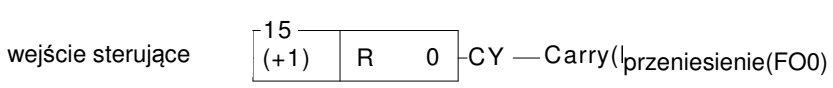
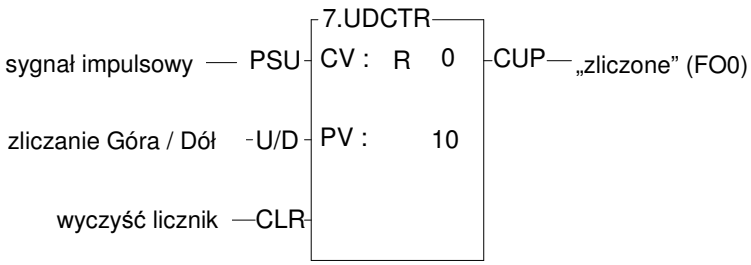
Rozdział 5Opis instrukcji funkcyjnych

5.1 Format instrukcji funkcyjnych

Niniejszy rozdział zawiera szczegóły dotyczące instrukcji funkcyjnych FBs-PLC. Opisy każdej funkcji podzielone zostały na cztery części: wejście sterujące funkcją, numer/nazwa instrukcji, argument i wyjście funkcji. W przypadku zastosowania FP-08, wprowadzenia instrukcji mnemonicznej (poza instrukcjami T, C, SET, RST i SFC, które można wprowadzić bezpośrednio poprzez naciśnięcie przycisku) dokonuje się poprzez wpisanie numeru instrukcji zamiast jej nazwy. Przykład przedstawiony został poniżej.

Schemat drabinkowy	Kod mnemoniczny FP-08
Przykład 1: Instrukcja z 1 wejściem sterującym 	FUN 15 D: R 0
Przykład 2: Instrukcja dla wielu wejśćsterujących 	FUN 7 CV: R 0 PV: 10

Uwaga : Słowa w polach kodu mnemonicznego są komunikatami z FP-08, takimi jak D:, CV:, i Pr: i nie są one wprowadzane przez użytkownika.

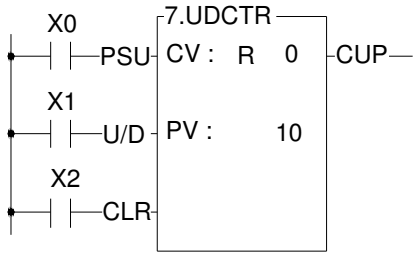
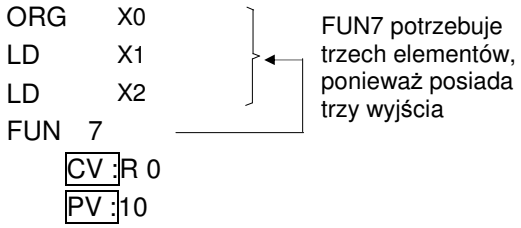
5.1.1 Wejścia sterujące

Poza siedmioma instrukcjami funkcyjnymi nieobsługującymi wejść sterujących, liczba takich wejść dla funkcji mieści się w zakresie od jednego do czterech. Realizacja instrukcji i operacji zależy od wejściowego sygnału sterującego lub kombinacji kilku wejściowych sygnałów sterujących. Oprogramowanie WinProladder umożliwia użytkownikowi realizację skomplikowanych projektów. W oknie programu drabinkowego wyświetlane są wszystkie instrukcje funkcyjne jako bloki oraz skróty oznaczające wejścia, wyjścia, nazwę funkcji oraz nazwy parametrów.

Uwaga: Istnieje siedem instrukcji, dla których wejścia sterujące powinny być bezpośrednio połączone z linią źródłową. Są to instrukcje MCE, SKPE, LBL, RTS, RTI, FOR i NEXT. Więcej szczegółów znajduje się w rozdziale 6 i 7.

Wszystkie wejścia sterujące instrukcji funkcyjnych powinny być łączone za pomocą odpowiednich elementów. W innym przypadkupojawi się błąd syntaktyczny. Jak pokazano na poniższym przykładzie nr 3, instrukcja funkcyjna FUN7 wyposażona jest w trzy wejścia do których podłączone są sygnały sterujące funkcją.

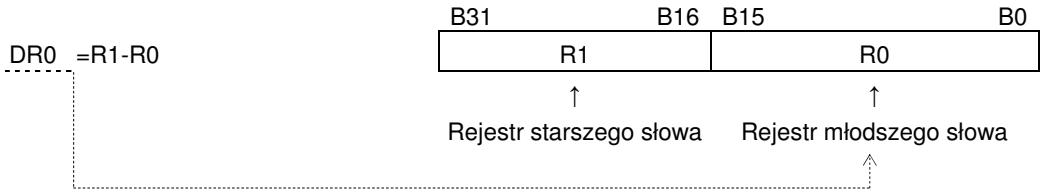
Przykład 3:

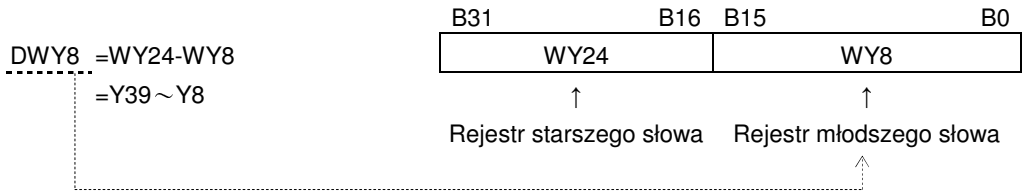
Schemat drabinkowy	Kod mnemoniczny FP-08
	

5.1.2 Numer instrukcji i instrukcje pochodne

Jak napisano wcześniej, poza dziewięcioma instrukcjami możliwymi do wprowadzenia za pomocą dedykowanych klawiszy, pozostałe instrukcje funkcyjne muszą być wprowadzane przy użyciu "numeru instrukcji". Za numerami instrukcji można umieścić trzy przyrostki **D**, **P**, **DP** oznaczające trzy dodatkowe funkcje.

D: Oznacza podwójne słowo (32-bitowe). Słowo 16-bitowe jest podstawową jednostką rejestrów w FBs-PLC. Długość danych rejestrów R, T i C (poza C200~C255) wynosi 16 bitów. W przypadku, gdy wymagany jest rejestr o długości danych wynoszącej 32-bity, niezbędne jest wówczas połączenie dwóch kolejnych rejestrów 16-bitowych, takich jak R1-R0, R3-R2 itp. Nazwy takich rejestrów poprzedza przedrostek D. I tak, na przykład DR0 reprezentuje R1-R0, a DR2 reprezentuje R3-R2. Po wprowadzeniu DR0 lub DWY8, wyświetlona zostanie wartość 32-bitowa (R1-R0 lub WY24-WY8).



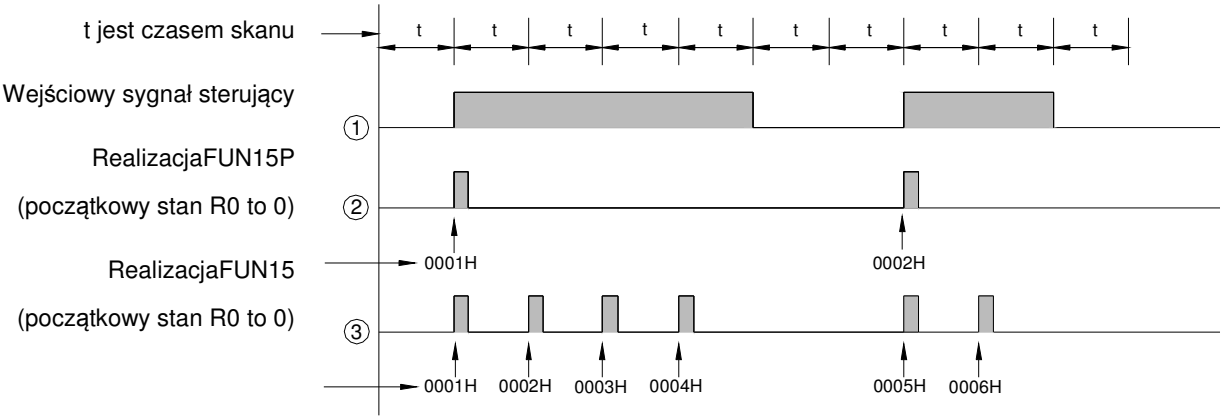


Uwaga: W celu rozróżnienia pomiędzy instrukcjami 16-bitowymi a 32-bitowymi podczas wykorzystywania schematu drabinkowego oraz kodu mnemonicznego, po "numerze instrukcji" dodaliśmy przyrostek D reprezentujący instrukcje 32-bitowe. Wielkość ich argumentu wynosi 32 bity jak pokazano w przykładzie nr 4 na stronie 6-6. Instrukcja FUN 11D charakteryzuje się przyrostkiem D. Oznacza to, że argumenty źródłowe i docelowe powinny również charakteryzować się tym przyrostkiem; zatem dodajna Sa : R0 jest faktycznie Sa=DR0=R1-R0 i Sb=DR2=R3-R2. Należy zwracać także szczególną uwagę na długość pozostałych argumentów używanych w instrukcji. Zawsze należy być pewnym czy dany argument jest słowem 16-to czy 32-bitowym.

P: Oznacza instrukcję trybu impulsowego. Instrukcja zostanie zrealizowana w momencie zmiany stanu wejścia sterującego z 0 na 1 (zbocze narastające). Jak pokazano w przykładzie 1, jeżeli po nazwie instrukcji występuje przyrostek P (FUN 15P), to instrukcja FUN 15P zostanie zrealizowana tylko wtedy, gdy nastąpi zmiana wejściowego sygnału sterującego z 0 na 1. W przypadku braku przyrostka P, oznacza to, że instrukcja będzie realizowana dla każdego skanu do momentu zmiany wejściowego sygnału sterującego z 1 na 0. Poniżej przedstawiony został przykład instrukcji funkcyjnej.

- W przypadku, gdy 〝EN〞 =1 lub （instrukcja **P**） od 0→1, ……

Pierwszy przykład pokazuje realizację instrukcji "nie P" (tryb poziomów), a drugi wskazuje na wymaganie realizację funkcji „**P**" (tryb impulsowy). Poniższy przebieg przedstawia wynik (R0) FUN15 i FUN15P dlatych samych warunkówwejściowych.



DP: wskazuje na to, że instrukcja jest 32-bitowa i pracuje w trybie impulsowym.

Uwaga: Instrukcja **P** jest o wiele bardziej oszczędna czasowo niż instrukcja wykonywana w każdym skanie programu. Dlatego też, użytkownik powinien jak najczęściej wykorzystać instrukcję **P**.

5.1.3 Argument

Argument wykorzystywany jest jako dana wejściowaoraz do zapisu danych. Dane argumentu źródłowego (S) służą jedynie jako dane wejściowei nie zostaną zmienione wraz z realizacją instrukcji. Argument docelowy (D) wykorzystywany jest do zapisywania wyników operacji i jego dane mogą być zmienione po realizacji instrukcji. Poniższa tabela przedstawia nazwy i funkcje argumentów instrukcji FATEK PLC oraz rodzaje styków, cewek lub rejestrów, które mogą być wykorzystane jako argument.

■ Nazwy i funkcje głównych argumentów:

Skrót	Nazwa	Opis
S	Źródłowy	Dane argumentu źródłowego (S) przeznaczone są tylko do odczytu, służą jako dane wejściowei nie ulegną zmianie po realizacji instrukcji. W przypadku więcej niż jednego argumentu źródłowego, każdy z nich oznaczony będzie indeksem dolnym, tj. Sa i Sb.
D	Docelowy	Argument docelowy (D) wykorzystywany jest do zapisywania wyniku operacji. Oryginalne dane ulegną zmianie po operacji. Tylko cewki i rejestry niezabezpieczone przed zapisem mogą być argumentem docelowym.
L	Długość	Opisuje rozmiar danych lub długość tabeli. Wartości te zazwyczaj są stałe.
N	Numer / Liczba	Stała wartość wykorzystywana często jako numery i czasy. W przypadku więcej niż jednej wartości, każda z nich oznaczona będzie indeksem dolnym, tj. Na, Nb, Ns, Nd, itp.
Pr	Wskaźnik	Służy do wskazywania na określony blok danych, określone dane lub rejestr w tabeli. Na ogół, wartość Pr może być zmieniana, dlatego też nie może być stałą ani rejestrem wejściowym.
CV	Aktualna wartość	Wykorzystywana w instrukcjach T i C do zapisywania aktualnej wartości T i C
PV	Wartość zadana	Wykorzystywana w instrukcjach T i C jako odniesienie i porównanie z CV
T	Tabela	Tabelę tworzy zestaw kolejnych rejestrów. Podstawowymi jednostkami operacyjnymi jest słowo i podwójne słowo. W przypadku więcej niż jednej tabeli, każda z nich oznaczona będzie indeksem dolnym, tj. Ta, Tb, Tc i Td, itp.
M	Macierz	Macierz tworzy zestaw kolejnych rejestrów. Podstawową jednostką operacyjną jest bit. W przypadku więcej niż jednej macierzy, każda z nich oznaczona będzie indeksem dolnym, tj. Ma, Mb, Mc i Md, itp.

Poza wymienionymi powyżej głównymi argumentami, istnieją inne argumenty, które są wykorzystywane do specjalnych celów, na przykład Fr - częstotliwość, ST - stos, QU - kolejka, itp. Więcej szczegółów znajduje się w opisach instrukcji.

- Typy argumentów i ich zakres: Typami argumentów dla instrukcji funkcyjnych są: argument dyskretny, rejestr i stała.

  a) Argument dyskretny (bit) :

  Istnieje w sumie pięć instrukcji funkcyjnych odnoszących się do argumentu dyskretnego. Są to SET, RST, DIFU, DIFD i TOGG. Instrukcje te mogą być wykorzystane tylko do operacji przeprowadzanych przez przekaźniki Y (wyjściowe zewnętrzne), M (wewnętrzne i specjalne) i S (krokowe). Poniższa tabela zawiera argumenty i zakresy dla pięciu instrukcji funkcyjnych.

Zakres / Argument	Y	M	SM	S
	Y0 □ Y255	M0 □ M1911	M1912 □ M2001	S0 □ S999
D	○	○	○*	○

Symbol „o" oznacza, że argument D (docelowy) może wykorzystać dany rodzaj cewek jako argumenty. Indeks "*" nad symbolem „o" w kolumnie SM oznacza, że przekaźników zabezpieczonych przed zapisem nie należy stosować jako argumenty. Opis instrukcji dla przekaźników specjalnych znajduje się na stronie 2-3.

  b) Argument - rejestr:

  Głównym argumentem dla instrukcji funkcyjnych jest rejestr. Istnieją dwa typy rejestrów: rejestry rodzime będące słowami lub słowami podwójnymi, takimi jak R, T, C oraz rejestry pochodne (WX, WY, WM, WS) tworzone przez dyskretne bity. Rodzaje rejestrów, które mogą zostać wykorzystane jako argumenty instrukcji oraz ich zakresy wymienione zostały w tabeli poniżej:

Zakres	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR

Argu-ment	WX0 □ WX240	WY0 □ WY240	WM0 □ WM1896	WS0 □ WS984	T0 □ T255	C0 □ C255	R0 □ R3839	R3840 □ R3903	R3904 □ R3967	R3968 □ R4167	R5000 □ R8071	D0 □ D4095	16/32-bit +/- numer	V, Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○*	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○
⋮														

Symbol "○" w tabeli oznacza, że określony rodzaj danych można zastosować jako argument. Symbol "○*" oznacza, że określony rodzaj danych można zastosować jako argument, oprócz rejestrów zabezpieczonych przed zapisem. Szczegóły dotyczące rejestrów zabezpieczonych przed zapisem znajdują się we wstępie do rejestrów specjalnych na stronie 2-8.
Jeżeli R5000～R8071 nie są ustawione jako rejestry tylko do odczytu, można je wykorzystać jako normalne rejestry (z możliwością odczytu i zapisu).

Uwaga 1: Rejestry z przedrostkami, takie jak WX, WY, WM i WS składają się z 16 bitów. Na przykład, WX0 oznacza, że rejestr składa się z bitów X0(bit 0)~X15(bit 15). WY144 oznacza, że rejestr składa się z bitów Y144(bit 0)~Y159(bit 15). Należy pamiętać, że numer dyskretny musi być wielokrotnością 8, czyli 0, 8, 16, 24 …

Uwaga 2: Ostatni rejestr (słowo) w tabeli nie może być reprezentowany argumentem 32-bitowym, ponieważ dla takiego argumentu wymagane są 2 słowa.

Uwaga 3: Rejestry TMR（T0～T255）i CTR（C0～C255）są rejestrami odpowiednio zegarów i liczników. Mimo, że mogą być one wykorzystane jako rejestry ogólne, to komplikują pracę systemów i utrudniają debugowanie. Dlatego też należy unikać zapisywania czegokolwiek w rejestrach TMR i CTR.

Uwaga 4: T0～T255 i C0～C199 są rejestrami 16-bitowymi,a C200~C255 są rejestrami 32-bitowymi, a w związku z tym nie mogą być wykorzystywane jako argumenty 16-bitowe.

Uwaga 5: Oprócz bezpośredniego oznaczenia za pomocą numeru rejestru (adres), argumenty rejestrów w zakresie R0～R8071 mogą być łączone z rejestrami wskaźników V, Z lub P0~P9 w celu realizacji adresowania pośredniego. Opis wykorzystania rejestru wskaźnika (XR) do realizacji adresowania pośredniego znajduje się w następnym rozdziale (rozdział 5.2).

c) Argumenty stałe :

Zakresem stałych wartości 16-bitowych jest -32768~32767. Zakresem stałych wartości 32-bitowych jest -2147483648~2147483647. Wartością stałą dla kilku instrukcji funkcyjnych może być tylko wartość dodatnia. Zakresy dla wartości stałych 16-bitowych i 32-bitowych wymienione są w poniższej tabeli.

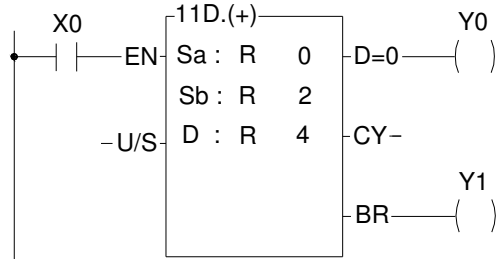
Klasyfikacja	Zakres
16-bitowa liczba ze znakiem	-32768～32767
16-bitowa liczba bez znaku	0～65535
32-bitowa liczba ze znakiem	-2147483648～2147483647
32-bitowa liczba bez znaku	0～4294967295
16 / 32-bitowa liczba ze znakiem	-32768～32767 lub -2147483648～2147483647
16 / 32-bitowa liczba bez znaku	0～65535 lub 0～4294967295

Długość i rozmiar określonego argumentu, tj. L, rozmiar bitu, N, itp. mogą być różne. W takiej sytuacji, różnice są oznaczone bezpośrednio w kolumnie argumentu. Należy odnieść się do opisów instrukcji funkcyjnych.

5.1.4 Wyjście funkcji(FO)

"Wyjście funkcji" (FO) wykorzystywane jest do wskazania wyniku instrukcji funkcyjnej. Tak jak w przypadku wejścia sterującego, wszystkie wyjścia funkcji oznaczane są skrótem od funkcji wyjścia. Na przykład skrót CY od słowa CarrY. Maksymalna liczba wyjść funkcyjnych wynosi 4. Oznaczone są one jako FO0, FO1, FO2, FO3. Stan FO musi być wywołany przez instrukcję FO (na urządzeniu do programowania FP-08 znajduje się specjalny przycisk FO). Nieużywany FO może być pozostawiony w stanie niepodłączonym do żadnego elementu, tj. FO1 (CY) w przykładzie 4.

Przykład 4 :

Schemat drabinkowy	Kody mnemoniczne
	ORG X 0 FUN 11D Sa: R 0 Sb: R 2 D : R 4 FO 0 OUT Y 0 FO 2 OUT Y 1

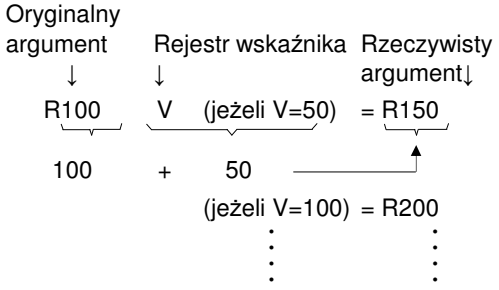
Jeżeli M1919=0, to stan FO będzie aktualizowany tylko po realizacji instrukcji. Aktualny stan utrzyma się do momentu wygenerowania nowego stanu FO po ponownej realizacji instrukcji (utrzymywanie pamięci).

Jeżeli M1919=1, w przypadku braku realizacji instrukcji, stan FO zostanie wyzerowany (brak utrzymywania pamięci).

5.2 Wykorzystanie rejestrów wskaźnikowych (XR) do adresowania pośredniego

Instrukcje funkcyjne FBs-PLC zawierają argumenty, które mogą być łączone z rejestrem wskaźnikowym (V,Z, P0~P9) w celu realizacji adresowania pośredniego (będą one wymienione w tabeli argumentów). Jednakże, tylko rejestry w zakresie R0～R8071 mogą być łączone z rejestrem wskaźnikowym w celu realizacji adresowania pośredniego (inne argumenty, takie jak dyskretny, stały i D0～D3071 nie mogą zostać wykorzystane do adresowania pośredniego).

Istnieje dwanaście rejestrów wskaźników XR (V, Z, P0~P9). Rejestr V jest rejestrem specjalnym R4164, rejestr Z jest rejestrem R4165, a rejestr P0~P9 jest rejestrem (D4080~D4089). Rejestr adresowany wskaźnikiem przesunięty jest względem oryginalnego argumentu o wartość rejestru wskaźników.



Jak pokazano na powyższym schemacie, aby zmienić adres argumentu należy jedynie zmienić wartość V. Po połączeniu adresowania pośredniegoz instrukcjami funkcyjnymi FBs-PLC, można uzyskać efektywne sterowanie za pomocą bardzo prostych instrukcji. Wykorzystując jako przykład program przedstawiony poniżej, wystarczy jedynie wykorzystać instrukcję przesunięcia bloku (BT_M), aby uzyskać dynamiczne wyświetlanie danych bloku.

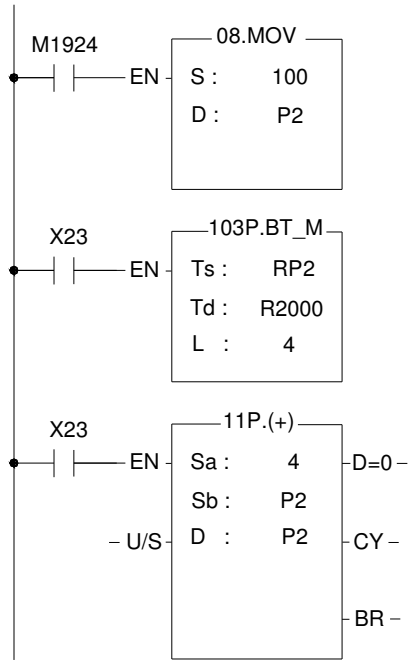
Rejestr wskaźników P0~P9. Wstęp

W przypadku adresowania pośredniego, rejestr Rxxxx może łączyć się z V、Z & P0～P9 w celu realizacji adresowania pośredniego; natomiast rejestr Dxxxx nie może łączyć się z V, Z w celu realizacji adresowania pośredniego, ale dozwolony jest rejestr P0～P9.

W przypadku połączenia rejestru pośredniegoz rejestrem Rxxxx, na przykład R0 z V, Z, to formatem instrukcji będzie R0Z (jeżeli V=100, to R100) lub R0Z (jeżeli Z=500, to R500); w przypadku połączenia rejestru pośredniegoP0～P9 z rejestrem Rxxxx, formatem instrukcji będzie RPn (n=0～9) lub RPmPn (m,n=0～9), na przykład RP5 (jeżeli P5=100, to R100) lub RP0P1(jeżeli P0= 100, P1=50, to150).

W przypadku połączenia rejestru pośredniegoP0～P9 z rejestrem Dxxxx, formatem instrukcji będzie DPn (n=0～9) lub DPmPn (m,n=0～9), na przykład DP3 (jeżeli P3=10, to D10) lub DP4P5 (jeżeli P4=100, P5=1, to D101).

Możliwe jest też połączenie z dwoma rejestramiP0～P9 (na przykład P2=20, P3=30). W przypadku połączenia Rxxxx lub Dxxxx z oboma tymi rejestrami, to RP2P3 wskaże na R50, a DP2P3 wskaże na D50.

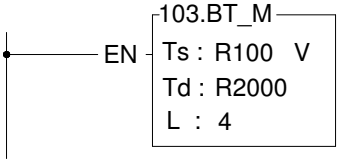
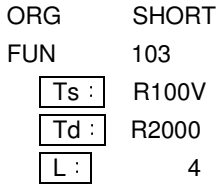


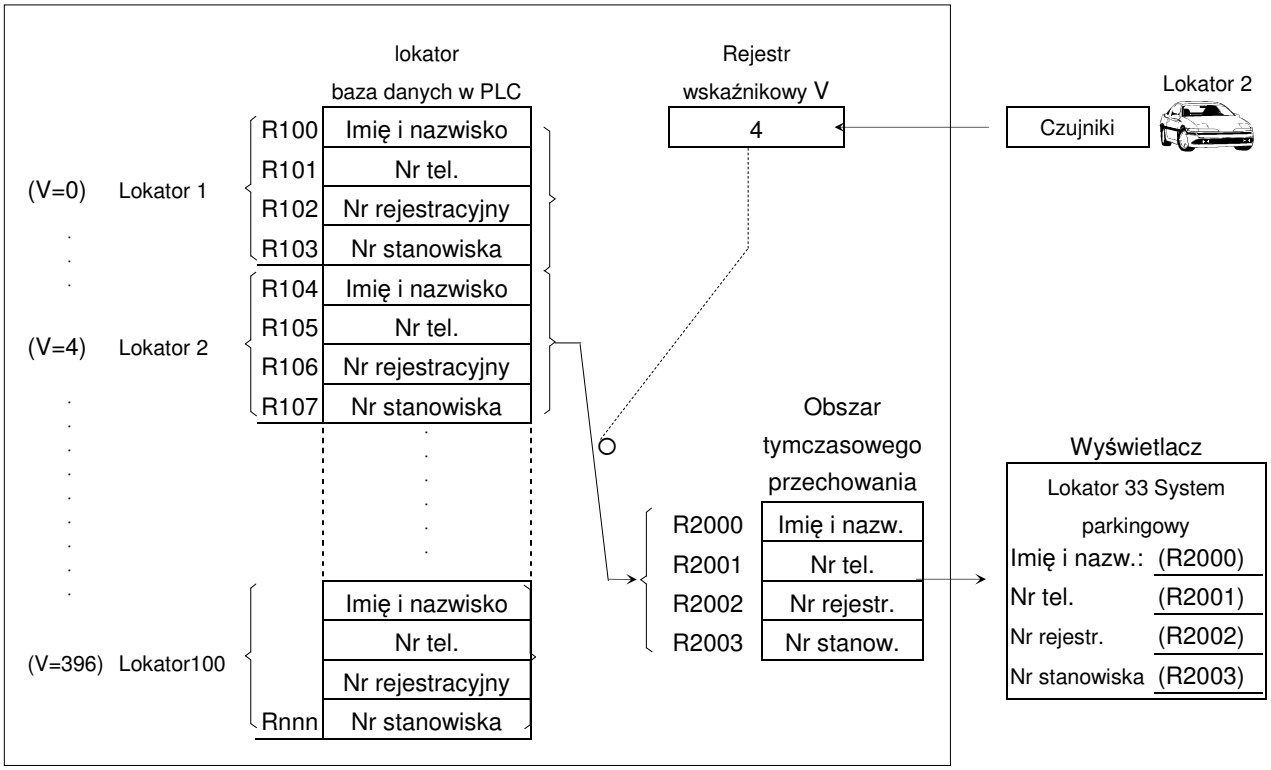
1. Rejestr adresowania pośredniegoP2=100 przy pierwszym uruchomieniu.

2. Kiedy X23 zmienia się z 0→1, FUN103 zrealizuje przesunięcie tabeli, rejestr źródłowy rozpocznie się od R100 (P2=100), a docelowy od R2000. Długość tabeli 4.
   Realizacja zawartości R100~R103 dla R2000~R2003 przy pierwszej próbie; realizacja zawartości R104~R107dla R2000~R2003 przy drugiej próbie…

3. Zwiększenie rejestru wskaźnika o 4 w celu wskazania następnej wartości.

Przykład programu adresowania pośredniego

Schemat drabinkowy	Kody mnemoniczne
	



Opis

Załóżmy, że w ramach systemu zarządzania parkingowego znajduje się 100 miejsc parkingowych przeznaczonych dla lokatorów wspólnoty mieszkaniowej. Każdego lokatora charakteryzuje zestaw podstawowych informacji, takich jak imię i nazwisko, numer telefonu, numer rejestracyjny pojazdu i numer stanowiska parkingowego. Jak pokazano na powyższym schemacie, informacje te zapisane są w czterech kolejnych rejestrach PLC. Łącznie, informacje zajmują 400 rejestrów (R100～R499). Każdy lokator otrzymał kartę z unikalnym numerem (0 dla lokatora 1, 4 dla lokatora 2, itd.), która służy jako przepustka przy głównym wjeździe na parking. Numer karty jest wykrywany przez PLC i zapisywany w rejestrze wskaźnikowym „V". Na wyświetlaczu parkingowym (LCD lub CRT) wyświetlają się tylko dane zawarte w R2001～R2003 w PLC. Na przykład: wykryta została karta lokatora 2 z numerem 4. Rejestr V=4 oraz PLC przeniesie dane z R104～R107 na obszar czasowego przechowywania (R2000～R2003). W ten sposób, na wyświetlaczu wyświetlą się dane lokatora 2 zaraz po tym jak jego karta zostanie wykryta.

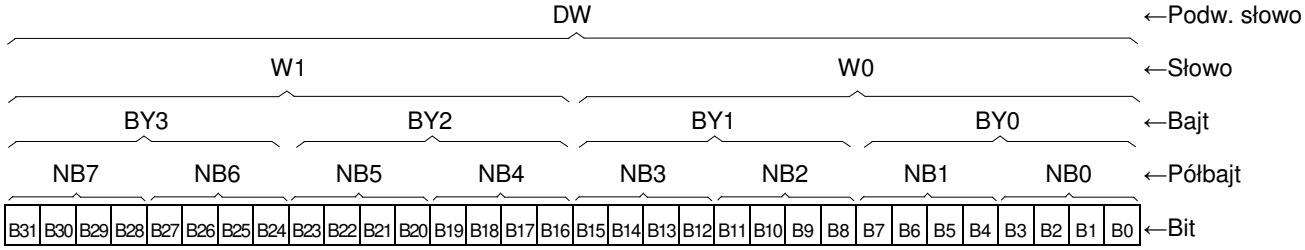
Ostrzeżenie
1. Pomimo, że wykorzystanie rejestru wskaźnikowego do adresowania pośredniego jest efektywnym i elastycznym rozwiązaniem, to dowolne i nieprzemyślane zmiany wartości V i Z mogą spowodować ogromne szkody poprzez nieprawidłowy zapis danych. Dlatego też, użytkownik powinien zachować szczególną ostrożność. 2. W zakresie rejestrów danych, które mogą być wykorzystane do adresowania (R0～R8071), 328 rejestrów R3840～R4167 (tj. IR, OR i SR) zarezerwowanych jest do zastosowań systemowych oraz jako we/wy. Dowolne zapisywanie danych w tych rejestrach może wywołać błędy systemowe lub we / wy i spowodować ogromne szkody. Ponieważ użytkownicy mogą mieć trudności z wykryciem i kontrolą zmian adresowych rejestru spowodowanych zmianą wartości V i Z, PLC automatycznie sprawdza, czy docelowy adres mieści się w zakresie R3840～R4067. Jeżeli tak, operacja zapisu nie zostanie przeprowadzona, a znacznik M1969 „Niedozwolony zapis adresowania pośredniego" zostanie ustawiony jako 1. W przypadku konieczności zapisu w rejestrach R3840～R4067, należy wykorzystać adresowanie bezpośrednie.

5.3 Układ numerowania

5.3.1 Kod binarny i powiązana terminologia

Kod binarny jest podstawowym układem numerowania stosowanym w komputerach cyfrowych. Kod system stosowany jest także dla PLC z uwagi na obsługę dyskretnych wartości WŁ / WYŁ. Należy dokładnie zapoznać się z poniższa terminologią.

- Bit: (W skrócie B, np.: B0, B1, itd.) Jest podstawową jednostką dla wartości binarnych. Bit może mieć status „0" lub „1".
- Półbajt: (W skrócie: NB, np.: NB0, NB1, itd.) Składa się z czterech kolejnych bitów (np.: B3～B0) i może być wykorzystany do wyrażenia liczby dziesiętnej 0～9 lub szesnastkowej 0～F.
- Bajt: (W skrócie: BY, np.: BY0, BY1, itd.) Składa się z dwóch kolejnych półbajtów (lub 8 bitów, np.: B7～B0) i może być wykorzystany do wyrażenia dwucyfrowej liczby szesnastkowej 00～FF.
- Słowo: (W skrócie W, np.: W0, W1, itd.) Składa się z dwóch kolejnych bajtów (lub 16 bitów, np.: B15～B0) i może być wykorzystane do wyrażenia 4-cyfrowej liczby szesnastkowej 0000～FFFF.
- Podwójne słowo: (W skrócie DW, np.: DW0, DW1, itd.) Składa się z dwóch kolejnych słów (lub 32 bitów, np.: B31～B0) i może być wykorzystane do wyrażenia 8-cyfrowej liczby szesnastkowej 00000000～FFFFFFFF.



- Liczba zmiennoprzecinkowa: Także składa się z dwóch kolejnych słów. Maksymalny zakres, który może być wyrażony taką liczbą to $\pm(1.8*10^{-38}\sim3.4*10^{38})$. Szczegóły w rozdziale 5.3.6.

5.3.2 Kodowanie wartości numerycznych dla FBs-PLC

FBs-PLC wykorzystuje kodowanie BCD do operacji wewnętrznych, co oznacza, że dane z zewnętrznych wejść BCD muszą być przekonwertowane na wartość binarną przed rozpoczęciem pracy PLC. Jak wiadomo, kod BCD jest bardzo trudny do odczytu i zapisu na PLC dla człowieka. Dlatego też, FP-08 i WinProladder wykorzystuje jednostki dziesiętne lub szesnastkowe do wprowadzania lub wyświetlania danych. W rzeczywistości zaś, wszystkie operacje w PLC przeprowadzane są za pomocą kodu binarnego.

Uwaga: W przypadku wprowadzenia lub wyświetlenia danych bez użycia FP-08 ani WinProladder (na przykład wprowadzenie lub pobranie danych z PLC przez złącza we / wy za pomocą przełącznika tarczowego lub siedmiosegmentowego wyświetlacza), należy wykorzystać program drabinkowy w celu przeprowadzenia konwersji wartości z dziesiętnych na binarne. Umożliwi to wprowadzanie i wyświetlanie danych bez potrzeby użycia FP-08 ani WinProladder. Należy odnieść się do działania funkcji FUN20(BIN→BCD) i FUN21(BCD→BIN).

5.3.3 Zakres wartości numerycznych

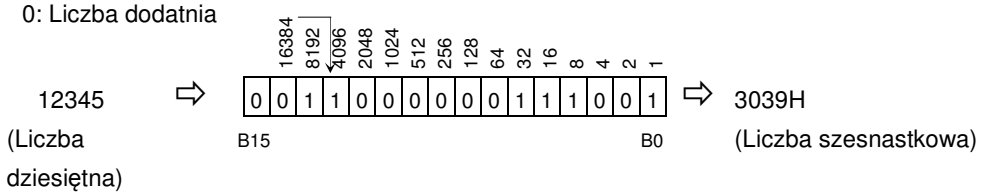
Jak wspomniano powyżej, FBs-PLC wykorzystuje wartości binarne do realizacji wewnętrznych operacji. Wartość 16-bitowa, 32-bitowa i liczba zmiennoprzecinkowa to trzy różne dane numeryczne wykorzystywane przez FBs-PLC. Poniżej przedstawione są zakresy dla tych wartości numerycznych.

16-bitów	-32768～32767
32-bity	-2147483648～2147483647
Liczba zmiennoprzecinkowa	$\pm(1.8*10^{-38}\sim3.4*10^{38})$

5.3.4 Wyrażenie wartości numerycznej (Początkujący mogą pominąć ten punkt)

W celu umożliwienia użytkownikowi dokładniejszego zrozumienia operacji na wartościach numerycznych przy bardziej skomplikowanych zastosowaniach, poniżej opisano sposób wyrażenia i charakterystykę 16-bitowych i 32-bitowych wartości numerycznych.

Najbardziej znaczące bity MSB spośród liczb 16-bitowych i 32-bitowych (B15 dla 16-bitowych i B31 dla 32-bitowych) wykorzystywane są w celu identyfikacji liczb dodatnich i ujemnych (0: dodatnia i 1: ujemna). Pozostałe bity (B14~B0 lub B30~B0) wyrażają wielkość liczby. Poniższy przykład wykorzystuje wartość 16-bitową. Należy pamiętać, że przykład ten ma też przełożenie na liczby 32-bitowe. Jedyną różnica jest długość.

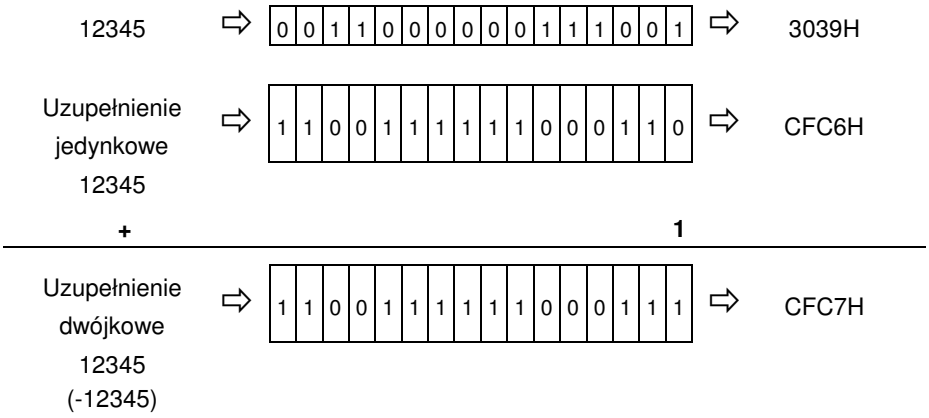


8192+4096+32+16+8+1=12345

W powyższym przykładzie, niezależnie od wielkości liczby (16-bitowa lub 32-bitowa), najmniej znaczący bit LSB (B0) równy jest 1, B1 = 2, B2 = 4, B3 = 8, itd. Liczba wyrażona przez lewy sąsiadujący bit będzie dwa razy większa (1, 2, 4, 8, 16, itd.), a wartość będzie sumą liczb wyrażonych przez bity równe 1.

5.3.5 Wyrażenie liczby ujemnej (Początkujący mogą pominąć ten punkt)

Jak opisano powyżej, jeżeli MSB równy jest 1, liczba będzie miała wartość ujemną. Liczby ujemne FBs-PLC wyrażane są poprzez uzupełnienie dwójkowe, tj. odwrócenie wszystkich bitów (B15～B0 lub B31～B0) równorzędnej liczby dodatniej (tzw. uzupełnieniem jedynkowym określa się zmianę bitów o statusie 1 na 0 i bitów o statusie 0 na 1) i dodatnie 1. W powyższym przykładzie, liczbą dodatnią jest 12345. Poniżej opisano uzupełnienie dwójkowe (tj. -12345):



5.3.6 Wyrażenie liczby zmiennoprzecinkowej (Początkujący mogą pominąć ten punkt)

Format liczby zmiennoprzecinkowej FATEK-PLC jest zgodny z normą IEEE-754. PLC wykorzystuje podwójne słowo do zapisu. Może być ono wyrażone w następujący sposób:

liczba zmiennoprzecinkowa = znak + wykładnik + mantysa

Znak	Wykładnik	Mantysa
b_{31}	b_{30} ～ b_{23}	b_{22} ～ b_0
1 bit	8 bitów	23 bity

32 bity

- ▲ Jeżeli bit znaku ma status 0, liczba jest dodatnia; jeżeli bit znaku ma status 1, liczba jest ujemna.
- ▲ Wykładnik wyznaczany jest jako 8 bitowa wartość pomniejszona o 127.
- ▲ Mantysa jest 23 bitowa z niejawnym zwiększeniem zakresu do 24 bitów. Znormalizowana mantysa zawsze zaczyna się bitem 1, za którym znajduje się kropka i pozostała część mantysy. Pierwszy bit 1, który jest zawsze obecny w znormalizowanej mantysie, jest ukryty i nie jest w żaden sposób wyrażany.

- ● Zasada konwersji liczby całkowitej na zmiennoprzecinkową jest następująca:

$$N = (-1)^{S} * 2^{(E-127)} * (1.M) \qquad 0 < E < 255$$

Na przykład :

(1). $1 = (-1)^{0} * 2^{(01111111)} * (1.000\ldots\ldots0)$

Znak wyrażony jest za pomocą 0, kod wykładnika pomniejszonego o 127 wynosi 127 = 01111111, a istotny bit wynosi 1, a co za tym idzie, mantysa składa się z samych zer. W związku z tym, prostym

wyrażeniem wartość 1 według IEEE 754 równa jest :

Kod (1) =

0	0	1	1	1	1	1	1	1	0	0	0	0	0·····0	0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m m····· m	m	m

= 3F800000H

(2). 0.5 = (-1) ^{0} * 2 $^{(01111110)}$ * (1.000……0)

Znak wyrażony jest za pomocą 0, kod wykładnika pomniejszonego o 127 wynosi 126 - 127 = 01111110, a istotny bit wynosi 1, a co za tym idzie, mantysa składa się z samych zer. W związku z tym, prostym wyrażeniem wartość 0.5 według IEEE 754 równa jest :

Code(0.5) =

0	0	1	1	1	1	1	1	0	0	0	0	0	0·····0	0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m m····· m	m	m

= 3F000000H

(3). -500.125 = (-1) ^{1} * 2 $^{(10000111)}$ * (1. 11110100001000000000000)

Znak wyrażony jest za pomocą 1, kod wykładnika pomniejszonego o 127 wynosi 135 - 127 = 10000111, a istotny bit wynosi 1, a co za tym idzie, mantysa przybiera postać 11110100001000000000000. W związku z tym, prostym wyrażeniem wartość -500.125, według IEEE 754 równa jest:

Kod(-500.125) =

1	1	0	0	0	0	1	1	1	1	1	1	1	0	1	0	0	0	0	1	0	0	0	00·····0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	mm·····m	m

= C3FA1000H

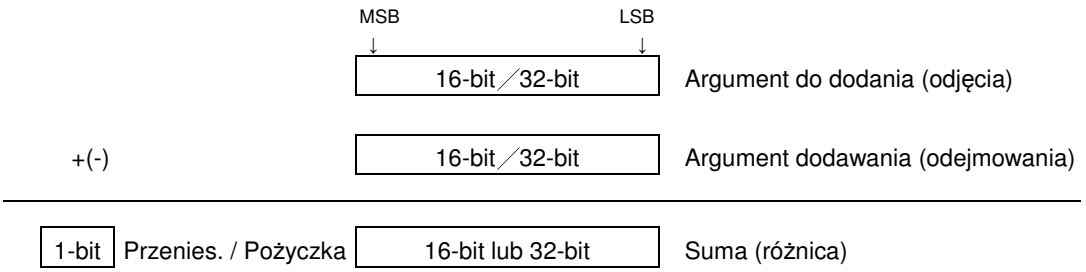
5.4 Przekroczenie zakresu dodatniego lub ujemnego przy wykonaniu instrukcji zwiększenia (+1) lub zmniejszenia (-1) wartości.

Maksymalnymi dodatnimi wartościami, które mogą zostać wyrażone za pomocą 16-bitowych i 32-bitowych argumentów są odpowiednio 32767 i 2147483647, natomiast minimalnymi wartościami ujemnymi są odpowiednio –32768 i –2147483648. W przypadku, gdy przy zwiększeniu lub zmniejszeniu argumentu (np.: gdy wartość zliczania w górę / w dół licznika lub wartość rejestru równa jest +1 lub -1) wynik przekracza dodatnią wartość graniczną argumentu, następuje przekroczenie dodatniego zakresu (OVF). Spowoduje to zmianę wartości na ujemną wartość graniczną (np.: dodanie 1 do 16-bitowej granicznej wartości dodatniej 32767 spowoduje jej zmianę na -32768). W przypadku, gdy wynik jest mniejszy niż ujemna wartość graniczna, następuje przekroczenie ujemnego zakresu (UDF). Spowoduje to zmianę wartości na dodatnią wartość graniczną (np.: odjęcie 1 od granicznej wartości ujemnej -32768 spowoduje jej zmianę na 32767), jak pokazano w poniższej tabeli. Marker znakujący przekroczenie dodatniego lub ujemnego zakresu znajduje się w FO FBs-PLC i może być wykorzystane w instrukcjach w celu osiągnięcia odpowiednich wyników w przypadku wartości 16-bitowych i 32-bitowych.

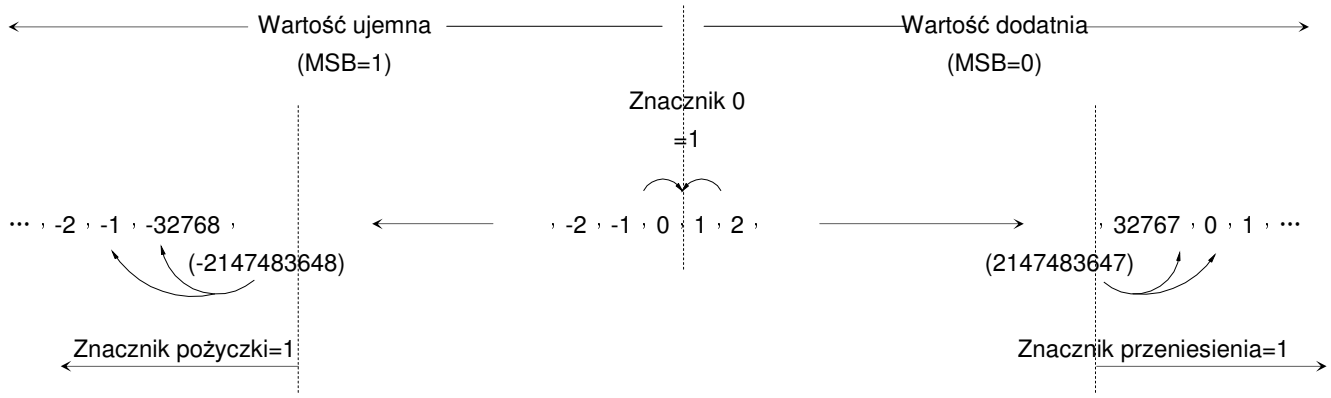
Zwiększ (zmniejsz) wynik / Przekroczenie zakr.dodatniego ujemnego	Argument 16-bitowy	Argument 32-bitowy
Zwiększ	OVF=1 → - 32767, - 32768, 32767, 32766, 32765	OVF=1 → - 2147483646, - 2147483647, - 2147483648, 2147483647, 2147483646
Zmniejsz	UDF=1 → - 32767, - 32768, 32767, 32766, 32765	UDF=1 → - 2147483647, - 2147483648, 2147483647, 2147483646, 2147483645

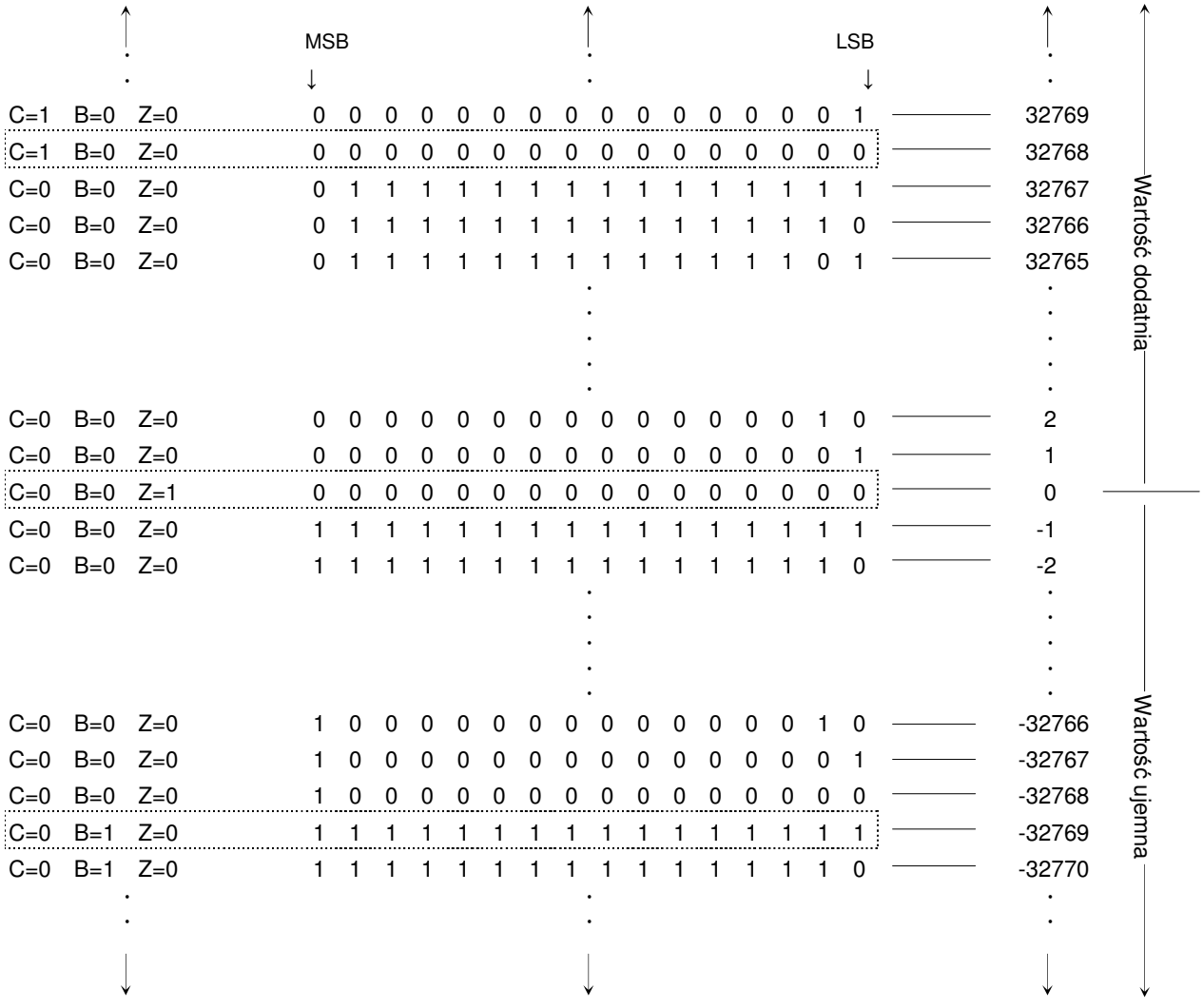
5.5 Przeniesienie i pożyczka przy dodawaniu / odejmowaniu

Przekroczenie dodatniego / ujemnego zakresu zachodzi w momencie, gdy operacja dodawania / odejmowania powoduje, że wartość argumentu przekracza dodatnią/ujemdną wartość graniczną. W wyniku tego, pojawia się znacznik przekroczenia dodatniego / ujemnego zakresu. Znacznik przeniesienia / pożyczki rożni się od znacznika przekroczenia dodatniego / ujemnego zakresu. Po pierwsze, do wykonania operacji dodawania (odejmowania) potrzebne są dwa argumenty, dla których wyznaczona zostanie suma (różnica) oraz znacznik przeniesienia / pożyczki. Ponieważ liczba bitów dodawanych (odejmowanych) wartości jest taka sama (zarówno w przypadku liczb 16-bitowych i 32-bitowych), wynik dodawania (odejmowania) może przekroczyć wartość 16-bitową lub 32-bitową. Dlatego też, ważne jest, aby wykorzystać znacznik przeniesienia / pożyczki w celu wyrażenia rzeczywistej wartości. Znacznik przeniesienia ustawiany jest, gdy wynik dodawania (odejmowania) przekroczy dodatnią wartość graniczną (32767 lub 2147483647) argumentu sumy (różnicy). Znacznik pożyczki ustawiany jest, gdy wynik dodawania (odejmowania) przekroczy ujemną wartość graniczną (-32768 lub -2147483648) argumentu sumy (różnicy). Instrukcja dodawania / odejmowania FO FBs-PLC posiada zarówno znacznik przeniesienia jak i pożyczki.



Ponieważ wszystkie operacje numeryczne FBs-PLC wykorzystują uzupełnienie dwójkowe, wyrażenie wartości ujemnej sumy (różnicy) uzyskanej z dodawania (odejmowania) różni się od wyrażenia standardowej liczby ujemnej. W MSB sumy (różnicy) nie może nigdy pojawić się 0. Znacznik przeniesienia wyraża dodatnią wartość 32768 (2147483648), natomiast znacznik pożyczki wyraża wartość ujemną -32768 (-2147483648).





C = Przeniesienie B = Pożyczka Z = Zero