PODSTAWY INFORMATYKI

Przygotowanie: Przemysław Sołtan
e-mail: kerk@moskit.ie.tu.koszalin.pl

Mikrooperacje

Mikrooperacja to elementarna operacja wykonywana podczas jednego taktu zegara mikroprocesora na informacji przechowywanej w jednym lub większej liczbie rejestrów. Wynik operacji może zastąpić poprzednią informację w danym rejestrze lub może zostać przesłany do innego rejestru.

Mikrooperacje arytmetyczne

Mikrooperacja arytmetyczna jest to operacje przesłań miedzy rejestrami zmieniająca zawartość informacji podczas przesłania modyfikowaną przez układy logiczne wykonujące zadaną funkcję arytmetyczną.

Mikrooperacje arytmetyczne.

Oznaczenie symboliczne	Opis
$A + B$	Dodawanie
$A - B$	Odejmowanie
$A + 1$	Zwiększenie o jeden (*inkrementacja*)
$A - 1$	Zmniejszenie o jeden (*dekrementacja*)
$\overline{A}$	*Dope*łnienie (*uzupełnienie do 1*)
$\overline{A} + 1$	Utworzenie uzupełnienia do 2
$A + \overline{B}$	Zawartość A plus uzupełnienie do 1 zawartości B
$A + \overline{B} + 1$	Zawartość A plus uzupełnienie do 2 zawartości B (*równoważne odejmowaniu*)

Symbolem mikrooperacji dopełnienia jest kreska nad literą symbolizującą rejestr *(symbol ten jest używany do określenia dopełnienia jednej zmiennej dwójkowej)*.

Mikrooperacje logiczne

Mikrooperacje logiczne są to operacje dwójkowe wykonywane na łańcuchach bitów przechowywanych w rejestrach. W tych operacjach każdy bit jest rozważany oddzielnie i traktowany jako zmienna dwójkowa .

Tablice funkcji dla 16 funkcji dwóch zmiennych.

x	Y	F0	F1 AND	F2	F3	F4	F5	F6 XOR	F7 OR	F8 NOR	F9 NXOR	F10	F11	F12	F13	F14 NAND	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Mikrooperacje logiczne

Funkcja	*Mikrooperacja*	*Nazwa*
$F_o = 0$	$F \leftarrow 0$	Zerowanie
$F_1 = xy$	$F \leftarrow A \wedge B$	AND
$F_2 = x\overline{y}$	$F \leftarrow A \wedge \overline{B}$	
$F_3 = x$	$F \leftarrow A$	Przesłanie A
$F_4 = \overline{x}y$	$F \leftarrow \overline{A} \wedge B$	
$F_5 = y$	$F \leftarrow B$	Przesłanie B
$F_6 = x \oplus y$	$F \leftarrow A \oplus B$	XOR
$F_7 = x + y$	$F \leftarrow A \vee B$	OR
$F_8 = \overline{x + y}$	$F \leftarrow \overline{A \vee B}$	NOR
$F_9 = \overline{x \oplus y}$	$F \leftarrow \overline{x \oplus y}$	NXOR
$F_{10} = \overline{y}$	$F \leftarrow \overline{B}$	Dopełnienie B
$F_{11} = x + \overline{y}$	$F \leftarrow A \vee \overline{B}$	
$F_{12} = \overline{x}$	$F \leftarrow \overline{A}$	Dopełnienie A
$F_{13} = \overline{x} + y$	$F \leftarrow \overline{A} \wedge B$	
$F_{14} = \overline{xy}$	$F \leftarrow \overline{A \wedge B}$	NAND
$F_{15} = 1$	$F \leftarrow$ same jedynki	Ustawienie jedynek

Mikrooperacje logiczne są bardzo użyteczne przy przetwarzaniu poszczególnych bitów lub części słowa przechowywanego w rejestrze. Można je stosować do zmiany wartości bitów, usunięcia grupy bitów lub wstawiania do rejestru nowych wartości bitów.

Operacja ustawiania selektywnego (operacja **OR** $A \leftarrow A \vee B$)[1]:

1010 **A** przed wykonaniem operacji
1100 **B**
1110 **A** po wykonaniu operacji

Do rejestru A wstawiane są jedynki w tych miejscach, w których w rejestrze B znajdują się jedynki.

[1] Operację ustawiania selektywnego stosujemy w przypadku, gdy chcemy dokonać ustawienia odpowiednich bitów na wartość 1.

Operacja dopełnienia selektywnego (operacja **XOR** $F \leftarrow A \oplus B$):

1010 **A** przed wykonaniem operacji
1100 **B**
0110 **A** po wykonaniu operacji

Do rejestru A wstawiane są dopełnienia rejestru A w tych miejscach, w których w rejestrze B znajdują się jedynki. Nie zmienia się wartość bitów rejestru A, którym odpowiadają jedynki w rejestrze B.

Operacja zerowania selektywnego(operacja **AND** A z dopełnieniem B $F \leftarrow A \wedge \overline{B}$):

1010 **A** przed wykonaniem operacji
1100 **B**
0010 **A** po wykonaniu operacji

Do rejestru A wstawiane są zera w tych miejscach, w których w rejestrze B znajdują się jedynki.

Operacja maskowania (operacja **AND** $F \leftarrow A \wedge B$):

1010 **A** przed wykonaniem operacji
1100 **B**
1000 **A** po wykonaniu operacji

Do rejestru A wstawiane są zera w tych miejscach, w których w rejestrze B znajdują się zera.

Operacja wstawiania:

Operacja ta powoduje wstawianie nowej wartości do danej grupy bitów. Dokonuje się tego przez uprzednie maskowanie *(AND)* danych bitów, a następnie logiczne zsumowanie *(OR)* ich z żądaną wartością.

By zastąpić cztery znajdujące się po lewej stronie bity wartością 1001, poddajemy niepotrzebne bity najpierw operacji maskowania *(AND),*

0110 1010 **A** przed wykonaniem operacji
0000 1111 **B** (maskowanie)
0000 1010 **A** po wykonaniu operacji

a następnie wstawiamy *(OR)* nową wartość:

0000 1010 **A** przed wykonaniem operacji
1001 1111 **B** (wstawianie)
1001 1010 **A** po wykonaniu operacji

Operacja porównania:

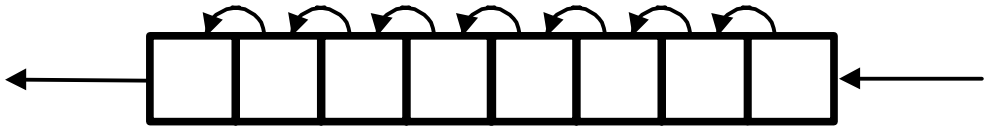
Operacja polega na porównaniu słów przechowywanych w rejestrach A i B. W przypadku, gdy dwa odpowiadające sobie bity są jednakowe to generowane jest zero. W przeciwnym wypadku otrzymujemy jedynkę. Osiąga się to przez wykorzystanie operacji **XOR**.

1010 **A** przed wykonaniem operacji
1010 **B**
0000 **A** po wykonaniu operacji

Następnie sprawdza się, czy otrzymano w wyniku same zera, co jest dowodem, że słowa A i B są sobie równe.

Mikrooperacje przesuwania

Operacja przesuwania bitów wykorzystywana jest do przesyłania informacji między rejestrami, także do wykonywania arytmetycznych, logicznych i innych operacji przetwarzania danych.



Rys. 0-1.Przykład operacji przesuwania w lewo (SL) na rejestrze 8-bitowym (układ przerzutników).

Istnieją cztery rodzaje przesuwania:

- szeregowe przesuwanie przesyłające*(proste)* (**SL**, **SR**);
- przesuwanie logiczne (**SLL**, **SRL**);
- przesuwanie arytmetyczne (**SLA**, **SRA**);
- przesuwanie cykliczne (**SLC**, **SRC**).

Przy ***przesuwaniu szeregowym*** wejście szeregowe rejestru otrzymuje bit z innego rejestru *(źródłowego)*.
W przypadku ***przesuwania logicznego*** bit ten nie jest pobierany z innego rejestru, zamiast tego jest wstawienie 0 do skrajnej pozycji rejestru.

Dla ***przesuwania cyklicznego*** następuje przesuwanie bitów rejestru między dwoma jego końcami. uzyskuje się to przez połączenie wyjścia szeregowego rejestru przesuwającego z jego wejściem szeregowym.
Ze względu na możliwość realizacji przesuwania cyklicznego przy pomocy przesuwania szeregowego *(prostego)* nie zaimplementowano w programie tego typy mikrooperacji[2].

[2] Aby realizować przesuniecie cykliczne należy użyć dyrektywy LINK łącząc wyjście CL z wejściem CR. W takim przypadku mamy możliwość wykonywania przesunięcia cyklicznego w lewo. Dla przesunięcia w prawo należy dodatkowo wykonać połączenie LINK łącząc wyjście CR z wejściem CL.

Przesuwanie arytmetyczne jest mikrooperacją przesuwającą liczbę wraz z jej znakiem. Arytmetyczne przesuwanie w lewo liczby dwójkowej powoduje mnożenie jej przez dwa, a arytmetyczne przesuwanie w prawo odpowiada zaś podzieleniu jej przez dwa. Przy takim przesyłaniu najstarszy bit rejestru (skrajny lewy przerzutnik) przechowuje bit znaku. Przy przesuwaniu bit znaku powinien pozostać niezmieniony ponieważ, gdy mnożymy lub czy dzielimy, znak liczby nie ulega zmianie.