania przerw muszą być spełnione następujące punkty:
 - etykiety przerw nie mogą mieć tych samych numerów na pozycji setek, tzn. I610 i I650 nie są dozwolone

4.6.3 Blokowanie poszczególnych przerw

Poszczególne przerwy mogą być blokowane czasowo lub na stałe poprzez ustawienie w stan ON przypisanych specjalnych pomocniczych przełączników M8050-M8058. Odpowiednie przełączniki są oznaczone w tabelach urządzeń w rozdziale 6. Dla wszystkich typów PLC, przełącznikiem początkowym jest M8050, blokujący przerwy I0□□



Sterowanie specjalnych przełączników pomocniczych

- Nigdy nie należy sterować przełącznikiem bez uprzedniego sprawdzenia jego funkcji. Nie we wszystkich PLC ich funkcja jest taka sama.

Blokowanie przerw od szybkiego licznika

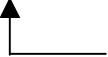
- Te przerwy mogą być blokowane tylko jako pojedyncza grupa, poprzez ustawienia znacznika M8059 w stan ON. Więcej szczegółów na temat przerw od licznika w następnej sekcji

4.6.4 Przerwy od liczników

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

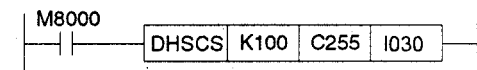
Identyfikacja numeru wskaźnika przerwy:

I 0 □ 0



numer przerwy licznikowej z zakresu od 1 do 6. Przerwy licznikowe mogą być wprowadzone jako urządzenia wyjściowe dla instrukcji HSCS FNC 53. Aby zablokować przerwy od licznika, znacznik M8059 musi być w stanie ON.

Przykład:



Sekwencja instrukcji umieszczona po etykiecie (oznaczonej I030) jest wykonywana kiedy licznik C255 osiągnie ustawiony limit K100, przedstawiony w przykładzie instrukcji HSCS.



Uwagi dodatkowe:

- Więcej szczegółów na temat instrukcji specjalnej HSCS FNC 53 na stronie 5-55

4.7 Stała K

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic urządzenia: K

Zastosowanie: oznaczenie stałej podanej w systemie dziesiętnym

Oznaczenie: Stała

K (wartość/stała)

K

Dostępne formy: wartość numeryczna; dla danych 16-bitowych wybierana z zakresu -32.768 do 32.767
dla danych 32-bitowych wybierana z zakresu -2.147.483.648 do 2.147.483.647

Numerowanie urządzeń: brak; Stała K służy do wprowadzenia danych do instrukcji. Nie ma ograniczeń na używanie stałej K w programie.

Dalsze użycie: Stała K może być używana z timerami, licznikami oraz z instrukcjami specjalnymi

Przykładowe użycie: patrz przykład z p. 4.6.4

4.8 Stała H

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic urządzenia: H

Zastosowanie: oznaczenie stałej podanej w systemie szesnastkowym

Oznaczenie: Stała

H (wartość/stała)

HEX (wartość/stała)

H

Dostępne formy: wartość alfanumeryczna tzn. 0-9, A-F (podstawa 16), dla danych 16-bitowych wybierana z zakresu 0 do FFFF
dla danych 32-bitowych wybierana z zakresu 0 do FFFFFFFF

Numerowanie urządzeń: brak; Stała H służy do wprowadzenia danych do instrukcji. Nie ma ograniczeń na używanie stałej H w programie.

Dalsze użycie: Stała H może być używana z instrukcjami specjalnymi

Przykładowe użycie: brak

4.9 Timery

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic urządzenia: T

Zastosowanie: wprowadzanie opóźnień czasowych

Oznaczenie: Timer(y) - przekaźnik czasowy

T

Dostępne formy: pobudzony przekaźnik (cewka) ustawia wewnętrzne programowe styki PLC (dostępne są styki normalnie rozwarte i normalnie zwarte). Są możliwe różne rozdzielczości timerów (1-100 ms), ale ich dostępność i ilość jest różna na różnym PLC. Dostępne są również następujące odmiany:

- przełączane rozdzielczości timera - zobacz strona 4-16
- timery z pamięcią - zobacz strona 4-17
- timery używane w przerwaniach i podprogramach (CALL) - zobacz strona 4-18

Numerowanie urządzeń: dziesiętne, tzn. T0-T9 i T10-T19

Dalsze użycie: brak

Przykładowe użycie urządzenia



Dostępne urządzenia:

Rozdzielczość timera	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
100 ms	56 (T0-55)	63 (T0-62)		200 (T0-199)	
10 ms	★24 (T32-55)	★31 (T32-62)		46 (T200-245)	
1 ms	brak	1 (T63)		brak	
1 ms z pamięcią	brak	brak		4 (T246-249)	
100 ms z pamięcią	brak	brak		6 (T250-255)	

★ timery o przełączanej rozdzielczości wybrane z głównego zakresu timerów 100 ms - zobacz strona 4-16



Dokładność timera:

- zobacz strona 4-18

4.9.1 Zasada działania timera

Zasada działania timera opiera się na zliczaniu systemowych impulsów zegarowych (1, 10, 100 ms). Styk wyjściowy timera jest aktywowany, gdy liczba zliczonych impulsów osiągnie zadaną wartość K. Całkowite opóźnienie lub odmierzony czas, dla cyklu operacji timera, jest obliczany poprzez mnożenie wartości stałej K przez rozdzielczość timera, tzn.

timer 10 ms z ustawioną wartością 567 przełącza styk wyjściowy po:

$$\begin{aligned} &567 * 10 \text{ ms} \\ &567 * 0,01 \text{ s} = 5,67 \text{ s} \end{aligned}$$

Wartość zadana timera może być podana bezpośrednio, przy użyciu stałej K lub pośrednio, przy użyciu danej zapamiętanej w rejestrze (np. D). Dla podania pośredniego zwykle używane są rejestry danych z podtrzymaniem/zatrzaszkowe; to zapewnia przechowanie danych w przypadku zaniku zasilania. Jeśli jednak bateria podtrzymująca będzie wyczerpana może dojść do błędnego działania timera.

4.9.2 Timery o przełączanej rozdzielczości

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

W niektórych sterownikach sterując specjalnym przekaźnikiem pomocniczym można redefiniować prawie połowę timerów 100 ms do rozdzielczości 10 ms. Następujące PLC i timery podlegają temu typowi selekcji

- FX₀, FX_{0S}, ustawiając M8028 w stan ON, zmienia się rozdzielczość timerów T32-T55 (24 sztuki) na 10 ms.
- FX_{0N}, ustawiając M8028 w stan ON, zmienia się rozdzielczość timerów T32-T62 (31 sztuk) na 10 ms.



Wybór rozdzielczości timera

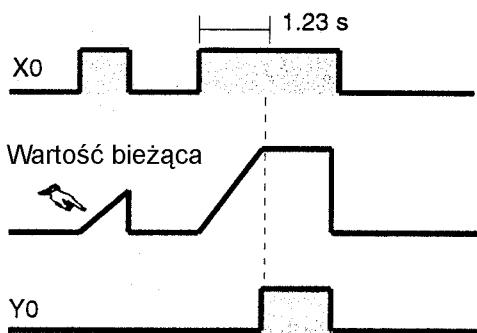
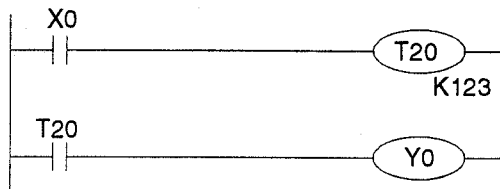
Należy sprawdzić rozdzielczość timera przed jego użyciem. Nie wszystkie PLC przyporządkowują tę samą rozdzielczość do tego samego numeru timera.

4.9.3 Timery z pamięcią

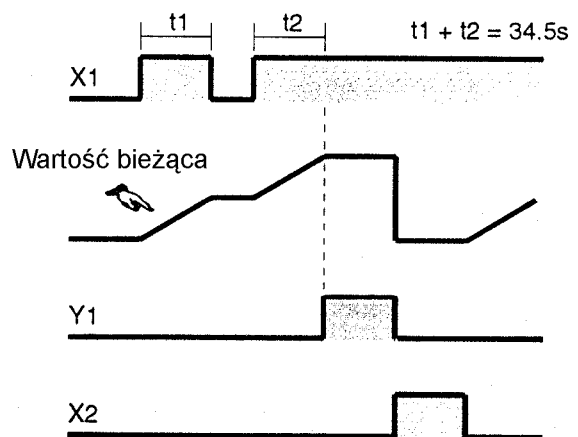
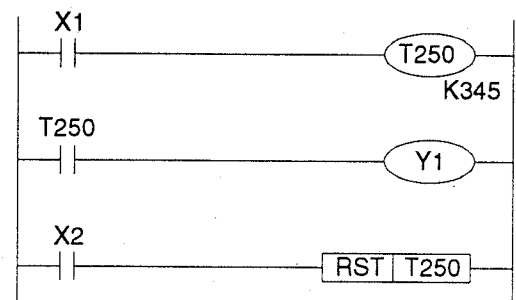
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Timer z pamięcią ma możliwość zapamiętania bieżącego stanu zliczenia po zaniku sygnału pobudzającego. Po ponownym jego pojawieniu się timer kontynuuje zliczanie od zapamiętanej wartości. Ponieważ timer z pamięcią nie jest zerowany po zaniku sygnału pobudzającego, należy to zerowanie wymusić. Poniższy przykład ilustruje taką sytuację.

Działanie timera bez pamięci



Działanie timera z pamięcią



Timery wykorzystywane w przerwaniach i podprogramach (CALL)

- zobacz strona 4-18

Dostępność urządzeń

- zobacz tabelę informacyjną na stronie 4-15

4.9.4 Timery wykorzystywane w przerwaniach i podprogramach (CALL)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Jeżeli timery T192-T199 i T246-T249 są używane w podprogramach "CALL" lub w programach obsługi przerw, zliczanie jest aktualizowane w trakcie wykonywania instrukcji END. Styk wyjściowy jest aktywowany, kiedy wykonywana jest instrukcja OUT timera lub instrukcja END, po osiągnięciu przez timer zadanej wartości opóźnienia.

Inne timery, niż wyszczególnione powyżej, nie mogą funkcjonować poprawnie w omawianych warunkach.

Kiedy timer "przerwaniowy" (o rozdzielczości 1 ms) jest używany w programie obsługi przerwa lub w podprogramie 'CALL', to jego styk wyjściowy jest aktywowany, po wykonaniu pierwszej instrukcji OUT dla tego timera, po osiągnięciu przez niego zadanej wartości opóźnienia.

4.9.5 Dokładność timera

Na dokładność timera może mieć wpływ konfiguracja programu. To znaczy, jeżeli styk timera jest użyty przed przypisaną mu cewką, wtedy dokładność timera jest zredukowana. Następujące formuły podają maksymalne i minimalne błędy dla takiej sytuacji.

Jednakże, średni oczekiwany błąd wynosi około:

$$1,5 * \text{czas przebiegu programu}$$

Przypadek 1:

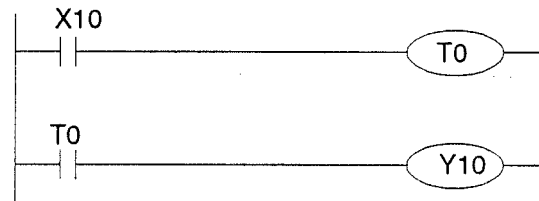
Styk timera pojawia się za cewką timera

Maksymalny błąd:

$$2 * \text{czas przebiegu programu} + \text{stała czasowa filtru wejściowego}$$

Minimalny błąd:

$$\text{stała czasowa filtru wejściowego} - \text{rozdzielczość timera}$$



Przypadek 2:

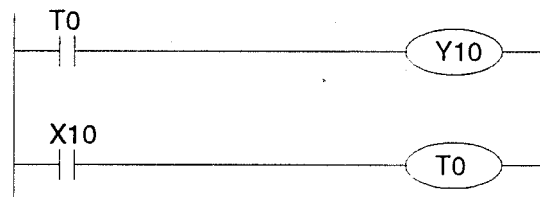
Styk timera pojawia się przed cewką timera

Maksymalny błąd:

$$3 * \text{czas przebiegu programu} + \text{stała czasowa filtru wejściowego}$$

Minimalny błąd:

$$\text{stała czasowa filtru wejściowego} - \text{rozdzielczość timera}$$



Dokładność timera wewnętrznego:

Faktyczna dokładność elementów czasowych w sprzęcie PLC wynosi:

+/- 10 impulsów na milion impulsów, tzn. jeżeli timer 100 ms jest używany do odmierzenia czasu jednej doby, zrealizuje pomiar z dokładnością 0.8 s na 24 godz (86.400 s). Timer zliczy w przybliżeniu 86.400/0,01 impulsów.

4.10 Liczniki

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemoniczny urządzenie: C

Zastosowanie: zliczanie zdarzeń zewnętrznych i wewnętrznych

Oznaczenie: Licznik(i)

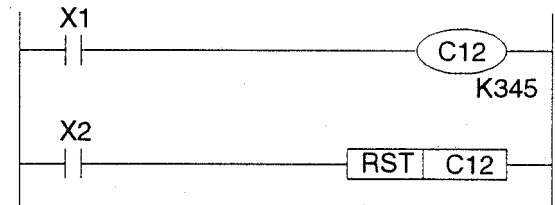
C

Dostępne formy: Pobudzana impulsowo cewka licznika ustawia wewnętrzne programowe styki PLC (dostępne są styki normalnie rozwarte i normalnie zwarte)
 Rodzaje liczników:
 liczniki zwykłe/zatraskowe, 16-bitowe, dodające - zobacz strona 4-20
 liczniki zwykłe/zatraskowe, 32-bitowe, dwukierunkowe - zobacz strona 4-21
 (dostępność i sposób użycia jest zależny od rodzaju sterownika - sprawdzić przed użyciem)

Numerowanie urządzeń: dziesiętne, tzn. C0-C9, C10-C19

Dalsze użycie: brak

Przykład zastosowania:



Dostępne urządzenia:

Rozdzielczość licznika	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
Licznik 16-bitowy, dodający	14 (C0-13)	16 (C0-15)		100 (C0-99)	
Licznik 16-bitowy, dodający, zatraskowy	2 (C14-15)	16 (C16-31)		100 (C100-199)	
Licznik 32-bitowy, dwukierunkowy	brak	brak		20 (C200-219)	
Licznik 32-bitowy, dwukierunkowy, zatraskowy	brak	brak		15 (C220-234)	



Liczniki szybkie:

- zobacz strona 4-22

Ustawianie zakresów dla liczników:

- liczniki 16-bitowe: -32.768 do 32.767
- liczniki 32-bitowe: -2.147.483.648 do 2.147.483.647

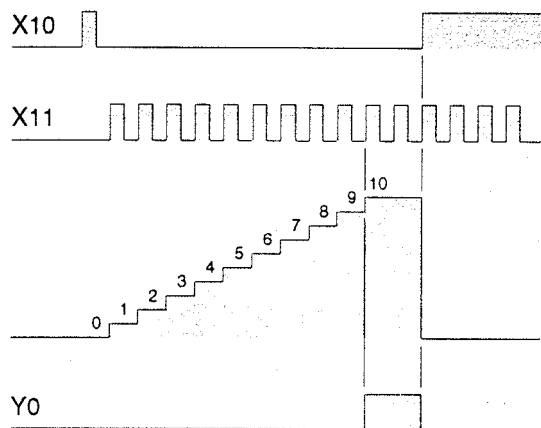
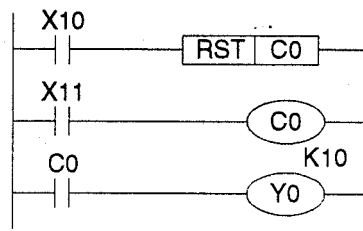
4.10.1 Liczniki 16-bitowe, dodające, zwykłe i zatraskowe

Bieżąca wartość licznika rośnie za każdym razem, kiedy cewka C0 jest pobudzona zmianą ze stanu OFF na stan ON za pomocą X11. Styk wyjściowy jest aktywowany, kiedy cewka jest ustawiona w stan ON za 10 razem (zobacz diagram). Potem wartość licznika pozostaje niezmienną, niezależnie od zmian na wejściu X11. W przykładzie bieżąca wartość licznika jest wyzerowana w czasie działania instrukcji RST, poprzez ustawienie X10 w stan ON. Styk wyjściowy Y0 jest również zerowany w tym samym czasie.

Stała licznika może być podana bezpośrednio, stałą K, lub pośrednio, przez użycie danej zapamiętanej w rejestrze (np. D). W ustawianiu pośrednim, przyporządkowanie rejestrowi D10, zawierającego np. wartość "123" ma ten sam efekt, co ustawienie "K123"

Zmiana stałej licznika rejestru pośredniego na wartość większą od wartości bieżącej licznika powoduje, że licznik liczy w górę przy następnym impulsie ON. Obowiązuje to we wszystkich typach liczników.

Częstotliwość impulsów zliczanych powinna wynosić ok. kilka cykli na sekundę.



Liczniki z podtrzymaniem/zatraskowe:

- Liczniki z podtrzymaniem/zatraskowe są w stanie przechować swój status nawet po wyłączeniu zasilania sterownika. Oznacza to, że po włączeniu zasilania, liczniki zatraskowe mogą natychmiast wznowić działanie od stanu przed wyłączeniem zasilania.



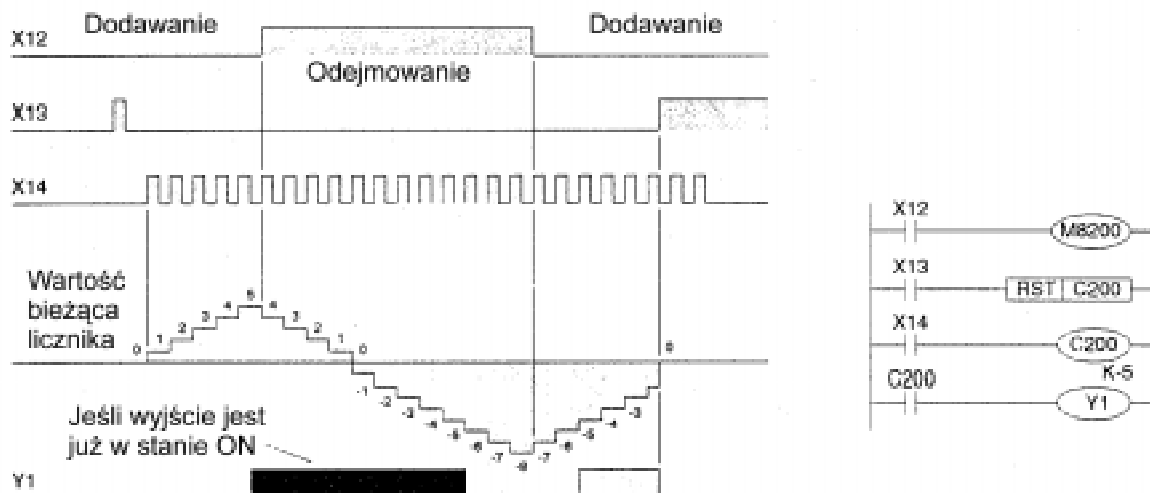
Dostępne urządzenia:

- zobacz tabela informacyjna na stronie 4-19

4.10.2 Liczniki 32-bitowe, dwukierunkowe, zwykle i zatraskowe

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Licznik pokazany na przykładzie poniżej aktywuje się po wystereowaniu jego cewki, C200. Przy każdej zmianie na wejściu X14, z OFF na ON, bieżąca wartość/zliczanie C200 jest inkrementowane



Styk licznika C200 jest aktywowany (styk NO przejdzie w stan ON), kiedy bieżąca wartość zmieni się z "-6" na "-5". Jednak, jeżeli wartość licznika maleje, z "-5" do "-6", styk licznika dezaktywuje się. Bieżąca wartość licznika rośnie lub maleje niezależnie od stanu jego styków wyjściowych (ON/OFF). Jeśli licznik zliczy powyżej 2.147.483.647, wartość bieżąca automatycznie zmieni się na -2.147.483.648. Podobnie, jeśli licznik zliczy poniżej -2.147.483.648, wartość bieżąca zmieni się na 2.147.483.647. Przepiętnienie licznika nie zmienia stanu jego styku. Ten typ sposobu liczenia jest charakterystyczny dla "liczników cyklicznych". Wartość bieżąca aktywnego licznika może być wyzerowana przez wymuszone resetowanie cewki licznika; w przykładowym programie przez przełączenie wejścia X13 w stan ON, co powoduje wykonanie instrukcji RST.

Kierunek zliczania jest przyporządkowany specjalnym pomocniczym przekaźnikiem M8200-M8234



Liczniki z podtrzymaniem/zatraskowe:

- Liczniki z podtrzymaniem/zatraskowe są w stanie przechować swój status nawet po wyłączeniu zasilania sterownika. Oznacza to, że po włączeniu zasilania, liczniki zatraskowe mogą natychmiast wznowić działanie od stanu przed wyłączeniem zasilania.



Dostępne urządzenia:

- zobacz tabelę informacyjną na stronie 4-19

Wybór kierunku zliczania:

- Jeśli M8☆☆☆ dla C☆☆☆ jest w stanie ON, licznik liczy w dół (licznik odejmujący). Odpowiednio, jeśli M8☆☆☆ jest w stanie OFF, licznik liczy w górę (licznik dodający).

4.11 Liczniki szybkie

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik urządzenia: C

Zastosowanie: zliczanie impulsów zewnętrznych o dużej częstotliwości

Oznaczenie: Licznik(i)

C

Licznik(i) szybkie

Liczniki fazowe

Dostępne formy: Pobudzana impulsowo cewka ustawia wewnętrzne programowe styki PLC (dostępne są styki normalnie rozwarne i normalnie zwarte)
Dostępne są różne odmiany szybkich liczników, ale ich liczba i funkcje są różne w różnych sterownikach.

Sprawdź następujące sekcje pod kątem dostępności urządzenia:

FX₀, FX_{0S} i FX_{0N} - zobacz strona 4-24

FX, FX_{2C}, FX_{2N} - zobacz strona 4-25

Następujące sekcje odnoszą się do typów liczników:

- liczniki 1-fazowe (uruchamiane i zerowane przez użytkownika) - zobacz strona 4-29
- liczniki 1-fazowe (z przydzielonym uruchamianiem i zerowaniem) - zobacz strona 4-30
- liczniki 2-fazowe, dwukierunkowe - zobacz strona 4-31
- liczniki A/B fazowe - zobacz strona 4-32

Numerowanie urządzeń: dziesiętne, tzn. C235-C255

Dalsze użycie: brak

Przykład użycia urządzenia: przykłady użycia dostępnych form znajdują się w odpowiednich sekcjach



Podstawowe operacje liczników szybkich

- zobacz strona 4-23

4.11.1 Podstawowe operacje na liczniku szybkim

Pomimo, że liczniki C235-C255 (21 sztuk) są licznikami szybkimi, wszystkie korzystają z tych samych wejść. Dlatego, jeżeli wejście jest już używane przez szybki licznik, nie może być użyte przez żadne inne szybkie liczniki, ani do innego celu, np. jako wejście przerwania.

Wybór szybkich liczników nie jest dowolny, jest on bezpośrednio zależny od typu wymaganego licznika i od dostępności wejść.

Dostępne typy liczników:

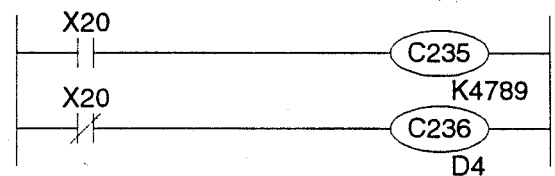
- 1-fazowy z uruchamianiem i zerowaniem programowym: C235-C240
- 1-fazowy z uruchamianiem i zerowaniem sprzętowym: C241-C245
- 2-fazowy, dwukierunkowy: C246-C250
- A/B fazowy: C251-C255

Należy zauważyć, że wszystkie te liczniki są 32-bitowe.

Liczniki szybkie działają na zasadzie przerwań. To znaczy, że są wyzwalane zdarzeniami i są niezależne od czasu cyklu. Cewka wybranego licznika powinna być sterowana w sposób ciągły dla zaznaczenia, że żaden inny szybki proces nie może jednocześnie korzystać z tego licznika i związanych z nim wejść.

Przykład:

Kiedy X20 jest w stanie ON, wybrany jest szybki licznik C235. Licznikowi C235 przyporządkowane jest wejście X0. X20 nie jest zliczanym sygnałem. Jest on ciągłym sterowaniem, wspomnianym wcześniej. X0 nie musi występować w programie. Przyporządkowanie wejścia jest uzależnione sprzętowo i nie może być zmienione przez użytkownika.



Kiedy X20 jest w stanie OFF, cewka C235 jest ustawiona w stan OFF a cewka C236 jest ustawiona w stan ON. Licznik C236 ma przyporządkowane wejście X1, i jak poprzednio X20 nie jest wejściem zliczanym.

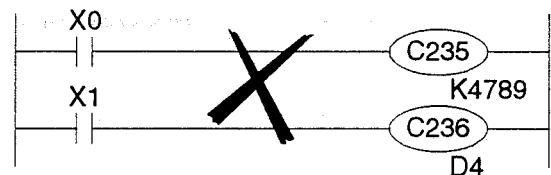
Przyporządkowanie liczników i urządzeń wejściowych zależy od wybranego PLC. Zostało to opisane w odpowiednich, późniejszych sekcjach.



Sterowanie cewek szybkich liczników:

- Wejścia zliczające nie są używane do sterowania cewek szybkich liczników.

Cewka licznika musi być wystawiona w stan ON w sposób ciągły dla obsługi szybkich wejść. Dlatego do sterowania cewką szybkiego licznika powinien być użyty inny styk. Idealnie nadaje się do tego celu styk M8000, jednak nie jest to obowiązkowe.



4.11.2 Dostępność liczników szybkich w programowalnych sterownikach FX₀, FX_{0S} i FX_{0N}

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Następująca tabela pokazuje zakres dostępnych szybkich liczników w sterownikach FX₀, FX_{0S} i FX_{0N}

Wejście	1-fazowy licznik ze startem/zerowaniem programowym				1-fazowy licznik ze startem/zerowaniem sprzętowym			2-fazowy licznik, dwukierunkowy			A/B-fazowy licznik		
	C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
X0	U/D				U/D		U/D	U	U	U	A	A	A
X1		U/D			R		R	D	D	D	B	B	B
X2			U/D			U/D			R	R		R	R
X3				U/D		R	S			S			S

Klucz: **U** - wejście licznika liczącego w górę

R - zerowanie licznika (wejście)

A - wejście fazowe A licznika

D - wejście licznika liczącego w dół

S - uruchomienie licznika (wejście)

B - wejście fazowe B licznika

C235 - licznik z podtrzymywaniem/zatrząskowy w sterownikach FX₀, FX_{0S} i FX_{0N}

C236 - licznik z podtrzymywaniem/zatrząskowy tylko w sterowniku FX_{0N}

(FX₀, FX_{0S} nie posiada urządzeń z podtrzymaniem/zatrząskowych)



Przyporządkowanie wejść:

Różne typy liczników mogą być używane w tym samym czasie, ale ich wejścia nie mogą być wykorzystane jednocześnie. Wejścia X0-X3 nie mogą być używane dla więcej niż jednego licznika np. jeżeli użyte jest C251, następujące liczniki i instrukcje nie mogą być użyte: C235, C236, C241, C244, C246, C247, C249, C252, C254, I0□□, I1□□

Szybkości liczników i reguły dotyczące działania:

Typ	Maks. szybkość zliczania 1-fazowego	Suma szybkości aktywnych liczników 1-fazowych	Maks. szybkość zliczania 2-fazowego	Maks. liczba liczników 2-fazowych	Mak. połączona suma szybkości zliczania 1- i 2-fazowych
FX ₀ , FX _{0N}	5 kHz	≤ 5kHz	2 kHz	1	liczniki 1- i 2-fazowe nie mogą być łączone
FX _{0S}	7 kHz	≤ 14kHz	2 kHz	1	≤ 14kHz, patrz uwaga niżej

- Maksymalna częstotliwość sygnału na wejściach 5 kHz.
- Tylko jeden 2-fazowy albo A/B fazowy licznik może być użyty w tym samym czasie.
- Licznik szybki, wyspecyfikowany w instrukcji specjalnej, nie może być modyfikowany przez rejestry indeksowe V lub Z.



Obliczanie maksymalnej połączonej szybkości liczenia w sterowniku FX_{0S}:

Sposób obliczenia: (2-fazowy licznik szybki x liczba zliczonych zboczy) + (suma szybkości aktywnych liczników 1-fazowych)

4.11.3 Dostępność liczników szybkich w programowalnych sterownikach FX, i FX_{2C}

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Następująca tabela pokazuje zakres dostępnych szybkich liczników w sterownikach FX i FX_{2C}

Wejście	1-fazowy licznik z programowym startem/zerowaniem						1-fazowy licznik ze sprzętowym startem/zerowaniem					2-fazowy licznik dwukierunkowy					Licznik A/B-fazowy				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X0 ◆	U/D						U/D		U/D		U	U		U		A	A		A		
X1 □		U/D					R		R		D	D		D		B	B		B		
X2 ◆			U/D					U/D		U/D			R		R			R		R	
X3 ◆				U/D				R		R				U	U				A	A	
X4 □					U/D				U/D					D	D				B	B	
X5 □						U/D			R					R	R				R	R	
X6									S						S				S		
X7										S					S					S	

Klucz: **U** - wejście licznika liczącego w górę

R - zerowanie licznika (wejście)

A - wejście fazowe A licznika

C235 - licznik z podtrzymywaniem/zatraskowy

D - wejście licznika liczącego w dół

S - uruchomienie licznika (wejście)

B - wejście fazowe B licznika



Przyporządkowanie wejść:

- X6 i X7 są również szybkimi wejściami, ale funkcjonują tylko jako sygnały startu dla wybranych typów liczników.
- Różne typy liczników mogą być używane w tym samym czasie, ale ich wejścia nie mogą być wykorzystane jednocześnie. Np. jeżeli użyte jest C247, następujące liczniki i instrukcje nie mogą być użyte: C235, C236, C237, C241, C242, C244, C245, C246, C249, C251, C252, C254, I0□□, I1□□, I2□□
- Wejścia oznaczone □ mogą pracować do 7 kHz, a oznaczone ◆ do 10 kHz.

**Szybkości licznika:**

- Maksymalna szybkość liczenia zależy od typu, ilości liczników i od tego, ile używanych jest instrukcji szybkich liczników. Następujące tabele pokazują przybliżoną maksymalną szybkość liczenia dla każdego zdefiniowanego przypadku.
- Należy ostrożnie używać instrukcji SPD FNC 56. Ta instrukcja jest traktowana, tak jakby to był licznik 1-fazowy. To należy wziąć pod uwagę przy obliczaniu sumy szybkości liczenia.

Liczniki 1-fazowe

Wejście licznika	Liczba liczników	Częstotliwość w kHz		
		Brak wykonania instrukcji dużej szybkości	Wykonanie (D)HSCS/R (1 do 6 instrukcji)	Wykonanie (D)HSZ (1 do 2 instrukcji)
X0, X2, X3 (wejścia 10 kHz)	1	10	7	5
	2	10	4	2,5
	3	6,6	2,5	
X0 do X5 (gdy X0, X2 i X3 są współużywane)	1	7	5	4
	2	3,5	2,5	1,5
	3	2,5	2	
	4			
	5		1,5	
	6			

Liczniki A/B-fazowe

Licznik	Liczba liczników	Częstotliwość w kHz		
		Brak wykonania instrukcji dużej szybkości	Wykonanie (D)HSCS/R (1 do 6 instrukcji)	Wykonanie (D)HSZ (1 do 2 instrukcji)
C251-C255	1	2	2	2
	2		1,5	1,5

Liczniki A/B-fazowe, używane albo z licznikiem 1- albo 2-fazowym.

Częstotliwość licznika A/B-fazowego musi być utrzymywana poniżej 1 kHz. Maksymalna częstotliwość 1- lub 2-fazowego podana jest w poniższej tabeli

Licznik	Liczba liczników	Częstotliwość w kHz		
		Brak wykonania instrukcji dużej szybkości	Wykonanie (D)HSCS/R (1 do 6 instrukcji)	Wykonanie (D)HSZ (1 do 2 instrukcji)
z 1 A/B-fazowym licznikiem przy 1 kHz	1	5	4	3
	2	4	2	1
	3	3		
	4	2		



Uwaga: Liczniki dwukierunkowe są zaprojektowane w ten sposób, że sygnał zliczania w górę i sygnał zliczania w dół nigdy nie działają jednocześnie. Dlatego używana jest tylko jedna faza w jednym czasie. Wobec tego mogą być traktowane jak liczniki 1-fazowe, podczas obliczania sumarycznej częstotliwości.

Częstotliwość sumaryczna:

- Częstotliwość sumaryczna jest sumą maksymalnych częstotliwości wszystkich sygnałów pojawiających się jednocześnie na wejściach sterownika. Kryterium poprawnego działania szybkich liczników jest takie, że sumaryczna częstotliwość musi być mniejsza od 20 kHz.

Przykład:

Liczniki 1-fazowe	Odpowiadające wejście	Maksymalna częstotliwość sygnału
C235	X0	4,2 kHz
C237	X2	4 kHz
C240	X3	6 kHz
Częstotliwość sumaryczna		14,2 kHz

Sumaryczna częstotliwość 14,2 kHz jest mniejsza od maksymalnej 20 kHz, tak więc ten przykład jest poprawny.

Liczniki A/B-fazowe:



Przy obliczaniu częstotliwości sumarycznej, która zawiera licznik A/B-fazowy, maksymalna częstotliwość liczenia powinna być pomnożona przez 4 przed dodaniem maksymalnej częstotliwości współbieżnych liczników.

Przykład:

Liczniki	Odpowiadające wejście	Maksymalna częstotliwość sygnału
1-fazowy C237	X2	3 kHz
dwukierunkowy C246	X0, X1	4 kHz
A/B-fazowy C255	X3, X4	1 kHz x 4
Częstotliwość sumaryczna		$3 + 4 + (1 \times 4) = 11 \text{ kHz}$

Sumaryczna częstotliwość 11 kHz jest mniejsza od maksymalnej 20 kHz, tak więc ten przykład jest poprawny.



Liczniki 2-fazowe:

- Kiedy impulsy docierają na oba wejścia liczące w górę i w dół w tym samym czasie, należy to traktować jako dwa 1-fazowe liczniki podczas obliczania częstotliwości sumarycznej. Kodery formatu ClockWise (zgodnie z ruchem wskazówek zegara) i Counter-ClockWise (przeciwnie do ruchu wskazówek zegara):
- Kiedy używane są kodery z wejściami w formacie CW i CCW, liczniki dwukierunkowe mogą liczyć z dużo wyższą częstotliwością niż liczniki A/B-fazowe, bez utraty rozdzielczości.

4.11.4 Dostępność liczników szybkich w programowalnych sterownikach FX_{2N}

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Następująca tabela pokazuje zakres dostępnych szybkich liczników w sterownikach FX_{2N}

Wejście	1-fazowy licznik z program. startem/ zerowaniem						1-faz. licznik ze sprzęt. startem/ zerowaniem					2-fazowy licznik dwukierunkowy					Licznik A/B-fazowy				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X0	U/D						U/D			U/D		U	U		U		A	A		A	
X1		U/D					R			R		D	D		D		B	B		B	
X2			U/D					U/D		U/D			R		R			R		R	
X3				U/D				R		R				U	U				A		A
X4					U/D				U/D					D	D				B		B
X5						U/D			R					R	R				R		R
X6										S					S					S	
X7										S					S						S

Klucz:
U - wejście licznika liczącego w górę
R - zerowanie licznika (wejście)
A - wejście fazowe A licznika
 C235 - licznik z podtrzymaniem/zatrząskowy

D - wejście licznika liczącego w dół
S - uruchomienie licznika (wejście)
B - wejście fazowe B licznika

**Przyporządkowanie wejść:**

- X6 i X7 są również szybkimi wejściami, ale funkcjonują tylko jako sygnały startu dla wybranych typów liczników.
- Różne typy liczników mogą być używane w tym samym czasie, ale ich wejścia nie mogą być wykorzystane jednocześnie. Np. jeżeli użyte jest C247, następujące liczniki i instrukcje nie mogą być użyte: C235, C236, C237, C241, C242, C244, C245, C246, C249, C251, C252, C254, I0□□, I1□□, I2□□

Szybkości licznika:

- Ogólne częstotliwości zliczania:
 - liczniki 1-fazowe i dwukierunkowe - do 10 kHz
 - liczniki A/B-fazowe - do 5 kHz
 - maksymalna sumaryczna częstotliwość zliczania - 20 kHz (licznik A/B-fazowy liczy podwójnie)



- Wejścia X0 i X1 mają specjalne wyposażenie sprzętowe, które umożliwia następujące bardzo szybkie zliczanie:
 - zliczanie 1-fazowe lub dwukierunkowe C235, C236 lub C246 - do 60 kHz
 - zliczanie 2-fazowe C251 - do 30 kHz



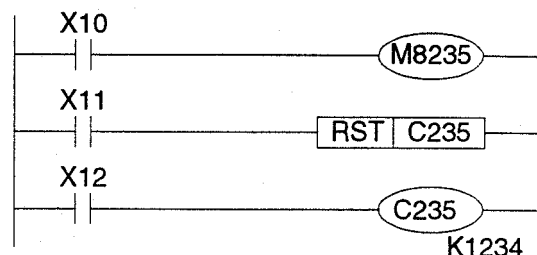
Jeśli użyte są instrukcje szybkiego porównania (FNC 53, 54, 55), wejścia X0 i X1 muszą się odwołać do liczenia programowego. W tym przypadku stosuje się ogólne częstotliwości zliczania.

4.11.5 Liczniki jednofazowe - uruchamiane i zerowane przez program użytkownika (C235-C240)

Każdy z tych liczników korzysta tylko z jednego wejścia. Kiedy znacznik kierunku M8235 jest w stanie ON, licznik C235 zlicza w dół. Kiedy znacznik jest w stanie OFF, C235 zlicza w górę.

Kiedy X11 jest w stanie ON, C235 zeruje się. Wszystkie styki licznika C235 również są wyzerowane.

Kiedy X12 jest w stanie ON, C235 jest wybrany. W poprzednich tablicach liczników X0 odpowiada wejściu liczącemu licznika C235. Dlatego licznik C235 liczy ile razy stan X0 zmienił się z OFF na ON.



Specyfikacja urządzenia:

- Wszystkie te liczniki są 32-bitowymi, liczącymi w górę i w dół licznikami cyklicznymi. Ich działanie jest takie samo, jak normalnych 32-bitowych, liczących w górę i w dół, liczników, opisanych na stronie 4-21. Kiedy wartość bieżąca liczników osiągnie swoją wartość maksymalną lub zadaną, styki przyporządkowane licznikom są ustawiane i stan ich jest utrzymywany, kiedy licznik liczy w górę. Jednak, kiedy licznik liczy w dół, styki są zerowane.

Zakres ustawień:

- -2.147.483.648 do 2.147.483.647

Ustawienia kierunku:

- Kierunek zliczania dla liczników 1-fazowych zależy od stanu odpowiadającego im znacznika M8☆☆☆, gdzie ☆☆☆ jest numerem odpowiedniego licznika (C235-C240).
 - Kiedy M8☆☆☆ jest w stanie ON, licznik liczy w dół
 - Kiedy M8☆☆☆ jest w stanie OFF, licznik liczy w górę



Używanie instrukcji SPD:

- Należy zachować ostrożność podczas używania instrukcji specjalnej SPD (FNC 56). Instrukcja ta ma charakter zarówno licznika szybkiego jak i przerwania, dlatego urządzenia wejściowe X0-X5 mogą być użyte jako źródło dla instrukcji SPD. Tak jak dla wszystkich szybkich procesów, wybrane urządzenie źródłowe instrukcji SPD nie może kolidować z żadną inną szybką funkcją, która właśnie działa, tzn. szybkie liczniki lub przerwania używające tego samego wejścia.

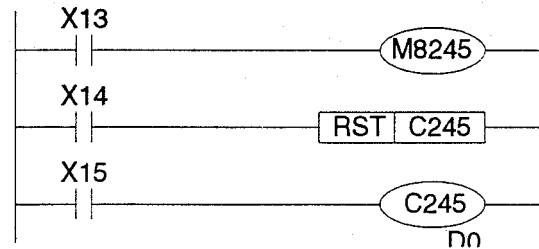
Kiedy używana jest instrukcja SPD, rozpoznawana jest przez system jako 1-fazowy, szybki licznik. Powinno to być wzięte pod uwagę podczas obliczania maksymalnej częstotliwości sumarycznej sygnałów wejściowych - zobacz poprzednią sekcję.

4.11.6 Liczniki jednofazowe - z uruchamianiem i zerowaniem sprzętowym (C246-C250)

Te liczniki mają po jednym wejściu liczącym i po jednym wejściu zerującym każdy. Liczniki C244 i C245 mają dodatkowo wejście startu.

Kiedy znacznik kierunku M8245 jest w stanie ON, C245 liczy w dół. Kiedy znacznik ten jest w stanie OFF, C245 liczy w górę.

Kiedy X14 jest w stanie ON, C245 zeruje się w taki sam sposób, jak normalne 32-bitowe liczniki wewnętrzne, ale C245 może być również wyzerowane wejściem X3. Odbywa się to automatycznie, kiedy używany jest licznik C245 (zobacz poprzednie tabele licznika).



Licznik C245 ma również zewnętrzny automatyczny styk startowy. Jest to wejście X7. Te dane można również znaleźć w poprzednich tabelach licznika.

Kiedy X7 jest w stanie ON, C245 zaczyna liczenie i odwrotnie, kiedy X7 jest w stanie OFF, C245 zatrzymuje się. Wejście X15 wybiera i rezerwuje to przyporządkowane wejście dla wybranego licznika, w tym przypadku C245.

Powód, dla którego te liczniki używają przydzielonych wejść startu (X7) i zerowania (X3), jest taki, że nie są związane z cyklem programu sterownika, tzn. ich działanie jest natychmiastowe i bezpośrednie.

W tym przykładzie C245 liczy zmiany stanu z OFF na ON na wejściu X2.

Uwaga: ponieważ C245 jest licznikiem 32-bitowym, musi być ustawiany przez daną 32-bitową. To oznacza, że rejestry danych D1, D0 są używane jako para dla uzyskania wymaganego formatu 32-bitowego.



Specyfikacja urządzeń:

- Wszystkie liczniki szybkie są licznikami 32-bitowymi, liczącymi w górę i w dół oraz licznikami cyklicznymi. Ich działanie jest takie samo, jak normalnych 32-bitowych liczników, liczących w górę i w dół, opisanych na stronie 4-21. Kiedy wartość bieżąca liczników osiągnie swoją wartość zadaną, styki przyporządkowane licznikom są ustawiane i stan ich jest utrzymywany, kiedy licznik liczy w górę. Jednak, kiedy licznik liczy w dół, osiągnięcie wartości o 1 mniejszej od wartości zadanej powoduje, że styki są zerowane.

Zakres ustawień:

- -2.147.483.648 do 2.147.483.647

Ustawienia kierunku:

- Kierunek zliczania dla liczników 1-fazowych zależy od stanu odpowiadającego im znacznika M8☆☆☆, gdzie ☆☆☆ jest numerem odpowiedniego licznika (C241-C245).
 - Kiedy M8☆☆☆ jest w stanie ON, licznik liczy w dół
 - Kiedy M8☆☆☆ jest w stanie OFF, licznik liczy w górę

4.11.7 Liczniki dwufazowe, dwukierunkowe (C246-C250)

Te liczniki mają jedno wejście dla liczenia w górę i jedno wejście dla liczenia w dół. Niektóre liczniki mają również wejścia zerowania i startu.

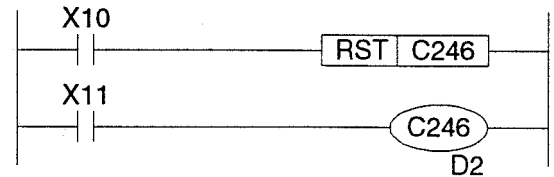
Kiedy X10 jest w stanie ON, C246 zeruje się w taki sam sposób jak standardowy licznik 32-bitowy.

Licznik C246 używa wejść:

X0 do liczenia w górę i

X1 do liczenia w dół.

W celu umożliwienia liczenia, wejście sterujące X11 musi być w stanie ON, aby ustawić i zarezerwować przypisane do licznika wejścia.



Uwaga:

Zmiana stanu wejścia X0 z OFF na ON zwiększa licznik C246 o 1

Zmiana stanu wejścia X1 z ON na OFF zmniejsza licznik C246 o 1

Licznik dwukierunkowy C250 może być widziany jako licznik mający wejście X5 jako zerujące i X7 jako zezwalające. Dlatego operacja zerowania może być przeprowadzona zewnętrznie, bez instrukcji RST C250.



X13 musi być w stanie ON, żeby wybrać C250, ale

wejście zezwalające X7 musi być w stanie ON, żeby umożliwić C250 liczenie. Jeżeli X7 przechodzi w stan OFF, liczenie zostaje wstrzymane.

Licznik C250 używa wejścia X3 do liczenia w górę i wejścia X4 do liczenia w dół.



Specyfikacja urządzeń:

- Wszystkie te liczniki są 32-bitowe.

Zakres ustawień:

- -2.147.483.648 do 2.147.483.647

Ustawienie kierunku:

- Kierunek zliczania dla liczników 1-fazowych zależy od stanu odpowiadającego im znacznika M8☆☆☆, gdzie ☆☆☆ jest numerem odpowiedniego licznika (C241-C245).
 - Kiedy M8☆☆☆ jest w stanie ON, licznik liczy w dół
 - Kiedy M8☆☆☆ jest w stanie OFF, licznik liczy w górę

4.11.8 Liczniki A/B fazowe (C252-C255)

Dla tych liczników tylko wejście opisane w poprzednich tabelach szybkich liczników może być użyte do liczenia. Liczenie wykonywane przez te urządzenia jest niezależne od czasu przebiegu programu sterownika. W zależności od użytego licznika, start, zerowanie i inne przyporządkowane wejścia są przydzielane automatycznie.

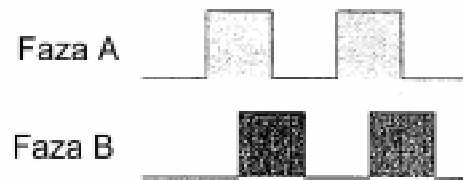
Sygnaly wejściowe fazy A i B nie tylko dostarczają impulsów do zliczania, ale również wg zależności między nimi określany jest kierunek zliczania, wg następującej reguły:

Jeśli sygnał fazy A jest w stanie ON i sygnał fazy B przechodzi ze stanu OFF na ON, to licznik będzie liczył w górę.

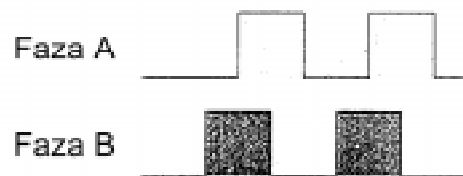
Jeśli sygnał fazy A jest w stanie OFF i sygnał fazy B przechodzi ze stanu OFF na ON, licznik będzie liczył w dół.

Zliczanie w danym kierunku zachodzi po tym, jak sygnały obu faz A i B wystąpią w odpowiednim porządku.

Dodawanie



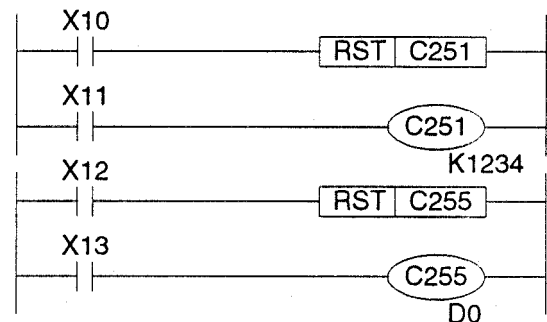
Odejmowanie



Licznik C251 liczy zmiany ON/OFF na wejściu X0 (wejście fazy A) i wejściu X1 (wejście fazy B) podczas gdy X11 jest w stanie ON.

C255 zaczyna zliczać natychmiast po ustawieniu X7 w stan ON, podczas gdy X13 jest już w stanie ON. Wejściami liczącymi są X3 (faza A) i X4 (faza B).

C255 jest zerowany, kiedy X5 jest ustawione w stan ON. Może on być również wyzerowany wejściem X12.



Specyfikacja urządzeń:

Maksymalnie można użyć jednocześnie 2 liczników - 2-fazowych, 32-bitowych, liczących w górę/w dół. Operacje na stykach wyjściowych w zależności od zliczonych wartości są takie same jak dla standardowych liczników, opisanych w punkcie 4.11

Zakres ustawień:

- 2.147.483.648 do 2.147.483.647

Ustawienie kierunku:

- Aby określić, czy licznik liczy w górę czy w dół, należy sprawdzić stan odpowiadającego mu przekaźnika M8☆☆☆.

4.12 Rejestry danych

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic urządzenia: D

Zastosowanie: komórka pamięci, urządzenie przechowujące, zdolne do przechowania danych numerycznych lub wzorców 16/32 bitowych

Oznaczenie: Dane (rejestr/urządzenie/słowo)

D (rejestr)

D

Słowo

Dostępne formy: rejestry ogólnego przeznaczenia - zobacz strona 4-34
 rejestry z podtrzymaniem/zatrzaskowe - zobacz strona 4-35
 specjalne rejestry diagnostyczne - zobacz strona 4-35
 rejestry plikowe - zobacz strona 4-36
 rejestry plikowe RAM - zobacz strona 4-36
 rejestry ustawiane zewnętrznie - zobacz strona 4-37

Numerowanie urządzeń: dziesiętne, tzn. D0-D9, D10-D19

Dalsze użycie: może być użyte do pośredniego ustawiania stałych dla liczników i timerów.

Przykład zastosowania: brak

Dostępne urządzenia:

	FX ₀	FX _{0N}	FX (CPU v 2.3)	FX _(2C)	FX _{2N}
Rejestry ogólnego przeznaczenia	30 (D0-29)	128 (D0-127)	200 (D0-199)		
Rejestry zatrzaskowe	2 (D30-31)	128 (D128-256)	312 (D200-512)	800 (D200-999)	7800 (D200-7999)
Rejestry diagnostyczne	27 (D8000-8069)	39 (D8000-8255)	256 (D8000-8255)		
Rejestry plikowe ★	Brak	1500 (D1000-2499)	2000 (D1000-2999)		7000 (D1000-7999)
Rejestry plikowe RAM ★	brak			2000 (D6000-7999)	brak
Rejestry ustawiane ♦	1 (D813)	2 (D8030-8031)	brak		

★ - te urządzenia są przydzielone przez użytkownika kosztem dostępnych kroków programu.

W sterownikach FX2N te urządzenia są podzbiorem rejestrów zatrzaskowych

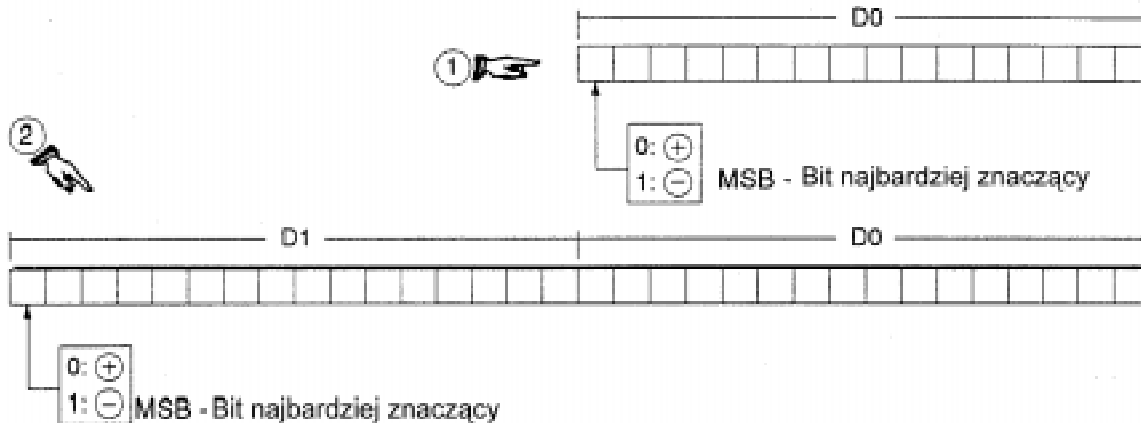
♦ - te urządzenia są również zaliczone do rejestrów diagnostycznych

★ - te urządzenia są aktywowane, kiedy specjalny przełącznik pomocniczy M8074 jest ustawiony w stan ON. Rejestry plikowe RAM nie zajmują kroków programu.

4.12.1 Rejestry ogólnego przeznaczenia

Rejestry danych, jak sama nazwa wskazuje, przechowują dane. Przechowywane dane mogą być zinterpretowane jako wartość numeryczna lub jako szereg bitów, które mogą być w stanie ON lub OFF.

Pojedynczy rejestr danych zawiera 16 bitów lub jedno słowo. Jednakże dwa kolejne rejestry danych mogą być użyte do tworzenia urządzenia 32-bitowego, znanego jako podwójne słowo. Jeśli zawartość rejestru danych jest traktowana jako dana numeryczna, wtedy najbardziej znaczący bit (MSB) jest używany do oznaczenia znaku (dodatni lub ujemny) danej. Wartość 0 oznacza liczbę dodatnią, a 1 - liczbę ujemną.



Powyższy diagram pokazuje zarówno pojedynczą jak i podwójną konfigurację rejestru. W diagramie oznaczonym (2) należy zauważyć, że młodszy rejestr D0 nie ma już MSB. Tak się dzieje ponieważ jest on teraz uważany jako część 32-bitowego podwójnego słowa. MSB zawsze będzie się znajdował na najstarszym, 16. bicie, w tym przypadku w D1. Gdy używa się 32-bitowego rejestru w instrukcji programu, zawsze podaje się młodszy rejestr, np. użyty w powyższym przykładzie 32-bitowy argument instrukcji będzie oznaczony jako D0. Drugi rejestr, D1, będzie przyporządkowany automatycznie.

Po wpisaniu danej do rejestru, pozostaje ona niezmieniona, aż do następnego wpisu. Kiedy sterownik zostanie przełączony ze stanu RUN do STOP, wszystkie rejestry ogólnego przeznaczenia są zerowane.



Zachowywanie danych

- Dane mogą zostać zachowane w rejestrach ogólnego przeznaczenia podczas przełączenia sterownika ze stanu RUN na STOP, jeżeli specjalny przełącznik pomocniczy M8033 jest w stanie ON.



Aktualizacja danych w rejestrach

- Zapisanie nowych wartości danych do rejestrów jest dokonywane na końcu bieżącego przebiegu programu.

4.12.2 Rejestry z podtrzymaniem/zatraskowe

Jeśli dane są zapisane do rejestrów z podtrzymaniem, pozostają niezmienione aż do następnego wpisu. Kiedy sterownik zostanie przełączony ze stanu RUN do STOP, wszystkie dane w tych rejestrach są zachowane.

Zakres podtrzymywanych urządzeń może być zmieniony poprzez ustawienie parametrów sterownika. Więcej szczegółów w odpowiednim podręczniku na temat narzędzi programujących.



Użycie FX2-40AW/AP:

- Podczas używania FX z modułem FX2-40AW albo FX2-40AP, część rejestrów zatraskowych jest automatycznie przyporządkowana do celów komunikacyjnych przez moduł FX2-40AW/AP

Komunikacja pomiędzy Master i Slave zajmuje: 100 markerów M800 do M899
10 rejestrów D490-D499

Komunikacja pomiędzy Slave i Master zajmuje: 100 markerów M900 do M999
10 rejestrów D500-D509

4.12.3 Rejestry diagnostyczne

Do sterowania lub monitorowania różnych trybów lub urządzeń wewnątrz sterownika służą specjalne rejestry. Dane zapisane w tych rejestrach są danymi domyślnymi podczas włączania zasilania PLC.

- Uwaga: podczas włączania zasilania wszystkie rejestry są najpierw zerowane i następnie wartości domyślne są automatycznie wpisywane do odpowiednich rejestrów przez program systemu. Np. dane timera watchdoga są wpisywane do D8000. Żeby zmienić ustawienie, użytkownik musi wpisać wymaganą wartość na miejsce aktualnie przechowywanej.

Dane przechowywane w specjalnym rejestrze diagnostycznym pozostają niezmienione podczas przełączenia sterownika ze stanu STOP na RUN.



Użycie rejestrów diagnostycznych:

- W żadnym przypadku nie należy używać niezidentyfikowanego rejestru. Należy używać urządzeń zgodnie z przeznaczeniem, opisanym w tym podręczniku. W rozdziale 6 zamieszczono tabele, zawierające dane i opisy dostępnych urządzeń dla każdego programowalnego sterownika.

4.12.4 Rejestry plikowe

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Rejestry plikowe są dostępne w dwóch formach:

- rejestry pamięci programu - zajmują kroki programu w blokach 500-punktowych i są dostępne w sterownikach FX_{0N}, FX, FX_{2C}, FX_{2N}
- rejestry RAM - zajmują specjalne obszary danych i są dostępne na wszystkich jednostkach FX_{2C} i FX w wersji CPU 3.07 i wyższych

Rejestry pamięci programu

Rejestry plikowe mogą być umiejscowione w obszarze pamięci programu (RAM, EEPROM lub EPROM) w jednostkach 500-punktowych. Rejestry te mogą być dostępne z urządzeń programujących. Podczas pracy sterownika dane z rejestrów plikowych mogą być wczytane do rejestrów ogólnego przeznaczenia z podtrzymaniem/zatrzaskowych przy użyciu instrukcji BMOV. Transfer w drugą stronę nie jest jednak możliwy. Rejestry plikowe są ustawione w obszarze parametrów PLC. Dla każdego przydzielonego bloku rejestrów plikowych tracone jest 500 kroków programu.

Uwaga: zakres urządzeń dla rejestrów plikowych pokrywa się z rejestrami zatrzaskowymi. Rozmieszczenie tych urządzeń jako rejestrów plikowych powoduje, że dane są przechowywane razem z programem.



Zapis do rejestrów plikowych:

- Dane rejestru plikowego sterowników FX_{0N} i FX mogą być zmienione tylko za pomocą urządzenia programującego, takiego jak podręczny programator lub komputer (osobisty) z odpowiednim oprogramowaniem. Szczegóły opisujące przeprowadzenie zmian można znaleźć w odpowiednim podręczniku.
- Dane rejestru plikowego sterowników FX_(2C) i FX_{2N} mogą być również zmienione przez program, używając instrukcji BMOV.
- Tylko rejestry plikowe w pamięci RAM lub wewnętrznej mogą być zmienione podczas stanu RUN, ale kasyety pamięci EEPROM, wewnętrznej i RAM mogą być zmienione gdy sterownik jest w stanie STOP.

Specjalne ostrzeżenie podczas użytkowania FX_{0N}:

- Żadne rejestry plikowe nie mogą być modyfikowane w stanie RUN.

Specjalne ostrzeżenie podczas użytkowania FX:

- Podczas gdy sterownik FX jest w stanie RUN, zmiana rejestrów plikowych D1000-D1119 (120 punktów) nie jest dozwolona. Próba zmiany tych urządzeń podczas stanu RUN może spowodować błąd programu, kiedy PLC jest ponownie przełączany w stan RUN.

Rejestry RAM

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Rejestry plikowe RAM zajmują 2000 punktów i mogą być aktywowane przez wystawienie specjalnego przekaźnika pomocniczego M8074 w stan ON. Te rejestry mogą być dostępne tak jak normalne rejestry plikowe programu. Rejestry RAM nie zajmują kroków programu ale zajmują obszar rejestru danych próbkowanych podczas gdy M8074 jest aktywny.



Dostępne urządzenia:

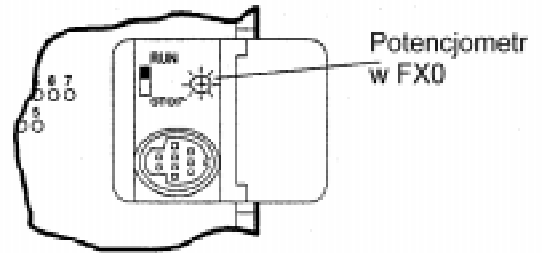
Więcej informacji na temat dostępności urządzeń można znaleźć w tabelach na stronie 4-33 lub w rozdziale 6 i 8.

4.12.5 Rejestry ustawiane zewnętrznie

Sterowniki FX₀ i FX_{0N} mają wbudowane potencjometry, używane do ustawienia wartości dedykowanych rejestrów danych. Wartość tych rejestrów może mieć zakres 0-255. To jest ich wbudowana cecha, która nie wymaga żadnego dodatkowego ustawienia ani programowania.

Sterowniki FX, FX_{2C} i FX_{2N} nie mają tej cechy, jednak jest dla nich dostępna specjalna jednostka, umożliwiającą tę samą funkcję.

Dla sterowników FX i FX_{2C} jednostką tą jest FX-8AV. Dla sterownika FX_{2N} tą jednostką jest FX_{2N}-8AV-BD. Do obsługi tej jednostki wymagane są instrukcje specjalne VRRD (FNC 85) i VRSC (FNC 86).



	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
Liczba ustawianych potencjometrów	1	2	8: dostarczane w dodatkowym bloku funkcyjnym FX-8AV lub FX _{2N} -8AV-BD		
Liczba ustawianych rejestrów	1: D8013	1: D8013 lub D8030, 2: D8031	wybierane przez użytkownika poprzez użycie instrukcji specjalnych VRRD i VRSC		



Zastosowania:

To udogodnienie jest często używane do zmiennego ustawienia timera. Może być użyte w każdej aplikacji, gdzie normalnie stosowany jest rejestr danych, np. ustawianie liczników, dostarczanie danych pierwotnych, nawet wybór operacji może być wykonywany przez tę opcję.

4.13 Rejestry indeksowe

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik urządzenia: V, Z

Zastosowanie: modyfikowanie adresu specyfikowanego urządzenia poprzez ustawienie offsetu

Oznaczenie: Rejestr (V/Z)

Indeks (rejestr/adresowanie/modyfikator)

Offset(y) (rejestr/adresowanie/modyfikator)

Modyfikator

Dostępne formy: dla danych 16-bitowych V lub Z (2 urządzenia)

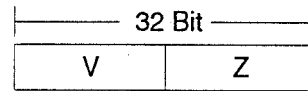
dla danych 32-bitowych połączone V i

Z (1 urządzenie, V jest specyfikowane)

operacja jest podobna jak dla rejestru danych



Numerowanie urządzeń: brak. Istnieje tylko 1 rejestr V i 1 rejestr Z



Dalsze użycie: może być używane do modyfikowania następujących urządzeń pod pewnymi warunkami: X, Y, M, S, P, T, C, D, K, H, KnX, KnY, KnM, KnS

Przykład zastosowania:

Program pokazany po prawej stronie przenosi dane z D5V do D10Z

Jeśli dana zawarta w rejestrze V równa się 8 a dana

w rejestrze Z równa się 14, wtedy:

$$V = 8$$

D5V

$$D5 + 8 = 13 \Rightarrow D13$$

$$Z = 14$$

D10Z

$$D10 + 14 = 24 \Rightarrow D24$$



Stąd, urządzeniami użytymi z wykorzystaniem modyfikatorów V i Z, są D13 i D24, a nie, odpowiednio, D5 i D10.



Użycie modyfikatorów z parametrami instrukcji specjalnych

- Wszystkie parametry instrukcji specjalnych mogą używać rejestrów indeksowych do zmodyfikowania argumentu, z wyjątkiem wyraźnie określonych przypadków.

Specjalne uwagi dla użytkowników FX₀ i FX_{0N}:

- Użytkownicy sterowników FX₀ i FX_{0N} powinni pamiętać, że kiedy jako argumenty w instrukcjach specjalnych używane są szybkie liczniki (C235-C255), nie mogą być one modyfikowane rejestrami V i Z.

4.13.1 Modyfikowanie stałej

Stałe mogą być modyfikowane tak łatwo jak rejestry danych lub urządzenia bitowe. Jeśli np. stała K20 została zapisana jako K20V, końcowy wynik będzie równy:

$K20 + \text{zawartość rejestru } V$

Przykład:

jeśli $V = 3276$ to $K20V \Rightarrow 3296$

4.13.2 Niewłaściwe użycie modyfikatorów

Modyfikowanie urządzeń Kn, kiedy Kn tworzy część opisu urządzenia, takiego jak KnY nie jest możliwe. tzn. podczas gdy dozwolone są następujące użycia modyfikatorów:

K3Z
K1M10V
Y20Z

Wyrażenia w formacie: K4ZY30 nie są dopuszczalne.



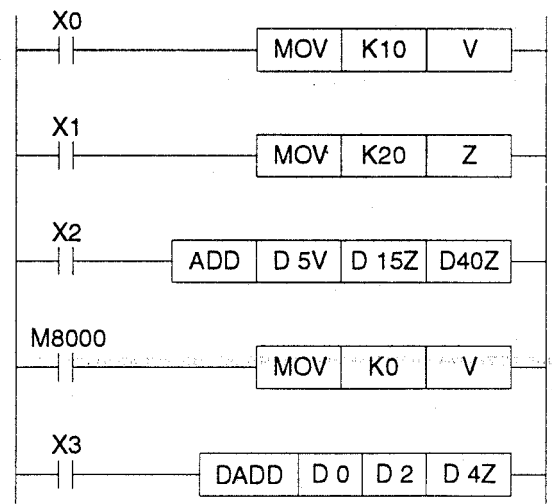
Niedopuszczalne jest używanie i modyfikacja parametrów jakiegokolwiek z 20 podstawowych instrukcji tzn. LD, AND, OR itd.

4.13.3 Użycie wielokrotne rejestrów indeksowych

Użycie wielokrotne rejestrów indeksowych jest czasem potrzebne w dużych programach lub w programach, które przetwarzają dużą ilość danych. W sterownikach programowalnych nie ma problemu z używaniem rejestrów V i Z wiele razy w programie. To czego należy się obawiać, to że czytanie takich programów przez programistę lub użytkownika jest kłopotliwe, ponieważ nie zawsze jest jasne, jaka jest bieżąca wartość V lub Z.

Przykład:

$V = 10$ (K10)
 $Z = 20$ (K20)
 $D5V = D15$ ($D5 + V = D5 + 10 = D15$)
 $D15Z = D35$ ($D15 + Z = D15 + 20 = D35$)
 $D40Z = D60$ ($D40 + Z = D40 + 20 = D60$)



Zarówno rejestr V i Z są wstępnie ustawione na, odpowiednio, K10 i K20.

Zawartość D15 jest dodana do D35 i przechowana w D60

V jest następnie zerowany i zarówno V i Z są użyte w instrukcji DADD.

Zawartość D1 i D0 są następnie dodane do D3, D2 i w końcowej fazie przechowane w D25, D24

4.14 Bity, Słowa, zapis BCD i zapis szesnastkowy

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Bieżąca sekcja omawia dokładnie organizację i sposoby prezentacji danych. Ta sekcja podzielona jest na kilka małych części, z których każda zawiera jeden lub kilka grup tematów. Niektóre z tych tematów to:

Urządzenia bitowe, pojedyncze i grupowe - zobacz strona 4-40

Urządzenie słowowe - zobacz strona 4-42

Interpretacja danej słowowej - zobacz strona 4-22

Uzupełnienie do 2 - zobacz strona 4-45



Dostępne urządzenia:

Informacje o dostępnych w PLC układach można znaleźć w rozdziale 8.

4.14.1 Urządzenia bitowe, pojedyncze i grupowe

Urządzenia, takie jak X, Y, M i S są urządzeniami bitowymi. Urządzenia bitowe są dwustanowe, to znaczy mają tylko dwa stany, ON i OFF lub 1 i 0. Urządzenia bitowe mogą być grupowane razem w większe reprezentacje danych, np. 8 kolejnych urządzeń bitowych tworzy bajt. Co więcej, 16 kolejnych urządzeń bitowych tworzy słowo, a 32 kolejne urządzenia bitowe tworzy podwójne słowo.

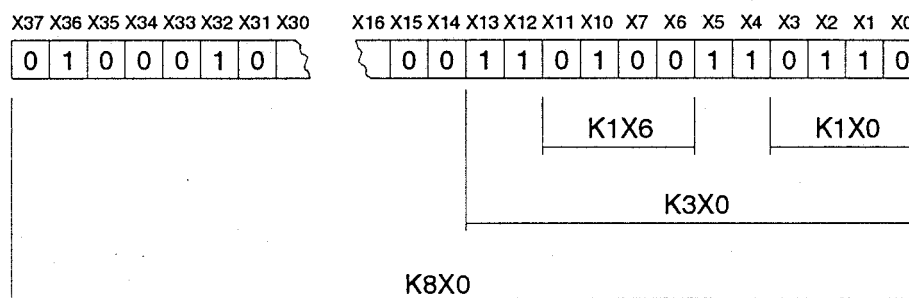
Sterownik programowalny identyfikuje grupy urządzeń bitowych, które powinny być uważane za pojedyncze jednostki, poprzez wyszukanie znacznika zakresu, poprzedzanego przez adres nagłówka. Jest to w formie Kn☆, gdzie ☆ reprezentuje adres nagłówka użytego urządzenia bitowego. Część Kn wyrażenia identyfikuje zakres zawartego urządzenia (liczbę czwórek). "n" może być liczbą z zakresu 0-8. Każda cyfra "n" reprezentuje urządzenie 4-bitowe, tzn. K1 = urządzenia 4-bitowe, a K8 = urządzenia 32-bitowe. Stąd wszystkie grupy urządzeń bitowych są podzielne przez 4.

Diagram i przykład na następnej stronie objaśnia tę ideę dalej.

Przyporządkowanie grupowanych urządzeń bitowych:

Jak już wyjaśniono, urządzenia bitowe mogą być grupowane w jednostki 4-bitowe. "n" w KnM0 definiuje liczbę grup 4 bitów, które mają być połączone dla przetworzenia danych. Dla K1-K4 dozwolone są operacje 16-bitowe, ale dla K1-K8 dostępne są operacje 32-bitowe.

K2M0, np. identyfikuje dwie grupy po 4 bity; M0-M3 i M4-M7, dając razem w sumie urządzenie 8-bitowe lub 1-bajtowe. Rysunek poniżej przedstawia więcej przykładów użycia Kn☆.



- K1X0 : X0 do X3 ⇒ urządzenie 4-bitowe z adresem nagłówka X0
- K1X6 : X6 do X11 ⇒ urządzenie 4-bitowe z adresem nagłówka X6
- K3X0 : X0 do X13 ⇒ urządzenie 12-bitowe z adresem nagłówka X0
- K8X0 : X0 do X37 ⇒ urządzenie 32-bitowe z adresem nagłówka X0

**Przesuwanie grupowanych urządzeń bitowych:**

Jeżeli przesunięcie danych wymaga zabrania danych źródłowych i przesunięcia ich w miejsce przeznaczenia, które jest mniejsze niż oryginalne źródło, wtedy nadmiar danych źródłowych jest ignorowany, np.

Jeżeli K3M20 jest przesunięte do K1M0, wtedy tylko M20-M23 jest przesunięte. Pozostałe dane K2M24 (M24-M31) są zignorowane.

**Przyporządkowanie wejścia/wyjścia**

Każda wartość wzięta z dostępnego zakresu urządzeń może być użyta jako znacznik adresu nagłówka grupy urządzeń bitowych. Jednakże, zaleca się użycie "0" na pozycji jednostek urządzeń X i Y (X0, X10, X20, itd.). Dla urządzeń M i S najbardziej efektywne jest użycie mnożnika "8". Jednak, ponieważ użycie takich liczb może prowadzić do zamieszania przy przyporządkowaniu numerów urządzeń, zaleca się używanie wielokrotności "10". To umożliwi właściwe zależności między urządzeniami X i Y.

4.14.2 Urządzenia słowowe

Urządzenia słowowe, takie jak T, C, D, V, Z mogą przechowywać dane o poszczególnym zdarzeniu lub działaniu wewnątrz sterownika. Najczęściej te urządzenia są rejestrami 16-bitowymi. Jednakże niektóre odmiany mają pojemność 32-bitową, tak jak pary kolejnych rejestrów danych lub połączone rejestry V i Z.

Może wydawać się dziwne podawanie rozmiaru urządzenia słowowego w bitach. Przestaje to dziwić, jeśli weźmie się pod uwagę, że bit jest najmniejszą jednostką danych wewnątrz sterownika. Tak więc określanie wszystkiego w formacie bitowym ułatwia porównywanie itp.

Dodatkowym udogodnieniem takiej interpretacji bitowej jest to, że aktualne dane mogą być różnie interpretowane. Fizyczny wzorec aktywnych bitów może być ważną cechą albo też numeryczna interpretacja wzorca bitowego może być kluczem do programu. Wszystko zależy od tego jak jest odczytywana informacja.

4.14.3 Interpretacja danych słowowych

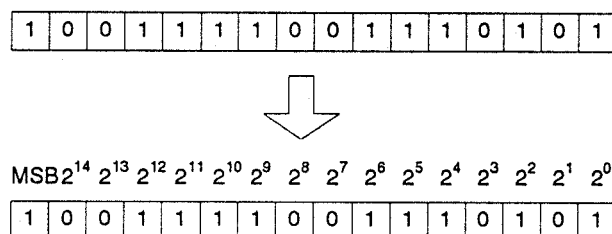
Jako że dane słowowe mogą być odczytywane na wiele sposobów, znaczenie poszczególnych części danych może się zmieniać. Sterowniki mogą odczytywać dane słowowe jako:

- czysty wzorec bitowy
- liczba binarna
- liczba szesnastkowa
- liczba BCD (liczba dziesiętna kodowana dwójkowo)

Następujące przykłady pokazują, jak ta sama dana może mieć wiele znaczeń, w zależności od sposobu, w jaki informacja jest czytana bądź interpretowana.

a) interpretacja wzorca bitowego

Następujący wzorec bitowy nic nie oznacza - jest to po prostu zbiór 16 urządzeń dwustanowych. Niektóre z tych urządzeń są losowo ustawione w jeden z dwu stanów. Jednakże, gdy dodać zapis pozycyjny o podstawie 2 do 16-bitowej danej, można obliczyć dziesiętną sumę aktywnych bitów.



$$\begin{aligned} \text{Wartość dziesiętna} &= (2^0 \times 1) + (2^2 \times 1) + (2^4 \times 1) + (2^5 \times 1) \\ &\quad + (2^5 \times 1) + (2^9 \times 1) + (2^{10} \times 1) + (2^{11} \times 1) + (2^{12} \times 1) \end{aligned}$$

$$\text{Wartość dziesiętna} = 7797$$

W rzeczywistości jest to niepoprawne!

Pominięto jedno z urządzeń bitowych. Jeśli będzie dokładnie przestudiowany jego zapis, okaże się, że jest to MSB (bit najbardziej znaczący). To pojedyncze urządzenie bitowe określi, czy dane będą interpretowane jako liczba dodatnia czy ujemna. W tym przykładzie MSB równa się 1, tzn. dana jest ujemna.

Mimo to rozwiązaniem nie jest -7797.

Wynikiem nie może być -7797, ponieważ wartość ujemna jest obliczana przy użyciu uzupełnienia do dwóch (opisanego później), ale wynik może być szybko obliczony w następujący sposób:

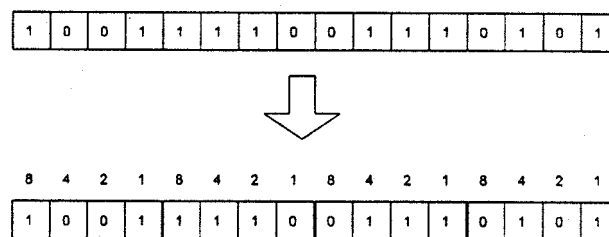
Ponieważ jest to liczba ujemna, bazą jest -32.768. Jest to liczba 16-bitowa, najmniejsza z dostępnych. Do niej zostaje dodana suma dodatnia bitów aktywnych, tzn. $-32.768 + 7797$

Poprawna odpowiedź wynosi -24.971.

Należy pamiętać, że jest to teraz przedstawienie dziesiętne oryginalnego wzorca 16-bitowego. Gdyby oryginalny wzór został ponownie przeliczony jako liczba szesnastkowa, odpowiedź byłaby inna.

b) zapis szesnastkowy

Wykorzystując taki sam oryginalny wzorzec bitów, jaki został omówiony w a) i wprowadzając zapis szesnastkowy zamiast dwójkowego (o podstawie 2), znaczenie nowego wzorca bitowego staje się:



Wartość szesnastkowa = $((1 \times 8) + (1 \times 1)), ((1 \times 8) + (1 \times 4) + (1 \times 2)),$
 $((1 \times 4) + (1 \times 2) + (1 \times 1)), ((1 \times 4) + (1 \times 1))$

Wartość szesnastkowa = 9E75

Dwie rzeczy stają się natychmiast jasne po konwersji szesnastkowej. Pierwszą jest to, że nie istnieje bit znaku, ponieważ liczby szesnastkowe są zawsze dodatnie.

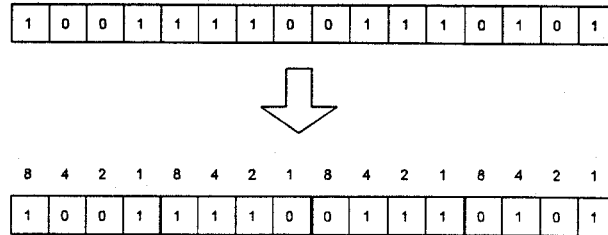
Drugą rzeczą jest to, że pojawia się cyfra "E" w obliczonej danej. Jest to dopuszczalne, ponieważ w układzie szesnastkowym liczy się od 0 do 15. Ponieważ w zapisie dziesiętnym jest tylko 10 cyfr (0-9), należy znaleźć podstawienia dla pozostałych liczb podstawy 16, tzn. 10, 11, 12, 13, 14 i 15. Użyto do tego pierwszych sześć liter alfabetu, odpowiednio od A do F.

Do reprezentacji cyfry szesnastkowej wymagane są 4 bity. Stąd 16-bitowe słowo danych będzie zawierać 4 pozycje (cyfry) szesnastkowe.

Istnieje jeszcze trzecia interpretacja wzorca bitowego - BCD (liczba dziesiętna kodowana dwójkowo). Konwersja oryginalnej postaci binarnej do formatu BCD zostanie przedstawiona w następnej sekcji.

c) konwersja BCD

Użycie oryginalnego wzorca bitowego, jako podstawy, z dodaniem nagłówka BCD umożliwia konwersję danych binarnych do formatu BCD.



wartość BCD = Błąd !!!!!

Od razu widać błąd. Konwersja nie będzie poprawna. Dzieje się tak dlatego, ponieważ liczby BCD mogą przyjmować tylko wartości od 0 do 9, a drugi od lewej blok urządzeń 4-bitowych będzie miał wartość 14, stąd błąd.

Proces konwersji jest bardzo podobny do szesnastkowego, z wyjątkiem wspomnianego zakresu wartości 0-9. Jeżeli inne bloki byłyby konwertowane tak jak w przykładzie, uzyskano by następujące wartości:

$$\text{blok pierwszy od lewej} = ((1 \times 8) + (1 \times 1)) = 9$$

$$\text{blok drugi od prawej} = ((1 \times 4) + (1 \times 2) + (1 \times 1)) = 7$$

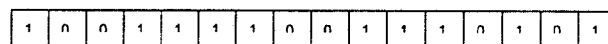
$$\text{blok pierwszy od prawej} = ((1 \times 4) + (1 \times 1)) = 5$$

Dane BCD są czytane od lewej do prawej, tak jak byłyby czytane normalne liczby. Dlatego w tym przykładzie "9" odpowiadałoby właściwie "9000". Blok drugi od prawej ma wartość "70" a nie "7". Jednostki liczby są umieszczone w bloku pierwszym od prawej, tzn. "5". Setki byłyby umieszczone w bloku drugim od lewej (w którym jest błąd).

Ważne jest również to, że nie ma znaku w konwertowanej liczbie BCD. Maksymalna liczba dopuszczalna BCD dla danej słowowej wynosi "9999", a najmniejsza "0000".

Podsumowanie danych słowowych

W każdym z poprzednio omówionych przypadków oryginalny wzór bitowy miał dalsze znaczenie. Podsumowanie trzech nowych interpretacji i oryginalnego wzorca bitowego



binarnie	:	-24971
szesnastkowo	:	9E75
BCD	:	Błąd (9?75)

Każde znaczenie jest zupełnie różne od następnego, istnieją różne sposoby opisanego znaczenia. W gruncie rzeczy one są sobie równe!

4.14.4 Uzupelnienie do 2

Sterowniki programowalne, komputery itd, używają formatu, zwanego uzupełnieniem do 2. Jest to wygodna postać prezentacji liczb, do wykonywania operacji arytmetycznych w systemach mikroprocesorowych. Jest wykorzystana do reprezentowania liczb ujemnych i do wykonywania operacji odejmowania. Procedura jest bardzo prosta, w następującym przykładzie ma być rozwiązane działanie "15 - 7"

Krok 1: Znaleźć wartości dwójkowe (w tym przykładzie używa się 8 bitów)

$$\begin{array}{rcl} 15 & = & 00001111 \\ 7 & = & 00000111 \end{array}$$

Krok 2: Znaleźć negację odjemnika

Procedura: zamienić wszystkie 1 na 0 i 0 na 1.

$$\begin{array}{rcl} 7 & = & 00000111 \\ \text{negacja } 7 & = & 11111000 \end{array}$$

Krok 3: dodać 1 do zanegowanego odjemnika

Procedura: dodać 1 do najmniej znaczącego bitu. Należy pamiętać, że jest to dodawanie binarne, stąd kiedy otrzyma się wartość 2, 1 jest przesuwana do następnej lewej pozycji, a bieżąca pozycja jest zerowana.

$$\begin{array}{rcl} \text{negacja } 7 & & 11111000 \\ \text{dodawane } 1 & & 00000001 \\ \hline \text{wynik} & & 11111001 \end{array}$$

Wynik jest taki sam jak wartość ujemna 7, tzn. -7

Krok 4: dodać wynik do odjemnej (15)

Procedura: należy pamiętać, że w układzie dwójkowym $1 + 1 = 0$ przeniesienie 1

$$\begin{array}{rcl} 15 & = & 00001111 \\ \text{wynik z kroku 3} & = & 11111001 \\ \hline \text{rozwiązanie} & = & (1)00001000 \end{array}$$

(1) jest przeniesieniem i jest pominięte, ponieważ ten przykład ograniczony jest do 8 bitów.

Krok 5: określić wartość dziesiętną:

$$00001000 = 8$$

Wartość jest dodatnia, ponieważ MSB (pierwszy bit z lewej) ma wartość 0. Po dokonaniu szybkiego sprawdzenia w pamięci uzyska się istotnie wynik "15 - 7 = 8"

Można zauważyć, że nie zostało wykonane żadne odejmowanie. Każdy z kroków albo konwertował albo dodawał dane. Pomimo tego odpowiedź jest poprawna: $15 - 7 = 8$. Przykład ten był oparty na liczbach 8-bitowych, ale będzie działał tak samo na każdej innej liczbie bitów.

4.15 Zapis zmiennoprzecinkowy i szesnastkowy

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Sterowniki programowalne mogą korzystać z wielu systemów i metod do przechowania danych. Najbardziej rozpowszechnione zostały już omówione w poprzednich sekcjach, np. BCD, dwójkowy, dziesiętny, szesnastkowy. Są one znane jako formaty całkowite. Jak wynika to z nazwy, te formaty używają tylko liczb całkowitych, które nie posiadają części ułamkowych. Jednakże istnieją jeszcze dwa inne formaty, które stają się bardzo ważne:

- a) zapis zmiennoprzecinkowy
- b) zapis naukowy

Oba te formaty są ściśle połączone. Są wykorzystywane do tworzenia bardzo dużych lub bardzo małych liczb, które mogą opisywać składniki całkowite i ułamkowe.



Uwaga ogólna:

Czasami terminy: "format", "tryb", "zapis" są używane wymiennie przy opisach procesów numerycznych. Jednakże wszystkie z tych terminów mają taki sam sens opisowy i użytkownicy powinni zdawać sobie sprawę z ich obecności.

Niektóre przydatne stałe:

π	$3,141 \times 10^0$
2π	$6,283 \times 10^0$
$\pi/4$	$7,853 \times 10^{-1}$
π^2	$9,869 \times 10^0$
prędkość światła	$2,997 \times 10^8$ m/s
stała grawitacji	$9,807 \times 10^0$ m/s ²
e (stała Eulera)	$2,718 \times 10^0$

Stale fizyczne:

punkt wrzenia ciekłego tlenu	$-1,8297 \times 10^2$ °C
punkt zamarzania wody	$0,00 \times 10^0$ °C
punkt potrójny wody	$1,00 \times 10^{-2}$ °C
punkt wrzenia wody	$1,00 \times 10^2$ °C

4.15.1 Zapis naukowy

Ten format może być nazwany krokiem pomiędzy formatami całkowitymi a pełnymi formatami zmiennoprzecinkowymi. W podstawowych warunkach notacja naukowa używa dwóch urządzeń do przechowania informacji o liczbie lub wartości. Jedno urządzenie zawiera ciąg znaków liczby (mantysa) podczas gdy drugie urządzenie zawiera informację o liczbie miejsc dziesiętnych w liczbie (wykładnik). Stąd notacja naukowa może pomieścić wartości większe/mniejsze niż normalny limit 32-bitowy, tzn. $-2.147.483.648$ do $2.147.483.647$, podczas gdy granicami notacji naukowej są:

Maksymalne
 9999×10^{35}
 -9999×10^{35}

Minimalne
 9999×10^{-41}
 -9999×10^{-41}

Zapis naukowy może być otrzymany przez użycie instrukcji BCD (FNC 18) ze znacznikiem M8023 ustawionym w stan ON. W tej sytuacji liczby zmiennoprzecinkowe są konwertowane przez instrukcję BCD do notacji naukowej - szczegóły na stronie 5-22.

Notacja naukowa może być z powrotem konwertowana do formatu zmiennoprzecinkowego przy użyciu instrukcji BIN (FNC 19) ze znacznikiem M8023 ustawionym w stan ON. Szczegóły na stronie 5-22.

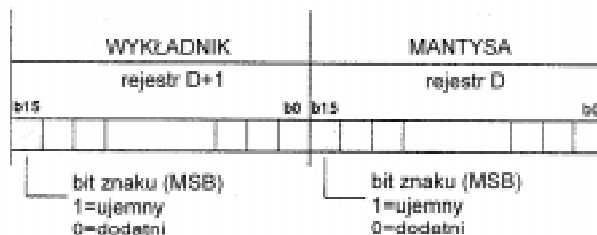
Używając notacji naukowej w odpowiednich jednostkach sterowników FX należy pamiętać, że:

- Mantysa i wykładnik są przechowywane w kolejnych rejestrach danych. Każda część jest utworzona z 16 bitów i jest jej przypisana wartość dodatnia lub ujemna, zgodnie z wartością MSB (bit 15 rejestru) dla każdej liczby.

Mantysa jest przechowywana jako pierwsze 4 najbardziej znaczące cyfry, bez żadnych

zaokrążeń, tzn. liczba zmiennoprzecinkowa o wartości $2,34567 \times 10^3$ będzie przechowana jako mantysa 2345 w rejestrze danych D i jako wykładnik 0 w rejestrze danych D+1

- Zakres dostępnych wartości mantysy wynosi od 0,1000 do 9999 i od -1000 do -9999
- Zakres dostępnych wartości wykładnika wynosi od 35 do -45
- Format naukowy nie może być używany bezpośrednio w obliczeniach, ale zapewnia idealny sposób prezentacji danych na urządzeniach monitorujących.



4.15.2 Format zmiennoprzecinkowy

Format zmiennoprzecinkowy rozszerza możliwości i zakresy notacji naukowej z możliwością reprezentacji części ułamkowych całych liczb, np.

Wykonanie i wyświetlenie obliczenia dzielenia 22 przez 7 będzie dawało następujące wyniki:

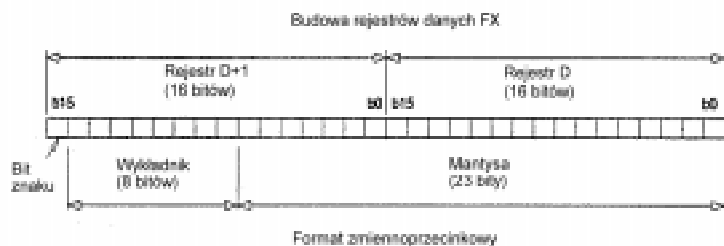
- Normalna operacja FX, używająca dziesiętnych liczb całkowitych da wynik 3 z resztą 1
- W zmiennym przecinku uzyska się wynik 3,14285 (w przybliżeniu)
- W notacji naukowej to obliczenie wyniesie 3142×10^{-3}

Widać, że większą dokładność uzyskuje się stosując liczby zmiennoprzecinkowe, tzn. poprzez użycie większych zakresów liczbowych i możliwości operowania na większej ilości cyfr. Stąd, obliczenia wykorzystujące dane zmiennoprzecinkowe są dogodniejsze.

Dana dziesiętna może być konwertowana do formatu zmiennoprzecinkowego przy użyciu instrukcji FLT (FNC 49). Kiedy ta sama instrukcja jest użyta ze znacznikiem M8023, ustawionym w stan ON, liczby zmiennoprzecinkowe mogą być z powrotem konwertowane do dziesiętnych. Więcej szczegółów na stronie 5-50.

Używając notacji zmiennoprzecinkowej w odpowiednich jednostkach sterowników FX należy pamiętać, że:

- liczby zmiennoprzecinkowe, bez względu na wartość, będą zajmowały 2 kolejne rejestry danych (32 bity)
- wartości zmiennoprzecinkowe nie są bezpośrednio odczytywalne, ponieważ są przechowywane w specjalnym formacie, rekomendowanym przez I.E.E.E. dla aplikacji mikrokomputerów i komputerów osobistych.
- liczby zmiennoprzecinkowe mają zarówno mantysę jak i wykładnik (wyjaśnienie tych terminów w sekcji o notacji naukowej). Do zapisu wykładnika używa się tylko 8 bitów, dodatkowo wykorzystuje się pojedynczy bit znaku mantysy. Pozostałe 23 bity z 32 są używane do opisanie wartości mantysy.

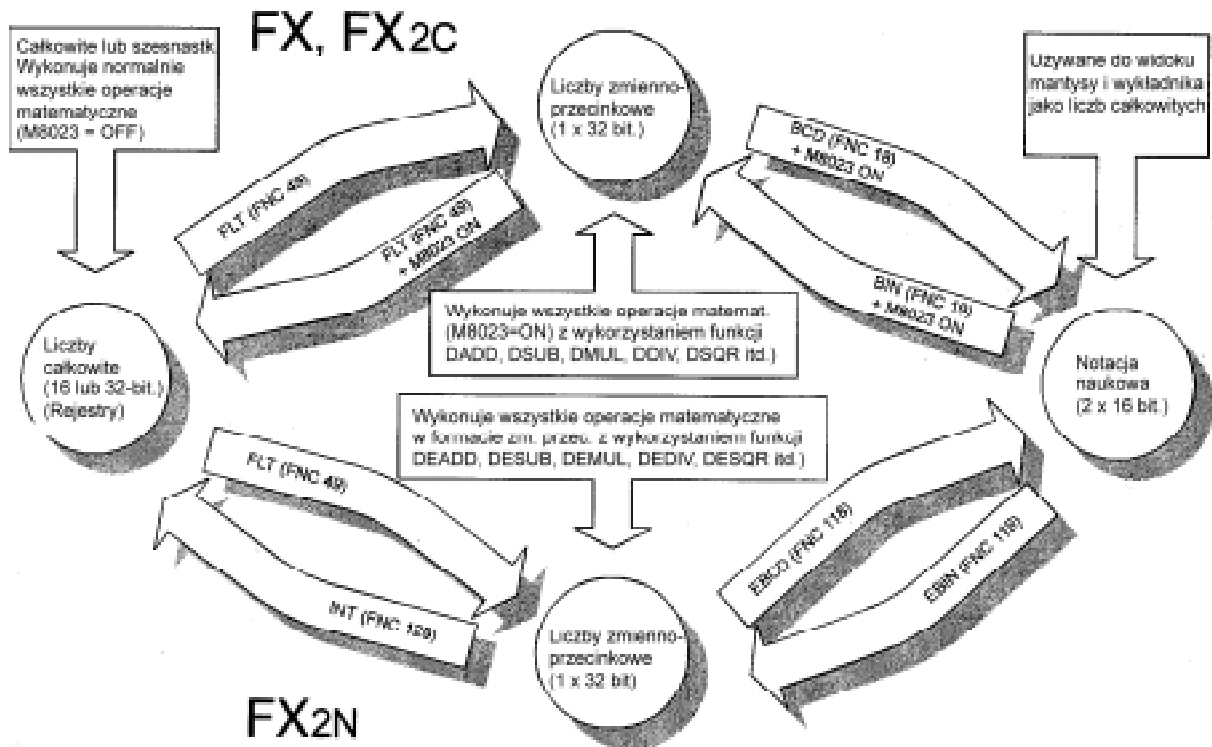


Zakresy liczb zmiennoprzecinkowych używanych w jednostkach FX

Opis	Znak	Wykładnik (wzorzec bitowy)	Mantysa (wzorzec bitowy)	Komentarz
Normalna liczba	0 lub 1	11111110 00000001	111111111111111111111111 111111111111111111111110 000000000000000000000001 000000000000000000000000	Największa liczba: $\pm 3,403 \times 10^{38}$ Dokładność: 7 cyfr znaczących Najmniejsza liczba: $\pm 1,175 \times 10^{-38}$
Zero	0 lub 1	00000000	000000000000000000000000	Wszystkie cyfry są zerami

4.15.3 Zestawienie notacji naukowej i liczb zmiennoprzecinkowych

Instrukcje potrzebne do konwersji pomiędzy każdym formatem liczbowym pokazane są poniżej w formie diagramu.



Notatki

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

5. Instrukcje specjalne				5-199	
5.1 Sterowanie przebiegiem programu - funkcje 00-09				5-4	
5.1.1 CJ (FNC 00)	5-5	5.1.2 CALL (FNC 01)	5-7	5.1.3 SRET (FNC 02)	5-8
5.1.4 IRET,EI,DI (FNC 03, 04, 05)	5-9	5.1.5 FEND (FNC 06)	5-11	5.1.6 WDT (FNC 07)	5-12
5.1.7 FOR, NEXT (FNC 08, 09)	5-13				
5.2 Instrukcje przesłania i porównania - funkcje 10-19				5-16	
5.2.1 CMP (FNC 10)	5-17	5.2.2 ZCP (FNC 11)	5-17	5.2.3 MOV (FNC 12)	5-18
5.2.4 SMOV (FNC 13)	1-18	5.2.5 CML (FNC 14)	5-19	5.2.6 BMOV (FNC 15)	5-20
5.2.7 FMOV (FNC 16)	5-21	5.2.8 XCH (FNC 17)	5-21	5.2.9 BCD (FNC 18)	5-22
5.2.10 BIN (FNC 19)	5-22				
5.3 Operacje arytmetyczne i logiczne - funkcje 30-39				5-24	
5.3.1 ADD (FNC 20)	5-25	5.3.2 SUB (FNC 21)	5-26	5.3.3 MUL (FNC 22)	5-27
5.3.4 DIV (FNC 23)	5-28	5.3.5 INC (FNC 24)	5-29	5.3.6 DEC (FNC 25)	5-29
5.3.7 WAND (FNC 26)	5-30	5.3.8 WOR (FNC 27)	5-30	5.3.9 WXOR (FNC 28)	5-31
5.3.10 NEG (FNC 29)	5-31				
5.4 Rotacje i przesunięcia - funkcje 30-39				5-34	
5.4.1 ROR (FNC 30)	5-35	5.4.2 ROL (FNC 31)	5-35	5.4.3 RCR (FNC 32)	5-36
5.4.4 RCL (FNC 33)	5-36	5.4.5 SFTR (FNC 34)	5-37	5.4.6 SFTL (FNC 35)	5-37
5.4.7 WSFR (FNC 36)	5-38	5.4.8 WSFL (FNC 37)	5-38	5.4.9 SFWR (FNC 38)	5-39
5.4.10 SFRD (FNC 39)	5-40				
5.5 Operacje na danych - funkcje 40-49				5-42	
5.5.1 ZRST (FNC 40)	5-43	5.5.2 DECO (FNC 41)	5-43	5.5.3 ENCO (FNC 42)	5-44
5.5.4 SUM (FNC 43)	5-45	5.5.5 BON (FNC 44)	5-45	5.5.6 MEAN (FNC 45)	5-46
5.5.7 ANS (FNC 46)	5-47	5.5.8 ANR (FNC 47)	5-47	5.5.9 SQR (FNC 48)	5-48
5.5.10 FLT (FNC 49)	5-49				
5.6 Szybkie przetwarzanie - funkcje 50-59				5-52	
5.6.1 REF (FNC 50)	5-53	5.6.2 REFF (FNC 51)	5-53	5.6.3 MTR (FNC 52)	5-54
5.6.4 HSCS (FNC 53)	5-55	5.6.5 HSCR (FNC 54)	5-56	5.6.6 HSZ (FNC 55)	5-57
5.6.7 ALT (FNC 56)	5-60	5.6.8 PLSY (FNC 57)	5-61	5.6.9 PWM (FNC 58)	5-62
5.6.10 PLSR (FNC 59)	5-63				
5.7 Instrukcje podręczne - funkcje 60-69				5-66	
5.7.1 IST (FNC 60)	5-67	5.7.2 SER (FNC 61)	5-69	5.7.3 ABSD (FNC 62)	5-70
5.7.4 INCD (FNC 63)	5-71	5.7.5 TTMR (FNC 64)	5-72	5.7.6 STMR (FNC 65)	5-72
5.7.7 ALT (FNC 66)	5-73	5.7.8 RAMP (FNC 67)	5-73	5.7.9 ROTC (FNC 68)	5-75
5.7.10 SORT (FNC 69)	5-77				
5.8 Zewnętrzne urządzenia wejścia/wyjścia - funkcje 70-79				5-80	
5.8.1 TKY (FNC 70)	5-81	5.8.2 HKY (FNC 71)	5-82	5.8.3 DSW (FNC 72)	5-83
5.8.4 SEGD (FNC 73)	5-84	5.8.5 SEGL (FNC 74)	5-85	5.8.6 ARWS (FNC 75)	5-87
5.8.7 ASC (FNC 76)	5-88	5.8.8 PR (FNC 77)	5-89	5.8.9 FROM (FNC 78)	5-90
5.8.10 TO (FNC 79)	5-91				
5.9 Zewnętrzne urządzenia szeregowo - funkcje 80-89				5-94	
5.9.1 RS (FNC 80)	5-95	5.9.2 PRUN (FNC 81)	5-99	5.9.3 ASCI (FNC 82)	5-101
5.9.4 HEX (FNC 83)	5-102	5.9.5 CCD 9FNC 84)	5-103	5.9.6 VRRD (FNC 85)	5-104
5.9.7 VRSC (FNC 86)	5-104	5.9.8 PID (FNC 88)	5-105		
5.10 Zewnętrzne jednostki F2 - funkcje 90-99				5-112	
5.10.1 MNET (FNC 90)	5-113	5.10.2 ANRD (FNC 91)	5-113	5.10.3 ANWR (FNC 92)	5-114
5.10.4 RMST (FNC 93)	5-114	5.10.5 RMWR (FNC 94)	5-115	5.10.6 RMRD (FNC 95)	5-116
5.10.7 RMMN (FNC 96)	5-116	5.10.8 BLK (FNC 97)	5-117	5.10.9 MCDE (FNC 98)	5-118
5.11 Operacje zmiennoprzecinkowe 1 & 2 - funkcje 110-129				5-120	
5.11.1 ECMP (FNC 110)	5-121	5.11.2 EZCP (FNC 111)	5-121	5.11.3 EBDC (FNC 118)	5-122
5.11.4 EBIN (FNC 119)	5-122	5.11.5 EADD (FNC 120)	5-123	5.11.6 ESUB (FNC 121)	5-124
5.11.7 EMUL (FNC 122)	5-124	5.11.8 EDIV (FNC 123)	5-125	5.11.9 ESQR (FNC 127)	5-125
5.11.10 INT (FNC 129)	5-126				
5.12 Operacje trygonometryczne - funkcje 130-139				5-128	
5.12.1 SIN (FNC 130)	5-129	5.12.2 COS (FNC 131)	5-130	5.12.3 TAN (FNC 132)	5-130
5.13 Operacje na danych 2 - funkcje 140-149				5-132	
5.13.1 SWAP (FNC 147)	5-133				
5.14 Sterowanie zegarem czasu rzeczywistego - funkcje 160-169				5-136	
5.14.1 TCMP (FNC 160)	5-137	5.14.2 TZCP (FNC 161)	5-138	5.14.3 TADD (FNC 162)	5-139
5.14.4 TSUB (FNC 163)	5-140	5.14.5 TRD (FNC 166)	5-141	5.14.6 TWR (FNC 167)	5-142
5.15 Kody Graya - funkcje 170-179				5-144	
5.15.1 GRY (FNC 170)	5-145	5.15.2 GBIN (FNC 171)	5-145		
5.16 Porównania operacyjne - funkcje 220-249				5-148	
5.16.1 LD compare (FNC 224-230)	5-149	5.16.2 AND compare (FNC 232-238)	5-150	5.16.3 OR compare (FNC 240-246)	5-151

5. Instrukcje specjalne

FX_{0(S)}FX_{0N}

FX

FX_(2C)FX_{2N}

Instrukcje specjalne są instrukcjami specjalistycznymi rodziny sterowników programowalnych FX. Pozwalają one użytkownikowi wykonywać złożone manipulacje danymi, obliczenia matematyczne, przy zachowaniu możliwości programowania i monitorowania programu. Każda instrukcja specjalna ma własny unikalny mnemonik i numer, i będzie wyrażona przy użyciu tabeli podobnej do przedstawionej poniżej.

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
CJ FNC 00 (skok warunkowy)	metoda skoku do wybranej etykiety	prawidłowe etykiety z zakresy 0 do 63	CJ, CJP : 3 kroki etykieta skoku P☆☆: 1 krok

Tabela ta będzie się znajdowała na początku opisu każdej nowej instrukcji. Kolumna nazwana "Argumenty" będzie zawierać różne urządzenia (będące argumentami), które mogą być użyte w instrukcji. Różne litery identyfikacyjne będą użyte do scharakteryzowania każdego argumentu, np. D - przeznaczenie, S - źródło, n, m - liczba elementów. Dodatkowe przyrostki numeryczne zostaną dołączone jeżeli instrukcja zawiera więcej niż jeden argument.

Nie wszystkie instrukcje i warunki odnoszą się do każdego typu sterownika. Odpowiednie CPU są wskazane w prostokątnych polach w górnym prawym rogu strony. Dla pokazania bardziej szczegółowych odmian instrukcji użyto następnego prostokątnego pola z zaznaczeniem: dostępności instrukcji impulsowej, pojedynczego (16-bitowego) słowa, podwójnego (32-bitowego) słowa oraz znaczników, ustawianych przez instrukcje.

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

Operacje 16-bitowe nie wymagają modyfikacji mnemoniki instrukcji. Jednak operacja impulsowa wymaga dodania 'P' bezpośrednio za mnemonikiem, podczas gdy 32-bitowa operacja wymaga dodania 'D' przed mnemonikiem. Tzn. jeżeli byłaby użyta instrukcja zarówno impulsowa jak i 32-bitowa, wyglądałaby następująco:

D☆☆☆P, gdzie ☆☆☆ byłyby podstawowym mnemonikiem.

Funkcja impulsowa umożliwi aktywowanie się skojarzonej z nią instrukcji przy narastającym zboczu sygnału na wejściu sterującym. Instrukcja jest w stanie ON podczas jednego przebiegu programu. Następnie, podczas gdy wejście sterujące pozostaje w stanie ON, połączona z nim instrukcja nie jest aktywna. Aby ponownie wykonać instrukcję, wejście sterujące musi być ponownie przełączone ze stanu OFF na ON.

Kolumna ZNACZNIKI zawiera znaczniki, używane w instrukcji. Szczegóły, dotyczące funkcji znaczników są wyjaśnione w tekście, dotyczącym instrukcji.



Dla instrukcji pracujących w sposób ciągły, tzn. w każdym przebiegu programu instrukcja jest wykonywana i dostarcza nowych, różnych wyników, będzie używany następujący symbol '→' do reprezentacji szybkich zmian stanu. Instrukcje oznaczone w ten sposób zawierają inkrementowany, bądź indeksowany argument.

W większości przypadków argumenty instrukcji specjalnych mogą być indeksowane programowo. Dla tych argumentów, które **nie mogą** być indeksowane, używany jest symbol '☒', do oznaczenia argumentu, który pozostaje "stały" po jego zapisaniu.



Niektóre instrukcje wykorzystują dodatkowe rejestry danych i/lub znaczniki stanu np. funkcja matematyczna, taka jak ADD (FNC 20) może sygnalizuje wynik zerowy, warunki pożyczki lub przeniesienia poprzez użycie wstępnie ustawionych pomocniczych znaczników M8020-M8021.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------



1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.1 Sterowanie przebiegiem programu - funkcje 00 – 09

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Zawartość:

			Strona
CJ	skok warunkowy	FNC 00	5-5
CALL	wywołanie podprogramu	FNC 01	5-7
SRET	powrót z podprogramu	FNC 02	5-8
IRET	powrót z obsługi przerwania	FNC 03	5-9
EI	odblokowanie przerw	FNC 04	5-9
DI	zablokowanie przerw	FNC 05	5-9
FEND	pierwszy koniec	FNC 06	5-11
WDT	timer watchdog	FNC 07	5-12
FOR	początek pętli FOR/NEXT	FNC 08	5-13
NEXT	koniec pętli FOR/NEXT	FNC 09	5-13



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D₁, S₃ lub dla urządzeń stabilizowanych D₃₊₀, S₊₉ itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową.

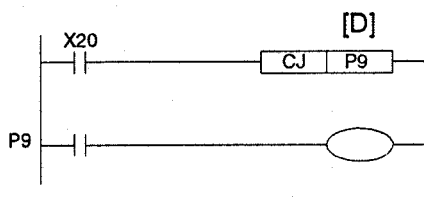
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.1.1 CJ (FNC 00)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
CJ FNC 00 (skok warunkowy)	metoda skoku do wybranej etykiety	prawidłowe etykiety z zakresy 0 do 63	CJ, CJP : 3 kroki etykieta skoku P☆☆: 1 krok

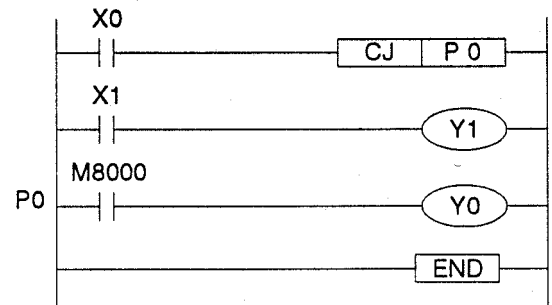
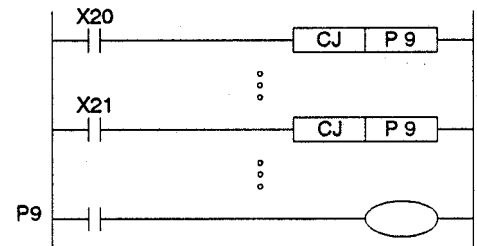
Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

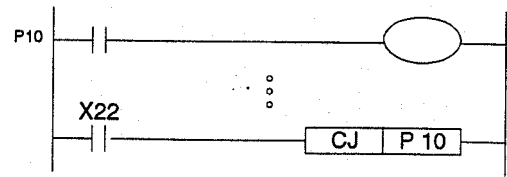
Kiedy instrukcja CJ jest aktywna, następuje skok programu do oznaczonego etykietą miejsca programu. Podczas skoku omijane są kroki programu między początkiem skoku a jego końcem. To znaczy, że nie są one w żaden sposób przetwarzane. Efektem końcowym jest przyspieszenie czasu przebiegu programu.

Uwagi:

- Do danej etykiety skoku może się odwoływać wiele instrukcji CJ.
- Każda etykieta musi mieć swój unikatowy numer. Użycie etykiety P63 jest równoważne ze skokiem do instrukcji END.
- Żaden przeskoczony obszar programu nie będzie aktualizował statusów wyjścia, nawet jeżeli zostanie zmieniony stan urządzenia wejścia. Przykład obok pokazuje sytuację, w której wejście X1 steruje wyjściem Y1. Zakładając, że X1 jest w stanie ON, a instrukcja CJ jest aktywowana, odczytanie stanu X1 i sterowanie wyjściem Y1 jest pominięte. Teraz nawet jeśli X1 jest ustawiony w stan OFF, Y1 pozostanie w stanie ON, podczas gdy instrukcja CJ powoduje skok programu do etykiety P0. Możliwa jest również sytuacja odwrotna, tzn. jeżeli na początku X1 jest w stanie OFF i wykonywana jest instrukcja CJ, Y1 nie przejdzie w stan ON, jeżeli X1 jest ustawione w stan ON. W momencie, gdy instrukcja CJ jest nieaktywna, X1 będzie sterować wyjściem Y1 w normalny sposób. Ta sytuacja odnosi się do wszystkich typów wyjść, np. SET, RST, OUT, Y, M i S.
- Instrukcja CJ może wykonać skok do każdego punktu w programie lub poza instrukcją FEND.



e) Instrukcja CJ może być użyta do skoku w przód, tzn. w kierunku instrukcji END lub do skoku w tył, w kierunku kroku 0. Wykonując skok w tył, należy uważać, by nie nastąpiło przepełnienie timera watchdoga, w przeciwnym razie sterownik zasygnalizuje błąd programu. Więcej informacji na temat timera watchdoga na stronie 5-12.



f) Skoki bezwarunkowe mogą być wprowadzone przez użycie specjalnych znaczników pomocniczych, np. M8000. W tej sytuacji sterownik w stanie RUN będzie zawsze wykonywał instrukcję CJ w sposób bezwarunkowy.

Ważne:



Timery i liczniki zachowują swoją bieżącą wartość, jeżeli znajdują się w przeskoczonym fragmencie programu. Np. jeżeli Y1 w poprzednim przykładzie (zobacz punkt c) byłby zamieniony na T0 K100 i wykonywana byłaby instrukcja CJ, zawartość T0 nie zmieniłaby się/nie wzrosłaby dopóki instrukcja CJ byłaby wykonywana. Tzn. bieżąca wartość timera zostałaby zachowana.

Szybkie liczniki są jedynym wyjątkiem w tej sytuacji ponieważ są sterowane niezależnie od głównego programu.



Użycie instrukcji specjalnych

Instrukcje specjalne są również przeskakiwane, jeśli znajdują się pomiędzy instrukcją CJ a etykietą skoku. Instrukcje PLSY (FNC 57) i PWM (FNC 58) będą działać w sposób ciągły, jeśli były aktywne zanim wykonana była instrukcja CJ. W przeciwnym razie będą nie aktywowane i przeskoczone, tak jak normalne instrukcje.



Szczegóły użycia CJ z innymi instrukcjami przebiegu programu

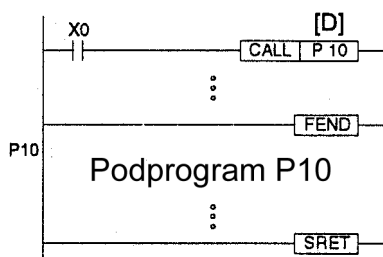
Dalsze szczegóły można znaleźć na stronie 7-12 i na stronie 7-1, gdzie opisano szczegóły dotyczące połączonego użycia różnych technik przebiegu programu (takich jak master control, MC, itd).

5.1.2 CALL (FNC 01)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D		
CALL FNC 01 (wywołanie podprogramu)	wykonuje podprogram od pozycji wskazanej etykiety	prawidłowe etykiety z zakresy 0 do 63	poziomy zagnieżdżeń: 5, włączając w to CALL początkowy	CALL, CALLP : 3 kroki etykieta podprogramu P☆☆: 1 krok

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

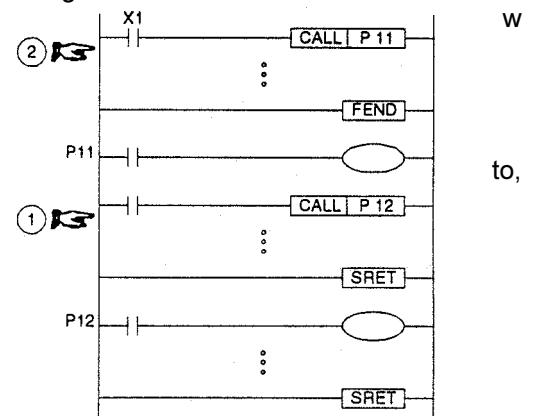
Kiedy instrukcja CALL jest aktywna, wywołuje ona podprogram od etykiety definiującej jego początek (obszar podprogramu oznaczony jako P10). Instrukcja CALL musi być używana w połączeniu z instrukcjami FEND (FNC 06) i SRET (FNC 02). Wywołanie podprogramu powoduje, że program przeskakuje z miejsca wywołania do etykiety początku podprogramu (podprogram umieszczony jest za instrukcją FEND) i wykonuje jego instrukcje aż do momentu napotkania instrukcji SRET. W miejscu tym następuje powrót do linii w schemacie drabinkowym bezpośrednio po wywołującej instrukcji CALL.

jego instrukcje aż do momentu napotkania instrukcji SRET. W miejscu tym następuje powrót do linii w schemacie drabinkowym bezpośrednio po wywołującej instrukcji CALL.

Uwagi:

- Do jednego podprogramu może się odnosić kilka instrukcji CALL.
- Każdy podprogram musi mieć unikatowy numer etykiety, które mogą być wybierane z zakresu P0-P62. Etykiety podprogramu i etykiety używane w instrukcjach CJ (FNC 00) nie mogą się pokrywać.
- Podprogramy nie są przetwarzane w kolejności występowania po instrukcji FEND. Podczas wywoływania podprogramów należy uważać, by nie nastąpiło przepełnienie timera watchdoga, w przeciwnym razie sterownik zasygnalizuje błąd programu. Więcej informacji na temat timera watchdoga na stronie 5-12.
- Podprogramy mogą być zagnieżdżane do 5 poziomów, włączając to początkową instrukcję CALL. W przykładzie obok pokazano zagnieżdżenie 2-poziomowe. Kiedy X1 jest aktywowane, wywołany jest podprogram P11. Wewnątrz podprogramu występuje instrukcja CALL do drugiego podprogramu P12. Jeżeli oba podprogramy, P11 i P12, są jednocześnie aktywne, oznacza że są one zagnieżdżone.

Kiedy podprogram P12 osiąga swoją instrukcję SRET, następuje powrót sterowania do kroku programu bezpośrednio następującego po wywołującej go instrukcji CALL (zobacz ①). P11 kończy swoje działanie i po wykonaniu swojej instrukcji SRET program wraca ponownie do kroku, który następuje po instrukcji CALL P11 (zobacz ②).



**Specjalne timery podprogramów**

Jeżeli funkcje czasowe, użyte w podprogramach, wymagają wykorzystania timerów, muszą być one wybierane z zakresu T129-T199 i T246-T249.

**Szczegóły użycia CALL z innymi instrukcjami przebiegu programu.**

Dalsze szczegóły można znaleźć na stronie 7-12 i na stronie 7-13, gdzie podano szczegóły dotyczące połączonego użycia różnych technik przebiegu programu (takich jak master control, MC, itd).

5.1.3 SRET (FNC 02)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty	Kroki programu
		D	
SRET FNC 02 (powrót z podprogramu)	zwraca sterowanie z podprogramu	brak automatycznie powraca do kroku bezpośrednio za instrukcją CALL, która wywołała podprogram	SRET: 1 krok

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

Działanie:

Instrukcja SRET wyznacza koniec bieżącego podprogramu i zwraca sterowanie programu do kroku bezpośrednio występującego po instrukcji CALL, która uaktywniła ten podprogram.

Uwagi:

- Instrukcja SRET może być używana tylko z instrukcją CALL.
- Instrukcja SRET jest zawsze programowana po instrukcji FEND - więcej szczegółów w instrukcji CALL (FNC 01)

5.1.4 IRET, EI, DI (FNC 03, 04, 05)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	
		D	Kroki programu
IRET FNC 03 (powrót z przerwania)	wymusza powrót z programu obsługi aktywnego przerwania	brak automatycznie powraca do kroku głównego programu, który był przetwarzany w momencie wystąpienia przerwania	IRET: 1 krok
EI FNC 04 (odblokowanie przerwania)	umożliwia obsługę przerwania	brak każdy sygnał przerwania, który wystąpi po instrukcji EI i przed instrukcją FEND lub DI jest natychmiast przetwarzany, chyba że został specjalnie zablokowany	EI: 1 krok
DI FNC 05 (zablokowanie przerwania)	uniemożliwia obsługę przerwania	brak każdy sygnał przerwania, który wystąpi po instrukcji DI i przed instrukcją EI zostanie zapamiętany aż do następnego wystąpienia instrukcji EI	DI: 1 krok
I (etykieta przerwania)	identyfikuje początek programu obsługi przerwania	3-cyfrowy kod numeryczny, związany z typem przerwania i operacją	I☆☆☆: 1 krok

Ogólny opis procedury przerwania:

Procedura przerwania jest częścią programu, która jest natychmiast wykonywana po wywołaniu, które przerywa główny przebieg programu. Po zakończeniu obsługi przerwania, główny przebieg programu jest wznowiany od miejsca, w którym został przerwany.

Działanie:

Przerwania są wyzwalane przez różne warunki wejściowe, czasami używane jest bezpośrednio wejście X0, czasami interwał czasowy np. 30 ms. Dostępność różnych typów przerwania oraz liczba punktów operacyjnych dla każdego typu sterownika są szczegółowo omówione na stronie 4-12 - Etykiety przerwania.

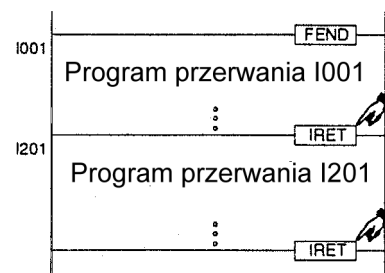
Zaprogramowanie i obsługa przerwania wymaga użycia co najmniej 3 dedykowanych instrukcji (wyszczególnionych w tej sekcji) i specjalnych etykiet przerwania.

Definiowanie procedury przerwania:

Procedura przerwania jest specyfikowana pomiędzy swoją własną, unikatową etykietą przerwania i pierwszym pojawieniem się instrukcji IRET.

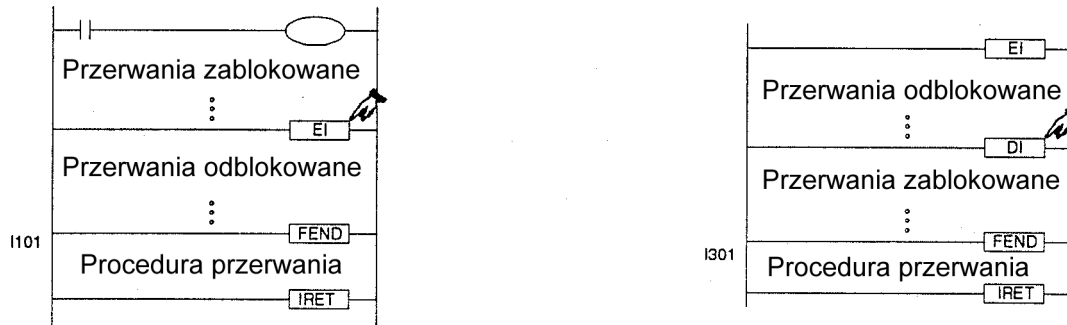
Procedury przerwania są zawsze umieszczane za instrukcją FEND.

Instrukcja IRET jest na ogół instrukcją bezwarunkową i może być używana tylko wewnątrz procedur przerwania.



Sterowanie działaniem przerwania:

Sterownik ma domyślnie zablokowane przerwanie. Aby uaktywnić przerwanie musi być użyta instrukcja EI. Wszystkie przerwanie, które pojawiają się fizycznie podczas okresu przebiegu programu od instrukcji EI do instrukcji FEND lub DI będą obsługiwane przez powiązane z nimi procedury przerwania. Jeżeli te przerwanie są zgłoszone poza zakresem instrukcji (EI-FEND lub EI-DI, zobacz diagram poniżej) to ich działanie będzie wstrzymane (zablokowane), aż do wykonania instrukcji EI w następnym cyklu przebiegu programu. W tym miejscu obsługa przerwania będzie uruchamiana.



Jeżeli pojedyncze przerwanie ma być wyłączone, powiązany z nim znacznik M musi być ustawiony w stan ON. Podczas gdy znacznik M będzie pozostawał w stanie ON, procedura przerwania nie będzie uaktywniona. Więcej szczegółów na temat blokujących znaczników M w tabelach urządzeń PC w rozdziale 8.

Zagnieżdżanie przerwania:

Przerwanie mogą być zagnieżdżane do 2 poziomów. To oznacza, że przerwanie może być przerwane podczas swojego działania. Jednakże, aby to osiągnąć, procedura obsługi przerwania, która może być przerwana, musi zawierać instrukcje EI i DI (na ogół bezwarunkowe). W przeciwnym razie, tak jak w normalnym działaniu, kiedy procedura przerwania jest aktywowana, wszystkie inne przerwanie będą zablokowane.

Równoczesne pojawienie się przerwania:

Jeżeli pojawi się równocześnie kilka przerwania, pierwszeństwo otrzymuje przerwanie, które pojawiło się pierwsze. Jeżeli dwa lub więcej przerwania pojawia się w tym samym momencie, procedura przerwania z niższym numerem wskaźnika otrzymuje wyższy priorytet.

Użycie timerów wewnątrz procedur obsługi przerwania:

Sterowniki FX mają szereg specjalnych timerów, które mogą być używane w procedurach obsługi przerwania. Więcej szczegółów na ten temat na stronie 4-8 - "Timery wykorzystywane w przerwaniach i podprogramach (CALL)".

Wejściowe sygnały wyzwalające - czas trwania impulsu:

Procedury obsługi przerwania, które są wyzwalane bezpośrednio przez wejścia przerywające, takie jak X0 itd., wymagają czasu trwania impulsu wynoszącego w przybliżeniu 200 μ s, tzn. szerokość impulsu wejściowego musi być równa lub większa niż 200 μ s. Jeżeli wybrany jest ten typ przerwania, stałe czasowe filtrów wejściowych są automatycznie ustalane na 50 μ s (w normalnych warunkach wynoszą one 10 ms).

Funkcja łapania impulsu:

Szybkie wejścia bezpośrednio mogą być używane do "łapania" krótkich sygnałów impulsowych. Kiedy impuls jest wykryty na wejściu, odpowiadający mu specjalny znacznik M jest ustawiany w stan ON. To umożliwia wykorzystanie "schwytanego" impulsu do wyzwolenia dalszych akcji, nawet jeżeli sygnał wejściowy przejdzie w stan OFF. Ta funkcja jest przez cały czas aktywna w jednostkach FX0, FX0S, FX0N dla wejść X0-X3 ze specjalnymi znacznikami M8056-M8059, przechowującymi dane impulsu. Jednostki FX(2C) i FX2N wymagają instrukcji EI (FNC 04) aby uaktywnić łapanie impulsu dla wejść X0-X5 ze znacznikami M8170-M8175, wskazujących złapany impuls. Należy zauważyć, że jeżeli urządzenie wejściowe jest używane dla innej szybkiej funkcji, wtedy łapanie impulsu dla tego urządzenia jest zablokowane.

5.1.5 FEND (FNC 06)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

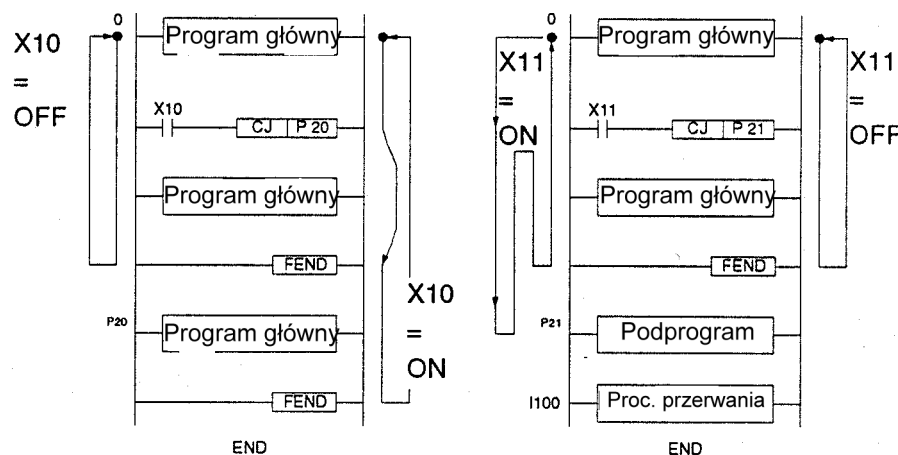
Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
FEND FNC 06 (pierwszy koniec)	używany do zaznaczenia końca bloku głównego programu	brak Uwaga: może być użyty z CJ (FNC 00), CALL (FNC 01) i programami obsługi przerwań	FEND: 1 krok

Działanie:

Bezwarunkowa instrukcja FEND wskazuje pierwszy koniec głównego programu i początek obszaru programu, który ma być użyty dla podprogramów. W normalnych warunkach instrukcja FEND spełnia podobną rolę do instrukcji END, tzn. wykonywane jest przetwarzanie wyjść, wejść i odświeżanie timera watchdoga.

Uwagi:

- a) Instrukcja FEND jest powszechnie używana z konstrukcjami CJ-P-FEND, CALL-P-SRET i I-IRET (P jest etykietą programu, I jest etykietą przerwania). Zarówno etykiety P/podprogramy CALL jak i etykiety I/podprogramy przerwań są zawsze umieszczane po instrukcji FEND, tzn. te cechy programu nigdy nie pojawiają się w programie głównym.



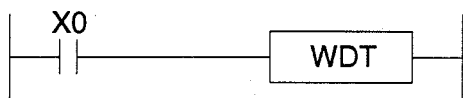
- b) Wielokrotne umieszczenie instrukcji FEND może być użyte do oddzielenia różnych podprogramów - zobacz diagram powyżej.
 c) Konstrukcja przebiegu programu nie może być rozdzielona przez instrukcje FEND.
 d) Instrukcja FEND nie może być użyta po instrukcji END.

5.1.6 WDT (FNC 07)

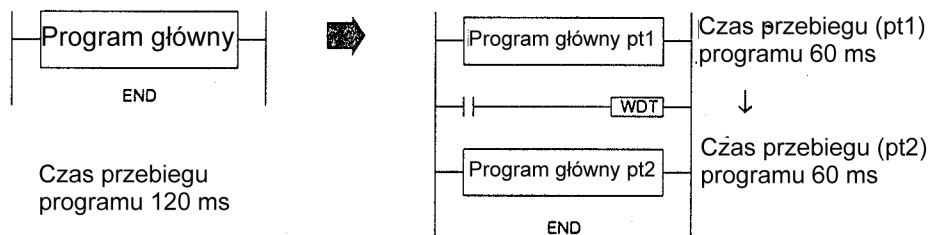
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
WDT FNC 07 (odświeżanie timera watchdoga)	używana do odświeżania timera watchdoga w czasie przebiegu programu	brak może być sterowana w każdym czasie wewnątrz głównego bloku programu	WDT, WDTP: 1 krok

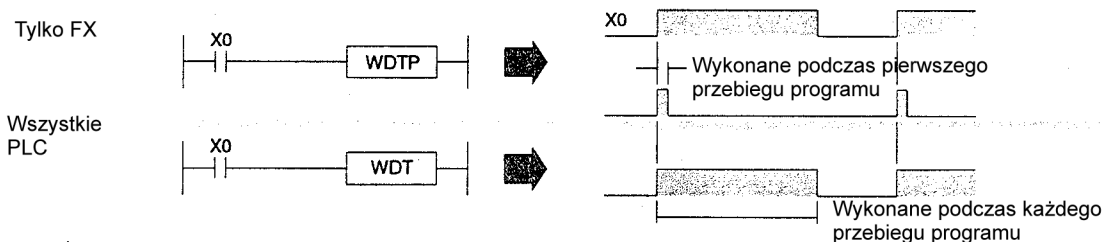
Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

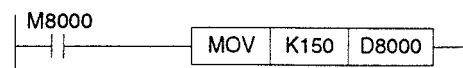
Instrukcja WDT odświeża timer watchdoga sterownika. Timer watchdoga sprawdza, czy czas przebiegu programu nie przekroczył zadanego limitu. Zakłada się, że jeżeli limit czasowy został przekroczony, w jakimś miejscu wystąpił błąd. Sterownik przerwie wtedy wykonywanie programu, aby zapobiec dalszemu pojawianiu się błędów. Przez odświeżanie timera watchdoga (działanie instrukcji WDT) czas przebiegu programu jest efektywnie podwyższany.

**Uwagi:**

- a) Kiedy używana jest instrukcja WDT, będzie ona działała przy każdym przebiegu programu, tak długo, jak pozwalają na to warunki wejściowe. Aby wymusić zadziałanie instrukcji WDT tylko jeden raz, należy użyć w programie pewnych form połączeń. Użytkownicy sterowników FX mają dodatkową opcję wykorzystującą format impulsowy (P) instrukcji WDT, tzn. WDTP.



- b) Timer watchdoga ma domyślne ustawienie 100 ms dla sterowników FX i 200 ms dla sterowników FX0, FX0N, FX2N. Ten limit czasowy może być dostosowany dla własnych potrzeb użytkownika poprzez zmianę zawartości rejestru D8000.

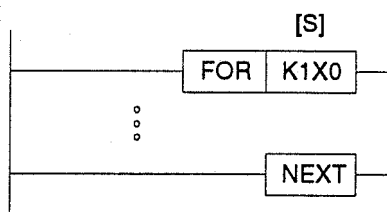


5.1.7 FOR, NEXT (FNC 08, 09)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
FOR FNC 08 (początek pętli FOR-NEXT)	identyfikuje pozycję startową oraz liczbę powtórzeń pętli	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	FOR: 3 kroki
NEXT FNC 09 (koniec pętli FOR-NEXT)	identyfikuje pozycję końcową pętli	brak Uwaga: pętla FOR-NEXT może być zagnieżdżona do 5 poziomów, tzn. można zaprogramować 5 pętli wewnątrz siebie	NEXT: 1 krok

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcje FOR i NEXT umożliwiają specyfikację obszaru programu, który będzie powtarzany S razy.

Uwagi:

- Instrukcja FOR pracuje w trybie 16-bitowym, stąd wartość argumentu S może być w zakresie od 1 do 32.767. Jeżeli liczba jest w zakresie od -32.768 do 0, jest automatycznie zastąpiona przez wartość 1, tzn. pętla FOR-NEXT będzie wykonana 1 raz.
- Instrukcja NEXT nie ma argumentu.
- Instrukcje FOR-NEXT **muszą** być zawsze użyte jako para, tzn. dla każdej instrukcji FOR **musi** istnieć związana z nią instrukcja NEXT. To samo odnosi się do instrukcji NEXT. Instrukcje FOR-NEXT muszą być programowane w odpowiednim porządku, tzn. programowanie pętli NEXT-FOR (instrukcja NEXT występuje przez związaną z nią instrukcją FOR) jest błędne.
Umieszczanie instrukcji FEND pomiędzy instrukcjami FOR-NEXT, tzn. FOR-FEND-NEXT jest niedozwolone. Miałoby to taki sam efekt, jak programowanie instrukcji FOR bez instrukcji NEXT, poprzedzającej instrukcję FEND i pętli z instrukcją NEXT bez instrukcji FOR.
- Pętla FOR-NEXT wykonuje się zadaną ilość razy **zanim** zakończy się główny przebieg programu.
- Podczas używania pętli FOR-NEXT należy uważać aby nie nastąpiło przepelnienie timera watchdoga. W przeciwnym razie sterownik zasygnalizuje błąd programu. Zaleca się użycie instrukcji WDT i/lub zwiększenie wartości timera watchdoga.

Zagnieżdżanie pętli FOR-NEXT:

Instrukcje FOR-NEXT mogą być zagnieżdżone do 5 poziomów. To oznacza, że 5 pętli FOR-NEXT może być kolejno zaprogramowanych wewnątrz siebie.

W przykładzie zostały zaprogramowane 3 poziomy zagnieżdżenia. Za każdym razem, gdy napotykane jest nowe zagnieżdżenie pętli FOR-NEXT, ilość powtórzeń tej pętli jest zwiększana tyle razy, ile wynosi iloczyn wszystkich poprzednich pętli.

Na przykład, pętla C wykonuje się 4 razy, ale wewnątrz tej pętli jest zagnieżdżona pętla B. Dla każdego ukończonego cyklu pętli C, pętla B będzie całkowicie wykonana D0Z razy. Sytuacja wygląda podobnie dla pętli B i A.

Całkowita liczba powtórzeń wykonania pętli A dla jednego przejścia programu wynosi:

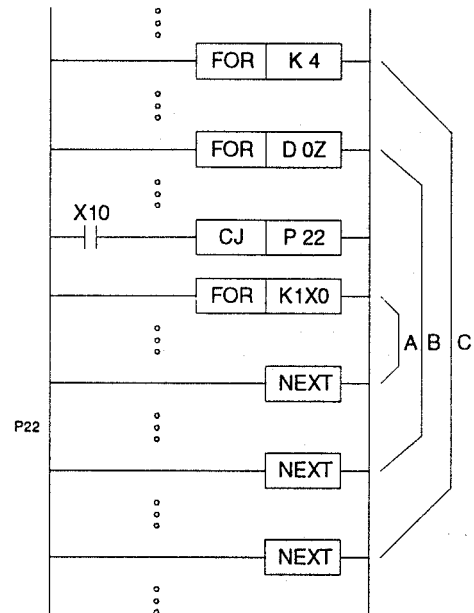
- 1) liczbę operacji pętli A, pomnożoną przez
- 2) liczbę operacji pętli B, pomnożoną przez
- 3) liczbę operacji pętli C

Jeżeli wartości iteracji dla pętli A, B i C wynosiłyby, odpowiednio, 7, 6 i 4, to w jednym przejściu programu poszczególne pętle zostałyby wykonane następującą liczbę razy:

Liczba operacji pętli C = 4 razy,

Liczba operacji pętli B = 24 razy (C x B, 4 x 6)


Liczba operacji pętli A = 168 razy (C x B x A, 4 x 6 x 7)

**Uwaga:**

Użycie instrukcji CJ powodujące skok do P22 pozwala na wyselekcjonowanie, która pętla będzie wykonana i kiedy, tzn. jeżeli X10 byłoby ustawione w stan ON, pętla A nie będzie się dłużej wykonywała.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

	1. Sterowanie przebiegiem programu	5-4
	2. Instrukcje przesłania i porównania	5-16
	3. Operacje arytmetyczne i logiczne	5-24
	4. Rotacje i funkcje przesunięcia	5-34
	5. Operacje na danych	5-42
	6. Szybkie przetwarzanie	5-52
	7. Instrukcje podręczne	5-66
	8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
	9. Zewnętrzne urządzenia szeregowo	5-94
	10. Zewnętrzne jednostki F2	5-112
	11. Operacje zmiennoprzecinkowe 1 & 2	5-120
	12. Operacje trygonometryczne	5-128
	13. Operacje na danych 2	5-132
	14. Sterowanie zegarem czasu rzeczywistego	5-136
	15. Kody Graya	5-144
	16. Porównania operacyjne	5-148

5.2 Instrukcje przesłania i porównania - funkcje 10 - 19

Zawartość:

			Strona
CMP	porównanie	FNC 10	5-17
ZCP	porównanie zakresów	FNC 11	5-17
MOV	przesłanie	FNC 12	5-18
SMOV	przesłanie z przesunięciem	FNC 13	5-18
CML	dopełnienie	FNC 14	5-19
BMOV	przesłanie blokowe	FNC 15	5-20
FMOV	przesłanie z wypełnieniem	FNC 16	5-21
XCH	zamiana	FNC 17	5-21
BCD	konwersja BCD	FNC 18	5-22
BIN	konwersja binarna	FNC 19	5-22



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową.

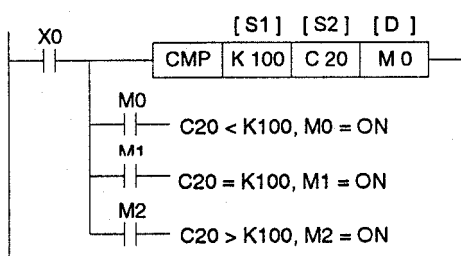
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.2.1 CMP (FNC 10)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
CMP FNC 10 (porównanie)	porównuje dwie wartości, zwracając wynik $<. = i >$	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		Y, M, S Uwaga: używane są 3 kolejne urządzenia	CMP, CMPP: 7 kroków DCMP, DCMPP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Dana S₁ jest porównywana z daną S₂. Wynik porównania jest wpisywany do 3 kolejnych urządzeń bitowych o adresie początkowym D. Urządzenia bitowe wskazują:

S₂ jest mniejsze od S₁ - urządzenie bitowe D jest w stanie ON

S₂ jest równe S₁ - urządzenie bitowe D₊₁ jest w stanie ON

S₂ jest większe od S₁ - urządzenie bitowe D₊₂ jest w stanie ON

**Uwaga:**

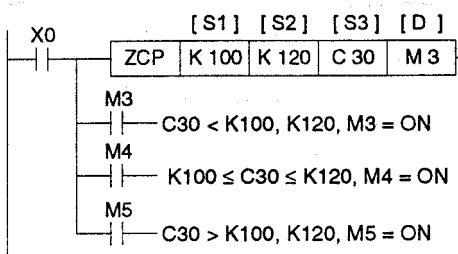
Stany urządzeń wyniku porównania są utrzymywane jeśli instrukcja CMP jest nieaktywna. Używane są porównania algebraiczne, tzn. $-10 < +2$ itd.

5.2.2 ZCP (FNC 11)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	S ₃	
ZCMP FNC 11 (porównanie zakresów)	porównuje wartość danej z zakresem danych, zwraca wynik $<, = i >$	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		Y, M, S Uwaga: używane są 3 kolejne urządzenia	ZCP, ZCPP: 9 kroków DZCP, DZCPP: 17 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Działanie instrukcji jest takie samo, jak instrukcji CMP, z tym, że wartość pojedynczej danej S₃ jest porównywana z zakresem danych (S₁-S₂).

S₃ jest mniejsze niż S₁ i S₂ - urządzenie bitowe D jest w stanie ON

S₃ jest równe, lub jest pomiędzy S₁ i S₂ - urządzenie bitowe D₊₁ jest w stanie ON

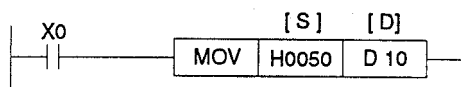
S₃ jest większe od S₁ i S₂ - urządzenie bitowe D₊₂ jest w stanie ON

5.2.3 MOV (FNC 12)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
MOV FNC 12 (przesłanie)	przesyła dane z jednego obszaru w drugi	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	MOV, MOV _P : 5 kroków DMOV, DMOV _P : 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Kiedy wejście sterujące jest aktywne, zawartość urządzenia źródłowego S jest wpisywana (kopiowana) do urządzenia docelowego D. Jeśli instrukcja MOV nie jest aktywna, żadna operacja nie jest wykonywana.

**Uwaga:**

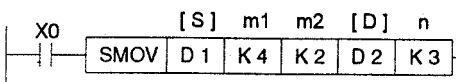
Ta instrukcja daje specjalną technikę programowania, która umożliwia naśladowanie operacji nowych instrukcji specjalnych przy użyciu starszych narzędzi programowania. Szczegóły na stronie 1-5.

5.2.4 SMOV (FNC 13)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty					Kroki programu
		m ₁	m ₂	n	S	D	
SMOV FNC 13 (przesłanie z przesunięciem)	pobiera elementy 4-cyfrowej liczby dziesiętnej i wstawia je do nowej liczby 4-cyfrowej	K, H			K, H KnX, KnY, KnM, KnS, T, C, D, V, Z	K, H KnY, KnM, KnS, T, C, D, V, Z	SMOV, SMOV _P : 11 kroków
		Uwaga: dostępny zakres 1 do 4 ☒			zakres 0 do 9.999 (dziesiętnie) lub 0 do 9.999 (BCD), gdy użyty jest M8168 – patrz uwaga obok		

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie 1:**

Instrukcja kopiuje określoną liczbę cyfr z 4-cyfrwego źródła dziesiętnego (S) i umieszcza je w określonym miejscu wewnątrz docelowej liczby (D, również dziesiętnej, 4-cyfrowej). Dana występująca w miejscu przeznaczenia jest nadpisywana. Tryb manipulacji

dziesiętnej jest dostępny na wszystkich jednostkach FX i FX2C. Klucz:

m₁ - pozycja źródłowa pierwszej przesyłanej cyfry

m₂ - liczba źródłowych cyfr do przesłania

n - pozycja docelowa dla pierwszej cyfry

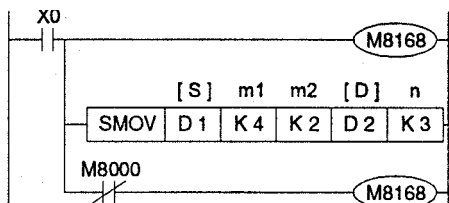
Uwaga: Wybrane miejsce przeznaczenia nie może być mniejsze niż ilość danych źródłowych. Pozycje cyfr są oznaczone przez liczby: 1 - jednostki, 2 - dziesiątki, 3 - setki, 4 - tysiące

Działanie 2:

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

(Dotyczy jednostek FX z CPU wersja 3.07 i późniejsze oraz FX_{2C})

Ta modyfikacja operacji SMOV pozwala manipulowanie liczbami BCD w taki sam sposób, jak normalna instrukcja SMOV manipuluje liczbami dziesiętnymi, tzn. ta instrukcja kopiuje określoną liczbę cyfr z 4-cyfrowego źródła BCD (S) i umieszcza je w określonym miejscu wewnątrz liczby przeznaczenia (D, 4-cyfrowa liczba BCD).



Żeby wybrać tryb BCD, instrukcja SMOV jest łączona ze znacznikiem M8168, który jest ustawiony w stan ON. Należy pamiętać, że jest to operacja ustawiana 'trybem' i będzie aktywna, tzn. wszystkie instrukcje SMOV będą działać w formacie BCD, aż tryb zostanie wyzerowany, tzn. M8168 będzie ustawiony w stan OFF.

**Ogólna uwaga:**

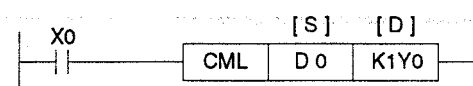
Więcej informacji na temat liczb dziesiętnych i liczb BCD w sekcji zatytułowanej "Interpretacja danych słowowych" na stronie 4-42.

5.2.5 CML (FNC 14)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
CML FNC 14 (uzupełnienie)	kopiuje i odwraca źródłowy wzorzec bitowy w określonym urządzeniu docelowym	K, H KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	CML, CMLP: 5 kroków DCML, DCMLP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

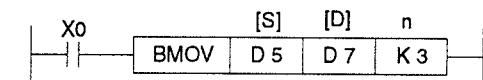
Kopia każdego bitu danych urządzenia źródłowego S jest odwracana i przesyłana do urządzenia docelowego D. Oznacza to, że każde pojawienie się "1" w danej źródłowej stanie się "0" w danej docelowej, podczas gdy każda cyfra źródłowa, która jest "0" stanie się "1". Jeżeli obszar przeznaczenia jest mniejszy niż dana źródłowa, wtedy odwzorowanie zostanie wykonane tylko dla pasujących bitów.

5.2.6 BMOV (FNC 15)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
BMOV FNC 15 (przesłanie bloku)	kopiuje określony blok danych do nowego miejsca	KnX, KnY, KnM, KnS, T, C, D, V, Z rejstry plikowe (RAM)	KnY, KnM, KnS, T, C, D, V, Z rejstry plikowe (RAM), patrz uwaga d)	K, H ☒ uwaga: n ≤ 512	BMOV, BMOV P: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

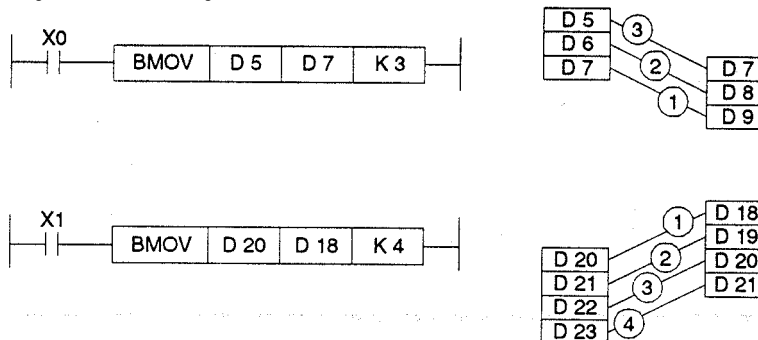
Działanie:

Następujące po sobie elementy danych są kopiowane do miejsca przeznaczenia. Dana źródłowa definiuje adres początkowy (S) i ilość występujących po sobie elementów danych (n). Do urządzenia przeznaczenia (D) przesyłana jest ta sama liczba elementów danych (n).

Do urządzenia przeznaczenia (D) przesyłana jest ta sama liczba elementów danych (n).

Uwagi:

- Jeżeli ilość danych w urządzeniu źródłowym (n) przewyższa bieżącą ilość dostępnych urządzeń źródłowych, to wykorzystane zostaną tylko te urządzenia, które mieszczą się w zakresie.
- Jeżeli ilość urządzeń źródłowych przewyższa dostępną przestrzeń w obszarze docelowym, wtedy zapisane zostaną tylko dostępne urządzenia docelowe.
- Instrukcja BMOV ma wbudowaną cechę, która zapobiega pojawieniu się błędów nadpisania, kiedy zakresy źródła (S - n) i miejsca przeznaczenia (D - n) danych zachodzą na siebie. Jest to pokazane na następującym diagramie: (Uwaga: numerowane linie wskazują kolejność, w jakiej wykonywana jest instrukcja BMOV).



FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

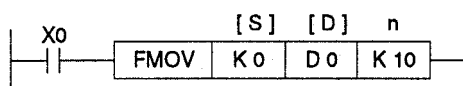
- Transfer do rejestrów plikowych jako urządzenia docelowego [D] może być wykonywane tylko na sterownikach FX z wersją CPU 3.07 i wyższą, lub na każdym FX2C lub FX2N.

5.2.7 FMOV (FNC 16)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
FMOV FNC 16 (wypełnienie bloku)	kopiuje pojedynczą daną do określonego zakresu urządzeń docelowych	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	K, H <input type="checkbox"/> uwaga: n ≤ 512	FMOV, FMOV _P : 7 kroków DFMOV, DFMOV _P : 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie: Dane zapamiętane w urządzeniu źródłowym S są kopiowane do każdego urządzenia w zakresie przeznaczenia. Zakres jest określany przez adres początkowy (D) i liczbę następujących po sobie elementów (n). Jeżeli podana liczba urządzeń przeznaczenia (n)

przewyższa dostępną przestrzeń w obszarze docelowym, wtedy zostaną zapisane tylko dostępne urządzenia docelowe. Należy zauważyć, że operacje na podwójnych słowach (32 bity) mogą być wykonywane tylko na sterownikach FX z wersją CPU 3.07 i wyższą, lub na każdym FX2C.



Uwaga: Ta instrukcja daje specjalną technikę programowania, która umożliwia naśladowanie operacji nowych instrukcji specjalnych przy użyciu starszych narzędzi programowania. Szczegóły na stronie 1-5.

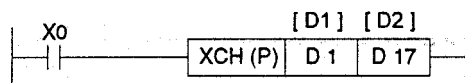
5.2.8 XCH (FNC 17)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D ₁	D ₂	
XCH FNC 17 (zamiana) →	dane w wyszczególnionych urządzeniach są zamieniane	KnY, KnM, KnS, T, C, D, V, Z Uwaga: jeśli XCH operuje na bajtach (M8160 = ON), D ₁ i D ₂ muszą być tym samym urządzeniem, w przeciwnym razie wystąpi błąd programu, a M8067 ustawi się w stan ON.		XCH, XCH _P : 5 kroków DXCH, DXCH _P : 9 kroków

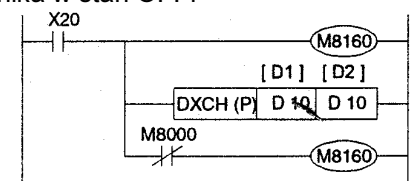
Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

Działanie 1: (Dostępne na jednostkach FX i FX2C). Zawartość dwóch urządzeń docelowych D₁ i D₂ jest zamieniona miejscami. Np.



Rejestr	Przed XCH	Po XCH
D ₁	20	530
D ₁₇	530	20

Działanie 2: (Dostępne na jednostce FX(2C)). Ta funkcja jest równoważna z funkcją SWAP (FNC 147). Bajty wewnątrz każdego słowa urządzenia docelowego D₁ są wymieniane, kiedy znacznik trybu bajtowego M8160 jest w stanie ON. Należy zauważyć, że tryb pozostaje aktywny do momentu wyzerowania znacznika, tzn. ustawienia znacznika w stan OFF.



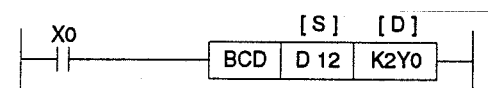
	Wartości szesnastkowe	Przed DXCH	Po DXCH
D10	Bajt 1	1FH	8BH
	Bajt 2	8BH	1FH
D11	Bajt 1	C4H	35H
	Bajt 2	35H	C4H

5.2.9 BCD (FNC 18)

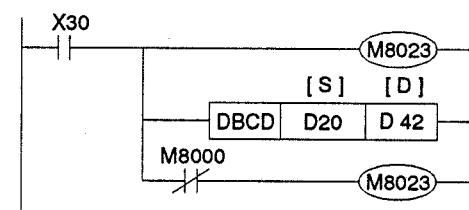
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
BCD FNC 18 (liczba dziesiętna kodowana dwójkowo)	zamienia liczby dwójkowe na ich odpowiedniki w kodzie BCD/ zamienia dane zmiennoprzecinkowe na format naukowy	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BCD, BCDP: 5 kroków DBCD, DBCDP: 9 kroków
		jeśli używany jest znacznik M8023, mogą być wykorzystane tylko rejestry podwójnych słów. Więcej szczegółów na temat formatów zmiennoprzecinkowych na stronie 4-46		

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie 1: (Dostępne na wszystkich jednostkach). Binarne dane źródłowe (S) są konwertowane na odpowiadającą im liczbę BCD i przechowane w urządzeniu docelowym (D). Jeżeli konwertowana liczba BCD przewyższa granice zakresu 0-9.999 (operacja 16-bitowa) i 0-99.999.999 (operacja 32-bitowa), pojawi się błąd. Instrukcja może być użyta do bezpośredniego wysłania danej do wyświetlacza 7-segmentowego.



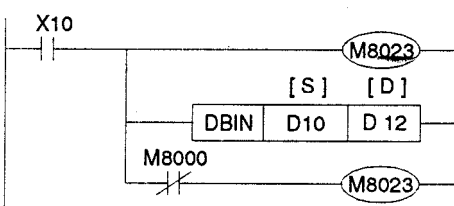
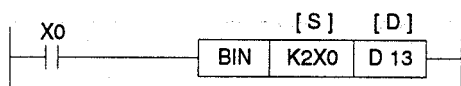
Działanie 2: (Dostępne na jednostkach FX_(2C)). Dana (S) jest konwertowana z formatu zmiennoprzecinkowego na format naukowy (D). Do prawidłowego działania instrukcja ta wymaga operacji na 32-bitowych rejestrach danych jako urządzeniach S i D.

5.2.10 BIN (FNC 19)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
BIN FNC 19 (dwójkowa)	zamienia liczby BCD na ich odpowiedniki dwójkowe / zamienia format naukowy na zmiennoprzecinkowy	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BIN, BINP: 5 kroków DBIN, DBINP: 9 kroków
		jeśli używany jest znacznik M8023, mogą być wykorzystane tylko rejestry podwójnych słów. Więcej szczegółów na temat formatów zmiennoprzecinkowych na stronie 4-46		

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie 1: (Dostępne na wszystkich jednostkach). Dana źródłowa BCD (S) jest konwertowana na odpowiadającą jej liczbę binarną i przechowana w urządzeniu docelowym (D). Jeżeli dana źródłowa nie będzie podana w formacie BCD, wystąpi błąd. Ta instrukcja może być użyta do bezpośredniego odczytu danych z zadajników kodowych.

Działanie 2: (Dostępne na jednostkach FX_(2C)). Ta funkcja jest równoważna funkcji EBIN (FNC 119). Dana (S) jest konwertowana z formatu naukowego do formatu zmiennoprzecinkowego (D). Do prawidłowego działania instrukcja ta wymaga operacji na 32-bitowych rejestrach danych jako urządzeniach S i D.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------



1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.3 Operacje arytmetyczne i logiczne - funkcje 20 - 29

Zawartość:

			Strona
ADD	dodawanie	FNC 20	5-25
SUB	odejmowanie	FNC 21	5-26
MUL	mnożenie	FNC 22	5-27
DIV	dzielenie	FNC 23	5-28
INC	inkrementacja	FNC 24	5-29
DEC	dekrementacja	FNC 25	5-29
WAND	operacja AND na słowach	FNC 26	5-30
WOR	operacja OR na słowach	FNC 27	5-30
WXOR	operacja XOR na słowach	FNC 28	5-31
NEG	negacja	FNC 29	5-31



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1 , S_3 lub dla urządzeń stabilizowanych D_{3+0} , S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

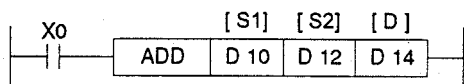
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.3.1 ADD (FNC 20)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
ADD FNC 20 (dodawanie)	wartości dwu urządzeń źródłowych są dodawane a wynik jest przechowany w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	jeśli używany jest znacznik M8023, mogą być wykorzystane tylko rejestry podwójnych słów lub stałe (K/H). Więcej szczegółów na temat formatów zmiennoprzecinkowych na stronie 4-46	ADD, ADDP: 7 kroków DADD, DADDP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Zero M8020 Pożyczka M8021 Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

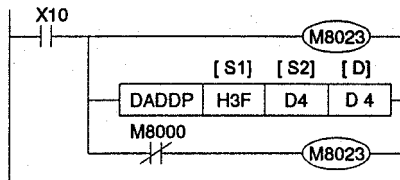


Działanie 1: (Dostępne na wszystkich jednostkach). Dane zawarte w urządzeniach źródłowych (S1, S2) są sumowane a wynik jest zapisywany we wskazanym urządzeniu docelowym (D).

Uwaga:

- Wszystkie obliczenia są wykonywane algebraiczne, tzn. $5 + (-8) = -3$
- Te same urządzenia mogą być wykorzystane jako źródłowe (S1 lub S2) i jako docelowe (D). W takim przypadku instrukcja ADD będzie działać w sposób ciągły. Oznacza to, że przy każdym przebiegu programu instrukcja doda wynik ostatniego przebiegu programu do drugiego urządzenia źródłowego. Aby temu zapobiec, należy użyć wywołania impulsowego lub zaprogramować wewnętrzną blokadę.
- Jeżeli wynik obliczenia wynosi "0", specjalny znacznik M8020 jest ustawiony w stan ON.
- Jeżeli wynik operacji przekracza 32.767 (ograniczenie 16-bitowe) lub 2.147.483.647 (ograniczenie 32-bitowe), znacznik przeniesienia M8022 ustawiany jest w stan ON. Jeżeli wynik obliczenia przekroczy -32.768 lub -2.147.483.648, znacznik pożyczki M8021 jest ustawiany w stan ON. Jeśli wynik przekroczy któryś z zakresów liczby, odpowiedni znacznik jest ustawiany w stan ON (M8021 lub M8022) i część przeniesienia/pożyczki jest przechowana w urządzeniu docelowym. Znak matematyczny tej przechowanej danej jest odzwierciedleniem zakresu, który został przekroczony, tzn. kiedy przekroczony jest zakres -32.768, w urządzeniu docelowym przechowane są liczby ujemne, ale kiedy zostanie przekroczony zakres 32.767, w urządzeniu docelowym D będą przechowane liczby dodatnie.
- Jeżeli miejsce docelowe jest mniejsze od otrzymanego wyniku, wtedy tylko część wyniku będzie zapisana do obszaru przeznaczenia, tzn. jeżeli wynikiem jest 25 (dziesiętnie) i ma być przechowany w K1Y4, wtedy tylko Y4 i Y7 będzie aktywne. W układzie dwójkowym jest to równoważne wartości dziesiętnej równej 9, dalekiej od prawdziwego wyniku, równego 25.

Ciąg dalszy na następnej stronie...



Działanie 2: (Dostępne dla sterowników FX_(2C)). Ta funkcja jest równoważna funkcji EADD (FNC 120). Gdy znacznik trybu zmiennoprzecinkowego M8023 jest w stanie ON, instrukcje DADD i DADDP mogą być użyte do wykonania dodawania zmiennoprzecinkowego.

Kiedy M8023 jest w stanie OFF, nie można wykonywać manipulacji zmiennoprzecinkowych. Stałe (K/H) i liczby zmiennoprzecinkowe (przechowywane w podwójnym rejestrze danych D) mogą być dodawane w każdej konfiguracji. Stałe (K/H) będą automatycznie konwertowane do formatu zmiennoprzecinkowego dla operacji dodawania. Wyniki operacji mogą być przechowywane tylko w podwójnym rejestrze danych (32 bity). Punkty a) i b) również obowiązują dla tego trybu operacyjnego.

FX2N realizacja operacji zmiennoprzecinkowych



Uwaga: W jednostkach FX2N nie używa się znacznika M8023 dla operacji zmiennoprzecinkowych.

Zamiast tego powinna być użyta dedykowana instrukcja zmiennoprzecinkowa, np. zamiast DADD ze znacznikiem M8023 w stanie ON należy użyć DEADD (FNC 120).



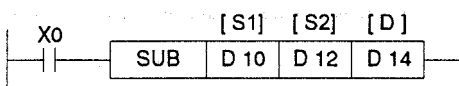
Zobacz sekcja 5-11.

5.3.2 SUB (FNC 21)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
SUB FNC 21 (odejmowanie)	zawartość jednego urządzenia źródłowego jest odejmowana od drugiego, wynik jest zachowany w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	jeśli używany jest znacznik M8023, mogą być wykorzystane tylko rejestry podwójnych słów lub stałe (K/H). Więcej szczegółów na temat formatów zmiennoprzecinkowych na stronie 4-46	SUB, SUBP: 7 kroków DSUB, DSUBP: 13 kroków

Instrukcja impulsowa (P)				Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Zero M8020 Pożyczka M8021 Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)		



Działanie 1: (Dostępne na wszystkich jednostkach).

Dane zawarte w urządzeniu źródłowym S2 są odejmowane od zawartości urządzenia źródłowego S1. Wynik jest zapamiętany w urządzeniu docelowym D. Uwaga: "Uwagi"

pod instrukcją ADD (poprzednia strona) mogą być zastosowane do instrukcji SUB

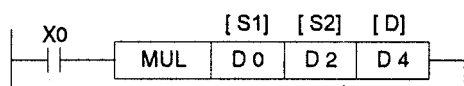
Działanie 2: (Dostępne w jednostkach FX_(2C)). Ta funkcja jest równoważna funkcji ESUB (FNC 121). Informacja dotycząca "Działanie 2" instrukcji ADD jest podobna do tej drugiej operacji instrukcji SUB (z wyjątkiem tego, że zamiast operacji dodawania wykonywana jest operacja odejmowania). Podobnie, tylko stałe i podwójne słowa danych mogą być wykorzystane z instrukcjami DSUB, DSUBP.

5.3.3 MUL (FNC 22)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
MUL FNC 22 (mnożenie)	wartości dwu urządzeń źródłowych są mnożone a wynik jest przechowywany w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, Z(V)	Uwaga: rejestr Z(V) nie może być używany w operacjach 32-bitowych	ADD, ADDP: 7 kroków DADD, DADDP: 13 kroków
		jeśli używany jest znacznik M8023, mogą być wykorzystane tylko rejestry podwójnych słów lub stałe (K/H). Więcej szczegółów na temat formatów zmiennoprzecinkowych na stronie 4-46			

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



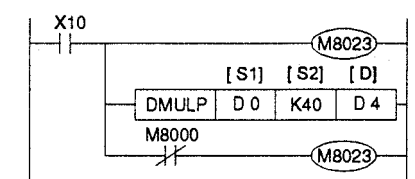
Działanie 1: (Dostępne na wszystkich jednostkach). Zawartość dwóch urządzeń źródłowych (S1, S2) są mnożone, a wynik jest przechowywany w urządzeniu docelowym D. Obowiązują normalne reguły algebraiczne.

Uwagi:

- Podczas działania instrukcji MUL w trybie 16-bitowym, mnożone są dwa 16-bitowe źródła danych. Daje to wynik 32-bitowy. Urządzenie oznaczone jako adres docelowy wskazuje młodsze z dwóch urządzeń, użytych do przechowania wyniku 32-bitowego. Dla powyższego rysunku z przykładowymi danymi otrzymuje się: $5(D0) \times 7(D2) = 35$ - wartość 35 jest przechowana w (D4, D5) jako pojedyncze słowo 32-bitowe.
- Podczas działania instrukcji MUL w trybie 32-bitowym, mnożone są dwa 32-bitowe źródła danych. Daje to wynik 64-bitowy. Urządzenie oznaczone jako adres docelowy wskazuje najmłodsze z czterech urządzeń, użytych do przechowania wyniku 64-bitowego.
- Jeżeli miejsce urządzenia docelowego jest mniejsze od otrzymanego wyniku, wtedy tylko część wyniku będzie zapamiętana, tzn. jeżeli wynik wynoszący 72 (dziesiętne), ma być przechowany w K1Y4, wtedy tylko Y7 będzie aktywne. W układzie dwójkowym jest to równoważne wartości dziesiętnej równej 8, dalekiej od prawdziwego wyniku, równego 72.

**Przeglądanie liczb 64-bitowych**

W chwili obecnej jest niemożliwe przeglądanie zawartości wyniku 64-bitowego. Jednakże, wynik może być przeglądany w dwóch 32-bitowych blokach, tzn. wynik 64-bitowy składa się z następujących części: (starsze 32 bity) $\times 2^{32}$ + (młodsze 32 bity)



Działanie 2: (Dostępne w jednostkach FX_(2C)). Ta funkcja jest równoważna funkcji EMUL (FNC 122). Gdy znacznik trybu zmiennoprzecinkowego M8023 jest w stanie ON, instrukcje DMUL i DMULP mogą być użyte do wykonania mnożenia zmiennoprzecinkowego. Kiedy M8023 jest w stanie OFF, nie

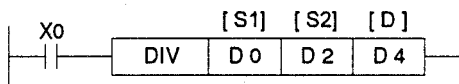
można wykonywać manipulacji zmiennoprzecinkowych. Stałe (K/H) i liczby zmiennoprzecinkowe (przechowywane w podwójnym rejestrze danych D) mogą być mnożone w każdej konfiguracji. Stałe (K/H) będą automatycznie konwertowane do formatu zmiennoprzecinkowego dla operacji mnożenia. Wyniki operacji mogą być przechowywane (całkowicie) w parze podwójnych rejestrów danych (32 bity), a nie w dwóch parach (64 bity), jak to się dzieje w "Działaniu 1". Obowiązują normalne algebraiczne zasady mnożenia.

5.3.4 DIV (FNC 23)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
DIV FNC 23 (dzielenie)	dzieli wartość jednego urządzenia źródłowego przez drugie i zachowuje wynik w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, Z(V)	DIV, DIVP: 7 kroków DDIV, DDIVP: 13 kroków
		jeśli używany jest znacznik M8023, mogą być wykorzystane tylko rejestry podwójnych słów lub stałe (K/H). Więcej szczegółów na temat formatów zmiennoprzecinkowych na stronie 4-46		Uwaga: rejestr Z(V) nie może być używany w operacjach 32-bitowych	

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie 1: (Dostępne na wszystkich jednostkach). Dana z pierwszego źródła (S1) jest dzielona przez daną z drugiego źródła (S2). Wynik jest zapamiętany w urządzeniu docelowym (D). Obowiązują normalne reguły algebraiczne.

Uwagi:

- a) Podczas działania instrukcji DIV w trybie 16-bitowym, dwa 16-bitowe źródła danych są dzielone przez siebie. Daje to 2 wyniki 16-bitowe. Urządzenie, oznaczone w instrukcji jako docelowe, jest młodszym z dwóch urządzeń, użytych do przechowania tych wyników i zawiera wynik dzielenia (iloraz).

Drugi, kolejny rejestr zawiera resztę z dzielenia. Dla powyższego rysunku z przykładowymi danymi otrzymuje się:

$$51(D0) / 10(D2) = 5(D4) 1(D5)$$

Wynik ten jest interpretowany jako 5 reszta 1 ($5 \times 10 + 1 = 51$)

- b) Podczas działania instrukcji DIV w trybie 32-bitowym, dwa 32-bitowe źródła danych są dzielone przez siebie. Daje to 2 wyniki 32-bitowe. Urządzenie, oznaczone w instrukcji jako docelowe, jest młodszym z dwóch urządzeń i służy do przechowania wyniku (ilorazu). Kolejne (starsze) dwa urządzenia są użyte do przechowania reszty. Np.: jeżeli D30 zostało wybrane jako przeznaczenie 32-bitowej operacji dzielenia wtedy D30, D31 przechowują iloraz a D32, D33 przechowują resztę. Jeżeli miejsce urządzenia docelowego jest mniejsze od otrzymanego wyniku, wtedy tylko część wyniku będzie zapamiętana. Jeżeli jako urządzenia docelowe są użyte urządzenia bitowe, wtedy reszta nie jest obliczana.
- c) Jeżeli wartość urządzenia źródłowego jest równa 0, pojawia się błąd operacji i instrukcja DIV jest anulowana.

Działanie 2: (Dostępne w jednostkach FX_(2C)). Ta funkcja jest równoważna funkcji EDIV (FNC 123). Informacja dotycząca "Działanie 2" instrukcji MUL jest podobna do tej drugiej operacji instrukcji DIV (z wyjątkiem tego, że zamiast operacji mnożenia wykonywana jest operacja dzielenia). Podobnie, tylko stałe i podwójne słowa danych mogą być wykorzystane z instrukcjami DDIV, DDIVP. Wyniki operacji są przechowywane (w całości) w jednej parze podwójnych (32-bitowych) rejestrów danych, tzn. wyniki nie są rozbite na iloraz i resztę jak w "Działaniu 1". Obowiązują normalne reguły algebraiczne dzielenia zmiennoprzecinkowego.

5.3.5 INC (FNC 24)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
INC FNC 24 (inkrementacja) →	zawartość określonego urządzenia jest inkrementowana o 1 za każdym wykonaniem instrukcji	KnY, KnM, KnS, T, C, D, V, Z dla operacji 32-bitowych obowiązują standardowe reguły rejestrów V, Z	INC, INCP: 3 kroki DINC, DINCP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Podczas każdego wykonania instrukcji wartość urządzenia, podanego jako docelowe jest inkrementowana (zwiększana) o 1.

Dla operacji 16-bitowych, gdy wartość osiągnie 32.767, następna inkrementacja spowoduje w urządzeniu docelowym zapis -32.768.

Dla operacji 32-bitowych, gdy wartość osiągnie 2.147.483.647, następna inkrementacja spowoduje w urządzeniu docelowym zapis -2.147.483.648.

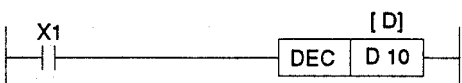
W obu przypadkach nie są ustawiane żadne dodatkowe znaczniki, sygnalizujące zmianę wartości danej.

5.3.6 DEC (FNC 25)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
DEC FNC 25 (dekrementacja) →	zawartość określonego urządzenia jest dekrementowana o 1 za każdym wykonaniem instrukcji	KnY, KnM, KnS, T, C, D, V, Z dla operacji 32-bitowych obowiązują standardowe reguły rejestrów V, Z	DEC, DECP: 3 kroki DDEC, DDECP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Podczas każdego wykonania instrukcji wartość urządzenia, podanego jako docelowe jest dekrementowana (zmniejszana) o 1.

Dla operacji 16-bitowych, gdy wartość osiągnie -32.768, następna dekrementacja spowoduje w urządzeniu docelowym zapis 32.767.

Dla operacji 32-bitowych, gdy wartość osiągnie -2.147.483.648, następna dekrementacja spowoduje w urządzeniu docelowym zapis 2.147.483.647.

W obu przypadkach nie są ustawiane żadne dodatkowe znaczniki, sygnalizujące zmianę wartości danej.

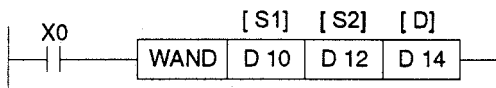
5.3.7 WAND (FNC 26)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
----------	---------	-----------	----------------

		S ₁	S ₂	D	
WAND FNC 26 (logiczne AND na słowach)	na zawartości urządzeń źródłowych wykonywana jest operacja logiczna AND, wynik zachowany w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WAND, WANDP: 7 kroków DAND, DANDP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Analizowany jest wzorec bitowy dwóch urządzeń źródłowych (zawartość S2 jest mnożona logicznie (AND) z zawartością S1). Wynik operacji logicznej WAND jest przechowany w urządzeniu

docelowym (D).

Do określenia wyniku logicznego instrukcji WAND stosuje się następujące reguły. Operację wykonuje się na każdej parze bitów urządzeń źródłowych:

Reguła ogólna: (S1) Bit n AND (S2) Bit n = (D) Bit n

1 WAND 1 = 1

0 WAND 1 = 0

1 WAND 0 = 0

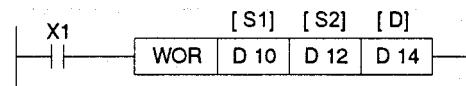
0 WAND 0 = 0

5.3.8 WOR (FNC 27)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
WOR FNC 27 (logiczne OR na słowach)	na zawartości urządzeń źródłowych wykonywana jest operacja logiczna OR, wynik zachowany w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WOR, WORP: 7 kroków DOR, DORP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Analizowany jest wzorec bitowy dwóch urządzeń źródłowych (zawartość S2 jest sumowana logicznie (OR) z zawartością S1). Wynik operacji logicznej OR jest przechowany w urządzeniu docelowym (D).

Do określenia wyniku logicznego instrukcji WOR stosuje się następujące reguły. Operację wykonuje się na każdej parze bitów urządzeń źródłowych:

Reguła ogólna: (S1) Bit n OR (S2) Bit n = (D) Bit n

1 OR 1 = 1

0 OR 1 = 1

1 OR 0 = 1

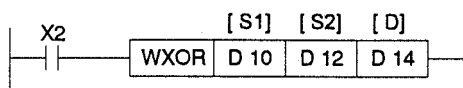
0 OR 0 = 0

5.3.9 WXOR (FNC 28)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
WXOR FNC 28 (logiczne XOR na słowach)	na zawartości urządzeń źródłowych wykonywana jest operacja logiczna XOR, wynik zachowany w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WXOR, WXORP: 7 kroków DXOR, DXORP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Analizowany jest wzorzec bitowy dwóch urządzeń źródłowych (zawartość S2 jest sumowana logicznie z wyłączeniem (XOR) z zawartością S1). Wynik operacji logicznej WXOR jest przechowywany w urządzeniu docelowym (D).

Do określenia wyniku logicznego instrukcji WXOR stosuje się następujące reguły. Operację XOR wykonuje się na każdej parze bitów urządzeń źródłowych:

Reguła ogólna: (S1) Bit n XOR (S2) Bit n = (D) Bit n

1 XOR 1 = 0

0 XOR 1 = 1

1 XOR 0 = 1

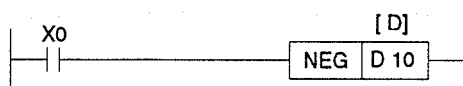
0 XOR 0 = 0

5.3.10 NEG (FNC 29)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
NEG FNC 29 (negacja)	zawartość określonego urządzenia jest odwracana logicznie	KnY, KnM, KnS, T, C, D, V, Z	NEG, NEGP: 3 kroki DNEG, DNEGP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Wzorzec bitowy wybranego urządzenia jest odwracany. To oznacza, że każde wystąpienie "1" jest zamieniane na "0", a każde wystąpienie "0" jest zamieniane na "1". Po zakończeniu operacji odwracania do wzorca dodawana jest binarna 1. Wynikiem jest zmiana znaku zawartości wybranego urządzenia, tzn. liczba dodatnia staje się ujemną, a ujemna dodatnią.

Notatki

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------



1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.4 Rotacje i przesunięcia - funkcje 30 - 39

Zawartość:

			Strona
ROR	rotacja w prawo	FNC 30	5-35
ROL	rotacja w lewo	FNC 31	5-35
RCR	rotacja w prawo z przeniesieniem	FNC 32	5-36
RCL	rotacja w lewo z przeniesieniem	FNC 33	5-36
SFTR	bitowe przesunięcie w prawo	FNC 34	5-37
SFTL	bitowe przesunięcie w lewo	FNC 35	5-37
WSFR	słowne przesunięcie w prawo	FNC 36	5-38
WSFL	słowne przesunięcie w lewo	FNC 37	5-38
SFWR	zapis z przesunięciem	FNC 38	5-39
SFRD	odczyt z przesunięciem	FNC 39	5-40



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

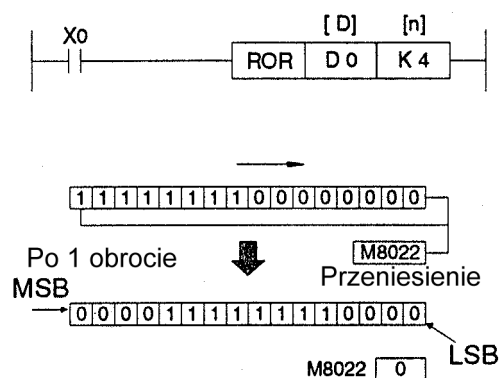
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.4.1 ROR (FNC 30)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D	n	
ROR FNC 30 (rotacja w prawo) →	wzorec bitowy urządzenia docelowego jest rotowany (obracany) o 'n' miejsc w prawo	KnY, KnM, KnS, T, C, D, V, Z uwaga: operacje 16-bitowe Kn=K4, operacje 32-bitowe Kn=K8	K, H, ☒ uwaga: operacja 16-bitowe n ≤ 16 operacja 32-bitowa n ≤ 32	ROR, RORP: 5 kroków DROR, DRORP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Wzorec bitowy urządzenia docelowego jest rotowany (obracany) o n miejsc w prawo.

Stan ostatniego przesuniętego bitu jest kopiowany do znacznika przeniesienia M8022.

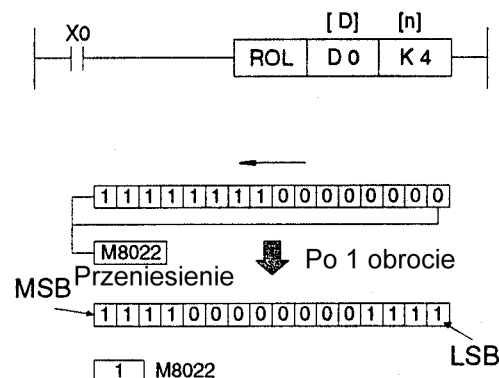
Przykład po lewej jest oparty na omówionej wyżej instrukcji, gdzie wzorec bitowy reprezentuje zawartość D0.

5.4.2 ROL (FNC 31)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D	n	
ROL FNC 31 (rotacja w lewo) →	wzorec bitowy urządzenia docelowego jest rotowany (obracany) o 'n' miejsc w lewo	KnY, KnM, KnS, T, C, D, V, Z uwaga: operacje 16-bitowe Kn=K4, operacje 32-bitowe Kn=K8	K, H, ☒ uwaga: operacja 16-bitowe n ≤ 16 operacja 32-bitowa n ≤ 32	ROL, ROLP: 5 kroków DROL, DROLP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Wzorec bitowy urządzenia docelowego jest rotowany (obracany) o n miejsc w lewo.

Stan ostatniego przesuniętego bitu jest kopiowany do znacznika przeniesienia M8022.

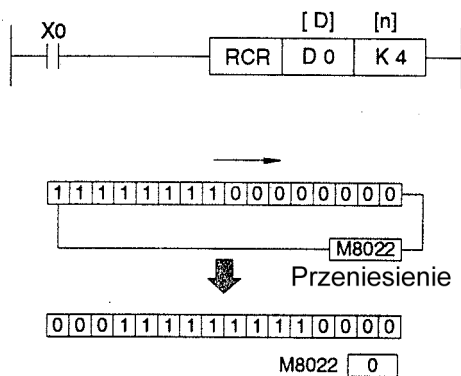
Przykład po lewej jest oparty na omówionej wyżej instrukcji, gdzie wzorec bitowy reprezentuje zawartość D0.

5.4.3 RCR (FNC 32)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D	n	
RCR FNC 32 (rotacja w prawo z przeniesieniem) →	wzorec bitowy jest rotowany (obracany) w prawo z 1 bitem wstawionym do znacznika przeniesienia	KnY, KnM, KnS, T, C, D, V, Z uwaga: operacje 16-bitowe Kn=K4, operacje 32-bitowe Kn=K8	K, H, ☒ uwaga: operacja 16-bitowe n ≤ 16 operacja 32-bitowa n ≤ 32	RCR, RCRP: 5 kroków DRCR, DRCRP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Wzorec bitowy urządzenia docelowego jest rotowany (obracany) o n miejsc w prawo.

Stan ostatniego przesuniętego bitu jest przesuwany do znacznika przeniesienia M8022. W następnej operacji tej instrukcji zawartość znacznika M8022 jest pierwszym bitem, który ma być przesunięty z powrotem do urządzenia docelowego.

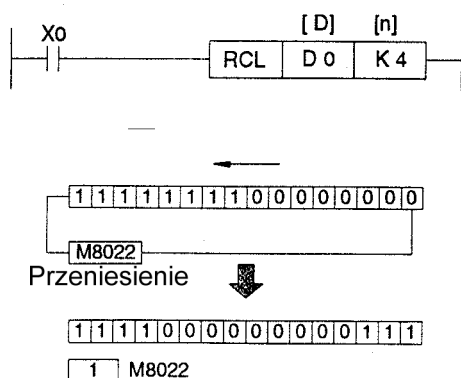
Przykład po lewej jest oparty na omówionej wyżej instrukcji, gdzie wzorec bitowy reprezentuje zawartość D0.

5.4.4 RCL (FNC 33)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D	n	
RCL FNC 33 (rotacja w lewo z przeniesieniem) →	wzorec bitowy jest rotowany (obracany) w lewo z 1 bitem wstawionym do znacznika przeniesienia	KnY, KnM, KnS, T, C, D, V, Z uwaga: operacje 16-bitowe Kn=K4, operacje 32-bitowe Kn=K8	K, H, ☒ uwaga: operacja 16-bitowe n ≤ 16 operacja 32-bitowa n ≤ 32	RCL, RCLP: 5 kroków DRCL, DRCLP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Wzorec bitowy urządzenia docelowego jest rotowany (obracany) o n miejsc w lewo.

Stan ostatniego przesuniętego bitu jest przesuwany do znacznika przeniesienia M8022. W następnej operacji tej instrukcji zawartość znacznika M8022 jest pierwszym bitem, który ma być przesunięty z powrotem do urządzenia docelowego.

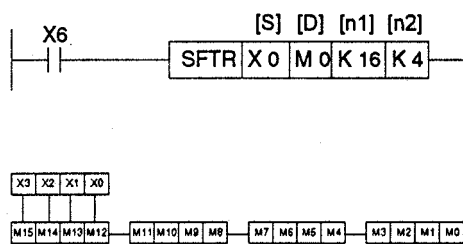
Przykład po lewej jest oparty na omówionej wyżej instrukcji, gdzie wzorec bitowy reprezentuje zawartość D0.

5.4.5 SFTR (FNC 34)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D	n ₁	n ₂	
SFTR FNC 34 (bitowe przesunięcie w prawo) →	stan urządzenia źródłowego jest kopiowany do stosu bitowego z przesunięciem danych w prawo	X, Y, M, S	Y, M, S	K, H, ☒ użytkownicy FX: n ₂ ≤ n ≤ 1024 użytkownicy FX ₀ , FX _{0N} n ₂ ≤ n ≤ 512		SFTR, SFTRP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja kopiuje n₂ urządzeń źródłowych na stos bitowy o długości n₁. Przy każdym nowym dodaniu n₂ bitów, istniejące dane w stosie bitowym są przesuwane o n₂ bitów w prawo. Każda dana bitowa przesuwana na pozycję powyżej zakresu n₁ jest tracona.

Operacja przesuwania bitów dokonuje się za każdym razem wykonywania instrukcji, chyba że instrukcja ta jest

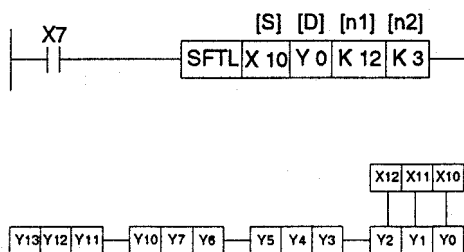
wyzwalana impulsowo lub ma ustawianą blokadę.

5.4.6 SFTL (FNC 35)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D	n ₁	n ₂	
SFTL FNC 35 (bitowe przesunięcie w lewo) →	stan urządzenia źródłowego jest kopiowany do stosu bitowego z przesunięciem danych w lewo	X, Y, M, S	Y, M, S	K, H, ☒ użytkownicy FX: n ₂ ≤ n ≤ 1024 użytkownicy FX ₀ , FX _{0N} n ₂ ≤ n ≤ 512		SFTL, SFTLP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja kopiuje n₂ urządzeń źródłowych na stos bitowy o długości n₁. Przy każdym nowym dodaniu n₂ bitów, istniejące dane w stosie bitowym są przesuwane o n₂ bitów w lewo. Każda dana bitowa przesuwana na pozycję powyżej zakresu n₁ jest tracona.

Operacja przesuwania bitów dokonuje się za każdym razem wykonywania instrukcji, chyba że instrukcja ta jest wyzwalana

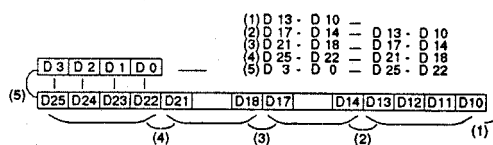
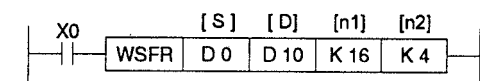
impulsowo lub ma ustawianą blokadę.

5.4.7 WSFR (FNC 36)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D	n ₁	n ₂	
WSFR FNC 36 (słowne przesunięcie w prawo) →	wartość urządzenia źródłowego jest kopiowana do stosu słownego z przesunięciem istniejących danych w prawo	KnX, KnY, KnM, KnS, T, C, D	KnY, KnM, KnS, T, C, D	K, H, ☒ użytkownicy FX: n ₂ ≤ n ≤ 512		WSFR, WSFRP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja kopiuje n₂ urządzeń źródłowych na stos słowny o długości n₁. Przy każdym nowym dodaniu n₂ słów, istniejące dane w stosie słownym są przesuwane o n₂ słów w prawo. Każda dana słowna przesuwana na pozycję powyżej zakresu n₁ jest tracona.

Operacja przesuwania słów dokonuje się za każdym razem wykonywania instrukcji, chyba że instrukcja ta jest wyzwalana impulsowo lub ma ustawianą blokadę.

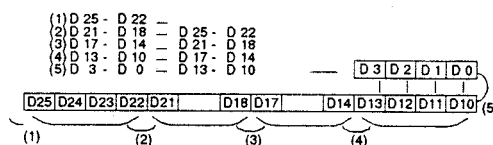
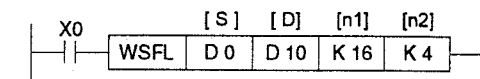
Uwaga: Podczas używania urządzeń bitowych jako źródła (S) i przeznaczenia (D), wartości Kn muszą być równe.

5.4.8 WSFL (FNC 37)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D	n ₁	n ₂	
WSFL FNC 37 (słowne przesunięcie w lewo) →	wartość urządzenia źródłowego jest kopiowana do stosu słownego z przesunięciem istniejących danych w lewo	KnX, KnY, KnM, KnS, T, C, D	KnY, KnM, KnS, T, C, D	K, H, ☒ użytkownicy FX: n ₂ ≤ n ≤ 512		WSFL, WSFLP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja kopiuje n₂ urządzeń źródłowych na stos słowny o długości n₁. Przy każdym nowym dodaniu n₂ słów, istniejące dane w stosie słownym są przesuwane o n₂ słów w lewo. Każda dana słowna przesuwana na pozycję powyżej zakresu n₁ jest tracona.

Operacja przesuwania słów dokonuje się za każdym razem wykonywania instrukcji, chyba że instrukcja ta jest wyzwalana impulsowo lub ma ustawianą blokadę.

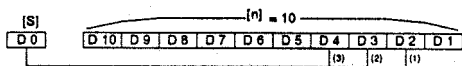
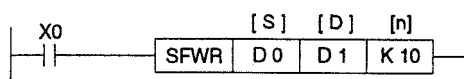
Uwaga: Podczas używania urządzeń bitowych jako źródła (S) i przeznaczenia (D), wartości Kn muszą być równe.

5.4.9 SFWR (FNC 38)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
SFWR FNC 38 (zapis z przesunięciem) →	instrukcja tworzy i rozbudowuje stos FIFO o długości n – musi być użyta z instrukcją SFRD FNC 39	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D	K, H, ☒ 2 ≤ n ≤ 512	SFWR, SFWRP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Zawartości urządzenia źródłowego (S) są zapisane do stosu FIFO. Pozycja wstawiania do stosu jest automatycznie obliczana przez sterownik.

Urządzenie docelowe (D) jest adresem początkowym stosu FIFO. Zawartości D pokazują, gdzie będzie przechowany następny zapis (jako offset z D+1). Jeżeli zawartości D przekroczą wartość "n-1" (n jest długością stosu FIFO), wtedy wstawienie do stosu jest wstrzymane. Znacznik przeniesienia M8022 jest ustawiony w stan ON aby

zasygnalizować tę sytuację.

Uwagi:

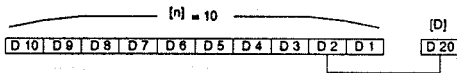
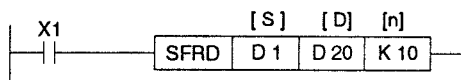
- FIFO jest skrótem od "First In/First Out" - „Pierwszy wszedł/pierwszy wyszedł”
- Chociaż do stosu FIFO jest przydzielonych n urządzeń, tylko n-1 informacje mogą być wpisane do stosu. Tak się dzieje, ponieważ adres początkowy urządzenia (D) wykorzystuje pierwszy dostępny rejestr do przechowania informacji o punkcie wstawienia następnej danej do stosu FIFO.
- Przed użyciem stosu FIFO należy się upewnić, że zawartości rejestru adresu początkowego (D) są równe "0".
- Ta instrukcja powinna być używana w połączeniu z instrukcją SFRD (FNC 39). Parametr n w obu instrukcjach powinien być taki sam.

5.4.10 SFRD (FNC 39)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
SFWL FNC 39 (odczyt z przesunięciem) →	instrukcja czyta i redukuje stos FIFO – musi być użyta z instrukcją SFWR FNC 38	KnY, KnM, KnS, T, C, D	KnY, KnM, KnS, T, C, D, V, Z	K, H, ☒ 2 ≤ n ≤ 512	SFRD, SFRDP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Urządzenie źródłowe (S) określa adres początkowy stosu FIFO. Jego zawartość odzwierciedla adres ostatnio wprowadzonej danej do stosu FIFO, tzn. określa szczyt FIFO (bieżącą pozycję).

Instrukcja czyta pierwszą daną ze stosu FIFO (rejestr S+1), przenosi wszystkie dane w stosie w górę o jedna pozycję, żeby wypełnić czytany obszar i dekrementuje zawartość adresu początkowego FIFO (S) o 1. Przeczytana dana jest wpisywana do urządzenia

docelowego (D).

Kiedy zawartości urządzenia źródłowego (S) równe są "0", tzn. stos FIFO jest pusty, znacznik M8022 jest ustawiony w stan ON.

Uwagi:

- FIFO jest skrótem od "First In/First Out"
- Tylko n-1 danych może być odczytanych ze stosu FIFO. Tak się dzieje, ponieważ stos wymaga aby pierwszy rejestr, zawierający adres początkowy stosu (S), zawierał informację o bieżącej długości stosu FIFO.
- Ta instrukcja będzie zawsze czytała dane źródłowe z rejestru S₊₁.
- Ta instrukcja powinna być używana w połączeniu z instrukcją SFWR (FNC 38). Parametr „n” w obu instrukcjach powinien być taki sam.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------



1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.5 Operacje na danych - funkcje 40 - 49

Zawartość:

			Strona
ZRST	zerowanie zakresu	FNC 40	5-43
DECO	dekodowanie	FNC 41	5-43
ENCO	kodowanie	FNC 42	5-44
SUM	suma aktywnych bitów	FNC 43	5-45
BON	stan określonego bitu	FNC 44	5-45
MEAN	średnia	FNC 45	5-46
ANS	ustawienie annunciator	FNC 46	5-47
ANR	zerowanie annunciator	FNC 47	5-47
SQR	pierwiastek kwadratowy	FNC 48	5-48
FLT	konwersja formatów zmiennoprzecinkowych	FNC 49	5-49



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

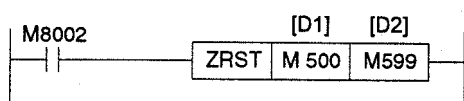
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.5.1 ZRST (FNC 40)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D ₁	D ₂	
ZRST FNC 40 (zerowanie zakresu)	zeruje zakres podanych urządzeń	Y, M, S, T, C, D Uwaga: D1 musi być mniejsze lub równe D2. Liczniki standardowe i szybkie nie mogą być używane jednocześnie		ZRST, ZRSTP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie: Zakres urządzeń, włącznie z tymi, określonymi jako docelowe, jest zerowany, tzn. dla urządzeń danych wartość bieżąca jest ustawiana na "0", a dla elementów bitowych urządzenia są ustawiane w stan OFF, tzn.

również wyzerowane. Określony zakres urządzeń nie może zawierać urządzeń różnych typów, tzn. C000 określony jako pierwsze urządzenie docelowe (D1) nie może być w parze z T199, jako drugim urządzeniem docelowym (D2). Podczas zerowania liczniki standardowe i liczniki szybkie nie mogą być zerowane jako część tego samego zakresu.

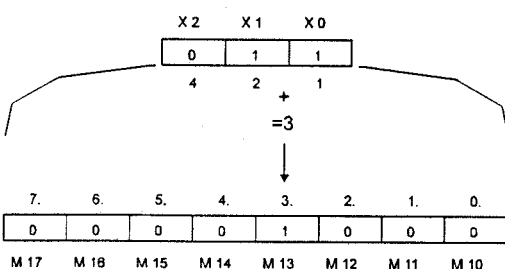
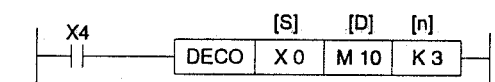
Jeżeli D1 jest większe niż D2 wtedy tylko urządzenie D1 jest zerowane.

5.5.2 DECO (FNC 41)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
DECO FNC 41 (dekodowanie)	wartość Q danej źródłowej identyfikuje Q-ty bit urządzenie docelowego, ustawiany w stan ON	K, H, X, Y, M, S, T, C, D, V, Z	Y, M, S, T, C, D	K, H, <input checked="" type="checkbox"/> Uwaga: D=Y,M,S to zakres n=1-8 D=T,C,D to zakres n=1-4 n=0 to bez przetwarzania	DECO, DECOP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



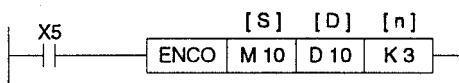
Działanie: Dane źródłowe są kombinacją argumentów S i n, gdzie S określa adres początkowy danej źródłowej a n określa liczbę kolejnych bitów źródłowych, które będą dekodowane. Dane źródłowe są czytane jako pojedyncza liczba, która w wyniku dekodowania wyznacza numer bitu w urządzeniu docelowym (D), które będzie ustawione w stan ON (zobacz przykład obok). Kiedy urządzenie docelowe jest urządzeniem danych, n musi być z zakresu od 1 do 4, ponieważ istnieje tylko 16 dostępnych bitów docelowych w pojedynczym słowie. Wszystkie nieużywane bity danych w słowie są wyzerowane.

5.5.3 ENCO (FNC 42)

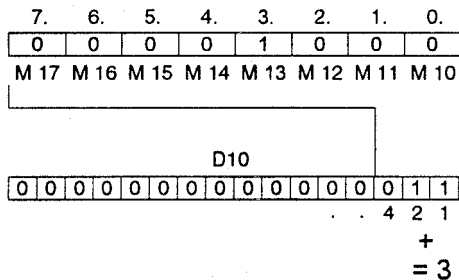
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
ENCO FNC 42 (kodowanie)	pozycja najstarszego aktywnego bitu jest przechowana jako wartość numeryczna liczona od adresu początkowego	X, Y, M, S, T, C, D, V, Z	T, C, D, V, Z	K, H, ☒ Uwaga: S=X,Y,M to zakres n=1-8 S=T,C,D to zakres n=1-4 n=0 to bez przetwarzania	ENCO, ENCO P: 7 kroków

Instrukcja impulsowa (P)				Operacje 16-bitowe				Operacje 32-bitowe						
FX _{0(S)}	FX _{0N}	FX	FX _{2C}	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _{2C}	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _{2C}	FX _{2N}

**Działanie:**

Funkcja odwrotna do DECO (FNC 41). Numer najwyższego bitu w czytany zakresie ustawiony na ON jest wyznaczany jako offset źródłowego adresu początkowego (S). Numer ten jest wpisywany do rejestru docelowego (D).

**Uwagi:**

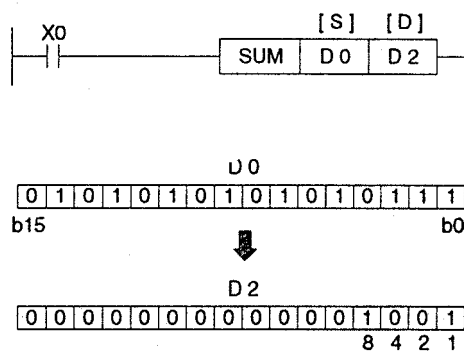
- Czytany zakres jest zdefiniowany przez największą liczbę przechowaną w formacie dwójkowym w liczbie przechowanych bitów przeznaczenia określonych przez "n", tzn. jeżeli n równa się 4 bity, w urządzeniu docelowym może być zapisana maksymalna liczba z zakresu 0-15. Stąd, jeżeli jako dana źródłowa było użyte urządzenie bitowe, będzie użyte 16-bitowe urządzenie, tzn. początkowe urządzenie bitowe i 15 kolejnych urządzeń.
- Jeżeli przechowana liczba docelowa jest równa "0", wtedy bit adresu początkowego źródła jest w stanie ON. tzn. aktywny bit ma zerowy offset adresu początkowego. Jednak, jeżeli żaden bit w danych źródłowych nie jest w stanie ON, do urządzenia docelowego jest wpisane 0 i generowany jest błąd.
- Kiedy urządzenie źródłowe jest daną albo urządzeniem słownym, n musi być poza zakresem od 1 do 4, ponieważ osiągalnych jest tylko 16 bitów źródłowych w pojedynczym słowie danych.

5.5.4 SUM (FNC 43)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
SUM FNC 43 (suma aktywnych bitów)	liczba aktywnych bitów w urządzeniu źródłowym jest zapisana w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, Y, Z	KnY, KnM, KnS, T, C, D, V, Z	SUM, SUMP: 7 kroków DSUM, DSUMP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Przenies. M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

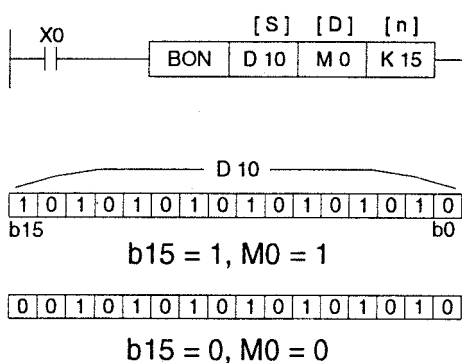
Ilość aktywnych (będących w stanie ON) bitów w urządzeniu źródłowym (S), tzn. bitów, które mają wartość "1", jest zliczona. Wynik jest zapamiętany w rejestrze docelowym (D). Jeżeli jest użyty format podwójnego słowa, zarówno źródło, jak i urządzenie docelowe używają 32-bitowych podwójnych rejestrów. Urządzenie docelowe będzie zawsze miało swoje górne 16 bitów wyzerowane, ponieważ zliczona wartość nigdy nie przekroczy 32. Jeśli żadne bity nie są w stanie ON, wtedy znacznik zera, M8020, jest ustawiony w stan ON.

5.5.5 BON (FNC 44)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
BON FNC 44 (stan określonego bitu)	status określonego bitu urządzenia źródłowego jest zapamiętywany w urządzeniu docelowym	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y, M, S	K, H, <input checked="" type="checkbox"/>	BON, BONP: 7 kroków DBON, DBONP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie: Pozycja pojedynczego bitu (n) jest wyspecyfikowana w urządzeniu/obszarze źródłowym (S). Wartość "n" może być uważana jako offset źródłowego adresu początkowego (S), tzn. 0 jest pierwszym urządzeniem (offset 0), podczas gdy offset 15 będzie urządzeniem 16.

Jeżeli wyspecyfikowany bit stanie się aktywny, tzn. będzie w stanie ON, urządzenie docelowe (D) jest również ustawiane w stan ON i odwrotnie.

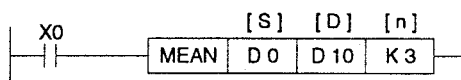
Można powiedzieć, że urządzenie docelowe jest lustrzanym odbiciem stanu wybranego bitu źródłowego.

5.5.6 MEAN (FNC 45)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
MEAN FNC 45 (średnia)	oblicza średnią z zakresu urządzeń	KnX, KnY, KnM, KnS, T, C, D	KnY, KnM, KnS, T, C, D, V, Z	K, H, ☒ uwaga: n = 1 do 64	MEAN, MEANP: 7 kroków DMEAN, DMEANP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Reguła ogólna

$$D = \frac{\sum_{s_0}^{s_n} S}{n} = \frac{(S_0 + S_1 + \dots + S_n)}{n}$$

Przykład

$$D10 = \frac{(D0) + (D1) + (D2)}{3}$$

Działanie:

Zakres danych źródłowych jest zdefiniowany argumentami S i n. S jest adresem początkowym danej źródłowej a n określa liczbę kolejno używanych urządzeń źródłowych. Wartość wszystkich urządzeń w zakresie źródłowym jest sumowana, a potem dzielona przez liczbę zsumowanych urządzeń, tzn. "n". Uzyskuje się wartość średnią (całkowitą), która jest przechowywana w urządzeniu docelowym (D). Reszta z obliczonej wartości średniej jest pomijana.

Uwagi:

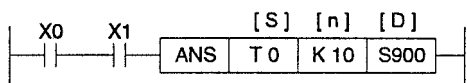
Jeżeli określony obszar źródłowy jest mniejszy, niż obszar fizycznie dostępny, wtedy używane są tylko dostępne urządzenia. Bieżąca wartość n, użyta do wyliczenia średniej, odzwierciedla użyte, dostępne urządzenia. Jednak wartość n, która została wprowadzona do instrukcji, będzie wciąż wyświetlana. To może spowodować zamieszanie, jako że średnia wartość, wyliczona ręcznie przy wykorzystaniu początkowej wartości n będzie inna od tej, która jest wyświetlana. Jeżeli wartość n jest określona poza ustalonym zakresem (1-64), generowany jest błąd.

5.5.7 ANS (FNC 46)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
ANS FNC 46 (ustawianie annunciatora)	instrukcja uruchamia timer, po zakończeniu cyklu timera wybrany annunciator ustawiany jest w stan ON	T uwaga: dozwolony zakres: T0 do T199	S uwaga: zakres annunciatorów: S900 do S999	K ☒ uwaga: zakres n 1 do 32767 – w jednostkach 100 ms	ANS: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Ta, instrukcja uruchamia timer (S) dla n kroków po 100 ms. Kiedy timer kończy swój cykl, przyporządkowany wskaźnik (annunciator, D) jest ustawiony w stan ON. Jeżeli

instrukcja jest przełączona w stan OFF podczas lub po ukończeniu cyklu, timer jest automatycznie zerowany. Jednak stan aktualny wskaźnika (cewki annunciatora) pozostaje niezmieniony.

**Uwaga:**

To jest tylko jedna z metod sterowania cewkami annunciatorów. Inne, takie jak bezpośrednie sterowanie, także może być użyte.

5.5.8 ANR (FNC 47)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		D	
ANR FNC 47 (zerowanie annunciatora)	najmłodszy aktywny annunciator jest zerowany przy każdym wykonaniu instrukcji	brak	ANR, ANRP: 1 krok

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Annunciatory, które zostały aktywowane, są kolejno zerowane (jeden za drugim) za każdym razem, kiedy działa instrukcja ANR. Jeżeli instrukcja ANR jest

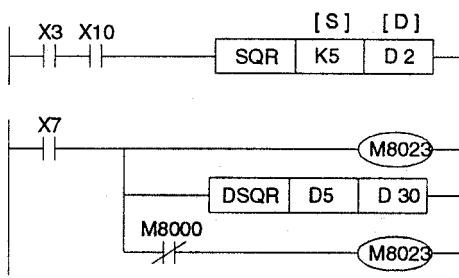
wykonywana w sposób ciągły, będzie wykonywała operacje zerowania przy każdym przebiegu programu, chyba że zostanie wyzwolona impulsowo (przedrostek P) lub przy użyciu programowej blokady.

5.5.9 SQR (FNC 48)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
SQR FNC 48 (pierwiastek kwadratowy)	oblicza pierwiastek kwadratowy, np. $D = \sqrt{S}$	K, H, D	D	SQR, SQRP: 5 kroków DSQR, DSQRP: 9 kroków
		jeśli używany jest znacznik M8023, mogą być wykorzystane tylko podwójne słowa. Więcej szczegółów na temat formatów zmiennoprzecinkowych na stronie 4-46		

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	Zero M8020 Pożyczka M8021
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie 1:**

Ta instrukcja wykonuje operację pierwiastka kwadratowego na danej źródłowej (S) i przechowuje wynik w urządzeniu docelowym (D). Operacja jest przeprowadzona w całości na liczbach całkowitych, zaokrąglając pierwiastek do najniższej liczby całkowitej. Na przykład, jeżeli (S) = 154, to (D) jest równe 12. Znacznik M8020 jest ustawiony w stan ON, kiedy wynik pierwiastkowania jest równy 0. Wyniki zaokrąglone uaktywniają znacznik M8021.

Działanie 2:

Ta funkcja jest równoważna funkcji ESQR (FNC 127).

To działanie jest podobne do "Działania 1". Jednak jest aktywowana tylko wtedy, kiedy używany jest znacznik trybu zmiennoprzecinkowego, M8023. To wtedy umożliwia instrukcji SQR przetworzenie wyników w formacie zmiennoprzecinkowym. Dane źródłowe (S) muszą wcześniej być dostarczone w formacie zmiennoprzecinkowym do rejestru danych, lub mogą być dostarczone jako stała (K/H). Jeżeli jako źródła użyte są stałe, są one automatycznie konwertowane do formatu zmiennoprzecinkowego. Działanie 2 jest ważne tylko dla operacji na podwójnych słowach (32-bitowych), stąd zarówno (S) i (D) będą wartościami 32-bitowymi, a instrukcja SQR będzie wprowadzona jako DSQR lub DSQRP.

**Uwaga ogólna:**

Wykonywanie operacji pierwiastkowania kwadratowego (nawet na kalkulatorze) na liczbach ujemnych daje błędny wynik. Jest to wykazywane przez uaktywnienie specjalnego znacznika M8067.

$$\sqrt{-168} = \text{błąd i M8067 będzie ustawiony w stan ON}$$

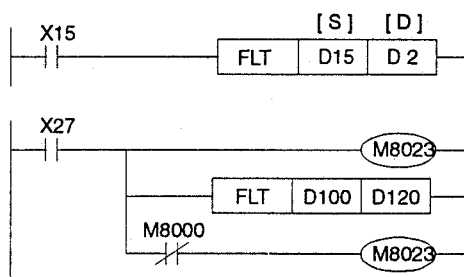
Obowiązuje to dla obu trybów operacji.

5.5.10 FLT (FNC 49)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
FLT FNC 49 (zmienny przecinek)	konwertuje dane na/z format zmiennoprzecinkowy	D	D	FLT, FLTP: 5 kroków DFLT, DFLTP: 9 kroków
		M8023=OFF dane są konwertowane z postaci dziesiętnej na zmiennoprzecinkową M8023=ON dane są konwertowane z postaci zmiennoprzecinkowej na dziesiętną		

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie 1:**

Kiedy instrukcja zmiennoprzecinkowa używana jest, gdy znacznik M8023 jest w stanie OFF, dana źródłowa (S) jest konwertowana do równoważnej wartości, przechowanej w formacie zmiennoprzecinkowym w urządzeniu docelowym (D).

Należy zauważyć, że dwa kolejne urządzenia (D i D₊₁) będą użyte do przechowania przekonwertowanej liczby zmiennoprzecinkowej. Niezależnie od rozmiaru danej

źródłowej (S) nie ma to wpływu na liczbę urządzeń docelowych (D), użytych do przechowania liczby zmiennoprzecinkowej.

Przykłady:

Dziesiętna dana źródłowa	Zmiennoprzecinkowa dana docelowa
1	1
-26700	$-2,67 \times 10^4$
404	$4,40 \times 10^2$

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Działanie 2: (Dostępne w jednostkach FX_(2C)) Ta funkcja jest równoważna funkcji INT (FNC 129). Kiedy wykonywana jest instrukcja zmiennoprzecinkowa i znacznik M8023 jest w stanie ON, operacja będzie przeprowadzana odwrotnie do "Działania 1". Każda liczba w formacie zmiennoprzecinkowym, przechowana w źródle (S) będzie konwertowana do swojego dziesiętnego odpowiednika i przechowana w urządzeniu docelowym (D).

Ciąg dalszy na stronie następnej...

Uwagi:

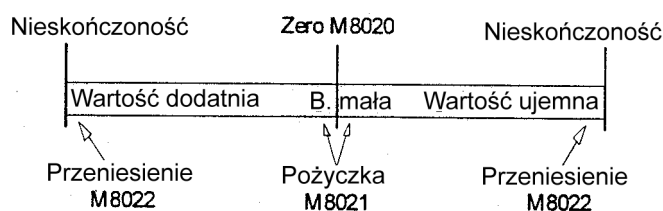
a) Kiedy użyte są liczby zmiennoprzecinkowe, znaczniki zera, pożyczki i przeniesienia (odpowiednio M8020, M8021 i M8022) działają w następujący sposób:

M8020 - Zero: znacznik jest aktywowany, kiedy wynik jest zerowy.

M8021 - Pożyczka: znacznik jest aktywowany, kiedy wynik jest mniejszy od najmniejszej możliwej liczby. Wymuszony jest wynik równy najmniejszej liczbie, a znacznik ustawiany jest w stan ON.

M8022 - Przeniesienie: znacznik jest aktywowany, gdy wynik jest większy od największej możliwej liczby. Wymuszony jest wynik równy największej liczbie, a znacznik ustawiany jest w stan ON.

b) Liczby zmiennoprzecinkowe zawsze zajmują 32 kolejne bity, tzn. 2 kolejne rejestry danych. Podczas konwersji pomiędzy liczbami zmiennoprzecinkowymi i dziesiętymi należy zapewnić wystarczającą ilość urządzeń docelowych, tzn.



Instrukcja	Operacja na podwójnym słowie	Stan M8023	Liczba rejestrów źródłowych	Liczba rejestrów docelowych	Komentarz
FLT	NIE	OFF	1(S)	2(D, D ₊₁)	konwersja na zmienny przec.
FLT (INT)		ON	2 (S, S ₊₁)	1 (D)	konwersja na dziesiętną
DFLT	TAK	OFF	2 (S, S ₊₁)	2(D, D ₊₁)	konwersja na zmienny przec.
DFLT (DINT)		ON	2 (S, S ₊₁)	2(D, D ₊₁)	konwersja na dziesiętną




Uwaga końcowa:

Więcej informacji na temat notacji zmiennoprzecinkowej i naukowej w rozdziale 4, Zaawansowane Urządzenia, strona 4-46.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

	1. Sterowanie przebiegiem programu	5-4
	2. Instrukcje przesłania i porównania	5-16
	3. Operacje arytmetyczne i logiczne	5-24
	4. Rotacje i funkcje przesunięcia	5-34
	5. Operacje na danych	5-42
	6. Szybkie przetwarzanie	5-52
	7. Instrukcje podręczne	5-66
	8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
	9. Zewnętrzne urządzenia szeregowo	5-94
	10. Zewnętrzne jednostki F2	5-112
	11. Operacje zmiennoprzecinkowe 1 & 2	5-120
	12. Operacje trygonometryczne	5-128
	13. Operacje na danych 2	5-132
	14. Sterowanie zegarem czasu rzeczywistego	5-136
	15. Kody Graya	5-144
	16. Porównania operacyjne	5-148

5.6 Szybkie przetwarzanie - funkcje 50 - 59

Zawartość:

			Strona
REF	odświeżanie	FNC 50	5-53
REFF	odświeżanie i dopasowanie filtra	FNC 51	5-53
MTR	matryca wejściowa	FNC 52	5-54
HSCS	ustawianie licznika szybkiego	FNC 53	5-55
HSCR	zerowanie licznika szybkiego	FNC 54	5-56
HSZ	porównanie zakresu licznika szybkiego	FNC 55	5-57
SPD	detekcja szybkości	FNC 56	5-60
PLSY	wyjście impulsowe	FNC 57	5-61
PWM	modulacja szerokości impulsu	FNC 58	5-62
PLSR	wyjście impulsu schodkowego	FNC 59	5-63



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1 , S_3 lub dla urządzeń stabilizowanych D_{3+0} , S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

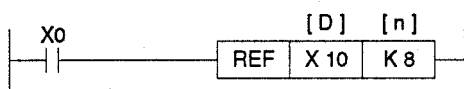
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.6.1 REF (FNC 50)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		D	n	
REF FNC 50 (odświeżanie)	wymusza natychmiastową aktualizację wyspecyfikowanych wejść lub wyjść	X, Y ☒ uwaga: D powinno być zawsze wielokrotnością 10, tzn. 00, 10, 20, 30 itd.	K, H ☒ uwaga: n powinno być zawsze wielokrotnością 8, tzn. 8, 16, 24, 32, itd.	REF, REFP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie: Standardowa operacja sterownika: aktualizuje stany wyjść i wejść pomiędzy instrukcją END jednego przebiegu programu i krokiem zerowym następnego przebiegu programu. Instrukcji REF używa się wówczas, gdy wymagana jest natychmiastowa aktualizacja stanu urządzeń wejścia/wyjścia. Instrukcja może być użyta tylko do

aktualizacji lub odświeżenia bloków zawierających po 8 (n) kolejnych urządzeń. Adres początkowy urządzeń odświeżanych powinien mieć zawsze ostatnią cyfrę równą 0.

**Uwaga:**

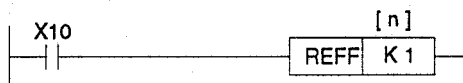
Zanim urządzenie wejścia/wyjścia zostanie fizycznie uaktualnione, pojawi się krótkie opóźnienie, w przypadku wejść będzie to czas równy ustawieniu filtra, natomiast wyjścia będą opóźnione o czas ich pobudzenia.

5.6.2 REFF (FNC 51)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		n		
REFF FNC 51 (odświeżanie i dopasowanie filtra)	wejścia są odświeżane i ustawiane są nowe wartości stałych czasowych filtrów	K, H, ☒ uwaga: n=0 do 60 ms (0=50µs) X000 do X007 (X000 do X017 dla FX _{2N}) są automatycznie przyporządkowywane podczas użycia tej instrukcji		REFF, REFFP: 3 kroki

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Aby uniknąć problemów związanych z mechanicznym odbiciem styków, sterowniki są wyposażone w filtry wejściowe. Ponieważ wymaga to upewnienia się, czy stały sygnał wejściowy został odebrany w niezmiennym okresie czasu, stosowanie filtrów wejściowych

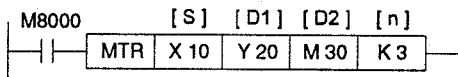
spowalnia czas reakcji sterownika. Dla bardzo szybkich zadań, szczególnie gdy sygnał wejściowy jest dostarczany przez przełączniki półprzewodnikowe, czasy filtrów mogą zostać zredukowane. Domyślne ustawienie dla filtrów wejściowych wynosi ok. 10 ms. Używając tej instrukcji czasy te mogą być wybrane z przedziału 0-60 ms. Ustawienie 0 odpowiada 50 µs, tzn. minimalne dostępne ustawienie. Jest ono wybierane automatycznie, jeśli używane są wejścia bezpośrednie, przerwania lub funkcje szybkiego zliczania. Aby instrukcja REFF działała efektywnie, musi być uruchamiana dla każdego przebiegu programu, w przeciwnym razie zostaną użyte standardowe (10 ms) czasy filtrów wejściowych.

5.6.3 MTR (FNC 52)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D ₁	D ₂	n	
MTR FNC 52 (matryca wejściowa)	multipleksuje bank wejść na zbiór urządzeń. Może być użyta TYLKO RAZ	X ☒	Y ☒	Y, M, S ☒	K, H, ☒	MTR: 9 kroków
uwaga: te argumenty muszą być zawsze wielokrotnością 10, tzn. 00, 10, 20, 30, itd.					uwaga: n=2 do 8	

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					Znaczniki	zakończ. operacji M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Ta instrukcja umożliwia cykliczny odczyt matrycy sygnalizacyjnej składającej się z (n) banków (słów) o długości 8-bitów. Wprowadzanie każdego banku odbywa się poprzez 8 kolejnych wejść X sterownika (S - adres pierwszego wejścia). Wybór kolejnego banku odbywa się kolejnym wyjściem Y sterownika (D₁ - adres pierwszego wyjścia wyboru banku). Wprowadzane banki umieszczane są w macierzy binarnej deklarowanej w D₂.

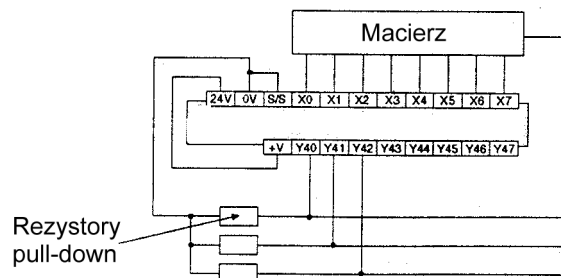
Uwagi:

- Instrukcja MTR wymaga szybkiego przełączania wejść/wyjść. Z tego powodu możliwe jest użycie tej instrukcji tylko w modułach z wyjściami tranzystorowymi.
- Aby instrukcja MTR działała poprawnie, musi być aktywowana w sposób ciągły. Zaleca się użycie znacznika M8000 stanu RUN sterownika. Po ukończeniu pierwszego pełnego odczytu matrycy, znacznik ukończenia operacji M8029 jest ustawiany w stan ON. Znacznik ten jest automatycznie zerowany, kiedy instrukcja MTR jest deaktywowana.
- Każdy odczytany zbiór 8 sygnałów wejściowych nazywany jest "bankiem"; istnieje "n" banków.
- Każdy bank jest wybierany przez dedykowane wyjście (adres początkowy D₁). Oznacza to, że liczba wyjść wg D₁, użytych jako sygnały wyboru banku, równa się liczbie banków n. Ponieważ matryca wprowadza „dodatkowe fizyczne wejścia” do sterownika, dlatego stan każdego z nich musi być zapamiętany. Jest on przechowywany w macierzy binarnej. Macierz zaczyna się od adresu początkowego D₂. Konfiguracja macierzy odpowiada konstrukcji matrycy: 8 bitów * n banków. Stąd też, kiedy ma być odczytane jakieś wejście w wybranym banku matrycy, jego stan należy odczytać z równoważnej pozycji w macierzy.
- Instrukcja działa pod wpływem przerwania czasowego, odczytując każdy bank wejściowy co 20 ms. Ten czas wynika ze stałej filtrów wejściowych, ustawionych na 10 ms. Oznacza to, że matryca 8 banków, tzn. 64 wejść (8 wejść x 8 banków), jest czytana przez 160 ms.



Jeżeli wejście szybkie (np. X0) jest wyspecyfikowane dla argumentu S, czas czytania każdego banku wynosi tylko 10 ms., tzn. czas czytania skraca się dwukrotnie.

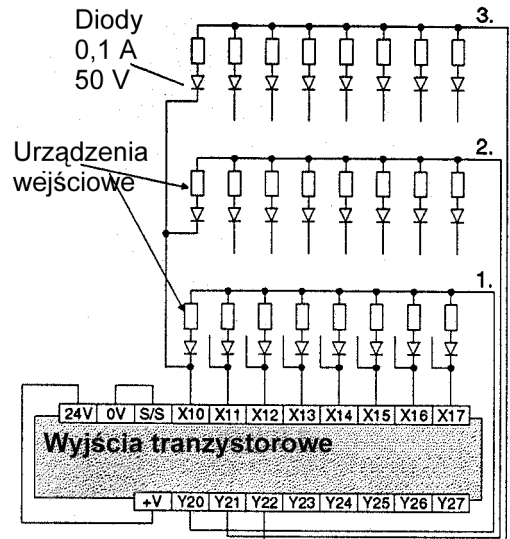
Jednakże wtedy wymagane są dodatkowe rezystory, przyspieszające wyłączenie wyjść sterujących, aby zapobiec nakładaniu się prądów w obwodach matrycy pozostałych z poprzedniej operacji wyboru. Te rezystory powinny być umieszczone równolegle do banków wejściowych i mieć wartość ok. 3,3 kohm/0,5 W. W celu uniknięcia kłopotów z instalacją rezystorów, szybkie wejścia nie powinny być specyfikowane w S.



f) Ponieważ ta instrukcja używa serii multipleksowanych sygnałów, wymaga ona pewnej ilości okablowania tworzącego matrycę sygnalizacyjną. Rysunek po prawej stronie pokazuje jej strukturę dla przypadku instrukcji według powyższego przykładu. Aby operacja odczytu styku wejściowego była poprawna, w szereg z każdym stykiem wejściowym musi być umieszczona dioda o parametrach 0,1A/50V.

g) **Przykład działania:**

Kiedy wyjście Y20 jest w stanie ON, czytane są tylko wejścia pierwszego banku. Wyniki są następnie zapamiętane; w tym przykładzie w znacznikach M30-M37. Następnym krokiem jest ustawienie Y20 w stan OFF, a Y21 w stan ON. Tym razem czytane są tylko wejścia drugiego banku. Te wyniki są przechowane w urządzeniach M40-M47. Ostatnim krokiem tego przykładu jest ustawienie Y21 w stan OFF, a Y22 w stan ON. Tym razem czytane są wszystkie wejścia trzeciego banku i przechowane w urządzeniach M50-M57. Czas odczytu wszystkich banków zajmie $20 \times 3 = 60$ ms.



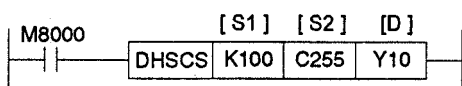
Należy pamiętać, że macierz wyników, zrealizowana na urządzeniach S lub M, nie używa urządzeń bitowych o numerach x8 i x9.

5.6.4 HSCS (FNC 53)

FX_{0(S)} FX_{0N} FX FX_(2C) FX_{2N}

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
HSCS FNC 53 (ustawianie licznika szybkiego)	ustawia wybrane wyjścia, gdy wartość specyfikowanego szybkiego licznika osiągnie wartość testową	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	C uwaga: C=235 do 254 lub dostępne szybkie liczniki	Y, M, S etykiety przerwania I010 do I060 mogą być ustawiane w FX z CPU ver. 3.07 i FX _{2C}	DHSCS: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie: Instrukcja HSCS porównuje bieżącą wartość wybranego szybkiego licznika S2 z wybraną wartością S1. Kiedy bieżąca wartość licznika zmieni się na wartość równą S1, urządzenie docelowe (D) jest ustawione w stan ON.

Przykład powyżej pokazuje, że Y10 będzie ustawione w stan ON tylko wtedy, gdy wartość C255 przejdzie z 99 na 100 lub ze 101 na 100. Jeżeli wartość bieżąca licznika została wymuszona na 100, wyjście Y10 nie zostanie ustawione w stan ON.

Uwagi:

a) Zaleca się, by wejściem dla funkcji szybkiego licznika HSCS, HSCR, HSCZ był znacznik M8000 stanu RUN.

- b) Jeżeli dla pojedynczego licznika użyta jest więcej niż jedna funkcja szybkiego licznika, wybrane urządzenia znaczników (D) powinny być utrzymane w grupie 8 urządzeń, np. Y0-Y7, M10-M17.
- c) Wszystkie funkcje szybkich liczników wykorzystują procesy przerwania, stąd wszystkie urządzenia docelowe D są natychmiast aktualizowane.

**Uwaga:**

Użytkownicy FX₀, FX_{0N} - dopuszcza się max. 4 jednocześnie aktywne instrukcje HSCS/R

Użytkownicy FX - dopuszcza się max. 6 jednocześnie aktywnych instrukcji HSCS/R i HSZ

Należy pamiętać, że użycie funkcji szybkich liczników ma bezpośredni wpływ na maksymalną dostępną szybkość liczenia. Więcej szczegółów na stronie 4-22.

Użycie wskaźników przerwań

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------



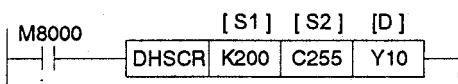
Sterowniki FX_(2C) i FX_{2N} mogą używać wskaźników przerwań I010-I060 (6 punktów) jako urządzeń docelowych (D). To umożliwia bezpośrednie wyzwolenie podprogramów obsługi przerwań, kiedy określony szybki licznik osiąga wartość podaną w instrukcji HSCS.

5.6.5 HSCR (FNC 54)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
HSCR FNC 54 (zerowanie licznika szybkiego)	zeruje wybrane wyjścia, gdy wartość specyfikowanego szybkiego licznika osiągnie wartość testową	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	C uwaga: C=235 do 254 lub dostępne szybkie liczniki	Y, M, S C uwaga: jeśli C, użyj tego samego licznika co w S ₂	DHSCR: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja HSCR porównuje bieżącą wartość wybranego szybkiego licznika z wybraną wartością S₁. Kiedy bieżąca wartość licznika zmienia się na wartość równą S₁,

urządzenie docelowe D jest zerowane. W przykładzie powyżej Y10 zostanie zresetowane, tylko wtedy, kiedy wartość C255 przejdzie ze 199 na 200 lub z 201 na 200. Jeżeli wartość bieżąca licznika została wymuszona na 200, wyjście Y10 **nie zostanie** wyzerowane.

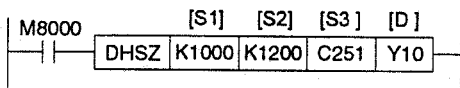
Więcej ogólnych uwag na temat funkcji szybkich liczników w sekcji "Uwagi" w opisie funkcji HSCS (FNC 53). Odnośne uwagi: a, b i c. Należy również zwrócić uwagę co do liczby dozwolonych szybkich instrukcji.

5.6.6 HSZ (FNC 55)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S ₁	S ₂	S ₃	D	
HSZ FNC 55 (porównanie zakresu licznika szybkiego)	Działanie 1: bieżąca wartość licznika szybkiego jest porównywana z określonym zakresem	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		C uwaga: C=235 do 254	Y, M, S C uwaga: użyte 3 kolejne urządzenia	DHSZ: 17 kroków
	Działanie 2: określony zakres jest przechowywany w tabeli danych sterując bezpośrednio wyjściami Y	D	K, H wartości od 1 do 128 (dziesiętnie)		M8130 (wyłącznie) ten znacznik może być użyty z jedną instrukcją DHSZ jednocześnie	
	Działanie 3: określony zakres jest przechowywany w tabeli danych sterując bezpośrednio częstotliwością PLSY używając D8132				M8132 (wyłącznie) ten znacznik może być użyty z jedną instrukcją DHSZ jednocześnie	

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie 1 - Standard: (Dostępne na wszystkich jednostkach) Ta instrukcja działa w taki sam sposób, jak standardowy ZCP (FNC 11). Jedyńa różnicą jest to, że porównywanym urządzeniem jest szybki licznik (określony jako S₃).

Również wszystkie wyjścia (D) są natychmiast aktualizowane, dzięki operacji przerwania wykonywanej przez instrukcję DHSZ. Należy pamiętać, że kiedy urządzenie jest określone w argumencie D, jest ono adresem początkowym dla 3 kolejnych urządzeń. Każde z nich jest użyte do reprezentowania statusu bieżącego porównania. Na przykład, bazując na powyższym przykładzie,

Y10 (D) C251 jest mniejsze niż S₁ = K1000 (S₃ < S₁)
 Y11 (D₊₁) C251 jest większe niż S₁ = K1000 ale mniejsze niż S₂, K1200 (S₃ > S₁, S₃ < S₂)
 Y12 (D₊₂) C251 jest większe niż S₂ = K1200 (S₃ > S₂)

- Dalsze szczegóły i uwagi na temat użycia funkcji szybkich liczników w sekcji "Uwagi" pod HSCS (FNC 52).
 Odnośne uwagi: a, b i c. Należy również zwrócić uwagę co do liczby dozwolonych szybkich instrukcji.

Działanie 2 - Użycie HSZ z tablicą danych: (Dostępne na FX2C i FX2N) Jest wybierane, kiedy urządzeniu docelowemu D przypisany jest specjalny znacznik M8130. Urządzenia S₁ i S₂ mogą być wtedy użyte do zdefiniowania tablicy danych, używając S₁ jako adresu początkowego i S₂ jako liczby zapisów w tabeli - maksymalna ilość zapisów wynosi 128. Każdy zapis zajmuje 4 kolejne rejestry danych, podzielonych w następujący sposób (dla pojedynczego zapisu rejestrów danych od D do D₊₃):

Rejestry danych	Pojedynczy rekord	
	D, D ₊₁	Używane jako podwójny (32-bitowy) rejestr zawierający dane do porównania
	D ₊₂	Przechowuje numer urządzenia I/O (HEX), sterowanego wyjścia „Y”, tzn. H10=Y10
	D ₊₃	Przechowuje działanie (SET/RESET) wykonywane na urządzeniu D ₊₂ . Uwaga: dla SET (ON) D ₊₃ =1, dla RESET (OFF) D ₊₃ =0

Należy zapoznać się z następującymi punktami podczas analizowania przykładu po prawej stronie. Należy zauważyć, że obowiązują wszystkie podstawowe zasady odnoszące się do szybkich liczników.

Tabela danych jest przetwarzana wiersz po wierszu, tzn. tylko jeden wiersz jest aktywny jako dana do porównania. Numer bieżącego aktywnego wiersza jest przechowywany w rejestrze danych D8130. Kiedy jest osiągnięta wartość porównania aktywnego wiersza, przypisane urządzenie Y jest ustawiane lub zerowane, a numer aktywnego wiersza jest zwiększany o 1. Kiedy już wszystkie wiersze w tabeli danych zostaną przetworzone, wskaźnik bieżącego wiersza (D8130) jest zerowany (tabela jest gotowa do następnego przetwarzania) a znacznik ukończenia operacji, M8131, jest ustawiany w stan ON.

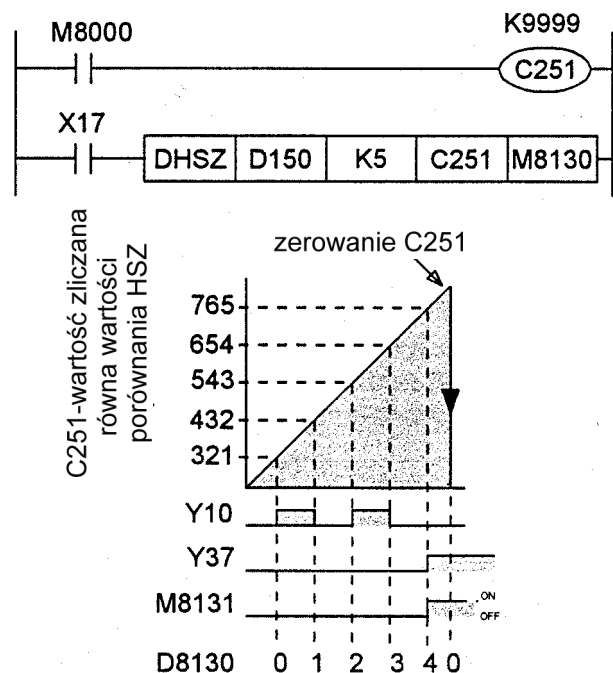
Numer rekordu [D8130]	Wartość porównywana (rejestr młodszy/starszy) [D, D+1]	Wybrane urządzenie wyjściowe „Y” [D+2]	SET/RESET Urządzenia „Y” (1 = SET, 0 = RESET) [D+3]
0	[D150, D151] K321	[D152] H10 (Y10)	[D153] K1
1	[D154, D155] K432	[D156] H10 (Y10)	[D157] K0
2	[D158, D159] K543	[D160] H10 (Y10)	[D161] K1
3	[D162, D163] K654	[D164] H10 (Y10)	[D165] K0
4	[D166, D167] K765	[D168] H37 (Y37)	[D169] K1

Jeżeli szybki licznik jest zerowany (przez program lub wejście sprzętowe), gdy wznowia liczenie i osiąga wartość porównania pierwszego rekordu, znacznik M8131 będzie wyzerowany. Użytkownik nie może zmieniać ani stanu M8131, ani zawartości D8130. Jeżeli instrukcja DHSZ jest wyłączona wtedy wszystkie przypisane znaczniki są zerowane.

Należy zachować ostrożność podczas zerowania szybkiego licznika lub wyłączania instrukcji DHSZ, ponieważ wszystkie przypisane urządzenia wyjściowe Y pozostaną w swoim ostatnim stanie, tzn. jeżeli wyjście było w stanie ON, pozostanie ono w stanie ON aż do momentu niezależnego wyzerowania przez użytkownika.

Dane w wierszach nieaktywnych mogą być zmieniane podczas działania umożliwiając aktualizację tabel danych. Każda zmiana jest przetwarzana na końcu bieżącego przebiegu programu. Instrukcja HSZ będzie przetwarzała tylko aktywne wiersze danych, tzn. nie wyzeruje się z powodu uaktualnienia nieaktywnego wiersza danych.

Kiedy instrukcja DHSZ jest inicjalizacyjnie aktywowana, nie będzie przetwarzać porównania aż do następnego przebiegu programu, ponieważ CPU wymaga niewielkiego opóźnienia do zainicjowania tabeli porównań.



Działanie 3 - Połączone operacje HSZ i PLSY: (Dostępne na jednostkach FX_(2C) i FX_(2N)). Działanie 3 umożliwia wspólne użycie instrukcji HSZ i PLSY jako pętli sterującej. Ta operacja jest wybierana kiedy urządzeniu docelowemu (D) jest przyporządkowany znacznik M8132. Wtedy urządzenia S1 i S2 mogą być użyte do zdefiniowania tabeli danych, używając S1 jako adresu początkowego i S2 jako liczby wierszy w tabeli - maksymalna liczba wierszy wynosi 128. Każdy wiersz zajmuje 4 kolejne rejestry danych (D do D₊₃), podzielonych na dwa 32-bitowe rejestry danych.

Pierwsza para rejestrów danych (D, D₊₁) zawiera wartość porównania używaną przez szybki licznik. Druga para rejestrów danych (D₊₂, D₊₃) zawiera wartość (od 0 do 1000), która reprezentuje częstotliwość wyjścia w Hz. Ta wartość jest wpisana do specjalnego rejestru danych D8132, kiedy porównanie wykonane przez instrukcję DHSZ daje na wyjściu prawdę.

Specjalny rejestr danych D8132 może być użyty jako dana źródłowa dla wyjścia instrukcji PLSY (FNC 57) umożliwiając zmianę wyjścia z zależności od odpowiednich danych zliczania.

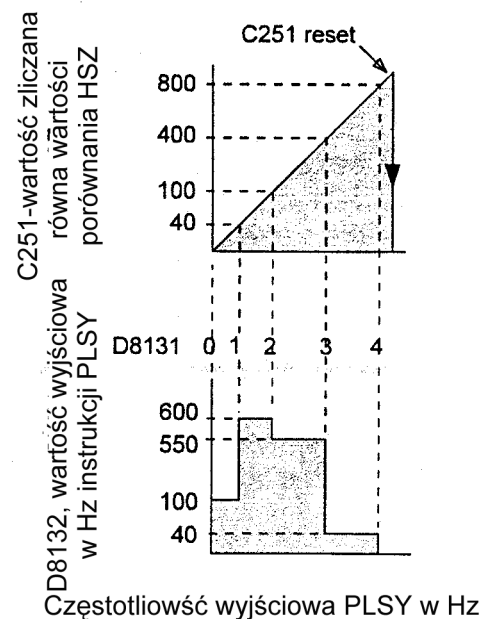
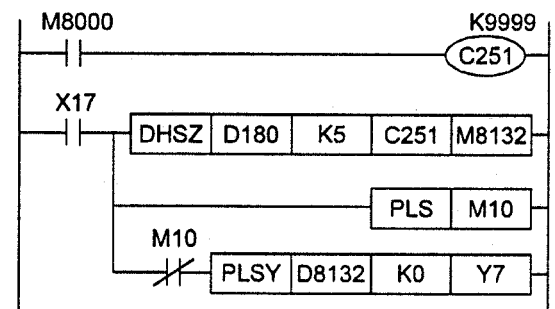
Numer rekordu [D8131]	Wartość porównywana (rejestr młodszy/starszy) [D, D ₊₁]	Częstotliwość wyjściowa instrukcji PLSY [D ₊₂ , D ₊₃]
0	[D180, D181] K40	[D182, D183] K100
1	[D184, D185] K100	[D186, D187] K600
2	[D188, D189] K400	[D190, D191] K550
3	[D192, D193] K800	[D194, D195] K40
4	[D196, D197] K0	[D198, D199] K0

Tak jak w Działaniu 2 tylko jeden wiersz w tabeli danych jest aktywny w danym czasie. Numer bieżącego wiersza jest przechowywany w rejestrze danych D8131. Aby śledzić bieżącą wartość porównania, rejestry danych D8134 i D8135 powinny być monitorowane jako urządzenie 32-bitowe.

Kiedy już zostanie przetworzony końcowy zapis w tabeli danych, znacznik ukończenia operacji, M8133, jest ustawiany w stan ON, a licznik wierszy D8131 wraca do pierwszego rekordu.

Jeżeli szybki licznik i operacja PLSY tworzą zamkniętą pętlę, wtedy ostatni wiersz wprowadzony do tabeli jest ustawiony na K0 dla wartości porównania i K0 dla częstotliwości wyjścia instrukcji PLSY. To zatrzyma sterowany system i licznik rekordów nie będzie w stanie wrócić do początku tabeli danych do czasu, kiedy przypisany szybki licznik będzie wyzerowany przez program lub wejście sprzętowe. Ta sytuacja może być łatwo monitorowana przez sprawdzanie par rejestrów danych D8134, D8135 dla wartości 0.

Zaleca się, aby działanie instrukcji PLSY było opóźnione o jeden przebieg programu, tak, aby tabela danych instrukcji DHSZ mogła być utworzona w operacji inicjującej. Sugerowany program, używający znacznika impulsu, jest pokazany w przykładzie na tej stronie.

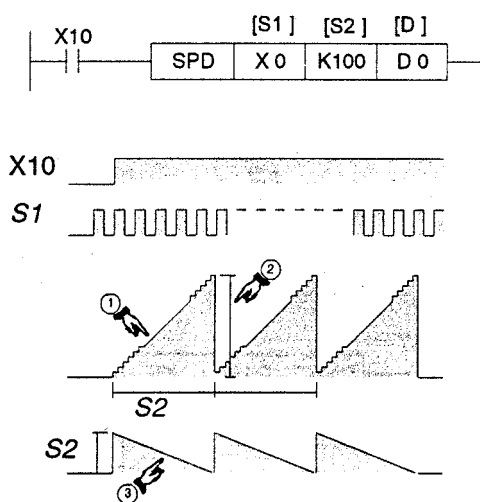


5.6.7 SPD (FNC 56)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
SPD FNC 56 (detekcja szybkości)	Wykrywa liczbę impulsów enkodera w podanym czasie ramki. Wynik może być użyty do obliczenia prędkości	X0 do X5	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	T, C, D, Z, (V) Uwaga: Użyte są 3 kolejne urządzenia. W przypadku D=Z monitor D8028, D8029 i D8030	SPD: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Liczba impulsów podawanych na wejściu S1 jest zliczana i przechowywana w D₊₁; jest to bieżąca wartość zliczania. Zliczanie odbywa się w ustalonym przedziale czasu, określonym w ms przez S2. Pozostały do zliczenia czas jest wyświetlany na urządzeniu D+2. Liczba zliczanych impulsów od ostatnio mierzonego zliczania jest przechowana w D. Wykres obok pokazuje operację SPD w sensie graficznym:

bieżąca wartość zliczana, urządzenie D+1

zapamiętana, ostatnio zliczona wartość w czasie S2, urządzenie D

pozostały czas bieżący w ms, urządzenie D+2

Uwagi:

- Kiedy zakończy się okres zliczania, wartość zliczona w D₊₁ jest natychmiast wpisana do D. D₊₁ jest zerowane i rozpoczyna się nowy okres zliczania.
- Ponieważ instrukcja wykorzystuje zarówno szybki proces jak i przerwanie, tylko wejścia X0-X5 mogą być użyte jako urządzenia źródłowe S1. Należy pamiętać, że określone jako S1 urządzenie nie może kolidować z żadną inną szybką funkcją, aktualnie wykonywaną, np. szybkim licznikiem, używającym tego samego wejścia. Instrukcja SPD jest traktowana jako licznik 1-fazowy.
- Instrukcja SPD może być użyta wielokrotnie, ale urządzenie źródłowe S1 ogranicza to do 6 razy.
- Kiedy już wartości zostały zapamiętane w D, można obliczyć różne rodzaje prędkości. Prędkości te mogą być kątowe, np. w radianach na min, liniowe w m/min, itp.. Zależy to całkowicie od użytych operacji matematycznych. Mogą być zastosowane następujące interpretacje:

$$\text{Prędkość liniowa } N \text{ (km/h)} = \frac{3600 \times (D)}{n \times S_2} \times 10^3$$

gdzie n = liczba liniowych podziałek enkodera (tarczy impulsowej) na km.

$$\text{Prędkość kątowna } N \text{ (rpm)} = \frac{60 \times (D)}{n \times S_2} \times 10^3$$

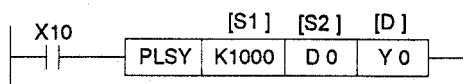
gdzie n = liczba impulsów na obrót tarczy enkodera

5.6.8 PLSY (FNC 57)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
PLSY FNC 57 (wyjście impulsowe)	Generuje określoną liczbę impulsów o określonej częstotliwości	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		Y Uwaga: FX0S/FX0N: tylko Y000 ☒ FX: dowolne Y☆☆☆ FX2N: tylko Y000 lub Y001☒	PLSY: 7 kroków DPLSY: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

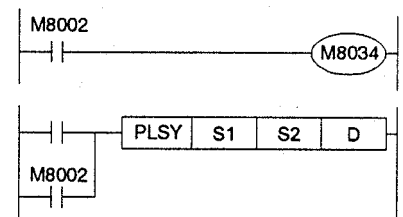
**Działanie:**

Określona liczba impulsów S2 o częstotliwości określonej w S1 jest generowana na urządzenie D. Ta instrukcja jest używana w sytuacjach, gdzie liczba impulsów ma

podstawowe znaczenie.

Uwagi:

- Użytkownicy sterowników FX mogą zadeklarować częstotliwość wyjściową (S1) w zakresie od 1 do 1000 Hz. Użytkownicy FX, wersja 2.2 lub wcześniejsza będą musieli zainicjować instrukcję PLSY. Do osiągnięcia tego celu może być użyty program pokazany po prawej stronie. Użytkownicy FX0/0N mogą używać częstotliwości w zakresie 10-2000 Hz. Użytkownicy FX2N mogą używać częstotliwości w zakresie 2-20.000 Hz.
- Maksymalna liczba impulsów: operacja 16-bitowa: od 1 do 32.767 impulsów, operacja 32-bitowa: od 1 do 2.147.483.647 impulsów.
Uwaga: znacznik M8029 jest ustawiony w stan ON, kiedy wygenerowano określoną liczbę impulsów. Licznik impulsów i znacznik ukończenia (M8029) są zerowane, kiedy instrukcja PLSY jest deaktywowana. Jeżeli przypiszemy wartość "0" liczbie impulsów, instrukcja PLSY będzie stale generowała impulsy, tak długo, jak instrukcja będzie aktywowana.
- Pojedynczy impuls ma 50% wypełnienie. Oznacza to, że jest w stanie ON przez 50% czasu trwania impulsu i konsekwentnie w stanie OFF przez pozostałych 50% czasu impulsu. Wyjście jest sterowane przez przerwanie, tzn. czas przebiegu programu (czas skanowania) nie ma wpływu na generowany cykl wyjścia.
- Dane w argumentach S1 i S2 mogą być zmienione podczas wykonywania instrukcji. Jednak nowe dane w S2 nie będą wzięte pod uwagę, dopóki bieżąca operacja nie zostanie ukończona, tzn. aż instrukcja nie zostanie wyzerowana przez rozłączenie styku sterowania.
- Ta instrukcja może być użyta tylko raz w jednym przebiegu programu. Również, tylko jedna z następujących instrukcji: PLSY (FNC 57), PLSR (FNC 59) może być użyta w programie tylko raz.



Możliwe jest użycie podprogramów lub innych tego typu technik programowania do izolowania różnych instancji tej instrukcji. W takim przypadku bieżąca instrukcja musi być deaktywowana przed zmianą na nową instancję.

- f) Ze względu na naturę szybkiego wyjścia, zaleca się użycie tej instrukcji tylko w modułach z wyjściami tranzystorowymi. Wyjściowe przekaźniki mechaniczne wraz z upływem czasu będą się zużywać, co spowoduje pojawienie się fałszywych wyjść, z powodu mechanicznego "odbijania się" styków. Aby zapewnić "czysty" sygnał wyjściowy prąd obciążenia powinien wynosić 200 mA lub więcej. Może się okazać konieczne zastosowanie rezystorów podciągających.
- g) Jednostki FX_(2C) i FX_(2N) mogą używać instrukcji HSZ (FNC 55) z instrukcją PLSY, kiedy urządzenie źródłowe S1 jest ustawione na D8132. Więcej szczegółów na stronie 5-59.
- h) Jednostki FX z wersją CPU 3.07 i późniejszą oraz jednostki FX_{2C} mogą śledzić liczbę impulsów, które zostały wygenerowane, jako podwójne słowo, używając urządzeń D8136 i D8137.

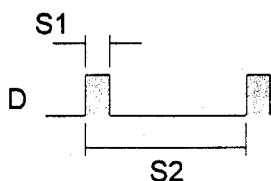
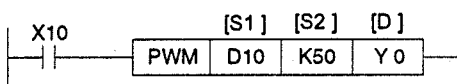
Jednostki FX_{2N} mogą również śledzić liczbę impulsów wyjściowych na Y0, używając urządzeń D8140 i D8141 i liczbę impulsów wyjściowych na Y1, używając urządzeń D8142 i D8143. Całkowita liczba impulsów wyjściowych może być monitorowana przy użyciu D8136 i D8137.

5.6.9 PWM (FNC 58)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
PWM FNC 58 (modulacja szerokości impulsu)	Generuje ciąg impulsów o określonej charakterystyce	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z Uwaga: S1 S2		Y Uwaga: FX0S/FX0N: tylko Y001 ☒ FX: dowolne Y☆☆☆ FX2N: tylko Y000 lub Y001☒	PWM: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie:

Podczas działania tej instrukcji jest generowany nieprzerwany ciąg impulsów na urządzenie D. Charakterystyka impulsów jest następująca: (S2) oznacza odległość w ms pomiędzy dwoma identycznymi częściami kolejnych impulsów (S1) oznacza czas trwania (w ms) pojedynczego impulsu.

Uwagi:

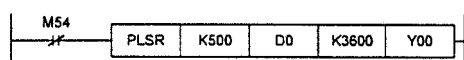
- Ponieważ jest to instrukcja 16-bitowa, dostępne zakresy czasu dla S1 i S2 wynoszą od 1 do 32.767
- Wypełnienie oblicza się poprzez dzielenie S1 przez S2. Stąd S1 nie może mieć wartości większej niż S2, ponieważ to oznaczałoby, że impuls trwa dłużej niż wynosi odległość między dwoma impulsami, tzn. drugi impuls zaczynałby się zanim pierwszy byłby skończony. Jeżeli tak się stanie, pojawi się błąd. Ta instrukcja jest używana, kiedy długość impulsu jest sprawą nadrzędną.
- Instrukcja PWM może być użyta tylko raz w programie użytkownika.
- Ze względu na naturę szybkiego wyjścia, zaleca się użycie tej instrukcji tylko w modułach z wyjściami tranzystorowymi. Przekazniki wyjściowe wraz z upływem czasu będą się zużywać, co spowoduje pojawienie się fałszywych wyjść, z powodu mechanicznego "odbijania się" styków. Aby zapewnić "czysty" sygnał wyjściowy, prąd obciążenia powinien wynosić 200 mA lub więcej. Może się okazać konieczne zastosowanie rezystorów podciągających.

5.6.10 PLSR (FNC 59)

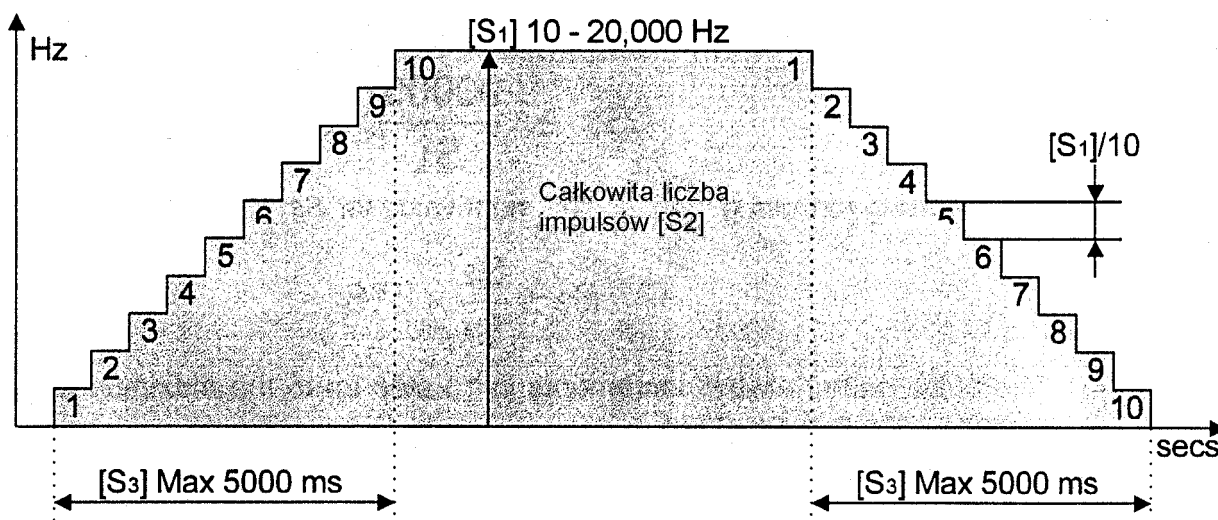
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S ₁	S ₂	S ₃	D	
PLSR FNC 59 (impulsy schodkowe)	Generuje określoną liczbę impulsów schodkowych rosnących i malejących o określonej częstotliwości	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z			Y	PLSR: 9 kroków DPLSR: 17 kroków
		Uwaga: S1 S2				

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Określona liczba impulsów S₂ jest generowana na urządzenie wyjściowe D. Częstotliwość wyjściowa rośnie w 10 krokach do maksymalnej częstotliwości S₁ w czasie S₃ ms, potem maleje również w czasie S₃ ms. Ta instrukcja jest używana do generowania nieskomplikowanych krzywych ze strefą zwiększania/zmniejszania częstotliwości, gdzie najważniejsza jest liczba impulsów.

**Uwagi:**

- Użytkownicy FX_{2N} mogą używać częstotliwości z zakresu od 10 do 20.000 Hz. Częstotliwość powinna być ustawiana jako wielokrotność 10. Jeżeli nie, będzie zaokrąglona w górę do najbliższej wielokrotności 10. Kroki narastania i opadania są ustawione na 1/10 maksymalnej częstotliwości. Należy wziąć to pod uwagę, stosując silniki krokowe.
- Maksymalna liczba impulsów: operacja 16-bitowa: od 110 do 32.767 impulsów
operacja 32-bitowa: od 110 do 2.147.483.647 impulsów
Przy ustawieniu 110 nie ma gwarancji poprawnej ilości impulsów na wyjściu.
- Czas narastania musi mieścić się w granicach opisanych na następnej stronie.

- d) Urządzenie wyjściowe jest ograniczone tylko do wyjść Y0 lub Y1 i powinno być typu tranzystorowego.
- e) Ta instrukcja może być użyta tylko raz w jednym przebiegu programu. Również, tylko jedna z następujących instrukcji: PLSY (FNC 57), PLSR (FNC 59) może być użyta w programie tylko raz.



Możliwe jest użycie podprogramów lub innych tego typu technik programowania do izolowania różnych instancji tej instrukcji. W takim przypadku bieżąca instrukcja musi być deaktywowana przed zmianą na nową instancję.

- f) Jeżeli liczba impulsów nie wystarcza do osiągnięcia maksymalnej częstotliwości, wtedy częstotliwość jest automatycznie obniżana.
- g) Znacznik M8029 ustawia się w stan ON, kiedy została osiągnięta określona liczba impulsów. Licznik impulsów i znacznik ukończenia (M8029) są zerowane, kiedy instrukcja PLSR jest deaktywowana.
- h) Ograniczenia czasu narastania

Maksymalny czas narastania S_3 wynosi 5.000 ms. Jednak ograniczenia S_3 zależą od innych parametrów systemu, zgodnie z następującymi 4 punktami:

- 1) Ustawienie S_3 powinno być większe niż 10 x maksymalny czas przebiegu programu (D8012)
Jeżeli ustawienie jest mniejsze, wtedy wykres kroków narastania staje się nierównomierny.
- 2) Następujące równanie podaje minimalną wartość S_3

$$S_3 \geq \frac{90000}{S_1} \times 5$$

- 3) Następujące równanie podaje maksymalną wartość S_3

$$S_3 \leq \frac{S_2}{S_1} \times 818$$

- 4) Wyjście impulsowe zawsze rośnie o 10 kroków do maksymalnej częstotliwości, jak pokazano na poprzedniej stronie.




Jeżeli parametry nie spełniają powyższych warunków należy zredukować rozmiar S_1 .

- Możliwa częstotliwość wyjściowa jest ograniczona od 2 do 20.000 Hz. Jeżeli maksymalna częstotliwość lub rozmiar kroku narastania są poza tym ograniczeniem, są one automatycznie dostosowane do granic zakresu.
- Jeśli sygnał sterujący jest wyłączony, wszystkie wyjścia zatrzymują się. Po ponownym uruchomieniu cały proces zaczyna się od początku.
- Nawet jeżeli argumenty są zmienione podczas operacji, profil wyjścia nie zmienia się. Nowe wartości dają efekt od następnej operacji.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

	1. Sterowanie przebiegiem programu	5-4
	2. Instrukcje przesłania i porównania	5-16
	3. Operacje arytmetyczne i logiczne	5-24
	4. Rotacje i funkcje przesunięcia	5-34
	5. Operacje na danych	5-42
	6. Szybkie przetwarzanie	5-52
	7. Instrukcje podręczne	5-66
	8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
	9. Zewnętrzne urządzenia szeregowo	5-94
	10. Zewnętrzne jednostki F2	5-112
	11. Operacje zmiennoprzecinkowe 1 & 2	5-120
	12. Operacje trygonometryczne	5-128
	13. Operacje na danych 2	5-132
	14. Sterowanie zegarem czasu rzeczywistego	5-136
	15. Kody Graya	5-144
	16. Porównania operacyjne	5-148

5.7 Instrukcje podręczne - funkcje 60 - 69

Zawartość:

			Strona
IST	stan inicjujący	FNC 60	5-67
SER	wyszukiwanie	FNC 61	5-69
ABSD	programator bębnowy absolutny	FNC 62	5-70
INCD	programator bębnowy przyrostowy	FNC 63	5-71
TTMR	timer śledzący	FNC 64	5-72
STMR	timer specjalny - definiowany	FNC 65	5-72
ALT	przerzutnik	FNC 66	5-73
RAMP	wartość zmienna	FNC 67	5-73
ROTC	sterowanie stołem obrotowym	FNC 68	5-75
SORT	sortowanie danych	FNC 69	5-77



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

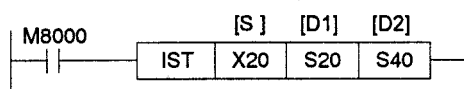
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.7.1 IST (FNC 60)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D ₁	D ₂	
IST FNC 60 (stan inicjujący)	Automatycznie ustawia wielomodowy system operacyjny STL	X, Y, M, S, Uwaga: użytych jest 8 kolejnych urządzeń	S, Uwaga: FX0: S20 do S63 FX0N: S20 do S127 FX: S20 do S899 D1 musi być mniejsze od D2		IST: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja ustawia automatycznie wielomodowy system operacyjny STL. Składa się on z ręcznych i automatycznych trybów operacyjnych.

Uwagi:

- Instrukcja automatycznie przyporządkowuje i używa wielu znaczników i urządzeń słowowych; są one przedstawione w kolumnie po prawej stronie.
- Instrukcja może być użyta **tylko raz**. Powinna być zaprogramowana blisko początku programu przed sterowanym obwodem STL.
- Wymagany tryb operacyjny jest wybierany przez sterowanie urządzeniem skojarzonym z argumentami S₊₀ do S₊₄ (5 wejść). Żadne z urządzeń w tym zakresie nie powinno być w stanie ON w tym samym czasie (tylko jedno może być załączone). Zaleca się, aby te "wejścia" były wybierane przy użyciu przełącznika rotacyjnego. Jeżeli bieżąco wybrany tryb operacyjny będzie zmieniony zanim zostanie ustawiony znacznik M8043, wszystkie wyjścia zostaną przełączone w stan OFF.
- "Pozycja zero" jest terminem używanym do identyfikowania pozycji odniesienia, skąd sterowane urządzenie startuje i wraca po ukończonym zadaniu. Stąd tryb operacyjny "zero return - powrót do zera" powoduje, że sterowany system wraca do pozycji odniesienia.

Przyporządkowane urządzenia**Urządzenia wybierane pośrednio przez użytkownika:**

- S₊₀ Operacja ręczna
- S₊₁ Powrót do zera
- S₊₂ Operacja krokowa
- S₊₃ Operacja jednego cyklu
- S₊₄ Operacja cykliczna
- S₊₅ Start powrotu do zera
- S₊₆ Start operacji automatycznej
- S₊₇ Stop

Stany inicjujące:

- S0 inicjuje operację ręczną
- S1 inicjuje operację powrotu do zera
- S2 inicjuje operację automatyczną

Stany ogólne:

- S10-S19 sekwencja powrotu do zera
- D₁, D₂ sekwencja powrotu automatycznego

Znaczniki specjalne:

- M8040 = ON transfer stanu STL jest wstrzymany
- M8041 = ON stany inicjujące są odblokowane
- M8042 = ON wystąpił impuls startu na wejściu startowym
- M8043 = ON zakończono powrót do zera
- M8044 = ON wykryto zero maszynowe
- M8047 = ON monitor STL odblokowany



"Pozycja zero" jest czasem określa jako pozycja spoczynkowa, bezpieczna, neutralna lub pozycja odniesienia

- Dostępne tryby operacyjne są podzielone na dwie główne grupy: ręczną i

automatyczną. Dla każdej grupy istnieją tryby podrzędne. Ich operacje zdefiniowane są następująco:

Grupa ręczna

Tryb ręczny (wybierany przez urządzenie S+0) - zasilanie dla poszczególnych obciążeń jest włączane i wyłączane przy pomocy oddzielnie dostępnych środków, często są to dodatkowe przyciski

Powrót do zera (wybierany przez S+1) - układy wykonawcze powracają do swoich pozycji początkowych, gdy pojawi się sygnał powrotu do zera (S+5)

Grupa automatyczna

Tryb jeden krok (wybierany przez S+2) - Sterowana sekwencja jest wykonywana krok po kroku. Każdy nowy krok wymaga podania sygnału na wejście "start" (S+6).

Tryb jeden cykl (wybierany przez S+3) - Układy wykonawcze działają przez jeden cykl. Po zakończeniu cyklu układy powracają do pozycji spoczynkowej. Następny cykl zostanie wykonany po podaniu sygnału na wejście "start" (S+6).

Wykonywany cykl można zatrzymać w każdej chwili, podając sygnał na wejście "stop". Aby wznowić sekwencję cyklu od zatrzymanej pozycji należy podać ponownie sygnał na wejście "start".

Tryb automatyczny (wybierany przez S+4) - W tym trybie możliwa jest praca w pełni automatyczna. Wykonanie zaprogramowanego cyklu jest powtarzane, po podaniu sygnału "start" (S+6). Bieżąco wykonywana operacja nie jest zatrzymywana natychmiast po podaniu sygnału "stop", lecz dopiero po zakończeniu bieżącego cyklu.



Uwaga: Sygnały start, stop, i zero są często zadawane poprzez dodatkowe, ręcznie obsługiwane przyciski.

Należy zauważyć, że wejście stop jest tylko sygnałem zatrzymania programu i **nie może** zastępować wyłącznika bezpieczeństwa. Wszystkie urządzenia bezpieczeństwa powinny być systemem, który efektywnie zatrzyma maszynę i odłączy zewnętrzne zasilanie. Więcej szczegółów w normach określających zasady bezpieczeństwa



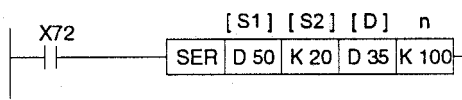
Program przykładowy będzie przedstawiony w rozdziale opisującym użycie funkcji IST.

5.7.2 SER (FNC 61)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S1	S2	D	n	
SER FNC 61 (wyszukiwanie)	Generuje listę statystyk o danej umieszczonej-znalezionej w stosie	KnX, KnY, KnM, KnS, T, C, D	KnX, KnY, KnM, KnS, T, C, D, V, Z, K, H	KnY, KnM, KnS T, C, D Uwaga: użytych jest 5 kolejnych urządzeń	K, H, D ☒ Uwaga: n=256 dla operacji 16- bitowych n=128 dla operacji 32- bitowych	SER, SERP: 9 kroków DSER, DSERP: 17 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja przeszukuje stos danych o długości n od adresu początkowego S1. Szukana dana jest wyspecyfikowana w S2, a wyniki przeszukania są zachowane w urządzeniu D

dla 5 kolejnych urządzeń.

Urządzenie docelowe	Opis urządzenia
D	Całkowita liczba wystąpień szukanej wartości S2 (0, gdy nie znaleziono wystąpienia)
D+1	Pozycja (w przeszukiwanym stosie danych) pierwszego wystąpienia szukanej wartości S2
D+2	Pozycja (w przeszukiwanym stosie danych) ostatniego wystąpienia szukanej wartości S2
D+3	Pozycja (w przeszukiwanym stosie danych) najmniejszej wartości znalezionej w stosie (jeżeli istnieje kilka tych samych wartości, zwracane jest ostatnie wystąpienie)
D+4	Pozycja (w przeszukiwanym stosie danych) największej wartości znalezionej w stosie (jeżeli istnieje kilka tych samych wartości, zwracane jest ostatnie wystąpienie)

Uwagi:

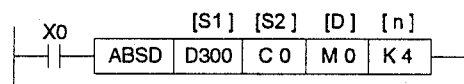
- Do określenia najmniejszej i największej wartości stosuje się normalne reguły algebraiczne, tzn. -30 jest mniejsze niż 6 itd.
- Jeżeli nie można znaleźć żadnego pojawienia się szukanej danej wtedy urządzenia docelowe D, D+1 i D+2 będą równe 0.
- Należy pamiętać, że podczas używania rejestru danych S, jako urządzenia docelowego D operacja 16-bitowa zajmie 5 kolejnych rejestrów danych, ale operacja 32-bitowa zajmie 10 rejestrów danych tworząc w parach 5 podwójnych słów.
- Kiedy używa się wielokrotnych urządzeń bitowych do przechowania wyniku (bez względu na to, czy są to operacje 16- czy 32-bitowe), tylko określony rozmiar grupy jest zapisywany dla 5 kolejnych wystąpień, tzn. K1Y0 zajmie 20 urządzeń bitowych od Y0 (K1 = 4 urządzenia bitowe, wtedy będzie 5 grup dla 5 wyników). Maksymalna wielkość stosu wynosi 256 (0-255) pozycji, optymalną grupą wymaganych urządzeń bitowych jest K2, tzn. 8 urządzeń bitowych.

5.7.3 ABSD (FNC 62)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty				Kroki programu
		S1	S2	D	n	
ABSD FNC 62 (programator bębnowy absolutny)	Generuje różnorodne wzory bitowe w oparciu o wartość bieżącą licznika	KnX, KnY, KnM, KnS, (w grupach po 8) T, C, D uwaga: liczniki szybkie nie są dozwolone	C uwaga: użyte są 2 kolejne urządzenia	Y, M, S uwaga: użytych jest n kolejnych urządzeń	K, H ☒ Uwaga: n ≤ 64	ABSD: 9 kroków DABSD: 17 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

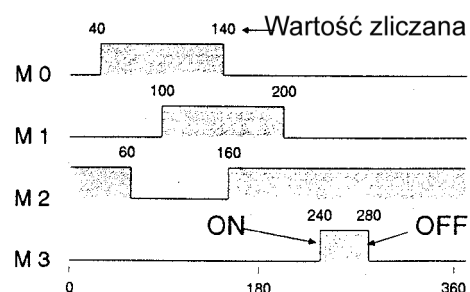
Instrukcja generuje różnorodne wzory bitowe (istnieje n adresowanych wyjść) w oparciu o wartość bieżącą wybranego licznika S2.

Uwagi:

- Wartość bieżąca wybranego licznika (S2) jest porównywana ze zdefiniowaną przez użytkownika tabelą danych. Adres początkowy tej tabeli zawarty jest w argumencie S1. S1 powinien zawsze zawierać parzystą liczbę urządzeń.
- Dla każdego bitu docelowego (D) istnieją dwie kolejne wartości danych przechowywane w tabeli danych. Pierwsza wartość reprezentuje liczbę zdarzeń kiedy urządzenie docelowe D jest ustawione w stan ON. Druga wartość identyfikuje zdarzenie zerowania. Wartości tabeli danych są rozmieszczone jako kolejna para elementu pomiędzy D i D+n.
- Tabela danych ma długość równą 2 x n pozycji danych. W zależności od formatu tabeli danych pojedyncze wprowadzenie danej może być słowem, takim jak D300 lub grupą 16 urządzeń bitowych, np. K4X000.
- W tabeli danych mogą być użyte wartości od 0 do 32.767
- Instrukcja ABSD może być użyta **tylko raz**.
- Jednostki FX z CPU w wersji 3.07 lub wyższej i FX2C mają w tej instrukcji opcję podwójnego słowa.

Dla elementów M0-M3 może być skonstruowany następujący przebieg czasowy, zgodnie z poniższym przykładem instrukcji i tabelą danych.

Gdy licznik S2 osiągnie wartość poniżej, urządzeniem docelowym D jest		Przyporządkowane urządzenie docelowe D
ON	OFF	
D300-40	D301-140	M0
D302-100	D303-200	M1
D304-160	D305-60	M2
D306-240	D307-280	M3

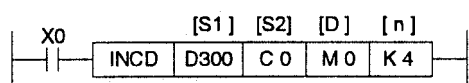


5.7.4 INCD (FNC 63)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S1	S2	D	n	
INCD FNC 63 (programator bębnowy przyrostowy)	Generuje ciąg impulsów na różnych wyjściach w oparciu o zawartość pary liczników	KnX, KnY, KnM, KnS, (w grupach po 8) T, C, D uwaga: liczniki szybkie nie są dozwolone	C	Y, M, S uwaga: użytych jest n kolejnych urządzeń	K, H ☒ Uwaga: n ≤ 64	INCD: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	Operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

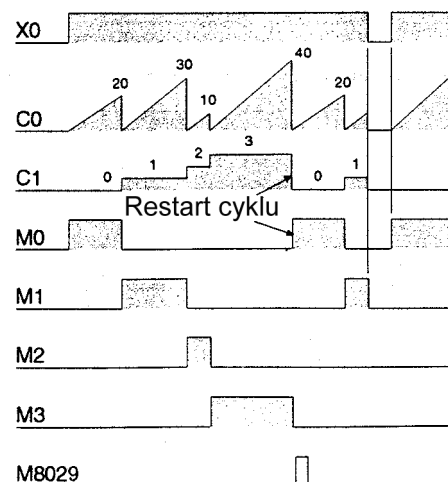
Instrukcja generuje ciąg impulsów kolejno na różnych wyjściach w oparciu o aktualną zawartość pary wybranych liczników (S2, S2+1)

Uwagi:

- Ta instrukcja wykorzystuje tabelę danych, która zawiera pojedynczą listę wartości, które mają być wybrane i porównane przez dwa kolejne liczniki (S2, S2+1). Tabela danych ma adres początkowy S1 i składa się z n elementów danych.
- Licznik S2 jest zaprogramowany w konwencjonalny sposób. Wartość ustawiona dla licznika S2 musi być większa od każdej z wartości wprowadzonej do tabeli danych. Licznik S2 liczy zdarzenia użytkownika i porównuje je z wartością bieżąco wybranego elementu z tabeli. Kiedy wartość licznika i danych jest równa, S2 zwiększa stan licznika S2+1 i zeruje swoją własną wartość bieżącą. Ta nowa wartość licznika S2+1 wybiera nowy element z tabeli danych, a licznik S2 porównuje nowe wartości elementów.
- Licznik S2+1 może mieć wartość od 0 do n. Kiedy zostanie przetworzony n-ty element, znacznik ukończenia operacji, M8029, zostanie ustawiony w stan ON. To automatycznie zeruje licznik S2+1 i cały cykl zaczyna się od początku z elementem S1+0.
- W tabeli danych mogą być użyte wartości od 0 do 32.767.
- Instrukcja INCD może być użyta **tylko raz**.

Dla elementów M0-M3 może być skonstruowany następujący przebieg czasowy, zgodnie z przykładem instrukcji i tabelą danych obok.

Tabela danych		
Element danych	Wartość danych	Wartość zliczana dla licznika S2
D300	20	0
D301	30	1
D302	10	2
D303	40	3

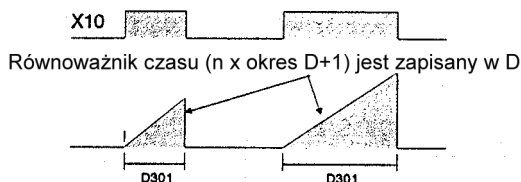
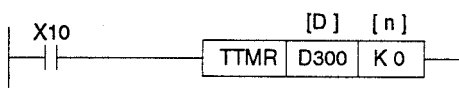


5.7.5 TTMR (FNC 64)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty		Kroki programu
		D	n	
TTMR FNC 64 (timer śledzący)	Monitoruje czasy trwania sygnałów i wpisuje zmierzone dane do rejestru	D uwaga: używane są 2 urządzenia 16-bitowe: D i D+1	K, H ☒ Uwaga: n=0: (D)=(D+1) x 1 n=1: (D)=(D+1) x 10 n=2: (D)=(D+1) x 100	TTMR: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Okres czasu, przez który aktywna jest instrukcja TTMR, zostaje zmierzony i przechowany w urządzeniu D+1 (jako liczba okresów 100 ms).

Wartość danej D+1 (w s) pomnożona przez współczynnik (wybrany przez argument n) zostaje przesunięta do rejestru D. Zawartość D może być użyta jako dana źródłowa dla pośredniego ustawienia timera lub jako surowa dana do dalszej obróbki.

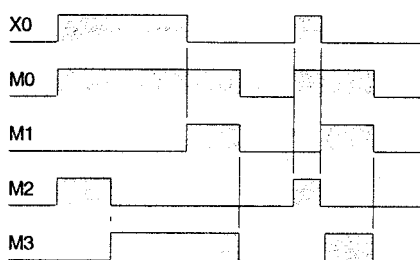
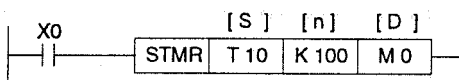
Kiedy instrukcja TTMR jest nieaktywna, D+1 jest automatycznie zerowane (D pozostaje niezmienione).

5.7.6 STMR (FNC 65)

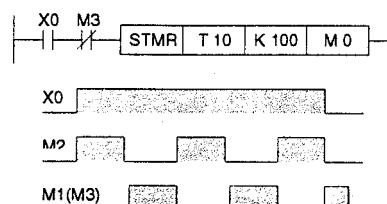
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty			Kroki programu
		S1	S2	n	
STMR FNC 65 (timer specjalny)	dostarcza dedykowanych, określonych opóźnień	T uwaga: timery 0 do 199 (100 ms)	K, H ☒ Uwaga: n =1 do 32767	Y, M, S uwaga: użyte są 4 kolejne urządzenia	STMR: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Określony timer S będzie pracował przez okres czasu n z wynikiem działania zaznaczonym przez urządzenia D+0 do D+3. Urządzenie D+0 jest timerem wyłączenia, D+1 jest uniwbiratorem wyłączenia, D+2 jest uniwbiratorem załączenia. Gdy D+3 jest użyte w konfiguracji jak pokazano poniżej, D+1 i D+2 pracują w odwracanej sekwencji.

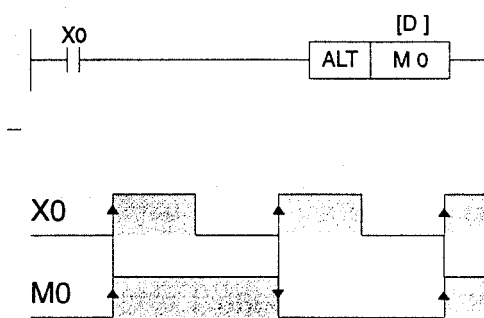


5.7.7 ALT (FNC 66)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		D			
ALT FNC 66 (przerzutnik)	stan urządzenia jest odwracany	Y, M, S			ALT, ALTP: 3 kroki

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Stan urządzenia docelowego (D) jest odwracany po każdym wykonaniu instrukcji ALT.

Oznacza to, że stan każdego bitu będzie zamieniany ze stanu ON na OFF i odwrotnie. Będzie to się pojawiało w każdym przebiegu programu (skanowaniu), chyba że zostanie użyty modyfikator impulsu lub blokada programowa.

Instrukcja ALT najbardziej nadaje się do przełączania pomiędzy dwoma trybami operacji, np. start/stop, ON/OFF

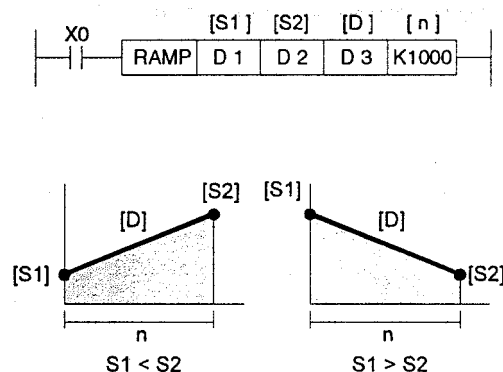
itp.

5.7.8 RAMP (FNC 67)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S1	S2	D	n	
RAMP FNC 67 (wartość zmienna) →	zmienia wartość urządzenia w określonych granicach i z określonym krokiem	D uwaga: użyte są 2 kolejne urządzenia D (D, D+1), tylko do odczytu			K, H ☒ uwaga: n=1 do 32767	RAMP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Instrukcja zmienia wartość bieżącą (D) pomiędzy granicami ustalonymi przez użytkownika (S1 i S2).

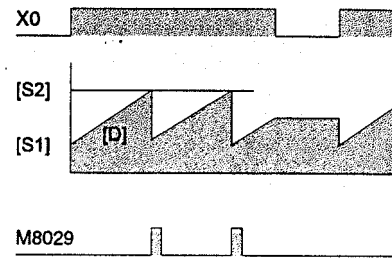
Przejście pomiędzy granicami trwa n przebiegów programu (cykli skanowania). Numer bieżącego przebiegu jest przechowywany w urządzeniu D+1.

Kiedy już wartość bieżąca D osiągnie wartość S2 znacznik ukończenia operacji, M8029, przyjmuje stan ON.

W zależności od wzajemnej relacji S1 i S2, zawartość D może rosnąć lub maleć.

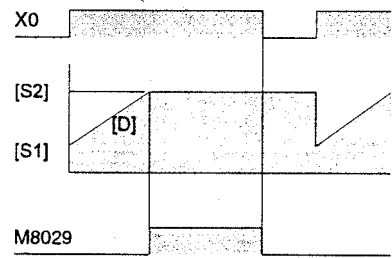
Uwagi:

- a) Użytkownicy FX mogą ustawiać tryb operacyjny instrukcji RAMP przez sterowanie stanem znacznika M8026. Kiedy znacznik jest w stanie OFF, instrukcja RAMP będzie w trybie powtarzania. Oznacza to, że kiedy wartość bieżąca D jest równa S2, instrukcja automatycznie się zeruje i ponownie uruchamia, tzn. zawartość D przyjmie wartość S1, a urządzenie D+1 (numer bieżącego przebiegu programu) przyjmie wartość 0. Jest to pokazane na diagramie obok.



Gdy M8026 jest w stanie ON, użytkownicy FX będą używać instrukcji RAMP w trybie "hold". Oznacza to, że kiedy wartość bieżąca D jest równa S2, instrukcja RAMP zostaje "zamrożona" w tym stanie, tzn. M8029 będzie w stanie ON tak długo, jak instrukcja pozostanie aktywna, a wartość D nie będzie wyzerowana, dopóki instrukcja RAMP nie będzie ponownie zainicjowana.

- b) Użytkownicy FX0 i FX0N nie mogą zmienić trybu operacyjnego instrukcji RAMP. Dla tych PLC tryb jest ustalony w takim samym przypadku, jak na jednostkach FX, kiedy M8026 jest w stanie ON, tzn. realizowany jest tryb "hold".
- c) Jeżeli instrukcja RAMP jest przerywana przed ukończeniem, wtedy bieżąca pozycja nachylenia jest "zamrożona" do momentu, kiedy sygnał sterujący zostanie wznowiony. Po ponownym uruchomieniu instrukcji rejestry D i D+1 zerują się, a cały cykl rozpoczyna się od początku.
- d) Jeżeli instrukcja RAMP jest wykonywana w trybie stałego przebiegu programu (stała długość cyklu), wtedy w D8039 jest wpisany żądany czas przebiegu (nieco dłuższy niż bieżący czas przebiegu) i M8039 jest w stanie ON. To umożliwi skojarzenie liczby przebiegów programu n (użytej do stworzenia nachylenia pomiędzy S1 i S2) z czasem. Jeżeli jeden przebieg (jeden cykl skanowania) jest równy zawartości D8039, wtedy czas potrzebny do przejścia nachylenia jest równy $n \times D8039$.



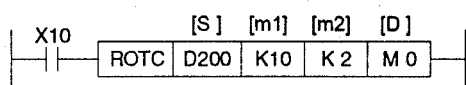
Instrukcja RAMP może być również użyta ze specjalnymi znacznikami M8193 i M8194 do odwzorowania operacji, odpowiednio, SER (FNC 61) i RS (FNC 80), kiedy jest używana ze starszą wersją oprogramowania peryferyjnego. Więcej szczegółów na stronie 1-5.

5.7.9 ROTC (FNC 68)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S1	S2	D	n	
ROTC FNC 68 (sterowanie stołem obrotowym)	steruje ruchem stołu obrotowego	D uwaga: użyte są 3 kolejne urządzenia $S+1 \leq m1$	K, H <input checked="" type="checkbox"/> uwaga: $m1=2$ do 32767 $m1 \geq m2$	K, H <input checked="" type="checkbox"/> uwaga: $m2=0$ do 32767	Y, M, S uwaga: użyte jest 8 kolejnych urządzeń	ROTC: 9 kroków

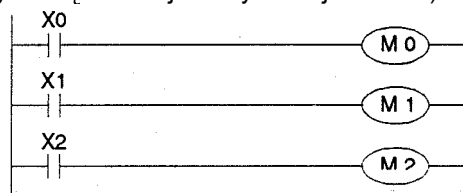
Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

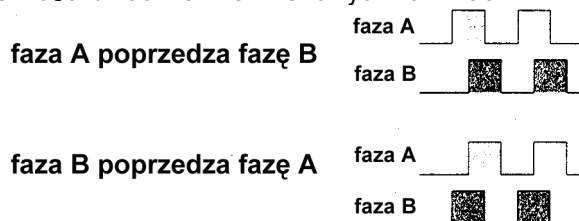
Instrukcja jest używana do wspomagania śledzenia i pozycjonowania stołu obrotowego w czasie jego przesuwania do określonego położenia.

Uwagi:

- Ta instrukcja zawiera wiele automatycznie zdefiniowanych urządzeń. Są one wyliczone po prawej stronie
- Instrukcja ROTC może być użyta w programie tylko raz.
- Instrukcja używa wbudowanego 2-fazowego licznika do wykrycia zarówno kierunku ruchu jak i przesunięcia. Urządzenia D+0 i D+1 są używane do wprowadzania impulsów fazowych, podczas gdy urządzenie D+2 jest używane do wprowadzania pozycji zerowej do stołu obrotowego. Te urządzenia powinny być zaprogramowane, jak pokazano w przykładzie poniżej, (gdzie fizyczne zakończenie odbywa się na skojarzonych wejściach X).



Kierunek ruchu jest wykrywany przez sprawdzanie zależności dwóch faz na 2-fazowych licznikach.

**Przyporządkowane urządzenia****Urządzenia wybierane pośrednio przez użytkownika:**

- D+0 sygnał A-fazy licznika - wejście
- D+1 sygnał B-fazy licznika - wejście
- D+2 wykrywanie punktu zerowego - wejście
- D+3 sygnał szybko naprzód - wyjście
- D+4 sygnał wolno naprzód - wyjście
- D+5 sygnał stop - wyjście
- D+6 sygnał wolno wstecz - wyjście
- D+7 sygnał szybko wstecz - wyjście

Stałe stołu obrotowego

- m1 liczba impulsów enkodera na obrót stołu
- m2 odległość do przebycia przy małej prędkości (w impulsach enkodera)

Zmienne operacyjne

- S+0 pozycja bieżąca w punkcie zero - tylko odczyt
- S+1 pozycja docelowa (wybranej stacji, do której ma być przesunięty) względem punktu zero - zdefiniowana przez użytkownika
- S+2 pozycja początkowa (wybranej stacji, do której ma być przesunięty) względem punktu zero - zdefiniowana przez użytkownika

- d) Kiedy otrzymany jest sygnał wejściowy z punktu zerowego (D+2), zawartość urządzenia S+0 jest zerowana. Przed rozpoczęciem nowej operacji należy upewnić się, że stół obrotowy jest zainicjalizowany przez przesunięcie do punktu zerowego lub poza. Może być to uznane za technikę kalibracji. Rekalibracja stołu obrotowego powinna być przeprowadzana okresowo, aby zapewnić poprawne działanie.
- e) Urządzenia D+3 do D+7 są automatycznie ustawiane przez instrukcję ROTC podczas jej działania. Są one używane jako znaczniki pokazujące operację, która powinna być wykonana jako następna.
- f) Wszystkie pozycje są wprowadzone w formie impulsów wymaganych przez enkoder. Można to zilustrować na następującym przykładzie:

Przykład:

Stół obrotowy zawiera enkoder, który generuje 400 (m1) impulsów na obrót. W stole jest 8 stacji (0 do 7). To znaczy, że kiedy stół przesuwana się od jednej stacji do kolejnej, bezpośrednio następującej, zliczone jest 50 impulsów enkodera. Stacja "0" jest pozycją zerową. Aby przesunąć element ze stacji 7 do stacji 3, do instrukcji ROTC muszą być wpisane następujące wartości:

$S+1 = 3 \times 50 = 150$ (pozycja stacji 3 w impulsach enkodera od punktu zero)

$S+2 = 7 \times 50 = 350$ (pozycja stacji 7 w impulsach enkodera od punktu zero)

$m1 = 400$ (całkowita liczba impulsów enkodera na obrót)

Założmy, że stół obrotowy ma dotrzeć do stacji przeznaczenia z małą prędkością, zaczynając 1,5 stacji przed celem. Wtedy:

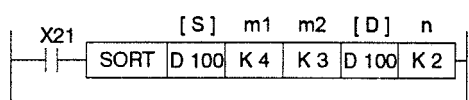
$m2 = 1,5 \times 50 = 75$ odległość małej prędkości od jakiegokolwiek strony stacji docelowej (w impulsach enkodera).

5.7.10 SORT (FNC 69)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty					Kroki programu
		S	m1	m2	D	n	
SORT FNC 69 (sortowanie danych w tabelach)	dane w zdefiniowanej tabeli są sortowane według wybranego pola	D	K, H ☒ uwaga: m1=1 do 32 m2=0 do 6		D	K, H ☒ uwaga: n=1 do m2	SORT: 11 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Instrukcja konstruuje tabelę danych z m1 rekordów o m2 polach pod adresem początkowym S. Następnie rekordy są sortowane względem danej w polu n, w porządku

numerycznym, z zachowaniem integralności rekordów. Powstała nowa tabela danych jest przechowywana pod adresem D.

Uwagi:

- Każdy rekord jest sortowany rosnąco, względem danej w wybranym polu n.
- Obszary S i D mogą się pokrywać, ale jeżeli są rozłączne, obszary zajmowane przez tablice nie powinny zachodzić na siebie.
- Kiedy operacja SORT została ukończona, znacznik M8029 jest ustawiany w stan ON. Aby posortować tablicę, instrukcja SORT będzie przetworzona m1 razy.
- Podczas operacji SORT dane w tabeli nie mogą być zmieniane, w przeciwnym razie wynik nie będzie poprawny.
- Instrukcja SORT może być użyta w programie **tylko raz**.

Na poniższym przykładzie pokazano manipulację danymi podczas sortowania tabeli, względem zdefiniowanego pola n.

Tabela oryginalna

		Pole(m2)		
		1	2	3
Rekord (m1)	1	(D100) 32	(D104) 162	(D108) 4
	2	(D101) 74	(D105) 6	(D109) 200
	3	(D102) 100	(D106) 80	(D110) 62
	4	(D103) 7	(D107) 34	(D111) 6

Tabela posortowana dla n=2

		Pole(m2)		
		1	2	3
Rekord (m1)	1	(D100) 32	(D104) 162	(D108) 4
	2	(D101) 74	(D105) 6	(D109) 200
	3	(D102) 100	(D106) 80	(D110) 62
	4	(D103) 7	(D107) 34	(D111) 6


Tabela posortowana dla n=1

		Pole(m2)		
		1	2	3
Rekord (m1)	1	(D100) 32	(D104) 162	(D108) 4
	2	(D101) 74	(D105) 6	(D109) 200
	3	(D102) 100	(D106) 80	(D110) 62
	4	(D103) 7	(D107) 34	(D111) 6

Notatki

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu 5-4
2. Instrukcje przesłania i porównania 5-16
3. Operacje arytmetyczne i logiczne 5-24
4. Rotacje i funkcje przesunięcia 5-34
5. Operacje na danych 5-42
6. Szybkie przetwarzanie 5-52
7. Instrukcje podręczne 5-66
-  8. Zewnętrzne urządzenia wejścia/wyjścia 5-80
9. Zewnętrzne urządzenia szeregowo 5-94
10. Zewnętrzne jednostki F2 5-112
11. Operacje zmiennoprzecinkowe 1 & 2 5-120
12. Operacje trygonometryczne 5-128
13. Operacje na danych 2 5-132
14. Sterowanie zegarem czasu rzeczywistego 5-136
15. Kody Graya 5-144
16. Porównania operacyjne 5-148

5.8 Zewnętrzne urządzenia wejścia/wyjścia - funkcje 70 - 79

Zawartość:

			Strona
TKY	klawiatura 10-klawiszowa	FNC70	5-81
HKY	klawiatura szesnastkowa	FNC 71	5-82
DSW	zadajnik cyfrowy	FNC 72	5-83
SEGD	dekoder 7-segmentowy	FNC 73	5-84
SEGL	dekoder 7-segmentowy z zatraskiem	FNC 74	5-85
ARWS	przełączniki kierunku	FNC 75	5-87
ASC	kod ASCII	FNC 76	5-88
PR	drukuj	FNC 77	5-89
FROM	czytaj z bloku funkcyjnego	FNC 78	5-90
TO	zapisz do bloku funkcyjnego	FNC 79	5-91



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1 , S_3 lub dla urządzeń stabilizowanych D_{3+0} , S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

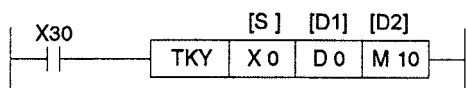
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.8.1 TKY (FNC 70)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D ₁	D ₂	
TKY FNC 70 (klawiatura 10- klawiszowa)	odczytuje 10 urządzeń jako pojedynczą liczbę całkowitą	X, Y, M, S uwaga: użyto 10 kolejnych urządzeń, oznaczonych jako S ₊₀ do S ₊₉	KnY, KnM, KnS, T, C, D, V, Z uwaga: dla operacji 32-bitowych użyto 2 kolejnych urządzeń	Y, M, S uwaga: użyto 11 kolejnych urządzeń, oznaczonych jako D ₂₊₀ do D ₂₊₁₀	TKY: 7 kroków DTKY: 13 kroków

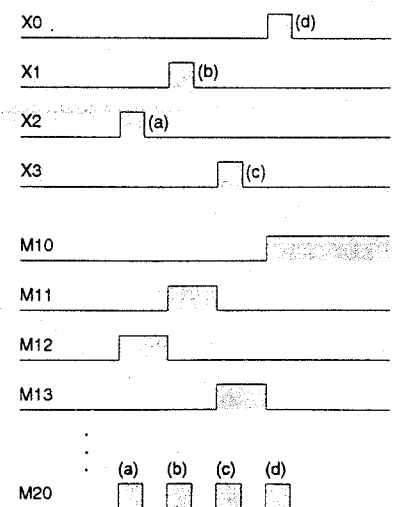
Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Ta instrukcja czyta 10 kolejnych urządzeń (S+0 do S+9) i przechowuje wprowadzony ciąg numeryczny w urządzeniu D1.

Uwagi:

- Kiedy urządzenie źródłowe staje się aktywne, skojarzone z nim bitowe urządzenie docelowe D2 również staje się aktywne i pozostaje aktywne aż do momentu, dopóki nie zostanie uaktywnione następne urządzenie źródłowe. Każde urządzenie źródłowe jest odwzorowane bezpośrednio w jego własnym urządzeniu D2, tzn. S+0 odwzorowuje się w D2+0, S+7 odwzorowuje się w D2+7 itd. Te urządzenia odwzorowują się bezpośrednio w wartościach dziesiętnych, które są następnie przechowane w urządzeniu docelowym określonym jako D1.
- Tylko jedno urządzenie źródłowe może być aktywne w dowolnym czasie. Urządzenie docelowe D2+10 jest używane do wskazania, że klawisz (jedno z 10 urządzeń źródłowych) został wciśnięty. D2+10 będzie aktywne tak długo, jak długo wciśnięty jest klawisz. Kiedy instrukcja TKY jest aktywna, każde przyciśnięcie klawisza dodaje cyfrę do liczby przechowanej w D1. Kiedy instrukcja TKY nie jest aktywna, wszystkie urządzenia D2 są zerowane, ale wartość danej w D1 jest zachowana.
- Kiedy instrukcja TKY jest użyta w operacji 16-bitowej, D1 może przechować liczby od 0 do 9.999, tzn. maksymalnie 4 cyfry. Kiedy jest użyta instrukcja DTKY (operacja 32-bitowa), można przechować wartości od 0 do 99.999.999 (maksymalnie 8 cyfr) w dwóch kolejnych urządzeniach D1 i D1 + 1. W obu przypadkach jeżeli liczba, która ma być przechowana przewyższa dostępny zakres (przepełnienie), najwyższe cyfry będą tracone aż do momentu osiągnięcia dopuszczalnej liczby. Cyfry przepełnienia są stracone i stają się niedostępne dla użytkownika. Zera wiodące nie są przechowywane, tzn. 0127 będzie przechowane jako 127.
- Instrukcja TKY może być użyta tylko raz.
- Użycie powyższej instrukcji w krótkim przykładzie: Jeżeli klawisze zdefiniowane jako (a) do (d) są wciśnięte w tej kolejności, liczba 2.130 będzie wprowadzone do D1. Jeżeli klawisz zdefiniowany jako (e) zostanie wciśnięty jako następny, wartość w D1 będzie równa 1.309. Początkowe "2" zostało stracone.



Klucze wejściowe i ich wartości dziesiętne



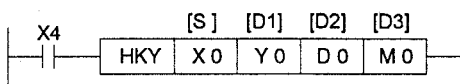
Przykład podłączenia kluczy

5.8.2 HKY (FNC 71)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D1	D2	D3	
HKY FNC 71 (klawiatura szesnastkowa)	multipleskuje wejścia i wyjścia tworząc klawiaturę numeryczną z 6 klawiszami funkcyjnymi	X uwaga: użyto 4 kolejnych urządzeń	Y uwaga: użyto 4 kolejnych urządzeń	T, C, D, V, Z uwaga: dla operacji 32-bitowych użyto 2 kolejnych urządzeń	Y, M, S uwaga: użyto 8 kolejnych urządzeń	HKY: 9 kroków DHKY: 17 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

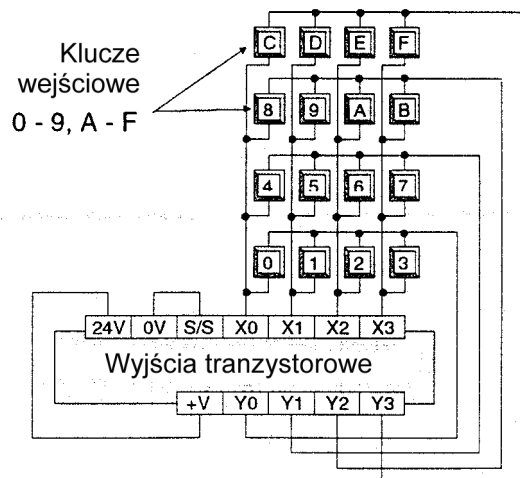
**Działanie 1 - Standardowe:**

Ta instrukcja tworzy multipleksowane połączenie 4 wyjść (D1) i 4 wejść (S) (matrycę) do odczytania w 16 różnych urządzeniach. Zapamiętane mogą być wartości dziesiętne

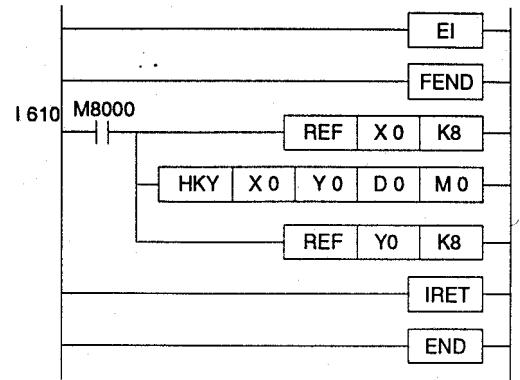
(0-9) oraz 6 dalszych znaczników funkcyjnych.

Uwagi:

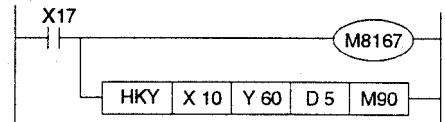
- Każde z pierwszych 10 multipleksowanych urządzeń źródłowych (zdefiniowanych jako 0-9) odwzorowuje bezpośrednio wartości od 0 do 9. Po naciśnięciu klawisza (0 - 9), tzn. kiedy urządzenie źródłowe jest aktywowane, wtedy skojarzona z nim wartość dziesiętna jest dodana do łańcucha danych, przechowywanego w D2. Naciśnięcie któregośkolwiek z tych klawiszy spowoduje, że urządzenie bitowe D3+7 będzie ustawione w stan ON na czas naciśnięcia klawisza.
- Pozostałe 6 multipleksowanych urządzeń źródłowych (zdefiniowanych jako klawisze funkcyjne A-F) jest używanych do ustawienia urządzeń bitowych, odpowiednio, D3+0 do D3+5. Te znaczniki bitowe, ustawione w stan ON, pozostają w tym stanie do momentu, kiedy zostanie naciśnięty następny klawisz funkcyjny. Naciśnięcie któregośkolwiek z tych klawiszy spowoduje, że urządzenie bitowe D3+6 będzie ustawione w stan ON na czas naciśnięcia klawisza.
- W przypadku naciśnięcia dwu lub więcej klawiszy, tylko klawisz naciśnięty jako pierwszy jest efektywny. Kiedy wykryte zostanie naciśnięcie klawisza, M8029 (znacznik ukończenia) ustawia się w stan ON. Kiedy instrukcja HKY jest nieaktywna, wszystkie urządzenia D3 są zerowane, ale wartość danej w D2 jest zachowana.
- Kiedy instrukcja HKY jest użyta w operacji 16-bitowej, D2 może przechować liczby od 0 do 9.999, tzn. maksymalnie 4 cyfry. Kiedy jest użyta instrukcja DHKY (operacja 32-bitowa), można zapamiętywać wartości od 0 do 99.999.999 (maksymalnie 8 cyfr) w dwóch kolejnych urządzeniach D2 i D2 + 1. W obu przypadkach jeżeli liczba, która ma być zapamiętana przewyższa dostępny zakres (przepełnienie), najwyższe cyfry będą tracone, aż do momentu osiągnięcia dopuszczalnej liczby. Cyfry przepełnienia są stracone i stają się niedostępne dla użytkownika. Zera wiodące nie są przechowywane, tzn. 0127 będzie przechowywane jako 127. Ta operacja jest podobna do instrukcji TKY.



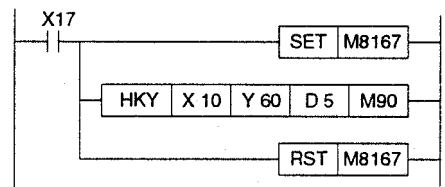
- e) Instrukcja HKY może być użyta tylko raz.
- f) Normalne działanie wymaga 8 przebiegów programu aby odczytać wejścia klawiszy. Do uzyskania stałych i powtarzalnych osiągnięć należy użyć trybu stałego przebiegu (stały cykl skanowania), tzn. znacznik M8039 jest ustawiony w stan ON, a zdefiniowany przez użytkownika czas przebiegu jest wpisany do rejestru D8039. Jednak, aby uzyskać szybszą odpowiedź, instrukcja HKY powinna być zaprogramowana w programie obsługi przerwania timera, jak pokazano na przykładzie obok.



Działanie 2 - Użycie instrukcji HKY ze znacznikiem M8167(Dostępne na jednostkach FX(2C) i FX2N) Kiedy instrukcja HKY jest używana ze znacznikiem M8167, ustawionym w stan ON (jak pokazano po prawej stronie), zmienia działanie klawiszy A-F, co umożliwi wprowadzanie wartości szesnastkowych, (0-9, A-F) do urządzenia danych D2. Jest to rozszerzenie do standardu klawiszy od 0 do 9. Wszystkie pozostałe działania, tak jak opisano w "Działaniu 1 - Standardowym". Maksymalny zakres wprowadzanych wartości dla tej operacji wynosi FFFF w trybie 16-bitowym i FFFFFFFF w trybie 32-bitowym.



Te dwa przykładowe programy wykonują to samo zadanie

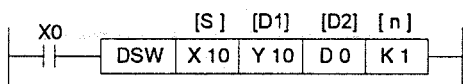


5.8.3 DSW (FNC 72)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty				Kroki programu
		S	D ₁	D ₂	n	
DSW FNC 72 (zadajnik cyfrowy)	multipleksowany odczyt n zestawów cyfrowych nastawników	X uwaga: jeśli n=2 wtedy 8 urządzeń, w przec. wyp. 4	Y uwaga: użyto 4 kolejnych urządzeń	T, C, D, V, Z uwaga: jeśli n=2 wtedy 2 urządzenia w przec. wyp. 1	K, H <input checked="" type="checkbox"/> uwaga: n=1 lub 2	DSW: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		



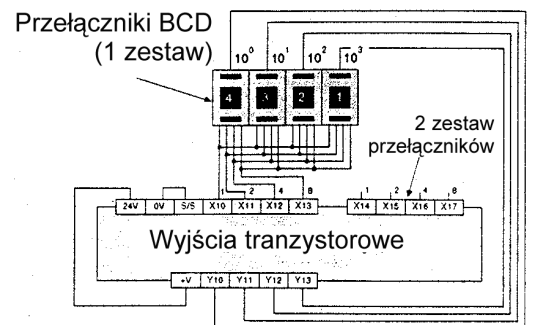
Działanie:
Ta instrukcja multipleksuje 4 wyjścia (D1) przez 1 lub 2 (n) zbiory

przełączników. Każdy zbiór przełączników zawiera 4 nastawniki, tworzące pojedyncze wejścia cyfrowe.

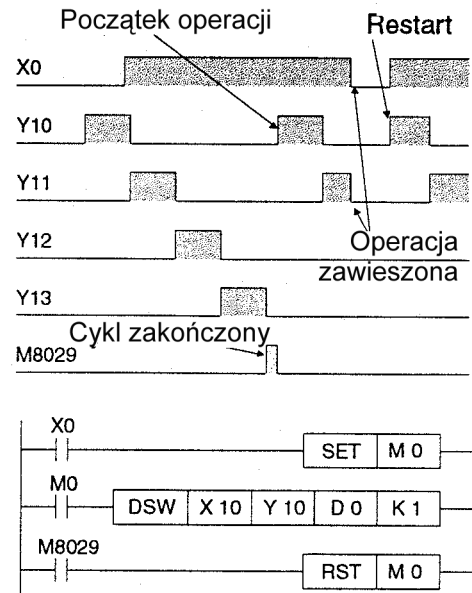
Uwagi:

- a) Kiedy n = 1, czytany jest tylko jeden zestaw przełączników. Multipleksowanie jest wykonane poprzez podłączenie nastawników równolegle do 4 kolejnych wejść od adresu początkowego, podanego w argumencie S. Wczytana dana (4 cyfry) jest przechowana w urządzeniu D2.

Ciąg dalszy na następnej stronie...



- b) Kiedy $n = 2$, czytane są dwa zestawy przełączników. Ta konfiguracja wymaga 8 kolejnych wejść od adresu początkowego określonego w argumencie S. Dane z pierwszego zestawu przełączników, tzn. używających pierwszych 4 wejść, są wczytywane do urządzenia danych D2. Dane z drugiego zestawu (znów 4 cyfry) są wczytywane do urządzenia danych D2+1.
- c) Wyjścia, użyte do multipleksowania, są sterowane cyklicznie, tak długo, jak działa instrukcja DSW. Po ukończeniu jednego odczytu ustawiany jest znacznik M8029. Liczba użytych wyjść nie zależy od liczby nastawników (n).
- d) Jeżeli instrukcja DSW jest zawieszona podczas cyklu, po ponownym uruchomieniu zacznie działać od początku cyklu a nie od ostatniego stanu.
- e) Zaleca się użycie z tą instrukcją jednostek z wyjściami tranzystorowymi. Jednak jeżeli użyje się techniki programowania, pokazanej po prawej stronie, można użyć jednostek z wyjściami przekaźnikowymi, ponieważ wyjścia nie będą aktywne w sposób ciągły.
- f) Instrukcja DSW może być użyta raz na sterownikach FX z wersją CPU niższą niż 3.07. W jednostkach FX z wersją CPU 3.07 lub późniejszą i w jednostkach FX2C instrukcja ta może być użyta maksymalnie dwa razy.

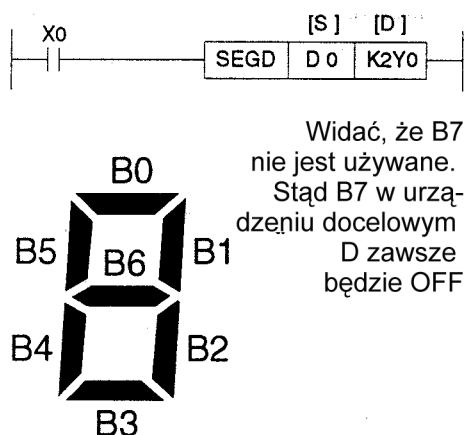


5.8.4 SEGD (FNC 73)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty		Kroki programu
		S	D	
SEGD FNC 73 (dekoder 7-segmentowy)	dana szesnastkowa jest dekodowana do formatu używanego w wyświetlaczach 7-segmentowych	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z uwaga: wykorzystywane tylko młodsze 4 bity	KnY, KnM, KnS, T, C, D, V, Z uwaga: starsze 8 bitów pozostaje nie zmienione	SEGD, SEGDP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		



Działanie:

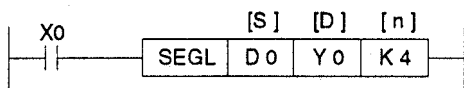
Pojedyncza cyfra szesnastkowa (0-9, A-F), zajmująca 4 niższe bity urządzenia źródłowego S, jest dekodowana do formatu, użytego do sterowania wyświetlaczem 7-segmentowym. Umożliwia to wyświetlanie reprezentacji cyfry szesnastkowej. Zdekodowana dana jest przechowywana w 8 niższych bitach urządzenia docelowego D. Pozostałych 8 bitów tego urządzenia nie jest wykorzystane. Rysunek obok pokazuje sterowanie wyświetlaczem 7-segmentowym poszczególnymi bitami. Aktywne bity odpowiadają tym ustawionym na "1" w urządzeniu docelowym D.

5.8.5 SEGL (FNC 74)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
SEGL FNC 74 (dekoder 7-segmentowy z zatraskiem)	zapisuje dane do multipleksowanych pojedynczych cyfr wyświetlacza - 4 cyfry na zestaw, max. 2 zestawy	K, H KnX, KnY, KnM, KnS T, C, D, V, Z	Y uwaga: n=0 do 3, użyto 8 wyjść n=4 to 7, użyto 12 wyjść	K, H ☒ uwaga: n=0 do 3 aktywny 1 zestaw n=4 to 7, aktywne 2 zestawy	SEGL: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

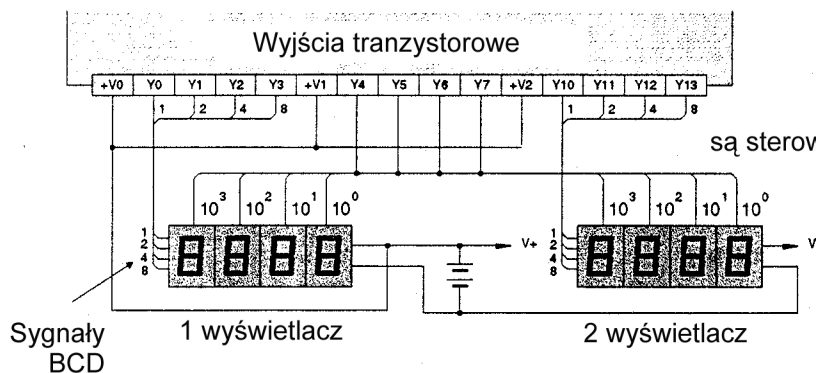
**Działanie:**

Instrukcja ta zapisuje daną dziesiętną ze źródła (S) do zbioru 4 multipleksowanych wyjść (D). Ze względu na to, że różni producenci oferują różne rozwiązania logiki zatraskowych wyświetlaczy 7-segmentowych, instrukcja daje się modyfikować, by obsłużyć większość dostępnych rozwiązań. Wybrane konfiguracje zależą od wartości argumentu (n), zobacz na następnej stronie.

segmentowych, instrukcja daje się modyfikować, by obsłużyć większość dostępnych rozwiązań. Wybrane konfiguracje zależą od wartości argumentu (n), zobacz na następnej stronie.

Uwagi:

- Dane są wpisywane do zbioru multipleksowanych wyjść (D+0-D+7, 8 wyjść). Zestaw wyświetlaczy składa się z 4 jednocyfrowych jednostek 7-segmentowych. Tą instrukcją mogą być sterowane maksymalnie 2 zestawy wyświetlaczy. Kiedy używane są 2 zestawy, wyświetlacze używają tych samych strobowanych wyjść (D+4-D+7). Aby dostarczyć nowe dane do drugiego zestawu wyświetlaczy, wymagany jest dodatkowy zbiór 4 urządzeń wyjściowych (D+10-D+13). Strobowane wyjścia powodują zatrzaśnięcie wpisanych danych w wyświetlaczu 7-segmentowym.
- Dane źródłowe z zakresu od 0 do 9.999 (dziesiętnie) są przepisywane na multipleksowane wyjścia. Kiedy używany jest jeden zestaw wyświetlaczy, dane te są odczytywane z urządzenia określonego jako argument S. Kiedy używane są dwa zestawy wyświetlaczy, danych dla drugiego zbioru dostarcza urządzenie źródłowe S1. Te dane muszą być z zakresu 0-9.999. Kiedy używane są dwa zestawy wyświetlaczy, dane są traktowane jako dwie oddzielne liczby i nie są łączone aby utworzyć pojedyncze wyjście z zakresu 0-99.999.999.
- Instrukcja SEGL potrzebuje 12 przebiegów programu do ukończenia jednego cyklu wyjścia, niezależnie od liczby użytych zbiorów wyświetlaczy. Po ukończeniu ustawiany jest znacznik M8029.



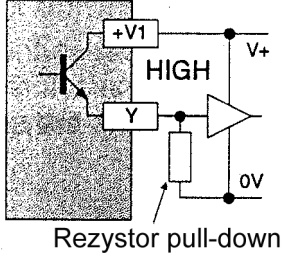
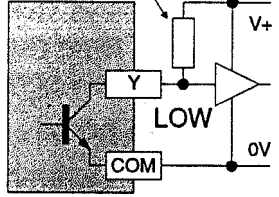
W tym przykładzie założono, że wyświetlacze 7-segmentowe są sterowane wysokim poziomem sygnału

- d) Jeżeli instrukcja SEGL jest zawieszona podczas cyklu, po ponownym uruchomieniu zacznie od początku cyklu a nie od ostatniego stanu.
- e) Instrukcja SEGL może być użyta raz na sterownikach FX z wersją CPU niższą niż 3.07. W jednostkach FX z wersją CPU 3.07 lub późniejszą i w jednostkach FX2C instrukcja ta może być użyta maksymalnie **dwa** razy.

Wybór odpowiedniej wartości dla argumentu n

Wybór parametru n zależy od 4 czynników:

- 1) Typ logiki, używanej na wyjściach PLC
- 2) Typ logiki, używanej na liniach danych 7-segmentowych
- 3) Typ logiki 7-segmentowego sygnału strobowego
- 4) Liczba używanych zbiorów wyświetlaczy

Omawiane urządzenie		Logika pozytywna	Logika negatywna
Logika PLC		<p>Wyjście ze źródłem</p>  <p>Rezystor pull-down</p>	<p>Wyjście pływające</p> <p>Rezystor pull-up</p> 
		Na wyjściu ze źródłem, gdy wyjście jest w stanie HIGH, logika wewnętrzna wynosi '1'	Na wyjściu pływającym, gdy wyjście jest w stanie LOW, logika wewnętrzna wynosi '1'
Logika wyświetlaczy 7-segmentowych	Logika sygnału strobowego	Dane są zatrzaśnięte i przechowywane, gdy sygnał jest HIGH	Dane są zatrzaśnięte i przechowywane, gdy sygnał jest LOW
	Logika sygnału danych	Aktywne linie danych są w stanie HIGH, tzn. mają logiczną wartość '1'	Aktywne linie danych są w stanie LOW, tzn. mają logiczną wartość '1'

Dostępne są dwa typy logiki, pozytywna i negatywna. W zależności od typu systemu, tzn. które elementy mają logikę pozytywną, a które negatywną, wartość n może być wybierana z poniższej tabeli w zależności od liczby użytych zestawów wyświetlaczy 7-segmentowych.

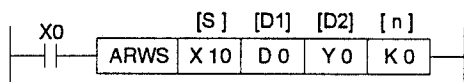
Logika PLC	Logika wyświetlacza		n	
	Logika danych	Logika strobu	1 zestaw	2 zestawy
Pozytywna (źródło)	Pozytywna (High)	Pozytywna (High)	0	4
Negatywna (pływ.)	Negatywna (Low)	Negatywna (Low)		
Pozytywna (źródło)	Pozytywna (High)	Negatywna (Low)	1	5
Negatywna (pływ.)	Negatywna (Low)	Pozytywna (High)		
Negatywna (pływ.)	Pozytywna (High)	Negatywna (Low)	2	6
Pozytywna (źródło)	Negatywna (Low)	Pozytywna (High)		
Negatywna (pływ.)	Pozytywna (High)	Pozytywna (High)	3	7
Pozytywna (źródło)	Negatywna (Low)	Negatywna (Low)		

5.8.6 ARWS (FNC 75)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D ₁	D ₂	n	
ARWS FNC 75 (przełączniki kierunku)	tworzy zdefiniowana przez użytkownika 4-klawiszowy panel wejściowy	X, Y, M, S uwaga: użyte 4 kolejnych urządzeń	T, C, D, V, Z uwaga: dane pamiętane w formacie binarnym	Y uwaga: użyte 8 kolejnych urządzeń	K, H ☒ uwaga: n=0 do 3	ARWS: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

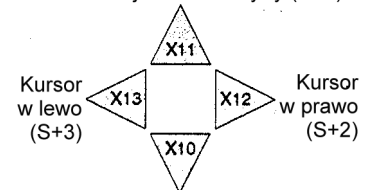
**Działanie:**

Ta instrukcja wyświetla zawartość pojedynczego urządzenia danych D1 na zestawie 4 cyfrowego wyświetlacza 7-segmentowego. Dane w D1 są w formacie dziesiętnym, ale są automatycznie konwertowane na kod BCD dla wyświetlenia na wyświetlaczu 7-segmentowym. Każda cyfra wyświetlanej liczby może być wybierana i edytowana. Procedura edycji zmienia bezpośrednio zawartość urządzenia określonego jako D1.

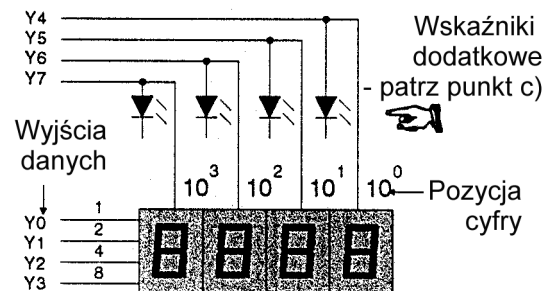
Uwagi:

- Dane przechowywane w urządzeniu D1 mogą mieć wartość z zakresu 0-9.999 (dziesiętnie), tzn. 4-cyfrowa liczba dziesiętna. Wartość każdej cyfry może być zwiększana (S+1) lub zmniejszana (S+0) poprzez naciśnięcie odpowiedniego klawisza. Edytowane liczby automatycznie zmieniają się z 9 - 0 - 1 i 1 - 0 - 9. Cyfra jest wyświetlana przez młodsze 4 urządzenia bitowe z D2, tzn. D2+0-D2+3
- Przy rozpoczęciu instrukcji ARWS wybrana jest cyfra pozycji 103. Każda pozycja cyfry może być sekwencyjnie wybierana kursorem w kierunku lewym (S+2) i prawym (S+3). Po osiągnięciu ostatniej cyfry instrukcja ARWS automatycznie przenosi pozycje kursora, tzn. po pozycji 103 jest wybierana pozycja 100 i odwrotnie. Każda cyfra jest fizycznie wybierana przez różne strobowane wyjścia.
- Aby wspomóc użytkownika przy obsłudze panelu sterowania przy pomocy instrukcji ARWS, zaleca się podłączenie dodatkowych lampek kontrolnych, równoległe do strobowanych wyjść dla każdej cyfry. To wskazałoby, która cyfra została wybrana do edycji.
- Parametr n ma tę samą funkcję, co parametr n w instrukcji SEGL - zobacz strona 5-86 - Wybór odpowiedniej wartości dla argumentu n. Uwaga: ponieważ instrukcja ARWS steruje tylko jednym zbiorem wyświetlaczy, n może przyjmować tylko wartości 0-3.
- Instrukcja ARWS może być użyta **tylko raz**. Ta instrukcja powinna być użyta tylko na wyjściach tranzystorowych sterowników PLC.

Inkrementacja wartości cyfry (S+1)



Dekrementacja wartości cyfry (S+0)



5.8.7 ASC (FNC 77)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
ASC FNC 76 (kod ASCII)	ciąg alfanumeryczny jest konwertowany na kody ASCII	dane alfanumeryczne np. 0-9, A-Z i a-z uwaga: jednorazowo można wprowadzić tylko jeden 8-znakowy ciąg	T, C, D uwaga: użyte 4 kolejne urządzenia	ASC: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

Działanie:

Źródłowy ciąg danych S zawiera do 8 znaków z drukowalnego zakresu zbioru znaków ASCII. Jeżeli użytych jest mniej niż 8 znaków, reszta jest wypełniona znakiem NULL (ASCII 00). Dana źródłowa jest konwertowana do kodów ASCII. Kody są następnie przechowane w urządzeniu docelowym D, zobacz przykład poniżej.

D	Bajt	
	starszy	młodszy
D300	58 (X)	46 (F)
D301	36 (6)	2D (-)
D302	4D (M)	34 (4)
D303	21 (!)	52 (R)

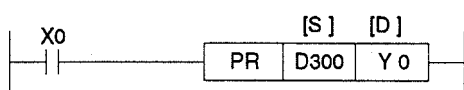
Uwaga: Znaki ASCII **nie mogą** być wprowadzane z podręcznego programatora.

5.8.8 PR (FNC 77)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
PR FNC 77 (drukuj)	wyprowadza dane ASCII na urządzenie typu wyświetlacz	T, C, D uwaga: tryb 8-bajtowy (M8027=OFF) używa 4 kolejnych urządzeń tryb 16-bajtowy (M8027=ON) używa 8 kolejnych urządzeń	Y uwaga: użyto 10 kolejnych urządzeń	PR: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	operacja zakończona M8029
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

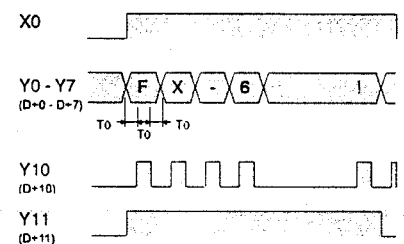
Dana źródłowa (przechowywana jako wartości ASCII) jest odczytywana bajt po bajcie z urządzenia źródłowego.

Każdy bajt jest odwzorowany bezpośrednio do pierwszych

8 kolejnych urządzeń docelowych (D+0 - D+7). Dwa końcowe bity to sygnał strobujący (D+10, ósemkowo) i znacznik zajętości (D+11, ósemkowo).

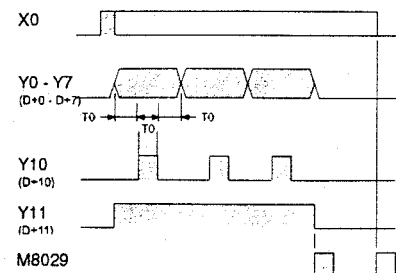
Uwagi:

- Bajtowa dana źródłowa odwzorowuje najniższy bit do pierwszego urządzenia docelowego D+0. Konsekwentnie najwyższy bit bajtu jest wysłany do urządzenia docelowego D+7.
- Instrukcja PR może być użyta raz na sterownikach FX z wersją CPU niższą niż 3.07. W jednostkach FX z wersją CPU 3.07 lub późniejszą i w jednostkach FX2C instrukcja ta może być użyta maksymalnie dwa razy. Ta instrukcja powinna być używana tylko dla PLC z wyjściami tranzystorowymi. Instrukcja PR będzie automatycznie powtarzała swoje działanie, chyba że wejście sterujące zostanie ustawione w stan OFF, a później w stan ON.
- Działanie instrukcji PR zależy od przebiegu programu. W normalnych warunkach wysłanie jednego bajtu trwa przez 3 przebiegi programu. Jednak aby uzyskać szybszy przebieg operacji, instrukcja PR powinna być zaprogramowana w programie obsługi przerwania timera, podobnie jak dla instrukcji HKY na stronie 5-82.
- Operacja 8-bajtowa ma następujący przebieg czasowy. Należy zauważyć, że kiedy wejście sterujące (w przykładzie X0) jest ustawione w stan OFF, instrukcja PR przerwie swoje działanie. Po ponownym uruchomieniu, instrukcja PR zacznie się od początku ciągu. Po przesłaniu wszystkich 8 bajtów, znacznik zajętości jest zerowany, a instrukcja PR zostaje zawieszona.
- Operacja 16-bajtowa wymaga ustawienia znacznika M8027 w stan ON (zaleca się aby jako wejście sterujące był użyty znacznik M8000). W tym trybie działania wejście sterujące (w przykładzie było X0) nie musi być aktywne przez cały czas. Po uaktywnieniu instrukcji PR będzie ona działała w sposób ciągły aż do momentu kiedy zostaną przesłane wszystkie 16 bajtów danych lub wartość 00h (NULL). Po zakończeniu działania znacznik zajętości (D+11, ósemkowo) jest ustawiony w stan OFF, a znacznik M8029 w stan ON.



Uwaga:

To=czas przebiegu programu

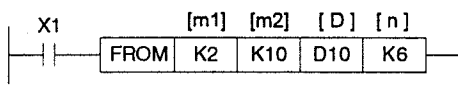


5.8.9 FROM (FNC 78)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		m1	m2	D	n	
FROM FNC 78 (FROM)	czyta dane z przyłączonego bloku funkcyjnego	K, H ☒ uwaga: m1= 0 do 7	K, H ☒ uwaga: m1= 0 do 31	KnY, KnM, KnS, T, C, D, V, Z	K, H ☒ uwaga: operacja 16-bitowa: n=1 do 32 operacja 32-bitowa: n=1 do 16	FROM, FROMP: 9 kroków DFROM, DFROMP: 17 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

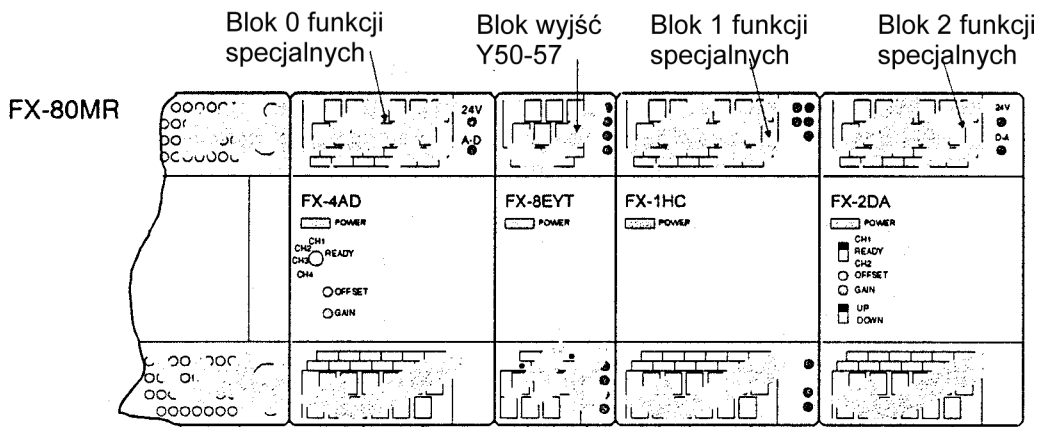
**Działanie:**

Instrukcja FROM czyta n słów danych z bufora pamięciowego zaczynając od adresu m2 bloku funkcyjnego umieszczonego na pozycji określonej przez m1.

Odczytywane dane są umieszczane w PLC od adresu początkowego D dla n urządzeń słowowych.

Uwagi:

- a) Wszystkie bloki funkcyjne, adresowane instrukcjami FROM/TO są połączone magistralą rozszerzającą z prawej strony PLC. Każdy blok funkcyjny może być wstawiony w dowolnym punkcie łańcucha jednostek rozszerzających (tak długo, jak zachowane są reguły konfiguracji systemu). Każdy blok jest kolejno adresowany, od 0 do 7, począwszy od bloku najbliższego jednostce podstawowej.



- b) Każda jednostka funkcyjna ma inne rejestry bufora pamięciowego. Rejestry te często są dedykowane dla każdej indywidualnej jednostki. Przed odczytaniem lub zapisaniem danych należy się upewnić, że znane są prawidłowe miejsca buforów pamięciowych dla używanej jednostki.
m2 określa adres początkowy dostępnego bufora pamięciowego; może mieć wartość z zakresu 0-31.
n określa liczbę przenoszonych słów, pomiędzy blokiem funkcyjnym i jednostką podstawową PLC; może mieć wartość 1-31 dla operacji 16-bitowych i 1-16 dla operacji 32-bitowych
- c) Docelowy adres początkowy danych czytanych z bloku funkcyjnego jest określony w argumencie D i zajmuje n kolejnych urządzeń
- d) Instrukcja będzie działała tylko wtedy, gdy wejście sterujące jest aktywne.

- e) Użytkownicy sterowników FX mogą wybrać, czy zezwolić na natychmiastowe pojawianie się przerw, tzn. w czasie trwania instrukcji FROM/TO, lub czekać do zakończenia działania bieżącej instrukcji FROM/TO. Osiąga się to przez sterowanie znacznikiem M8028. Następująca tabela pokazuje określone punkty, związane z tym sterowaniem i działaniem.

Przerwania zablokowane	Przerwania odblokowane
M8028 = OFF	M8028 = ON
Skoki powodowane przez przerwania są opóźniane do chwili zakończenia transferu danych przez instrukcje FROM/TO	Skoki powodowane przez przerwania wykonywane są natychmiast
W najgorszym przypadku można spodziewać się małego opóźnienia (800 m+ 200) μs Uwaga: m = liczba słów 32-bitowych	Transfer danych jest wstrzymywany do chwili powrotu z programu przerwania. Nie jest to pożądane, gdy instrukcje FROM./TO są zawarte w programie wywoływanego przerwania
Instrukcje FROM/TO zawarte w programie przerwania nie wchodzi w interakcję z czymkolwiek	Znacznik M8028 powinien być używany tylko wtedy, gdy wymagane jest bardzo krótkie opóźnienie w aplikacjach uwarunkowanych czasowo

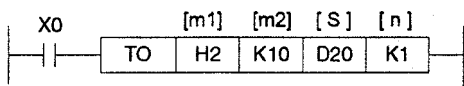
Użytkownicy FX_{0N} nie mają opcji dla przerw instrukcji FROM/TO i dlatego zawsze pracują w trybie równoważnym stanowi znacznika M8028 OFF.

5.8.10 TO (FNC 79)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		m1	m2	S	n	
TO FNC 79 (TO)	zapisuje dane do przyłączonego bloku funkcyjnego	K, H ☒ uwaga: m1= 0 do 7	K, H ☒ uwaga: m1= 0 do 31	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K, H ☒ uwaga: operacja 16- bitowa: n=1 do 32 operacja 32- bitowa: n=1 do 16	TO, TOP: 9 kroków DTO, DTOP: 17 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie:

Instrukcja TO zapisuje n słów danej do bufora pamięciowego pod adres początkowy m2 bloku funkcyjnego umieszczonego na pozycji określonej przez

m1. Zapisywane dane pobierane są z PLC od adresu początkowego S dla n urządzeń słowowych.

Uwagi:


Wszystkie punkty są takie same jak dla instrukcji FROM (zobacz poprzednią stronę, oprócz punktu c), który brzmi następująco:

- c) Adres początkowy źródła dla danych zapisywanych w bloku funkcyjnym jest określony argumentem S.

Notatki

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
 9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.9 Zewnętrzne urządzenia szeregowo - funkcje 80 - 89

Zawartość:

			Strona
RS	komunikacja RS	FNC 80	5-95
PRUN	praca równoległa FX2-40A☆	FNC 81	5-99
ASCII	konwersja szesnastkowa do ASCII	FNC 82	5-101
HEX	konwersja ASCII na szesnastkową	FNC 83	5-102
CCD	kod kontrolny	FNC 84	5-103
VRRD	odczyt wartości FX-8AV	FNC 85	5-104
VRSC	odczyt zakresu FX-8AV	FNC 86	5-104
☆☆☆	brak	FNC 87	
PID	pętla sterująca PID	FNC 88	5-105
☆☆☆	brak	FNC 89	



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1 , S_3 lub dla urządzeń stabilizowanych D_{3+0} , S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

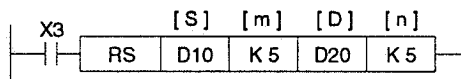
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.9.1 RS (FNC 80)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	m	D	n	
RS FNC 80 (komunikacja RS)	używana do sterowania komunikacją szeregową z/do PLC	D (w tym rejestry plikowe)	K, H, D ☒	D	K, H, D ☒	RS: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	opóźnienie nadawania M8121 żądanie nadawania M8122 dane odebrane M8123
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Instrukcja zapewnia bezpośrednie sterowanie komunikacją pomiędzy FX i adapterami komunikacyjnymi FX0N, przyłączonymi do lewego portu jednostki głównej MPU,

tzn. FX0N-232ADP, FX-232ADP itd.

Uwagi:

- Instrukcja zawiera wiele automatycznie definiowanych urządzeń. Są one przedstawione po prawej stronie.
- Instrukcja ma 2 części, nadawanie (transmisja) i odbiór. Pierwsze elementy instrukcji RS określają bufor transmitowanych danych (S) jako adres początkowy, który zawiera m elementów w stosie sekwencyjnym. Ostatnie 2 parametry instrukcji RS określają obszar danych odbieranych. Przeznaczenie (D) dla otrzymywanych wiadomości ma bufor lub stos o długości n elementów. Rozmiar buforów nadawania i odbioru określa, jak duża może być pojedyncza wiadomość. Rozmiary buforów mogą być aktualizowane w następujących okresach czasu:
 - Bufor nadawania - zanim pojawi się transmisja, tzn. zanim M8122 będzie ustawiony w stan ON.
 - Bufor odbioru - po otrzymaniu wiadomości, ale zanim M8123 będzie wyzerowany.
- Dane nie mogą być wysłane podczas otrzymywania wiadomości, transmisja będzie opóźniona - zobacz M8121
- Można zaprogramować więcej niż jedną instrukcję RS, ale tylko jedna może być w danym momencie aktywna.

Przyporządkowanie urządzeń:**Urządzenia danych:**

D8120 - zawiera konfigurację parametrów dla komunikacji, tzn. prędkość transmisji, ilość bitów stopu, itd. Pełne informacje na następnej stronie.
 D8122 - zawiera liczbę bajtów pozostałych do wysłania w aktualnie wysyłanej wiadomości
 D8123 - zawiera liczbę otrzymanych bajtów w odbieranej wiadomości
 D8124 - zawiera kod ASCII znaku używanego do określenia nagłówka wiadomości - domyślnie "STX", 02 HEX
 D8125 - zawiera kod ASCII znaku używanego do zakończenia wiadomości - domyślnie "ETX", 03 HEX

Znaczniki:

M8121 - jest w stanie ON, dla wskazania, że transmisja będzie opóźniona do momentu zakończenia aktualnej operacji odbioru.
 M8122 - gdy jest ustawiony w stan ON, powoduje wyzwolenie transmisji danych,
 M8123 - jest w stanie ON, gdy odbiór wiadomości został zakończony
 M8124 - obecność nośnej; używany przez MPU FX i FX2C w komunikacji modemowej
 M8161 - tryb pracy 8-bitowy lub 16-bitowy; stan ON - tryb 8-bitowy, używane jest tylko dolne 8 bitów urządzenia, tzn. jeden znak ASCII jest przechowywany w jednym rejestrze; stan OFF - tryb 16-bitowy, wykorzystywane są pełne rejestry urządzeń, tzn. w każdym rejestrze przechowywane są dwa znaki ASCII.

Specyfikacja parametrów komunikacji:

Te parametry muszą być ustawione przed rozpoczęciem komunikacji pomiędzy dwoma urządzeniami szeregowymi. Elementy takie jak prędkość transmisji, ilość bitów stopu, parzystość muszą być identycznie ustawione pomiędzy dwoma urządzeniami komunikacyjnymi. Parametry komunikacyjne są wybierane przez wzorzec bitowy, który jest przechowywany w rejestrze danych D8120, np. jeżeli D8120 byłby równy 0F9Eh wtedy ustawiono by następujące parametry:
EH = dana 7-bitowa, parzystość, 2 bity stopu,
9H = prędkość transmisji 19200 bps
FH = znak nagłówka,
znak końca transmisji,
sprzętowa kontrola transmisji typ 1 (handshaking type 1)
tryb linii sterujących - pojedynczy
0H = typ 2 kontroli transmisji (handshaking type 2)
- wyłączony

D8120				
	Opis	Status bitu		
		0(OFF)	1(ON)	
b0	Długość słowa danych	7 bitów	8 bitów	
b1 b2	Parzystość (b2, b1)	(00) - brak (01) - nieparzystość (11) - parzystość		
b3	Liczba bitów stopu	1	2	
b4 b5 b6 b7	prędkość transmisji [bps] (b7, b6, b5, b4)	(0011): 300 (0100): 600 (0101): 1200 (0110): 2400 (0111): 4800 (1000): 9600 (1001): 19200		
☆b8	znak nagłówka	brak	D8124	
☆b9	znak końca transmisji	brak	D8125	
☆b10	handshake typ 1	brak	H/W 1	
☆b11	tryb (linia sterująca)	normalny	pojedynczy	
☆b12	handshake typ 2	brak	H/W 2	
*b13	sieć FX-485	suma kontrolna	bez sprawdzania	dodana
*b14		sieć	wył.	aktywny
*b15		protokół	Format 1	Format 4



Urządzenia używane przez sieć FX-485, zaznaczone *, są używane zamiast tych oznaczonych ☆; więcej informacji w "FX-485PC-IF User's Manual (JY992D53501)".

Znaki nagłówka i końca transmisji mogą być zmienione przez użytkownika w zależności od potrzeb. Domyślnym ustawieniem dla nagłówka przechowanym w D8124 jest znak 'STX' (02h), a znaku końca transmisji (w D8125) jest 'ETX' (03h). Te wartości są przechowane w młodszym bajcie odpowiedniego rejestru.

Znaki nagłówka i końca transmisji są automatycznie dodawane do wysyłanej wiadomości w czasie nadawania. Podczas odbioru dane są ignorowane do momentu odbioru znaku nagłówka. Dane będą teraz czytane w sposób ciągły, do momentu wykrycia znaku końca transmisji lub wypełnienia bufora odbioru. Jeżeli bufor będzie wypełniony zanim zostanie odebrany znak końca transmisji, wtedy wiadomość jest uważana za niekompletną.

Jeżeli nie używa się znaku końca transmisji, wtedy odczytywanie będzie kontynuowane do momentu wypełnienia bufora odbioru. Tylko w tym punkcie wiadomość będzie zaakceptowana i kompletna. Nie będzie możliwe dalsze buforowanie komunikacji, stąd jeżeli przesłanych jest więcej danych niż wynosi rozmiar bufora, wtedy nadwyżka będzie stracona, ponieważ bufor jest pełny.

W takim razie ważne jest aby bufor odbioru miał taki sam rozmiar jak najdłuższa wiadomość, która ma być odebrana.



Ogólna informacja dotycząca parametrów kontroli transmisji:

Jednostki MPU sterownika FX0N mają tylko kontrolę typu 1 (b10), podczas gdy jednostki FX i FX2C mają typ 1 i typ 2 (b12). Jednakże tylko jeden typ może być wybrany w danej chwili.



Ogólna informacja dotycząca użycia rejestru danych D8120:

Ten rejestr jest używany jako rejestr ogólnych ustawień dla wszystkich jednostek ADP. Kiedy używane są jednostki 232ADP, urządzenia bitowe od *b13 do *b15 nie powinny być używane. Kiedy używana jest sieć FX-485 z odpowiednimi jednostkami 485ADP, urządzenia bitowe od *b13 do *b15 powinny być używane zamiast konfiguracji bitów od ☆b8 do ☆b12.

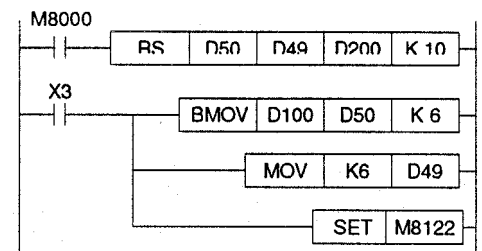
Podstawowe przebiegi czasowe dla różnych metod kontroli transmisji

PLC	D8120 Wybór metody sprzętowego potwierdzenia	Nadawanie	Odbiór
FX _{0N} , FX, FX _{2C} i FX _{2N}	b10: 1 (ON) b11: 0 (OFF) typ potwierdzenia 1 tryb normalny	SD ^{Nadawanie} DATA ER DR	RD ^{Odbiór} DATA ER DR
	b10: 1 (ON) b11: 1 (ON) typ potwierdzenia 1 tryb pojedynczej linii	SD ^{Nadawanie} RD ^{Odbiór} ER DR	SD ^{Nadawanie} RD ^{Odbiór} ER DR
FX, FX _{2C} i FX _{2N}	b12: 1 (ON) b11: 0 (OFF) typ potwierdzenia 2 tryb normalny	SD ^{Nadawanie} DATA ER DR	RD ^{Odbiór} DATA ER DR
	b12: 1 (ON) b11: 1 (ON) typ potwierdzenia 2 tryb pojedynczej linii	SD ^{Nadawanie} RD ^{Odbiór} ER DR	SD ^{Nadawanie} RD ^{Odbiór} ER DR

Uwaga: SD = (TxD), RD = (RxD), ER = (DTR), DR = (DSR)

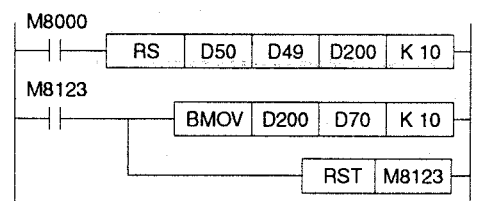
Zdarzenia potrzebne do ukończenia transmisji:

Instrukcja RS powinna być ustawiona i aktywna. Dane, które mają być transmitowane, powinny być umieszczone w buforze transmisji. Jeżeli do określenia długości wiadomości w instrukcji RS jest użyta zmienna, powinna być ustawiona na długość nowej wiadomości. Znacznik nadawania M8122 powinien następnie być ustawiony w stan ON. Jest on automatycznie zerowany po wysłaniu wiadomości - przykład programu po prawej stronie.

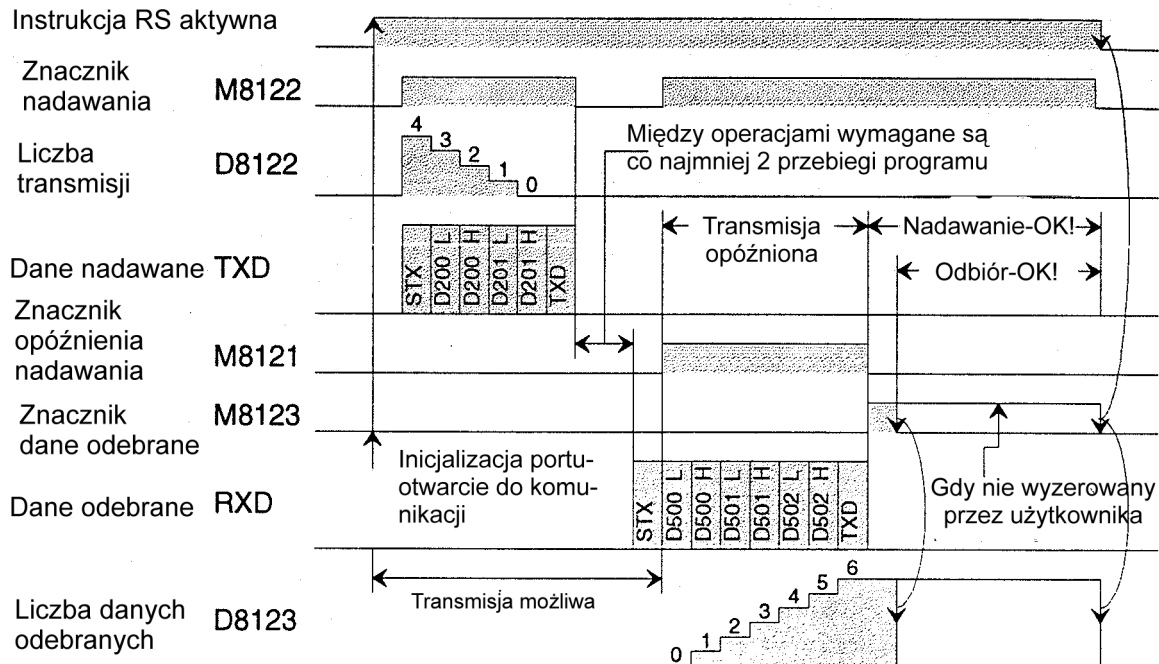
**Zdarzenia powstałe podczas odbierania wiadomości:**

Instrukcja RS powinna być ustawiona i aktywna. Po odebraniu danej i próbie wysłania danej, znacznik M8121 jest ustawiony w stan ON, sygnalizując opóźnienie transmisji. Po skompletowaniu odebranej wiadomości, znacznik M8123 jest ustawiany w stan ON. W tym samym czasie, jeżeli M8121 był ustawiony w stan ON, jest automatycznie zerowany, umożliwiając transmisję następnych wiadomości (opóźnionych bądź nie).

Zaleca się usuwać odebrane dane z bufora odbioru tak szybko jak to jest możliwe. Po wykonaniu tego M8123 powinien być wyzerowany przez użytkownika. Jest wtedy gotów na wysłanie lub przyjęcie nowej wiadomości.



Ogólny przebieg czasowy komunikacji



Opóźnienie podwójnego przebiegu programu (2 x cykl skanowania) wymagane w urządzeniu RS232 pomiędzy transmisjami ma umożliwić instrukcji RS ukończenie przetwarzania. To opóźnienie nie jest wymagane w jednostkach MPU FX2N.

Instrukcja RS może być używana przez jakiekolwiek bloki funkcyjne ADP lub tablice BD. Komunikacja może być nawiązana między takimi elementami jak drukarki, komputery, czytniki kodu kreskowego i modemy FX-232ADP.

Komunikacja 8- lub 16-bitowa

Jest ona przełączana przy użyciu znacznika M8161. Kiedy znacznik jest w stanie OFF, ma miejsce komunikacja 16-bitowa. Oznacza to, że w transmisji i odbiorze wiadomości używane są oba bajty 16-bitowego urządzenia danych. Jeżeli znacznik M8161 jest w stanie ON, wtedy wybierany jest tryb 8-bitowy. W tym trybie tylko 8 młodszych bitów (bajt) jest używanych do wykonania operacji nadawania-odbierania. Przełączenie znacznika M8161 powinno nastąpić tylko wtedy, kiedy instrukcja RS nie jest aktywna, tzn. jest w stanie OFF.

Kiedy obszar bufora jest określony w instrukcji RS, należy sprawdzić, czy został wybrany właściwy tryb, tzn. obszar bufora określony jako D50 K3 da następujące wyniki:

Tryb 16-bitowy - M8161 = OFF		
Rejestr	Starszy bajt	Młodszy bajt
D50	X	F
D51	---	0
---	---	---

Tryb 8-bitowy - M8161 = ON		
Rejestr	Starszy bajt	Młodszy bajt
D50	---	F
D51	---	X
D52	---	0



Ogólna informacja dotycząca sprzętu:

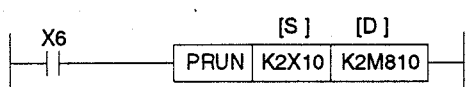
Informacja dotycząca wyprowadzeń odpowiednich bloków funkcyjnych ADP może być znaleziona razem ze szczegółami, dotyczącymi okablowania w odpowiednich podręcznikach dotyczących sprzętu.

5.9.2 PRUN (FNC 81)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
PRUN FNC 81 (transmisja równoległa)	używana do sterowania równoległymi adapterami serii FX: FX2-40AW/AP	KnX, KnM uwaga: n = 1 do 8 dla łatwego i wygodnego użycia bit adresu początkowego powinien być wielokrotnością 10, tzn. X10, M100, Y30 itd.	KnY	PRUN, PRUNP: 5 kroków DPRUN, DPRUNP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

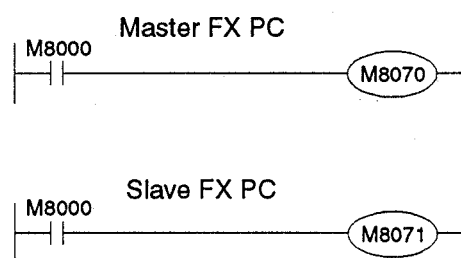
**Działanie:**

Ta instrukcja jest używana z adapterami FX połączeń równoległych. Umożliwia ona przesunięcie danych źródłowych do obszaru transmisji bitowych. Sterowanie komunikacją połączeń równoległych odbywa się przy pomocy znaczników M.

komunikacją połączeń równoległych odbywa się przy pomocy znaczników M.

Uwagi:

- a) Komunikacja połączeń równoległych odbywa się automatycznie, kiedy oba systemy są połączone a znaczniki stacji Master (M8070) i stacji Slave (M8071) są ustawione w stan ON (komunikacja nie wymaga użycia instrukcji PRUN). W systemie łączącym tylko dwa PLC może wystąpić tylko jedna stacja Master i jedna stacja Slave. Programy pokazane obok powinny być włączone do odpowiednich programów PLC.



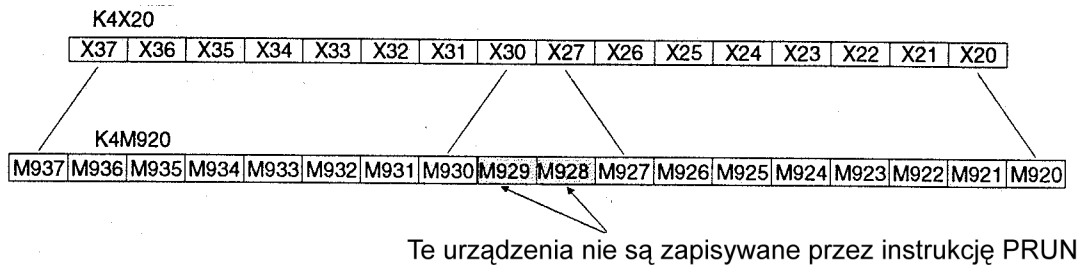
Znaczniki stacji, po ustawieniu, mogą być wyzerowane tylko gdy PLC jest w trybie STOP lub przez wyłączenie i włączenie zasilania sterownika.

- b) Podczas automatycznej komunikacji pomiędzy sterownikami Master i Slave są wymieniane następujące dane :

Stacja Master		Kierunek komunikacji	Stacja Slave	
M8070=ON	Dane bitowe		Dane bitowe	M8071=ON
	M800 do M899 (100 szt.)	→	M800 do M899 (100 szt.)	
	M900 do M999 (100 szt.)	←	M900 do M999 (100 szt.)	
	Dane słowowe	Dane słowowe		
	D490 do D499 (10 szt.)	→	D490 do D499 (10 szt.)	
D500 do D509 (10 szt.)	←	D500 do D509 (10 szt.)		

Ciąg dalszy nastąpi...

- c) Instrukcja PRUN umożliwia przesunięcie danych do obszaru transmisji bitowej lub z obszaru odebranych danych bitowych. Instrukcja różni się od instrukcji przesuwania tym, że działa ósemkowo. Oznacza to, że jeżeli K4X20 byłoby przesunięte przy użyciu instrukcji PRUN do K4M920, dane nie byłyby zapisane w M928 i M929, ponieważ te urządzenia mieszczą się poza ósemkowym systemem liczenia. Można to zobaczyć na diagramie poniżej.



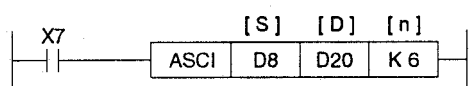
- d) Więcej informacji na stronie 9-6.

5.9.3 ASCII (FNC 82)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
ASCII FNC 82 (konwersja HEX na ASCII)	konwertuje dane z formatu HEX na ASCII	K, H, KnX, KnY, KnM, KnS T, C, D, V, Z	KnY, KnM, KnS T, C, D	K, H uwaga: n=1 do 256 <input checked="" type="checkbox"/>	ASCII, ASCIP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja czyta n szesnastkowych znaków danych od źródłowego adresu początkowego (S), konwertuje je do równoważnego kodu ASCII i umieszcza w urządzeniu

docelowym (D) jako n bajtów.

Uwagi:

Należy zauważyć, że dane są konwertowane tak jak są czytane, tzn. jak pokazano w przykładzie powyżej, z następującymi danymi w (D9, D8) ABCDh, **EF26h**. Biorąc pierwsze n znaków szesnastkowych (cyfr) z prawej (w tym przypadku n = 6), po konwersji do postaci ASCII, wartości te zostaną przechowane w 6 kolejnych bajtach, od D20, tzn. odpowiednio D20 = (67,68), D21 = (69, 70) i D22 = (50, 54). W symbolach znakowych będzie to odczytane jako **CDEF26**.

Można to pokazać graficznie, tak jak w tabeli z prawej strony. Należy zwrócić szczególną uwagę na to, że dana źródłowa S jest czytana od najbardziej znaczącego urządzenia do najmniej znaczącego, podczas gdy dana docelowa D jest czytana w odwrotnym kierunku.

Instrukcja ASCII może być użyta ze znacznikiem trybu 8/16-bit, M8161. Efekt działania tego znacznika jest dokładnie taki sam, jak opisany na stronie 5-98.

Źródło (S)	Dane	Przeznaczenie (D)		Kod ASCII		Symbol
		HE	DE	X	C	
D9	b12-15			A		
	b8-11			B		
	b4-7			C		
	b0-3			D		
D8	b12-15			E		
	b8-11			F		
	b4-7			2		
	b0-3			6		
D20	b8-15	43	67	'C'		
	b0-7	44	68	'D'		
D21	b8-15	45	69	'E'		
	b0-7	46	70	'F'		
D22	b8-15	32	50	'2'		
	b0-7	36	54	'6'		

Przykład po prawej pokazuje działanie, gdy znacznik M8161 jest w stanie OFF.

Gdyby znacznik M8161 był ustawiony w stan ON, wtedy tylko najmłodszy bajt docelowy (b0-b7) zostałby użyty do przechowania danych i stałbyby wymaganych 6 rejestrów danych (D20-D25).

**Kody znaków ASCII**

Tabela poniżej określa używane cyfry szesnastkowe i związane z nimi kody ASCII.

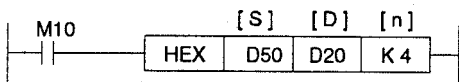
Znak HEX		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Kod ASCII	HEX	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
	DEC	48	49	50	51	52	53	54	55	56	57	65	66	67	68	69	70
Symbol znaku		'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'

5.9.4 HEX (FNC 83)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
HEX FNC 83 (konwersja ASCII na HEX)	konwertuje dane z formatu ASCII na HEX	K, H, KnX, KnY, KnM, KnS T, C, D	KnY, KnM, KnS T, C, D, V, Z	K, H uwaga: n=1 do 256 <input type="checkbox"/>	HEX, HEXP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja odczytuje n bajtów danych ASCII spod źródłowego adresu początkowego (S) i konwertuje je do równoważnych znaków szesnastkowych, po czym

przechowuje w urządzeniu docelowym (D) jako n bajtów.

Uwagi:

Należy zauważyć, że ta instrukcja działa odwrotnie niż instrukcja ASCII, tzn. dane ASCII przechowywane w bajtach, są konwertowane do postaci szesnastkowej. Instrukcja HEX może być użyta ze znacznikiem trybu 8/16-bit, M8161. W tym przypadku dana źródłowa (S) jest czytana z najmłodszego bajtu (8 bitów), przy M8161 w stanie ON, lub z całego słowa, przy M8161 w stanie OFF, tzn. jak pokazano w przykładzie powyżej, z następującymi danymi w urządzeniach, odpowiednio, D50 i D51 (43h, 41h) (43h, 31h) i przy założeniu, że M8161 jest w stanie ON.

Dane ASCII są konwertowane do swoich szesnastkowych równoważników i przechowywane kolejno, cyfra po cyfrze, od docelowego adresu początkowego.

Jeżeli M8161 byłby w stanie OFF, wtedy zawartość D20 byłaby odczytana jako CAB1h.



Dalsze szczegóły dotyczące użycia instrukcji HEX i dostępnych zakresów danych ASCII występują w tabeli "Kody znaków ASCII" pod instrukcją ASCII na poprzedniej stronie.

**Ważne:**

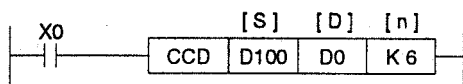
Instrukcja nie będzie wykonana przy próbie konwersji znaków ASCII spoza zakresu określonego w tabeli na poprzedniej stronie. Błąd nr 8067 będzie wykazany w rejestrze D8004, a błąd 6706 w rejestrze D8067. Szczególna ostrożność powinna być zachowana przy użyciu znacznika M8161 i dodatkowo przy określaniu liczby przetwarzanych elementów 'n', ponieważ są to najbardziej prawdopodobne miejsca spowodowania błędu.

5.9.5 CCD (FNC 84)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D	n	
CCD FNC 84 (kontrola parzystości)	sprawdza "pionową" parzystość w stosie danych	KnX, KnY, KnM, KnS T, C, D	KnY, KnM, KnS T, C, D	K, H, D uwaga: n=1 do 256 ☒	CCD, CCDP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja przegląda bajtowy stos 'n' bajtów danych od adresu początkowego (S) i sprawdza pionowy wzorec bitowy parzystości oraz oblicza sumę bajtów stosu. Te dane

są następnie zapamiętane w urządzeniu przeznaczenia (D).

Uwagi:

Suma danych stosu jest przechowana w urządzeniu D, podczas gdy parzystość w urządzeniu D+1

Podczas kontroli parzystości wystąpienie parzystości jest sygnalizowane "0", podczas gdy brak parzystości (nieparzystość) sygnalizowany jest "1".

Instrukcja może być użyta ze znacznikiem trybu 8/16-bit, M8161. Dla przyjętych poniżej warunków pojawią się następujące wyniki. Więcej szczegółów na temat znacznika M8161 na stronie 5-98.

M8161 = OFF										
Źródło (S)		Wzorec bitowy								
D100	H	FF	1	1	1	1	1	1	1	1
	L	FF	1	1	1	1	1	1	1	1
D101	H	FF	1	1	1	1	1	1	1	1
	L	00	0	0	0	0	0	0	0	0
D102	H	F0	1	1	1	1	0	0	0	0
	L	0F	0	0	0	0	1	1	1	1
Parzystość pion. D1			0	0	0	0	0	0	0	0
Suma D0			3FC							

M8161 = ON										
Źródło (S)		Wzorec bitowy								
D100 L	FF	1	1	1	1	1	1	1	1	1
D101 L	00	0	0	0	0	0	0	0	0	0
D102 L	0F	0	0	0	0	1	1	1	1	1
D103 L	F0	1	1	1	1	0	0	0	0	0
D104 L	F0	1	1	1	1	0	0	0	0	0
D105 L	0F	0	0	0	0	1	1	1	1	1
Parzystość pion.D1		1	1	1	1	1	1	1	1	1
Suma D0			2FD							

Należy zauważyć, że kiedy M8161 jest w stanie OFF, 'n' przedstawia liczbę kolejnych bajtów, sprawdzonych przez instrukcję CCD. Kiedy M8161 jest w stanie ON, używane są tylko najmłodsze bajty 'n' kolejnych słów.

Suma jest po prostu sumą danych w stosie.

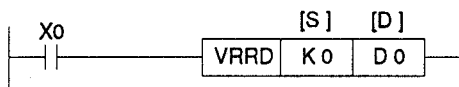
Parzystość jest sprawdzana pionowo przez stos danych, jak pokazano w zaciemnionych obszarach.

5.9.6 VRRD (FNC 85)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
VRRD FNC 85 (odczyt poziomu)	odczytuje wartość analogową z 1 z 8 wejść FX-8AV	K, H uwaga: S=0 do 7, odpowiednio do 8 dostępnych poziomów FX-8AV	KnY, KnM, KnS T, C, D, V, Z	VRRD, VRRDP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Wartość (S) w FX-8AV jest czytana jako wejście analogowe. Dana analogowa jest w formacie 8-bitowym, tzn. przyjmuje wartości 0-255. Czytana dana jest przechowywana w urządzeniu docelowym, określonym argumentem D.

Uwaga:

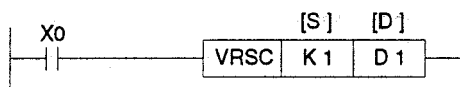
Wejścia analogowe FX-8AV mogą być czytane w dwóch formatach, a) jako wartość analogowa i b) jako 11-pozycyjny (0-10) przełącznik obrotowy. Drugi format jest opisany w instrukcji VRSC (FNC 86).

5.9.7 VRSC (FNC 86)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
VRRD FNC 85 (skalowanie poziomu)	odczytuje wartość ustawienia 0 do 10 wejść FX-8AV	K, H uwaga: S=0 do 7, odpowiednio do 8 dostępnych poziomów FX-8AV	KnY, KnM, KnS T, C, D, V, Z	VRSC, VRSCP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

11-pozycyjny (0-10) przełącznik obrotowy jest czytany w FX-8AV (S), jako dana analogowa. Dane pozycji są przechowywane w urządzeniu D jako liczba całkowita z

zakresu 0-10.

**Uwaga:**

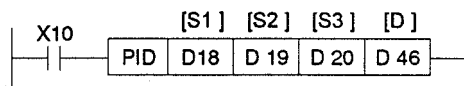
Wejścia analogowe FX-8AV mogą być czytane w dwóch formatach, a) jako 11-pozycyjny (0-10) przełącznik obrotowy i b) jako wartość analogowa. Drugi format jest opisany w instrukcji VRRD (FNC 85).

5.9.8 PID (FNC 88)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty				Kroki programu
		S1	S2	S3	D	
PID FNC 88 (pętla sterująca PID)	odbiera dane z wejścia, oblicza wartość korygującą PID na określonym poziomie	D uwaga: S1 i S2 wykorzystują pojedyncze rejestry danych każdy		D uwaga: S3 używa 25 kolejnych rejestrów danych	D uwaga: D używa pojedynczego rejestru danych	PID: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja pobiera bieżącą wartość (S_2) i porównuje ją z predefiniowaną ustawioną wartością (S_1). Różnica lub błąd pomiędzy dwiema wartościami jest przetwarzana w

pętli PID do wytworzenia współczynnika korygującego, uwzględniając poprzednio wyliczone wartości i trend błędu obliczenia. Proces PID oblicza współczynnik korygujący, jako bieżącą wartość wyjściową i zapamiętaną w urządzeniu docelowym (D). Parametry ustawione dla pętli PID są przechowane w 25 kolejnych rejestrach od S_3+0 do S_3+24

Uwagi:

- Instrukcja PID jest dostępna tylko na jednostkach FX i FX2C z wersją CPU 3.11 i późniejszą.
- Ponieważ do przechowania parametrów pętli PID wymagane jest 25 rejestrów, adres początkowy stosu danych nie może być większy niż D975. Zawartość danych na stosie będzie później omówiona w tej sekcji.
- Można zaprogramować wielokrotne instrukcje PID, jednakże każda pętla PID nie może wykorzystywać tych samych rejestrów.
- Instrukcja PID zawiera własny zbiór kodów błędów. Błędy są identyfikowane w normalny sposób. Kody błędów związanych z pętlą PID są sygnalizowane przez znacznik M8067 z odpowiednim kodem błędu zapisanym w rejestrze D8067. Te urządzenia nie są wykorzystywane wyłącznie przez instrukcję PID, stąd należy zachować ostrożność przy śledzeniu błędów. Więcej szczegółów w rozdziale 6 - "Urządzenia Diagnostyczne".
- Iteracja PID nie musi być w pełni wykorzystywana. Poprzez odpowiednie ustawienia parametrów pętli P (proporcjonalnej), I (całkującej) i D (różniczkowej) można dostosować je indywidualnie. Zostanie to opisane na następnej stronie.
- Sterowanie PID jest często kojarzone z analogowymi systemami sterowania. Obecnie nie ma powodu, dla którego jakikolwiek cykl sterowania, wykorzystujący technikę analogową czy też cyfrową, nie mógł stosować PID. Sterowanie PID jest tylko metodą określenia błędu w systemie i wyliczającą współczynnik korygujący, przekazany systemowi, aby ten mógł osiągnąć żadaną wartość. PID może być właściwie określony jako system dynamicznej korekcji błędów.

Ustawianie parametrów PID; S3

Parametry PID są przechowywane w stosie 25 rejestrów. Niektóre z nich wymagają wprowadzenia danych, inne są zarezerwowane do operacji wewnętrznych, wreszcie inne zwracają dane wyjściowe operacji PID. Należy zwrócić uwagę na uproszczenie opisu parametrów zawartych w tabeli poniżej, a to dla ułatwienia ich zrozumienia. Instrukcja PID wykorzystuje skomplikowany aparat matematyczny i każdy z parametrów poniżej objaśnia tylko małą część całego procesu PID.

Parametr S ₃ + ☆	Funkcja/ nazwa parametru	Opis		Zakres ustawień
S ₃ +0	Ts - czas próbkowania	przedział czasowy pomiędzy odczytami próbek wartości bieżącej systemu		1 do 32767 ms
S ₃ +1	kierunek akcji-reakcji i sterowanie alarmem	b0	sprężenie dodatnie (0) sprężenie ujemne (1)	brak
		b1	alarm zmiany CV OFF(0)/ON(1)	
		b2	alarm zmiany OV OFF(0)/ON(1)	
		b3-15	zarezerwowane	
S ₃ +2	filtr wejściowy	zmienia efekt filtru wejściowego		0-99%
S ₃ +3	Kp	współczynnik, używany do wyrównania wyjścia proporcjonalnego przy znanym zwiększeniu zmiany CV (S ₂). Część P pętli PID		1 do 32767%
S ₃ +4	Stała czasu całkowania Ti	Część I pętli PID. Ustawienie na 0 wyłącza efekt I		(0 do 32767) x 100 ms
S ₃ +5	Kd	Współczynnik używany do wyrównania wyjścia różniczkującego w znanej proporcji zmian CV (S ₂)		1 do 100%
S ₃ +6	Td	Część D pętli PID. Ustawienie na 0 wyłącza efekt D		(0 do 32767) x 10 ms
S ₃ +7 do S ₃ +19	zarezerwowane dla procesów wewnętrznych			
S ₃ +20	Maks. dodatnia zmiana CV	Aktywne gdy S ₃ +1 b1 jest ON	zdefiniowany przez użytkownika limit ilości dodatnich zmian, występujących w jednym przebiegu PID. Jeśli CV go przekroczy, ustawiany jest bit S ₃ +24, b0	0 do 32767
S ₃ +21	Maks. ujemna zmiana CV		zdefiniowany przez użytkownika limit ilości ujemnych zmian, występujących w jednym przebiegu PID. Jeśli CV go przekroczy, ustawiany jest bit S ₃ +24, b1	
S ₃ +22	Maks. dodatnia zmiana OV	Aktywne gdy S ₃ +1 b2 jest ON	zdefiniowany przez użytkownika limit ilości dodatnich zmian, występujących w jednym przebiegu PID. Jeśli OV go przekroczy, ustawiany jest bit S ₃ +24, b2	
S ₃ +23	Maks. ujemna zmiana OV		zdefiniowany przez użytkownika limit ilości ujemnych zmian, występujących w jednym przebiegu PID. Jeśli CV go przekroczy, ustawiany jest bit S ₃ +24, b3	
S ₃ +24	znaczniki alarmu (Tylko do odczytu)	b0	przekroczenie dodatnich zmian CV (S ₂)	brak
		b1	przekroczenie ujemnych zmian CV (S ₂)	
		b2	przekroczenie dodatnich zmian OV (D)	
		b3	przekroczenie ujemnych zmian OV (D)	
		b4-15	zarezerwowane	



Zobacz też **Wartości początkowe pętli PID** jako podstawowy przewodnik wartości początkowych PID, strona 5-110.

Opis dodatkowych parametrów dostępnych w jednostkach FX2N na stronie 10-18.

Konfiguracja pętli PID

Pętla PID może być konfigurowana dla różnych wariantów sterowania PID.

Metoda sterowania	Wybór poprzez rejestry			Opis
	S ₃ +3 (Kp)	S ₃ +4 (Ti)	S ₃ +6 (To)	
P	użytkownik	0	0	tylko efekt proporcjonalny
PI	użytkownik	użytkownik	0	efekt proporcjonalny i całkujący
PD	użytkownik	0	użytkownik	efekt proporcjonalny i różniczkujący
PID	użytkownik	użytkownik	użytkownik	pełny PID

Należy zauważyć, że we wszystkich sytuacjach wykorzystywany jest element proporcjonalny (P) pętli.

Sprężenia dodatnie i ujemne (S₃+1, b0)

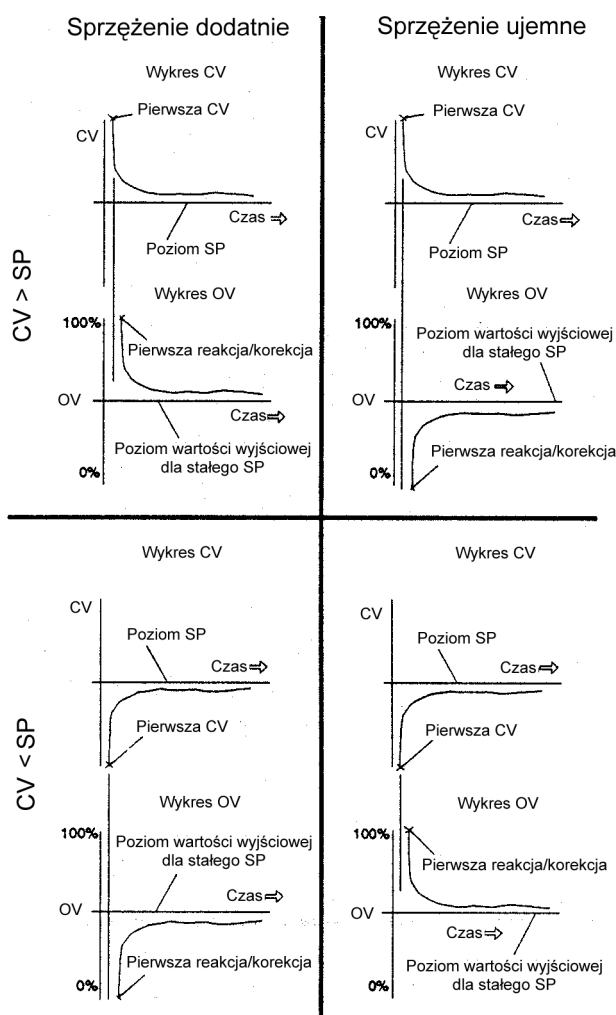
Te dwa parametry są mylące jeżeli chodzi o ich nazewnictwo i dlatego należy dokładnie zrozumieć efekty ich działania na pętli PID. Dla każdego z tych trybów istnieją dwa warunki:

- 1) wartość bieżąca (CV, rejestr danych S2) jest większa niż punkt ustawienia (SP lub wartość ustawienia, rejestr danych S1), tzn. $CV > SP$
- 2) punkt ustawienia (SP) jest większy niż wartość bieżąca (CV), tzn. $SP > CV$.

Ponieważ wartość wyjściowa (OV) jest modyfikowana, aby zastosować korekcję sterowanego systemu, błąd systemu, tzn. różnica pomiędzy wartością bieżącą (CV) a punktem ustawienia (SP), jest redukowany.

O wyborze trybu dodatniego lub ujemnego będą decydować wymagania systemowe, tzn. jeżeli wystąpi błąd $CV > SP$ i wybrany jest tryb dodatni, wtedy rośnie bezwzględna wartość wyjściowa OV. To spowoduje większy nakład pracy w skorygowaniu błędu systemowego. Np. dla klimatyzatorów typowym działaniem jest chłodzenie, tzn. jeśli w pomieszczeniu temperatura wzrośnie ponad ustawiony poziom, system chłodzenia musi wykonać więcej pracy aby obniżyć temperaturę. Z drugiej strony, gdyby zastosowano tryb ujemny dla tego samego błędu, OV zredukowałaby ilość pracy dostarczonej do systemu. Np. typowe dla działania ogrzewającego, tzn. jeśli w pomieszczeniu temperatura wzrosła ponad ustawiony poziom, nie ma potrzeby wykonywać żadnej pracy.

Stąd wybór trybu dodatniego lub ujemnego jest oparty na mechanice sterowanego systemu i metodach korygowania. We wszystkich sterowaniach PID istnieje opóźnienie pomiędzy rozpoznaniem błędu a pierwszym zastosowanym działaniem korekcyjnym. Aby pomóc w zrozumieniu tego i faktu, że korekcja błędu PID jest cykliczna, wykresy na tej stronie pokazują odpowiedź systemu (wartości CV) na zastosowane korekty (wartości OV) przez pewien okres czasu.



Efekty zastosowania elementów P, I, D na wartość wyjściową (OV)

**Ważne:**

Przed przeczytaniem kolejnych sekcji ważne jest dokładne zrozumienie danych przedstawionych na wykresach. Objasnienia graficzne pokazują wartość bieżącą i jej zmianę powodowaną przez błąd. Kiedy błąd jest pokazany jako wartość stała, oznacza to, że powtarza się błąd o takim samym poziomie. To może wydawać się oczywiste, ale wykresy mogą być błędnie interpretowane jako jedynie krótkie pojawienie się błędu (tzn. tylko punkt zmiany), co może spowodować wybór nowego poziomu, który jest stały. W dodatku punkt zmiany wartości bieżącej (tzn. błąd) jest pokazany jako chwilowy. W warunkach rzeczywistego zastosowania nie mogłoby się tak zdarzyć i prawdopodobnie trwałoby przez okres czasu równy operacji PID tworzącej następane wielokrotne próbki. Należy uważać próbując zrozumieć mechanizmy tych idealizowanych operacji.

P - zmiana proporcjonalna

Kiedy jest zastosowany czynnik proporcjonalny, pobierany jest obliczony błąd (różnica pomiędzy wartością bieżącą (S2) i punktem ustawienia (S1)) i wykonywany jest proces matematyczny, który może zastosować czynnik korekcyjny dla wartości wyjściowej (D). Czynnik korekcyjny oparty jest na zastosowaniu dla wartości wyjściowej, procentu bieżącego błędu, tzn.

Wartość Wyjściowa = $K_p \times \text{Błąd Wartości Bieżącej}$

Uwaga: Wartość K_p jest ustawiona w rejestrze S_3+3

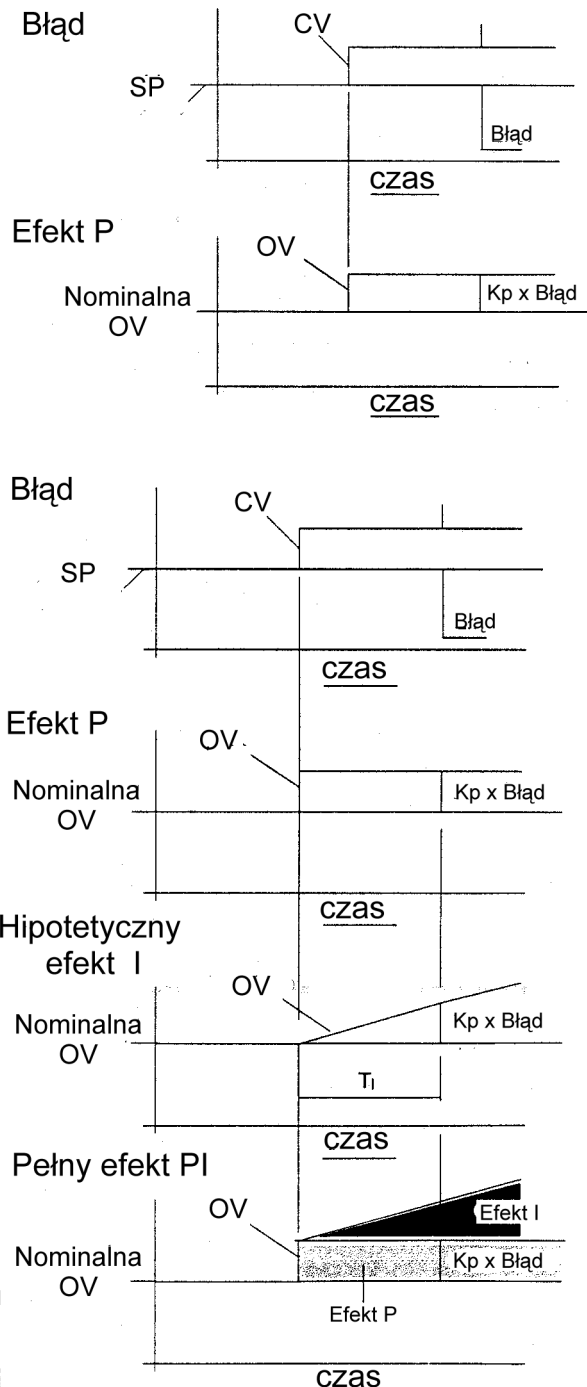
I - zmiana całkująca

Po zastosowaniu zmiany proporcjonalnej na wystąpienie błędu, można zastosować "dostrojenie" korekcji poprzez element całkujący (I).

Początkowo zastosowana jest tylko mała zmiana, ale w miarę upływu czasu rośnie błąd i efekt całkowania. Tempo wzrostu jest zdefiniowane przez czas T_i , który ustala punkt, kiedy wyjście całkujące równa się tej samej wartości bezwzględnej co korekcja proporcjonalna, tzn. ($K_p \times \text{Błąd}$).

Należy zwrócić uwagę jak zmienna T_i wpływa na to, jak szybko jest zastosowana całkowita korekcja całkująca. Stąd im mniejsze jest T_i , tym większy efekt będzie miało całkowanie.

Uwaga: wartość T_i jest ustawiona w rejestrze S_3+4



D - zmiana różniczkująca

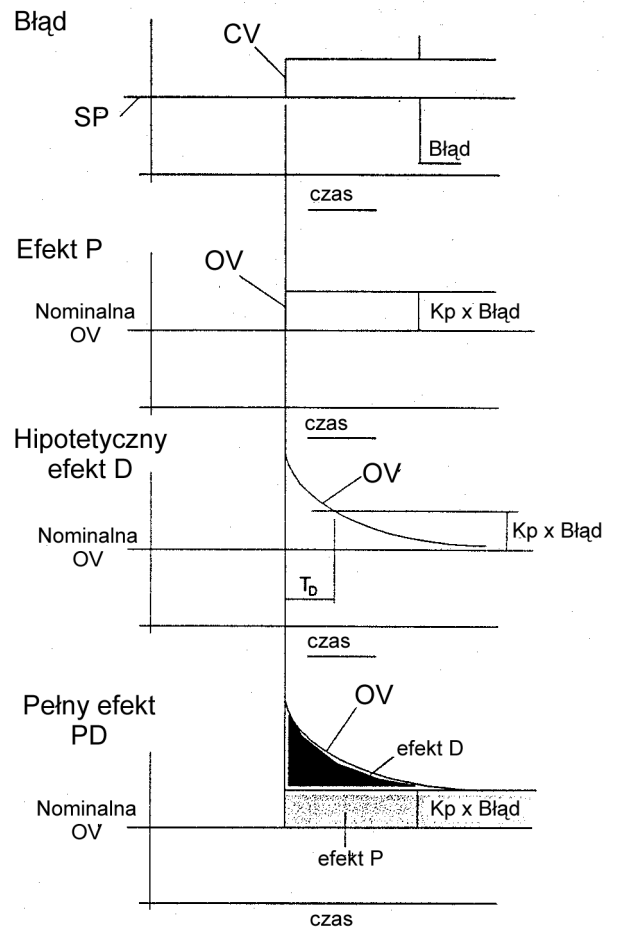
Funkcja różniczkująca wspomaga efekty spowodowane odpowiedzią proporcjonalną. Efekt różniczkowania jest wynikiem obliczenia, zawierającego elementy TD, TS i obliczonego błędu. To powoduje, że różniczkowanie generuje dużą odpowiedź korekcyjną, która rozprasza się gwałtownie w czasie. Szybkość rozpraszania może być kontrolowana przez wartość TD. TD jest właściwie określeniem upływu czasu, po którym wartość funkcji różniczkującej równa się wartości bezwzględnej efektu elementu proporcjonalnego.

Ponieważ początkowy efekt różniczkowania może być "surowy", istnieje efekt "zmiękczający", który może być zastosowany przez użycie K_d , zysku różniczkowania. Działanie K_d może być uważane jako filtr, umożliwiający skalowanie odpowiedzi różniczkowej w zakresie od 0 do 100%.

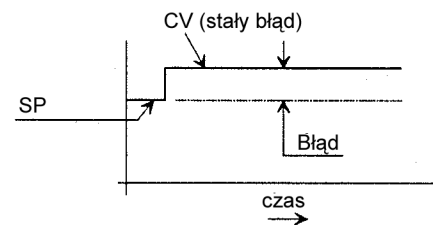
Obliczenie efektu różniczkowego jest skomplikowane matematycznie, ale istnieje prosta reguła, którą warto zapamiętać: jeżeli wartość TD jest mała, wtedy efekt zastosowania sterowania różniczkowego jest zwiększany.

Połączony efekt PID.

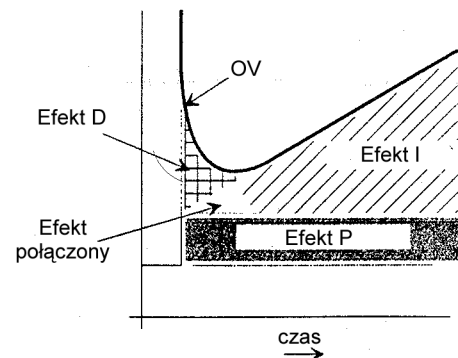
Do tej pory wyjaśniono, jaki wynik powstaje z połączenia maksimum 2 elementów korygujących, tzn. P, PI lub PD. Diagram po prawej pokazuje efekt połączenia wszystkich trzech elementów w pojedynczy system. Efektywny zakres/obszar każdej części instrukcji PID jest pokazany przez inny typ kreskowania.



Błąd



Efekt PID



Efektywne użycie filtrów wejściowych α S3 + 2

Aby zabezpieczyć instrukcję PID przed natychmiastowym i gwałtownym reagowaniem na błędy wartości bieżącej, istnieje mechanizm filtrujący, który pozwala instrukcji PID śledzić znaczące zmiany na trzech próbkach.

Efektem ilościowym filtra wejściowego jest liczenie filtrowanej wartości wejściowej dla instrukcji PID, wziętej ze zdefiniowanego procentu wartości bieżącej i poprzednich dwóch filtrowanych wartości wejściowych.

Ten typ filtrowania często jest zwany filtrem first-order lag. Jest on szczególnie użyteczny przy usuwaniu efektów szumów wysokiej częstotliwości, które mogą się pojawić w sygnałach wejściowych otrzymywanych z czujników.

Im wyżej jest ustawiony procent filtrowania, tym dłuższy jest czas opóźnienia. Kiedy filtr wejściowy jest ustawiony na 0, wtedy filtrowanie jest usunięte i wartość bieżąca może być użyta jako wartość wejściowa.

Wartości początkowe dla pętli PID.

Instrukcja PID ma wiele parametrów, które mogą być ustawiane i konfigurowane zgodnie z potrzebą użytkownika. Trudność polega na znalezieniu dobrego punktu, będącego początkiem dostrojenia pętli PID do wymagań systemu. Następujące propozycje nie będą idealne dla każdej sytuacji i do wszystkich zastosowań, ale dadzą użytkownikowi rozsądny punkt wyjścia.

Zalecane ustawienia początkowe

Ts = powinno być równe całkowitemu czasowi przebiegu programu lub wielokrotności tego czasu, tzn. 2 razy, 5 razy itp.

α = 50%

Kp = powinno być dostosowane do wartości zależnej od maksymalnego działania korekcyjnego w osiągnięciu punktu ustawienia - wartości powinny być dobrane doświadczalnie od 75%

Ti = optymalnie powinno być 4-10 razy większe niż czas **Td**

Kd = 50%

Td = zależy od całkowitej odpowiedzi systemu, tzn. nie tylko od szybkości reakcji sterownika, ale również wszystkich zaworów, pomp i silników.

Dla systemów o szybkiej reakcji **Td** będzie ustawione na krótki czas, jednakże nigdy mniej niż **Ts**. System o wolnej reakcji będzie wymagał, by opóźnienie **Td** było dłuższe.




Na sterownikach FX2N dostępna jest cecha autodostrajania, która może szybko zapewnić wartości początkowe dla procesu PID. Więcej szczegółów na stronie 10-22.

Ze wszystkimi wartościami PID można w pewnym stopniu eksperymentować, aby dostroić pętlę PID dokładnie do warunków lokalnych. Można to zrobić poprzez dopasowanie jednego parametru na raz przez ustalone procenty, tzn. zwiększając np. wartość **Kp** krokowo o 10%. Niewłaściwy wybór parametrów PID prowadzi do źle skonfigurowanego systemu, który nie będzie działał zgodnie z oczekiwaniami. Należy pamiętać, że proces PID jest czysto matematycznym wyliczeniem i jako taki nie ocenia jakości zmiennej danej, dostarczonej przez użytkownika/system. PID zawsze przetworzy swoją matematyczną funkcję dla takich danych, jakie otrzyma.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu 5-4
2. Instrukcje przesłania i porównania 5-16
3. Operacje arytmetyczne i logiczne 5-24
4. Rotacje i funkcje przesunięcia 5-34
5. Operacje na danych 5-42
6. Szybkie przetwarzanie 5-52
7. Instrukcje podręczne 5-66
8. Zewnętrzne urządzenia wejścia/wyjścia 5-80
9. Zewnętrzne urządzenia szeregowo 5-94
-  10. Zewnętrzne jednostki F2 5-112
11. Operacje zmiennoprzecinkowe 1 & 2 5-120
12. Operacje trygonometryczne 5-128
13. Operacje na danych 2 5-132
14. Sterowanie zegarem czasu rzeczywistego 5-136
15. Kody Graya 5-144
16. Porównania operacyjne 5-148

5.10 Zewnętrzne jednostki F2 - funkcje 90 - 99

Zawartość:

			Strona
MNET	F-16NX, sieć Melsec Net Mini	FNC 90	5-113
ANRD	F2-6A, odczyt analogowy	FNC 91	5-113
ANWR	F2-6A, zapis analogowy	FNC 92	5-114
RMST	F2-32RM, start RM	FNC 93	5-114
RMWR	F2-32RM, zapis RM	FNC 94	5-115
RMRD	F2-32RM, odczyt RM	FNC 95	5-116
RMMN	F2-32RM, monitor RM	FNC 96	5-116
BLK	F2-30GM, blok	FNC 97	5-117
MCDE	F2-30GM, kod maszynowy	FNC 98	5-118
☆☆☆	brak	FNC 99	

Uwaga: Dalsze informacje na temat wszystkich instrukcji w tej sekcji znajdują się w rozdziale "Przyporządkowanie Urządzeń Systemowych"



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1 , S_3 lub dla urządzeń stabilizowanych D_{3+0} , S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.10.1 MNET (FNC 90)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
MNET FNC 90 (sieć Melsec mini)	steruje siecią modułów serii F, używać z FX2-24Ei	X <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	Y <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	MNET, MNETP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja jest używana do komunikacji bitowych sygnałów statusu pomiędzy PLC i interfejsem F-16NP/NT sieci Melsec Net Mini.

Numery adresów początkowych wejścia/wyjścia dla S i D są określone przez pozycję FX2-24Ei (połączonego z F-16NP/NT) w łańcuchu rozszerzającym PLC. Urządzenia określone dla S i D mogą być użyte bezpośrednio w programie użytkownika FX.

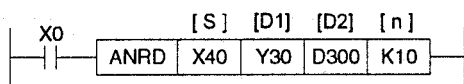
Więcej informacji na stronie 9-3.

5.10.2 ANRD (FNC 91)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D ₁	D ₂	n	
ANRD FNC 91 (odczyt analogowy)	odczytuje moduły analogowe serii F, używać z FX2-24Ei	X <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	Y <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	KnY, KnM, KnS, T, C, D, V, Z	K, H <input checked="" type="checkbox"/> uwaga: n = 10 do 13	ANRD, ANRDP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

Działanie:

Instrukcja jest używana do odczytania n kanału wejściowego modułu analogowego F2-6A. Odczytana wartość analogowa jest przechowana w urządzeniu docelowym D2.

Numery adresów początkowych wejścia/wyjścia dla S i D1 są określone przez pozycję FX2-24Ei (połączonego z F2-6A) w łańcuchu rozszerzającym PLC.

Dane analogowe przechowane w urządzeniu docelowym D2 są w formacie 8-bitowym.

Do określenia, który kanał analogowy jest czytany, użyty jest argument 'n', tzn. 10-13.

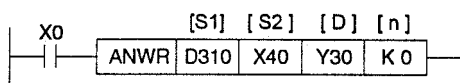
Więcej informacji na stronie 9-4.

5.10.3 ANWR (FNC 92)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S ₁	D ₂	D	n	
ANWR FNC 92 (zapis analogowy)	zapisuje do modułów analogowych serii F, używać z FX2-24EI	KnY, KnM, KnS, T, C, D, V, Z	X ☒ uwaga: użyte 8 kolejnych urządzeń	Y ☒ uwaga: użyte 8 kolejnych urządzeń	K, H ☒ uwaga: n = 0 do 1	ANWR, ANWRP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja jest używana do zapisania danej wyjściowej do kanału n modułu analogowego F2-6A. Zapisana wartość analogowa jest przechowana w urządzeniu źródłowym S1.

Numery adresów początkowych wejścia/wyjścia dla S2 i D są określone przez pozycję FX2-24EI (połączonego z F2-6A) w łańcuchu rozszerzającym PLC.

Dane analogowa, która ma być zapisana, jest przechowana w urządzeniu źródłowym S1 w formacie 8-bitowym.

Do określenia, który kanał analogowy jest zapisywany, użyty jest argument 'n', tzn. 0 lub 1.

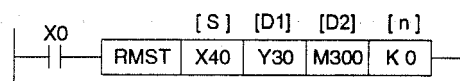
Więcej informacji na stronie 9-4.

5.10.4 RMST (FNC 93)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S	D ₁	D ₂	n	
RMST FNC 93 (start RM)	uruchamia moduły CAM serii F, używać z FX2-24EI	X ☒ uwaga: użyte 8 kolejnych urządzeń	Y ☒ uwaga: użyte 8 kolejnych urządzeń	Y, M, S ☒ uwaga: użyte 8 kolejnych urządzeń	K, H ☒ uwaga: n = 0 do 1	RMST: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja jest użyta do rozpoczęcia operacji F2-32RM pod kontrolą FX i do monitorowania bieżącego statusu F2-32RM.

Uwagi:

- Numery adresów początkowych wejścia/wyjścia dla S2 i D są określone przez pozycję FX2-24EI (połączonego z F2-6A) w łańcuchu rozszerzającym PLC.
- Argument 'n' jest używany do określenia, który program F2-32RM jest aktywny, tzn. 0 lub 1.
- Argument D2 przechowuje informację o statusie F2-32RM.

	Status urządzenia bitowego D_{2+m}	
	ON	OFF
D_{2+0}	wybrany BANK 1 (program 1)	wybrany BANK 0 (program 0)
D_{2+1}	-	normalnie OFF
D_{2+2}	START	STOP
D_{2+3}	kroki 1.0 st.	kroki 0,5 st.
D_{2+4}	normalnie ON	-
D_{2+5}	operacja Clockwise (CW)	operacja Counter-clockwise (CCW)
D_{2+6}	normalne działanie - brak błędu	błąd sprzętowy
D_{2+7}	normalne działanie - brak błędu	błąd programowy

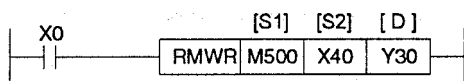
Więcej informacji na stronie 9-4.

5.10.5 RMWR (FNC 94)

$FX_{0(S)}$	FX_{0N}	FX	$FX_{(2C)}$	FX_{2N}
-------------	-----------	----	-------------	-----------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S_1	S_2	D	
RMWR FNC 94 (zapis RM)	blokuje wyjścia modułów CAM serii F, używać z FX2-24EI	Y, M, S uwaga: operacje 16-bitowe używają 16, a 32-bitowe - 32 kolejnych urządzeń	X uwaga: użyte 8 kolejnych urządzeń	Y ☒ uwaga: użyte 8 kolejnych urządzeń	RMWR, RMWRP: 7 kroków DRMWR, DRMWRP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
$FX_{0(S)}$	FX_{0N}	FX	$FX_{(2C)}$	FX_{2N}	$FX_{0(S)}$	FX_{0N}	FX	$FX_{(2C)}$	FX_{2N}	$FX_{0(S)}$	FX_{0N}	FX	$FX_{(2C)}$	FX_{2N}



Działanie:

Instrukcja przesyła dane wyjściowe blokujące do programowalnego przełącznika CAM F2-32RM ze sterownika FX.

Numerzy adresów początkowych wejścia/wyjścia dla S_2 i D są określone przez pozycję FX_{2-24EI} (połączonego z F2-32RM) w łańcuchu rozszerzającym PLC.

Argument S_1 jest adresem początkowym 16 lub 32 bitów źródłowych. Bity źródłowe odwzorowują się bezpośrednio na wyjścia F2-32RM Y0-Y37 (numeracja ósemkowa)

Kiedy urządzenie źródłowe w sterowniku jest włączone, przypisane urządzenie F2-32RM jest zablokowane.

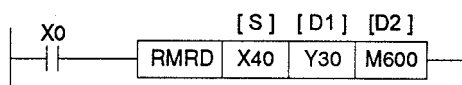
Więcej informacji na stronie 9-4.

5.10.6 RMRD (FNC 95)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	D ₁	D ₂	
RMRD FNC 95 (odczyt RM)	odczytuje status wyjść modułów CAM serii F, używać z FX2-24EI	X <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	Y <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	Y, M, S uwaga: operacje 16-bitowe używają 16, a 32-bitowe - 32 kolejnych urządzeń	RMRD, RMRDP: 7 kroków DRMRD, DRMRDP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja wczytuje bieżący status z wyjść programowalnego przełącznika CAM F2-32RM do sterownika FX.

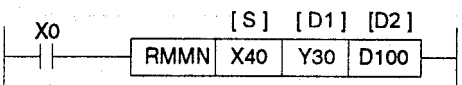
Numery adresów początkowych wejścia/wyjścia dla S i D1 są określone przez pozycję FX2-24EI (połączonego z F2-32RM) w łańcuchu rozszerzającym PLC. Statusy wyjść Y0-Y17 (operacja 16-bitowa) lub Y0-Y37 (operacja 32-bitowa) są odczytywane z F2-32RM. Dane odczytywane są bezpośrednio odwzorowywane na urządzeniu docelowym FX (adres początkowy D2). Urządzenia D2 zachowują swój ostatni stan, nawet jeżeli instrukcja RMRD jest wyłączona. Więcej informacji na stronie 9-4.

5.10.7 RMMN (FNC 96)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D ₁	D ₂	
RMMN FNC 96 (monitor RM)	monitoruje stany danych w modułach CAM serii F - używać z FX2-24EI	X <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	Y <input checked="" type="checkbox"/> uwaga: użyte 8 kolejnych urządzeń	KnY, YnM, YnS T, C, D, V, Z	RMMN, RMMNP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja jest używana do odczytania szybkości (rpm) lub bieżącej pozycji kątowej z F2-32RM do PLC. Następnie jest ona przechowana w urządzeniu określonym argumentem D2. Decyzja, która dana ma być odczytywana, tzn. szybkość czy pozycja, jest podejmowana przez przełącznik sprzętowy #4 w F2-32RM. Numery adresów początkowych wejścia/wyjścia dla S i D1 są określone przez pozycję FX2-24EI (połączonego z F2-32RM) w łańcuchu rozszerzającym PLC.

Więcej informacji na stronie 9-4.

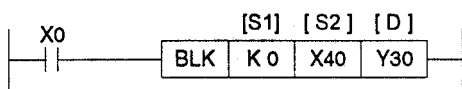
Przeł. #4	Argument	Wartość przykł.	Typ danej
ON	D ₂	350	kąt (st.)
OFF		830	prędkość (r.p.m.)

5.10.8 BLK (FNC 97)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
BLK FNC 97 (blok F2-30GM)	wyznacza numer bloku w F2-30GM, używać z FX2-24EI	K, H, KnX, KnY, KnM, KnS T, C, D, V, Z	X ☒ uwaga: użyte 8 kolejnych urządzeń	Y ☒ uwaga: użyte 8 kolejnych urządzeń	BLK, BLKP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja jest używana do wyznaczenia numeru bloku (S1) w jednostce wyjścia impulsowego F2-30GM przez sterowanie PLC.

Uwagi:

- Numery adresów początkowych wejścia/wyjścia dla S i D1 są określone przez pozycję FX2-24EI (połączonego z F2-30GM) w łańcuchu rozszerzającym PLC.
- Efektywnymi numerami bloku, które mogą być określone dla S1, są 0-31 (dziesiętnie). Dane, które mają być użyte dla S1 nie mogą być w formacie BCD, tzn. dane nie powinny być odczytywane bezpośrednio (używając np. KnX) z nastawników BCD.
- Kiedy używany jest F2-30GM i kiedy nie jest wymagana instrukcja BLK, program pokazany obok jest potrzebny do tego, aby FX2-24EI rozpoznał i poprawnie zadziałał dla dołączonego F2-30GM.
- Więcej informacji na stronie 9-5.

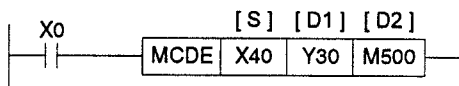


5.10.9 MCDE (FNC 98)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D ₁	D ₂	
MCDE FNC 98 (kod maszynowy F2-30GM)	odczytuje zbiór kodów z F2-30GM, używać z FX2-24EI	X ☒ uwaga: użyte 8 kolejnych urządzeń	X ☒ uwaga: użyte 8 kolejnych urządzeń	Y, M, S ☒ uwaga: użyte 64 kolejnych (numerowanych ósemkowo) urządzeń	MCDE, MCDEP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja umożliwi sterownikowi odczytanie kodów maszynowych F2-30GM. W F2-30GM istnieją 64 punkty kodu maszynowego, ponumerowane ósemkowo tzn. 0-77 i


są prefiksowane przez M.

Uwagi:

- Numery adresów początkowych wejścia/wyjścia dla S i D1 są określone przez pozycję FX2-24EI (połączonego z F2-30GM) w łańcuchu rozszerzającym PLC.
- Kiedy F2-30GM wykonuje kod maszynowy (jedno z urządzeń M0-M77), aktywowane jest przypisane urządzenie docelowe (D2). Urządzenia D2 są bezpośrednio odwzorowywane do kodów maszynowych F2-30GM, tzn. kiedy F2-30GM aktywuje M7, uaktywnia się również D2+7 (ósemkowo) w sterowniku. Jeżeli są użyte dane jak w przykładowej instrukcji powyżej i M77 w F2-30GM jest w stanie ON - to M577 (D2+77 ósemkowo) w PLC też będzie w stanie ON (podczas gdy instrukcja MCDE będzie aktywna).
- Aby ułatwić numerowanie pomiędzy FX i F2-30GM, zaleca się, aby niższe dwie cyfry adresu początkowego, użyte do określenia D2 były zero, tzn. '*00', jeśli urządzenia S lub M są użyte w PLC.
- Więcej informacji na stronie 9-5.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
 11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.11 Operacje zmiennoprzecinkowe 1 & 2 - funkcje 110 - 129

Zawartość:

		Strona
Zmienny przecinek 1:		
ECMP	porównanie zmiennoprzecinkowe	FNC110 5-121
EZCP	porównanie zakresu zmiennoprzecinkowe	FNC 111 5-121
☆☆☆	brak	FNC 112-117
EBCD	konwersja notacji zmiennoprzecinkowej na naukową	FNC 118 5-122
EBIN	konwersja notacji naukowej na zmiennoprzecinkową	FNC 119 5-122
Zmienny przecinek 2:		
EADD	dodawanie zmiennoprzecinkowe	FNC 120 5-123
ESUB	odejmowanie zmiennoprzecinkowe	FNC 121 5-124
EMUL	mnożenie zmiennoprzecinkowe	FNC 122 5-124
EDIV	dzielenie zmiennoprzecinkowe	FNC 123 5-125
☆☆☆	brak	FNC 124-126
ESQR	pierwiastek kwadratowy zmiennoprzecinkowy	FNC 127 5-125
☆☆☆	brak	FNC 128
INT	konwersja liczby zmiennoprzecinkowej na całkowitą	FNC 129 5-126



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

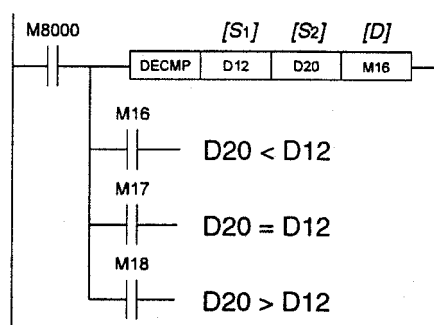
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.11.1 ECMP (FNC 110)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D ₁	D ₂	
ECMP FNC 110 (porównanie zmiennoprzecinkowe)	porównuje dwie wartości zmiennoprzecinkowe, zwraca wynik <, =, >	K, H - wartość całkowita, automatycznie konwertowana do postaci zmiennoprzecinkowej D - w formacie zmiennoprzecinkowym (32 bity)	Y, M, S uwaga: użyte 3 kolejne urządzenia	DECMP, DECMP: 13 kroków	

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Dane z S1 jest porównywana z daną z S2. Wynik jest określony przez 3 urządzenia bitowe, specyfikowane od adresu początkowego w D. Urządzenia bitowe określają:

- S2 jest mniejsze niż < S1 - D jest w stanie ON
- S2 jest równe = S1 - D+1 jest w stanie ON
- S2 jest większe niż > S1 - D+2 jest w stanie ON.

Uwagi:

Stan urządzeń docelowych będzie zachowany, nawet po deaktywowaniu instrukcji ECMP. Obowiązują reguły porównania algebraicznego, tzn. $-1,79 \times 10^{27}$ jest mniejsze niż $9,43 \times 10^{15}$

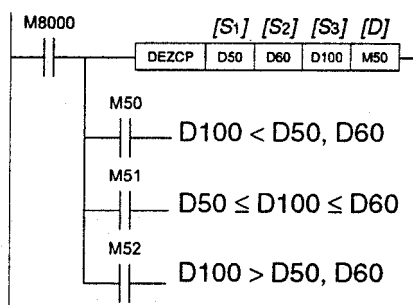


5.11.2 EZCP (FNC 111)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S ₁	S ₂	S ₃	D ₂	
EZCP FNC 111 (porównanie zakresu zmiennoprzecinkowe)	porównuje zakres wartości z jedną wartością zmiennoprzecinkową, zwraca wynik <, =, >	K, H - wartość całkowita, automatycznie konwertowana do postaci zmiennoprzecinkowej D - w formacie zmiennoprzecinkowym (32 bity) S1 musi być mniejsze od S2	Y, M, S uwaga: użyte 3 kolejne urządzenia	DEZCP, DEZCPP: 13 kroków		

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Działanie jest takie samo jak instrukcji ECMP, z tą różnicą, że wartość pojedynczej danej (S3) jest porównywana do zakresu danych (S1-S2).

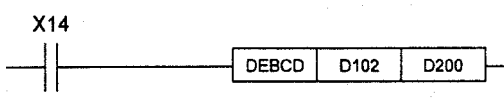
- S3 jest mniejsze niż S1 i S2 - D jest w stanie ON
- S3 jest pomiędzy S1 i S2 - D+1 jest w stanie ON
- S3 jest większe niż S2 - D+2 jest w stanie ON

5.11.3 EBCD (FNC 118)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D ₂	
EBCD FNC 118 (konwersja formatu)	konwertuje zapis zmiennoprzecinkowy na naukowy	D - w formacie zmiennoprzecinkowym (32 bity)	D - użyte 2 kolejne urządzenia D - mantysa, D+1 - wykładnik	DEBCD, DEBCDP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Instrukcja konwertuje wartość zmiennoprzecinkową w S na mantysę i wykładnik, które są przechowane w D i D+1 (notacja naukowa).

Uwagi:

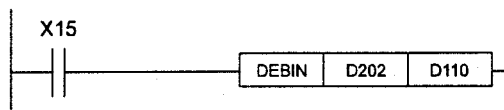
- Instrukcja musi być w formacie podwójnego słowa. Urządzenia docelowe D i D+1 reprezentują, odpowiednio, mantysę i wykładnik liczby zmiennoprzecinkowej.
- Aby zapewnić maksymalną dokładność konwersji mantysa D będzie w zakresie od 1000 do 9999 (lub 0), a wykładnik D+1 będzie skorygowany do odpowiedniej wartości.
- Przykład: $S = 3,4567 \times 10^{-5}$ wtedy $D = 3456$, $D+1 = -8$

5.11.4 EBIN (FNC 119)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D ₂	
EBIN FNC 119 (konwersja formatu)	konwertuje zapis naukowy na zmiennoprzecinkowy	D - użyte 2 kolejne urządzenia S - mantysa, S+1 - wykładnik	D - w formacie zmiennoprzecinkowym (32 bity)	DEBIN, DEBINP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Generuje liczbę zmiennoprzecinkową w D z danej w źródle S w notacji naukowej.

Uwagi:

Instrukcja musi być w formacie podwójnego słowa. Dane źródłowe S i S+1 reprezentują mantysę i wykładnik generowanej liczby zmiennoprzecinkowej.

Aby zapewnić maksymalną dokładność konwersji mantysa S musi być w zakresie od 1000 do 9999 (lub 0), a wykładnik S+1 musi być skorygowany do odpowiedniej wartości.

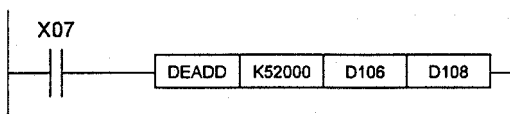
Przykład: $S = 5432$, $S+1 = 12$, wtedy $D = 5,432 \times 10^9$

5.11.5 EADD (FNC 120)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty			Kroki programu
		S	D ₁	D ₂	
EADD FNC 120 (dodawanie zmiennoprzecinkowe)	dodaje do siebie dwie wartości zmiennoprzecinkowe	K, H - wartość całkowita, automatycznie konwertowana do postaci zmiennoprzecinkowej D - w formacie zmiennoprzecinkowym (32 bity)		D - wartość zmiennoprzecinkowa (32 bity)	DEADD, DEADDP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	zero M8020 pożyczka M8021 przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Wartości zmiennoprzecinkowe przechowywane w urządzeniach źródłowych S1 i S2 są algebraicznie dodawane a wynik jest przechowywany w urządzeniu docelowym D.

Uwagi:

- Instrukcja musi być w formacie podwójnego słowa, tzn. DEADD lub DEADDP
Wszystkie dane źródłowe i docelowe będą podwójnym słowem, tzn. do przechowywania danych (32-bitowych) użyte będą dwa kolejne rejestry.
Z wyjątkiem K i H wszystkie dane źródłowe będą uważane za dane w formacie zmiennoprzecinkowym i wynik przechowany w urządzeniu docelowym będzie również w formacie zmiennoprzecinkowym.
- Jeżeli jako dana źródłowa jest użyta stała K lub H, wartość jest konwertowana do formatu zmiennoprzecinkowego przed operacją dodawania.
- Dodawanie jest algebraicznie poprawne, tzn. $2,3456 \times 10^2 + (-5,6 \times 10^{-1}) = 2,34 \times 10^2$
- To samo urządzenie może być użyte jako źródłowe i docelowe. W takim przypadku podczas wykonywania instrukcji DADD w sposób ciągły, wynik poprzedniej operacji będzie użyty jako nowa wartość źródłowa i będzie obliczany nowy wynik.
Tak się będzie działo w każdym przebiegu programu, chyba że użyty zostanie modyfikator impulsu lub blokada programowa.
- Jeżeli wynikiem obliczenia jest 0, wtedy znacznik zera, M8020, jest ustawiony w stan ON.
Jeżeli wynik obliczenia jest większy niż największa liczba zmiennoprzecinkowa, wtedy znacznik przeniesienia, M8021, jest ustawiony w stan ON, a wynik jest ustawiony na największą wartość.
Jeżeli wynik obliczenia jest mniejszy niż najmniejsza liczba zmiennoprzecinkowa, wtedy znacznik pożyczki, M8022, jest ustawiony w stan ON, a wynik jest ustawiony na najmniejszą wartość.



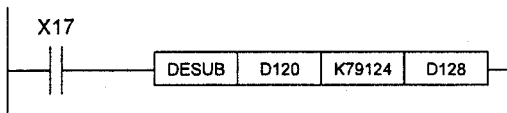
Więcej informacji na temat liczby zmiennoprzecinkowej na stronie 4-46.

5.11.6 ESUB (FNC 121)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S	D ₁	D ₂	
ESUB FNC 121 (odejmowanie zmiennoprzecinkowe)	odejmuje od siebie dwie wartości zmiennoprzecinkowe	K, H - wartość całkowita, automatycznie konwertowana do postaci zmiennoprzecinkowej D - w formacie zmiennoprzecinkowym (32 bity)		D - wartość zmiennoprzecinkowa (32 bity)	DESUB, DESUBP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	zero M8020 pożyczka M8021 przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Wartość zmiennoprzecinkowa z S2 jest odejmowana od wartości zmiennoprzecinkowej z S1 a wynik jest przechowywany w urządzeniu docelowym D

Uwagi:

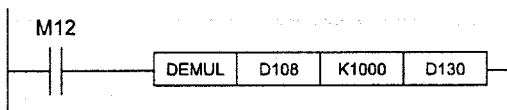
Te same uwagi, które były przy instrukcji EADD, z wyjątkiem tego, że wykonywane jest odejmowanie - zobacz strona 5-124

5.11.7 EMUL (FNC 122)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
EMUL FNC 122 (mnożenie zmiennoprzecinkowe)	mnoży dwie wartości zmiennoprzecinkowe	K, H - wartość całkowita, automatycznie konwertowana do postaci zmiennoprzecinkowej D - w formacie zmiennoprzecinkowym (32 bity)		D - wartość zmiennoprzecinkowa (32 bity)	DEMUL, DEMULP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Wartość zmiennoprzecinkowa z S1 jest mnożona przez wartość zmiennoprzecinkową z S2 a wynik jest przechowywany w urządzeniu docelowym D.

Uwagi:

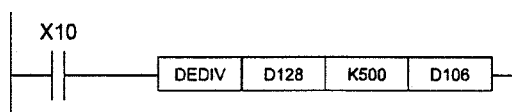
Te same uwagi, które były przy instrukcji EADD, z wyjątkiem punktu e) oraz tego, że wykonywane jest mnożenie - zobacz strona 5-124

5.11.8 EDIV (FNC 123)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
EDIV FNC 123 (dzielenie zmiennoprzecinkowe)	dzieli dwie wartości zmiennoprzecinkowe	K, H - wartość całkowita, automatycznie konwertowana do postaci zmiennoprzecinkowej D - w formacie zmiennoprzecinkowym (32 bity)		D - wartość zmiennoprzecinkowa (32 bity)	DEDIV, DEDIVP: 13 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Wartość zmiennoprzecinkowa z S1 jest dzielona przez wartość zmiennoprzecinkową z S2 a wynik jest przechowywany w urządzeniu docelowym D. Nie jest

obliczana reszta.

Uwagi:

Te same uwagi, które były przy instrukcji EADD, z wyjątkiem punktu e) oraz tego, że wykonywane jest dzielenie - zobacz strona 5-124

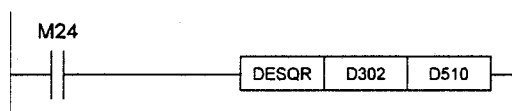
- Jeżeli S2 jest równe 0, wtedy pojawia się błąd dzielenia przez zero i operacja nie jest wykonywana.

5.11.9 ESQR (FNC 127)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu	
		S	D		
ESQR FNC 127 (pierwiastek kwadratowy zmiennoprzecinkowy)	oblicza wartość pierwiastka kwadratowego zmiennoprzecinkowego	K, H - wartość całkowita, automatycznie konwertowana do postaci zmiennoprzecinkowej D - w formacie zmiennoprzecinkowym (32 bity)		D - wartość zmiennoprzecinkowa (32 bity)	DESQR, DESQRP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	zero M8020
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**

Obliczany jest pierwiastek kwadratowy wartości zmiennoprzecinkowej z S a wynik jest przechowywany w D.

Uwagi:

Te same uwagi, które były przy instrukcji EADD, z wyjątkiem punktu e) oraz tego, że wykonywane jest pierwiastkowanie - zobacz strona 5-124

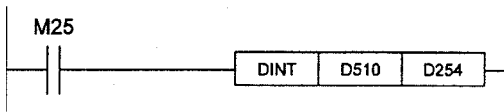
- Jeżeli S jest liczbą ujemną, pojawia się błąd i znacznik błęd, M8067, jest ustawiony w stan ON.

5.11.10 INT (FNC 129)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
INT FNC 129 (konwersja formatów)	konwertuje liczbę zmiennoprzecinkową na binarną	D - w formacie zmiennoprzecinkowym (32 bity)	D - format binarny dla INT, INTP: 16 bitów dla DINI, DINTP: 32 bity	INT, INTP: 5 kroków DINT, DINTP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	zero M8020
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Działanie:**


Wartość zmiennoprzecinkowa z S jest zaokrąglona w dół do najbliższej wartości całkowitej i przechowana w normalnym formacie binarnym w D.

Uwagi:

- Dana źródłowa jest zawsze podwójnym (32-bitowym) słowem; wartością zmiennoprzecinkową
Dla operacji 16-bitowej urządzenie docelowe jest 16-bitowe
Dla operacji 32-bitowej urządzenie docelowe jest 32-bitowe
 - Ta instrukcja jest odwrotną do instrukcji FLT (zobacz strona 5-49)
 - Jeżeli wynik jest równy 0, wtedy znacznik zera, M8020, jest ustawiony w stan ON.
Jeżeli dana źródłowa nie jest liczbą całkowitą, musi być zaokrąglona w dół. W takim przypadku znacznik pożyczki, M8021, jest ustawiony w stan ON, dla zaznaczenia, że wartość jest zaokrąglona.
Jeżeli wartość wynikowej liczby całkowitej jest poza zakresem urządzenia docelowego, pojawia się błąd przepełnienia. W tym przypadku znacznik przeniesienia, M8022 jest ustawiony w stan ON.
- Uwaga:** Jeżeli pojawi się przepełnienie, wartość wynikowa nie będzie poprawna.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
 12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.12 Operacje trygonometryczne - funkcje 130 - 139

Zawartość:

		Strona
Zmienny przecinek 3		
SIN	sinus	FNC 130 5-129
COS	cosinus	FNC 131 5-130
TAN	tangens	FNC 132 5-130
☆☆☆	niedostępne	FNC 133-139



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

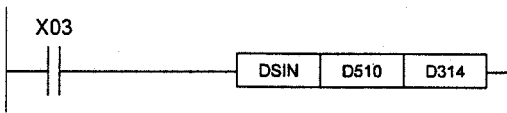
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.12.1 SIN (FNC 130)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
SIN FNC 130 (sinus)	oblicza wartość sinusa	D - w formacie zmiennoprzecinkowym (32 bity)	D - w formacie zmiennoprzecinkowym (32 bity)	DSIN, DSINP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Zawartość:**

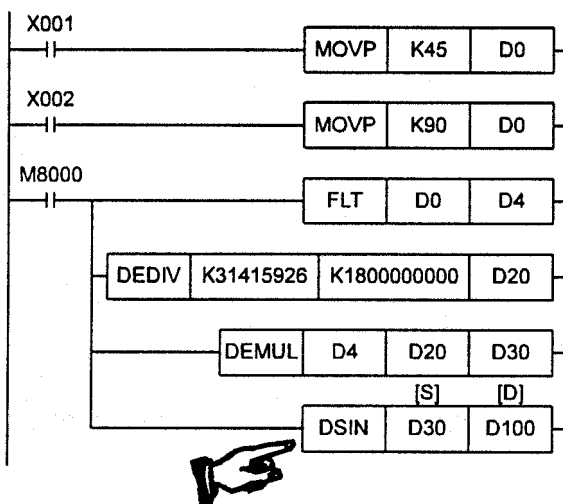
Instrukcja wykonuje operację matematyczną sinus na wartości zmiennoprzecinkowej w (S). Wynik jest przechowywany w (D).

Uwagi:

- Instrukcja musi używać formatu podwójnego słowa, tzn. DSIN lub DSINP
Wszystkie źródła i dane docelowe będą podwójnym słowem, tzn. będą używać dwóch kolejnych rejestrów do przechowania danych (32 bity).
Dana źródłowa i docelowa jest w formacie zmiennoprzecinkowym.
- Wartość źródłowa musi być kątem od 0 do 360 stopni, podanym w radianach, tzn.
 $0^\circ \leq S < 360^\circ$

Kąty w radianach

Poniżej znajduje się przykład programu, pokazującego, jak liczyć kąty w radianach, używając liczb zmiennoprzecinkowych



K45 stopni do D0

K90 stopni do D0

Konwersja D0 do liczby zmiennoprzecinkowej w D4, D5

Przeliczenie π na radiany ($\pi/180$)

Zapamiętanie jako liczby zmiennoprzecinkowej w D20, D21

Obliczenie kąta w radianach w D30, D31

(stopnie $^\circ \times \pi/180 =$ radiany)

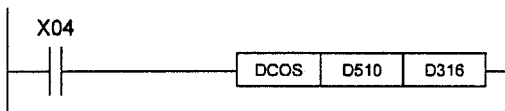
Obliczenie sinus kąta w D100

5.12.2 COS (FNC 131)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
COS FNC 131 (cosinus)	oblicza wartość cosinusa	D - w formacie zmiennoprzecinkowym (32 bity)	D - w formacie zmiennoprzecinkowym (32 bity)	DCOS, DCOSP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Zawartość:**

Instrukcja wykonuje operację matematyczną cosinus na wartości zmiennoprzecinkowej w (S). Wynik jest przechowywany w (D).

Uwagi:

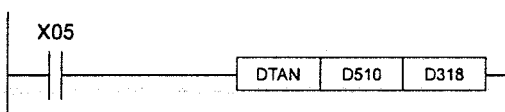
Mają zastosowanie wszystkie punkty poprzedniej sekcji z tym, że obliczany jest cosinus - zobacz strona 5-129

5.12.3 TAN (FNC 132)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
TAN FNC 131 (tangens)	oblicza wartość tangensa	D - w formacie zmiennoprzecinkowym (32 bity)	D - w formacie zmiennoprzecinkowym (32 bity)	DTAN, DTANP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

Zawartość:


Instrukcja wykonuje operację matematyczną tangens na wartości zmiennoprzecinkowej w (S). Wynik jest przechowywany w (D).

Uwagi:

Mają zastosowanie wszystkie punkty poprzedniej sekcji z tym, że obliczany jest tangens - zobacz strona 5-129

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
 13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.13 Operacje na danych 2 - funkcje 140 - 149

Zawartość:

		Strona
☆☆☆	niedostępne	FNC 140-146
SWAP	konwersja z notacji zmiennoprzecinkowej do naukowej	FNC147 5-133
☆☆☆	niedostępne	FNC 148-149



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

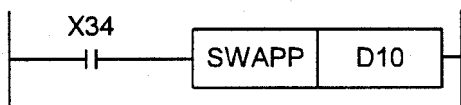
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.13.1 SWAP (FNC 147)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S		
SWAP FNC 147 (zamiana bajtów)	starszy i młodszy bajt oznaczonego urządzenia są zamieniane miejscami	KnY, KnM, KnS, T, C, D, V, Z		SWAP, SWAPP: 5 kroków DWAP, DSWAPP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Zawartość:**

Górne i dolne bajty urządzenia źródłowego są zamienione miejscami.

Ta instrukcja jest równoważna "działaniu 2" instrukcji XCH (FNC 17) - zobacz strona 5-21

Uwagi:

- W operacji na pojedynczym słowie (16-bitowym) górne i dolne bajty urządzenia źródłowego są zamienione miejscami.
- W operacji na podwójnym słowie (32-bitowym) górne i dolne bajty w każdym z dwóch 16-bitowych urządzeń są zamienione miejscami.

Wynik instrukcji DSWAP(P) D10:


Wartości szesnastkowe		Przed DSWAP	Po DSWAP
D10	Bajt 1	1FH	8BH
	Bajt 2	8BH	1FH
D11	Bajt 3	C4H	35H
	Bajt 4	35H	C4H

Jeżeli ta instrukcja będzie wykonana przy każdym przebiegu programu, wtedy pierwotna wartość urządzenia źródłowego będzie przywrócona w co drugim przebiegu. Zaleca się użycie modyfikatora impulsowego lub blokady programowej.

Notatki

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
 14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.14 Sterowanie zegarem czasu rzeczywistego - funkcje 160 - 169

Zawartość:

		Strona	
TCMP	porównanie czasu	FNC 160	5-137
TZCP	porównanie strefy czasu	FNC 161	5-138
TADD	zwiększanie czasu	FNC 162	5-139
TSUB	zmniejszanie czasu	FNC 163	5-140
☆☆☆	niedostępne	FNC 164-165	
TRD	konwersja z notacji zmiennoprzecinkowej do naukowej	FNC 166	5-141
TWR	konwersja z notacji zmiennoprzecinkowej do naukowej	FNC 167	5-142
☆☆☆	niedostępne	FNC 168-169	



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

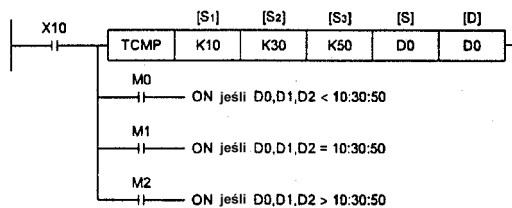
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.14.1 TCMP (FNC 160)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty					Kroki programu
		S ₁	S ₂	S ₃	S	D	
TCMP FNC 160 (porównanie czasu)	porównuje dwa czasy, zwracając wynik >, =, <	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z			T, C, D	Y, M, S	TCMP, TCMPP: 11 kroków
					uwaga: użyte 3 kolejne urządzenia		

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Zawartość:**

S1, S2 i S3 reprezentują, odpowiednio, godziny, minuty i sekundy. Czas ten jest porównywany z wartością czasu w 3 urządzeniach danych od adresu początkowego S. Wynik jest przechowywany w trzech urządzeniach bitowych od adresu początkowego D.

Urządzenia bitowe w D oznaczają:

D₊₀ jest ustawiony w stan ON, kiedy czas w S jest mniejszy niż czas w S1, S2 i S3

D₊₁ jest ustawiony w stan ON, kiedy czas w S jest taki sam, jak czas w S1, S2 i S3

D₊₂ jest ustawiony w stan ON, kiedy czas w S jest większy niż czas w S1, S2 i S3

Uwagi:

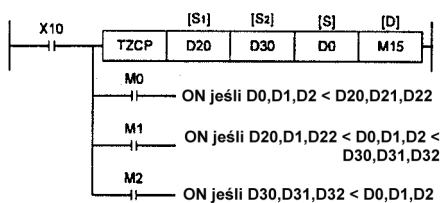
- Status urządzeń docelowych jest zachowany, nawet jeżeli instrukcja TCMP jest nieaktywna
- Porównanie jest oparte na wartości czasu określonej w urządzeniach źródłowych
 - dopuszczalny zakres wartości dla S1 i S+0 wynosi 0-23 (godziny)
 - dopuszczalny zakres wartości dla S2 i S+1 wynosi 0-59 (minuty)
 - dopuszczalny zakres wartości dla S3 i S+2 wynosi 0-59 (sekundy)
- Czas bieżący zegara czasu rzeczywistego może być porównany przez określenie D8015 (godziny), D8014 (minuty) i D8013 (sekundy) jako urządzeń dla, odpowiednio, S1, S2 i S3.

5.14.2 TCMP (FNC 161)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty				Kroki programu
		S ₁	S ₂	S ₃	D	
TZCP FNC 161 (porównanie zakresu czasu)	porównuje czas z określonym zakresem czasu, zwracając wynik >, =, <	T, C, D S1 musi być mniejsze lub równe S2			Y, M, S	TZCP, TZCPP: 9 kroków
uwaga: użyte 3 kolejne urządzenia						

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Zawartość:**

S₁, S₂ i S reprezentują wartości czasowe. Każde z nich określa adres początkowy 3 urządzeń danych. S jest porównywane z okresem czasu określonego przez S₁ i S₂. Wynik jest przedstawiony w 3 urządzeniach bitowych, określonych przez adres początkowy w D.

Urządzenia bitowe w D oznaczają:

D₊₀ jest ustawiony w stan ON, kiedy czas w S jest mniejszy niż czas w S₁ i S₂

D₊₁ jest ustawiony w stan ON, kiedy czas w S jest pomiędzy S₁ i S₂

D₊₂ jest ustawiony w stan ON, kiedy czas w S jest większy niż czas w S₁ i S₂

Uwagi:

- Status urządzeń docelowych jest zachowany, nawet jeżeli instrukcja TCMP jest nieaktywna
- Porównanie jest oparte na wartości czasu określonej w urządzeniach źródłowych
 - dopuszczalny zakres wartości dla S₁+0, S₂+0 i S+0 wynosi 0-23 (godziny)
 - dopuszczalny zakres wartości dla S₁+1, S₂+1 i S+1 wynosi 0-59 (minuty)
 - dopuszczalny zakres wartości dla S₁+2, S₂+2 i S+2 wynosi 0-59 (sekundy)

5.14.3 TADD (FNC 162)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

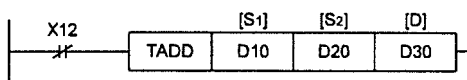
Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
TADD FNC 162 (sumowanie czasu)	dodaje dwie wartości czasu	T, C, D uwaga: do reprezentacji godzin, minut i sekund użyte są 3 kolejne urządzenia			TADD, TADDP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	zero M8020 przeniesienie M8022
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

Zawartość:

S₁, S₂ i D określają adres początkowy 3 urządzeń danych, użytych jako wartość czasu.

Wartość czasu w S₁ jest dodana do wartości czasu w S₂, wynik jest przechowywany w D jako nowa wartość czasu.

**Uwagi:**

- a) Dodawanie jest wykonywane zgodnie ze standardowymi wartościami czasu. Godziny, minuty i sekundy są utrzymane w odpowiednich zakresach. Wszelkie przepełnienia są korygowane.

S1	S2	D
D10: 10 h	D20: 3 h	D30: 13 h
D11: 30 min	D21: 10 min	D31: 41 min
D12: 27 s	D22: 49 s	D32: 16 s
10:30:27	03:10:49	13:41:16

- b) Jeżeli dodanie dwóch czasów daje wynik większy niż 24 godziny, wartość wyniku jest czasem pozostającym po 24 godzinach.

S1	S2	D
D10: 10 h	D20: 18 h	D30: 13 h
D11: 17 min	D21: 12 min	D31: 41 min
D12: 29 s	D22: 34 s	D32: 16 s
10:17:29	18:12:34	04:30:03

Kiedy tak się zdarzy, znacznik przeniesienia, M8022, jest ustawiony w stan ON.

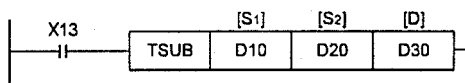
- c) Jeżeli dodanie dwóch czasów daje wynik równy 0 (0:00:00 0 godzin, 0 minut, 0 sekund) wtedy znacznik zera, M8020, jest ustawiony w stan ON.
- d) To samo urządzenie może być użyte jako źródłowe (S₁ lub S₂) i jako docelowe. W tym przypadku dodawanie wykonywane jest w sposób ciągły; wartość docelowa jest zmieniana przy każdym przebiegu programu. Aby tego uniknąć należy użyć modyfikatora impulsu albo blokady programowej.

5.14.4 TSUB (FNC 163)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
TSUB FNC 163 (odejmowanie czasu)	odejmuje jedną wartość czasu od drugiej	T, C, D uwaga: do reprezentacji godzin, minut i sekund użyte są 3 kolejne urządzenia			TSUB, TSUBP: 7 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe					znaczniki	zero M8020 pożyczka M8021
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}		

**Zawartość:**

S1, S2 i D określają adres początkowy 3 urządzeń danych, użytych jako wartość czasu.

Wartość czasu w S1 jest odejmowana od wartości czasu w S2, wynik jest przechowywany w D jako nowa wartość

czasu.

Uwagi:

- a) Odejmowanie jest wykonywane zgodnie ze standardowymi wartościami czasu. Godziny, minuty i sekundy są utrzymane w odpowiednich zakresach. Wszelkie niedopełnienia są korygowane.

S1			S2		=	D	
D10: 10 h			D20: 3 h			D30: 7 h	
D11: 30 min		-	D21: 10 min			D31: 19 min	
D12: 27 s			D22: 49 s			D32: 38 s	
10:30:27			03:10:49			07:19:38	

- b) Jeżeli odjęcie dwóch czasów daje wynik mniejszy niż 00:00:00 godzin, wartość wyniku jest czasem pozostającym poniżej 00:00:00 godzin.

S1			S2		=	D	
D10: 10 h			D20: 18 h			D30: 16 h	
D11: 17 min		-	D21: 12 min			D31: 04 min	
D12: 29 s			D22: 34 s			D32: 55 s	
10:17:29			18:12:34			16:04:55	

Kiedy tak się zdarzy, znacznik pożyczki, M8021, jest ustawiony w stan ON.

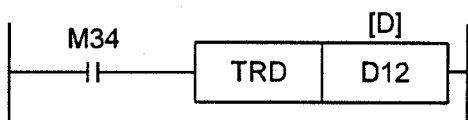
- c) Jeżeli odjęcie dwóch czasów daje wynik równy zero (00:00:00 godzin), wtedy znacznik zera, M8020, jest ustawiony w stan ON.
- d) To samo urządzenie może być użyte jako źródłowe (S1 lub S2) i jako docelowe. W tym przypadku odejmowanie wykonywane jest w sposób ciągły; wartość docelowa jest zmieniana przy każdym przebiegu programu. Aby tego uniknąć należy użyć modyfikatora impulsu albo blokady programowej.

5.14.5 TRD (FNC 166)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty			Kroki programu
		S ₁	S ₂	D	
TRD FNC 166 (odczyt czasu)	odczytuje bieżącą wartość czasu z RTC do rejestrów	T, C, D	uwaga: do reprezentacji godzin, minut i sekund użyte są 3 kolejne urządzenia		TRD, TRDP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Zawartość:**

Czas bieżący i data z zegara czasu rzeczywistego są odczytywane i zapamiętywane w 7 urządzeniach, określonych przez adres początkowy D.

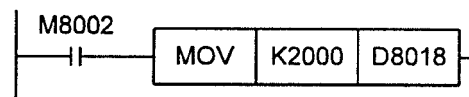
7 urządzeń jest ustawionych w sposób następujący:

Urządzenie	Znaczenie	Wartości
D8018	Rok	00-99
D8017	Miesiąc	01-12
D8016	Dzień	01-31
D8015	Godziny	00-23
D8014	Minuty	00-59
D8013	Sekundy	00-59
D8019	Dzień tygodnia	0-6 (Nd-Sob)

Urządzenie	Znaczenie
D+0	Rok
D+1	Miesiąc
D+2	Dzień
D+3	Godziny
D+4	Minuty
D+5	Sekundy
D+6	Dzień tygodnia

Uwagi:

Rok jest odczytywany jako liczba dwucyfrowa. Można to zmienić na liczbę czterocyfrową przestawiając D8018 na 2000 podczas pierwszego przebiegu programu; zobacz poniższy przykład:



Jeżeli tak zrobiono, wtedy wartość roku z zegara nie powinna być użyta podczas pierwszego przebiegu, ponieważ będzie to liczba 2-cyfrowa przed wykonaniem instrukcji, a wartość 2000 po wykonaniu instrukcji, aż do momentu, kiedy wykona się instrukcja END. Po pierwszym przebiegu rok jest odczytywany i zapisywany jako liczba 4-cyfrowa.



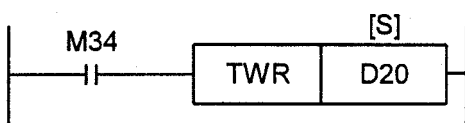
Sterowniki FX-10DU-E, FX-20DU-E i FX25DU-E wspomagają tylko rok 2-cyfrowy.

5.14.6 TWR (FNC 167)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty	Kroki programu
		S	
TWR FNC 167 (zapis czasu)	ustawia zegar czasu rzeczywistego	T, C, D uwaga: użytych jest 7 kolejnych urządzeń	TWR, TWRP: 5 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Zawartość:**

7 urządzeń określonych przez adres początkowy S użytych jest do ustawienia nowych wartości zegara czasu rzeczywistego

7 urządzeń jest ustawionych w sposób następujący:

Urządzenie	Znaczenie	Wartości
S ₊₀	Rok	00-99
S ₊₁	Miesiąc	01-12
S ₊₂	Dzień	01-31
S ₊₃	Godziny	00-23
S ₊₄	Minuty	00-59
S ₊₅	Sekundy	00-59
S ₊₆	Dzień tygodnia	0-6 (Nd-Sob)


Urządzenie	Znaczenie
D8018	Rok
D8017	Miesiąc
D8016	Dzień
D8015	Godziny
D8014	Minuty
D8013	Sekundy
D8019	Dzień tygodnia

Uwagi:

Ta instrukcja usuwa potrzebę użycia M8015 podczas ustawiania zegara czasu rzeczywistego. Podczas ustawiania czasu zaleca się wprowadzić dane źródłowe kilka minut naprzód i uruchomić zegar gdy czas rzeczywisty osiągnie tę wartość.

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
 15. Kody Graya	5-144
16. Porównania operacyjne	5-148

5.15 Kody Graya - funkcje 170 - 179

Zawartość:

			Strona
GRY	konwersja dziesiętna na kod Graya	FNC 170	5-145
GBIN	konwersja kodu Graya na dziesiętną	FNC 171	5-145
☆☆☆	niedostępna	FNC 172-177	



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

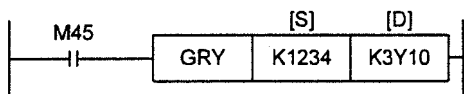
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.15.1 GRY (FNC 170)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
GRY FNC 170 (Kod Graya)	oblicza wartość kodu Graya	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	GRY, GRYP: 5 kroków DGRY, DGRYP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Całkowita wartość binarna w S jest konwertowana do równoważnej postaci w kodzie Graya i przechowywana w D

Uwagi:

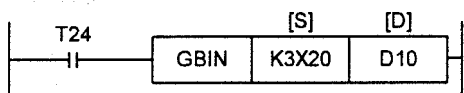
Natura liczb w kodzie Graya umożliwia ich szybkie użycie bez potrzeby stosowania sygnału strobującego. Na przykład, jeżeli dana źródłowa jest inkrementowana w sposób ciągły, nowe dane wyjściowe mogą być ustawiane w każdym przebiegu programu.

5.15.2 GBIN (FNC 171)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S	D	
GBIN FNC 171 (Kod Graya)	oblicza całkowitą wartość kodu Graya	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	GBIN, GBINP: 5 kroków DBIN, DBINP: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}

**Działanie:**

Wartość w kodzie Graya w S jest konwertowana do równoważnej postaci binarnej i przechowywana w D.


Uwagi:

Ta instrukcja może być użyta do czytania wartości z kodera Graya. Jeżeli źródło jest ustawione na wejścia X0-X17, możliwe jest przyspieszenie czasu czytania przez dopasowanie filtra odświeżającego w instrukcji REFF (FNC 51).

Notatki

Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

1. Sterowanie przebiegiem programu	5-4
2. Instrukcje przesłania i porównania	5-16
3. Operacje arytmetyczne i logiczne	5-24
4. Rotacje i funkcje przesunięcia	5-34
5. Operacje na danych	5-42
6. Szybkie przetwarzanie	5-52
7. Instrukcje podręczne	5-66
8. Zewnętrzne urządzenia wejścia/wyjścia	5-80
9. Zewnętrzne urządzenia szeregowo	5-94
10. Zewnętrzne jednostki F2	5-112
11. Operacje zmiennoprzecinkowe 1 & 2	5-120
12. Operacje trygonometryczne	5-128
13. Operacje na danych 2	5-132
14. Sterowanie zegarem czasu rzeczywistego	5-136
15. Kody Graya	5-144
 16. Porównania operacyjne	5-148

5.16 Porównania operacyjne - funkcje 220 - 249

Zawartość:

			Strona
LD□	LD z porównaniem	FNC 224-230	5-121
AND□	AND z porównaniem	FNC 232-238	5-122
OR□	OR z porównaniem	FNC 240-246	5-122



Lista symboli:

D - urządzenie przeznaczenia

S – urządzenie źródłowe

m, n - liczba aktywnych urządzeń, bitów lub stałych operacyjnych

Dodatkowe przyrostki numeryczne będą dołączone, jeżeli jest więcej niż jeden argument w tej samej funkcji, np. D_1, S_3 lub dla urządzeń stabilizowanych D_{3+0}, S_{+9} itd.

MSB - najbardziej znaczący bit, czasami używany do oznaczenia znaku matematycznego liczby, np. dodatni = 0, ujemny = 1

LSB - najmniej znaczący bit

Modyfikacje instrukcji:

☆☆☆ - instrukcja operująca na 16 bitach, gdzie xxx oznacza mnemonik instrukcji

☆☆☆P - instrukcja 16-bitowa, zmodyfikowana dla operacji impulsowej

D☆☆☆ - instrukcja operująca na 32 bitach

D☆☆☆P - instrukcja operująca na 32 bitach, zmodyfikowana dla operacji impulsowej

→ - instrukcja szybkiego powtarzania, która zmienia wartość końcową przy każdym cyklu programowym, jeśli nie jest wyzwalana przez funkcję impulsową

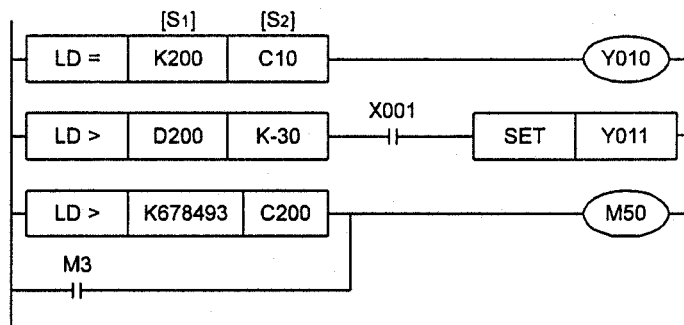
☒ - argument, który nie może być indeksowany, tzn. dodanie V lub Z jest albo nieprawidłowe, albo nie będzie miało wpływu na wartość argumentu.

5.16.1 LD compare (FNC 224 do 320)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S ₁	S ₂	
LD□ (LD z porównaniem) gdzie □: =, <, >, <>, ≤, ≥	inicjuje styk porównania Aktywna, gdy porównanie S1□S2 jest prawdziwe	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	LD□: 5 kroków DLD□: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie:

Wartości S1 i S2 są testowane zgodnie z warunkiem porównania instrukcji. Jeżeli porównanie jest prawdziwe, wtedy styk LD jest aktywny. Jeżeli porównanie jest fałszywe, wtedy styk LD nie jest aktywny.

Uwagi:

Funkcje porównania LD mogą być umieszczane gdziekolwiek w programie, gdzie może być użyta standardowa instrukcja LD, tzn. na początku nowego bloku.
(Zobacz strona 2-3 - instrukcja LD)

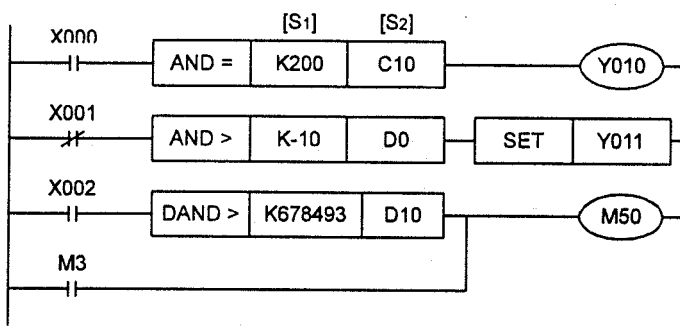
FNC	Mnemonik		Aktywna gdy	Nieaktywna gdy
	16-bitowy	32-bitowy		
224	LD=	DLD=	S ₁ = S ₂	S ₁ ≠ S ₂
225	LD>	DLD>	S ₁ > S ₂	S ₁ ≤ S ₂
226	LD<	DLD<	S ₁ < S ₂	S ₁ ≥ S ₂
228	LD<>	DLD<>	S ₁ ≠ S ₂	S ₁ = S ₂
229	LD≤	DLD≤	S ₁ ≤ S ₂	S ₁ > S ₂
230	LD≥	DLD≥	S ₁ ≥ S ₂	S ₁ < S ₂

5.16.2 AND compare (FNC 232 do 238)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S ₁	S ₂	
AND□ (AND□ z porównaniem) gdzie □: =, <, >, <>, ≤, ≥	inicjuje styk porównania szeregowego Aktywna, gdy porównanie S1□S2 jest prawdziwe	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	AND□: 5 kroków DAND□: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie:

Wartości S1 i S2 są testowane zgodnie z warunkiem porównaniem instrukcji. Jeżeli porównanie jest prawdziwe, wtedy styk AND jest aktywny. Jeżeli porównanie jest fałszywe, wtedy styk AND nie jest aktywny.

Uwagi:

Funkcje porównania AND mogą być umieszczane gdziekolwiek w programie, gdzie może być użyta standardowa instrukcja AND, tzn. w połączeniu szeregowym styków. (Zobacz strona 2-6 - instrukcja AND)

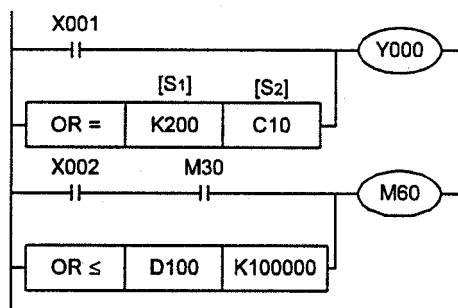
FNC	Mnemonik		Aktywna gdy	Nieaktywna gdy
	16-bitowy	32-bitowy		
232	AND=	DAND=	S ₁ = S ₂	S ₁ ≠ S ₂
233	AND>	DAND>	S ₁ > S ₂	S ₁ ≤ S ₂
234	AND<	DAND<	S ₁ < S ₂	S ₁ ≥ S ₂
236	AND<>	DAND<>	S ₁ ≠ S ₂	S ₁ = S ₂
237	AND≤	DAND≤	S ₁ ≤ S ₂	S ₁ > S ₂
238	AND≥	DAND≥	S ₁ ≥ S ₂	S ₁ < S ₂

5.16.3 OR compare (FNC 240 do 246)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonik	Funkcja	Argumenty		Kroki programu
		S ₁	S ₂	
OR□ (OR□ z porównaniem) gdzie □: =, <, >, <>, ≤, ≥	inicjuje styk porównania równoległego Aktywna, gdy porównanie S1□S2 jest prawdziwe	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	OR□: 5 kroków DOR□: 9 kroków

Instrukcja impulsowa (P)					Operacje 16-bitowe					Operacje 32-bitowe				
FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}	FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}



Działanie:

Wartości S1 i S2 są testowane zgodnie z warunkiem porównania instrukcji. Jeżeli porównanie jest prawdziwe, wtedy styk OR jest aktywny. Jeżeli porównanie jest fałszywe, wtedy styk OR nie jest aktywny.

Uwagi:

Funkcje porównania OR mogą być umieszczane gdziekolwiek w programie, gdzie może być użyta standardowa instrukcja OR, tzn.

w połączeniu równoległym styków.

(Zobacz strona 2-7 - instrukcja OR)

FNC	Mnemonik		Aktywna gdy	Nieaktywna gdy
	16-bitowy	32-bitowy		
240	OR=	DOR=	S ₁ = S ₂	S ₁ ≠ S ₂
241	OR>	DOR>	S ₁ > S ₂	S ₁ ≤ S ₂
242	OR<	DOR<	S ₁ < S ₂	S ₁ ≥ S ₂
244	OR<>	DOR<>	S ₁ ≠ S ₂	S ₁ = S ₂
245	OR≤	DOR≤	S ₁ ≤ S ₂	S ₁ > S ₂
246	OR≥	DOR≥	S ₁ ≥ S ₂	S ₁ < S ₂

Notatki

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

6	Urządzenia diagnostyczne	6-1
6.1	Status PLC (M8000-M8009 i D8000-D8009)	6-2
6.2	Urządzenia zegara (M8000- M8009 i D8000-D8009)	6-3
6.3	Znaczniki operacyjne	6-4
6.4	Tryb programowy PLC (M8030-M8039 i D8030-D8039)	6-5
6.5	Znaczniki STL (M8040-M8049 i D8040-D8049)	6-6
6.6	Znaczniki przerwania (M8050-M8059 i D8050-D8059)	6-7
6.7	Urządzenia detekcji błędów (M8060-M8069 i D8060-D8069)	6-8
6.8	Urządzenia operacji łączących (M8070-M8099 i D8070-D8099)	6-9
6.9	Urządzenia pomocnicze (M8100-M8119 i D8100-D8119)	6-10
6.10	Urządzenia adaptera komunikacyjnego, tzn. 232ADP, 485ADP (M8120-M8129 i D8120-D8129)	6-10
6.11	Znaczniki tabeli porównań HSZ (M8130-M8139 i D8130-D8139)	6-11
6.12	Urządzenia pomocnicze (M8160-M8199)	6-12
6.13	Rejestry indeksowe (D8180 do D8199)	6-13
6.14	Sterowanie licznika góra/dół (M8200-M8234 i D8200-D8234)	6-14
6.15	Sterowanie szybkiego licznika (M8235-M8255 i D8235-D8255)	6-14
6.16	Tabela kodów błędów	6-15

6. Urządzenia diagnostyczne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Następujące urządzenia diagnostyczne są wykorzystywane przez sterownik programowalny do wykazania bieżącego statusu programowego oraz do identyfikacji możliwych błędów. Istnieje kilka odmian w aplikacjach tych urządzeń w rodzinie sterowników FX, które to odmiany będą w odpowiednich miejscach omówione.

Wewnętrzne urządzenia diagnostyczne to zarówno przekaźniki pomocnicze (M) jak i rejestry danych (D). Często spotykana jest współzależność między urządzeniami M i D, np. M8039 identyfikuje stały tryb przebiegu programu, zaś rejestr D8039 zawiera stałą, określającą ustawioną długość czasu przebiegu programu.

Urządzenia, które nie są ustawialne:

- Urządzenie typu M i D, oznaczone „x”, nie może być ustawiane w programie użytkownika. W przypadku urządzeń typu M oznacza to, że cewka nie może byćysterowana, ale można odczytać stan styków. Z kolei dla urządzeń typu D oznacza to, że nie można zapisywać wartości w rejestrze, ale można jej użyć do operacji porównań.

Urządzenia ustawiane domyślnie:

- Urządzenia, oznaczone „☞”, przyjmują domyślne wartości, gdy PLC przechodzi ze stanu OFF na ON.

Podsumowanie użytych symboli:

- x - nie ustawialne przez użytkownika
- ☞ - przyjmują wartości domyślne po włączeniu zasilania
- ☞R - przyjmują wartości domyślne także po przejściu w stan RUN
- ☞S - przyjmują wartości domyślne także po przejściu w stan STOP

6.1 Status PLC (M8000-M8009 i D8000-D8009)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8000 (*) monitor stanu RUN styk normalnie otwarty	
M8001 (*) monitor stanu RUN styk normalnie zwarły	
M8002 (*) impuls początkowy, styk normalnie otwarty	
M8003 (*) impuls początkowy, styk normalnie zwarły	
M8004 (*) pojawienie się błędu	ON, jeśli jeden lub więcej znaczników błędów z zakresu M8060 do M8067 są w stanie ON
M8005 za niskie napięcie baterii	ON, jeśli napięcie baterii jest niższe od wartości ustawionej w D8006
M8006 zatrząsk błędu baterii	zatrząskuje błąd za niskiego napięcia baterii
M8007 chwilowy zanik zasilania	
M8008 błąd zasilania	wystąpił zanik zasilania
M8009 napięcie 24V DC wyłączone	zanik zasilania 24V DC

Urządzenie diagnostyczne	Operacja
D8000 (☞) timer watchdoga	FX, FX _{2C} : 100 ms FX ₀ , FX _{0S} , FX _{0N} , FX _{2N} : 200 ms Patrz uwaga 1
D8001 (*) typ i wersja PLC	20Vv: FX _{0(S)} , FX _{0N} , FX, FX _{2C} wersja V.vv 24Vv: FX _{2N} wersja V.vv
D8002 (*) pojemność pamięci	0002: 2K kroków 0004: 4K kroków 0008: 8K kroków (zobacz też D8102)
D8003 (*) typ pamięci	00H = RAM, 01H = EPROM 02H = EEPROM 0AH = EEPROM (prot.) 10H = pamięć MPU
D8004 (*) numer błędu M☆☆☆☆	zawartość tego rejestru☆☆☆☆ identyfikuje aktywny znacznik błędu, tzn. jeśli ☆☆☆☆=8060, to M8060 jest w stanie ON
D8005 napięcie baterii	Np. 36 = 3,6 V
D8006 poziom niskiego napięcia baterii	poziom, poniżej którego wykrywane jest za niskie napięcie baterii
D8007 liczba błędów zasilania	liczba chwilowych zaników zasilania od włączenia zasilania
D8008 wykrycie zaniku zasilania	okres czasu przed zamknięciem systemu po wykryciu zaniku zasilania (domyślnie 10 ms)
D8009 urządzenie z błędem z powodu zaniku zasilania	najniższy numer urządzenia z błędem z powodu zaniku zasilania 24 V DC

Objaśnienie użytych symboli na stronie 6-1

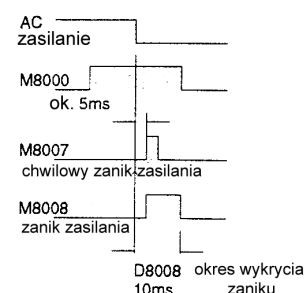


Uwaga 1:

Zawartość tego rejestru może być zmieniona przez użytkownika. Możliwa jest zmiana z krokiem 1 ms. Aby zapewnić stały przebieg programu, wartość powinna być większa od maksymalnego czasu przebiegu programu.

Uwaga ogólna:

Gdy używany jest zasilacz 200V AC, okres wykrywania zaniku zasilania jest określony przez wartość w D8008. Użytkownik może zmieniać tę wartość w zakresie 10-100 ms.



6.2 Urządzenia zegara (M8000-M8009 i D8000-D8009)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8010	zarezerwowane
M8011 (*) impuls zegarowy 10 ms	oscylacje 10 ms
M8012 (*) impuls zegarowy 100 ms	oscylacje 100 ms
M8013 (*) impuls zegarowy 1 s	oscylacje 1 s
M8014 (*) impuls zegarowy 10 min	oscylacje 1 min

Urządzenie diagnostyczne	Operacja
D8010 (*) bieżący czas przebiegu programu	bieżący cykl operacyjny/czas przebiegu programu w 0,1 ms
D8011 (*) minimalny czas przebiegu programu	minimalny cykl operacyjny/czas przebiegu programu w 0,1 ms
D8012 (*) maksymalny czas przebiegu programu	maksymalny cykl operacyjny/czas przebiegu programu w 0,1 ms

Następujące urządzenia dotyczą sterownika FX_{2N}
i FX_{0N}, FX i FX_{2C} gdy jest zainstalowana kasetka zegara czasu rzeczywistego

M8015 ustawianie czasu	gdy ON - zatrzymanie zegara, ON ⇔ OFF - restart zegara	D8013 sekundy	dane sekund, używane z kasetą RTC (0-59) (dla FX _{0(S)} , FX _{0N} patrz uwaga 2)
M8016 rejestr	gdy ON, rejestry D8013 do D8019 są "zamrożone", ale zegar pracuje	D8014 minuty	dane minut, używane z kasetą RTC (0-59)
M8017 zaokrąglenie minut	po impulsie ON ustawia RTC na następną minutę	D8015 godziny	dane godzin, używane z kasetą RTC (0-23)
M8018 (*) dostępność RTC	gdy ON, zegar czasu rzeczywistego jest zainstalowany	D8016 dni	dane dnia, używane z kasetą RTC (1-31)
M8019 błąd ustawienia	dane zegara ustawione poza zakresem	D8017 miesiące	dane miesiąca, używane z kasetą RTC (1-12)
		D8018 lata	dane roku, używane z kasetą RTC (0-99)
		D8019 dzień tygodnia	dane dnia tygodnia, używane z kasetą RTC (0-6)

Objaśnienie użytych symboli na stronie 6-1

Uwaga 2:

- Dla sterowników FX₀, FX_{0S} i sterowników FX_{0N} nie wyposażonych z RTC, rejestr D8013 reprezentuje wartość odczytaną w pierwszym potencjometru, z zakresu 0-255, w ms.

6.3 Znaczniki operacyjne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja	Urządzenie diagnostyczne	Operacja
M8020 (*) Zero	ON, gdy wynikiem instrukcji ADD (FNC 20) lub SUB (FNC 21) jest "0"	D8020 (tylko FX ₀ /FX _{0S} /FX _{0N})	ustawienie filtrów wejściowych dla X000 do X007, domyślnie 10 ms, (0-15)
M8021 (*) Pożyczka	ON, gdy wynik instrukcji SUB (FNC 21) jest mniejszy od najmniejszej liczby ujemnej	D8020 (tylko FX _{2N})	ustawienie filtrów wejściowych dla X000 do X017, domyślnie 10 ms, (0-15)
M8022 (*) Przeniesienie	ON, gdy w wyniku instrukcji ADD (FNC 20) pojawia się przeniesienie lub powstał nadmiar w wyniku operacji przesuwania	D8021 (tylko FX ₀ /FX _{0S})	ustawienie filtrów wejściowych dla X010 do X017, domyślnie 10 ms, (0-15)
M8023 (*) Operacja zmiennoprzecinkowa (tylko FX _{2C})	gdy ON, dostępne są operacje zmiennoprzecinkowe. Znacznik jest używany przez instrukcje: BCD, BIN, ADD, SUB, MUL, DIV, SQR i FLT		
M8024	zarezerwowane		
M8025 (nie dotyczy FX _{0(S)} , FX _{0N})	gdy ON, instrukcje HSC (FNC 53-55) są wykonywane, nawet gdy zewnętrzne wejścia zerujące HSC są aktywne	D8022-D8027	zarezerwowane
M8026(nie dotyczy FX _{0(S)} , FX _{0N})	tryb "hold" instrukcji RAMP (FNC 67)		
M8027(nie dotyczy FX _{0(S)} , FX _{0N})	16-elementowy ciąg danych instrukcji PR (FNC 77)		
M8028	FX _{0(S)} , FX _{0N} : zmiana timerów T32 do T55 na 10 ms FX, FX _{2C} , FX _{2N} : odblokowanie przerwań podczas instrukcji FROM/TO	D8028 (*)	bieżąca wartość rejestru indeksowego Z
M8029 (*)	ON po zakończeniu działania instrukcji DSW (FNC 72), RAMP (FNC 67) itd.	D8029 (*)	bieżąca wartość rejestru indeksowego V

Objaśnienie użytych symboli na stronie 6-1

Uwaga dotycząca filtrów wejściowych:

- Ustawienia filtrów wejściowych dotyczą tych MPU, które wykorzystują wejścia 24 V DC. Filtry AC nie są ustawialne.

6.4 Tryb programowy PLC (M8030-M8039 i D8030-D8039)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8030 (👉) LED baterii wyłączona	napięcie baterii jest za niskie, ale LED baterii nie świeci
M8031 (👉) zerowanie pamięci nie-zatraskowych	Ustawienia bieżącego urządzenia są zerowane przy następnym END, tzn. styki, cewki i bieżące wartości danych dla urządzeń, odpowiednio, Y, M, S, T, C i D. Urządzenia specjalne, które mają ustawienia domyślne, są ustawiane na te wartości.
M8032 (👉) zerowanie pamięci zatraskowych	
M8033 (👉) zachowanie pamięci w trybie 'stop'	Statusy urządzeń i ustawienia są zachowywane, gdy PLC przechodzi ze stanu RUN do STOP i znów na RUN
M8034 zablokowanie wszystkich wyjść	wszystkie fizyczne wyjścia są zablokowane, jednak program działa normalnie i wykazuje stany wyjść tylko poprzez LED-y wyjściowe
M8035 (👉S) tryb operacji wymuszonej	Poprzez użycie trybu operacji wymuszonej, tzn. M8035 w stanie ON, możliwe jest wykonanie operacji zdalnego lub impulsowego RUN/STOP. Przykład w rozdziale 10.
M8036 (👉S) wymuszenie sygnału RUN	
M8037 (👉S) wymuszenie sygnału STOP	
M8038 (👉) zerowanie rejestrów plikowych RAM (tylko FX _{2C})	Używane do zerowania zawartości 2000 rejestrów plikowych RAM
M8039 (👉) tryb stałego przebiegu	Gdy jest w stanie ON, program jest wykonywany ze stałym czasem przebiegu. Różnica pomiędzy bieżącym końcem operacji a ustawionym czasem przebiegu jest wypełniania „przerwą”.

Urządzenie diagnostyczne	Operacja
D8030 (*) (tylko FX _{0N})	wartość odczytana z pierwszego potencjometru w ms (0-255)
D8030 (nie FX _{0(S)} , FX _{0N})	Ten rejestr jest używany jako 3 urządzenie zapamiętujące, gdy rejestry Z lub V zostały wybrane jako przeznaczenie dla instrukcji SPD (FNC 56)
D8031 (*) (tylko FX _{0N})	wartość odczytana z drugiego potencjometru w ms (0-255)
D8032-D8038	zarezerwowane
D8039 (👉) stała przebiegu programu	W tym rejestrze użytkownik może zdefiniować czas przebiegu programu z rozdzielczością 1 ms. Domyślna wartość rejestru, ustawiana podczas włączania zasilania, wynosi 0.

Objaśnienie użytych symboli na stronie 6-1

6.5 Znaczniki STL (M8040-M8049 i D8040-D8049)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8040 (☞)	Gdy jest w stanie ON, zablokowany jest transfer STL
M8041 (☞S)	Gdy jest w stanie ON, transfer STL jest automatycznie odblokowany
M8042 (☞)	w odpowiedzi na sygnał wejściowy generowany jest impuls wyjściowy (patrz instrukcja IST (FNC 60))
M8043 (☞S)	ON podczas ostatniego stanu trybu ZERO RETURN (patrz instrukcja IST (FNC 60))
M8044 (☞S)	ON, gdy wykryto zero maszynowe (patrz instrukcja IST (FNC 60))
M8045 (☞)	blokuje funkcje zerowania wszystkich wyjść gdy zmieniany jest tryb operacji (patrz instrukcja IST (FNC 60))
M8046 (*)	ON, gdy włączone jest monitorowanie STL i stan STL jest aktywny
M8047 (☞)	Gdy ON, D8040 do D8047 są odblokowane do monitorowania aktywnych kroków STL
M8048 (*)	ON, gdy monitorowanie annunciatora jest odblokowane i znacznik annunciatora jest aktywny
M8049 (☞)	Gdy ON, rejestr D8049 jest wyznaczony do monitorowania stanu aktywnego annunciatora

Urządzenie diagnostyczne	Operacja
D8040 (*)	najniższy aktywny krok STL
D8041 (*)	drugi aktywny stan STL
D8042 (*)	trzeci aktywny stan STL
D8043 (*)	czwarty aktywny stan STL
D8044 (*)	piąty aktywny stan STL
D8045 (*)	szósty aktywny stan STL
D8046 (*)	siódmy aktywny stan STL
D8047 (*)	ósmo aktywny stan STL
D8048	zarezerwowany
D8049 (*)	najniższy aktywny annunciator (Nie dla FX _{0(S)} , FX _{0N})

Aktywne stany STL (do 8) z zakresu S0 do S999 są przechowywane w D8040 do D8047 w porządku narastającym (aktualizowane podczas END)

Objaśnienie użytych symboli na stronie 6-1

Uwaga ogólna: Wszystkie stany STL są aktualizowane podczas działania instrukcji END

6.6 Znaczniki przerwania (M8050-M8059 i D8050-D8059)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8050 (👉) blokada I00□	Podczas wykonania instrukcji EI (FNC 04) wszystkie przerwania są odblokowane, chyba że wyszczególniony obok znacznik M jest w stanie ON. W takim przypadku przerwanie jest zamaskowane, tzn. nie będzie działać. Należy zauważyć, że □□ opisuje wszystkie typy tego przerwania
M8051 (👉) blokada I10□	
M8052 (👉) blokada I20□	
M8053 (👉) blokada I30□	

Urządzenie diagnostyczne	Operacja
D8050-D8059	Zarezerwowane

Dotyczy tylko sterowników FX, FX_{2C},
FX_{2N}

M8054 (👉) blokada I40□	Podczas wykonania instrukcji EI (FNC 04) wszystkie przerwania są odblokowane, chyba że wyszczególniony obok znacznik M jest w stanie ON. W takim przypadku przerwanie jest zamaskowane, tzn. nie będzie działać. Należy zauważyć, że □□ opisuje wszystkie typy tego przerwania
M8055 (👉) blokada I50□	
M8056 (👉) blokada I60□	
M8057 (👉) blokada I7□□	
M8058 (👉) blokada I8□□	
M8059 (👉) blokada I010 do I060 jako pojedynczej grupy (tylko FX _(2C) i FX _{2N})	

Dotyczy tylko sterowników FX₀, FX_{0S} i FX_{0N}

M8054	Zarezerwowane
M8055	Zarezerwowane
M8056 (👉) przechwycenie impulsu X0	Gdy zostanie odebrane narastające zbocze impulsu na wejściu z zakresu X0 do XM, przyporządkowany do wyszczególnionego urządzenia znacznik M jest ustawiany w stan ON. Następne pojawienie się impulsu ponownie ustawia znacznik w stan ON, po uprzednim wyzerowaniu go w programie użytkownika. Stąd szybkie impulsy wejściowe są wychwytywane i zapamiętywane. Operacja działa niezależnie od instrukcji EI (FNC 04) i DI (FNC 05). Działanie dla FX, FX _{2C} i FX _{2N} opisano na stronie 6-12
M8057 (👉) przechwycenie impulsu X1	
M8058 (👉) przechwycenie impulsu X2	
M8059 (👉) przechwycenie impulsu X3	

Objaśnienie użytych symboli na stronie 6-1

6.7 Urządzenia detekcji błędów (M8060-M8069 i D8060-D8069)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Działanie		
	Wykrywanie	LED	Status PLC
M8060 (*) błąd konfiguracji I/O (nie dotyczy FX _{0(S)} , FX _{0N})	podczas stanu RUN	wył.	RUN
M8061 (*) błąd sprzętowy PLC		błyska	STOP
M8062 (*) błąd komunikacji PCL/HPP	podczas odbierania sygnału z HPP	wył.	RUN
M8063 (*) (R) błąd łącza równoległego ADP	podczas odbierania sygnału z połączonej stacji		
M8064 (*) błąd parametru	Podczas zmiany programu (PLC w stanie STOP) i podczas przenoszenia programu (PLC w stanie STOP)	błyska	STOP
M8065 (*) błąd syntaktyczny			
M8066 (*) błąd programu			
M8067 (*) (R) błąd instrukcji	podczas stanu RUN	wył.	RUN
M8068 () zatrzaśnięty błąd operacji			
M8069 (R) błąd magistrali I/O (nie dotyczy FX _{0(S)} , FX _{0N})	patrz uwaga 4.	-	-

Urządzenie diagnostyczne	Działanie
D8060 (*) (nie dotyczy FX _{0(S)} , FX _{0N})	Numer pierwszego I/O jednostki lub bloku, która spowodowała błąd – patrz uwaga 3.
D8061 (*)	Kod błędu sprzętowego – zobacz także odpowiedni dla PLC podręcznik
D8062 (*) (nie dotyczy FX _{0(S)} , FX _{0N})	Kod błędu komunikacji – zobacz odpowiednią tablicę błędów
D8063 (*) (R) (nie dotyczy FX _{0(S)} , FX _{0N})	Kod błędu połączenia równoległego – zobacz podręcznik, identyfikację błędów komunikacji ADP, podręcznik użytkownika FX-485PC-IF
D8064 (*)	Kod błędu parametru – zobacz odpowiednią tabelę kodów błędów
D8065 (*)	Kod błędu syntaktycznego – zobacz odpowiednią tabelę błędów
D8066 (*)	Kod błędu programu – zobacz odpowiednią tabelę błędów
D8067 (*) (R)	Kod błędu instrukcji
D8068 (*)	Zatrzaśnięty numer kroku błędu operacji
D8069 (*) (R)	Numery kroków dla znalezionych błędów, odpowiadającym znacznikom M8065 do M8067

Objaśnienie użytych symboli na stronie 6-1



Uwagi, dotyczące tej tabeli znajdują się na następnej stronie



Uwaga 3:

Jeżeli jednostka lub blok odpowiadający zaprogramowanemu numerowi I/O nie jest dołączony, znacznik M8060 jest ustawiony w stan ON, a numer pierwszego urządzenia błędnego bloku jest zapisywany w D8060.

Zawartość D8060

1	0	2	0
---	---	---	---

 = X 20

 Numer urządzenia:
10 do 177

 Typ urządzenia:
1 - Wejście X
0 - Wyjście Y

Uwaga 4:

- Kontrola stanu magistrali I/O jest wykonywana, gdy M8069 jest w stanie ON. Jeżeli pojawi się błąd, do rejestru D8069 wpisywany jest kod błędu 6103 lub 6104, a znacznik M8061 jest ustawiany w stan ON.

Uwaga ogólna:

- Skrót HPP oznacza podręczny panel programujący (Handy Programming Panel)

6.8 Urządzenia operacji łączących (M8070-M8099 i D8070-D8099)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8070 (R)	Aktywowany, gdy PLC jest stacją Master w połączeniu równoległym
M8071(R)	Aktywowany, gdy PLC jest stacją Slave w połączeniu równoległym
M8072 (*)	ON, gdy PLC jest aktywny w połączeniu równoległym
M8073(*)	ON, gdy M8070/M8071 są nieprawidłowo ustawione podczas operacji połączenia równoległego
M8074 (Tylko FX _(2C))	Aktywizacja rejestrów plikowych RAM
M8075-M8098	Zarezerwowane
M8099 (R)	Operacja szybkiego timera

Urządzenie diagnostyczne	Operacja
D8070 (*)	Czas watchdoga dla połączenia równoległego (500 ms)
D8071-D8098	Zarezerwowane
D8099	Timer cykliczny, zakres 0-32,767 w jednostkach 0,1 ms (używany podczas pomiaru czasu trwania impulsów wejściowych)

Objaśnienie użytych symboli na stronie 6-1

6.9 Urządzenia pomocnicze (M8100-M8119 i D8100-D8119)

Urządzenie diagnostyczne	Operacja
M8109 (*)	błąd odświeżania wyjścia

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
D8102 (*) pojemność pamięci	0002: 2K kroków 0004: 4K kroków 0008: 8K kroków 0016: 16K kroków
D8109 (*)	numer urządzenia, gdzie wystąpił błąd odświeżania wyjścia: 0, 10, 20, itd.

6.10 Urządzenia adaptera komunikacyjnego, tzn. 232ADP, 485ADP (M8120-M8129 i D8120-D8129)

Urządzenie diagnostyczne	Operacja
M8120	znacznik backup dla sieci 485 – tylko FX _{0N}
M8121 () (☞R)	RS – opóźnienie transmisji danych
M8122 (☞R)	RS – znacznik transmisji danych
M8123 (☞R)	zakończono odbieranie danych
M8124 (tylko FX/FX _{2C})	RS – znacznik detekcji nośnej
M8125	zarezerwowane
M8126	RS485 – znacznik globalny
M8127	RS485 - On Demand znacznik potwierdzenia
M8128	RS485 – On Demand znacznik błędu
M8129	RS485 – On Demand znacznik bajt/słowo, ON = bajt, OFF = słowo

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
D8120	format komunikacji
D8121	numer lokalny stacji dla sieci 485
D8122 (*) (☞R)	RS – ilość danych pozostałych do transmisji
D8123 (*) (☞R)	RS – ilość danych już odebranych
D8124	RS – znak nagłówek danych, domyślnie STX (02H)
D8125	232ADP – znak końca danych, domyślnie ETX (03H)
D8126	zarezerwowane
D8127	RS485 - On Demand rejestr nagłówek
D8128	RS485 – On Demand rejestr długości danych
D8129	wartość „time-out” sieci RS485

6.11 Znaczniki tabeli porównań HSZ (M8130-M8139 i D8130-D8139)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8130	wybiera tabele porównań do użycia z instrukcją HSZ
M8131 (✖) (☞)	wskazuje, kiedy tabela HSZ jest przetwarzana
M8132	wybiera użycie instrukcji PLSY z tabelą HSZ
M8133 (✖) (☞)	wskazuje kiedy tabela HSZ (z instrukcją PLSY) jest przetwarzana
M8134-M8139	zarezerwowane

Urządzenie diagnostyczne	Operacja
D8130 (✖) (☞)	zawiera numer bieżącego rekordu przetwarzanego w tabeli HSZ
D8131 (✖) (☞)	zawiera numer bieżącego rekordu przetwarzanego w tabeli HSZ, gdy odblokowana jest operacja PLSY
D8132 (✖) (☞)	zawiera wartość częstotliwości impulsów wyjściowych dla instrukcji PLSY
D8133	zarezerwowane
D8134, D8135 (✖) (☞)	zawiera kopię wartości bieżącego porównania gdy użyta jest tabela HSZ z instrukcją PLSY. Dana dostępna tylko w formacie 32-bitowym
D8136, D8137 (✖) (☞)	zawiera całkowitą liczbę impulsów, wygenerowanych na wyjściu Y0 przez instrukcję PLSY lub PLSR. Dana dostępna tylko w formacie 32-bitowym
D8138-D8139	zarezerwowane
D8140, D8141 (✖) (☞) (tylko FX _{2N})	zawiera całkowitą liczbę impulsów, wygenerowanych na wyjściu Y0 przez instrukcję PLSY lub PLSR. Dana dostępna tylko w formacie 32-bitowym
D8142, D8143 (✖) (☞) (tylko FX _{2N})	zawiera całkowitą liczbę impulsów, wygenerowanych na wyjściu Y1 przez instrukcję PLSY lub PLSR. Dana dostępna tylko w formacie 32-bitowym

Objaśnienie użytych symboli na stronie 6-1

6.12 Urządzenia pomocnicze (M8160-M8199)

Urządzenie diagnostyczne	Operacja
M8160	wybór operacji XCH do zamiany bajtów w pojedynczym słowie danych
M8161	wybór 8-bitowych operacji dla instrukcji specjalnych ASC, RS, ASCI, HEX, CCD
M8162	tryb szybki dla instrukcji PRUN, 2 słowa danych, tylko odczyt
M8163-M8166	zarezerwowane
M8167	wybiera tryb szesnastkowy dla instrukcji HKY
M8168	wybiera tryb BCD dla instrukcji SMOV
M8169	zarezerwowane
M8170 (R)	gdy zostanie odebrane narastające zbocze impulsu na wejściu X0-X5, przyporządkowany do niego znacznik M jest ustawiany w stan ON. Po wyzerowaniu znacznika przez program jest on ustawiany w stan ON przy następnym pojawieniu się impulsu. Stąd możliwość wychwytywania szybkich impulsów. Ta operacja wymaga aktywnej instrukcji EI (FNC 04). Więcej szczegółów na stronie 6-7
M8171 (R)	
M8172 (R)	
M8173 (R)	
M8174 (R)	
M8175 (R)	
M8176-M8179	

Objaśnienie użytych symboli na stronie 6-1

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja	
M8180	zarezerwowane	
M8181 (I010)	Te znaczniki mogą być wprowadzone do instrukcji HSCS (FNC 53) i HSCR (FNC 54) zamiast przyporządkowanych etykiet przerwań, gdy używane jest starsze oprogramowanie lub podręczne programatory. Więcej szczegółów we „Wprowadzeniu” do tego podręcznika	
M8182 (I020)		
M8183 (I030)		
M8184 (I040)		
M8185 (I050)		
M8186 (I060)		
M8187-M8189	zarezerwowane	
M8190 (+MOV=SQR)	Te znaczniki mogą modyfikować instrukcje pokazane w nawiasach jako ‘+☆☆☆’ aby wykonywały instrukcje pokazane jako ‘=☆☆☆’. Pozwala to na dostęp do instrukcji przedstawionych w wersjach CPU 3.07 i jednostkach FX _{2C} przy pomocy starszego oprogramowania lub podręcznych programatorów. Więcej szczegółów we „Wprowadzeniu” do tego podręcznika	
M8191 (+MOV=FLT)		
M8192 (+SMOV=SORT)		
M8193 (+RAMP=SER)		
M8194 (+RAMP=RS)		
M8195 (+FMOV=CCD)		
M8196 (+FMOV=ASCII)		
M8197 (+FMOV=HEX)		
M8198		gdy ON, odwraca operację urządzeń źródłowych i docelowych specyfikowanych w instrukcji. Więcej szczegółów we „Wprowadzeniu” do tego podręcznika
M8199		zarezerwowane

6.13 Rejestry indeksowe (D8180 do D8199)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
D8180 (✖)	zarezerwowane
D8181 (✖)	
D8182 (✖)	rejestr indeksowy Z1
D8183 (✖)	rejestr indeksowy V1
D8184 (✖)	rejestr indeksowy Z2
D8185 (✖)	rejestr indeksowy V2
D8186 (✖)	rejestr indeksowy Z3
D8187 (✖)	rejestr indeksowy V3
D8188 (✖)	rejestr indeksowy Z4
D8189 (✖)	rejestr indeksowy V4

Urządzenie diagnostyczne	Operacja
D8190 (✖)	rejestr indeksowy Z5
D8191 (✖)	rejestr indeksowy V5
D8192 (✖)	rejestr indeksowy Z6
D8193 (✖)	rejestr indeksowy V6
D8194 (✖)	rejestr indeksowy Z7
D8195 (✖)	rejestr indeksowy V7
D8196 (✖)	
D8197 (✖)	
D8198 (✖)	
D8199 (✖)	

6.14 Sterowanie licznika góra/dół (M8200-M8234 i D8200-D8234)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8200 – M8234 (☞)	gdy M8☆☆☆ jest aktywny, licznik C☆☆☆ pracuje jako licznik odejmujący. Gdy M8☆☆☆ nie jest aktywny, licznik C☆☆☆ pracuje jako licznik dodający.

Objaśnienie użytych symboli na stronie 6-1

Urządzenie diagnostyczne	Operacja
D8200-D8234	zarezerwowane

6.15 Sterowanie szybkiego licznika (M8235-M8255 i D8235-D8255)

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie diagnostyczne	Operacja
M8235 – M8245 (☞)	gdy M8☆☆☆ jest aktywny, 1-fazowy szybki licznik C☆☆☆ pracuje jako licznik odejmujący. Gdy M8☆☆☆ nie jest aktywny, licznik C☆☆☆ pracuje jako licznik dodający. Dostępność liczników zależy od typu PLC.
M8246 – M8255 (✕) (☞)	gdy M8☆☆☆ jest aktywny, 2-fazowy szybki licznik C☆☆☆ pracuje jako licznik odejmujący. Gdy M8☆☆☆ nie jest aktywny, licznik C☆☆☆ pracuje jako licznik dodający. Dostępność liczników zależy od typu PLC.

Objaśnienie użytych symboli na stronie 6-1

Urządzenie diagnostyczne	Operacja
D8235-D8255	zarezerwowane

6.16 Tabela kodów błędów

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8061 Błąd sprzętowy PLC	0000	brak błędu	sprawdzić połączenie pomiędzy urządzeniem programującym a PLC
	6101	błąd RAM	
	6102	błąd operacji układu	
	6103	błąd magistrali I/O (M8069 = ON)	
	6104	błąd jednostki rozszerzającej 24V (M8069 = ON)	
	6105	błąd timera watchdoga	czas wykonania programu przekroczył wartość ustawioną w D8000

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8062 Błąd komunikacji PLC/HPP	0000	brak błędu	sprawdzić połączenie pomiędzy urządzeniem programującym a PLC
	6201	błąd parzystości/przepelnienia/ramki	
	6202	błąd znaku komunikacji	
	6203	błąd sumy kontrolnej	
	6204	błąd formatu danych	
	6205	błąd komendy	

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8063 Błąd komunikacji szeregowej	0000	brak błędu	sprawdzić zasilanie i połączenia Uwaga, dotycząca sieci 485: ponieważ te błędy nie są transmitowane przez sieć, muszą być monitorowane przez jednostkę Master
	6301	błąd parzystości/przepelnienia/ramki	
	6302	błąd znaku komunikacji	
	6303	błąd sumy kontrolnej	
	6304	błąd formatu danych	
	6305	błąd komendy Sieć 485 - otrzymana komenda jest inna niż GW, gdy numer stacji był FF	
	6306	błąd timera watchdoga	
	6312	błąd znaku połączenia równoległego	
	6313	błąd sumy kontrolnej połączenia równoległego	
	6314	błąd formatu danych połączenia równoległego	

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8064 Błąd parametru	0000	brak błędu	zatrzymać PLC, wybrać tryb parametrów, wprowadzić prawidłowe dane
	6401	błąd sumy kontrolnej programu	
	6402	błąd ustawienia pojemności pamięci	
	6403	błąd ustawienia obszaru urządzeń zatraskowych	
	6404	błąd ustawienia obszaru komentarza	
	6405	błąd ustawienia obszaru rejestrów plikowych	
	6406-6408	zarezerwowane	
	6409	inny błąd ustawień	

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8065 Błąd syntaktyczny	0000	brak błędu	Podczas programowania każda instrukcja jest sprawdzana w momencie jej wprowadzenia. W przypadku wykrycia błędu syntaktycznego należy wprowadzić instrukcję poprawnie
	6501	nieprawidłowa instrukcja/symbol urządzenia/kombinacja numeru urządzenia	
	6502	brak cewki timera lub licznika przed ustawieniem wartości	
	6503	1) Brak ustawienia wartości poprzedzającej cewkę timera lub licznika 2) za dużo argumentów instrukcji specjalnej	
	6504	1) dwukrotne użycie tej samej etykiety 2) dwukrotne użycie tego samego wejścia przerwania lub wejścia szybkiego licznika	
	6505	numer wejścia poza dozwolonym zakresem	
	6506	nieprawidłowa instrukcja specjalna	
	6507	nieprawidłowe przypisanie etykiety P	
	6508	nieprawidłowe przypisanie etykiety I	
	6509	inny błąd	
	6510	błąd zagnieżdżenia	
	6511	nałożenie się przypisania wejść przerwania i szybkiego licznika	

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8066 Błąd układu	0000	brak błędu	Błąd układu pojawia się w przypadku nieprawidłowej kombinacji instrukcji lub ich złej specyfikacji. Wybrać tryb programowania i skorygować wskazany błąd
	6601	użyto instrukcji LD i LDI 9 i więcej razy z rzędu	
	6602	1) brak instrukcji LD/LDI; nieautoryzowane użycie instrukcji LD/LDI, AND/ANI 2) następujące instrukcje nie są przyłączone do linii zasilającej: STL, RET, MCR, (P)ointer, (I)nterrupt, EI, DI, SRET, RET, FOR, NEXT, FEND i END 3) zagubiona MPP	
	6603	użyto instrukcji MPS więcej niż 12 razy	
	6604	nieautoryzowane użycie instrukcji MPS/MRD/MPP	
	6605	1) pojedyncze rozgałęzienie STL steruje więcej niż 9 równoległymi obwodami 2) instrukcje MC/MCR lub przerwanie zostało przypisane wewnątrz stanu STL 3) nie przypisano instrukcji RET lub przypisano ją poza stanem STL	
	6606	1) brak etykiety P lub I 2) brak instrukcji SRET/IRET 3) instrukcję SRET/IRET przypisano w obszarze głównego programu 4) STL/RET/MC lub MCR przypisano wewnątrz podprogramu lub programu obsługi przerwania	
	6607	1) nieautoryzowane użycie FOR-NEXT. Przypisano 6 lub więcej poziomów 2) wykorzystano następujące instrukcje wewnątrz pętli FOR-NEXT: STL/RET/MC/MCR/IRET/SRET/FEND lub END	
	6608	1) nieautoryzowana relacja MC/MCR 2) zagubione MCR N0 3) instrukcje SRET/IRET lub przerwanie zostały przypisane wewnątrz bloku MC/MCR	
	6609	inny błąd	

Kontynuacja na następnej stronie...

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8066 Błąd układu (tylko FX _{2N})	6610	użyto instrukcji LD i LDI 9 i więcej razy z rzędu	Błąd układu pojawia się w przypadku nieprawidłowej kombinacji instrukcji lub ich złej specyfikacji. Wybrać tryb programowania i skorygować wskazany błąd
	6611	liczba instrukcji LD/LDI jest mniejsza niż instrukcji ANB/ORB	
	6612	liczba instrukcji LD/LDI jest większa niż instrukcji ANB/ORB	
	6613	użyto instrukcji MPS więcej niż 12 razy	
	6614	brak instrukcji MPS	
	6615	brak instrukcji MPP	
	6616	nieautoryzowane użycie instrukcji MPS/MRD/MPP; możliwy brak cewki	
	6617	jedna z poniższych instrukcji nie jest podłączona do linii zasilającej: STL, RET, MCR, (P)ointer, (I)nterrupt, EI, DI, ARET, IRET, FOR, NEXT, FEND i END	
	6618	instrukcje STL/RET/MC lub MCR zaprogramowano wewnątrz podprogramu lub programu obsługi przerwania	
	6619	nieprawidłowa instrukcja wewnątrz pętli FOR-NEXT: STL/RET/MC/MCR/ I /IRET/SRET	
	6620	przekroczony zakres zagnieżdżeń pętli FOR-NEXT	
	6621	liczba instrukcji FOR nie odpowiada liczbie instrukcji NEXT	
	6622	brak instrukcji NEXT	
	6623	brak instrukcji MC	
	6624	brak instrukcji MCR	
	6625	pojedyncze rozgałęzienie STL steruje więcej niż 9 układami równoległymi	
	6626	nieprawidłowa instrukcja wewnątrz bloku STL-RET: MC/MCR/ I /IRET/SRET	
	6627	brak instrukcji RET	
	6628	instrukcje I/SRET/IRET zaprogramowane w obszarze programu głównego	
	6629	brak etykiety P lub I	
6630	brak instrukcji SRET lub IRET		
6631	instrukcja SRET zaprogramowana w nieprawidłowym miejscu		
6632	instrukcja IRET zaprogramowana w nieprawidłowym miejscu		

Urządzenie wykrywające błąd	Numer błędu	Znaczenie błędu	Działanie
D8067 Błąd instrukcji	0000	brak błędu	Ten błąd pojawia się podczas wykonywania instrukcji. Gdy pojawi się błąd instrukcji należy zatrzymać PLC (STOP), wprowadzić w tryb programowania i skorygować błąd. Uwaga: błędy instrukcji mogą się pojawić nawet gdy ich syntaktyka lub projekt układu jest prawidłowy, np. D500Z jest instrukcją prawidłową. Gdy jednak rejestr Z zawiera wartość 100, wystąpi próba dostępu do danej D600. To spowoduje błąd, ponieważ taka dana nie jest dostępna.
	6701	1) brak oznaczenia docelowego dla instrukcji CJ lub CALL 2) etykieta wskazuje blok poza instrukcją END 3) etykieta wewnątrz pętli FOR-NEXT lub podprogramu	
	6702	wystąpiło zagnieżdżenie 6 lub więcej wywołań instrukcji CALL	
	6703	wystąpiło zagnieżdżenie ponad 3 przerw	
	6704	wystąpiło zagnieżdżenie ponad 6 pętli FOR-NEXT	
	6705	niekompatybilne urządzenie jako argument instrukcji specjalnej	
	6706	urządzenie poza dozwolonym zakresem instrukcji specjalnej	
	6707	próba dostępu do rejestru plikowego poza zakresem zdefiniowanym przez użytkownika	
	6708	błąd instrukcji FROM/TO	
6709	inny błąd, np. brakująca instrukcja IRET/SRET, nieautoryzowana relacja FOR-NEXT		
D8067 Błąd operacji PID	6730	czas próbkowania T_S ($T_S < 0$ lub > 32767)	Wskazany parametr jest poza dozwolonym zakresem. Wykonanie jest zakończone
	6732	stała czasowa filtru wejściowego α ($\alpha < 0$ lub ≥ 101)	
	6733	stała proporcjonalna K_P ($K_P < 0$ lub > 32767)	
	6734	stała czasu całkowania T_I ($T_I < 0$ lub > 32767)	
	6735	stałą różniczkowania K_D ($K_D < 0$ lub ≥ 101)	
	6736	stała czasu różniczkowania T_D ($T_D < 0$ lub > 32767)	
	6740	czas próbkowania T_S jest mniejszy od czasu przebiegu programu	Ts jest ustawiony na czas przebiegu programu. Wykonanie jest kontynuowane
	6742	bieżąca wartość Δ jest poza zakresem	Dane są ustawiane do wartości najbliższych granicom zakresu. Dal wszystkich błędów, z wyjątkiem 6745 będzie to minimum -32768 lub maksimum 32767.
	6743	obliczony błąd ε jest poza zakresem	
	6744	wynik całkowania jest poza zakresem	
	6745	wartości różniczkowania są poza zakresem	
6746	wynik różniczkowania jest poza zakresem		
6747	całkowity wynik PID jest poza zakresem	Wykonanie jest kontynuowane.	

Notatki

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

7	Czasy wykonania i hierarchia instrukcji	7-1
7.1	Instrukcje podstawowe	7-1
7.2	Instrukcje specjalne	7-3
7.3	Hierarchiczne powiązania instrukcji podstawowych	7-12
7.4	Przetwarzanie wsadowe	7-14
7.5	Podsumowanie przydziału pamięci urządzeń	7-14
7.6	Ograniczenia w użyciu instrukcji	7-16
7.6.1	Instrukcje, które mogą być używane tylko raz w głównym obszarze programu	7-16
7.6.2	Instrukcje, które nie są odpowiednie do stosowania z jednostkami wejściowymi 110V AC	7-16

7. Czasy wykonania i hierarchia instrukcji

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

7.1 Instrukcje podstawowe

Mnemonic	Urządzenie	Liczba kroków	Czas wykonania instrukcji w μ s									
			FX ₀ , FX _{0S}		FX _{0N}		FX (< v. 3.07)		FX (> v.3.07) FX _{2C}		FX _{2N}	
			ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
LD	X, Y, M, S, T, C, M	1	3,4		3,4		0,74		0,48		0,08	
LDI			3,4		3,4							
AND			3,2		3,2							
ANI			3,2		3,2							
OR			3,2		3,2							
ORI	X, Y, M, S, T, C	1	brak									
LDP			brak									
LDF			brak									
ANP			brak									
ANF			brak									
ORP	brak											
ORF	brak	1	2,2		2,2		0,74		0,48		0,08	
ANB			2,2		2,2							
ORB			2,0		2,0							
MPS			2,0		2,0							
MRD			2,0		2,0							
MPP	2,0		2,0									
INV	brak											
MC	poziom zagnieżdż. M, Y	3	17	18,2	19,2	20,4	42,8	47,8	27	30	24,8	27,5
MCR	poziom zagnieżdż.	2	6,0		6,2		40,4		19		20,8	
NOP	brak	1	1,6		1,6		0,74		0,48		0,08	
END			410		470		960		700		508	
STL	S (patrz uwaga 1)	1	4,2+8n		6,4 + 6,8n		39.1+21.1n		25+13,5n		27,3+12,6n	
RET	brak		8,0		12,4		40,5		20		21,6	

Ciąg dalszy na następnej stronie...

Mnemonik	Urządzenie	Liczba kroków	Czas wykonania instrukcji w μ s										
			FX ₀ , FX _{0S}		FX _{0N}		FX (< v. 3.07)		FX (> v.3.07) FX _{2C}		FX _{2N}		
			ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
OUT	Y,M	1	3,2		3,2		0,74		0,48		0,08		
	S	2	7,0	7,2	7,0	7,2	50,0	48,1	26	24	24,4	24,3	
	M spec.	2	7,8	7,4	8,2	7,8	38,1	38,8	28	20	0,16		
	T-K	3	21,8	19,4	25,2	21,0	72,4	52,6	45	34	42,3	37,4	
	T-D	3	23,4	21,0	27,2	23,0	80,0	52,6	53	34	42,2	37,2	
	C-K (16-bit)	3	14,6		17,8	15,6	67,9	40,3	42	25	25,5	24,9	
	C-D (16-bit)	3	16,2		19,8	17,6	75,5	40,3	47	25	25,3	25,0	
	C-K (32-bit)	5	12,5	6,0	16,0	8,6	82,3	40,3	51	25	25,3	24,9	
C-D (32-bit)	5	13,9	6,0	18,0	8,6	89,9	40,3	55	25	25,2	24,9		
SET	Y,M	1	3,6	2,0	3,6	2,0	0,74		0,48		0,08		
	S	2	6,8	2,6	7,0	2,8	39,0	25,5	24	16	23,7	17,2	
	S gdy używane w kroku STL (patrz uwaga 1)		brak				45,2+	14,2n	25,5	28+	9n	16	27,3+
	M spec.	2	7,4	2,4	7,8	2,6	41,9	28,5	26	18	0,16		
RST	Y,M	1	3,4	1,8	3,6	1,8	0,74		0,48		0,08		
	S	2	6,0	2,6	6,2	2,8	40,5	25,5	23	16	23,1	17,3	
	M spec.	2	7,4	2,4	7,8	2,6	41,8	28,9	26	18	0,16		
	T, C	2	20,8	18,0	22,4	19,6	50,1	38,3	31	24	27	25	
	D, V, Z, D spec.	3	10,0	2,8	9,2	3,0	35,5	25,5	22	16	21,9	17,1	
PLS	Y,M	2	19,4		21,8		41,9	41,5	27	26	0,32		
PLF	Y,M	2					42,7	40,6	26	25	0,32		
P	0 do 63	1	1,6		1,6		0,74		0,48		0,08		
I	I□□□	1											



Uwaga1:

„n” w formułach do obliczania czasu wykonania instrukcji odnosi się do liczby instrukcji STL w bieżącym rozgałęzieniu równoległym. Wartość „n” zawiera się w zakresie 1 do 8.

7.2 Instrukcje specjalne

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Mnemonic	16/32 bit	Czas wykonania instrukcji w μ s														
		FX ₀ , FX _{0S}		FX _{0N}		FX (< v.3.07)			FX (> v.3.07.), FX _{2C}				FX _{2N}			
		ON	OFF	ON	OFF	ON	OFF	P	ON	OFF	2 FNC ON	P	ON	OFF	P	
00 CJ	16	19,4	9,6	20,0	10,0	46,6	27,4	✓	29	8,8	-	✓	29,0	6,4	✓	
01 CALL	16	brak				49,5	27,4	✓	31	8,8	-	✓	32,2	6,4	✓	
02 SRET	16	brak				34,0			21		-		21,2			
03 IRET	*1	11,2		11,6		36,7			33		-		18,1			
04 EI	*1	6,4		7,8		62,6			34	8,8	-		55,8			
05 DI	*1					37,7			17	8,8	-	18,5				
06 FEND	*1	410		470		960			700		-		508			
07 WDT	16	9,2	5,6	9,8	7,0	35,9	25,1	✓	23	8,8	-	✓	26,3	6,4	✓	
08 FOR	*1	29,0		29,8		39,9			25		-		27,6			
09 NEXT	*1	12,4		12,4		29,1			19		-		5,2			
10 CMP	16	122,6	22,0	112,2	22,6	161,8	33,3	✓	49	8,8	-	✓	87,6	6,4	✓	
	32	129,2	30,4	118,6	31,6	189,0	39,9		57	12,2	-		91,9	6,4		
11 ZCP	16	140,0	25,0	128,4	25,8	186,9	33,3	✓	62	8,8	-	✓	103,2	6,4	✓	
	32	197,8	36,4	137,8	38,0	220,8	39,9		73	12,2	-		108,9	6,4		
12 MOV	16	46,2	18,0	47,2	18,4	78,4	33,3	✓	35	8,8	-	✓	1,52	1,52	✓	
	32	52,6	23,4	53,8	24,2	98,4	39,3		43	12,2	-		1,84	1,84		
13 SMOV	16	brak				302,9	33,3	✓	170	8,8	-	✓	155,2	6,4	✓	
14 CML	16	brak				74,0	33,3	✓	51	8,8	-	✓	51,4	6,4	✓	
	32	brak				95,9	33,3		64	12,2	-		55,9	6,4		
15 BMOV *2	16	brak		103,2 + 18,2n	20,8	180,5 + 17,1n	33,3	✓	118+ 10,6n	8,8	-	✓	97,0+ 1,7n	6,4	✓	
16 FMOV *2	16	brak				107,6 + 5,3n	33,3	✓	73 + 3,3n	8,8	-	✓	69,1+ 2,8n	6,4	✓	
	32	brak				brak			87 + 4,5n	12,2	-		73,2+ 5,2n	6,4		
17 XCHG	16	brak				90,3	33,3	✓	58	8,8	-	✓	57,2	6,4	✓	
	32	brak				113,8	39,8		72	12,2	-		64,0	6,4		
18 BCD	16	63,6		64,4	18,4	130,9	33,3	✓	82	8,8	306	✓	37,9	6,4	✓	
	32	100,2		101,4	24,2	342,0	39,9		218	12,2	-		57,6	6,4		
19 BIN	16	64,4		66,2	18,4	135,4	33,3	✓	85	8,8	157	✓	32,4	6,4	✓	
	32	113,4		114,8	24,2	314,3	39,9		203	12,2	-		44,5	6,4		

Uwagi dotyczące „*” znajdują się na końcu sekcji...

Mnemonic	16/32 bit	Czas wykonania instrukcji w μ s																																	
		FX ₀ , FX _{0S}		FX _{0N}		FX (< v.3.07)			FX (> v.3.07.), FX _{2C}				FX _{2N}																						
		ON	OFF	ON	OFF	ON	OFF	P	ON	OFF	2 FNC ON	P	ON	OFF	P																				
20 ADD	16	69,4		70,8	21,6	115,5	33,3	✓	51	8,8	-	✓	27,6	6,4	✓																				
	32	81,2		82,8	31,8	144,5	39,9		63	12,2	224		28,9	6,4																					
21 SUB	16	69,8		71,6	21,6	116,6	33,3	✓	52	8,8	-	✓	27,6	6,4	✓																				
	32	81,4		83,0	31,6	146,5	39,9		65	12,2	232		28,9	6,4																					
22 MUL	16	89,4		91,0	21,6	133,4	33,3	✓	54	8,8	-	✓	25,2	6,4	✓																				
	32	104,6		106,4	31,2	185,0	39,9		81	12,2	162		31,4	6,4																					
23 DIV	16	119,2		120,8	21,6	139,5	33,3	✓	56	8,8	-	✓	32,0	6,4	✓																				
	32	230,0		232,4	31,0	804,8	39,9		451	12,2	197		36,4	6,4																					
24 INC	16	28,4		29,0	14,8	55,3	33,3	✓	26	8,8	-	✓	18,8	6,4	✓																				
	32	33,4		34,2	17,4	65,4	34,4		29	12,2	-		20,2	6,4																					
25 DEC	16	28,4		29,0	14,8	55,4	33,3	✓	26	8,8	-	✓	18,9	6,4	✓																				
	32	33,6		34,4	17,4	65,1	34,4		29	12,2	-		20,0	6,4																					
26 WAND	16	64,2		65,5	21,6	108,0	33,3	✓	67	8,8	-	✓	23,4	6,4	✓																				
	32	73,0		74,6	30,6	135,4	39,9		83	12,2	-		24,8	6,4																					
27 WOR	16	64,2		65,6	21,6	107,9	33,3	✓	67	8,8	-	✓	23,5	6,4	✓																				
	32	73,0		74,6	30,6	135,5	39,9		83	12,2	-		24,7	6,4																					
28 WXOR	16	64,2		65,6	21,6	106,5	33,3	✓	67	8,8	-	✓	23,5	6,4	✓																				
	32	73,0		74,6	30,5	133,9	39,9		82	12,2	-		25,0	6,4																					
29 NEG	16	brak				55,1	33,3	✓	34	8,8	-	✓	35,3	6,4	✓																				
	32					65,5	34,4		41	12,2	-		38,4	6,4																					
30 ROR *3	16					brak				91,9+	33,3	✓	57+	8,8	-	✓	61,7	6,4	✓																
	32									113,8+	39,9		70+	12,2	-		65,3	6,4																	
31 ROL *3	16									brak				91,9+	33,3	✓	57+	8,8	-	✓	61,7	6,4	✓												
	32													113,8+	39,9		71+	12,2	-		65,2	6,4													
32 RCR *3	16													brak				99,0+	33,3	✓	61+	8,8	-	✓	66,3+	6,4	✓								
	32																	120,8+	39,9		75+	12,2	-		69,7+	6,4									
33 RCL *3	16																	brak				99,0+	33,3	✓	62+	8,8	-	✓	65,8+	6,4	✓				
	32																					120,8+	39,9		75+	12,2	-		69,5+	6,4					
34 SFTR *4	16																					145,2+	24,6	156,4+	25,4	180,8+	33,3	✓	172+	8,8	-	✓	107+	6,4	✓
35 SFTL *4	16																					150,6+	24,2	162,4+	25,0	180,8+	33,3	✓	172+	8,8	-	✓	105+	6,4	✓

Uwagi dotyczące „*” znajdują się na końcu sekcji...

Mnemonic	16/32 bit	Czas wykonania instrukcji w μs															
		FX ₀ , FX _{0S}		FX _{0N}		FX (< v.3.07)			FX (> v.3.07.), FX _{2C}				FX _{2N}				
		ON	OFF	ON	OFF	ON	OFF	P	ON	OFF	2 FNC ON	P	ON	OFF	P		
36 WSFR ★2	16	brak						218,6 + 18,0n	33,3	✓	147 +11n	8,8	-	✓	126+ 11,7n	6,4	✓
37 WSFL ★2	16	brak						218,6 + 18,0n	33,3	✓	147 +11n	8,8	-	✓	125+ 11,8n	6,4	✓
38 SFWR ★5	16	brak						138,1	33,3	✓	87	8,8	-	✓	83,9	6,4	✓
39 SFRD ★5	16	brak						143,1 + 6,8n	33,3	✓	87	8,8	-	✓	80,2	6,4	✓
40 ZRST ★6	16D	57,2+ 1,6n	12,8	65,0+ 1,6n	13,2	161,3 + 3,2n	39,9	✓	121+ 2n	8,8	-	-	✓	77+ 1,7n	6,4	✓	
	16S	127,0 + 2,9n		131,5 + 2,9n		161,3 + 16,5n			121+ 10,5n					83+ 11,1n			
	16C	64,4+ 3,3n		70,7+ 3,3n		161,3 + 13,5n			121+ 8,7n					89,2+ 9,4n			
	16T	65,2+ 3,3n		71,5+ 3,3n													
	16M	127,0 + 0,08n		131,5 + 3,5n		161,3 + 13,5n			121+ 8,7n					89,2+ 9,4n			
	16Y	127,0 + 3,5n		131,5 + 3,5n													
41 DECO	16	881,4	20,6	932,0	21,4	114,8	28,8	✓	72	8,8	-	✓	76,0	6,4	✓		
42 ENCO	16	618,3	20,6	692,4	21,4	125,6	28,8	✓	79	8,8	-	✓	81,8	6,4	✓		
43 SUM	16	brak						133,5	33,3	✓	84	8,8	-	✓	72,8	6,4	✓
	32	brak						196,6	39,9		123	12,2	-		94,6	6,4	
44 BON	16	brak						168,9	33,3	✓	98	8,8	-	✓	78,2	6,4	✓
	32	brak						177,6	39,9		112	12,2	-		82,3	6,4	
45 MEAN ★7	16	brak						133,4 + 12,2n	33,3	✓	84+ 7,7n	8,8	-	✓	83,8+ 3,4n	6,4	✓
	32	brak						brak			105+ 9,8n	12,1	-		90,9+ 6,7n	6,4	
46 ANS	16	brak						192,6	165,6		120	110	-	✓	100,8	96,2	
47 ANR	16	brak						86,5	25,5	✓	54	8,8	-	✓	37,7	6,4	✓
48 SQR	16	brak									208	8,8	-	✓	150,2	6,4	✓
	32	brak									220	12,2	344			154,8	
49 FLT	16	brak									98	8,8	-	✓	66,8	6,4	✓
	32	brak									114	12,2	-			66,8	
50 REF ★8	16	53,6	12,8	65,4+ 3,1n	13,4	145,3 + 3,6n	33,3	✓	62+ 3,2n	8,8	-	✓	99,6+ 0,6n	6,4	✓		

Uwagi dotyczące „★” znajdują się na końcu sekcji...

Mnemonic	16/32 bit	Czas wykonania instrukcji w μ s													
		FX ₀ , FX _{0S}		FX _{0N}		FX (< v.3.07)			FX (> v.3.07.), FX _{2C}				FX _{2N}		
		ON	OFF	ON	OFF	ON	OFF	P	ON	OFF	² FNC ON	P	ON	OFF	P
51 REFE ★9	16	brak				56,0+ 4,9n	33,3	✓	35+ 3,5n	8,8	-	✓	65,3+ 1,7n	6,4	✓
52 MTR	16	brak				87,3	39,3		53	26	-		39,1	23,6	
53 HSCS ★10	32	75,6	6,6	82,8	7,8	175,0	39,3		115	12,2	-		87,8	6,4	
54 HSCS ★10	32														
55 HSZ ★10	32	brak				240,3	39,3		142	12,2	-		100,6	6,4	
56 SPD ★1	★1	brak				164,4	163,0		102	101	-		80,2	80,2	
57 PLSY	16 32	189,4	10,0	212,4	21,4	154,5	173,6		101	136	-		85,0	73,3	
				223,4	31,4				115	136	-		86,6	75,8	
58 PWM	16	42,5	7,8	44,2	18,6	139,8	171,0		86	101	-		70,4	73,3	
59 PLSR	16 32	brak											122,6	87,5	
		brak											125,6	90,5	
60 IST	16	766,0	322,4	212,4	21,4	272,9	33,3		153	8,8	-		114,3	6,4	
61 SER ★14	16 32	brak							147+ 12,5n	29	-	✓	129,2 +8,6n	22,9	✓
		brak							168+ 17,4n	29	-		147+ 9,0n	22,9	
62 ABSD ★11	16 32	brak				141,4 + 61,4n	33,3		91+ 35n		-		91,8+ 20,2n	6,4	
		brak				brak			110+ 43n		-		97,5+ 21,5n	6,4	
63 INCD	16	brak				208,8	39,9		130	26	-		110,5	19,5	
64 TTMR	16	brak				81,3	69,6		48	43	-		54,9	44,9	
65 STMR	16	brak				176,6	167,8		106	104	-		84,4	84,4	
66 ALT	16	61,0	9,8	62,8	10,0	105,6	33,3	✓	66	8,8	-	✓	50,1	6,4	✓
67 RAMP	16	248,6	82,6			181,8	134,5		113	83	-		98,1	81,6	
68 ROTC	16	brak				232,5	209,1		144	130	-		118,4	107,2	
69 SORT ★15	16	brak							62+ 32,3m 1	23	-		50,5	19,5	

Uwagi dotyczące „★” znajdują się na końcu sekcji...

Mnemonik	16/32 bit	Czas wykonania instrukcji w μ s													
		FX ₀ , FX _{0S}		FX _{0N}		FX (< v.3.07)			FX (> v.3.07.), FX _{2C}				FX _{2N}		
		ON	OFF	ON	OFF	ON	OFF	P	ON	OFF	2 FNC ON	P	ON	OFF	P
70 TKY	16 32	brak				245,7	33,3		153	23	-		97,2	22,2	
		brak				229,1	39,9		145	23	-		98,7	22,2	
71 HKY	16 32	brak				318,8	39,9		189	29	-		92,2	27,4	
		brak				338,0	45,5		205	29	-		65,0	6,4	
72 DSW	16-1 zb.	brak				205,8	39,3		130	30	-		92,2	27,4	
	16-2 zb.	brak				208,1			brak	30	-		brak	brak	
73 SEGD	16	brak				142,1	33,3	✓	87	8,8	-	✓	65,0	6,4	✓
74 SEGL	16-1 zb.	brak				209,7	33,3		127	30	-		105,9	26,5	
	16-2 zb.	brak				246,9			148		-		brak	brak	
75 ARWS	16	brak				285,0	163,0		169	8,8	-		134,4	22,1	
76 ASC	16	brak				130,9	33,3		104	8,8	-		49,5	6,4	
77 PR	16 pr.	brak				207,1	112,6		127	58	-		114,8	88,5	
	16 rdy	brak				112,1			68		-		88,0		
78 FROM *12	16 32	brak		120+ 400n	26	170+ 406n	45,0	✓	120+ 325n	14	-	✓	97+ 487n	6,4	✓
		brak		120+ 800n	26	200+ 800n			140+ 640n		-		99+ 962n	6,4	
79 TO *12	16 32	brak		120+ 480n	38	151+ 480n	45,0	✓	106+ 384n	14	-	✓	94+ 557n	6,4	✓
		brak		120+ 950n	38	200+ 936n			140+ 749n		-		96+ 1009n	6,4	
80 RS	16	brak	125,2	20,5	brak			132	20	-		117,6	18,0		
81 PRUN *13	16 32	brak				137,1 +53,5 n	33,3	✓	91+3 2n	8,8	-	✓	65,6+ 17,0n	6,4	✓
		brak				154,5 +49,3 n			104+ 34n	12,2	-		67,0+ 17,7n	6,4	
82 ASCI	16	brak		115+ 9,7n	22,3	brak			94+1 2n	8,8	-	✓	88,2+ 10,8n	6,4	✓
83 HEX	16	brak		115+ 22,9n	22,1	brak			95+2 3n	8,8	-	✓	89,7+ 20,0n	6,4	✓
84 CCD	16	brak		115+ 11,7n	24,4	brak			96+8 n	8,8	-	✓	90,5+ 4,8n	6,4	✓
85 VRRD	16	brak				308,1	33,3	✓	209	21	-	✓	209,7	27,3	✓
86 VRSC	16	brak				319,1	33,3	✓	205	21	-	✓	202,4	27,3	✓
87	16/ 32	brak													

Uwagi dotyczące „*” znajdują się na końcu sekcji...

Mnemonic	16/32 bit	Czas wykonania instrukcji w μ s													
		FX ₀ , FX _{0S}		FX _{0N}		FX (< v.3.07)			FX (> v.3.07.), FX _{2C}				FX _{2N}		
		ON	OFF	ON	OFF	ON	OFF	P	ON	OFF	² FNC ON	P	ON	OFF	P
88 PID	16	brak						407	109	-		155,0	89,0		
89	16 32	brak													
90 MNET	16	brak			643,9	25,5	✓	brak							
91 ANRD	16				1,137	33,3	✓								
92 ANWR	16				1,387	470,9	✓								
93 RMST	16				948,8	950,0		691	692	-		brak			
94 RMWR	16				2,214	33,3	✓	1,612	8,8	-	✓				
	32				4,235	39,9		3,127	12,2	-					
95 RMRD	16				1,684	33,3	✓	1,254	8,8	-	✓				
	32				3,168	39,9		2,414	12,2	-					
96 RMMN	16				1,589	33,3	✓	1,195	8,8	-	✓				
97 BLK	16				672,4	669,3	✓	brak							
98 MCDE	16	740,3	33,3	✓											
99	16 32	brak													
110 ECMP	32	brak									104,4	6,4	✓		
111 EZCP	32	brak									124,5	6,4	✓		
118 EBCD	32	brak									106,9	6,4	✓		
119 EBIN	32										81,3	6,4	✓		
120 EADD	32										117,4	6,4	✓		
121 ESUB	32										117,4	6,4	✓		
122 EMUL	32										96,4	6,4	✓		
123 EDIV	32										100,4	6,4	✓		

Uwagi dotyczące „*” znajdują się na końcu sekcji...

Mnemonic	16/32 bit	Czas wykonania instrukcji w μ s													
		FX ₀ , FX _{0S}		FX _{0N}		FX (< v.3.07)			FX (> v.3.07.), FX _{2C}				FX _{2N}		
		ON	OFF	ON	OFF	ON	OFF	P	ON	OFF	2 FNC ON	P	ON	OFF	P
127 ESQR	32	brak											152,1	6,4	✓
128		brak											brak		
129 INT	16	brak											67,5	6,4	✓
	32	brak											70,4	6,4	
130 SIN	32	brak											199,5	6,4	✓
131 COS	32	brak											262,5	6,4	✓
132 TAN	32	brak											425,3	6,4	✓
147 SWAP	16	brak											36,1	6,4	✓
	32	brak											41,2	6,4	
160 TCMP	16	brak											134,2	6,4	✓
161 TZCP	16	brak											140,2	6,4	✓
162 TADD	16	brak											118,8	6,4	✓
163 TSUB	16	brak											109,4	6,4	✓
164	16	brak											brak		
165	16	brak											brak		
166 TRD	16	brak											46,2	6,4	✓
167 TWR	16	brak											112,0	6,4	✓
168	16	brak											brak		
169	16	brak											brak		
170 GRY	16	brak											102,5	6,4	✓
	32	brak											107,1	6,4	
171 GBIN	16	brak											103,4	6,4	✓
	32	brak											107,5	6,4	
224-230 LD□	16	brak											1,52		
	32	brak											1,84		
232-238 AND□	16	brak											1,52		
	32	brak											1,84		
240-246 OR□	16	brak											1,52		
	32	brak											1,84		

Uwagi dotyczące „*” znajdują się na końcu sekcji...

- ★1:
 - te instrukcje nie wymagają urządzeń stykowych, takich jak LD, AND, OR, itp.
- ★2:
 - „n” oznacza liczbę dostępnych rejestrów. „n” nie może być większe 512
- ★3:
 - „n” oznacza liczbę dostępnych urządzeń bitowych. „n” nie może być większe niż przyjęty tryb operacyjny, tzn. jeśli jest to tryb 32-bitowy, to „n” nie może być większe od 32
- ★4:
 - „n” oznacza liczbę dostępnych urządzeń bitowych. W przypadku użycia sterownika FX „n” nie może być większe niż 1024, w przypadku sterowników FX₀ i FX_{0N} „n” nie może być większe niż 512
- ★5:
 - „n” oznacza liczbę dostępnych urządzeń. „n” może przyjmować wartości z zakresu od 2 do 512
- ★6:
 - „n” oznacza zakres urządzeń do wyzerowania. Typ zerowanego urządzenia jest określony przez literę urządzenia w nawiasie w kolumnie „16/32 bit”.
- ★7:
 - „n” oznacza liczbę urządzeń. „n” może przyjmować wartości z zakresu od 1 do 64.
- ★8:
 - „n” oznacza zakres odświeżanych urządzeń. Wartość „n” jest zawsze określana w jednostkach po 8, tzn. 8, 16, 24, .., 128. Maksymalny dostępny zakres zależy od liczby dostępnych wejść/wyjść, tzn. w FX₀ -16, w FX - 128
- ★9:
 - „n” oznacza ustawienie stałej czasowej filtrów wejściowych. Może przyjmować wartości od 0 do 60 ms.
- ★10:
 - Istnieją ograniczenia kombinacji użycia tych instrukcji. Dla FX₀ i FX_{0N} nie powinno być więcej niż 4 jednocześnie aktywne instrukcje. Dla FX może być 6 jednocześnie aktywnych instrukcji.,
- ★11:
 - „n” oznacza liczbę wyjść. Nie może być większe niż 64
- ★12:
 - „n” oznacza liczbę słów odczytanych lub zapisanych z/do bloków FROM/TO
- ★13:
 - „n” oznacza liczbę słów odczytanych lub zapisanych, gdy dwa sterowniki FX pracują w połączeniu równoległym.

*14:

- „n” oznacza liczbę elementów na stosie, dla operacji 16-bitowych „n” wynosi maksymalnie 256. Dla operacji 32-bitowych „n” wynosi maksymalnie 128.

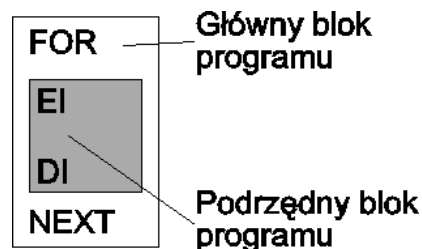
*15:

- „m1” oznacza liczbę elementów w tablicy danych. m1 może przyjmować wartości z zakresu od 1 do 32. Instrukcja SORT dla całkowitego przetworzenia tablicy wykona się m1 razy.

7.3 Hierarchiczne powiązania instrukcji podstawowych

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Następująca tabela pokazuje „powiązanie zawierające”. Oznacza to, że główny blok programu zawiera blok podrzędny ze wszystkimi ograniczeniami bloku głównego, np.



Główny blok programu	Podrzędny blok programu							
	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STR-RET	P-SRET	I-RET	FEND-END
MC-MCR	✓ - 8 poziomów zagnieżdżeń	✓	✓	✓	✓	x-(6608)	x-(6608)	x-(6608)
CJ-P	✓	✓	✓	✓	✓	●	●	x-(6701)
EI-DI	✓	✓	✓	✓	✓	✓	✓	✓①
FOR-NEXT	x-(6607)	✓	✓	✓ - 5 poziomów zagnieżdżeń	x-(6607)	x-(6607)	x-(6607)	x-(6607)
STL-RET	x-(6605)	●	✓	✓ - (wewnątrz 1 kroku STL)	✓	x-(6605)	x-(6605)	x-(6605)
P-SRET	x-(6606)	✓	✓	✓	x-(6606)	x-(6606)	x-(6606)	x-(6709)
I-RET	x-(6606)	✓	✓	✓	x-(6606)	x-(6606)	x-(6606)	x-(6606)
FEND-END	✓	✓	✓	✓	●	✓	✓	✓②
0-FEND	✓	✓	✓	✓	✓	x-(6606)	x-(6606)	✓②
0-END (bez FEND)	✓	✓	✓	✓	✓	x-(6606)	x-(6606)	✓②



✓: Kombinacja instrukcji dopuszczalna – ograniczenia opisane w odpowiedniej uwadze

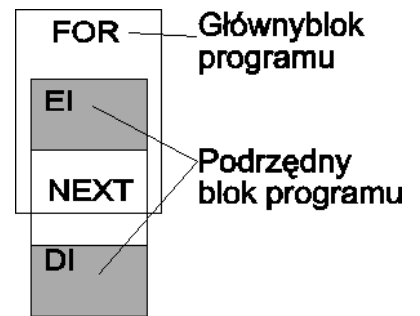
x: Kombinacja instrukcji niedozwolona – liczba w nawiasach oznacza kod błędu

●: Kombinacja niezalecana, nawet gdy nie występuje błąd instrukcji

Kombinacja instrukcji z „powiązaniem zawierającym” jest dopuszczalna. Należy jednak pamiętać o następujących wyjątkach:

- 1) Instrukcje MC-MCR i STL-RET nie mogą być użyte wewnątrz pętli FOR-NEXT i podprogramów P-SRET i I-RET.
- 2) Przebieg programu nie może być przerywany przez użycie którejkolwiek z metod wewnątrz konstrukcji MC-MCR, FOR-NEXT, P-SRET i I-RET, tzn. użycie przerwania, instrukcji IRET, SRET, FEND i END jest niedozwolone.

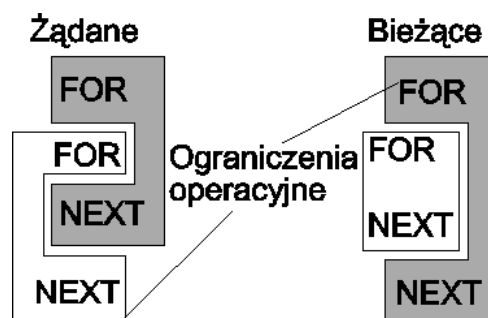
Następująca tabela pokazuje „powiązanie pokrywające”. Oznacza ono, że konstrukcja bloku podrzędnego rozpoczyna się z wszystkimi ograniczeniami bloku głównego, ale kończy się poza blokiem głównym, tzn.



Główny blok programu	Podrzędny blok programu							
	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STR-RET	P-SRET	I-IRET	FEND-END
MC-MCR	●	●	✓	x-(6607)	x-(6605)	x-(6606)	x-(6606)	x-(6608)
CJ-P	●	●	✓	●	●	●	●	✓
EI-DI	✓	✓	✓	✓	✓	✓	✓	✓
FOR-NEXT	x-(6607)	●	✓	✓ ^③	x-(6701)	x-(6607)	x-(6607)	x-(6607)
STL-RET	x-(6605)	●	✓	x-(6607)	✓	x-(6606)	x-(6606)	x-(6605)
P-SRET	x-(6608)	●	✓	x-(6607)	x-(6605)	x-(6606)	x-(6606)	x-(6709)
I-IRET	x-(6606)	●	✓	x-(6607)	x-(6606)	x-(6606)	x-(6606)	x-(6606)
FEND-END	x-(6608)	x-(6701)	✓ ^①	x-(6607)	x-(6605)	x-(6709)	x-(6709)	✓ ^②
0-FEND	x-(6608)	✓	✓	x-(6607)	x-(6605)	x-(6709)	x-(6606)	✓ ^②
0-END (bez FEND)	x-(6608)	x-(6701)	✓ ^①	x-(6607)	x-(6605)	x-(6709)	x-(6706)	✓ ^②



- ① Wprowadza stan jak w przypadku braku instrukcji DI. Nie ma błędu.
 ② Pierwsze wystąpienie instrukcji FEND lub END ma priorytet.
 ③ Ta sekwencja nie będzie wykonana, np.

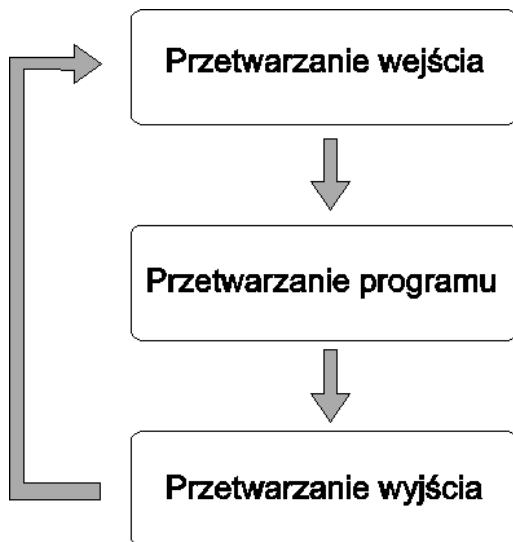


7.4 Przetwarzanie wsadowe

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Jest to system, używany przez wszystkie jednostki z rodziny sterowników FX. Podstawowa koncepcja opiera się na trzech stanach przebiegu programu. Innymi słowy, jeżeli program jest przetwarzany od początku do końca, występuje następująca sekwencja zdarzeń:

Przetwarzanie wejść:



Stany wszystkich wejść są wczytywane do pamięci tymczasowej, zwanej też pamięcią obrazu. PLC jest wówczas gotów do przetwarzania programu...

Przetwarzanie programu:

Wszystkie zaktualizowane wejścia są sprawdzane podczas przetwarzania programu. Jeżeli nowy status wejścia zmieni status sterowanego wyjścia, jest to odnotowane w pamięci obrazu do momentu...

Przetwarzanie wyjść:

Następuje fizyczna aktualizacja wyjść, zgodnie z nowym, bieżącym statusem, tzn. przekaźniki są przełączane ze stanu ON lub OFF, zgodnie z wymaganiami. Sterownik rozpoczyna nowy przebieg programu...

System nazywany jest „przetwarzaniem wsadowym” ponieważ wszystkie wejścia, instrukcje programu i w końcu wyjścia są przetwarzane wsadowo.

7.5. Podsumowanie przydziału pamięci urządzeń

Z punktu widzenia użytkownika można wyróżnić trzy główne obszary:

a) Pamięć Programu

Ten obszar pamięci przechowuje dane dotyczące parametrów, sekwencji programu, stałych K i H, informacji o etykietach P i I, poziomów zagnieżdżeń, zawartości rejestrów plikowych i komentarzy do programu.

- Pamięć ta jest albo podtrzymywana bateryjnie albo wykorzystuje EEPROM. Dane w tej pamięci są zachowywane po wyłączeniu zasilania PLC. Czas i gwarancja przechowania zależy od stanu baterii lub użytej pamięci EEPROM.

b) Pamięć Danych

Ten obszar pamięci przechowuje, zgodnie ze swoją nazwą, wszystkie dane związane z: rejestrami danych, rejestrami indeksowymi, wartościami timera, wartościami timera z podtrzymaniem (o ile są dostępne) oraz bieżącymi wartościami licznika.

- Wszystkie te urządzenia, które działają jako zatraskowe (włączając w to timery z pamięcią) są przechowywane w sposób podobny do opisanego w punkcie a).
- Rejestry indeksowe i rejestry specjalne (D8000 do D8255) działają w specyficzny sposób w następujących okolicznościach:

Okoliczności	Reakcja
zasilanie PLC jest wyłączone	wszystkie dane są zerowane
zasilanie PLC jest włączone	Niektóre urządzenia są ustawiany w swój stan domyślny – patrz rozdział 6
PLC jest przełączony ze stanu STOP do RUN PLC jest przełączony ze stanu RUN do STOP	Niektóre urządzenia są ustawiany w swój stan domyślny – patrz rozdział 6

- Wszystkie inne urządzenia, takie jak rejestry bez podtrzymania, timery i liczniki zachowują się w następujący sposób:

Okoliczności	Reakcja
zasilanie PLC jest wyłączone zasilanie PLC jest włączone	wszystkie dane są zerowane
PLC jest przełączony ze stanu STOP do RUN PLC jest przełączony ze stanu RUN do STOP	Bez zmian zerowane (chyba, że M8033 jest aktywny)

c) Pamięć Bitowa

Ten obszar pamięci przechowuje status styków wszystkich wejść, wyjść, przekaźników, znaczników, timerów i liczników.

- Wszystkie te urządzenia, które działają jako zatraskowe (włączając w to timery z pamięcią) są przechowywane w sposób podobny do opisanego w punkcie a)
- Znaczniki (M8000 do M8255) działają w sposób podobny do opisanego w punkcie b)
- Wszystkie inne urządzenia podlegają takim samym zmianom jak bieżące wartości rejestrów, timerów i liczników (zobacz ostatni punkt i tabelę w sekcji b).

Podsumowanie

Typ pamięci	Zasilanie		Status PLC	
	OFF	OFF⇒ON	STOP⇒RUN	RUN⇒STOP
wszystkie urządzenia podtrzymywane bateryjnie	bez zmian			
urządzenia M i D (8000 do 8255) oraz rejestry indeksowe (V i Z)	zerowane	ustawiane w stan domyślny	bez zmian	
wszystkie inne urządzenia	zerowane		bez zmiany	zerowane bez zmiany jeśli M8033 jest ustawiony

7.6 Ograniczenia w użyciu Instrukcji

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

7.6.1 Instrukcje, które mogą być używane tylko raz w głównym obszarze programu

Następujące instrukcje mogą być użyte tylko raz w obszarze głównym programu. Należy sprawdzić szczegółowe wyjaśnienie dotyczące instrukcji lub tabele wykonania instrukcji.



- Instrukcje, które mogą być użyte tylko jeden raz:

MTR FNC 52
PLSY FNC 57
PWM FNC 58
PLSR FNC 59

IST FNC 60
SORT FNC 61
ABSD FNC 62
INCD FNC 63
ROTC FNC 68

TKY FNC 70
HKY FNC 71
DWS FNC 72
SEGL FNC 74
ARWS FNC 75



Tylko jedna z instrukcji PLSY FNC 57 lub PLSR FNC 59 może być zaprogramowana w danym momencie. Obie instrukcje nie mogą jednocześnie wystąpić w tym samym aktywnym programie.

7.6.2 Instrukcje, które nie są odpowiednie do Stosowania z jednostkami wejściowymi 110V AC

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Podczas użycia jednostek wejściowych 110V AC, niektóre operacje, funkcje i instrukcje nie są polecane z powodu długich czasów ładowania/rozładowania urządzeń wejściowych 110 V.



- Operacje programowe, które nie są polecane, to:
 - programy obsługi przerw
 - liczniki szybkie

- Instrukcje, które nie są polecane, to:

REFF FNC 51
MTR FNC 52
SPD FNC 56

ROTC FNC 68
TKY FNC 70
HKY FNC 71

DSW FNC 72
ARWS FNC 75



Uwaga: pomimo tego, że wybrane jednostki FX_{0S} mają wejścia 110V AC, instrukcje opisane powyżej nie są zaimplementowane w CPU jednostki i stąd nie mogą być użyte.

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

8	Specyfikacja urządzeń sterownika	8-1
8.1	Specyfikacja urządzeń FX ₀ i FX _{0S}	8-1
8.2	Specyfikacja urządzeń FX _{0N}	8-2
8.3	Specyfikacja urządzeń FX (wersja CPU 2.0 do 3.06)	8-4
8.4	Specyfikacja urządzeń FX (wersja CPU od 3.07 i wszystkie FX _{2C})	8-6
8.5	Specyfikacja urządzeń FX _{2N}	8-8

8. Specyfikacja urządzeń sterownika

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

8.1 Specyfikacja urządzeń FX₀ i FX_{0S}

Element		Specyfikacja	Uwagi
Metoda wykonywania programu		Cykliczne wykonywanie zapamiętanego programu	
Metoda sterowania wyjścia/wejścia		Przetwarzanie wsadowe (gdy wykonywana jest instrukcja END)	Dostępne jest odświeżanie instrukcji wejścia/wyjścia
Czas wykonywania operacji		Instrukcje podstawowe: od 1,6 do 3,6 μs Instrukcje specjalne: od kilkudziesięciu s do 100 μs	
Język programowania		Język symboliki przekaźnikowej + schemat drabinkowy	Schemat drabinkowy może być użyty do tworzenia programów SFC
Pojemność programu		800 kroków	wbudowana pamięć EEPROM
Liczba instrukcji		Instrukcje podstawowe: 20 Instrukcje drabinkowe: 2 Instrukcje specjalne: 35	Dostępne jest 50 instrukcji specjalnych we wszystkich odmianach
Konfiguracja wejścia/wyjścia		Wejścia/wyjścia ustawiane przez MPU	
Przekaźniki pomocnicze (znaczniki M)	Ogólne	512 szt.	od M0 do M511
	Zatraskowe	16 szt. (podzbiór)	od M496 do M511
	Specjalne	56 szt.	z zakresu M8000 do M8255
Znaczniki stanu (S)	Ogólne	64 szt.	od S0 do S63
	Inicjujące	10 szt. (podzbiór)	od S0 do S9
Timery (T)	100 ms	Zakres: od 0 do 3.276,7 s 56 szt.	od T0 do T55
	10 ms	Zakres: od 0 do 327,67 s 24 szt.	od T32 do T55, gdy M8028 jest w stanie ON
Liczniki (C)	Ogólne	Zakres od 1 do 32.767 16 szt.	od C0 do C13 typ: 16-bitowy licznik dodający
	Zatraskowe	2 szt. (podzbiór)	od C14 do C15 typ: 16-bitowy licznik dodający
Liczniki szybkie (C)	1-fazowe	Zakres: 0d -2.147.483.648 do 2.147.483.647 zliczeń	od C235 do C238 (C235 nie jest zatraskowy), 4 szt.
	1-fazowe z wejściami start/stop	FX ₀ : Wybór do czterech 1-fazowych liczników z połączoną częstotliwością zliczania 5 kHz lub mniej. Alternatywny wybór jednego 2-fazowego lub A/B-fazowego licznika z częstotliwością zliczania 2 kHz lub mniej	C241 (zatraskowy), C242 i C244 (zatraskowy), 3 szt.
	2-fazowe	FX _{0S} : Gdy używanych jest wiele 1-fazowych liczników, suma częstotliwości musi być równa lub mniejsza od 14 kHz. Tylko jeden 2-fazowy szybki licznik może być użyty na raz. Gdy używane są 2-fazowe liczniki, maksymalna szybkość zliczania musi być równa lub mniejsza od 14 kHz, obliczana jako (szybkość licznika x liczba zboczy) + 1	C246, C247, C249 (wszystkie zatraskowe), 3 szt.
	A/B-fazowe		C251, C252 i C254 (wszystkie zatraskowe), 3 szt.

ciąg dalszy na następnej stronie...

Element		Specyfikacja	Uwagi
Rejestry danych (D)	Ogólne	32 szt.	od D0 do D31 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Zatraskowe	2 szt. (podzbiór)	od D30 do D31 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Ustawiane zewnętrznie	Zakres: od 0 do 255 1 szt.	D8013 Dana jest wprowadzana pośrednio poprzez zewnętrzny potencjometr
	Specjalne	27 szt. (wliczając D8013)	od D8000 do D8255 typ: 16-bitowy rejestr danych
	Indeksowe	2 szt.	V i Z typ: 16-bitowy rejestr danych
Wskaźniki (P)	używane z instrukcją CALL	64 szt.	od P0 do P63
	używane w przerwanich	4 szt.	od I00□ do I30□ wyzwalanie narastające □ = 1 wyzwalanie opadające □ = 0
Poziomy zagnieżdzeń		8 szt. do wykorzystania z MC i MCR	od N0 do N7
Stałe	Dziesiętne K	16-bitowe: od -32.768 do 32.767 32-bitowe: od -2.147.483.648 do 2.147.483.647	
	Szesnastkowe H	16-bitowe: 0000 do FFFF 32-bitowe: 00000000 do FFFFFFFF	

8.2 Specyfikacja urządzeń FX_{0N}

Element		Specyfikacja	Uwagi
Metoda wykonywania programu		Cykliczne wykonywanie zapamiętanego programu	
Metoda sterowania wyjścia/wejścia		Przetwarzanie wsadowe (gdy wykonywana jest instrukcja END)	Dostępne jest odświeżanie instrukcji wejścia/wyjścia
Czas wykonywania operacji		Instrukcje podstawowe: od 1,6 do 3,6 μs Instrukcje specjalne: od kilkudziesięciu s do 100 μs	
Język programowania		Język symboliki przekaźnikowej + schemat drabinkowy	Schemat drabinkowy może być użyty do tworzenia programów SFC
Pojemność programu		2000 kroków	wbudowana pamięć EEPROM
Liczba instrukcji		Instrukcje podstawowe: 20 Instrukcje drabinkowe: 2 Instrukcje specjalne: 42	Dostępne jest 59 instrukcji specjalnych we wszystkich odmianach
Konfiguracja wejścia/wyjścia		Maksimum 128 punktów konfiguracyjnych sprzętowych wejść/wyjść, zależnych od użytkownika (Maksymalna liczba adresowanych wejść: 128, wyjść: 128)	
Przekaźniki pomocnicze (znaczniki M)	Ogólne	512 szt.	od M0 do M511
	Zatraskowe	128 szt. (podzbiór)	od M384 do M511
	Specjalne	65 szt.	z zakresu M8000 do M8255

ciąg dalszy na następnej stronie...

Element		Specyfikacja	Uwagi
Znaczniki stanu (S)	Ogólne	128 szt.	od S0 do S127
	Inicjujące	10 szt. (podzbiór)	od S0 do S9
Timery (T)	100 ms	Zakres: od 0 do 3.276,7 s 63 szt.	od T0 do T62
	10 ms	Zakres: od 0 do 327,67 s 31 szt.	od T32 do T62, gdy M8028 jest w stanie ON
	1 ms	Zakres: od 0 do 32,767 s 1 szt.	T63
Liczniki (C)	Ogólne	Zakres od 1 do 32.767 32 szt.	od C0 do C31 typ: 16-bitowy licznik dodający
	Zatraskowe	16 szt. (podzbiór)	od C16 do C31 typ: 16-bitowy licznik dodający
Liczniki szybkie (C)	1-fazowe	Zakres: od -2.147.483.648 do 2.147.483.647 zliczeń Wybór do czterech 1-fazowych liczników z połączoną częstotliwością zliczania 5 kHz lub mniej. Alternatywny wybór jednego 2-fazowego lub A/B-fazowego licznika z częstotliwością zliczania 2 kHz lub mniej Wszystkie liczniki są zatraskowe	od C235 do C238, 4 szt.
	1-fazowe z wejściami start/stop		C241, C242 i C244, 3 szt.
	2-fazowe		C246, C247, C249, 3 szt.
	A/B-fazowe		C251, C252 i C254, 3 szt.
Rejestry danych (D)	Ogólne	256 szt.	od D0 do D255 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Zatraskowe	128 szt. (podzbiór)	od D128 do D255 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Plikowe	1500 szt.	od D1000 do D2499 ustawianych przez parametr w 3 blokach po 500 kroków programu typ: 16-bitowy rejestr danych
	Ustawiane zewnętrznie	Zakres: od 0 do 255 2 szt.	Dane są wprowadzane pośrednio poprzez zewnętrzne potencjometry D8030* i D8031 (*gdy nie ma RTC, używany jest D8013)
	Specjalne	42 szt. (wliczając D8013, D8030 i D8031)	od D8000 do D8255 typ: 16-bitowy rejestr danych
	Indeksowe	2 szt.	V i Z typ: 16-bitowy rejestr danych
Wskaźniki (P)	używane z instrukcją CALL	64 szt.	od P0 do P63
	używane w przerwaniach	4 szt.	od I00□ do I30□ wyzwalanie narastające □ = 1 wyzwalanie opadające □ = 0
Poziomy zagnieżdzeń		8 szt. do wykorzystania z MC i MCR	od N0 do N7
Stałe	Dziesiętne K	16-bitowe: od -32.768 do 32.767 32-bitowe: od -2.147.483.648 do 2.147.483.647	
	Szesnastkowe H	16-bitowe: 0000 do FFFF 32-bitowe: 00000000 do FFFFFFFF	

8.3 Specyfikacja urządzeń FX (wersja CPU 2.0 do 3.06)

Element		Specyfikacja	Uwagi
Metoda wykonywania programu		Cykliczne wykonywanie zapamiętanego programu	
Metoda sterowania wyjścia/wejścia		Przetwarzanie wsadowe (gdy wykonywana jest instrukcja END)	Dostępne jest odświeżanie instrukcji wejścia/wyjścia
Czas wykonywania operacji		Instrukcje podstawowe: 0,74 μ s Instrukcje specjalne: od kilkudziesięciu s do 100 μ s	
Język programowania		Język symboliki przekaźnikowej + schemat drabinkowy	Schemat drabinkowy może być użyty do tworzenia programów SFC
Pojemność programu		2000 kroków	rozszerzalna do 8000 kroków przy zastosowaniu kasety pamięci
Liczba instrukcji		Instrukcje podstawowe: 20 Instrukcje drabinkowe: 2 Instrukcje specjalne: 87	Dostępne jest 119 instrukcji specjalnych we wszystkich odmianach
Konfiguracja wejścia/wyjścia		Maksimum 256 punktów konfiguracyjnych sprzętowych wejść/wyjść, zależnych od użytkownika (Maksymalna liczba adresowanych wejść: 128, wyjść: 128)	
Przełączniki pomocnicze (znaczniki M)	Ogólne	1024 szt.	od M0 do M1023
	Zatraskowe	524 szt. (podzbiór)	od M500 do M1023
	Specjalne	256 szt.	z zakresu M8000 do M8255
Znaczniki stanu (S)	Ogólne	1000 szt.	od S0 do S999
	Zatraskowe	500 szt. (podzbiór)	od S500 do S999
	Inicjujące	10 szt. (podzbiór)	od S0 do S9
	Wskaźnikowe	100 szt.	od S900 do S999
Timery (T)	100 ms	Zakres: od 0 do 3.276,7 s 200 szt.	od T0 do T199
	10 ms	Zakres: od 0 do 327,67 s 46 szt.	od T200 do T245
	1 ms z podtrzymaniem	Zakres: od 0 do 32,767 s 4 szt.	od T246 do T249
	100 ms z podtrzymaniem	Zakres: od 0 do 3.276,7 s 6 szt.	od T250 do T255
Liczniki (C)	Ogólne 16-bitowe	Zakres od 1 do 32.767 200 szt.	od C0 do C199 typ: 16-bitowy licznik dodający
	Zatraskowe 16-bitowe	100 szt. (podzbiór)	od C100 do C199 typ: 16-bitowy licznik dodający
	Ogólne 32-bitowe	Zakres od -2.147.483.648 do 2.147.483.647 35 szt.	od C200 do C234 typ: 32-bitowy licznik dwukierunkowy
	Zatraskowe 32-bitowe	15 szt.	od C219 do C234 typ: 16-bitowy licznik dwukierunkowy

ciąg dalszy na następnej stronie...

Element		Specyfikacja	Uwagi
Liczniki szybkie (C)	1-fazowe	Zakres: od -2.147.483.648 do 2.147.483.647 zliczeń Reguła ogólna: wybierać kombinacje liczników z połączoną częstotliwością zliczania 20 kHz lub mniej. Wszystkie liczniki są zatraskowe	od C235 do C240, 6 szt.
	1-fazowe z wejściami start/stop		od C241 do C245, 5 szt.
	2-fazowe		od C246 do C250, 5 szt.
	A/B-fazowe		od C251 do C255, 5 szt.
Rejestry danych (D)	Ogólne	512 szt.	od D0 do D511 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Zatraskowe	312 szt. (podzbiór)	od D200 do D511 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Plikowe	2000 szt.	od D1000 do D2999 ustawianych przez parametr w 4 blokach po 500 kroków programu typ: 16-bitowy rejestr danych
	Specjalne	256 szt.	od D8000 do D8255 typ: 16-bitowy rejestr danych
	Indeksowe	2 szt.	V i Z typ: 16-bitowy rejestr danych
Wskaźniki (P)	używane z instrukcją CALL	64 szt.	od P0 do P63
	używane w przerwaniach	6 szt. i 3 timery	od I00□ do I50□ i od I6☆☆ do I8☆☆ wyzwalanie narastające □ = 1 wyzwalanie opadające □ = 0 (☆☆- czas w ms)
Poziomy zagnieżdzeń		8 szt. do wykorzystania z MC i MCR	od N0 do N7
Stałe	Dziesiętne K	16-bitowe: od -32.768 do 32.767 32-bitowe: od -2.147.483.648 do 2.147.483.647	
	Szesnastkowe H	16-bitowe: 0000 do FFFF 32-bitowe: 00000000 do FFFFFFFF	

8.4 Specyfikacja urządzeń FX (wersja CPU od 3.07 i wszystkie FX_{2C})

Element		Specyfikacja	Uwagi
Metoda wykonywania programu		Cykliczne wykonywanie zapamiętanego programu	
Metoda sterowania wyjścia/wejścia		Przetwarzanie wsadowe (gdy wykonywana jest instrukcja END)	Dostępne jest odświeżanie instrukcji wejścia/wyjścia
Czas wykonywania operacji		Instrukcje podstawowe: 0,48 μs Instrukcje specjalne: od kilkudziesięciu s do 100 μs	
Język programowania		Język symboliki przekaźnikowej + schemat drabinkowy	Schemat drabinkowy może być użyty do tworzenia programów SFC
Pojemność programu		2000 kroków	rozszerzalna do 8000 kroków przy zastosowaniu kasety pamięci
Liczba instrukcji		Instrukcje podstawowe: 20 Instrukcje drabinkowe: 2 Instrukcje specjalne: 96	Dostępne jest 119 instrukcji specjalnych we wszystkich odmianach
Konfiguracja wejścia/wyjścia		Maksimum 255 punktów konfiguracyjnych sprzętowych wejść/wyjść, zależnych od użytkownika (Maksymalna liczba adresowanych wejść: 255, wyjść: 255)	
Przekaźniki pomocnicze (znaczniki M)	Ogólne	1536 szt.	od M0 do M1535
	Zatrzaskowe	1024 szt. (podzbiór)	od M500 do M1535
	Specjalne	256 szt.	z zakresu M8000 do M8255
Znaczniki stanu (S)	Ogólne	1000 szt.	od S0 do S999
	Zatrzaskowe	500 szt. (podzbiór)	od S500 do S999
	Inicjujące	10 szt. (podzbiór)	od S0 do S9
	Wskaźnikowe	100 szt.	od S900 do S999
Timery (T)	100 ms	Zakres: od 0 do 3.276,7 s 200 szt.	od T0 do T199
	10 ms	Zakres: od 0 do 327,67 s 46 szt.	od T200 do T245
	1 ms z podtrzymaniem	Zakres: od 0 do 32,767 s 4 szt.	od T246 do T249
	100 ms z podtrzymaniem	Zakres: od 0 do 3.276,7 s 6 szt.	od T250 do T255
Liczniki (C)	Ogólne 16-bitowe	Zakres od 1 do 32.767 200 szt.	od C0 do C199 typ: 16-bitowy licznik dodający
	Zatrzaskowe 16-bitowe	100 szt. (podzbiór)	od C100 do C199 typ: 16-bitowy licznik dodający
	Ogólne 32-bitowe	Zakres od -2.147.483.648 do 2.147.483.647 35 szt.	od C200 do C234 typ: 32-bitowy licznik dwukierunkowy
	Zatrzaskowe 32-bitowe	15 szt. (podzbiór)	od C219 do C234 typ: 16-bitowy licznik dwukierunkowy

ciąg dalszy na następnej stronie...

Element		Specyfikacja	Uwagi
Liczniki szybkie (C)	1-fazowe	Zakres: od -2.147.483.648 do 2.147.483.647 zliczeń	od C235 do C240, 6 szt.
	1-fazowe z wejściami start/stop	Reguła ogólna: wybierać kombinacje liczników z połączoną częstotliwością zliczania 20 kHz lub mniej.	od C241 do C245, 5 szt.
	2-fazowe		od C246 do C250, 5 szt.
	A/B-fazowe	Wszystkie liczniki są zatraskowe	od C251 do C255, 5 szt.
Rejestry danych (D)	Ogólne	1000 szt.	od D0 do D999 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Zatraskowe	800 szt. (podzbiór)	od D200 do D999 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Plikowe	2000 szt.	od D1000 do D2999 ustawianych przez parametr w 4 blokach po 500 kroków programu typ: 16-bitowy rejestr danych
	Plikowe RAM	2000 szt.	od D6000 do D7999 aktywnych, gdy M8047 jest w stanie ON typ: 16-bitowy rejestr danych
	Specjalne	256 szt.	od D8000 do D8255 typ: 16-bitowy rejestr danych
	Indeksowe	2 szt.	V i Z typ: 16-bitowy rejestr danych
Wskaźniki (P)	używane z instrukcją CALL	128 szt.	od P0 do P127
	używane w przerwaniach	6 szt., 3 timery i 6 liczników	od I00□ do I50□ i od I6☆☆ do I8☆☆, od I010 do I060 wyzwalanie narastające □ = 1 wyzwalanie opadające □ = 0 (☆☆- czas w ms)
Poziomy zagnieżdzeń		8 szt. do wykorzystania z MC i MCR	od N0 do N7
Stałe	Dziesiętne K	16-bitowe: od -32.768 do 32.767 32-bitowe: od -2.147.483.648 do 2.147.483.647	
	Szesnastkowe H	16-bitowe: 0000 do FFFF 32-bitowe: 00000000 do FFFFFFFF	
	Zmienno-przecinkowe	32-bitowe: $0, \pm 1, 175 \times 10^{-38}, \pm 3,403 \times 10^{38}$ (nie do wprowadzenia bezpośrednio)	

8.5 Specyfikacja urządzeń FX_{2N}

Element		Specyfikacja	Uwagi
Metoda wykonywania programu		Cykliczne wykonywanie zapamiętanego programu	
Metoda sterowania wyjścia/wejścia		Przetwarzanie wsadowe (gdy wykonywana jest instrukcja END)	Dostępne jest odświeżanie instrukcji wejścia/wyjścia
Czas wykonywania operacji		Instrukcje podstawowe: 0,08 μs Instrukcje specjalne: od kilkudziesięciu s do 100 μs	
Język programowania		Język symboliki przekaźnikowej + schemat drabinkowy	Schemat drabinkowy może być użyty do tworzenia programów SFC
Pojemność programu		8000 kroków	rozszerzalna do 16000 kroków przy zastosowaniu kasety pamięci
Liczba instrukcji		Instrukcje podstawowe: 20 Instrukcje drabinkowe: 2 Instrukcje specjalne: 125	Dostępne jest ??? instrukcji specjalnych we wszystkich odmianach
Konfiguracja wejścia/wyjścia		Maksimum 255 punktów konfiguracyjnych sprzętowych wejść/wyjść, zależnych od użytkownika (Maksymalna liczba adresowanych wejść: 255, wyjść: 255)	
Przełączniki pomocnicze (znaczniki M)	Ogólne	3072 szt.	od M0 do M3071
	Zatrząskowe	2572 szt. (podzbiór)	od M500 do M3071
	Specjalne	256 szt.	z zakresu M8000 do M8255
Znaczniki stanu (S)	Ogólne	1000 szt.	od S0 do S999
	Zatrząskowe	500 szt. (podzbiór)	od S500 do S999
	Inicjujące	10 szt. (podzbiór)	od S0 do S9
	Wskaźnikowe	100 szt.	od S900 do S999
Timery (T)	100 ms	Zakres: od 0 do 3.276,7 s 200 szt.	od T0 do T199
	10 ms	Zakres: od 0 do 327,67 s 46 szt.	od T200 do T245
	1 ms z podtrzymaniem	Zakres: od 0 do 32,767 s 4 szt.	od T246 do T249
	100 ms z podtrzymaniem	Zakres: od 0 do 3.276,7 s 6 szt.	od T250 do T255
Liczniki (C)	Ogólne 16-bitowe	Zakres od 1 do 32.767 200 szt.	od C0 do C199 typ: 16-bitowy licznik dodający
	Zatrząskowe 16-bitowe	100 szt. (podzbiór)	od C100 do C199 typ: 16-bitowy licznik dodający
	Ogólne 32-bitowe	Zakres od -2.147.483.648 do 2.147.483.647 35 szt.	od C200 do C234 typ: 32-bitowy licznik dwukierunkowy
	Zatrząskowe 32-bitowe	15 szt. (podzbiór)	od C219 do C234 typ: 16-bitowy licznik dwukierunkowy

ciąg dalszy na następnej stronie...

Element		Specyfikacja	Uwagi
Liczniki szybkie (C)	1-fazowe	Zakres: od -2.147.483.648 do 2.147.483.647 zliczeń Reguła ogólna: wybierać kombinacje liczników z połączoną częstotliwością zliczania 20 kHz lub mniej. Wszystkie liczniki są zatraskowe	od C235 do C240, 6 szt.
	1-fazowe z wejściami start/stop		od C241 do C245, 5 szt.
	2-fazowe		od C246 do C250, 5 szt.
	A/B-fazowe		od C251 do C255, 5 szt.
Rejestry danych (D)	Ogólne	8000 szt.	od D0 do D7999 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Zatraskowe	7800 szt. (podzbiór)	od D200 do D7999 typ: 16-bitowy rejestr danych, para dla urządzeń 32-bitowych
	Plikowe	7000 szt. (podzbiór)	od D1000 do D7999 ustawianych przez parametr w 14 blokach po 500 kroków programu typ: 16-bitowy rejestr danych
	Specjalne	256 szt.	od D8000 do D8255 typ: 16-bitowy rejestr danych
	Indeksowe	16 szt.	od V0 do V7 i od Z0 do Z7 typ: 16-bitowy rejestr danych
Wskaźniki (P)	używane z instrukcją CALL	128 szt.	od P0 do P127
	używane w przerwaniach	6 szt., 3 timery i 6 liczników	od I00□ do I50□ i od I6☆☆ do I8☆☆, od I010 do I060 wyzwalanie narastające □ = 1 wyzwalanie opadające □ = 0 (☆☆- czas w ms)
Poziomy zagnieżdżeń		8 szt. do wykorzystania z MC i MCR	od N0 do N7
Stałe	Dziesiętne K	16-bitowe: od -32.768 do 32.767 32-bitowe: od -2.147.483.648 do 2.147.483.647	
	Szesnastkowe H	16-bitowe: 0000 do FFFF 32-bitowe: 00000000 do FFFFFFFF	
	Zmiennie-przecinkowe	32-bitowe: $0, \pm 1,175 \times 10^{-38}, \pm 3,403 \times 10^{38}$ (nie do wprowadzenia bezpośrednio)	

Notatki

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

9	System adresowania urządzeń	9-1
9.1	Adresowanie modułów rozszerzeń	9-1
9.2	Użycie modułu FX2-24EI do obsługi specjalnych bloków funkcjonalnych serii F	9-2
9.2.1	Użycie modułu FX2-24EI do obsługi F-16NP/NT	9-3
9.2.2	Użycie modułu FX2-24EI do obsługi F2-6A	9-4
9.2.3	Użycie modułu FX2-24EI do obsługi F2-32RM	9-4
9.2.4	Użycie modułu FX2-24EI do obsługi F2-30GM	9-5
9.3	Adaptory połączeń równoległych FX2-40AP/W	9-6
9.4	Funkcje zegara czasu rzeczywistego	9-7
9.4.1	Ustawianie zegara czasu rzeczywistego	9-8

9. System adresowania urządzeń



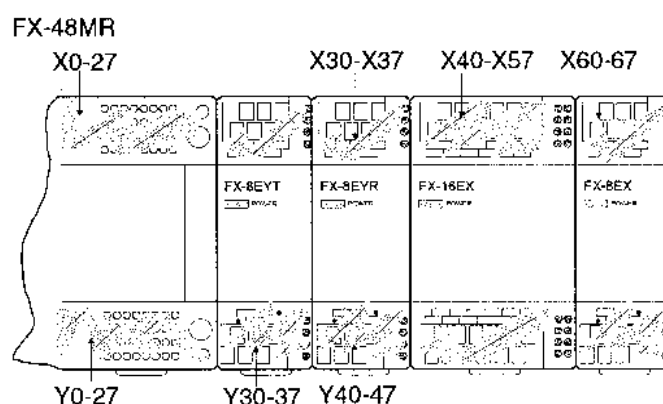
9.1. Adresowanie modułów rozszerzeń

Większość sterowników z rodziny FX posiada możliwość przyłączenia dodatkowych, dyskretnych modułów wejścia/wyjścia i/lub specjalnych modułów funkcjonalnych. Aby móc korzystać z tych dodatkowych jednostek, użytkownik musi adresować każdy blok niezależnie.

Adresowanie dodatkowych dyskretnych wejść/wyjść

Ten typ wejść/wyjść jest standardowym modułem wejścia i wyjścia. Każdy dodawany do systemu moduł (blok) rozszerzeniowy lub jednostka rozszerzeniowa (z zasilaniem) przyjmują kolejne, dostępne adresy. Stąd, jednostki najbliższe jednostce bazowej będą miały najniższe numery wejść/wyjść lub adresy. Numery wejść/wyjść są zawsze obliczane ósemkowo, tzn. 0 do 7, od 10 do 17 itd. W programie użytkownika te dodatkowe adresy są używane jak te z modułu bazowego. Dyskretne wejścia/wyjścia mogą być dodawane tak długo, jak długo spełnione są zasady konfiguracji systemu dla każdego typu sterownika. Tę informację można znaleźć w odpowiednim podręczniku.

Dla łatwego używania i identyfikacji każda dodatkowa jednostka wejścia/wyjścia powinna być etykietowana odpowiednimi numerami wejść/wyjść.



Ostrzeżenie przy korzystaniu z modułów FX-8ER i FX-24MR w systemie FX

Kiedy używane są ww. wymienione moduły musi być udostępnione 8 dodatkowych adresów (4 wejścia i 4 wyjścia). Dzieje się tak dlatego, ponieważ oba moduły dzielą na pół bloki 8 wejść i 8 wyjść, aby otrzymać fizyczną konfigurację 4 wejść/4 wyjść. Stąd blok FX-8ER zajmuje 8 adresów wejściowych i 8 adresów wyjściowych pomimo tego, że fizycznie posiada on tylko 4 wejścia i 4 wyjścia.

Adresowanie specjalnych bloków funkcjonalnych

Specjalnym blokiem funkcjonalnym są przydzielone logiczne adresy (numery) bloków od 0 do 7. Adresy te są używane przez instrukcje FROM/TO do bezpośredniego dostępu do każdego z nich, jako niezależnego, specjalnego modułu funkcjonalnego. Im mniejszy jest numer bloku, tym bliżej jest on jednostki bazowej. Bloki mogą być dodawane, jak długo spełnione są zasady konfiguracji systemu dla każdego typu sterownika. Uwagi dotyczące konfiguracji, znajdują się w odpowiednim podręczniku, omawiającym dany typ sterownika.

9.2. Użycie modułu FX2-24EI do obsługi specjalnych bloków funkcjonalnych serii F

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Moduł FX2-24EI umożliwia jednostce bazowej FX bezpośrednie przyłączenie jednego z następujących bloków funkcjonalnych serii F:

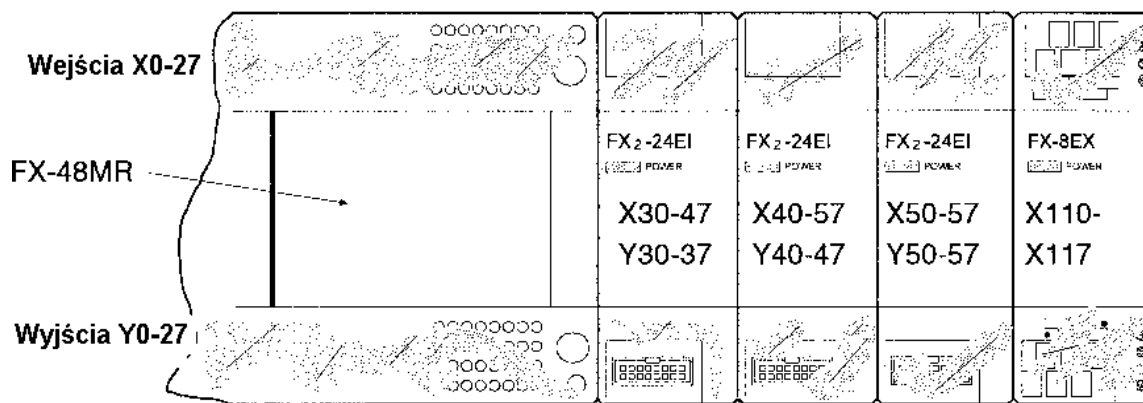
- F-16NT/NP, interfejs minisieci Melsec
- F2-6A, moduł 4 wejść i 2 wyjść analogowych
- F2-32RM, programowalny sekwencer CAM
- E2-30GM, jednostka pozycjonująca sterowana ciągiem impulsów



Jeden moduł 24EI może sterować jednym funkcjonalnym modułem serii F.

Moduły 24EI są dodawane do systemu FX w taki sam sposób, jak dodatkowe, dyskretne moduły wejścia/wyjścia. Każdy moduł 24EI zajmuje 16 adresów wejściowych i 8 adresów wyjściowych, tak jak pokazano na schemacie poniżej.

Doświadczeni użytkownicy mogą zauważyć, że przykład pokazany na tym schemacie, po zliczeniu wejść/wyjść, będzie wymagał zbyt dużej mocy z jednostki CPU (w tym przypadku FX-48MR).



Jednak w tym przypadku jest inaczej. FX2-24EI jest w rzeczywistości modułem komunikacyjnym. Nie steruje on bezpośrednio żadnymi dyskretnymi wejściami/wyjściami. Rozpatrując bilans mocy można założyć, że zajmuje on tylko 8 standardowych adresów wejścia/wyjścia. Oznacza to, że konfiguracja pokazana na schemacie zajmuje 32 adresy (trzy FX2-24EI i jedna FX-8EX) wejść/wyjść w bilansie mocy, natomiast w rzeczywistości zajmuje 80 adresowalnych wejść/wyjść.

Połączenie zacisków uziemiających podczas używania specjalnych bloków funkcjonalnych



Podczas używania specjalnych bloków funkcjonalnych ich zaciski uziemiające powinny być dołączone do zacisku [SG] jednostki CPU. W przypadku używania modułu F-16NP/NT jej zacisk [SG] nie powinien być podłączony do zacisku [SG] jednostki FX. Funkcje tych zacisków są zupełnie różne i stąd **NIE SĄ** kompatybilne.

9.2.1. Użycie modułu FX2-24EI do obsługi F-16NP/NT.

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Numery programowe wejść/wyjść (adresy) modułu F-16NP/NT wyznaczone są na podstawie pozycji lokacji przyporządkowanego mu modułu FX2-24EI w zestawie sterownika. Rysunek poniżej pokazuje, jak zmiana pozycji użytego modułu FX2-24EI zmienia adresy używane przez moduł F-16NP/NT. Patrz przykład 1A do 1C.

Szczegóły programowe, instalacji i okablowania modułu F-16NP/NT znajdują się w odpowiednim podręczniku. Ten podręcznik pokazuje przykłady użycia F-16NP/NT i numerację wejść/wyjść używanych do jego adresowania, które są ustalane przez sterowniki serii F. Te adresy wejść/wyjść mogą się zmieniać wg opisu zawartego w poprzednim paragrafie; tzn. urządzenia X/Y sterownika FX są przyporządkowane modułowi FX2-24EI wg jego miejsca w zestawie.

Przykład:

Następująca tabela pokazuje zależność pomiędzy systemami FX a F2.

Ustawienia FX	Ustawienia FX ₂	Komentarz
X54	X414	dane wejściowe prawidłowe
X55	X415	dane wejściowe nieprawidłowe
X56	X416	dane wyjściowe prawidłowe
X57	X417	dane wyjściowe nieprawidłowe
X60 do X67	X420 do X427	wejście do PLC
Y40 do Y47	Y440 do Y447	wyjście z PLC

Użyty system FX jest podobny do pokazanego na rysunku przykładu 1C. System F2 używa drugiego portu rozszerzającego, tzn. portu X400 sterownika serii F.

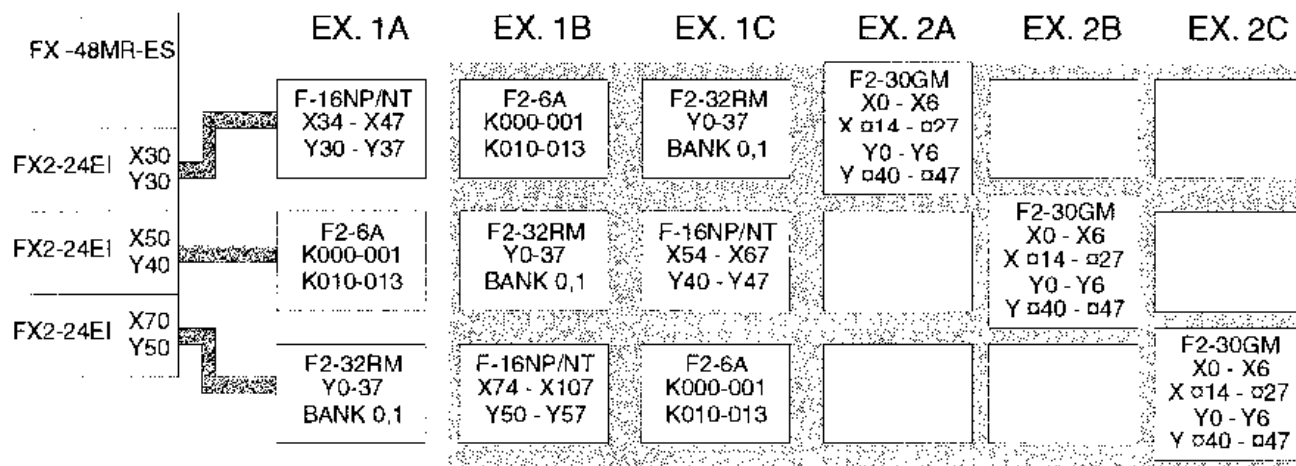


Zastosowanie instrukcji specjalnych FX.

(Brak zastosowania w jednostkach MPU FX2c lub FX z numerem wersji CPU większym od 3.06)

- MNET (FNC 90) - używana do odczytywania i wpisywania informacji sieci Net Mini.

Przykładowa konfiguracja FX2-24EI i bloków funkcjonalnych serii F.



9.2.2. Użycie modułu FX2-24EI do obsługi F2-6A.

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	-----------	--------------------	------------------

Numery programowe wejść/wyjść (adresy) modułu F2-6A wyznaczone są na podstawie pozycji lokacji przyporządkowanego mu modułu FX2-24EI w zestawie sterownika. Jednak na numery kanałów wejść/wyjść nie ma wpływu adres programowy. Kanał wejść/wyjść zawsze pozostanie jako K000 do K001 i K010 do K013. Rysunek na poprzedniej stronie pokazuje, jak przesuwanie pozycji użytej jednostki FX2-24EI zmienia adresy programowe użytego modułu F2-6A, lecz nie zmienia numeru kanału.

Zobacz przykłady 1A-1C

Szczegóły dotyczące użycia F2-6A znajdują się w podręczniku użytkownika tego modułu. Podczas czytania podręcznika należy pamiętać, że był on napisany tylko dla systemu F2 i stąd wszystkie adresy wejścia/wyjścia są określone jak w systemie F2.

Identyfikacja kanałów F2-6A

- analogowe kanały wyjściowe - K000 i K001 (2 kanały)
- analogowe kanały wejściowe - K010-K013 (4 kanały)



Zastosowanie instrukcji specjalnych FX.

(Brak zastosowania w jednostkach MPU FX2c lub FX z numerem wersji CPU większym od 3.06)

- ANRD (FNC 91) - używana do wczytywania danych analogowych do FX
- ANWR (FNC 92) - używana do przepisywania danych z FX na wyjścia F2-6A

9.2.3. Użycie modułu FX2-24EI do obsługi F2-32RM.

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	-----------	--------------------	------------------

Numery programowe wejść/wyjść (adresy) modułu F2-32RM wyznaczone są na podstawie pozycji lokacji przyporządkowanego mu modułu FX2-24EI w zestawie sterownika. Jednak numery wejść/wyjść jednostki 32RM zawsze pozostają takie same, ponieważ nie ma na nie wpływu adres programowy FX. Numery wejść/wyjść modułu 32RM pozostaną jako Y0 do Y37 i Bank 0, 1. Rysunek na poprzedniej stronie pokazuje, jak zmiana pozycji użytego modułu FX2-24EI zmienia adresy programowe, ale nie zmienia numerów wejść/wyjść, używanych przez F2-32RM.

Zobacz przykłady 1A-1C

Szczegóły dotyczące użycia F2-32RM znajdują się w podręczniku użytkownika tego modułu. Podczas czytania podręcznika należy pamiętać, że był on napisany tylko dla systemu F2 i stąd wszystkie adresy wejścia/wyjścia są określone jak w systemie F2.



Zastosowanie instrukcji specjalnych FX.

(dotyczy wszystkich jednostek MPU typu FX i FX2c)

- RMST (FNC 93) - używana do wyprowadzenia kodu startowego z FX do F2-32RM
- RMWR (FNC 94) - używana do zablokowania wyjść F2-32RM
- RMRD (FNC 95) - używana do odczytania statusu ON/OFF F2-32RM
- RMMN (FNC 96) - używana do monitorowania prędkości i pozycji bieżącej F2-32RM

9.2.4. Użycie modułu FX2-24EI do obsługi F2-30GM.

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	-----------	--------------------	------------------

Numer programowe wejść/wyjść (adresy) modułu F2-30GM wyznaczone są na podstawie pozycji lokacji przyporządkowanego mu modułu FX2-24EI w zestawie sterownika. Jednak adres programowy nie ma bezpośredniego wpływu na numery wejść/wyjść. Numery wejść/wyjść mogą być wybierane przez użytkownika. Rysunek na stronie 9-3 pokazuje, jak przesuwanie pozycji użytego modułu FX2-24EI zmienia adresy programowe, ale nie zmienia numerów wejść/wyjść używanych przez F2-30GM.

Zobacz przykłady 2A-2C. Tam gdzie jest użyty w diagramie symbol ⊗ można wstawić "0", "4" lub "5". Zaleca się użycie "0".

Szczegóły dotyczące użycia F2-30GM znajdują się w podręczniku użytkownika. Podczas czytania podręcznika należy pamiętać, że był on napisany tylko dla systemu F2 i stąd wszystkie adresy wejścia/wyjścia są określone jak dla systemu F2.

Przykład:

Następująca tabela pokazuje zależność pomiędzy systemami FX i F2-30GM.

Ustawienie F2-30GM	Komentarz	Ustawienie FX
X [⊗] 14 do X [⊗] 27	Wyłącz -	X54 do X67
X [⊗] 40 do X [⊗] 47	Włączone przez -	Y40 do Y47
X0 do X6	nie związane	brak
Y0 do Y6	nie związane	brak

Użyty system FX jest podobny do pokazanego na rysunku ze strony 9-3, przykład 2B.



Zastosowanie instrukcji specjalnych FX.

(Brak zastosowania w jednostkach MPU FX2c lub FX z numerem wersji CPU większym od 3.06)

- BLK (FNC 97) - używana do utworzenia numeru bloku w F2-30GM
- MCDE (FNC 98) - używana do odczytywania numeru kodu M z F2-30GM

9.3. Adaptery połączeń równoległych FX2-40AP/W.

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Adaptery połączeń równoległych zapewniają bezpośrednią komunikację pomiędzy dwoma zestawami PLC. Istnieją dwa typy adapterów połączeń równoległych, wykorzystujące dwa różne media komunikacyjne.

a) Łącze światłowodowe - FX2-40AP

odległość transmisji: 50 m
 światłowód: F-OFC-M10 - długość 10 m
 F-OFC-M30 - długość 30 m
 F-OFC-M50 - długość 50 m

Wszystkie wymienione kable dostarczane są ze złączem CA9104AP

b) Łącze przewodowe (skrętka) - FX2-40AW.

odległość transmisji: 10 m
 łączyć razem zaciski [SA] na jednostce 1 z [SA] na jednostce 2.

Powtórzyć dla [SB] i [SG]. Na końcu połączyć także zacisk [SG] każdej jednostki z zaciskiem [SG] PLC.



Specjalne urządzenia systemowe

- M8070 - gdy jest w stanie ON, PLC pracuje jako "Master"
- M8071 - gdy jest w stanie ON, PLC pracuje jako "Slave"
- M8072 - gdy jest w stanie ON, jest nawiązana komunikacja pomiędzy stacjami
- M8073 - gdy jest w stanie ON, powstał błąd przydziału "Master/Slave"
- M8063 - gdy jest w stanie ON, powstał błąd połączenia - szczegóły w tablicy błędów

Zastosowanie instrukcji specjalnych

- PRUN (FNC 81) - identyfikuje urządzenie wejściowe, użyte do wymiany danych

Urządzenia transmisyjne:

Master > Slave

urządzenia bitowe - 100 sztuk
 (M800 do 899)

urządzenia słowowe - 10 sztuk
 (D490 do D499)

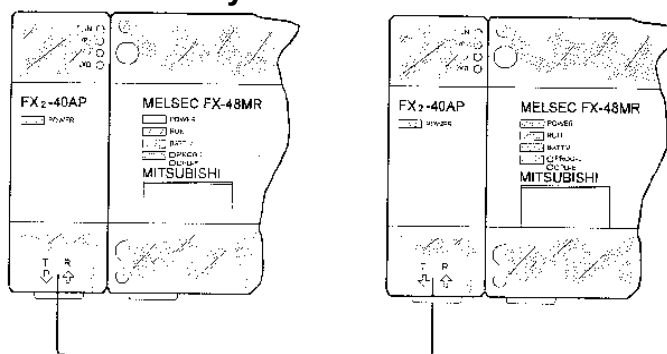
Czas komunikacji: 70 ms + (cykl stacji Master + cykl stacji Slave)

Slave > Master

urządzenia bitowe - 100 sztuk
 (M900 do M999)

urządzenia słowowe - 10 sztuk
 (D500 do D509)

Ogólne rozmieszczenie elementów systemu:



9.4 Funkcje zegara czasu rzeczywistego

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Kiedy kasetka pamięci zegara czasu rzeczywistego (RTC) jest zainstalowana w FX lub FX_{0N}, funkcje zegara czasu rzeczywistego dla tego PLC są automatycznie włączone.

Dane czasu kasety RTC są podtrzymywane bateryjnie. Oznacza to, że kiedy PLC jest wyłączony, dane czasu i ustawienia nie są tracone ani zmieniane. Okres przechowywania danych zależy od stanu baterii. Obwody RTC pobierają nieznaczny prąd, porównywalny do pamięci RAM (dostępnej tylko na FX).

Kasety RTC mają w najgorszym przypadku dokładność +/- 45 s na miesiąc w temperaturze otoczenia 25 C. Funkcja kalendarzowa RTC uwzględnia lata przestępne w okresie 1980 do 2079.



Sterowniki programowalne FX w wersji wcześniejszej od 2.0.

- Te sterowniki nie realizują funkcji RTC.



Dostępne kasety RTC:

Kaseta RTC	Komentarz
FX-RTC	Tylko funkcje zegara czasu rzeczywistego
FX-RAM-8C	Funkcje zegara czasu rzeczywistego + 8K RAM (uwaga: jednostka FX _{0N} nie może wykorzystywać tej kasety ze względu na brak gwarancji zachowania programu w RAM)
FX-EEPROM-4C	Funkcje zegara czasu rzeczywistego + 4K EEPROM (uwaga: jednostka FX _{0N} wykorzystuje tylko 2 K pamięci EEPROM)

Rejestry funkcyjne i znaczniki

- Lista dostępnych urządzeń zegarowych - strona 6-3
- Dostępność funkcji RTC na FX_{2N} - sekcja 5.14, strona 5-138



Ogólne zasady używania kaset pamięci

MPU FX i FX_{2c} mogą używać każdego typu dostępnych kaset pamięci
FX_{0N} mogą używać tylko kaset typu EEPROM lub EPROM i wtedy mogą osiągnąć pojemność tylko 2K kroków programu.



Należy zainstalować baterię FX_{0N}-40BL aby podtrzymać wskazanie czasu RTC w jednostkach FX_{0N}.



Jednostka FX_{2N} ma wbudowany do MPU zegar czasu rzeczywistego. Chociaż możliwe jest użycie kaset RTC w charakterze pamięci, nie ma potrzeby używania ich do otrzymania funkcji RTC.

9.4.1 Ustawianie zegara czasu rzeczywistego

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Zegar czasu rzeczywistego może być ustawiany przy użyciu następujących specjalnych rejestrów i znaczników:

Rejestry zegara czasu rzeczywistego

Numer Urządzenia	Funkcja	Zakres
D8013	sekundy	0 do 59
D8014	minuty	0 do 59
D8015	godziny	0 do 23
D8016	dzień	1 do 31 (prawidłowo dla bieżącego miesiąca)
D8017	miesiąc	1 do 12
D8018	rok	00 do 99 (1980 do 2979)
D8019	dzień tygodnia	0 do 6 (od niedzieli do soboty)

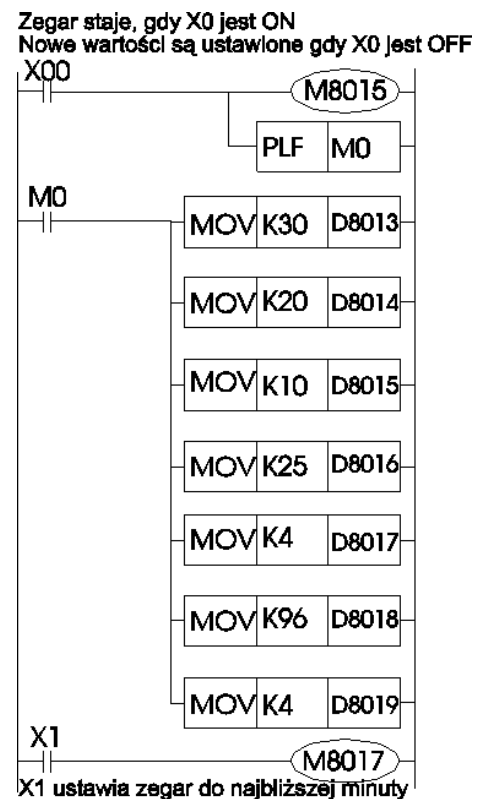
Znaczniki sterujące

Numer urządzenia	Komentarz
M8015 ustawianie czasu	Ustawienie w stan ON zatrzymuje zegar. Wtedy wartości czasu mogą być zmienione. Zegar startuje po ustawieniu znacznika w stan OFF
M8016 zatrask czasu	Dane zegara są zatraskiwane w rejestrze. Zegar pracuje nadal.
M8017 zaokrąglenie minuty	Ustawienie w stan ON zaokrągla czas w górę lub w dół do najbliższej minuty
M8018 dostępność zegara	Ustawiany automatycznie wskazuje dostępność zegara czasu rzeczywistego
M8019 błąd ustawiania	Ustawienie w stan ON sygnalizuje wykroczenie ustawień poza dopuszczalne zakresy

Te urządzenia używane są tak jak pokazano na przykładzie obok.



Uwaga: Jednostka FX_{2N} ma specjalną instrukcję, która upraszcza ustawianie i korzystanie z RTC. Szczegóły w sekcji 5.14



1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

10	Wskazówki techniczne	10-1
10.1	Wskazówki zaawansowanego programowania	10-1
10.2	Użytkownicy jednostek FX zasilanych prądem stałym	10-1
10.3	Wykorzystanie znaczników RUN/STOP	10-2
10.3.1	Konfiguracja przycisku RUN/STOP	10-2
10.3.2	Zdalne sterowanie RUN/STOP	10-3
10.4	Tryb stałego czasu przebiegu programu	10-4
10.5	Zamiana stanów ON/OFF	10-4
10.6	Stosowanie urządzeń z podtrzymaniem w celu osiągnięcia maksymalnej sprawności	10-5
10.7	Indeksowanie przy wyświetlaniu wielu wartości danych	10-5
10.8	Czytanie i obróbka danych z nastawników cyfrowych	10-6
10.9	Pomiary impulsów o wysokiej częstotliwości	10-6
10.9.1	Pomiar impulsu z wykorzystaniem timera 1 ms	10-6
10.9.2	Pomiar impulsu z wykorzystaniem timera 0,1 ms	10-7
10.10	Wykorzystanie znacznika zakończenia wykonywania operacji M8029	10-7
10.11	Tworzenie instrukcji MTR definiowanej przez użytkownika	10-8
10.12	Przykładowa aplikacja systemu, wykorzystująca STL i IST	10-8
10.13	Wykorzystanie instrukcji PWM do sterowania silnikiem	10-15
10.14	Techniki programowania PID	10-18
10.14.1	Ustawienie parametrów początkowych PID	10-18
10.14.2	Utrzymywanie MV w zadanym zakresie	10-19
10.14.3	Zmiana sterowania ręczna/automatyczna	10-19
10.14.4	Wykorzystanie sygnałów alarmowych PID	10-20
10.14.5	Inne wskazówki programowania PID	10-20
10.15	Dodatkowe funkcje PID	10-21
10.15.1	Sterowanie zakresem wartości wyjściowej	10-21
10.15.2	Operacja samodostrajania	10-22

10 Wskazówki techniczne

FX_{0(S)}FX_{0N}

FX

FX_(2C)FX_{2N}

10.1 Wskazówki zaawansowanego programowania

Rodzina FX sterowników programowalnych zawiera łatwy do nauki i prosty w użyciu zestaw instrukcji, który umożliwia wykonywanie skomplikowanych funkcji przy pomocy prostych programów. Ten rozdział omawia jedną lub dwie użyteczne techniki dostarczając użytkownikowi wartościowych przykładów programów.



Jeżeli niektóre z tych technik będą zastosowane w programach, użytkownik musi upewnić się, że wykonają one wymagane zadanie lub operację. Mitsubishi Electric nie ponosi odpowiedzialności za programy użytkownika zawierające jakiegokolwiek przykład z tego podręcznika. Każdy program będzie zawierał krótkie wyjaśnienie systemu. Należy zauważyć, że dla każdej instrukcji nie będą omówione metody "jak programować" i "jakie parametry są osiągalne". Te informacje znajdują się w odpowiednich poprzednich rozdziałach.

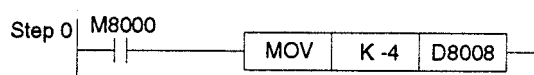
10.2 Użytkownicy jednostek FX zasilanych prądem stałym

FX_{0(S)}FX_{0N}

FX

FX_(2C)FX_{2N}

Kiedy używane są sterowniki FX zasilane prądem stałym, konieczne należy dodać następujące instrukcje na początku zainstalowanego programu:



Objaśnienie:

W sterownikach FX zasilanych prądem zmiennym, okres wykrycia zaniku napięcia może być dopasowany przez wpisanie wymaganej wartości okresu do rejestru D8008. Jednakże dla jednostek zasilanych prądem stałym okres ten musi być ustawiony na 5 ms.



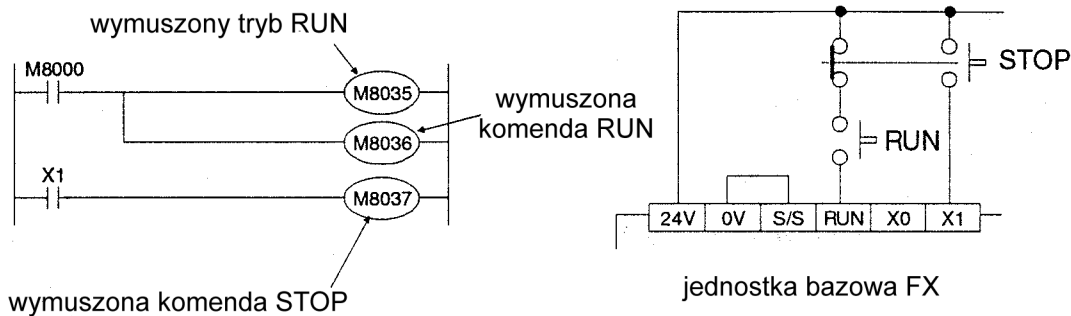
Osiąga się to przez wpisanie wartości -4 do D8008. W przeciwnym wypadku może to doprowadzić do utraty informacji wejściowych podczas zaniku napięcia zasilającego.

10.3 Wykorzystanie znaczników RUN/STOP

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

10.3.1 Konfiguracja przycisku RUN/STOP

Sterowniki FX posiadają pojedynczy zacisk RUN. Po podłączeniu zasilania do tej końcówki, PLC przechodzi w stan RUN, tzn. wykonywany jest program użytkownika. Konsekwentnie, kiedy zasilanie nie jest podłączone do zacisku RUN, PLC jest w stanie STOP. Ta właściwość może być wykorzystana do uzupełnienia PLC w zewnętrzny przycisk sterujący RUN/STOP. Wymagane jest następujące dodatkowe okablowanie i oprogramowanie PLC:



Objaśnienie:

Naciśnięcie przycisku sterującego RUN wprowadza PLC w stan RUN. Oznacza to, że M8000 jest w stanie ON. Zgodnie z programem, M8000 uaktywnia zarówno M8035 i M8036. Te dwa urządzenia ustawiają PLC w wymuszonym trybie RUN. Zwolnienie przycisku RUN wprowadziłoby PLC w stan STOP, ale ponieważ M8035 i M8036 są w stanie ON, PLC pozostaje w stanie RUN. Aby zatrzymać PLC, naciśnięcie przycisku STOP wymusza na wejściu X1 stan ON i konsekwentnie M8037 ustawia się w stan ON. To automatycznie wymusza stan OFF na obu M8035 i M8036 oraz M8037. Stąd PLC przechodzi w stan STOP i oczekuje na ponowne rozpoczęcie cyklu.

Priorytety wejść:



Stan wejścia STOP jest uwzględniany dopiero, gdy w programie zostanie wykonana instrukcja END. Tak się dzieje, ponieważ fizyczne wejście, tzn. urządzenie X jest normalnie w tym czasie aktualizowane i przetwarzane. Dlatego wejście RUN ma priorytet, gdy oba przyciski, RUN i STOP, zostaną naciśnięte równocześnie.

Aby zapewnić priorytet wejściu STOP i stworzyć bezpieczniejszy system, należy skonstruować pewien rodzaj mechaniczno/elektrycznej blokady między wejściami RUN i STOP. Prosty przykład jest pokazany na rysunku powyżej.



Aby sterowanie przyciskami RUN/STOP działało poprawnie, użytkownik jednostek FX2C i FX2N musi ustawić je w pozycji STOP.

Jednostki FX2N nie zawierają końcówki RUN. Jedno z wejść X0 do X17 (X0 do X7 w FX2N-16M) powinno być skonfigurowane poprzez ustawienie parametrów jako końcówka RUN.

10.3.2 Zdalne sterowanie RUN/STOP

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Rodzina sterowników programowalnych FX może być sterowana, tzn. przełączana w stan RUN lub STOP i umożliwia monitorowanie stanu PLC przy użyciu zewnętrznych, inteligentnych urządzeń sterujących, takich jak komputery i jednostki dostępu do danych Mitsubishi FX. Następujący przykład wykorzystuje jednostkę graficzną FX-DU.

Objaśnienie:

W tym przykładzie sterownik programowalny nie wymaga żadnego specjalnego okablowania czy też dodatkowego oprogramowania.

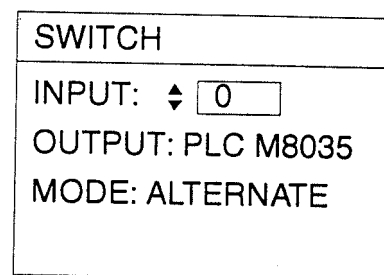
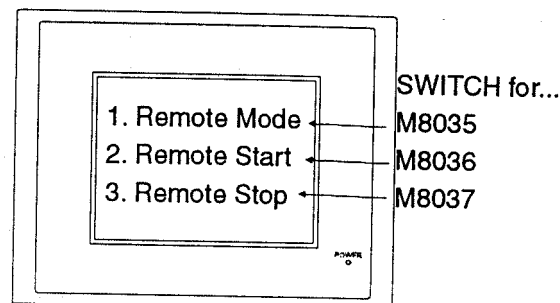


Jedynym wymaganym warunkiem jest, aby PLC nie był w stanie RUN, tzn. brak połączenia z zaciskiem RUN, a przełącznik RUN/STOP w tych PLC, które go zawierają, jest w położeniu STOP.



Jednostka DU może być programowana przełącznikami, sterującymi trzema znacznikami: M8035, M8036 i M8037. Poprzez aktywację przełączników dla M8035 i M8036 PLC może być przełączony w stan RUN, podczas gdy aktywowanie M8037 powoduje przejście PLC w stan OFF.

Przykład ustawienia przełączników znajduje się powyżej. Przełączniki dla M8035 i M8036 powinny być dwustanowe, natomiast przełącznik dla M8037 powinien być chwilowy (szczegóły programowania i działania przełączników w podręczniku obsługi DU).



Uwaga: Gdy M8035 i M8036 są w stanie ON, jednostka MPU nie może być przełączona w stan STOP przez odłączenie końcówki RUN lub przełącznik RUN/STOP. Należy ustawić M8037 w stan ON lub wyzerować M8035 i M8036 aby powrócić do normalnego stanu operacyjnego



Zakres jednostek graficznych Mitsubishi FX-DU:

FX-25DU-E - jednostka tekstowo/graficzna, 4-liniowa

FX-30DU-E - jednostka tekstowo/graficzna, 4-liniowa, z klawiaturą membranową

FX-40DU-TK-E - jednostka tekstowo/graficzna, 7-liniowa, z dotykową klawiaturą numeryczną

FX-50-DU-TK(S)-E - jednostka tekstowo/graficzna, kolorowa, 15-liniowa, z klawiszami dotykowymi

Zdalny STOP w FX2N

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

W jednostkach FX2N, nawet, gdy końcówka RUN lub przełącznik RUN/STOP jest w pozycji RUN, możliwe jest zdalne wprowadzenie PLC w stan STOP poprzez wprowadzenie M8037 w stan ON.

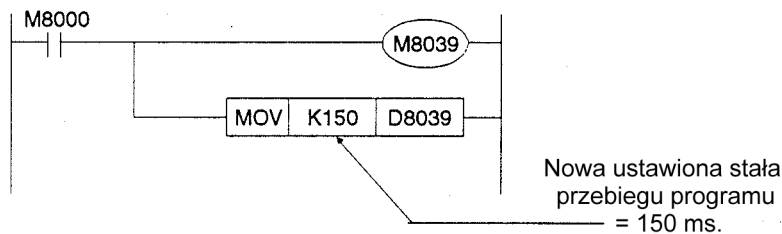
Powrót do stanu RUN następuje po wyzerowaniu M8037.

10.4 Tryb stałego czasu przebiegu programu

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Czasami czas trwania operacji może być problemem, szczególnie jeżeli kiedy usiłuje się osiągnąć koordynację z drugim systemem sterowania. W takim przypadku bardzo pomocne jest ustalenie czasu przebiegu programu. W normalnych warunkach czas przebiegu programu PLC będzie różny w kolejnych przebiegach. Tak się dzieje ponieważ normalnie czas przebiegu zależy od liczby i typu aktywnych instrukcji. Ponieważ zmieniają się one w różnych przebiegach programu, zmienia się również czas przebiegu. Stosując dodatkową funkcję programową opisaną poniżej, można ustalić czas przebiegu tak, aby był on jednakowy w każdym cyklu. Bieżący czas przebiegu ustawia się przez wpisanie wartości co najmniej najdłuższego czasu przebiegu programu do rejestru D8039 (w przykładzie poniżej użyto wartość K150). Jeżeli czas przebiegu jest krótszy od wpisanej wartości, pojawi się "przerwa" do momentu osiągnięcia ustalonego czasu przebiegu.

Poniższy fragment powinien być umieszczony na początku programu użytkownika.

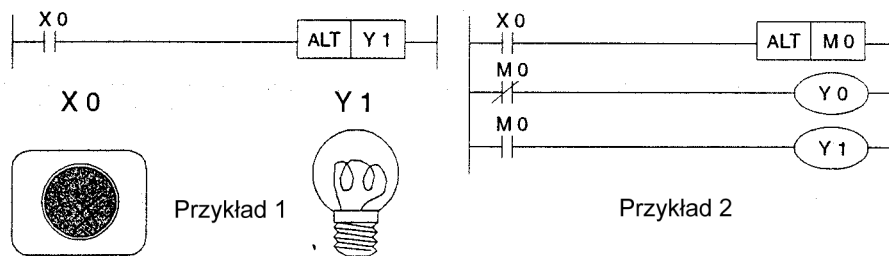


10.5 Zamiana stanów ON/OFF

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

W wielu przypadkach przydatne jest posiadanie pojedynczego wejścia sterującego lub przełączanie sytuacji. Podstawowym, ale bardzo typowym przykładem jest włączanie/wyłączanie światła. Można to łatwo osiągnąć, używając standardowego programu drabinkowego do odczytania stanu wejścia i przełączenia stanu wyjścia. Jednak, ten system wymaga wejścia zatraskowego. Użycie podstawowych instrukcji drabinkowych prowadzi szybko do skomplikowania programu. Użycie instrukcji ALT do przełączania stanów ON/OFF (SET/RESET, START/STOP, SLOW/FAST) jest dużo prostsze, szybsze i bardziej wydajne.

Objaśnienie:



Naciśnięcie (jeden raz) chwilowego przycisku X1 włącza lampę. Naciśnięcie przycisku po raz drugi wyłącza lampę. Jeżeli przycisk zostanie naciśnięty po raz trzeci, lampa będzie znowu włączona i w ten sposób przełączanie stanu będzie kontynuowane. Drugi pokazany program opisuje możliwą blokadę/sterowanie silnikiem, sytuację start/stop.

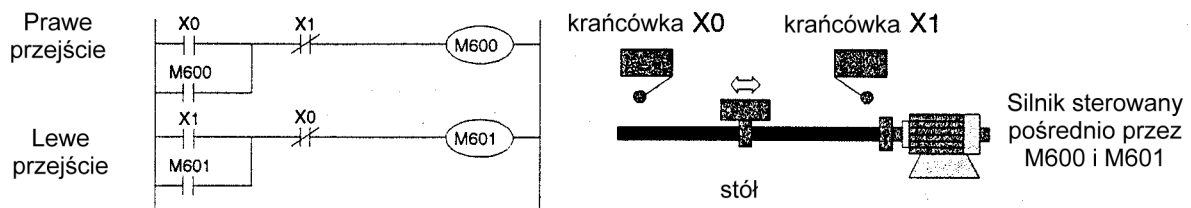
10.6 Stosowanie urządzeń z podtrzymaniem w celu osiągnięcia maksymalnej sprawności

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Urządzenia zasilane bateryjnie zachowują swój status w przypadku zaniku zasilania PLC. Te urządzenia mogą być użyte do osiągnięcia maksymalnej sprawności, poprzez umożliwienie kontynuowania pracy PLC od ostatniej operacji przed zanikiem zasilania.

Na przykład: system przesuwu poprzecznego przemieszcza stół roboczy naprzemiennie pomiędzy dwoma wyłącznikami krańcowymi. Gdy wystąpi zanik zasilania podczas przesuwu, maszyna zatrzyma się. Byłoby idealnie, gdyby PLC po wznowieniu zasilania kontynuował ruch z położenia, w którym pozostawał, tzn. w kierunku, jaki był przed zanikiem zasilania.

Objaśnienie:



Stan zatraskowych urządzeń (w tym przykładzie M600 i M601) jest zachowywany podczas zaniku zasilania. Po powrocie zasilania styki M ustawiają się w stan sprzed zaniku zasilania.

10.7 Indeksowanie przy wyświetlaniu wielu wartości danych

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

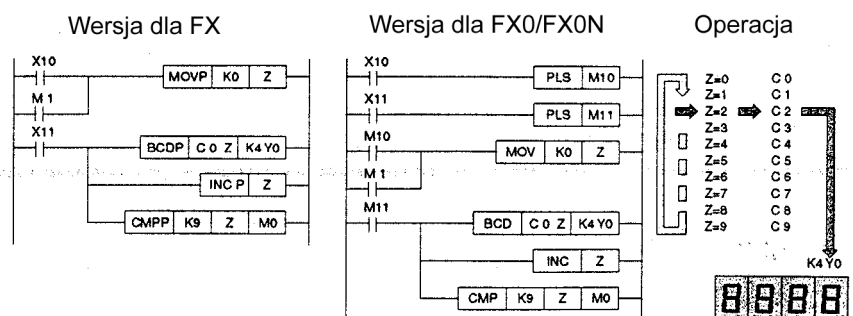
Wielu użytkowników nieświadomie wpada w pułapkę używania pojedynczego wyświetlacza 7-segmentowego do wyświetlania tylko jednej wartości danej. Bardzo prosta kombinacja instrukcji specjalnych pokazuje, jak użytkownik może "stronicować" wiele wartości danych wyświetlanych po kolei.

Objaśnienie:

Zawartość 10 liczników jest wyświetlana w operacji sekwencyjnego

"stronicowania". To działanie pojawia się za każdym razem, gdy jest odebrany sygnał z wejścia X11. Rejestr indeksowy Z jest w sposób ciągle inkrementowany do momentu, aż osiągnie wartość

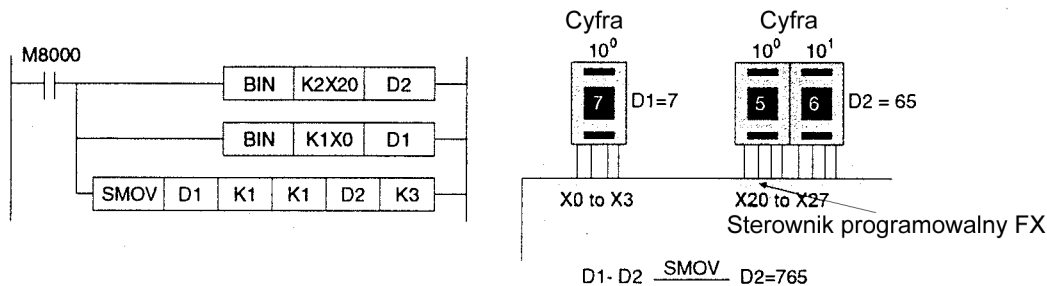
równą 9. Kiedy to się stanie, instrukcja porównująca ustawia M1 w stan ON, co z kolei zeruje zawartość rejestru Z. Stąd pojawia się efekt pętli, gdzie Z zmienia się od 0 do 9 (10 wartości). Wartość Z jest używana do wybrania następnego licznika, który ma być wyświetlony na wyświetlaczu 7-segmentowym. Tak się dzieje ponieważ modyfikator indeksowy Z jest użyty do offsetu licznika, który jest czytany przez instrukcję BCD.



10.8 Czytanie i obróbka danych z nastawników cyfrowych

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Dane mogą być łatwo wczytane do sterownika poprzez użycie instrukcji BIN. Gdy dane są czytane z wielu źródeł, są umieszczane w różnych miejscach. Może być wymagane łączenie poszczególnych danych w celu wytworzenia nowych danych. Można też wydzielać cyfry danych z formatów słowowych. Obsługa i manipulowanie takim rodzajem danych jest możliwa poprzez zastosowanie instrukcji SMOV. Przykład poniżej pokazuje, jak dwie wartości danych (pojedyncza cyfra i liczba dwucyfrowa) mogą być połączone w celu wytworzenia danej końcowej.



Objaśnienie:

Każda z dwu instrukcji BIN czyta jedną wartość danej. Pierwsza wartość, pojedyncza cyfra przechowana w D1, jest łączona z drugą wartością D2 (zawierającą 2 cyfry). Jest to wykonywane przez instrukcję SMOV. W wyniku tego zawartość D1 jest dopisywana jako trzecia cyfra do zawartości D2. Wynik jest następnie wprowadzony z powrotem do D2.

10.9 Pomiary impulsów o wysokiej częstotliwości

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

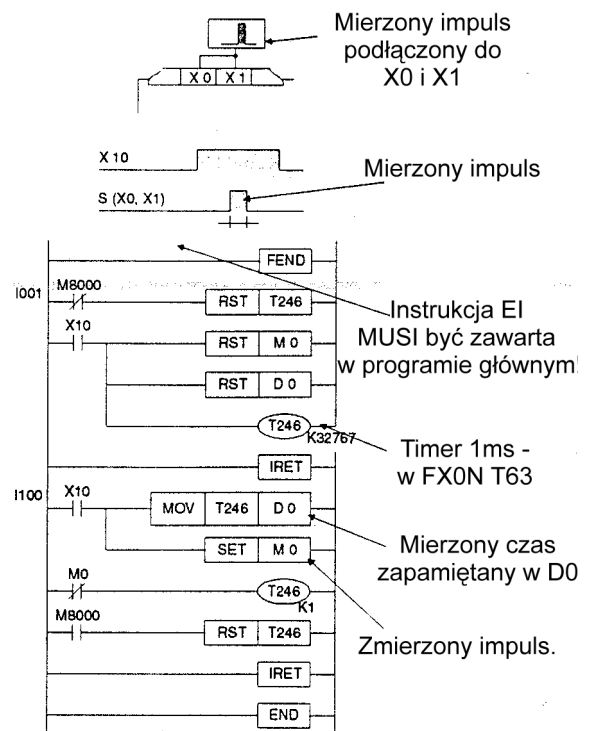
10.9.1 Pomiar impulsu z wykorzystaniem timera 1 ms

Czasami z powodu wymagań systemowych lub potrzeb serwisowych istnieje konieczność określenia czasu trwania impulsów wejściowych. Następujący program wykorzystuje dwa podprogramy przerwań do pomiaru szerokości impulsu przy pomocy timera 1 ms. Timer, użyty w przykładzie, jest jednym z timerów FX. Jednak w sterowniku FX_{0N}, w podobnej sytuacji, może być użyty T63.

Objaśnienie:

Timer 1 ms, T246 jest wysterowany przerwaniem I001. Kiedy sygnał na wejściu X1 zanika, wartość bieżąca timera T246 jest przesuwana do rejestru D0 przez podprogram obsługi przerwania I100. Znacznik ukończenia operacji M0 jest ustawiany w stan ON.

Uwaga: X10 działa jako znacznik odblokowania/zablokowania.



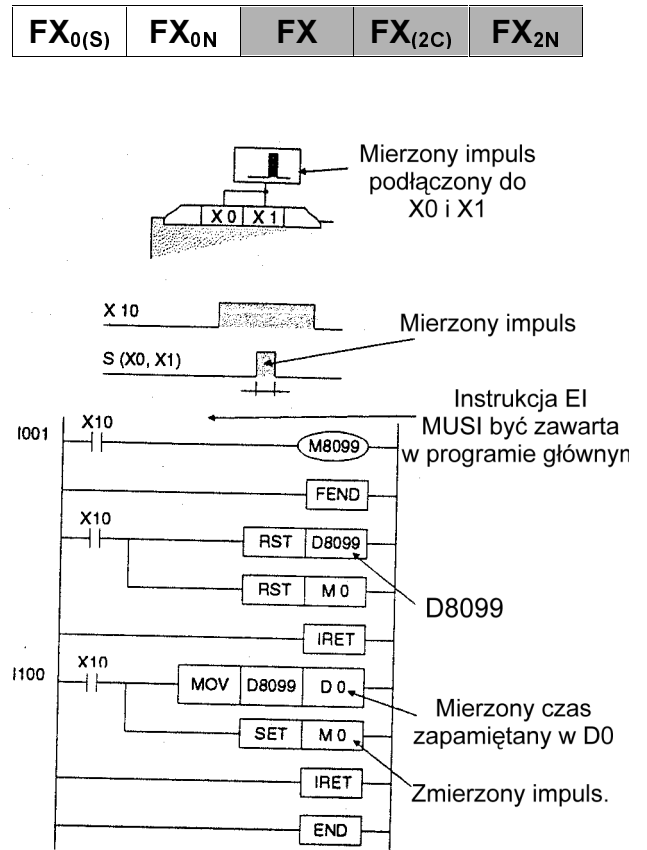
10.9.2 Pomiar impulsu z wykorzystaniem timera 0,1 ms

Jest to bardzo dokładny proces pomiaru dla wejść impulsowych. Użycie w tym przypadku standardowego timera nie jest dość dokładne, ponieważ najwyższa rozdzielczość wynosi 1 ms. Dlatego ten przykład pokazuje, jak urządzenia wysokiej dokładności M8099 i D8099 są wykorzystane do pomiaru szerokości impulsów z rozdzielczością 0,1 ms.

Objaśnienie:

Impuls jest mierzony przy wykorzystaniu dwóch przerwań. Podprogramy obsługi tych przerwań działają niezależnie od siebie, jeden wyzwalany jest zboczem narastającym, drugi opadającym, tego samego impulsu. Podczas trwania impulsu zawartość rejestru D8099 jest przesuwana do rejestru danych D0 w sposób ciągły. Po zakończeniu impulsu, zawartość D0 może być odczytana w dowolnej chwili.

Należy zauważyć, że aby ten tryb dużej szybkości/dokładności był aktywowany dla D8099, odpowiadający mu znacznik M8099 musi być w stanie ON w głównym programie.

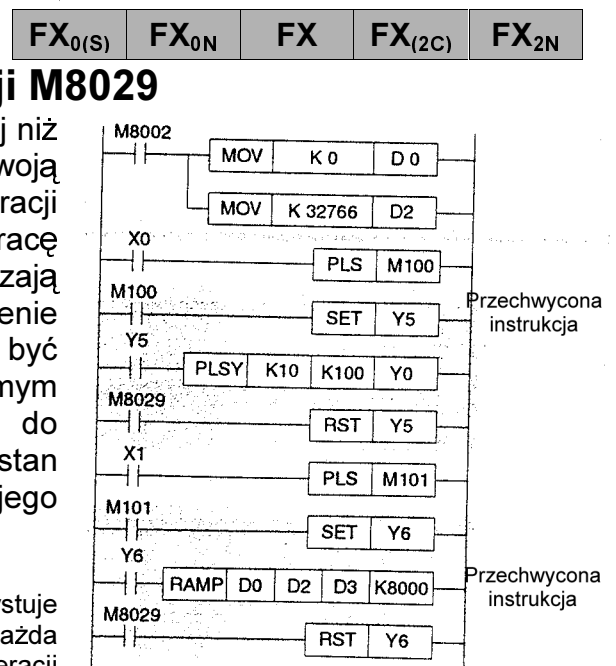


10.10 Wykorzystanie znacznika zakończenia wykonywania operacji M8029

Niektóre z instrukcji specjalnych potrzebują więcej niż jednego przebiegu programu, aby zakończyć swoją operację. Identyfikacja bieżącego statusu operacji staje się przez to trudna. Aby wspomóc pracę programisty, niektóre instrukcje specjalne oznaczają ukończenie swojej operacji poprzez ustawienie znacznika M8029. Ponieważ ten znacznik może być używany przez kilka różnych instrukcji w tym samym czasie, należy skorzystać z metody, podobnej do przedstawionej poniżej, przechwytywającej stan znacznika M8029 dla każdej instrukcji, która jego używa.

Objaśnienie:

Sekwencja "przechwytywająca" znacznik M8029 wykorzystuje wsadowe odświeżanie sterowników rodziny FX. Każda instrukcja wykorzystująca znacznik zakończenia operacji M8029, bezpośrednio po jej wykonaniu w każdym cyklu programu, powinna być powiązana z natychmiastowym sprawdzaniem stanu M8029. Stąd, poprzez natychmiastowe zerowanie (lub ustawianie) pomocniczego znacznika sterującego dla każdej instrukcji, przechwytywany jest jej status operacyjny. Tak więc po wsadowym odświeżeniu zerowane są tylko instrukcje, które zakończyły działanie. Przykład obok używa impulsu do ustawienia znaczników sterujących, tak więc łatwo jest monitorować ich stan i zauważyć, kiedy instrukcja zakończyła działanie (jeśli instrukcje działają w sposób ciągły, byłoby trudno zauważyć, kiedy się zakończyły!).



10.11 Tworzenie instrukcji MTR definiowanej przez użytkownika

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

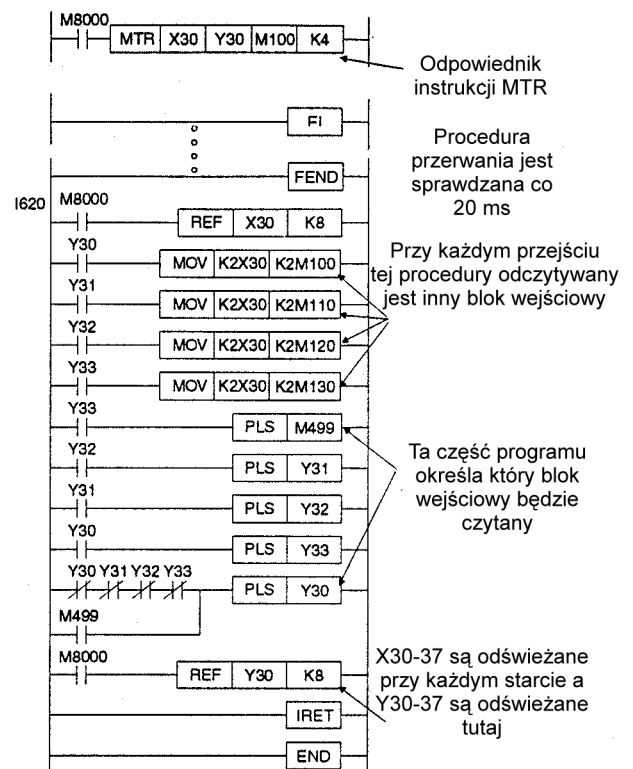
Dla tych użytkowników, którzy chcą wykorzystać zalety instrukcji MTR, którzy chcą określić więcej niż jeden obszar instrukcji MTR, możliwość definiowania instrukcji MTR będzie bardzo pomocna.

Objaśnienie:

Sterowanie programu opiera się na obsłudze przerwania czasowego I620. To przerwanie wyzwalane jest co 20 ms, niezależnie od tego, co wykonuje program główny. Podczas każdego przerwania odczytywany jest jeden bank ze zdefiniowanej przez użytkownika matrycy. Aktywny bank matrycy wybierany jest przez odpowiednie sterowanie wyjściami Y33-Y30 (w danej chwili aktywne może być tylko jedno wyjście). Odczytane dane są zapamiętywane w sekwencyjnym zbiorze rejestrów.

Każda instrukcja MOV odczytuje nowy bank multipleksowanych wejść.

Bezpośrednio przed instrukcją MTR, zdefiniowaną przez użytkownika pokazana jest równoważna instrukcja MTR. Więcej szczegółów na stronie 5-54.



10.12 Przykładowa aplikacja systemu, wykorzystująca STL i IST

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Następująca ilustracja pokazuje prosty system "podnieś i umieść" wykorzystująca ramię małego robota. Punkt zerowy jest zdefiniowany jako najwyższe i możliwie najbardziej odsunięte w lewo położenie ramienia robota.

Normalna sekwencja zdarzeń:

Ramię robota przenosi produkt z punktu 'A' do punktu 'B'. Aby wykonać taką operację musi zajść następująca sekwencja zdarzeń:

Pozycja początkowa: ramię robota jest w położeniu zerowym.

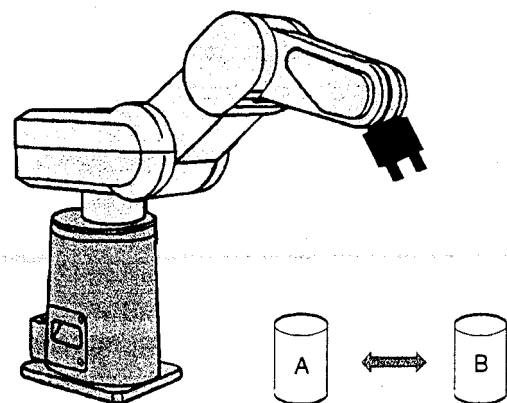
1) uchwyt robota jest opuszczany do położenia skrajnego dolnego

- wyjście Y0: ON, wejście X1: ON, wyjście Y0:

OFF

2) uchwyt obejmuje produkt w punkcie A

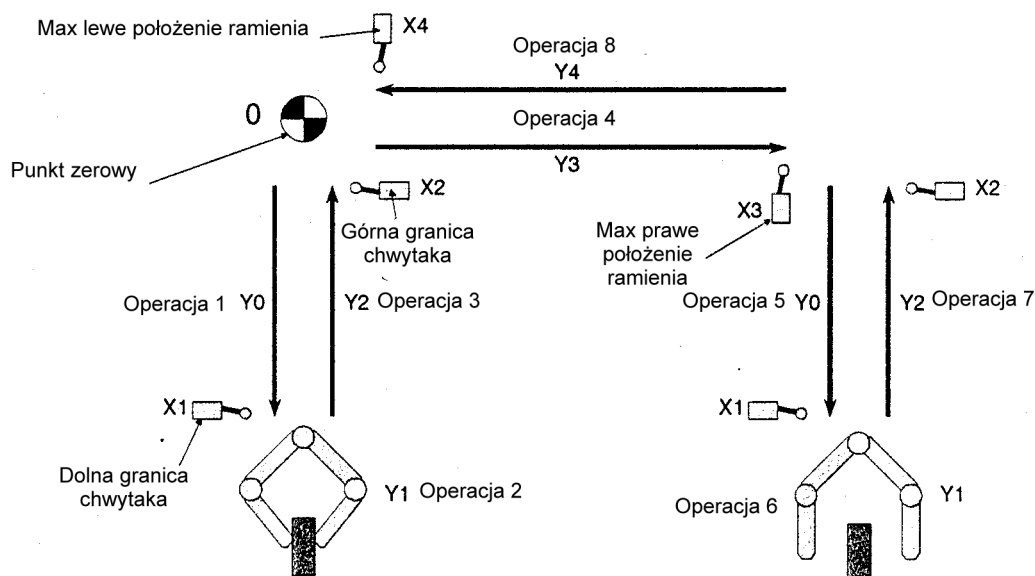
- wyjście Y1: ON



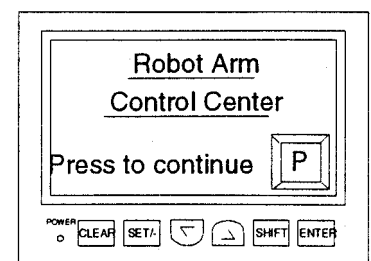
- 3) uchwyt, trzymający produkt, jest podnoszony do położenia skrajnego górnego
 - wyjście Y2: ON, wejście X2: ON, wyjście Y2: OFF
- 4) ramię robota przesuwane jest w położenie skrajne prawe
 - wyjście Y3: ON, wejście X3: ON, wyjście Y3: OFF
- 5) uchwyt i produkt są opuszczane do położenia skrajnego dolnego
 - wyjście Y0: ON, wejście X1: ON, wyjście Y0: OFF
- 6) uchwyt zostaje zwolniony i produkt pozostaje w punkcie B
 - wyjście Y1: OFF
- 7) uchwyt jest podnoszony do położenia skrajnego górnego
 - wyjście Y0: ON, wejście X2: ON, wyjście Y0: OFF
- 8) ramię przenoszone jest do punktu zerowego w położeniu skrajnym lewym.
 - wyjście Y4: ON, wejście X4: ON, wyjście Y4: OFF

Po czym cykl jest ponownie powtarzany.

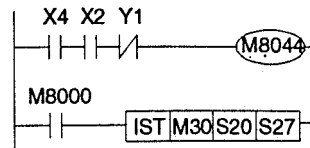
Parametry systemu



- I) Podwójne elektromagnesy są użyte do sterowania ruchem w górę (Y2)/w dół (Y0) oraz w prawo (Y3)/w lewo (Y4)
- II) Pojedynczy elektromagnes steruje ściskaniem i otwieraniem (Y1) chwytaka
- III) W system wykorzystuje się FX-40DU-TK do komunikacji z operatorem. FX-40DU-TK jest jednostką z ekranem dotykowym.

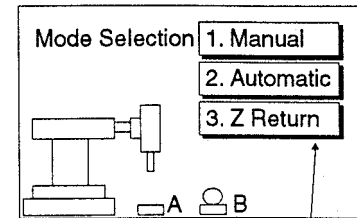


W tym przykładzie do sterowania trybem operacyjnym ramienia robota wykorzystano instrukcję IST (FNC 60). Przykład pokazany obok przedstawia, jak instrukcja IST jest zapisana w głównym programie.



Kiedy wszystkie warunki są spełnione to chwytak jest w punkcie 0
M8044 = ON
IST ustala tryb operacji

Podczas użycia instrukcji IST istnieje 5 możliwych trybów, które pozwalają na dostęp do trzech niezależnych programów. Przykład pokazuje programy, skojarzone z tymi trybami. Każdy tryb jest wybierany poprzez jednostkę FX-40DU-TK. Na ekranie obok pokazano początkowe menu trybów. Każda z opcji menu powoduje skok do ekranu wybranego trybu. Opcje menu 1 i 3 ustawiają także znaczniki, odpowiednio, M30 i M31 w stan ON. Aktywne bity wyzwalają następnie zmianę do ekranu odpowiedniego trybu. Należy zauważyć, że tryb 'Automatyczny' ma trzy dalsze tryby, wybierane z następujących ekranów.



Klawisze wyboru

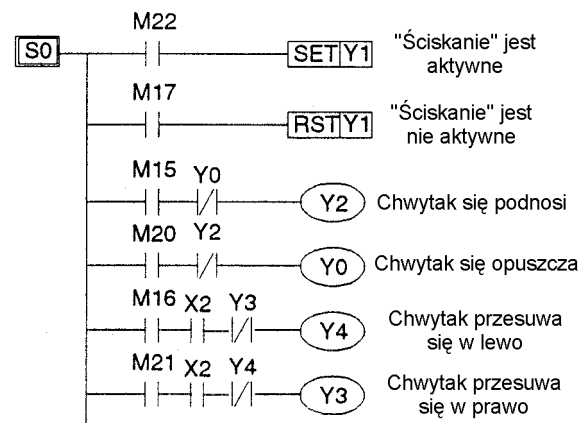
wybranych z następujących ekranów.

Tryb ręczny

W tym trybie wszystkie operacje, wykonywane przez robota, są kontrolowane przez operatora. Operacja lub ruch jest wybierany poprzez naciśnięcie odpowiedniej opcji na ekranie jednostki DU (patrz poniżej). Ta opcja wyzwala przełącznik DU, który steruje przyporządkowanym przełącznikiem wewnątrz sterownika. Przełączniki powinny działać jako chwilowe, a więc są aktywne tylko w chwili ich naciskania.

Stan operacji uchwycenia może być zidentyfikowany przez dwie funkcje INDICATOR (SCR) w jednostce DU. Mogą one monitorować stan ON/OFF wyjścia Y1. Stąd, gdy uchwyt jest aktywny (ON), obok przycisku CLAMP ON pojawia się czarny kwadrat. Gdy uchwyt nie jest aktywny (OFF), kwadrat pojawi się obok przycisku OFF.

Jednocześnie tylko jeden z kwadratów może być aktywny.



Przyporządkowanie klawiszy na ekranie DU obok

Góra (Up) = M15

Dół (Down) = M20

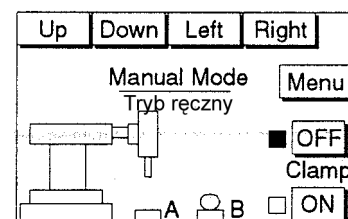
Lewo (Left) = M16

Prawo (Right) = M21

Uchwyt (Clamp) ON = M22

Uchwyt (Clamp) OFF = M17

Menu = zerowanie M30

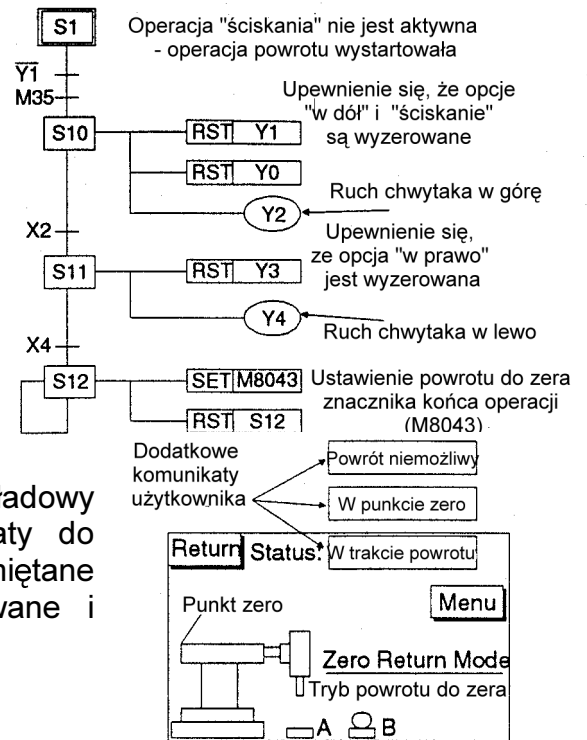


Po zakończeniu wykonywania operacji ręcznej operator może powrócić do ekranu wyboru poprzez naciśnięcie klawisza "Menu". Spowoduje to wyzerowanie znacznika trybu ręcznego M30 i wyświetlenie ekranu wyboru trybu.

Tryb z powrotem do zera

Ten tryb wypełnia funkcje inicjujące poprzez przesunięcie ramienia robota do ustalonego położenia. Po wybraniu trybu "Z Return" znacznik M35 ustawiany jest w stan ON. W tym momencie DU wyświetla ekran sterujący "powrotu do zera". Bieżąca operacja "powrotu do zera" rozpoczyna się gdy został naciśnięty przycisk "Return" (uaktywniony M25) i ramię robota nie jest aktywne, tzn. Y1 jest OFF (na schemacie STL obok stan Y1 OFF jest pokazany jako Y1).

Jednostka DU może być wykorzystana do informowania o statusie bieżącej operacji. Przykładowy ekran pokazany obok wykorzystuje 3 komunikaty do pokazania statusu. Komunikaty mogą być zapamiętane jako ciągi tekstowe w PLC. Są one odczytywane i wyświetlane poprzez opcję ASCII jednostki DU.



Po osiągnięciu punktu zerowego operator może powrócić do ekranu wyboru trybu, poprzez naciśnięcie klawisza "Menu". Spowoduje to wyzerowanie znacznika M31 i zmianę zawartości ekranu DU.

Przyporządkowanie klawiszy dla ekranu DU powyżej:

Return = M25

Menu = wyzerowanie M31

Tryb automatyczny

Ten tryb zawiera trzy tryby podrzędne:

tryb krokowy:

- program wykonywany jest krokowo, instrukcja po instrukcji, po każdorazowym naciśnięciu klawisza "Start"

tryb cykliczny:

- program wykonuje automatycznie jeden cykl operacyjny. Każdy cykl jest inicjowany naciśnięciem klawisza "Start". Jeżeli w trakcie wykonywania cyklu zostanie naciśnięty klawisz "Stop", cykl jest natychmiast zatrzymywany. Wznowienie cyklu następuje po naciśnięciu klawisza "Start"

tryb automatyczny (pełny)

- program wykonuje się w pełni automatycznie. Program może być zatrzymany po naciśnięciu klawisza "Stop", ale dopiero po zakończeniu bieżącego cyklu.

W tym przykładzie te trzy tryby są wybierane przełącznikiem obrotowym. Przełącznik nie jest podłączony do PLC, lecz do szyny I/O z tyłu jednostki DU. Użycie przełącznika a obrotowego zapewnia jednoznaczny wybór trybu. Aby zwiększyć przyjazność otoczenia operatora, bieżący tryb jest wyświetlany na ekranie DU (znów wykorzystana funkcja ASCII jednostki DU). Sterowanie start/stop wykorzystuje klawisze dotykowe na ekranie DU. Po wybraniu trybu sygnał odebrany przez DU natychmiast aktywizuje jeden z następujących przekaźników pomocniczych:

Przełącznik obrotowy:

pozycja 1 "Step" - operacja krokowa: wejście DU I0, bitowe urządzenie sterujące M32

pozycja 2 "Cycle" - praca pojedynczego cyklu: wejście DU I1, bitowe urządzenie sterujące M33

pozycja 3 "Auto" - praca automatyczna: wejście DU I2, bitowe urządzenie sterujące M34

Przyporządkowanie klawiszy dla ekranu DU powyżej:

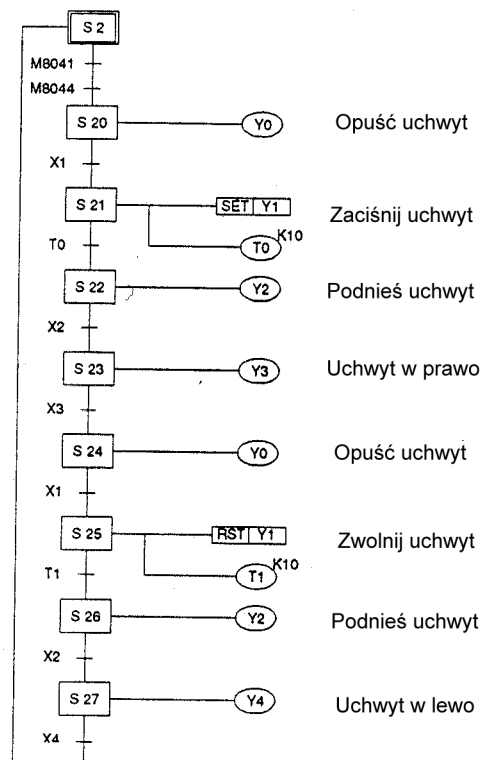
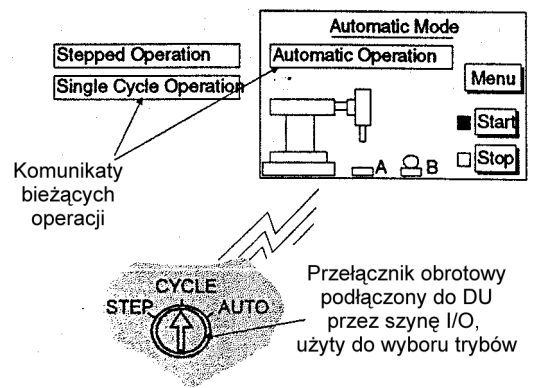
Start = M36

Stop = M37

Program pracuje w trzech trybach jak pokazano obok. Jak już wspomniano, tryb "Step" wymaga naciśnięcia klawisza "Start" w celu uaktywnienia kolejnego bloku STL. Może to być uważane za dodatkowy warunek transferu pomiędzy każdym stanem. Jednakże użytkownik nie musi tego programować, ponieważ instrukcja IST wykonuje to automatycznie.

Tryb "Cycle" przetwarza cały program od kroku S2, aż do momentu ponownego pojawienia się kroku S2. Jeszcze raz instrukcja IST zapewnia jednorazowe wykonanie cyklu po każdym naciśnięciu klawisza "Start".

Tryb "Auto" powoduje ciągłe wykonywanie programu do momentu naciśnięcia klawisza "Stop". Zatrzymanie programu nastąpi dopiero po ukończeniu bieżącego cyklu.



Uwagi:

- a) Użytkownicy, wykorzystujący instrukcję IST, powinni pamiętać, że w danej chwili powinien być aktywny tylko jeden tryb operacyjny. W tym przykładzie jest to zapewnione poprzez stosowanie oddzielnych ekranów w jednostce DU dla trybów "Manual" i "Zero return" oraz użycie przełącznika obrotowego dla wyboru podrzędnych trybów automatycznych. Możliwe jest także wykorzystanie przełącznika obrotowego dla wyboru wszystkich trybów.
- b) Dla użytkowników, którzy zamierzają przetestować przedstawiony przykład na symulatorze, tzn. bez użycia jednostki dostępu do danych, zamieszczono odpowiednie zmiany w listingu programu na końcu sekcji. Można też wykorzystać program w oryginalnej postaci z wszystkimi warunkami wejściowymi poprzez wymuszeniu stanu ON na stykach przy pomocy urządzenia programującego, np. podręcznego programatora, Medoc, itp.
- c) Znaczniki użyte w programie:
 - M8040: transfer stanu wstrzymany
 - **Tryb ręczny:** zawsze ON
 - Tryb powrotu do zera oraz cykliczny:** po podaniu sygnału "Stop" stan bieżący jest zachowany do momentu podania sygnału "Start".
 - Tryb krokowy:** OFF, gdy sygnał "Start" jest ON. W pozostałym czasie M8040 jest w stanie ON, co pozwala na wykonanie operacji pojedynczego kroku STL.
 - Tryb auto:** M8040 jest ustawiany w stan ON po przełączeniu PLC w stan RUN. Zerowany po podaniu sygnału "Start".
 - M8041: start transferu stanu
 - **Tryb ręczny oraz powrotu do zera:** nie używany
 - Tryb krokowy i cykliczny:** aktywny, gdy odbierany jest sygnał "Start"
 - Tryb auto:** ustawiony w stan ON po odebraniu sygnału "Start". Zerowany po odebraniu sygnału "Stop".
 - M8042: impuls startowy
 - aktywny chwilowo po odebraniu sygnału "Start".
 - M8043: powrót do zera zakończony
 - urządzenie aktywowane programowo przez użytkownika
 - M8044: pozycja/warunek Zero
 - urządzenie aktywowane programowo przez użytkownika

Pełny listing programu:

0	LD	X	4	35	STL	S	1	72	STL	S	21
1	AND	X	2	36	LD	M	35	73	SET	Y	1
2	ANI	Y	1	37	RST	M	8043	74	OUT	T	0
3	OUT	M	8044	39	ANI	Y	1			K	10
5	LD	M	8000	40	SET	S	10	77	LD	T	0
6	IST		60	42	STL	S	10	78	SET	S	22
		M	30	43	RST	Y	1	80	STL	S	22
		S	20	44	RST	Y	0	81	OUT	Y	2
		S	27	45	OUT	Y	2	82	LD	X	2
13	STL	S	0	46	LD	X	2	83	SET	S	23
14	LD	M	8044	47	SET	S	11	85	STL	S	23
15	OUT	M	8043	49	STL	S	11	86	OUT	Y	3
17	LD	M	22	50	RST	Y	3	87	LD	X	3
18	SET	Y	1	51	OUT	Y	4	88	SET	S	24
19	LD	M	17	52	LD	X	4	90	STL	S	24
20	RST	Y	1	53	SET	S	12	91	OUT	Y	0
21	LD	M	15	55	STL	S	12	92	LD	X	1
22	ANI	Y	0	56	SET	M	8043	93	SET	S	25
23	OUT	Y	2	58	RST	S	12	95	STL	S	25
24	LD	M	20		(RET)*			96	RST	Y	1
25	ANI	Y	2	60	STL	S	2	97	OUT	T	1
26	OUT	Y	0	61	LD	M	8041			K	10
27	LD	M	16	62	RST	M	8043	100	LD	T	1
28	AND	X	2	64	AND	M	8044	101	SET	S	26
29	ANI	Y	3	65	SET	S	20	103	STL	S	26
30	OUT	Y	4	67	STL	S	20	104	OUT	Y	2
31	LD	M	21	68	OUT	Y	0	105	LD	X	2
32	AND	X	2	69	LD	X	1	106	SET	S	27
33	ANI	Y	4	70	SET	S	21	108	STL	S	27
34	OUT	Y	3					109	OUT	Y	4
	(RET)*							110	LD	X	4
								111	OUT	S	2
								113	RET		
								114	END		

ta instrukcja zwraca przepływ programu do kroku STL S2

*: instrukcje w () nie są niezbędne

Opcje programu:

6	IST		60	17	LD	X	12	27	LD	X	6
		X	20	19	LD	X	7	31	LD	X	11
		S	20	21	LD	X	5	36	LD	X	25
		S	27	24	LD	X	10				

10.13 Wykorzystanie instrukcji PWM do sterowania silnikiem

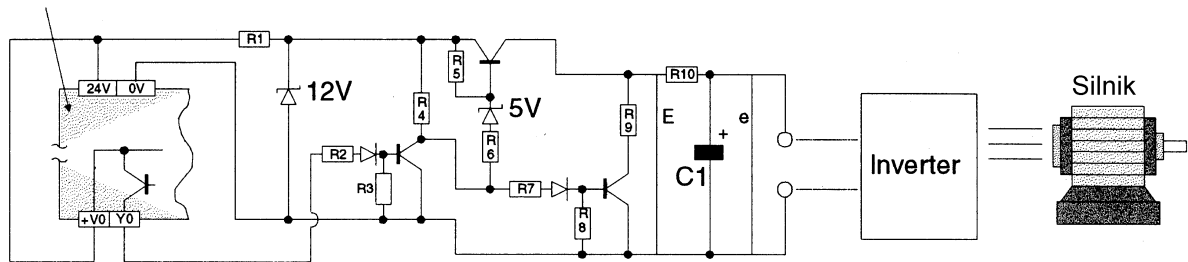
FX_{0(S)}FX_{0N}

FX

FX_(2C)FX_{2N}

Instrukcja PWM może być używana bezpośrednio z inwerterem (falownikiem) do sterowania silnikiem. W takiej konfiguracji należy zastosować pokazany niżej układ, pomiędzy wyjściem instrukcji PWM a końcówkami wejściowymi inwertera.

Sterownik programowalny



Oznaczenia:

R1 - 510Ω (1/2 W)

R2 - 3,3 kΩ (1/2W)

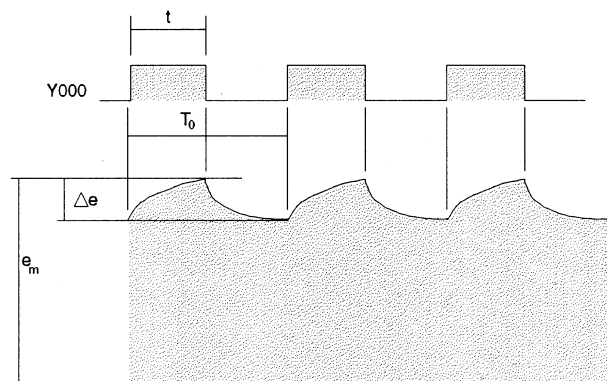
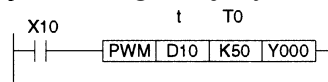
R3 do R8 - 1 kΩ (1/4W)

R9 - 22 Ω (1/4 W)

R10 zmienna, zależna od konfiguracji; w tym przykładzie 1kΩ (1W)

C1 - 470 μF

Uwaga: wartości R10 i C1 zależą od konfiguracji systemu.



Ustalenie parametrów systemu i wartości

Zakłada się, że impedancja wejściowa inwertera jest wysoka. Przyjmując to założenie, wybrano takie wartości R10 i C1 aby otrzymać stałą czasową (ms) w przybliżeniu 10 razy większą od wartości użytej dla T0 w instrukcji PWM:

$$\tau = R10 \text{ (k}\Omega\text{)} \times C1 \text{ (}\mu\text{F)}$$

Podczas tych obliczeń wartość R10 musi być znacznie większa od wartości R9. W tym przykładzie R9 jest równe 22 Ω, podczas gdy R10 - 1 kΩ. Proporcja wynosi w przybliżeniu 1:50.

Maksymalne napięcie wyjściowe (podane na inwerter) włączając w to napięcie pulsacji, można obliczyć, korzystając z równania:

$$e_m = E \frac{t}{T_0}$$

gdzie:

e_m = maksymalne napięcie wyjściowe

E = impulsowe (fala prostokątna) napięcie wyjściowe (zobacz układ na poprzedniej stronie)

t = czas trwania impulsu PWM (zobacz poprzednią stronę)

T_0 = okres impulsu PWM (zobacz poprzednią stronę)

Średnie napięcie wyjściowe (podane na inwerter) włączając w to napięcie pulsacji, można obliczyć, korzystając z równania:

$$\frac{\Delta e}{e} = \frac{T_0 - t}{\tau} \leq \frac{T_0}{\tau}$$

gdzie:

Δe = wartość napięcia pulsacji

e = pulsujące napięcie wyjściowe

t = czas trwania impulsu PWM

T_0 = okres impulsu PWM

τ = stała czasowa uśredniającego układu RC (tu R10, C1)

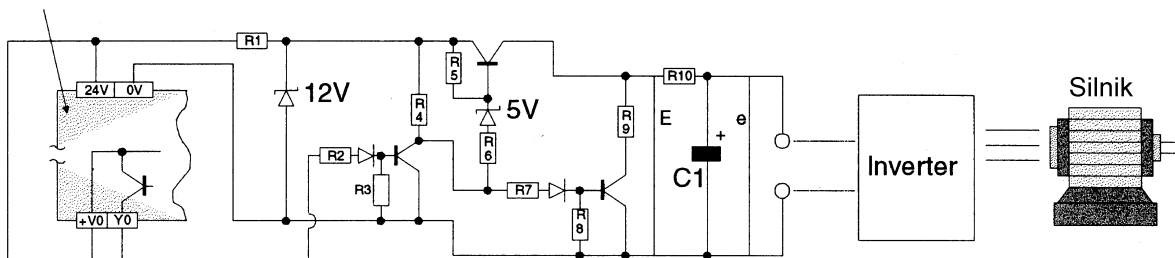
Szczegóły na poprzedniej stronie.

Działanie:

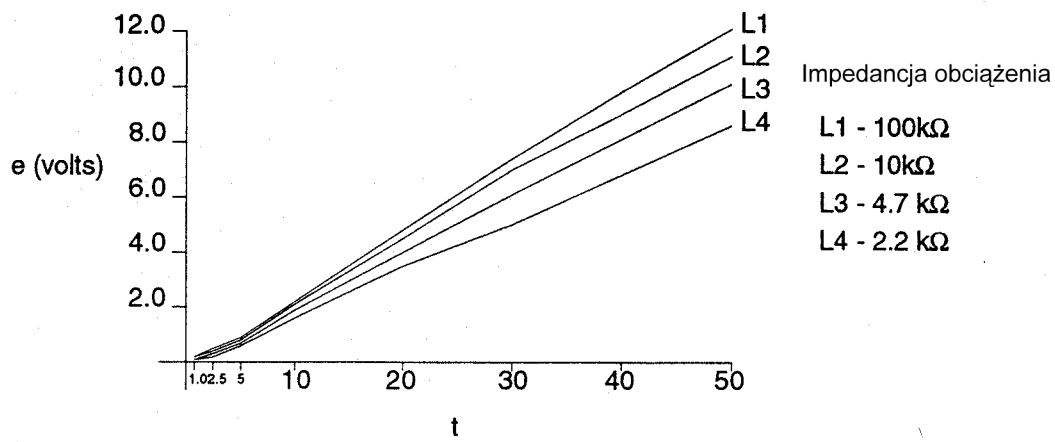
Po skonfigurowaniu systemu i podłączeniu układu uśredniającego, prędkość obrotowa silnika może być zmieniana poprzez dopasowanie wartości 't' instrukcji PWM. Im większa wartość 't', tym większa prędkość obrotowa silnika. Należy jednak pamiętać o tym, że im szybsze zmiany sygnału wyjściowego, tym większe wahania napięcia. Z drugiej strony wolne zmiany sygnału wyjściowego powodują mniejszy efekt uśredniający. Szybkość zmian sygnału zależy od pojemności C1. Duża wartość C1 powoduje mniejszy efekt uśredniający w długim okresie czasu.

Następujące charakterystyki były wzięte pod uwagę przy testowaniu układu poniżej:

Sterownik programowalny



Instrukcja PWM ustawia T_0 na K50. Wartość t oraz impedancja obciążenia była zmieniana, przez co uzyskano następującą charakterystykę



Opóźnienie T_0 także wpływa na wahania napięcia. Widać to na następnym zbiorze danych testowych:

Ustawienie parametru PWM			Zmierzone odchylenie napięcia
t	T_0	t/T_0	
100	200	0,5	1,27V
50	100		668 mV
25	50		350 mV
10	20		154 mV
5	10		82 mV

Zachowanie się układu opisanego powyżej jest podobne do układu opisanego wcześniej

10.14 Techniki programowania. PID

FX_{0(S)}FX_{0N}

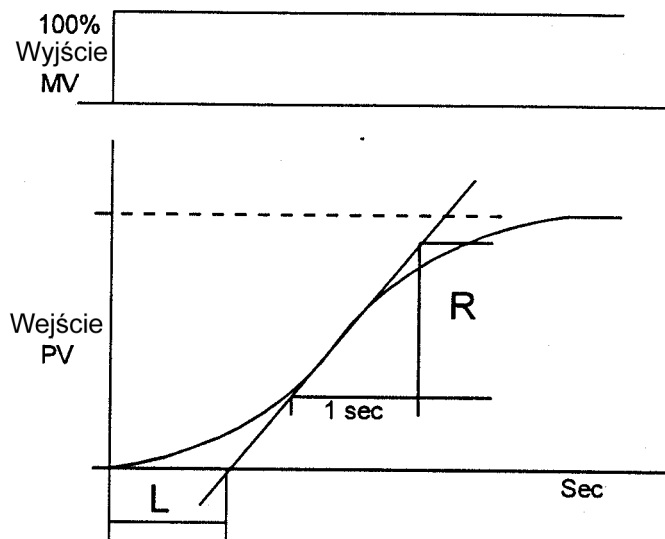
FX

FX_(2C)FX_{2N}

10.14.1 Ustawienie parametrów początkowych PID.

Metoda obliczania najlepszego punktu startowego dla Kp, Ti i Td

Aby znaleźć odpowiednie wartości niektórych parametrów, używanych przez PID, niezbędna jest znajomość odpowiedzi systemu. Najprostszym sposobem na wyznaczenie odpowiedzi systemu jest ustawienie wyjścia (MV) na znaną wartość (100%, o ile to możliwe) i monitorowanie wartości systemu (PV) aż do ustalenia poziomu. Zwykle odpowiedź ma kształt krzywej pokazanej poniżej.



Naniesienie na tym wykresie prostej (stycznej) dopasowanej do prostoliniowego odcinka krzywej wyznacza następujące wartości:

R - współczynnik odpowiedzi; zmiana PV w czasie 1 s na stycznej

L - czas opóźnienia systemu; punkt przecięcia stycznej z osią współrzędnych.

Wykorzystując te czynniki, następująca tabela podaje odpowiednie wartości dla Kp, Ti i Td

	Kp [1/100 %]	Ti [s]	Td [s]	
tylko P	1/RL	0	0	zalecane ustawienie w zakresie roboczym
3,1415	0,9/RL	3,3L	0	
PID	1,2/RL	2L	0,5L	

Jeżeli pobudzenie 100% nie jest możliwe (możliwe uszkodzenie maszyny), wartości mniejsze (np. 50%) są też wystarczające. W takim przypadku wartość Kp należy pomnożyć przez 50% (tzn. 0,5)

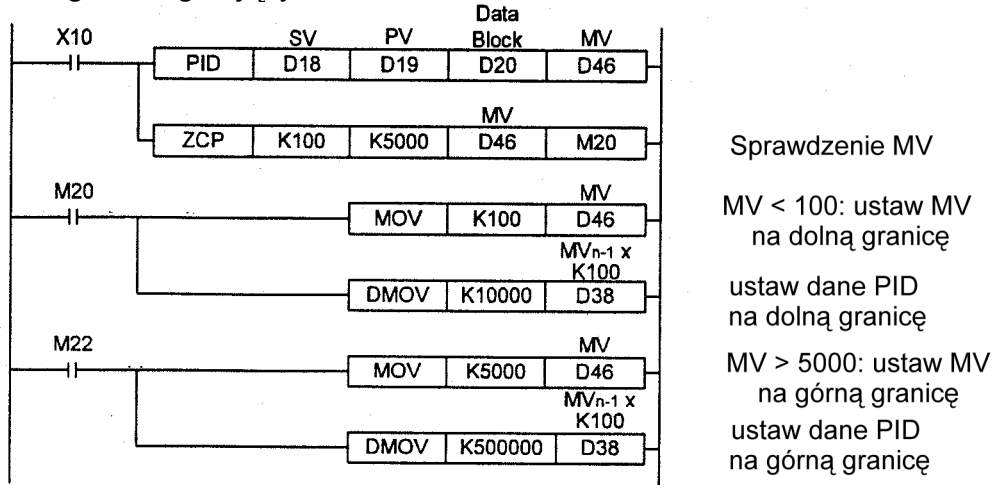


Uwaga: Ts powinno być zawsze mniejsze niż Ti i Td. Dodatkowo, Ts powinno być pomnożone przez czas przebiegu programu.

10.14.2 Utrzymywanie MV w zadanym zakresie.

W bloku danych S3₊₁₈ i S3₊₁₉ rejestrów zarezerwowanych dla PID należy uformować urządzenie podwójnego słowa, które zawiera poprzednią wartość MV x K100. Wykorzystuje je poniższy program do utrzymania kontroli nad M, gdy przekroczone zostaną limity operacyjne.

Przykład: Program regulujący MV w zakresie K100 do K5000



Jeżeli do przechowywania wartości ograniczających służą rejestry danych, możliwe jest użycie instrukcji MUL zamiast DMOV. Np. gdy D50 jest ograniczeniem górnym wystarczy użyć MUL D50 K100 D38, a ponieważ wynik instrukcji MUL jest podwójnym słowem, nie trzeba używać DMUL.

Zerowanie (S3₊₁₉, S3₊₁₈) zabezpiecza przed przeregulowaniem, które pojawia się, gdy tylko MV ulegnie zmianie.

10.14.3 Zmiana sterowania ręczna/automatyczna

Aby móc przełączać sterowanie automatyczne (PID) na ręczne i odwrotnie należy w procesie PID wykonać "śledzenie ręczne". Pomimo tego, że instrukcja PID sterownika FX nie zawiera możliwości śledzenia ręcznego, istnieją dwie metody, które można użyć do przełączania sterowania automatycznego na ręczne i odwrotnie bez żadnych kłopotów.

Aby zrozumieć powód użycia dwóch metod, niezbędne jest zapoznanie się z następującą uwagą. Instrukcja PID ustawia wartości początkowe wyjścia w oparciu o dane początkowe w rejestrze wyjściowym. Po włączeniu instrukcji PID możliwe jest wykonanie tylko części P, w następnym czytaniu części PI i w trzecim - wykonanie pełnego PID. Przy pierwszym czytaniu bieżąca wartość rejestru wyjściowego jest traktowana jako wartość początkowa, zanim zostanie dokonana korekta. Następnie użyta jest poprzednia wartość wyjściowa (zapamiętana w S3₊₁₈, S3₊₁₉).

Metoda 1

Zaleca się, aby, jeśli pożądanym jest przełączanie trybów ręczny/auto, instrukcja PID była wyłączona podczas operacji ręcznych, a wartość w rejestrze MV była kontrolowana przez operatora. Po powrocie do trybu auto i włączeniu instrukcji PID pobierze ona nową wartość i dokona stosownej korekty. Po 3 odczytach będzie działać pełne sterowanie PID i proces szybko znajdzie się pod kontrolą. (Zakłada się, że sterowanie ręczne nie spowodowało zbyt wielkiego oddalenia od punktu ustawienia).

Metoda 2

Podczas operacji ręcznych instrukcja PID jest wykonywana, ale wartość MV jest ignorowana; tą wartością steruje operator. Aby zabezpieczyć instrukcję PID przed wyjściem spod kontroli, wartość MV ustawiana przez operatora powinna być zapisana w rejestrach MV_{n-1} bloku danych PID, w taki sam sposób, jak wcześniej zakres sterujący MV, tzn. S3₊₁₈, S3₊₁₉ ustawić na MV x 100). Po przełączeniu na sterowanie PID wartości wewnętrzne instrukcji PID są już ustawione i pełna operacja PID startuje natychmiast.

10.14.4 Wykorzystanie sygnałów alarmowych PID

Blok danych zawiera cztery wartości alarmowe. Ustawione są na maksymalną dodatnią i ujemną zmianę, jaka może się pojawić w MV i PV. Sygnały alarmowe PID są wykorzystywane do ostrzegania, że system wymyka się spod kontroli.

Po "zimnym" uruchomieniu systemu zwykle nie jest pożądane uwzględnianie członu D w obliczeniach; zmiany PV są duże i człon D powoduje zbyt wielkie korekty. Zatem, gdy system nagle startuje od SP wtedy czasami użycie D może pogorszyć regulację.

Dysponując znacznikiem "alarmu" dla zmian PV i MV możliwe jest monitorowanie stanu systemu i dostosowanie parametrów PID do odpowiednich ustawień.

Gdy system znajduje się blisko SP, zmiany PV (i MV) powinny być minimalne. W tej sytuacji korzystne jest wykorzystanie członu D aby utrzymywać system blisko SP (pożądane jest pełne PID).

Jednakże, jeśli zmieniają się warunki (np. otwarcie drzwi chłodziarki, dodanie substancji do mieszaniny, zimny start itd.) system będzie reagować. W niektórych przypadkach (zwłaszcza zimnego startu) reakcje członu D są zbyt wielkie, by były użyteczne (lepsze byłoby PI lub czasem samo P). W takich przypadkach można wykorzystać znaczniki alarmowe do zmiany na sterowanie PI, dopóki system nie wróci do stanu na tyle stabilnego, w którym można wykorzystywać pełne sterowanie PID.

10.14.5 Inne wskazówki programowania PID

- Zaleca się, aby odczytać wartość wejściową zanim zostanie aktywowana instrukcja PID. W przeciwnym razie, PID zinterpretuje to jako wielką zmianę od 0 do pierwszej wartości i obliczy to tak, jak gdyby pojawił się wielki błąd.
- Instrukcja PID nie jest wykonywana pod przerwaniem. Jest zależna od czasu przebiegu programu i w związku z tym próbkowanie nie wystąpi szybciej niż wynika to z czasu przebiegu programu. Zaleca się, aby Ts był ustawiany jako wielokrotność czasu przebiegu programu.
- Aby zminimalizować błędy czasowe, zaleca się wykorzystywać stały czas przebiegu programu.
- Aby zwiększyć współczynnik próbkowania można umieścić instrukcję PID w procedurze przerwania od timera.
- Lepiej jest wykonywać tylko człon P, dopóki wartość wejściowa (PV) nie osiągnie zakresu roboczego.
- Podczas ustawiania dobrze jest monitorować wejście i wyjście instrukcji PID i sprawdzać, czy ich wartości mieszczą się w oczekiwanych zakresach.
- Jeżeli system PID nie działa poprawnie należy sprawdzić znaczniki błędów PID (rejestr D8067).

10.15 Dodatkowe funkcje PID

FX _{0(S)}	FX _{0N}	FX	FX _(2C)	FX _{2N}
--------------------	------------------	----	--------------------	------------------

Poniższa tablica zawiera dodatkowe parametry dostępne w jednostkach FX2N. Są to:

- S₃+1 bit 4: znacznik operacji samodostrajania
- S₃+1 bit 5: znacznik granicy zakresu wartości wyjściowej
- S₃+22: granica górna wartości wyjściowej
- S₃+23: granica dolna wartości wyjściowej

Parametr S ₃ +☆	Nazwa parametru/funkcja	Opis		Zakres ustawień
S ₃ +1	akcja-reakcja sterowania kierunku i alarmu	b0		brak
		b1	bieżąca wartość (S2) zmienia alarm OFF(0)/ON(1)	
		b2	wartość wyjściowa zmienia alarm OFF(0)/ON(1)	
		b3	zarezerwowane	
		b4	aktywacja samodostrajania	
		b5	wartość wyjściowa (D) zakresu ograniczeń OFF(0)/ON(1)	
		b6-15	zarezerwowane	
S ₃ +22	Wartość wyjściowa, maks. zmiana dodatnia	aktywny gdy S ₃ +1, b2 jest ON	Ograniczenie maks. zdefiniowane przez użytkownika dla ilości dodatnich zmian, które mogą się pojawić w jednym przebiegu PID. Jeśli wartość wyjściowa (D) przekracza ograniczenie, bit S ₃ +24, b2 jest ustawiany	0 do 32767
	Wartość wyjściowa, ograniczenie górne	aktywny, gdy S ₃ +1, b5 jest ON	Górne ograniczenie wartości wyjściowej (D), definiowane przez użytkownika. Podczas działania instrukcja PID uniemożliwia przekroczenie ograniczenia	-32768 do 32767
S ₃ +23	Wartość wyjściowa, maks. zmiana ujemna	aktywny, gdy S ₃ +1, b5 jest ON	Ograniczenie maks. zdefiniowane przez użytkownika dla ilości ujemnych zmian, które mogą się pojawić w jednym przebiegu PID. Jeśli wartość wyjściowa (D) przekracza ograniczenie, bit S ₃ +24, b3 jest ustawiany	0 do 32767
	Wartość wyjściowa, ograniczenie dolne	aktywny, gdy S ₃ +1, b5 jest ON	Górne ograniczenie wartości wyjściowej (D), definiowane przez użytkownika. Podczas działania instrukcja PID uniemożliwia przekroczenie ograniczenia	-32768 do 32767

Pełna lista innych parametrów znajduje się na stronie 5-105



Uwaga: S₃+1 b2 i b5 nie mogą być jednocześnie aktywne.

10.15.1. Sterowanie zakresem wartości wyjściowej

Gdy bit 5 parametru S₃+1 jest w stanie ON, powoduje, że S₃+22 i S₃+23 stanowią limity wartości wyjściowej (OV lub MV)

Ta właściwość utrzymuje wartość wyjściową w określonych granicach; w efekcie automatycznie wykonywana jest taka sama operacja, jaką opisano w sekcji 10.14.2.

10.15.2 Operacja samodostrajania

FX_{0(S)}FX_{0N}

FX

FX_(2C)FX_{2N}

Wykorzystywany jest proces podobny do opisanego w sekcji 10.14.1 w celu automatycznego wyliczenia:

- kierunku procesu; dodatni lub ujemny (S₃+1, bit 0)
- stałej wzmacnienia proporcjonalnego; K_p (S₃+3)
- stałej czasu całkowania; T_i (S₃+4)
- stałej czasu różniczkowania; T_d (S₃+6)

Wartość początkowa wartości wyjściowej MV (D) jest używana jako wartość krokowa dla tej operacji. Użytkownik powinien ustawić tę wartość w zakresie 50-100% maksymalnej wartości systemu.

Dodatkowo, przed uruchomieniem, powinny być ustawione wartości, które nie są ustawiane przez operację samodostrajania, tj. czas próbkowania, T_s (S₃+0); stała czasowa filtru wejściowego alfa, (S₃+2) itd. oraz SV.



Uwaga: Aby operacja samodostrajania działała poprawnie, różnica pomiędzy wartością bieżącą a nastawą musi być większa od 150.



Uwaga: Podczas operacji samodostrajania czas próbkowania powinien być większy od 1 s (1000 ms).
Zaleca się, by czas próbkowania był wielokrotnie większy od czasu przebiegu programu.

Ustawienie bitu 4 w S₃+1 uruchamia proces samodostrajania.

System utrzymuje wartość wyjściową (D) jako wartość początkową, monitorując wartość procesu, dopóki nie osiągnie ona jednej trzeciej ustawionej wartości. W tym punkcie znacznik samodostrajania (bit 4) jest zerowany i wznawiana jest normalna operacja PID.



Uwaga: System powinien być stabilny przez uruchomieniem operacji samodostrajania.
System niestabilny może doprowadzić operację samodostrajania do generowania błędnych wyników.

1	Wprowadzenie
2	Podstawowe instrukcje programu
3	Programowanie STL
4	Charakterystyka urządzeń
5	Instrukcje specjalne
6	Urządzenia diagnostyczne
7	Czasy wykonania i hierarchia instrukcji
8	Specyfikacja urządzeń sterownika
9	System adresowania urządzeń
10	Wskazówki techniczne
11	Indeks

Spis treści

Indeks	11-1
Tabela znaków ASCII	11-9
Lista instrukcji specjalnych	11-10

Indeks

A

adresowanie bloków funkcji specjalnych	5-9
annunciator – ustawianie, instrukcja ANS	5-47
annunciator – zerowanie, instrukcja ANR	5-47
aplikacja systemowa, przykład	10-8

B

bity, słowa, zapis BCD i szesnastkowy	4-40
---------------------------------------	------

C

czasy wykonania instrukcji, instrukcje podstawowe	7-1
czasy wykonania instrukcji, instrukcje specjalne	7-3
czytanie danych binarnych	4-42
czytanie i obróbka danych z nastawników cyfrowych (SMOV), przykład	10-6

D

dodawanie wartości danych, instrukcja ADD	5-25
dwukrotne użycie cewki	2-5

E

END	2-23
etykieta programu przerywania, patrz przerywania	
etykieta, szczegóły urządzeń i przykłady	4-10

F

F-16NP/NT – blok sieci Net Mini serii F	9-3
F2-30GM – blok wyjść impulsowych serii F	9-5
F2-32RM CAM – blok pozycjonujący serii F	9-4
F2-6A – blok analogowy serii F	9-4
funkcja komunikacji RS (FNC 80)	5-95
FX2-24EI – użycie z blokami serii F	9-2
FX2-40AP/AW, praca równoległa, instrukcja PRUN	5-99
FX-40AP/AW, adapter połączenia równoległego	9-6
FX-8AV – instrukcje sterujące, instrukcja VRRD	5-104
FX-8AV – instrukcje sterujące, instrukcja VRSC	5-104
FX-8AV – zewnętrzne ustawianie wartości danych	4-37
FX-PCS/AT-EE SW, środki ostrożności	3-15

H

hierarchia instrukcji przepływu programu	7-12
--	------

I	
impuls, instrukcje zbocza narastającego i opadającego	2-20
indeksowanie przy wyświetlaniu wielu wartości danych	10-5
instrukcja ABSD	5-70
instrukcja ANB	2-12
instrukcja AND BLOCK	2-12
instrukcja ARWS	5-87
instrukcja BCD, czytanie danych słowowych	4-44
instrukcja BCD, konwersja binarny na BCD	5-22
instrukcja BCD, wyjście BCD (Binary Coded Decimal)	5-22
instrukcja BIN, konwersja BCD na binarny	5-22
instrukcja BIN, wejście BIN	5-22
instrukcja BMOV, przesunięcie bloku danych	5-20
instrukcja BON	5-45
instrukcja CALL, wywołanie podprogramu	5-7
instrukcja CCD, suma kontrolna	5-103
instrukcja CJ, skok warunkowy	5-5
instrukcja CML, uzupełnienie wartości danych	5-19
instrukcja CMP, porównanie pojedynczych wartości danych	5-17
instrukcja DEC, dekrementowanie danej	5-29
instrukcja DECO, dekodowanie wartości danych	5-43
instrukcja DIV, dzielenie wartości danych	5-28
instrukcja DSW, wejście przełącznika cyfrowego	5-83
instrukcja ENCO, kodowanie danych	5-44
instrukcja END	2-23
instrukcja FEND, wymuszenie końca programu	5-11
instrukcja FMOV	5-21
instrukcja FROM, odczyt z bloków funkcyjnych	5-90
instrukcja HKY, klawiatura szesnastkowa	5-82
instrukcja HSCR, zerowanie szybkiego licznika	5-56
instrukcja HSCS, ustawianie szybkiego licznika	5-55
instrukcja HSZ, operacja standardowa (1)	5-57
instrukcja HSZ, połączenie operacji HSZ i PLSY (3)	5-59
instrukcja HSZ, porównanie zakresu szybkiego licznika	5-57
instrukcja HSZ, użycie z tabelą danych (2)	5-57
instrukcja INC, inkrementowanie danych	5-29
instrukcja INCD	5-71
instrukcja IST, sterowanie stanu inicjującego	5-67
instrukcja LOAD, LOAD INVERSE	2-3
instrukcja MEAN	5-46
instrukcja MOV	5-18
instrukcja MTR definiowana przez użytkownika	10-8
instrukcja MTR	5-54
instrukcja MUL, mnożenie danych	5-27
instrukcja NEG, negacja wartości danej	5-31
instrukcja NOP	2-22
instrukcja OR BLOCK	2-11
instrukcja OUT	2-4
instrukcja PLSR	5-63

instrukcja PLSY	5-61
instrukcja PLSY, inicjalizacja dla FX wersja 2.2 lub wcześniejsza	5-61
instrukcja PR	5-89
instrukcja PWM, modulacja szerokości impulsu	5-62
instrukcja RAMP	5-73
instrukcja RCL, rotacja z przeniesieniem w lewo	5-36
instrukcja RCR, rotacja z przeniesieniem w prawo	5-36
instrukcja REF, odświeżanie stanu wejścia/wyjścia	5-53
instrukcja REFF, ustawianie i odświeżanie filtrów	5-53
instrukcja ROL, rotacja wzorca bitowego w lewo	5-35
instrukcja ROR, rotacja wzorca bitowego w prawo	5-35
instrukcja ROTC, sterowanie stołem obrotowym	5-75
instrukcja SEGD, dekodery 7-segmentowy	5-84
instrukcja SEGL, sterowanie 7-segmentowe z zatraskiem	5-85
instrukcja SER, wyszukiwanie danych	5-69
instrukcja SFRD, odczyt danych FIFO	5-40
instrukcja SFTL, przesunięcie bitowe w lewo	5-37
instrukcja SFTR, przesunięcie bitowe w prawo	5-37
instrukcja SFWR, zapis danych FIFO	5-39
instrukcja słowowa AND	5-30
instrukcja słowowa OR	5-30
instrukcja słowowa XOR	5-31
instrukcja SMOV	5-18
instrukcja SMOV, przesuwanie danych BCD	5-19
instrukcja SMOV, przesuwanie danych dziesiętnych	5-18
instrukcja SORT (FNC 69)	5-77
instrukcja SPD	5-60
instrukcja SQR, pierwiatek kwadratowy	5-48
instrukcja SRET, powrót z podprogramu	5-8
instrukcja STMR, timer specjalny	5-72
instrukcja SUB, odejmowanie wartości danych	5-26
instrukcja SUM, sumowanie aktywnych bitów	5-45
instrukcja TKY, klawiatura 10-klawiszowa	5-81
instrukcja TO, zapis do bloków specjalnych	5-91
instrukcja TTMR	5-72
instrukcja WDT, odświeżanie timera watchdog	5-12
instrukcja WSFL, przesuw słowa w lewo	5-38
instrukcja WSFR, przesuw słowa w prawo	5-38
instrukcja XCH, zamiana bajtów danej	5-21
instrukcja XCH, zamiana wartości danych	5-21
instrukcja ZCP, porównanie danych z zakresem	5-17
instrukcja ZRST, zerowanie zakresu urządzeń	5-43
instrukcje AND i AND INVERSE	2-6
instrukcje AND, ANI	2-6
instrukcje FOR, NEXT, pętle FOR-NEXT	5-13
instrukcje MASTER CONTROL i MASTER CONTROL RESET	2-15
instrukcje nieodpowiednie do wykorzystania w jednostkach z wejściami 110V AC	7-16
instrukcje OR, OR INVERSE	2-7
instrukcje SET i RESET	2-17
instrukcje specjalne	5-1
instrukcje specjalne, które mogą być użyte tylko raz	7-16

instrukcje sterujące F-16NP/NT – FX2-24EI, instrukcja MNET	5-113
instrukcje sterujące F-16NP/NT – FX2-24EI, sterowanie mini siecią Melsec	5-113
instrukcje sterujące F2-30GM – FX2-24EI, instrukcja BLOCK, zapis bloku	5-117
instrukcje sterujące F2-30GM – FX2-24EI, instrukcja MCDE	5-118
instrukcje sterujące F2-32RM – FX2-24EI, monitor RM, instrukcja RMMN	5-116
instrukcje sterujące F2-32RM – FX2-24EI, start RM, instrukcja RMST	5-114
instrukcje sterujące F2-32RM – FX2-24EI, status odczytu RM, instrukcja RMRD	5-116
instrukcje sterujące F2-32RM – FX2-24EI, zapis RM, instrukcja RMWR	5-115
instrukcje sterujące F2-6AE – FX2-24EI, odczyt danych analogowych, instr. ANRD	5-113
instrukcje sterujące F2-6AE – FX2-24EI, zapis danych analogowych, instr. ANWR	5-114
interpretacja danych słownych	4-42
K	
kasety zegara czasu rzeczywistego	9-7
kody ASCII (Alpha to ASCII), instrukcja ASCII	5-88
kody błędów, komunikacji (D8062-D8063)	6-15
kody błędów, operacji (D8067)	6-19
kody błędów, parametrów (D8064)	6-16
kody błędów, sprzętowe (D8061)	6-15
kody błędów, syntaktyczne (D8065)	6-16
kody błędów, układu (D8066)	6-17; 6-18
kody znaków ASCII	11-9
konwersja ASCII na HEX przy użyciu instrukcji HEX (FNC 83)	5-102
konwersja HEX do ASCII przy użyciu instrukcji ASCII (FNC 82)	5-101
L	
LD, LDI	2-3
liczniki 16-bitowe	4-20
liczniki 32-bitowe, dwukierunkowe	4-21
liczniki cykliczne	4-21
liczniki podstawowe	2-18
liczniki szybkie, 1-fazowe, programowy start i zerowanie	4-29
liczniki szybkie, 1-fazowe, wejścia zerujące i startujące	4-30
liczniki szybkie, 2-fazowe, dwukierunkowe	4-31
liczniki szybkie, A/B-fazowe	4-32
liczniki szybkie, dostępność na sterownikach serii FX	4-25
liczniki szybkie, dostępność na sterownikach serii FX ₀ , FX _{0S} i FX _{0N}	4-24
liczniki szybkie, dostępność na sterownikach serii FX _{2N}	4-28
liczniki szybkie, operacje podstawowe	4-23
liczniki szybkie, słownik i podsumowanie	4-22
liczniki szybkie, szybkość zliczania	4-26
liczniki, szczegóły urządzeń i przykłady	4-19
lista instrukcji specjalnych	11-10
logika dodatnia i ujemna	5-86
M	
MC, MCR	2-15
MPS, MRD, MPP	2-13

N	
narzędzia programowania	1-3
NOP	2-22
notacja naukowa, format numeryczny	4-47
O	
OR, ORI	2-7
ORB	2-11
OUT, odmiany timerów i liczników	2-4
P	
P, patrz etykiety	
PID, sterowanie	5-108
PID, sterowanie, instrukcja specjalna PID, FNC 88	5-105
PID, sterowanie, parametry początkowe	5-106
PID, sterowanie, samodostrajanie	10-22
PID, sterowanie, technika programowania	10-18
PLS, PLF	2-20
pomiar impulsów wysokiej częstotliwości, wykorzystanie przerwań i M8099, D8099	10-7
pomiar impulsów wysokiej częstotliwości, wykorzystanie przerwań i timera 1 ms	10-6
potencjometry zewnętrzne – FX ₀ , FX _{0S} i FX _{0N}	4-37
program, co jest potrzebne do zaprogramowania PLC	1-3
program, czym jest	2-1
program, formaty: lista instrukcji, schemat drabinkowy, STL/SFC	2-1
program, jak czytać schemat drabinkowy	2-2
program, przebieg	2-23
przełączniki pomocnicze ogólne informacje o urządzeniach diagnostycznych	4-5
przełączniki pomocnicze ogólne zastosowanie	4-3
przełączniki pomocnicze przykłady i szczegóły urządzeń	4-3
przełączniki pomocnicze z podtrzymaniem/zatrząskowe	4-4
przerwania, blokada pojedynczych przerwań	4-13
przerwania, instrukcje IRET, EI, DI	5-9
przerwania, programy obsługi wyzwalane timerem	4-12
przerwania, programy obsługi wyzwalane wejściem	4-12
przerwania, przykład użycia	10-6
przerwania, szczegóły urządzeń i przykłady	4-11
przerwanie timera, przykład użycia	10-8
przetwarzanie wsadowe	7-14
przyporządkowanie numerów bloków funkcji specjalnych	9-1
przyporządkowanie urządzeń systemowych	9-1
R	
rejstry danych, ogólne działanie	4-34
rejstry diagnostyczne, ogólny opis	4-35
rejstry indeksowe, modyfikacja stałej	4-39
rejstry indeksowe, niewłaściwe użycie modyfikatorów	4-39
rejstry indeksowe, ogólne zastosowanie	4-38
rejstry indeksowe, szczegóły urządzeń i przykłady	4-38
rejstry indeksowe, wielokrotne użycie	4-39
rejstry plikowe FX i FX _{0N}	4-36

rejstry z podtrzymaniem/zatraskowe	4-35
rejstry, szczegóły urządzeń i przykłady	4-33
rejstry, ustawiane ręcznie/zewnętrznie	4-37
S	
schemat drabinkowy	3-1
schemat drabinkowy, czym jest STL, SFC i IEC 1131 cz.3	3-1
schemat drabinkowy, jak rozpocząć i zakończyć program STL	3-3
schemat drabinkowy, łączenie pierwszego stanu	3-11
schemat drabinkowy, łączenie wielokrotnych stanów	3-13
schemat drabinkowy, ogólne reguły rozgałęziania STL	3-14
schemat drabinkowy, ograniczenia niektórych instrukcji	3-10
schemat drabinkowy, prosty przepływ STL	3-16
schemat drabinkowy, reguły pisania programów STL	3-7
schemat drabinkowy, rozgałęzienie wyboru STL	3-18
SET, RST	2-17
słowa szesnastkowe, czytanie	4-43
specyfikacja urządzeń sterownika FX, wersje CPU od 2.0 do 3.06	8-4
specyfikacja urządzeń sterownika FX, wersje CPU od 3.07	8-6
specyfikacja urządzeń sterownika FX ₀ i FX _{0S}	8-1
specyfikacja urządzeń sterownika FX _{0N}	8-2
specyfikacja urządzeń sterownika FX _{2C}	8-6
specyfikacja urządzeń sterownika FX _{2N}	8-8
stałe dziesiętne (K)	4-14
stałe szesnastkowe (H)	4-14
stały przebieg programu, jak programować, przykład	10-4
stan zmienny, instrukcja ALT	5-73
stany zmienne przy użyciu instrukcji ALT, przykład	10-4
sterowanie IST i STL, przykład programu	10-8
sterownik programowalny, czym jest	1-3
STL, zobacz schemat drabinkowy	
T	
timery i liczniki, wyjścia i zerowanie	2-18
timery, dokładność	4-18
timery, operacje ogólne	4-16
timery, podstawowe	2-18
timery, szczegóły urządzeń i przykłady	4-15
timery, używane w przerwaniach i podprogramach CALL	4-18
timery, z pamięcią	4-17
timery, z wybieranym zakresem	4-16
U	
układ uśredniający do użycia z inwerterem	10-55
układy wielokrotnych wyjść	2-13
urządzenia bitowe	4-40
urządzenia bitowe	4-41
urządzenia C, patrz liczniki	
urządzenia D, patrz rejstry	
urządzenia diagnostyczne, status PLC (M8000-9 i D8000-9)	6-2

urządzenia diagnostyczne, sterowanie kierunkiem zliczania (M8200-34 i D8200-34)	6-14
urządzenia diagnostyczne, sterowanie połączeniem (M8070-99 i D8070-99)	6-9
urządzenia diagnostyczne, sterowanie przerwami (M8050-59 i D8050-59)	6-7
urządzenia diagnostyczne, tryb pracy PLC (M8030-39 i D8030-39)	6-5
urządzenia diagnostyczne, wykrywające błędy (M8060-69 i D8060-69)	6-8
urządzenia diagnostyczne, zegarowe (M8010-19 i D8010-19)	6-3
urządzenia diagnostyczne, znaczniki annunciatora/STL (M8040-49 i D8040-49)	6-6
urządzenia diagnostyczne, znaczniki operacji (M8020-29 i D8020-29)	6-4
urządzenia diagnostyczne, znaczniki szybkich liczników (M8235-55 i D8235-55)	6-14
urządzenia diagnostyczne, zobacz pomocnicze (M8100-19 i D8100-19)	
urządzenia i instrukcje podstawowe	2-1
urządzenia M, patrz przekaźniki pomocnicze	
urządzenia podstawowe przegląd	2-1
urządzenia podstawowe X, Y, T, C, M, S	2-1
urządzenia podtrzymywane bateryjnie/zatrząskowe – przykłady	10-5
urządzenia S, patrz znaczniki stanu	
urządzenia słowowe	4-42
urządzenia T, patrz timery	
urządzenia V, patrz rejestry indeksowe	
urządzenia X, patrz wejścia	
urządzenia Y, patrz wyjścia	
urządzenia Z, patrz rejestry indeksowe	
uzupełnienie do 2 – wyjaśnienie	4-45
W	
wartość H – zobacz stałe	
wartość K zobacz stałe	
wejście, szczegóły urządzeń i przykłady	4-1
wskazówki zaawansowanego programowania	10-1
wyjście, szczegóły urządzeń i przykłady	4-2
wykorzystanie instrukcji PWM do sterowania silnikiem	10-15
wymuszenie trybu RUN (M8035/36/37) – przykład z przyciskiem	10-2
wymuszenie trybu RUN (M8035/36/37) – zdalne sterowanie	10-3
wyświetlacze 7-segmentowe	5-85
Z	
zamiana formatów danych, zmiennoprzecinkowy na naukowy (FNC 18)	5-22
zamiana formatów danych, zmiennoprzecinkowy na naukowy (FNC 19)	5-22
zanik zasilania jednostek FX DC, środki ostrożności	10-1
zapis zmiennoprzecinkowy i notacja naukowa	4-46
zmiennoprzecinkowy zapis	4-48
zmienny przecinek, aplikacja – podsumowanie	4-49
zmienny przecinek, instrukcja FLT,	5-49
znacznik wykonania operacji, M8029	10-7
znaczniki stanu, ogólne zastosowanie	4-6
znaczniki stanu, szczegóły urządzenia i przykłady	4-6
znaczniki stanu, użycie jako numerów kroków STL	4-8
znaczniki stanu, używanie annunciatorów	4-9
znaczniki stanu, z podtrzymaniem/zatrząskowe	4-7

Kody znaków ASCII

Tabela kodów ASCII (HEX)		Starszy bit						
		1	2	3	4	5	6	7
Młodszy bit	0	niedostępne	(SP)	0	@	P	@	p
	1		!	1	A	Q	a	q
	2		"	2	B	R	b	r
	3		#	3	C	S	c	s
	4		\$	4	D	T	d	t
	5		%	5	E	U	e	u
	6		&	6	F	V	f	v
	7		'	7	G	W	g	w
	8		(8	H	X	h	x
	9)	9	I	Y	i	y
	A		*	:	J	Z	j	z
	B		+	;	K	[k	{
	C		,	<	L		l	
	D		-	=	M]	m	}
	E		.	>	N	(SP)	n	~
	F		/	?	O	_	o	CR

Uwaga:

(SP) = Spacja

CR = Powrót karetki (Return)

Lista instrukcji specjalnych

FX _{2N}			
FX (CPU wersja >=3.07) i FX _{2C}			
FX (CPU 2.0 do 3.06)			
FX _{0N}			
FX ₀ , FX _{0S}			
Mnemo- nik	FNC	Strona	
A	ABSD	62	5-70
	ADD	20	5-25
	ALT	66	5-73
	AND	232-238	5-150
	ANR	47	5-47
	ANRD	91	5-113
	ANS	46	5-47
	ANWR	92	5-114
	ARWS	75	5-87
	ASC	76	5-88
	ASCI	82	5-101
	BCD	18	5-22
B	BIN	19	5-22
	BLK	97	5-117
	BMOV	15	5-20
	BON	44	5-45
C	CALL	01	5-7
	CCD	84	5-103
	CJ	00	5-5
	CML	14	5-19
	CMP	10	5-17
	COS	131	5-17
D	DEC	25	5-29
	DECO	41	5-43
	DI	05	5-9
	DIV	23	5-28
	DSW	72	5-83
E	EADD	120	5-123
	EBCD	118	5-17
	EBIN	119	5-122
	ECMP	110	5-121
	EDIV	123	5-125
	EI	04	5-9
	EMUL	122	5-17
	ENCO	42	5-44
	ESQR	127	5-125
	ESUB	121	5-124
	EZCP	111	5-121
F	FEND	06	5-11
	FLT	49	5-49
	FMOV	16	5-21
	FOR	08	5-13
	FROM	78	5-90

FX _{2N}			
FX (CPU wersja >=3.07) i FX _{2C}			
FX (CPU 2.0 do 3.06)			
FX _{0N}			
FX ₀ , FX _{0S}			
Mnemo- nik	FNC	Strona	
G	GBIN	171	5-145
	GRY	170	5-145
	HEX	83	5-102
H	HKY	71	5-82
	HSCR	54	5-56
	HSCS	53	5-55
	HSZ	55	5-57
I	INC	24	5-29
	INCD	63	5-71
	INT	129	5-126
	IRET	03	5-9
	IST	60	5-67
L	LD	224-230	5-149
	MCDE	98	5-118
	MEAN	45	5-46
M	MNET	90	5-113
	MOV	12	5-18
	MTR	52	5-54
	MUL	22	5-27
N	NEG	29	5-31
	NEXT	09	5-13
O	OR	240-246	5-151
	PID	88	5-105
	PLSR	59	5-63
P	PLSY	57	5-61
	PR	77	5-89
	PRUN	81	5-99
	PWM	58	5-62
	RAMP	67	5-73
	RCL	33	5-36
	RCR	32	5-36
	REF	50	5-53
	REFF	51	5-53
R	RMMN	96	5-116
	RMRD	95	5-116
	RMST	93	5-114
	RMWR	94	5-115
	ROL	31	5-35
	ROR	30	5-35
	ROTC	68	5-75
	RS	80	5-95

FX _{2N}			
FX (CPU wersja >=3.07) i FX _{2C}			
FX (CPU 2.0 do 3.06)			
FX _{0N}			
FX ₀ , FX _{0S}			
Mnemo- nik	FNC	Strona	
S	SEGD	73	5-84
	SEGL	74	5-85
	SER	61	5-69
	SFRD	39	5-40
	SFTL	35	5-37
	SFTR	34	5-37
	SFWR	38	5-39
	SIN	130	5-129
	SMOV	13	5-18
	SORT	69	5-77
	SPD	56	5-60
	SQR	48	5-48
	SRET	02	5-8
	STMR	65	5-72
	SUB	21	5-26
	SUM	43	5-45
	SWAP	147	5-133
	TADD	162	5-139
	TAN	132	5-130
	TCMP	160	5-137
	TKY	70	5-81
T	TO	79	5-91
	TRD	166	5-141
	TSUB	163	5-140
	TTMR	64	5-72
	TWR	167	5-142
	TZCP	161	5-138
V	VRRD	85	5-104
	VRSC	86	5-104
	WAND	26	5-30
	WDT	07	5-12
W	WOR	27	5-30
	WSFL	37	5-38
	WSFR	36	5-38
	WXOR	28	5-31
X	XCH	17	5-21
Z	ZCP	11	5-17
	ZRST	40	5-43