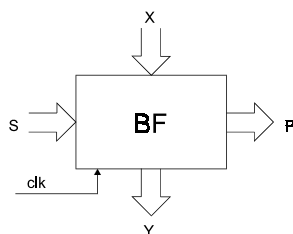


Bloki funkcjonalne

Złożone układy cyfrowe są budowane z wyspecjalizowanych układów logicznych, zwanych blokami funkcjonalnymi. Przez pojęcie bloku funkcjonalnego rozumie się jeden lub kilka układów fizycznych połączonych w sposób umożliwiający wykonanie określonej operacji przetwarzania danych wejściowych.



Blok funkcjonalny

X – wejścia sygnałów reprezentujących dane wejściowe,
Y – wyjścia sygnałów reprezentujących dane wyjściowe,
S – wejścia sterujące,
P – wyjścia predykatowe, sygnalizujące pewne szczególne stany przetwarzania danych oraz wejście zegarowe *clk*.

44

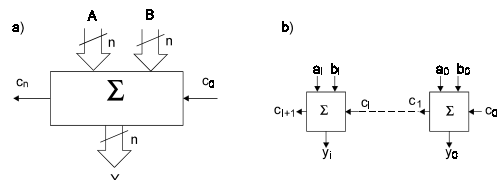
Bloki arytmetyczne

Operację sumowania arytmetycznego realizuje **sumator**, który w najprostszym przypadku może być zbudowany z kaskadowo połączonych sumatorów jednobitowych. Jego działanie można opisać następującymi zależnościami:

$$y_i = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i b_i \vee (a_i \vee b_i) c_i$$

Sumator



a) schemat ogólny,

b) kaskadowe połączenie

sumatorów jednobitowych

45

Sumator może być wykorzystany do realizacji operacji odejmowania. Działanie układu z rysunku można opisać:

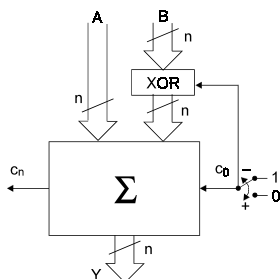
$$Y = A + B \oplus c_0 + c_0, \text{ gdzie } c_0 \in \{0,1\}$$

Dla $c_0 = 0$ mamy $Y = A + B$, czyli sumowanie.

Dla $c_0 = 1$ mamy $Y = A + \bar{B} + 1$;

$\bar{B} + 1$ oznacza liczbę $-B$ w kodzie U2,

zatem $Y = A - B$.



Układ odejmujący zbudowany z sumatora

46

Reprezentacje liczb

Naturalny kod binarny

Reprezentacja *LB* $A = (a_{n-1}, \dots, a_0)$

za pomocą liczby *LD* $L(A)$

$$L = \sum_{i=0}^{n-1} 2^i a_i$$

Kod uzupełnień do 2; U2

$$L = -2^{n-1} a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

n -bitowe słowo reprezentuje liczbę L :

$$-2^{n-1} \leq L \leq 2^{n-1} - 1$$

Np.: -5 1011 ; -3 1101 ; 3 0011

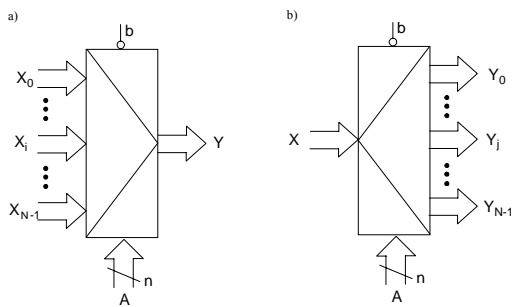
$$-X = \bar{X} + 1$$

Np.: -5

$$\begin{array}{r} 0101 \\ 1010 \\ + \quad 1 \\ \hline 1011 \end{array}$$

47

Bloki komutacyjne



Multiplexer (a)

demultiplexer (b)

$$N = 2^n$$

Multiplexer służy do wybierania jednego z wielu słów wejściowych i przesyłania go na wyjście. Na wyjściu \$Y\$ pojawia się słowo wejściowe wskazane adresem \$A\$ (wg naturalnego kodu binarnego).

Demultiplexer służy do przesyłania słowa wejściowego na jedno z wielu wyjść; numer tego wyjścia jest równy aktualnej wartości adresu.

48

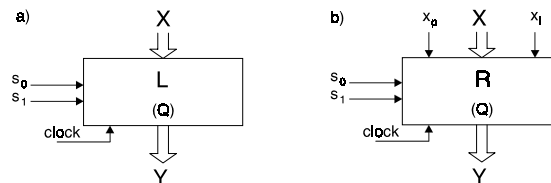
Liczniki i rejestry

Podstawowe sekwencyjne bloki funkcjonalne to liczniki i rejestry. Typowymi mikrooperacjami w tych układach są mikrooperacje wpisywania LOAD i pamiętania HOLD:

$Y := X$ LOAD

$Y := Y$ HOLD

oraz zliczanie (w licznikach) $Y := Y + 1$



Schemat ogólny: a) licznika, b) rejestru;

$s_0 \div s_1$ – wejścia sterujące

W rejestrze, oprócz typowych operacji LOAD i HOLD, występują mikrooperacje przesuwania:

SHR(x_p, Y) – przesunięcie logiczne słowa \$Y\$ w prawo z wpisaniem bitu x_p na zwalnianą pozycję;

SHL(Y, x_i) – przesunięcie logiczne słowa \$Y\$ w lewo z wpisaniem bitu x_i na zwalnianą pozycję.

49

Pamięci

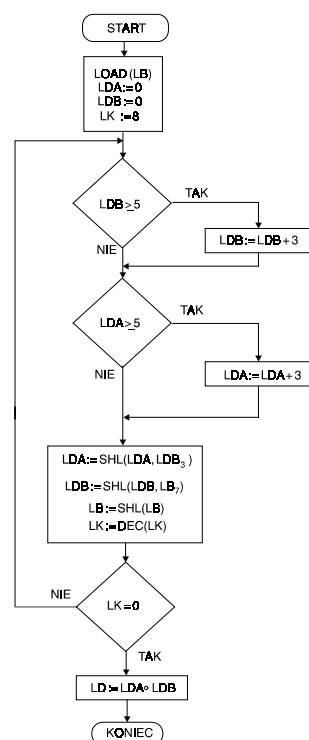
Pamięci stałe typu ROM (Read Only Memory) są układami kombinacyjnymi, w których jest możliwe tylko odczytanie informacji zawartej w strukturze matrycy pamięci. Zapisanie tej informacji odbywa się w procesie technologicznym, analogicznym do programowania układów PLD.

O pamięci, w której wektor adresowy \$A\$ ma długość \$p\$ a wektor wyjściowy \$Y\$ – długość \$n\$ mówimy, że ma pojemność \$m \times n\$ bitów (\$m\$ słów \$n\$-bitowych), przy czym $m = 2^p$.

W pamięciach typu RAM (Random Access Memory) możliwy jest również zapis informacji w postaci binarnych słów o ustalonej długości. Funkcją zapisu i odczytu steruje wejście w/r (write/read). Wyjście \$Y\$ jest jednocześnie wejściem danych zapisywanych do pamięci. W przypadku odczytu pamięć RAM działa podobnie jak pamięć ROM.

50

Algorytm konwersji liczby binarnej na liczbę w kodzie BCD



51

