

# STEROWANIE MASZYN I URZĄDZEŃ I

## Laboratorium

### 6. Działania na ciągach bitów

Opracował: dr hab. inż. Cezary Orlikowski

Instytut Politechniczny



## CIĄGI BITÓW I ICH ADRESOWANIE

Bloki funkcyjne z omawianej w tym punkcie grupy realizują operacje logiczne na ciągach bitów. Maksymalna długość tych ciągów wynosi 256 słów czyli 4096 bitów. Ciągi bitów są więc zarazem ciągami 16-bitowych słów.

Ciągi bitów można tworzyć z bitów przyporządkowanych zmiennym typu  $%R$  oraz  $%I$ ,  $%Q$ ,  $%M$ ,  $%T$ . Adresowanie (określenie) ciągu bitów odbywa się w sposób przedstawiony na rys. 1. Ogólnie, polega na podaniu liczby słów (16-tek bitów) oraz pierwszego z tych słów ( $%R$ ) lub pierwszego bitu (najmniej znaczącego - LSB) w pierwszym słowie utworzonym z bitów zmiennych typu  $%I$ ,  $%Q$ ,  $%M$ ,  $%T$ .

## BLOKI DZIAŁAŃ NA CIĄGACH BITÓW

W programie *CIMPLICITY* dostępne są między innymi następujące bloki funkcyjne:

### Operacje bitowe dwuargumentowe

*AND* – iloczyn logiczny dwóch ciągów bitów

*OR* – suma logiczna dwóch ciągów bitów

*XOR* – suma logiczna wyłączająca dwóch ciągów bitów

Parametry tych bloków (na rys. 2 pokazano blok *AND*) przedstawiono w tabelicy 1. Odpowiednie operacje logiczne wykonywane są na parach bitów ciągów bitów przypisanych wejściom *I1*, *I2*.

**Tablica 1**

Parametry bloków operacji bitowych dwuargumentowych

Parametr	Opis
<i>Enable</i>	Sygnal wejściowy. Gdy do bloku funkcyjnego dołączy sygnał, wykonywana jest operacja logiczna.
<i>I1</i>	<i>I1</i> – wartość stała lub adres zmiennej, stanowiącej pierwsze słowo, na którym ma być wykonywana operacja logiczna.
<i>I2</i>	<i>I2</i> – wartość stała lub adres zmiennej, stanowiącej drugie słowo, na którym ma być wykonywana operacja logiczna.
<i>Q</i>	Wynik operacji logicznej na słowach <i>I1</i> i <i>I2</i> .
<i>LEN</i>	Liczba słów bitowych obszaru, w obrębie którego ma zostać dokonane przesunięcie bitów.

### Operacja bitowa (jednoargumentowa) *NOT* - negacji

Na rys. 3 przedstawiono blok *NOT*, a w tabelicy 2 opis jego parametrów.

**Tablica 2**

Parametry bloku *NOT*

Parametr	Opis
<i>Enable</i>	Sygnal wejściowy. Gdy do bloku funkcyjnego dołączy sygnał, wykonywana jest operacja logiczna.
<i>I1</i>	<i>I1</i> – wartość stała lub adres zmiennej, stanowiącej słowo, które ma być zanegowane.
<i>Q</i>	Wynik operacji logicznej - negacja słowa <i>I1</i> .
<i>LEN</i>	Liczba słów bitowych obszaru, w obrębie którego ma zostać dokonane przesunięcie bitów.

### Przesuwanie bitów

*SHL* – przesuwanie bitów w lewo,

*SHR* – przesuwanie bitów w prawo.

Parametry tych bloków zamieszczono w tabl. 3, a widok bloku *SHL* na rys. 4.

**Tablica 3**

Parametry bloków przesuwania bitów

Parametr	Opis
<i>Enable</i>	Sygnal wejściowy. Gdy do bloku funkcyjnego dożywa sygnal, wykonywane jest przesunięcie.
<i>IN</i>	<i>IN</i> zawiera adres pierwszego słowa, którego bity mają zostać przesunięte.
<i>N</i>	Liczba miejsc (bitów), o które mają zostać przesunięte bity danego słowa (ciągu słów).
<i>B1</i>	Wartość bitu (bitów), które mają zostać wstawione w puste miejsca słowa, powstałe po przesunięciu jego zawartości.
<i>B2</i>	Wartość ostatniego bitu, który wyszedł poza zakres słowa po dokonaniu operacji przesunięcia.
<i>Q</i>	<i>Q</i> zawiera adres pierwszego słowa ciągu słów, otrzymanego po przesunięciu bitów słowa adresowanego przez parametr <i>IN</i> .
<i>LEN</i>	Liczba słów bitowych obszaru, w obrębie którego ma zostać dokonane przesunięcie bitów.

### Rotacja bitów

*ROL* – rotacja bitów w lewo,

*ROR* – rotacja bitów w prawo.

Działanie tych bloków jest podobne do działania bloków *SHL* i *SHR* ale w tym przypadku przesuwanie odbywa się w obiegu zamkniętym. Blok *SHL* pokazano na rys. 5. W tabl. 4 przedstawiono parametry tych bloków.

**Tablica 4**

Parametry bloków rotacji bitów

Parametr	Opis
<i>Enable</i>	Sygnal wejściowy. Gdy do bloku funkcyjnego dożywa sygnal, wykonywana jest rotacja.
<i>IN</i>	<i>IN</i> zawiera adres pierwszego słowa, którego bity mają zostać przesunięte.
<i>N</i>	Liczba miejsc (bitów), o które mają zostać przesunięte bity danego słowa (ciągu słów).
<i>Q</i>	<i>Q</i> zawiera adres pierwszego słowa ciągu słów, otrzymanego po rotacji bitów słowa adresowanego przez parametr <i>IN</i> .
<i>LEN</i>	Liczba słów bitowych obszaru, w obrębie którego ma zostać dokonana rotacja bitów.

### Ustawianie wartości danego bitu

*BSET* – ustawianie wartości danego bitu ciągu bitowego na „1”,

*BCLR* – ustawianie wartości danego bitu ciągu bitowego na „0”.

Na rys. 6. pokazano blok funkcyjny *BSET*, a w tabl. 5 parametry bloków *BSET* i *BCLR*.

**Tablica 5**

Parametry bloków ustawiania „0” lub „1”

Parametr	Opis
<i>Enable</i>	Sygnal wejściowy. Gdy do bloku funkcyjnego dopływa sygnał, wykonywana jest operacja ustawiania wartości bitu.
<i>IN</i>	<i>IN</i> zawiera adres pierwszego słowa ciągu słów, na którym ma zostać wykonana jest operacja.
<i>BIT</i>	Numer bitu słowa <i>IN</i> , którego wartość ma zostać ustawiona. Zakres wartości to $1 \leq BIT \leq 16   LEN$ .
<i>LEN</i>	Liczba słów bitowych ciągu słów, z którego wybierany jest bit, którego wartość ma zostać ustawiona (maksymalnie 256).

### Blok funkcyjny *BITSEQ*

Na rys. 7 przedstawiono blok *BITSEQ*, a w tablicy 6 opis jego parametrów. Ten blok funkcyjny umożliwia przesuwanie wartości „1” w zadanym polu ciągu bitów, w którym pozostałe bity mają wartość „0”. Przesuwanie (rotacja) „jedynek” może odbywać się w wybranym kierunku.

**Tablica 6**

Parametry bloku *BITSEQ*

Parametr	Opis
<i>Address</i>	Adres pierwszego z trzech rejestrów wykorzystywanego obszaru pamięci.
<i>Enable</i>	Przy każdym zboczu narastającym tego sygnału i gdy jednocześnie na wejście <i>R</i> nie jest podawany sygnał, funkcja <i>BITSEQ</i> wykonuje przemieszczenie bitu o wartości równej 1 w lewo lub w prawo.
<i>R</i>	Podanie sygnału na wejście <i>R</i> powoduje ustawienie warunków początkowych, tzn. skopiowanie bieżącego numeru bitu o wartości 1 z parametru <i>STEP</i> .
<i>DIR</i>	<i>DIR</i> określa kierunek przemieszczenia bitu o wartości równej 1. Jeśli na wejście <i>DIR</i> podawany jest sygnał, bit przemieszczany jest w lewo (bieżący numer rośnie). Jeśli sygnał nie jest podawany, bit przemieszczany jest w prawo (bieżący numer maleje).
<i>STEP</i>	<i>STEP</i> jest początkowym numerem bitu o wartości równej 1.
<i>ST</i>	Adres pierwszego bitu obszaru pamięci, na którym działa funkcja <i>BITSEQ</i> .
<i>lbr</i>	obszar rotacji (liczba bitów w rotacji)

### PRZYKŁADY

#### Przykład 1

Na rysunku 8 przedstawiono schemat układu realizującego zerowanie obszaru pamięci sterownika. Obszar ten jest zadanym ciągiem 16 bitów %Q1 - %Q16. Obydwu wejściom oraz wyjściu bloku *XOR* przyporządkowano ten sam ciąg bitów. Działanie tego programu wynika z własności funkcji *XOR* (dla tych samych wartości dwóch argumentów wynik zawsze jest równy „zero”).

#### Przykład 2

Na rysunku 9 pokazano program umożliwiający sprawdzenie działania bloku funkcyjnego *BITSEQ*.

#### Przykład 3

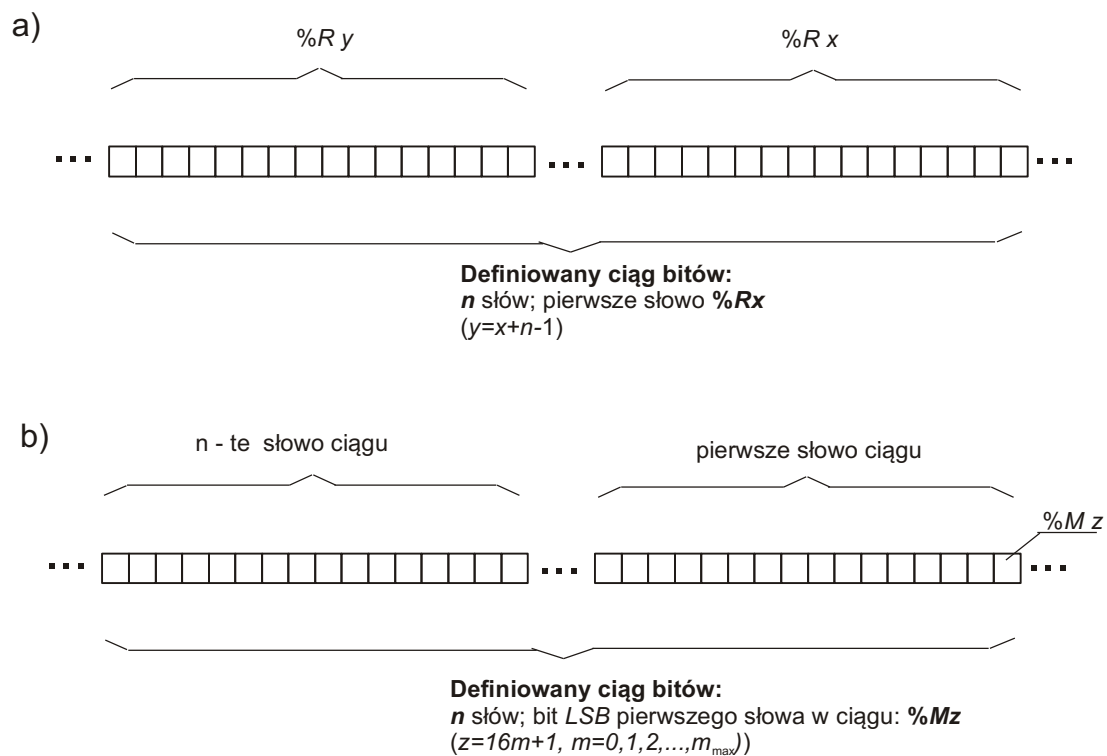
Na rysunku 10 pokazano program umożliwiający załączanie i wyłączanie urządzenia sterowanego przy pomocy jednego przycisku START/STOP (*STST*). Impulsy nieparzyste włączają, a parzyste wyłączają urządzenie.

## CEL I PRZEBIEG ZAJĘĆ

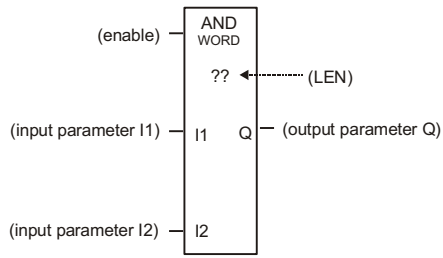
Cel: Programowanie sterownika *GE FANUC* do realizacji układów sterowania zawierających bloki funkcyjne opisane w tym punkcie.

### Przebieg

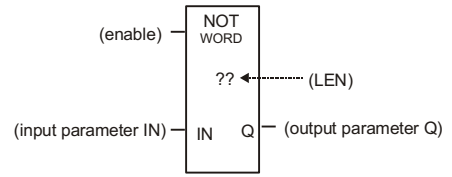
1. Zapoznać się z przykładami programów przedstawionych we wprowadzeniu do ćwiczenia.
2. Zaprogramować sterownik do realizacji programów przedstawionych we wprowadzeniu do ćwiczenia.
3. Zmodyfikować powyższe programy według wskazań prowadzącego. Zbudować nowe programy sterujące zgodnie z poleceniem prowadzącego ćwiczenia. Sprawdzić ich działanie.



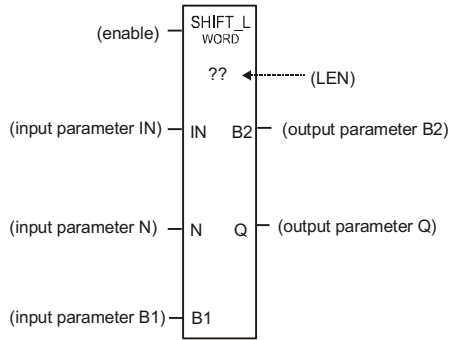
Rys. 1. Ciąg bitów i jego adresowanie



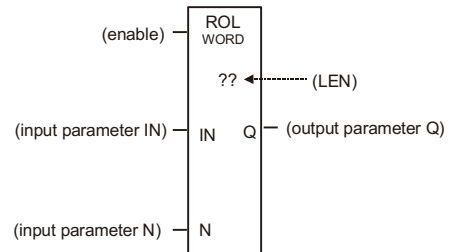
Rys. 2. Blok AND



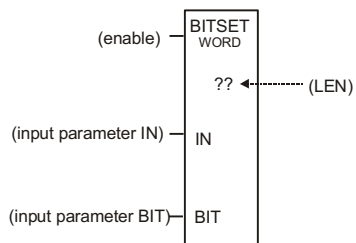
Rys. 3. Blok NOT



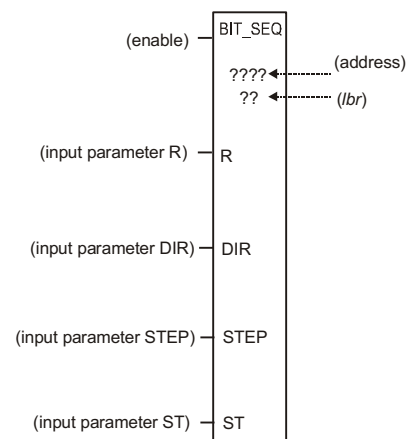
Rys. 4. Blok SHL



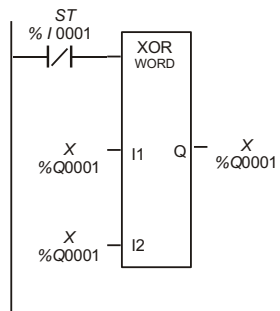
Rys. 5. Blok ROL



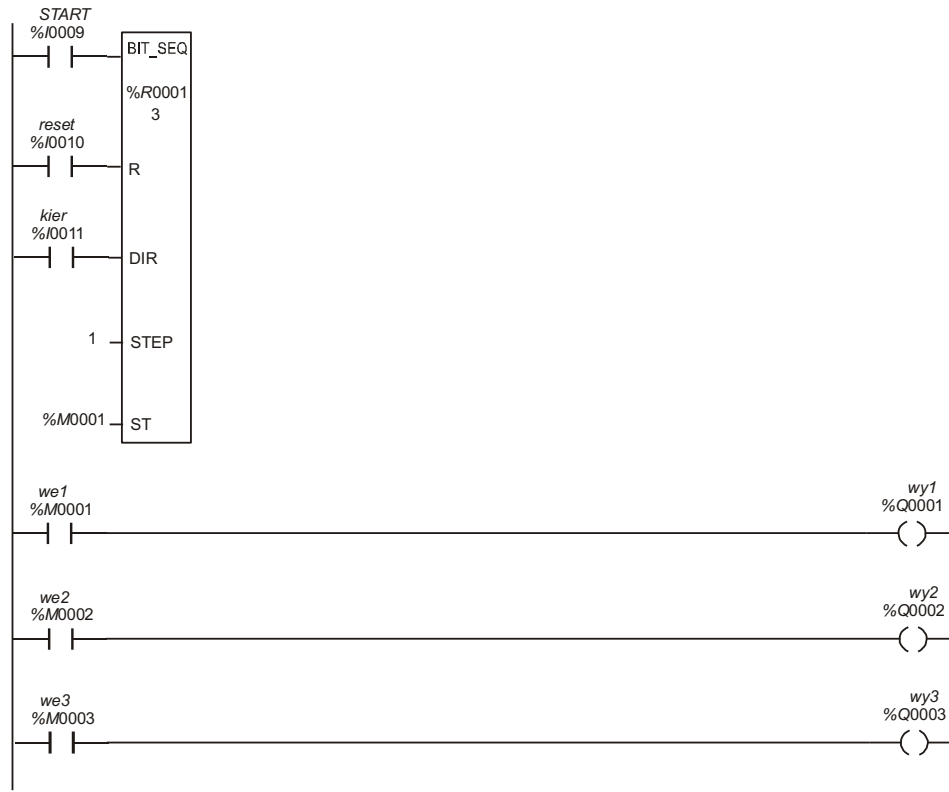
Rys. 6. Blok BSET



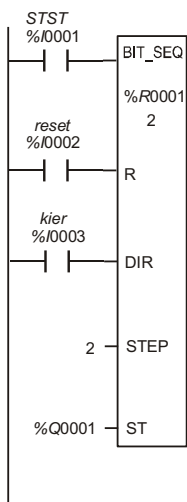
Rys. 7. Blok BITSEQ



Rys. 8. Schemat do przykładu 1



Rys. 9. Schemat do przykładu 2



Rys. 10. Schemat do przykładu 3