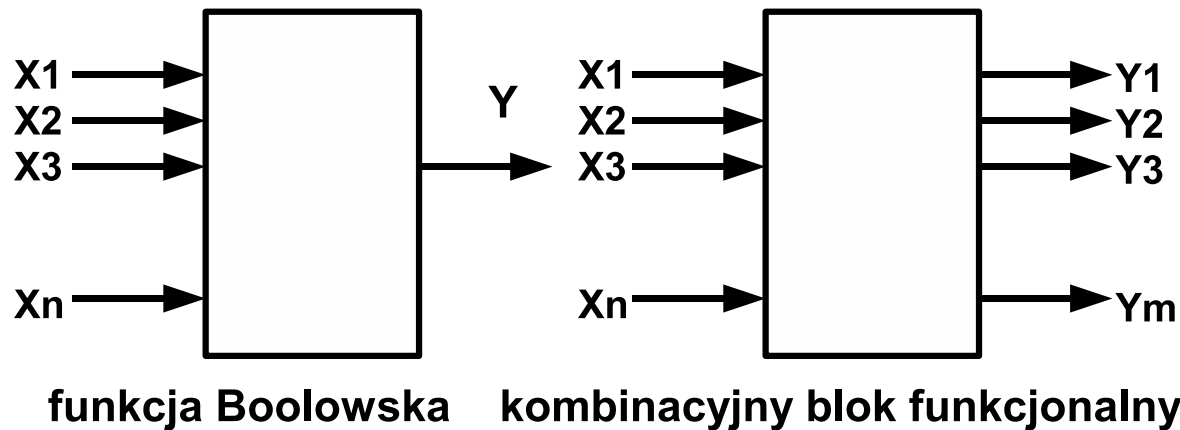


Funkcja Boolowska a kombinacyjny blok funkcjonalny



- *Kombinacyjny blok funkcjonalny* w technice cyfrowej jest układem kombinacyjnym złożonym z n wejściami i m wyjściami, gdzie $m, n = 1, 2, \dots$ są liczbami naturalnymi.
- *Funkcja Boolowska* jest szczególnym przypadkiem *kombinacyjnego bloku funkcjonalnego* - posiada tylko jedno wyjście $m = 1$.

Kombinacyjne bloki funkcjonalne

Kombinacyjne bloki funkcjonalne można podzielić na

- układy komutacyjne:
 - multipleksery MUX,
 - demultipleksery DMUX,
 - konwertery kodów, dekodery DEC,
- układy arytmetyczne:
 - sumatory,
 - komparatory,
 - inne.

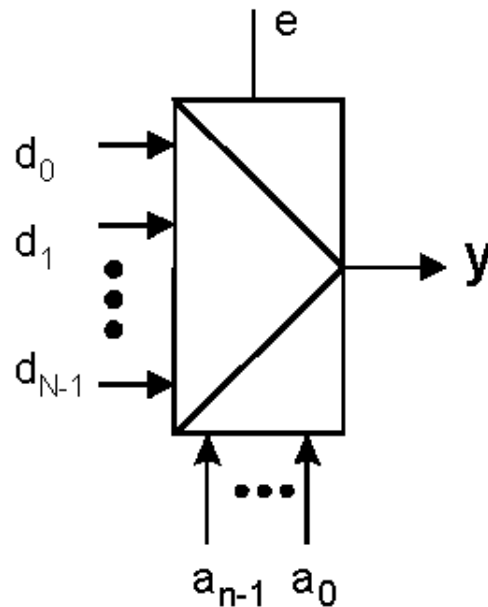
Układy komutacyjne

Multipleksery (MUX),

Demultipleksery (DMUX),

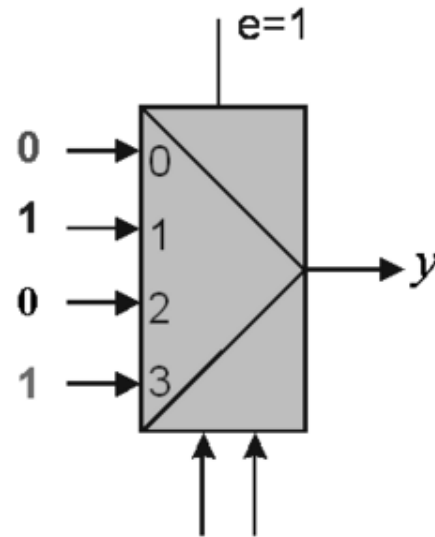
Konwertery kodów, dekodery (DEC).

Multiplekser (MUX)



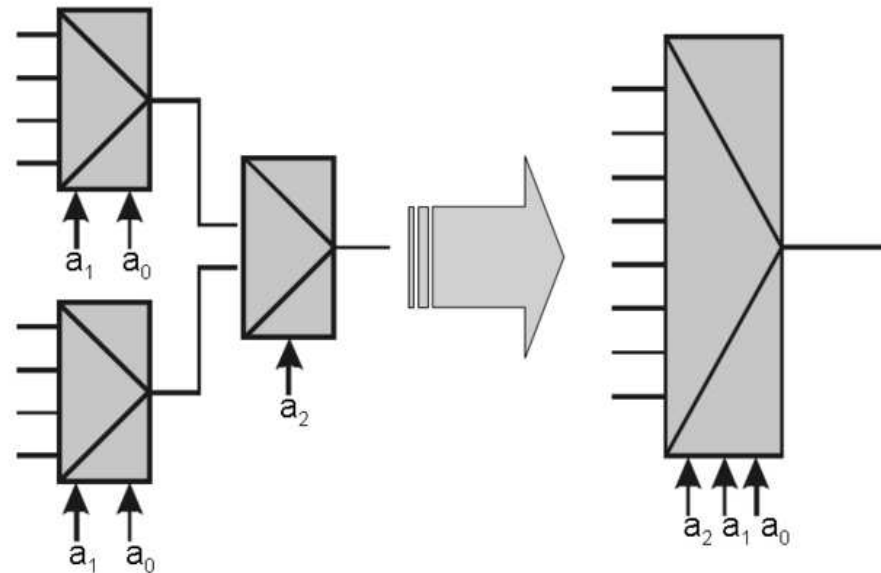
- W multiplekserze wyróżnia się dwa rodzaje wejść - *wejścia adresowe* i *wejścia informacyjne*,
- *Multiplekser* to funkcjonalny blok kombinacyjny, w którym jest n wejść adresowych i $N = 2^n$ wejść informacyjnych, wyjście oraz wejście zezwolenia (enable).

Multiplekser jako przełącznik



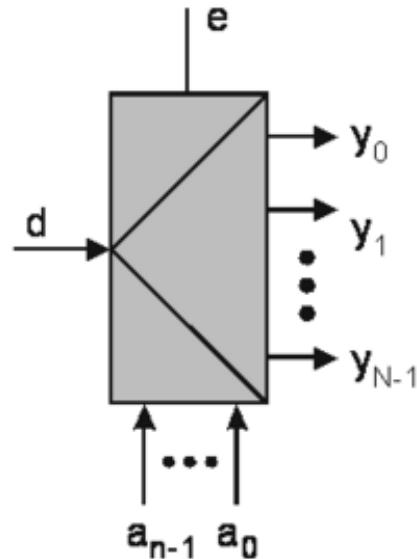
- Multiplekser pracuje jako przełącznik,
- Dany multiplekser realizuje funkcję $y = \overline{a_1}a_0d + a_1a_0d$
- Multiplekser wypisze na wyjściu taki sygnał jaki jest na wejściu informacyjnym wybranym przez wejścia adresowe.

Kaskadowe łączenie multiplexerów



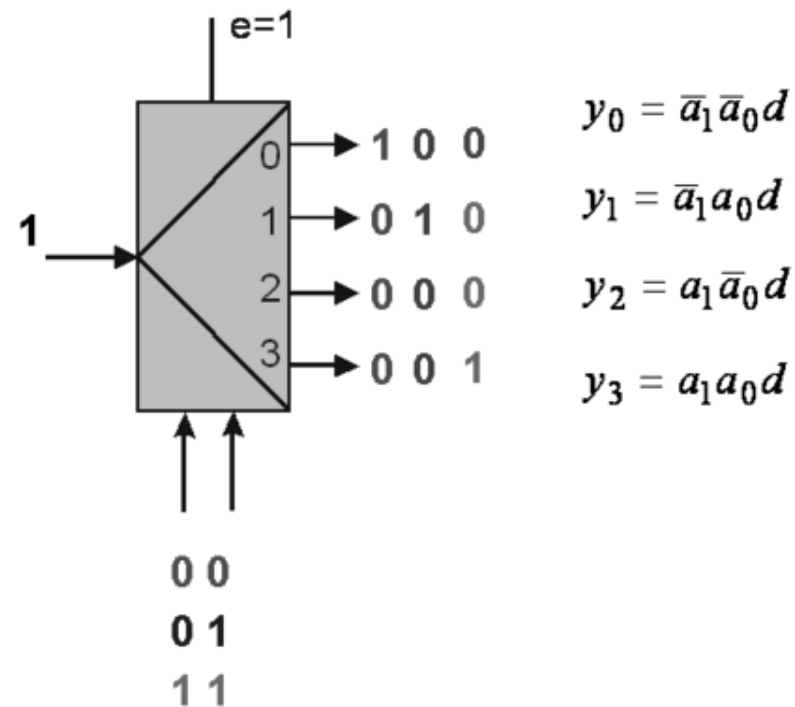
- Liczba wejść informacyjnych multiplexera rośnie wykładniczo dlatego nie realizuje się bezpośrednio multiplexerów o dużej liczbie wejść adresowych,
- Większe multiplexery można budować z mniejszych.

Demultiplekser (DMUX)



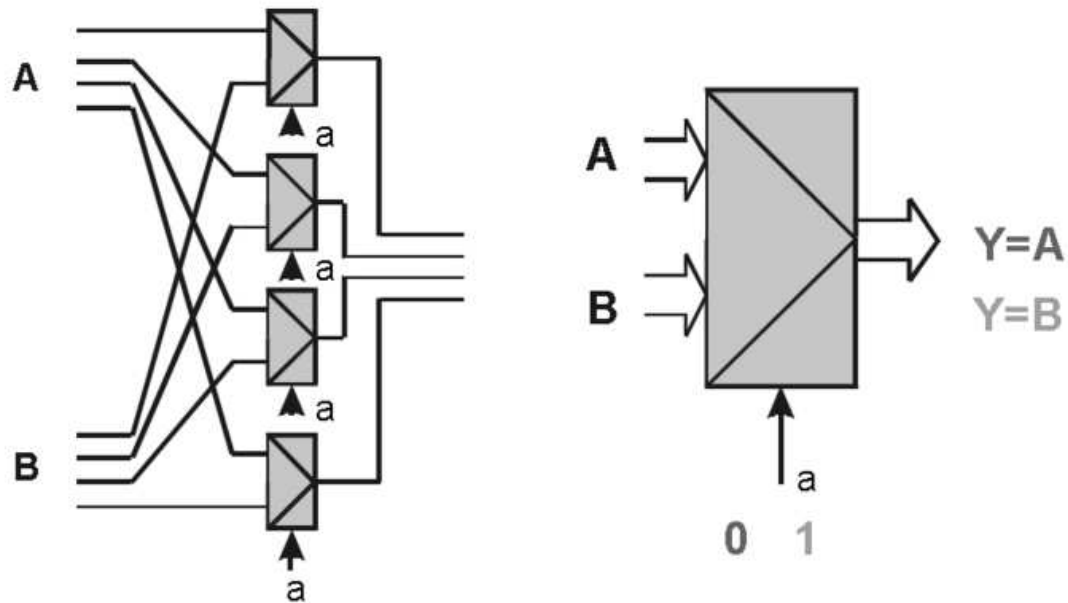
- *Demultiplekser* to układ kombinacyjny o jednym wejściu informacyjnym o n wejść adresowych i $N = 2^n$ wyjściach oraz wejściu zezwalającym .

Demultiplekser jako przełącznik



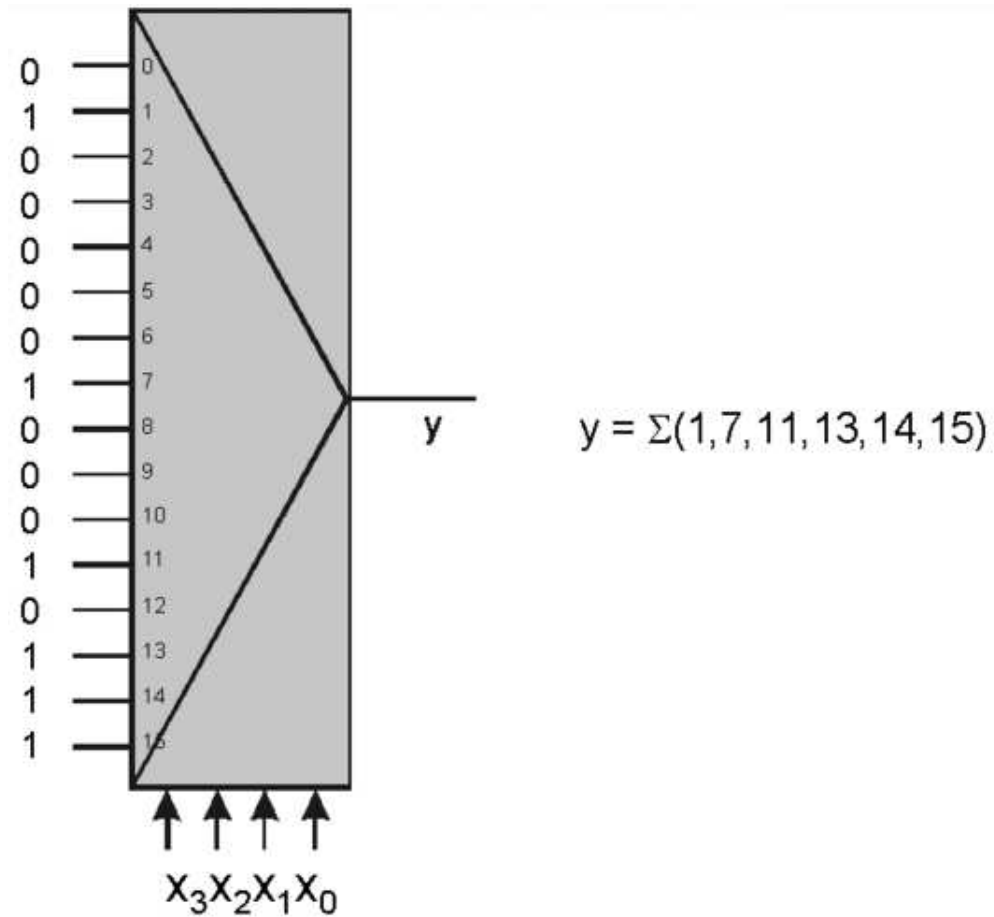
- Demultiplekser pracuje jako przełącznik,
- Demultiplekser wypisze sygnał z wejścia na wyjście wskazane przez stan wejść adresowych.

Multipleksery i demultipleksery grupowe



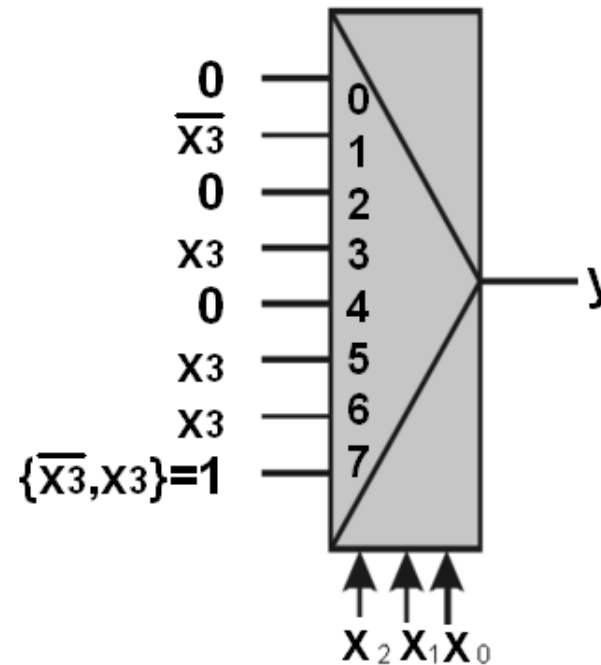
- Realizacja bloków komutacyjnych, czyli elementów umożliwiających proste przełączanie sygnałów, jest najczęściej grupowa,
- Multiplekser grupowy (w tym przypadku 4-bitowy) może być dołączane do szyny w zależności od stanu wejścia adresowego .

Realizacja funkcji Boolowskiej za pomocą multiplexera



Realizacja funkcji Boolowskiej za pomocą multipleksera o trzech wejściach adresowych - cd.

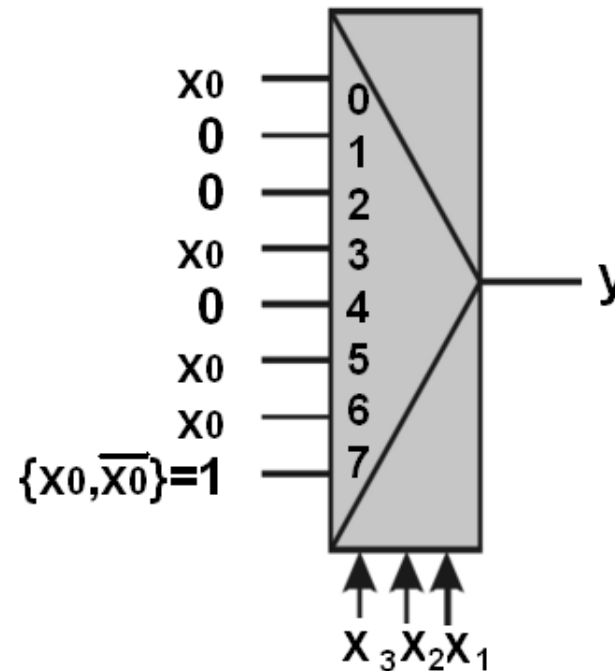
y	x_3	$x_2x_1x_0$	$x_2x_1x_0$
1	0	001	1
7	0	111	7
11	1	011	3
13	1	101	5
14	1	110	6
15	1	111	7



- $y = \sum(1, 7, 11, 13, 14, 15)$
- Na 1 wejściu MUX pojawia się $\overline{x_3}$ - potrzebna negacja.

Realizacja funkcji Boolowskiej za pomocą multipleksera o trzech wejściach adresowych - wybór zmiennych sterujących

y	$x_3x_2x_1$	x_0	$x_3x_2x_1$
1	000	1	0
7	011	1	3
11	101	1	5
13	110	1	6
14	111	0	7
15	111	1	7



- $y = \sum(1, 7, 11, 13, 14, 15)$
- Tym razem negacja jest niepotrzebna.

Realizacja funkcji Boolowskiej za pomocą multiplexera o dwóch wejściach adresowych

$x_3x_2 \setminus x_1x_0$	00	01	11	10
00	0	1	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	0

- $y = \sum(1, 7, 11, 13, 14, 15)$
- Jak wybrać wejścia adresowe ?

cd. - Wybór zmiennych adresowych

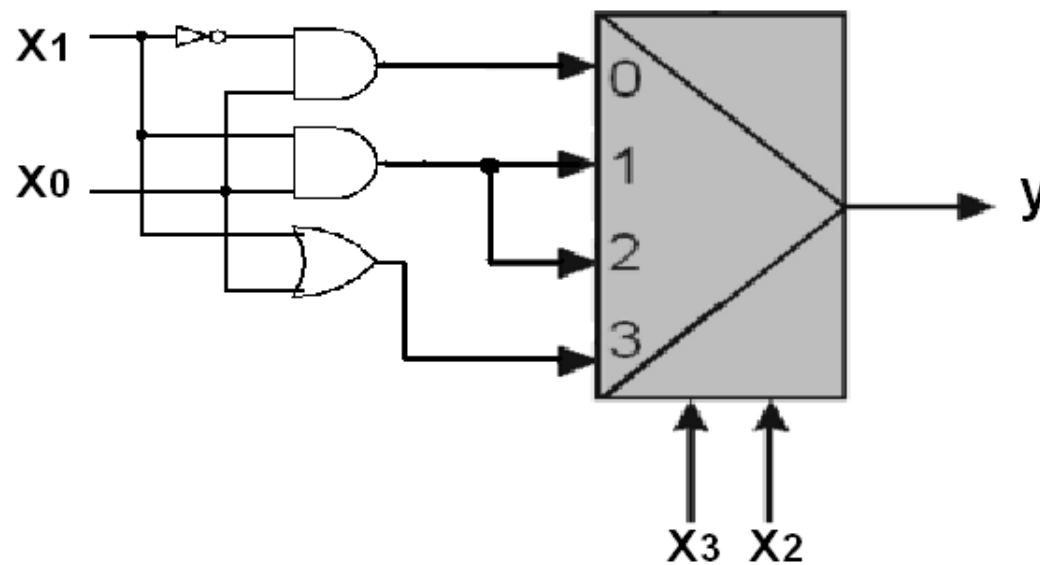
$x_3x_2 \setminus x_1x_0$	00	01	11	10
$x_3x_2 = 00 \rightarrow f(x_1, x_0)$	0	1	0	0
$x_3x_2 = 01 \rightarrow f(x_1, x_0)$	0	0	1	0
$x_3x_2 = 11 \rightarrow f(x_1, x_0)$	0	1	1	1
$x_3x_2 = 10 \rightarrow f(x_1, x_0)$	0	0	1	0

Na wejście adresowe wybraliśmy x_3x_2 wówczas na wejścia informacyjne podajemy wyjście funkcji $f(x_1, x_0)$ opisane poprzez odpowiednie wiersze mapy Karnough-a

- $x_3x_2 = 00 \implies f(x_1, x_0) = \overline{x_1}x_0$
- $x_3x_2 = 01 \implies f(x_1, x_0) = x_1x_0$
- $x_3x_2 = 11 \implies f(x_1, x_0) = x_1 + x_0$
- $x_3x_2 = 10 \implies f(x_1, x_0) = x_1x_0$

cd. - Realizacja

- $x_3x_2 = 00 \implies f(x_1, x_0) = \overline{x_1}x_0$
- $x_3x_2 = 01 \implies f(x_1, x_0) = x_1x_0$
- $x_3x_2 = 11 \implies f(x_1, x_0) = x_1 + x_0$
- $x_3x_2 = 10 \implies f(x_1, x_0) = x_1x_0$



Kody liczbowe i konwertery kodów

Najczęściej używane kody w technice cyfrowej

- **Kod 1 z N** - sposób kodowania, w którym słowa binarne o długości n bitów zawierają zawsze tylko jeden bit o wartości 1. Pozycja jedynki determinuje zakodowaną wartość (kod pozycyjny, bezwagowy)
- **NKB - Naturalny Kod Binarny** - to pozycyjny system liczbowy, w którym podstawą jest liczba 2.
- **Kod Graya** - dwójkowy kodem bezwagowy niepozycyjny, który charakteryzuje się tym, że dwa kolejne słowa kodowe różnią się tylko stanem jednego bitu.
- **kod BCD** (dziesiętny zakodowany dwójkowo) – sposób zapisu liczb polegający na zakodowaniu kolejnych cyfr dziesiętnych liczby dwójkowo przy użyciu czterech bitów.

Kod 1 z N

Wartość dziesiętna	Wartość binarna	Kod 1 z 10
0	0000	1000000000
1	0001	0100000000
2	0010	0010000000
3	0011	0001000000
4	0100	0000100000
5	0101	0000010000
6	0110	0000001000
7	0111	0000000100
8	1000	0000000010
9	1001	0000000001

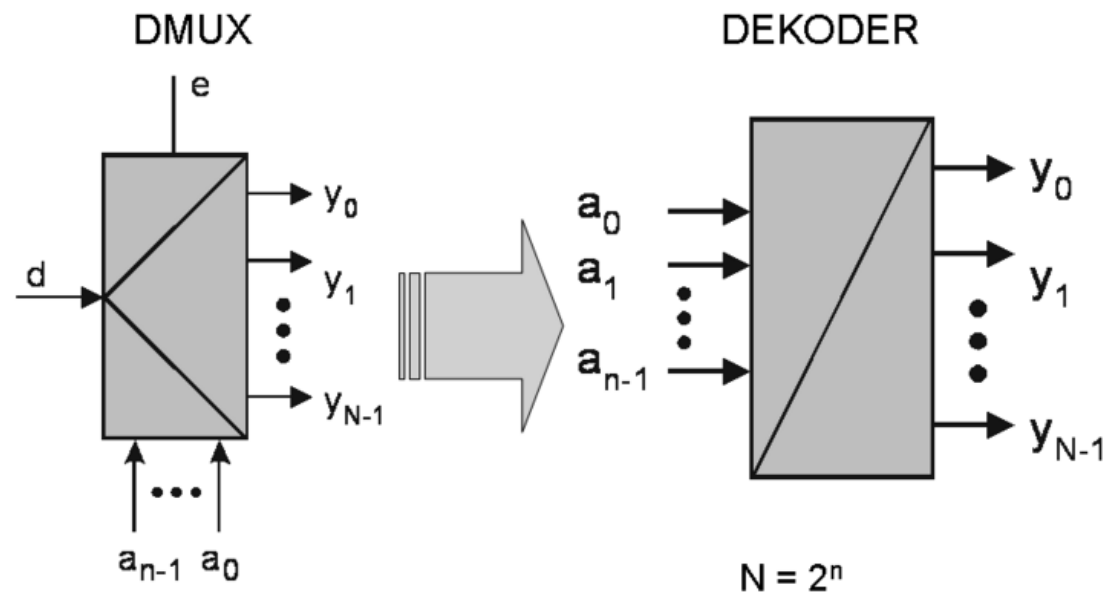
Kod BCD

Cyfra dziesiętna	zapis binarny cyfry
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

np. Liczba 123 składa się z trzech cyfr. Kodując każdą cyfrę binarnie otrzymujemy kod BCD: 0001 0010 0011.

Dekoder

Dekoder zamienia kod NKB na 1 z N .



- Szczególnym przypadkiem demultipleksera jest dekodery, w którym przyjmuje się, że do wejścia d zawsze jest dołączony sygnał o wartości logicznej 1. Wejście to nie jest dostępne na zewnątrz układu.

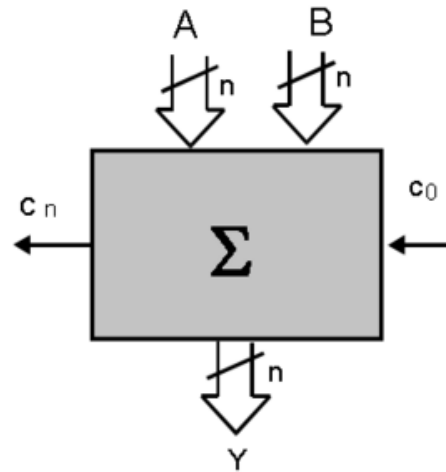
układy arytmetyczne

sumatory,

komparatory,

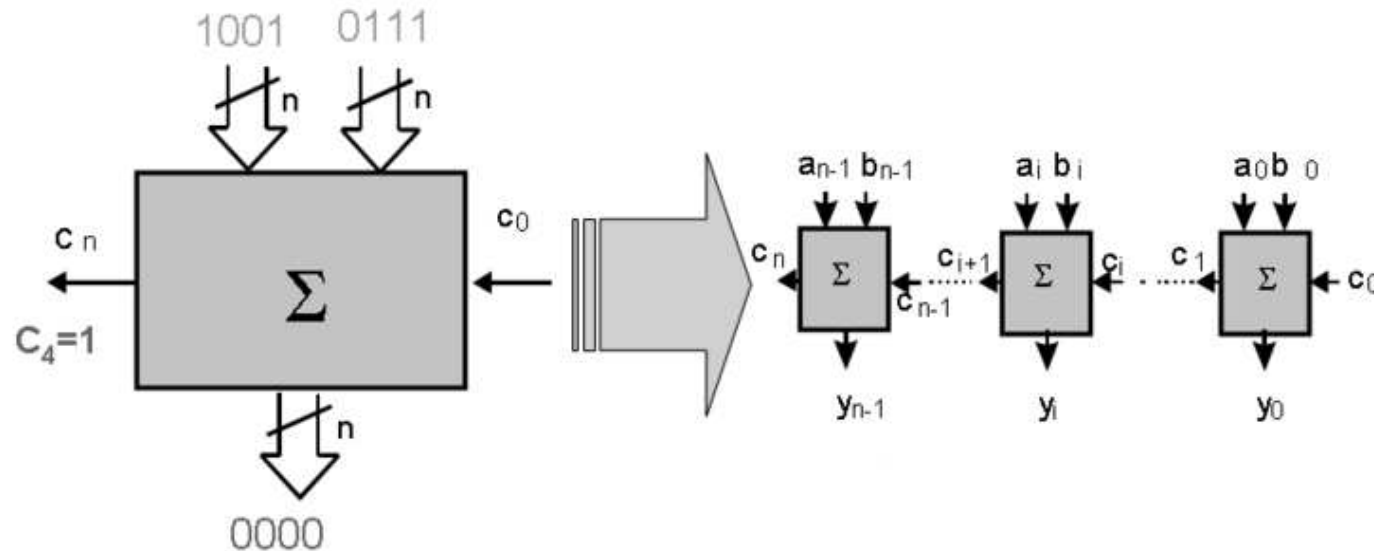
inne.

Sumator



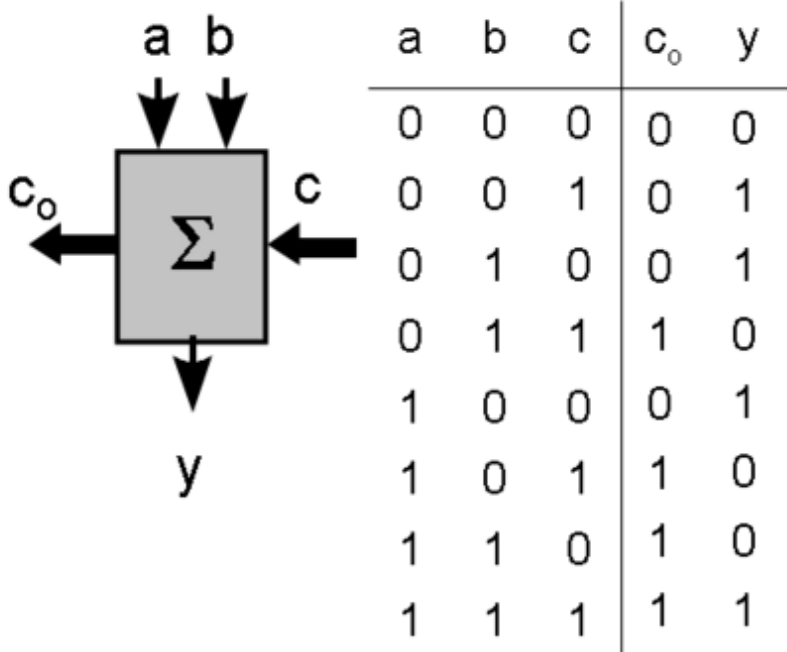
- Operację sumowania arytmetycznego $Y = A + B + c_0$ realizuje sumator. Na wyjściu sumatora powstaje suma n -bitowych liczb binarnych A i B .
- Przypadek przekroczenia zakresu sygnalizowany jest sygnałem przeniesienia c_n .
- Bit przeniesienia można traktować jako najstarszy bit wyniku.

Budowa kaskadowa sumatora



- W najprostszej realizacji sumator jest zbudowany z kaskadowo połączonych sumatorów jednobitowych, o wejściach a_i , b_i i c_i , wyjściach y_i i c_{i+1} .

Budowa sumatora jednobitowego



c \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$y = cab \vee c\bar{a}\bar{b} \vee c\bar{a}b \vee c\bar{a}\bar{b}$$

$$= c(a \oplus b) \vee c(a \oplus b)$$

$$y = c \oplus a \oplus b$$

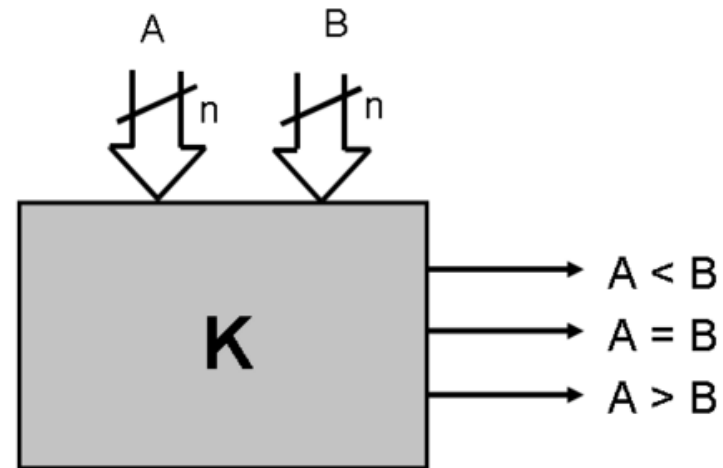
c \ ab	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$c_o = ab \vee c(avb) = ab \vee c(a \oplus b)$$

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

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

Komparator



- Komparator umożliwia porównanie dwóch liczb n -bitowych i określenie czy są sobie równe, a także która z liczb jest większa, a która mniejsza.

Inne układy arytmetyczne

Budowane są również inne bloki arytmetyczne, jak np.:

- układy odejmujące,
- układy mnożące,
- układy dzielenia,
- etc.

Zadania na ćwiczenia

Dana jest funkcja czterech zmiennych wskazana przez prowadzącego

$y = \sum(\dots\dots\dots)$ (dla każdego studenta inna).

1. Za pomocą multipleksera o czterech wejściach adresowych zrealizuj daną funkcję.
2. Za pomocą multipleksera o trzech wejściach adresowych i co najwyżej jednego negatora zrealizuj daną funkcję.
3. Za pomocą multipleksera o dwóch wejściach adresowych i dowolnej liczbie i rodzajach bramek zrealizuj daną funkcję. Zastosuj możliwie małą liczbę bramek.
4. Za pomocą multipleksera o dwóch wejściach adresowych i bramkach NAND zrealizuj daną funkcję. Zastosuj możliwie małą liczbę bramek.