BLOKI FUNKCJONALNE

KOMBINACYJNE

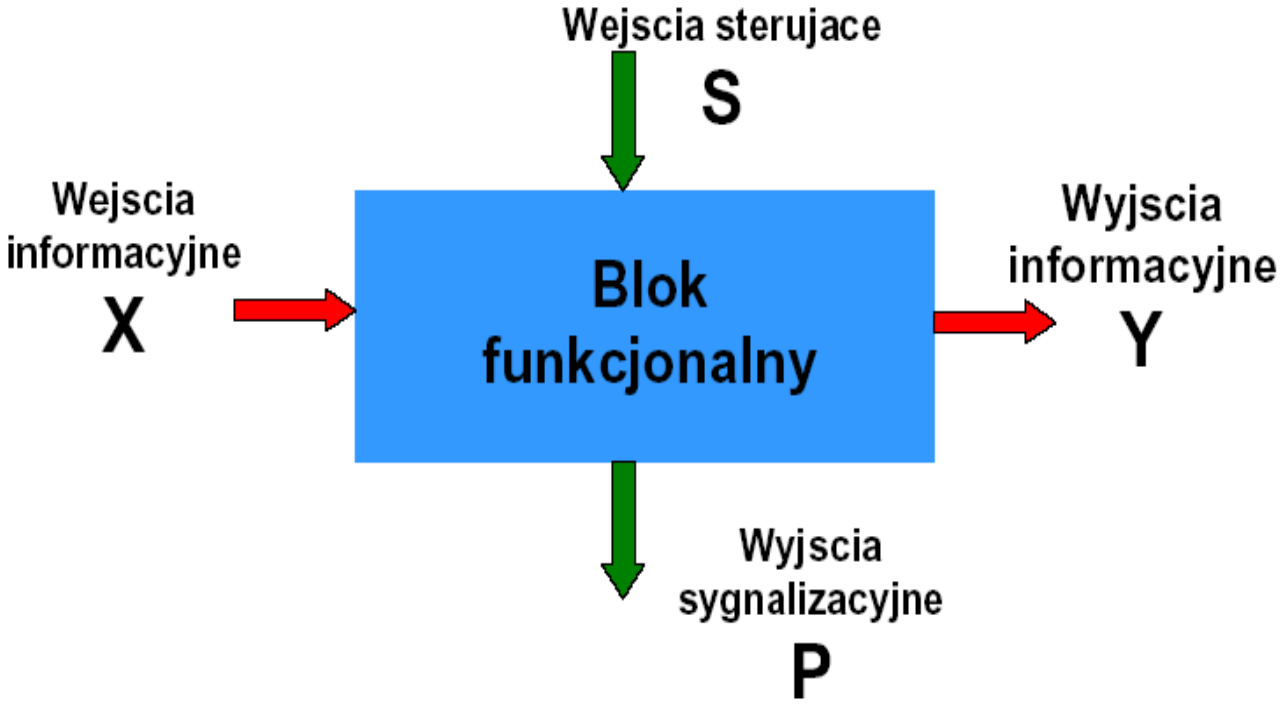
- **UKŁADY ARYTMETYCZNO - LOGICZNE**
  - **SUMATOR**
  - **ALU**
  - **KOMPARATOR**
- **UKŁADY KOMUTACYJNE**
  - **MULTIPLEKSER**
  - **DEMULTIPLEKSER**
  - **DEKODER**
  - **KODER**
  - **TRANSKODER**
  - **MAGISTRALA**

SEKWENCYJNE

- **REJESTRY**
  - **SISO**
  - **SIPO**
  - **PISO**
  - **PIPO**
- **LICZNIKI**
  - **SZEREGOWE**
  - **RÓWNOLEGŁE**

PAMIĘCI

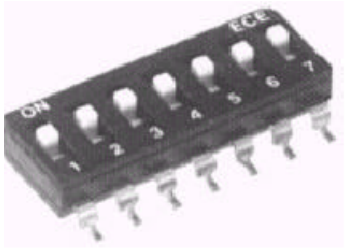
- **ROM**
  - **ROM**
  - **PROM**
  - **EPROM**
  - **EEPROM**
- **RAM**
  - **DRAM**
  - **SRAM**

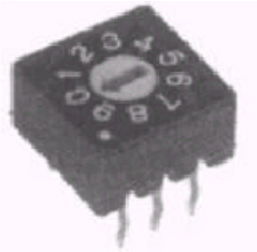


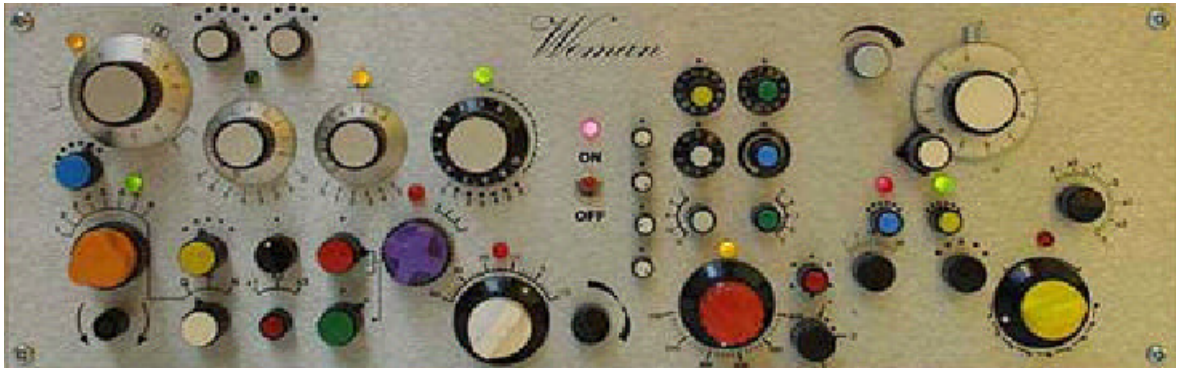
Uklad kombinacyjny $Y = f(X,S)$

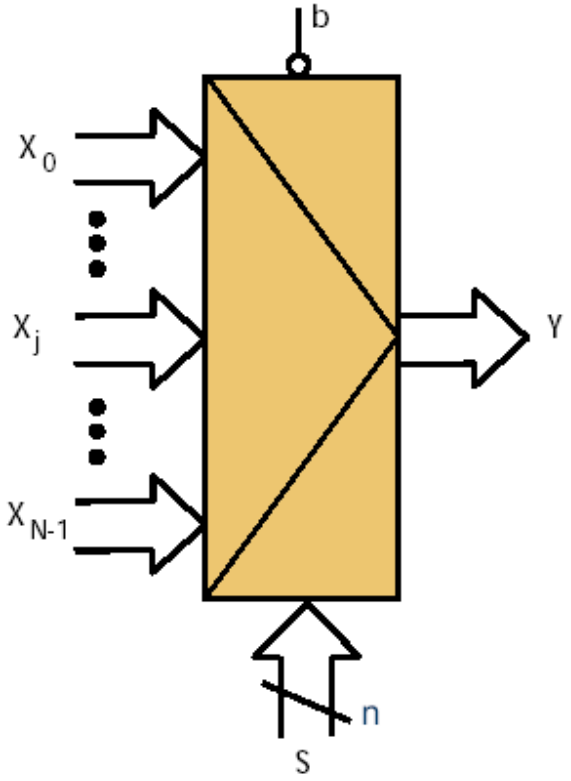
Uklad sekwencyjny $Y = f_s(Y,X)$

MULTIPLEKSERY – PO CO ?

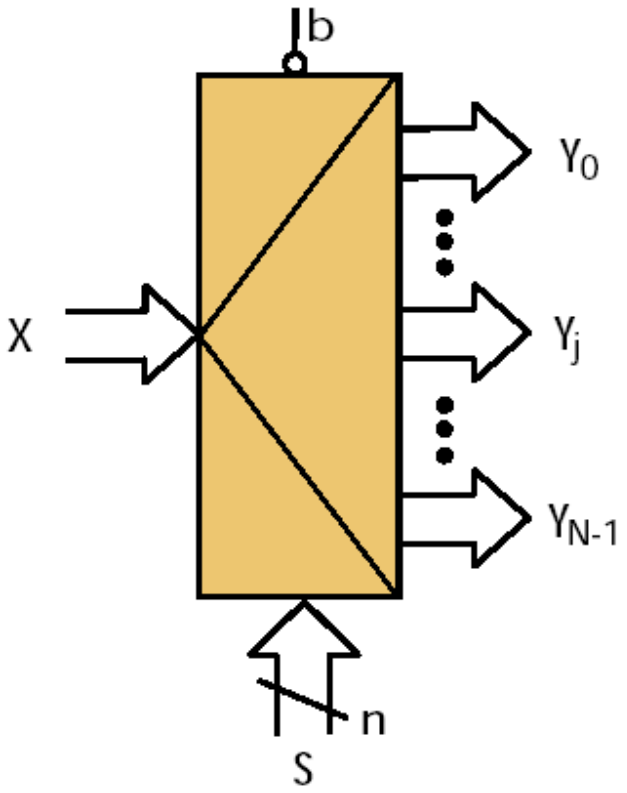






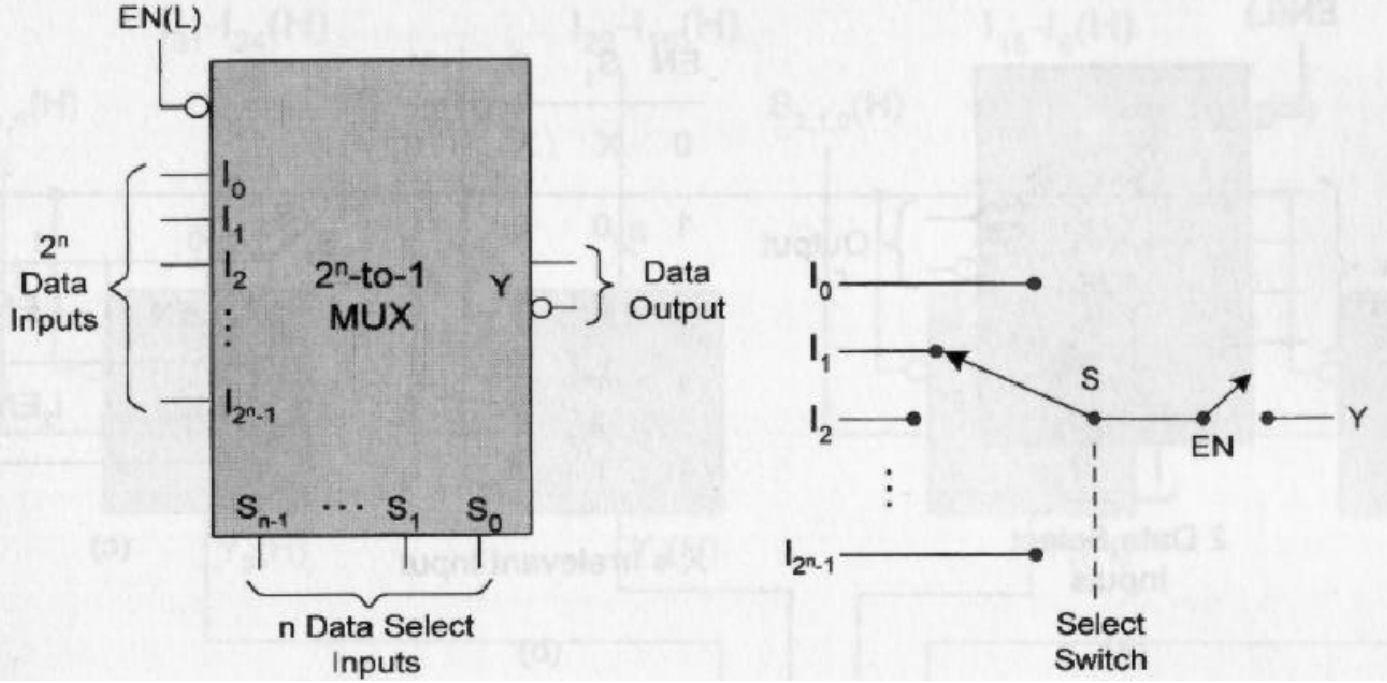


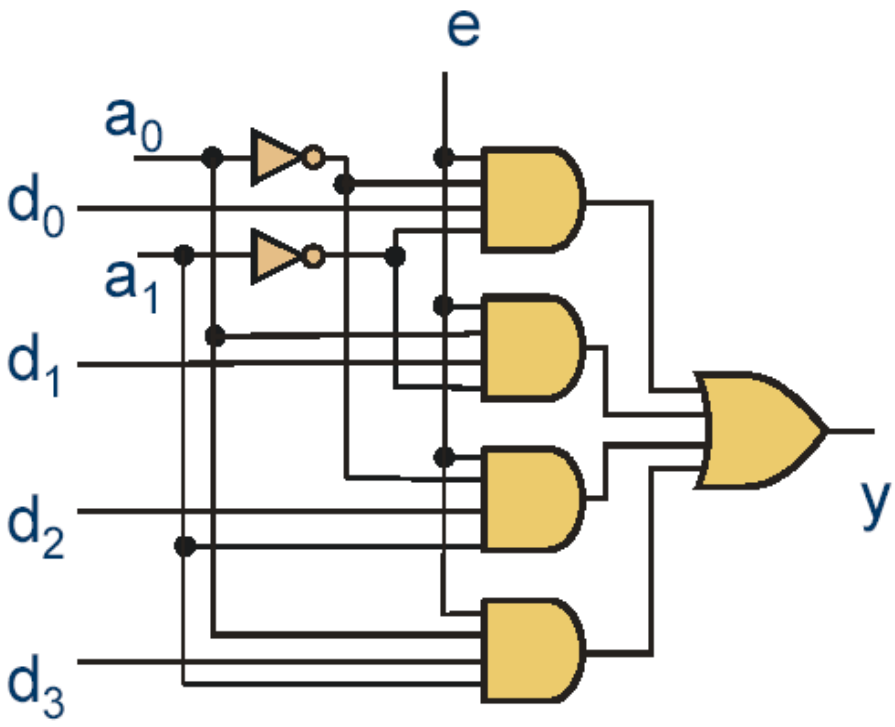
Multiplekser służy do wybierania jednego z wielu słów wejściowych i przesyłania go na wyjście. Na wyjściu Y pojawia się słowo wejściowe wskazane adresem A (wg naturalnego kodu binarnego).



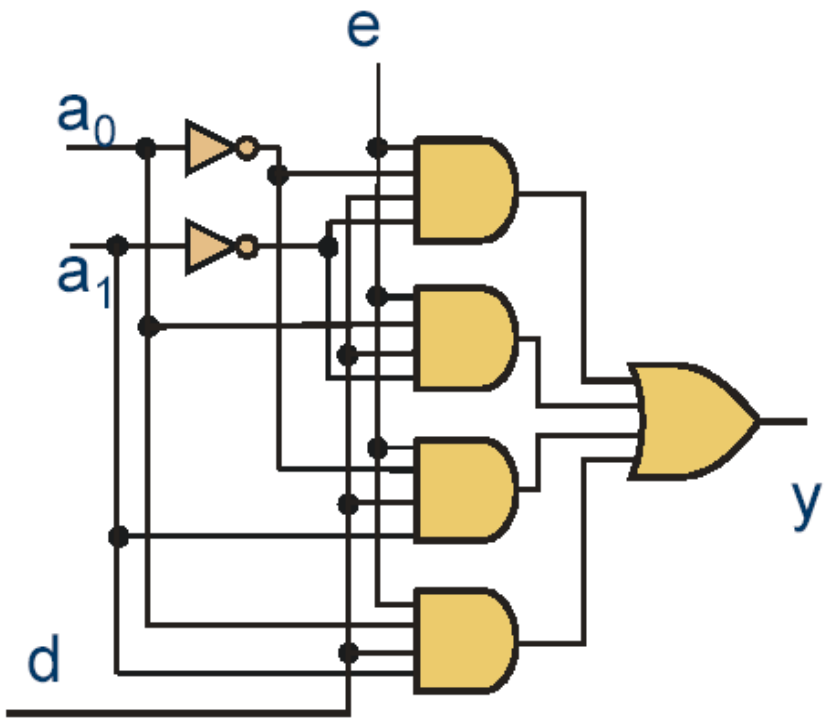
Demultiplekser służy do przesyłania słowa wejściowego na jedno z wielu wyjść; numer tego wyjścia jest równy aktualnej wartości adresu.

MULTIPLEKSER I JEGO SCHEMAT ZASTĘPCZY

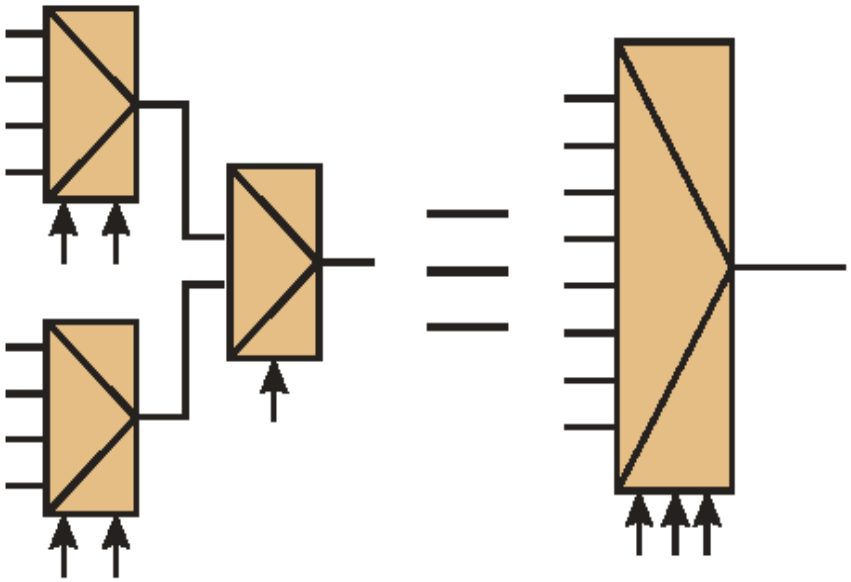




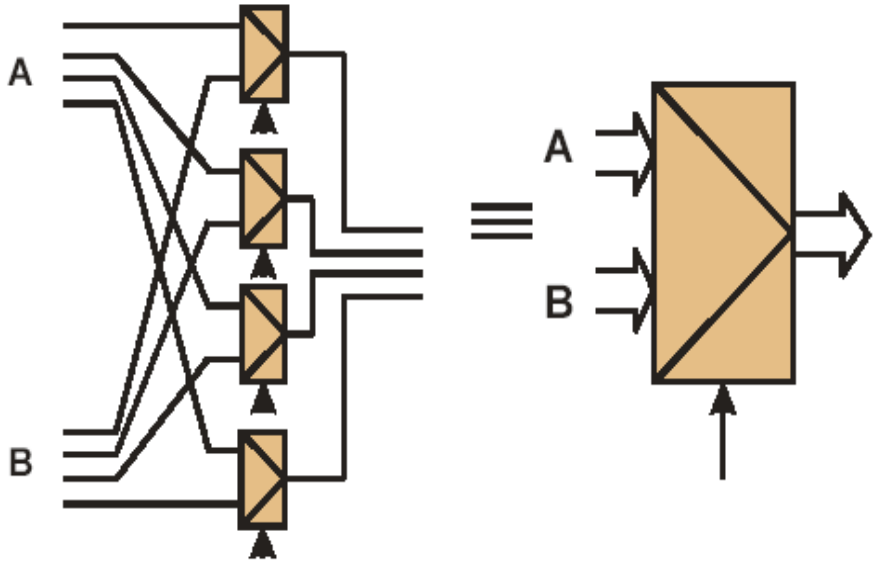
Multiplekser



Demultiplekser



Multiplekser kaskadowy

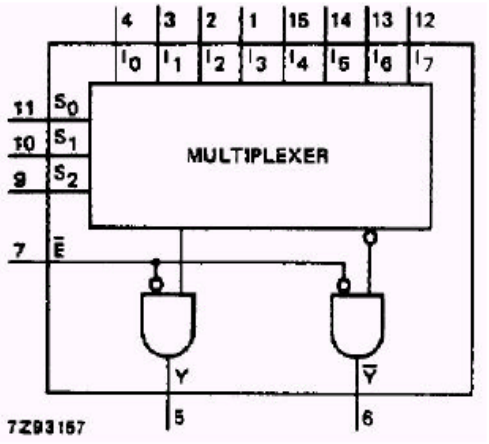


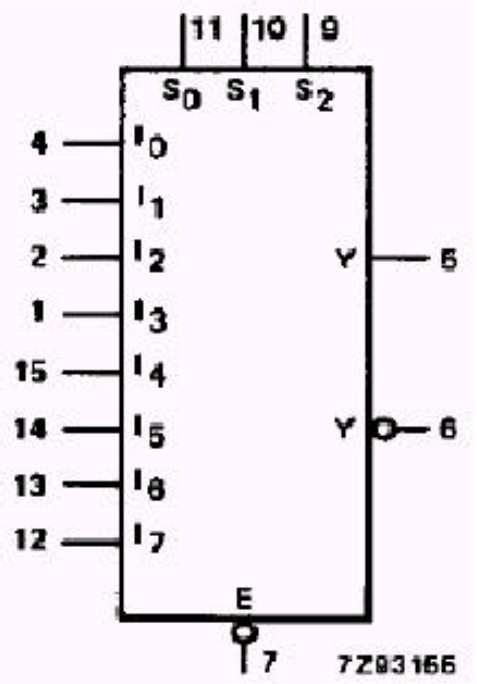
Multiplekser grupowy

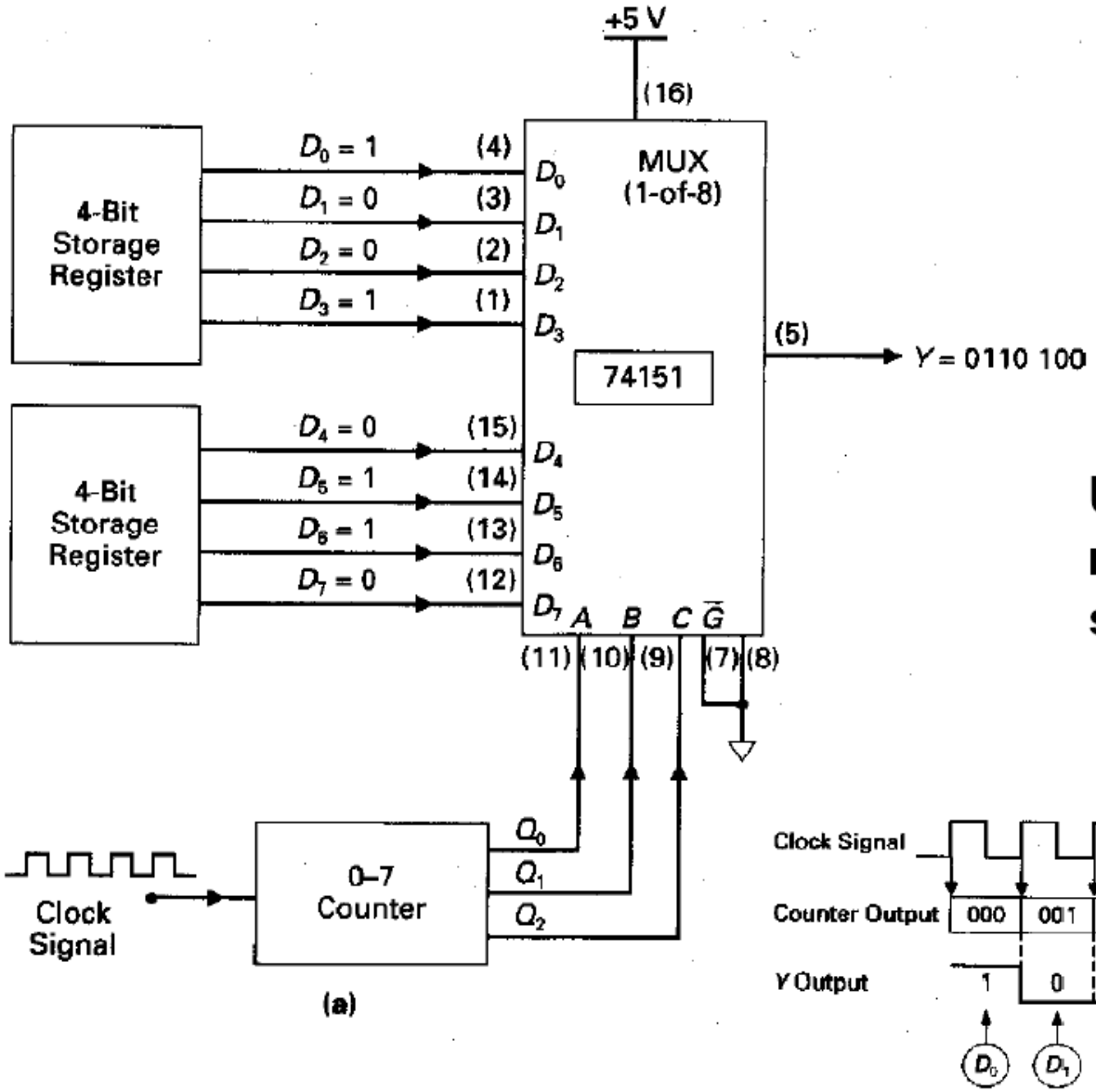
Uklad ‘151

- multiplekser 8 na 1
  (3 wejscia adresowe 8 wejsc informacyjnych)
- wyjscie wprost i zanegowane
- wejscie *„Enable”* (strobujace)

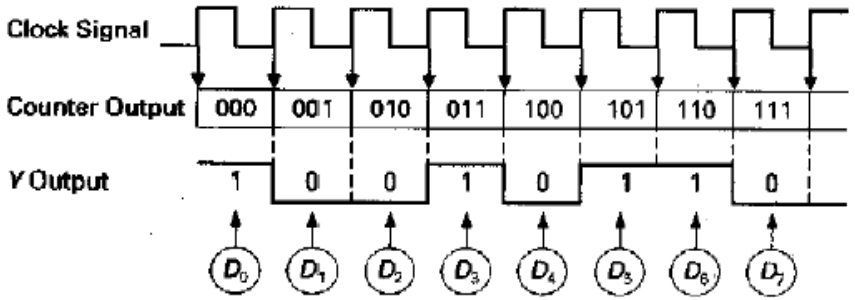
- W ofercie wersje:
  -LS -HC -F -HCT –ACT –LVQ
  ceny od 0,18 do 1 USD







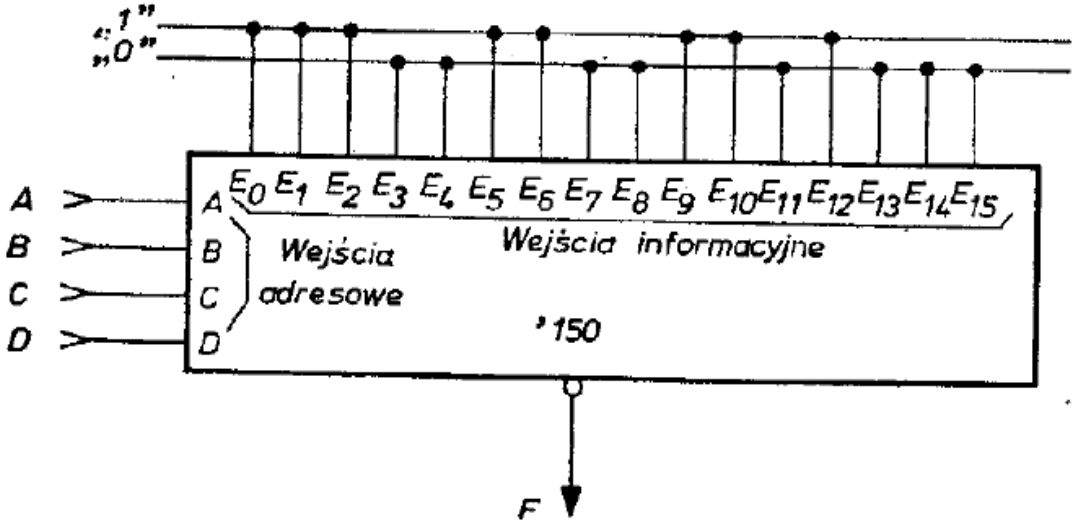
Uklad konwersji równoleglo-szeregowej slowa 8 bitowego



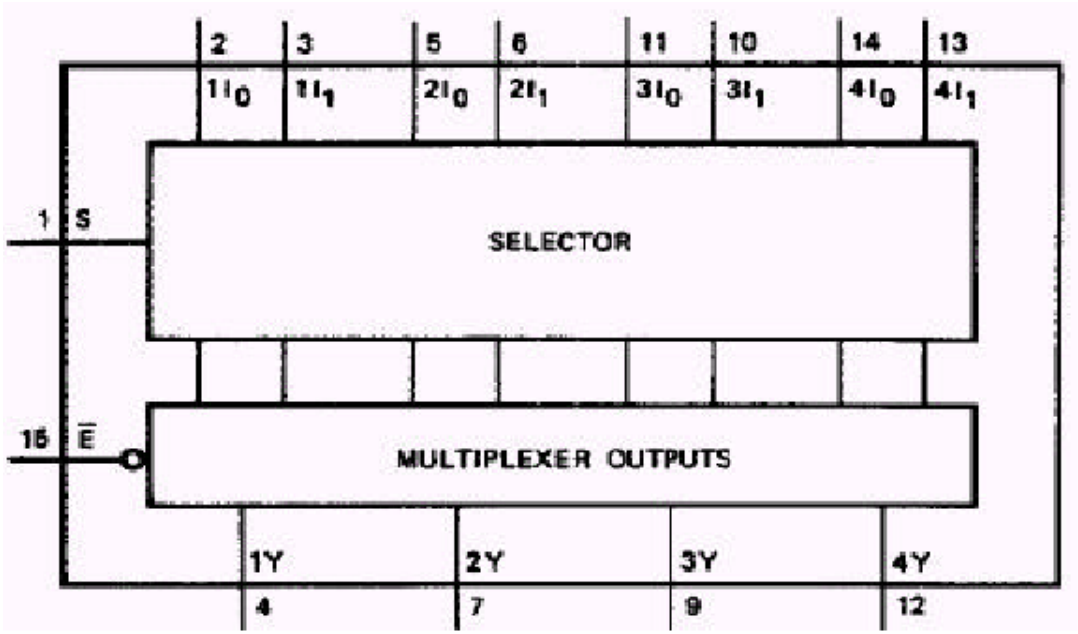
Wejścia				Wyjście
D	C	B	A	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Przy pomocy multipleksera zrealizowac kazda funkcje logiczna jezeli liczba argumentów funkcji równa jest liczbie wejsc adresowych multipleksera

Postac kanoniczna !!!



MULTIPLEKSER ‘157

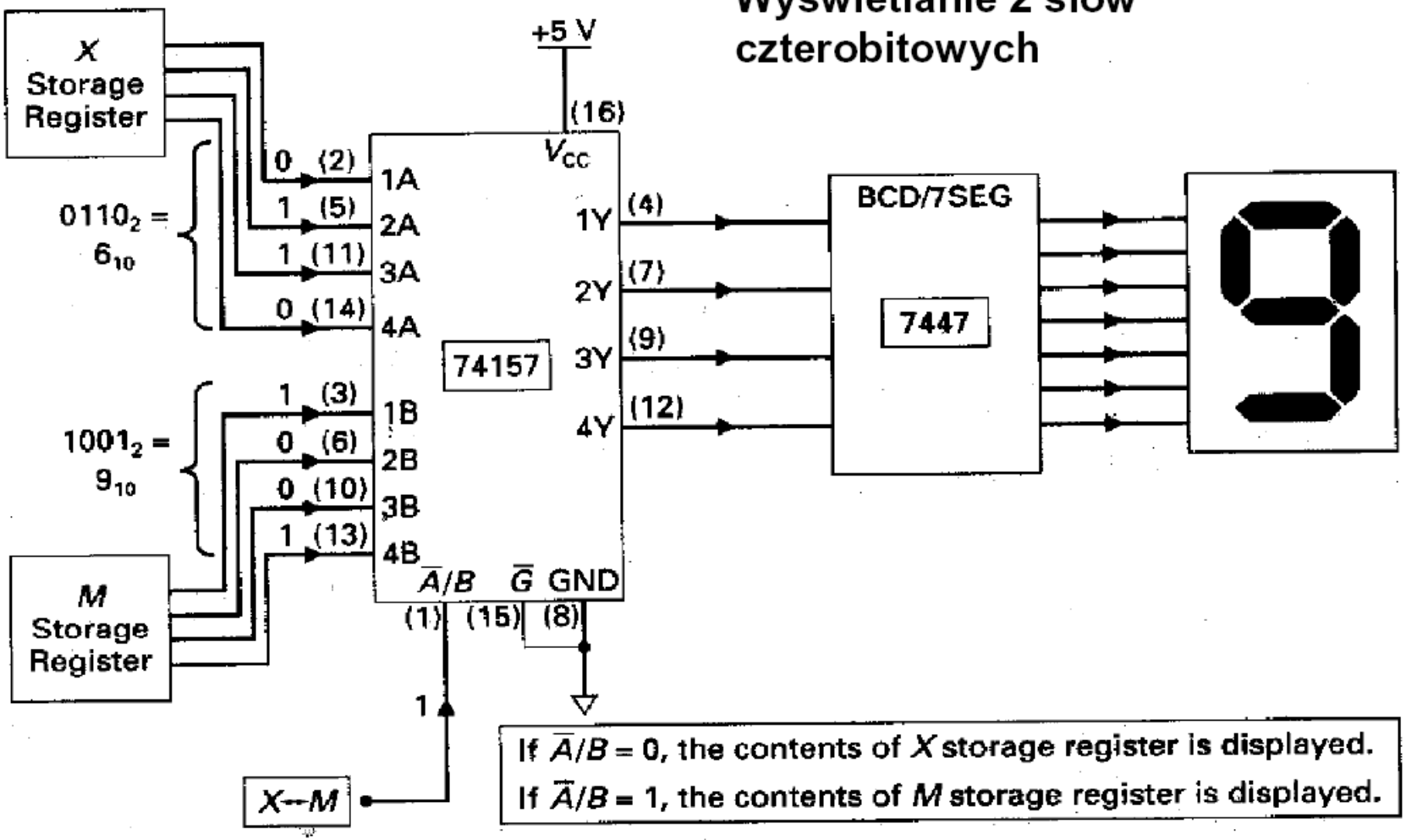


INPUTS				OUTPUT
$\overline{E}$	S	nI_0	nI_1	nY
H	X	X	X	L
L	L	L	X	L
L	L	H	X	H
L	H	X	L	L
L	H	X	H	H

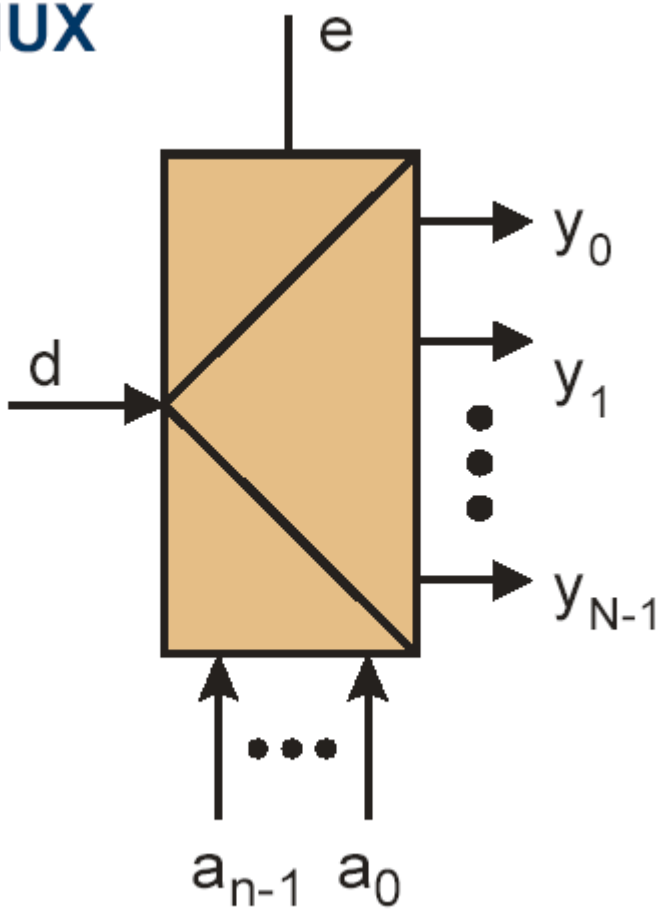
Układ ‘157

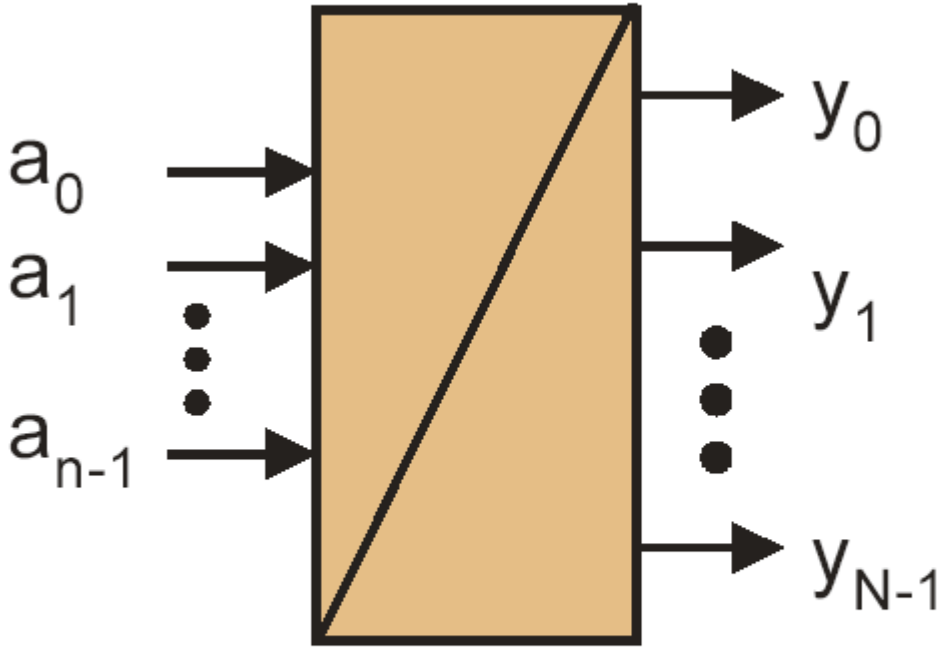
- Poczwórny multiplekser 2 na 1 (1 wspólne wejście adresowe 4x2 wejścia informacyjne
- Wejście *„Enable”* sterujące

Wyswietlanie 2 słów czterobitowych



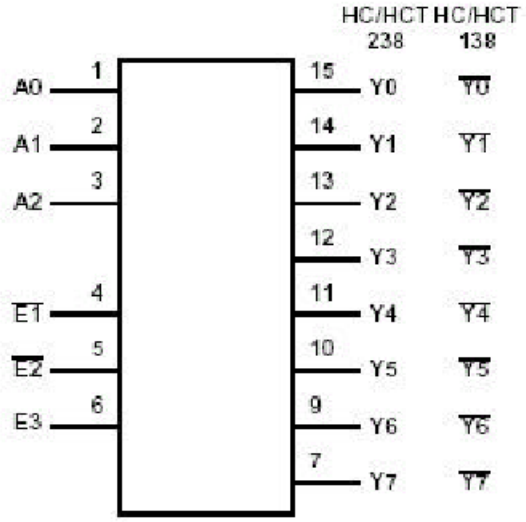
DMUX





$N = 2^n$

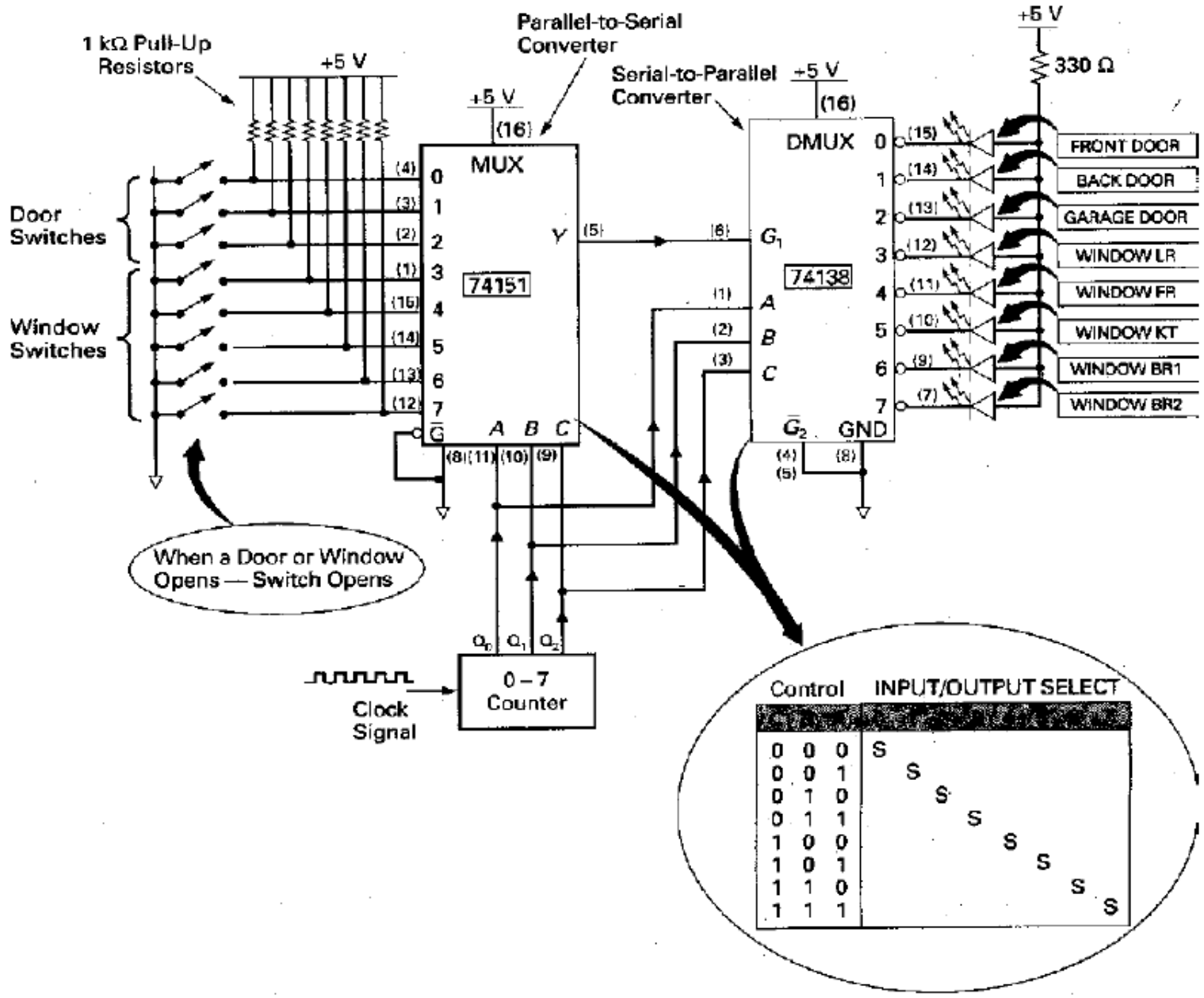
DEMULTIPLEKSER ‘138



Układ ‘138

- 3 na 8 dekoder demultiplekser

INPUTS						OUTPUTS							
ENABLE			ADDRESS										
E3	$\overline{E2}$	$\overline{E1}$	A2	A1	$\overline{A0}$	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$	$\overline{Y4}$	$\overline{Y5}$	$\overline{Y6}$	$\overline{Y7}$
X	X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H


1 kΩ Pull-Up Resistors
+5 V
Parallel-to-Serial Converter
Serial-to-Parallel Converter
+5 V
330 Ω
Door Switches
Window Switches
MUX
74151
DMUX
74138
FRONT DOOR
BACK DOOR
GARAGE DOOR
WINDOW LR
WINDOW FR
WINDOW KT
WINDOW BR1
WINDOW BR2
GND
When a Door or Window Opens — Switch Opens
0 – 7 Counter
Clock Signal
Control
INPUT/OUTPUT SELECT

DEKODERY, KONWERTERY KODÓW

KODER – konwertuje kod „1 z N” na BCD

DEKODER – konwertuje kod BCD na „1 z N”

	NKB (8421)	Aikena (2421)	XS3	1 z 10	abcdefg
0	0000	0000	0011	0000000001	1111110
1	0001	0001	0100	0000000010	0110000
2	0010	0010	0101	0000000100	1101101
3	0011	0011	0110	0000001000	1111001
4	0100	0100	0111	0000010000	0110011
5	0101	1011	1000	0000100000	1011011
6	0110	1100	1001	0001000000	1011111
7	0111	1101	1010	0010000000	1110010
8	1000	1110	1011	0100000000	1111111
9	1001	1111	1100	1000000000	1111011

KODER 1 z 10 na NKB

	1 z 10 y_9 .....y_0	NKB (8421) $b_3\, b_2\, b_1\, b_0$
0	0000000001	0000
1	0000000010	0001
2	0000000100	0010
3	0000001000	0011
4	0000010000	0100
5	0000100000	0101
6	0001000000	0110
7	0010000000	0111
8	0100000000	1000
9	1000000000	1001

RÓWNANIA KODERA:

$$b_0 = y_1 + y_3 + y_5 + y_7 + y_9$$

$$b_1 = y_2 + y_3 + y_6 + y_7$$

$$b_2 = y_4 + y_5 + y_6 + y_7$$

$$b_3 = y_8 + y_9$$

DEKODER NKB na 1 z 10

	NKB (8421) $b_3\, b_2\, b_1\, b_0$	1 z 10 y_9 .....y_0
0	0000	0000000001
1	0001	0000000010
2	0010	0000000100
3	0011	0000001000
4	0100	0000010000
5	0101	0000100000
6	0110	0001000000
7	0111	0010000000
8	1000	0100000000
9	1001	1000000000

DEKODER:

1. Odrzucający fałszywe kombinacje wejściowe

   Równania wprost z tabelki zerojedynkowej.

2. Nieodrzucający fałszywych kombinacji wejściowych

   Tabela Karnaugha, prostsze równania.

1. DEKODER odrzucajacy fałszywe kombinacje wejściowe

$$y_0 = \overline{b_3}\,\overline{b_2}\,\overline{b_1}\,\overline{b_0} \qquad y_4 = \overline{b_3} b_2 \overline{b_1}\,\overline{b_0}$$

$$y_1 = \overline{b_3}\,\overline{b_2}\,\overline{b_1} b_0 \qquad y_5 = \overline{b_3} b_2 \overline{b_1} b_0 \qquad y_8 = b_3 \overline{b_2}\,\overline{b_1}\,\overline{b_0}$$

$$y_2 = \overline{b_3}\,\overline{b_2} b_1 \overline{b_0} \qquad y_6 = \overline{b_3} b_2 b_1 \overline{b_0} \qquad y_9 = b_3 \overline{b_2}\,\overline{b_1} b_0$$

$$y_3 = \overline{b_3}\,\overline{b_2} b_1 b_0 \qquad y_7 = \overline{b_3} b_2 b_1 b_0$$

1. DEKODER nieodrzucajacy fałszywych kombinacji wejściowych

b_3b_2 \ b_1b_0	00	01	11	10
00	y_0	y_1	y_3	y_2
01	y_4	y_5	y_7	y_6
11	-	-	-	-
10	y_8	y_9	-	-

D

$$y_0 = \overline{b_3}\,\overline{b_2}\,\overline{b_1}\,\overline{b_0} \qquad y_5 = b_2\overline{b_1}b_0$$

$$y_1 = \overline{b_3}\,\overline{b_2}\,\overline{b_1}b_0 \qquad y_6 = b_2b_1\overline{b_0}$$

$$y_2 = \overline{b_2}b_1\overline{b_0} \qquad y_7 = b_2b_1b_0$$

$$y_3 = \overline{b_2}b_1b_0 \qquad y_8 = b_3\overline{b_0}$$

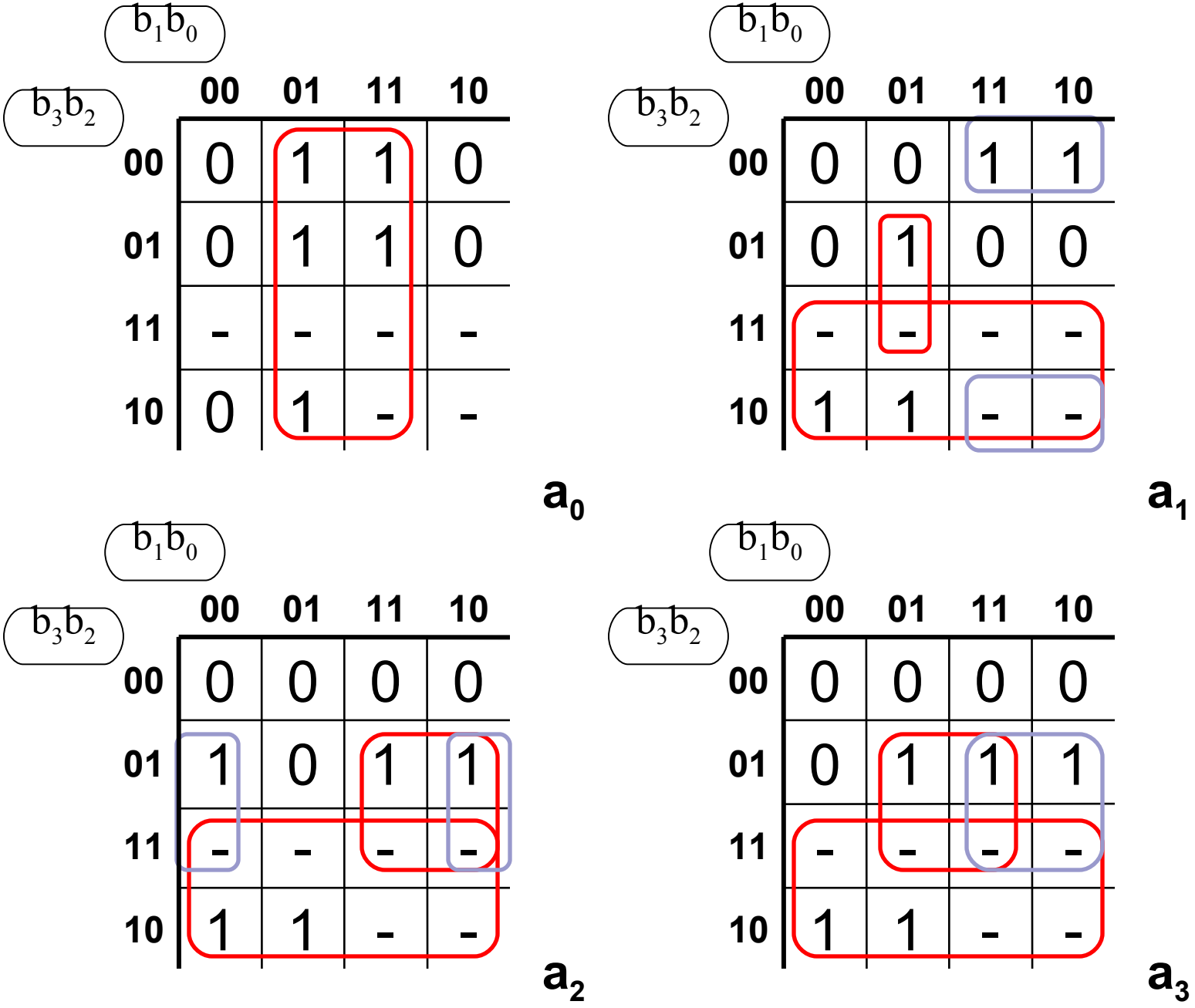
$$y_4 = b_2\overline{b_1}\,\overline{b_0} \qquad y_9 = b_3b_0$$

KONWERTER KODU (TRANSKODER) zamiana kodu o określonej pojemności na inny

NKB (8421) $b_3\,b_2\,b_1\,b_0$	Aikena (2421) $a_3\,a_2\,a_1\,a_0$
0000	0000
0001	0001
0010	0010
0011	0011
0100	0100
0101	1011
0110	1100
0111	1101
1000	1110
1001	1111

PRZYKŁAD 1:

konwerter kodu NKB na Aikena


b1b0
b3b2
00 01 11 10
00 0 1 1 0
01 0 1 1 0
11 - - - -
10 0 1 - -
a0
b1b0
b3b2
00 01 11 10
00 0 0 1 1
01 0 1 0 0
11 - - - -
10 1 1 - -
a1
b1b0
b3b2
00 01 11 10
00 0 0 0 0
01 1 0 1 1
11 - - - -
10 1 1 - -
a2
b1b0
b3b2
00 01 11 10
00 0 0 0 0
01 0 1 1 1
11 - - - -
10 1 1 - -
a3

RÓWNANIA TRANSKODERA:

$$a_0 = b_0$$

$$a_1 = b_3 + b_1\overline{b_2} + b_0\overline{b_1}b_2$$

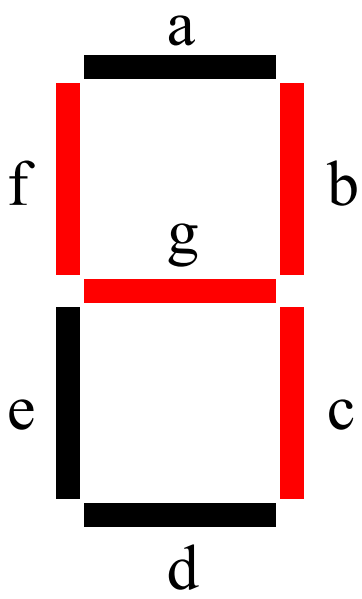
$$a_2 = b_3 + b_1 b_2 + \overline{b_0}b_2$$

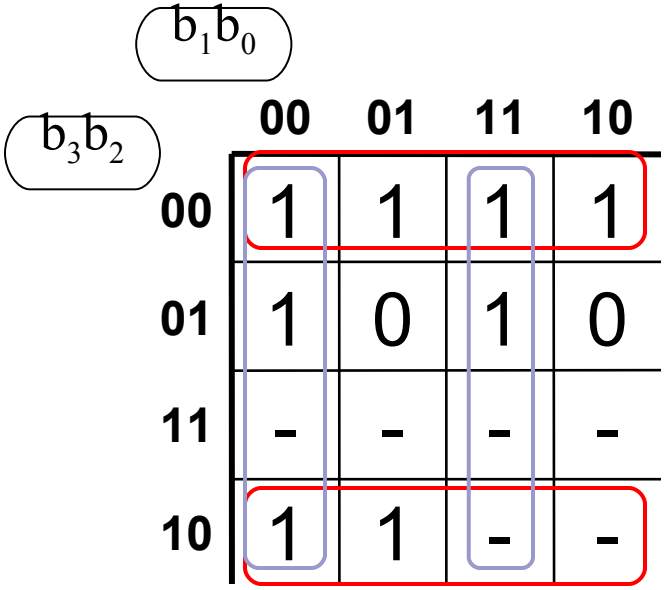
$$a_3 = b_3 + b_0 b_2 + b_1 b_2$$

PRZYKŁAD 2:

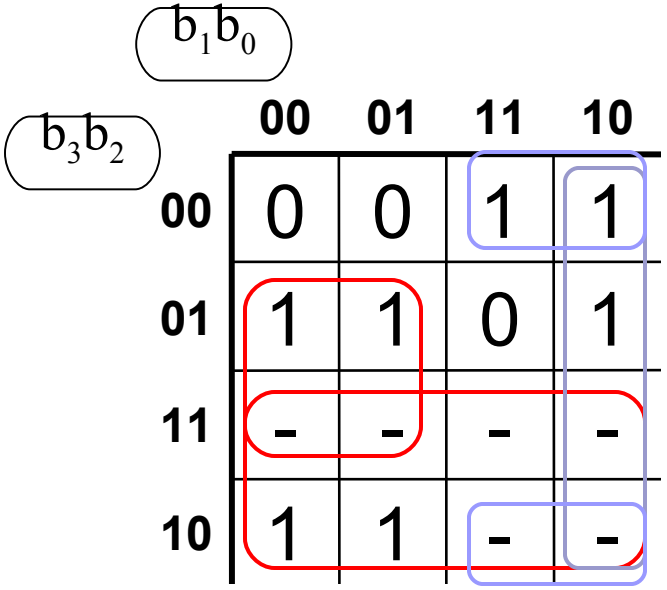
konwerter kodu NKB na 7-segmentowy

NKB (8421) $b_3\,b_2\,b_1\,b_0$	7-segment abcdefg
0000	1111110
0001	0110000
0010	1101101
0011	1111001
0100	0110011
0101	1011011
0110	1011111
0111	1110010
1000	1111111
1001	1111011





b



g

RÓWNANIA TRANSKODERA:

$$b = \overline{b_2} + \overline{b_0}\,\overline{b_1} + b_0 b_1$$

$$g = b_3 + \overline{b_1} b_2 + \overline{b_0} b_1 + b_1 \overline{b_2}$$

UKŁADY ARYTMETYCZNE
(sumatory, akumulatory i komparatory)

Podział sumatorów:

- *dwójkowe*
- *dziesiętne (na liczbach dziesiętnych zakodowanych binarnie)*

Inny podział:

- *szeregowe*
- *równoległe*
- *szeregowo - równoległe*

Akumulator:

sumator, w którym następuje dodanie liczby wprowadzanej na wejście do jego aktualnej zawartości (do budowy wykorzystuje się sumatory i elementy pamięci)

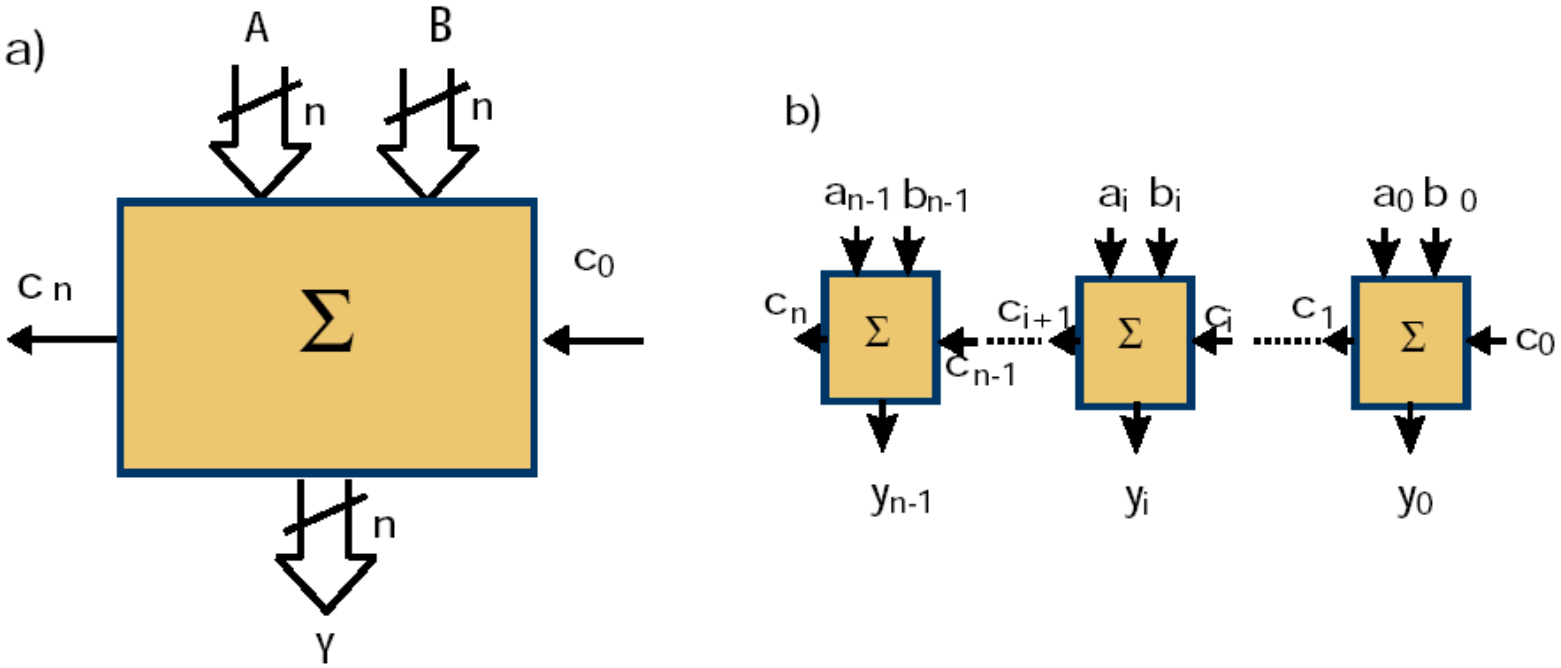
Wprowadzanie informacji do akumulatora:

- *równolegle*
- *szeregowo*

Komparatory:

służą do porównywania liczb binarnych

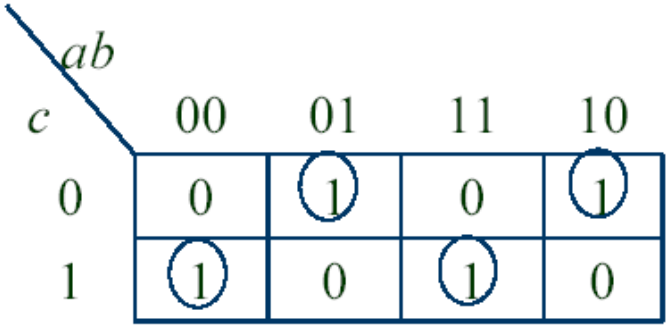
Najprostszy sumator – *ripple carry adder*



a	b	c	c_o	y
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

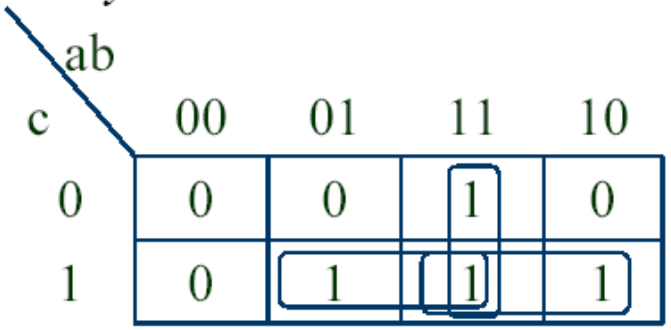
$$y_i = a_i \oplus b_i \oplus c_i$$

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



$$y = cab \vee c\bar{a}\bar{b} \vee \bar{c}a\bar{b} \vee \bar{c}ab = c\overline{(a \oplus b)} \vee \bar{c}(a \oplus b)$$

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



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

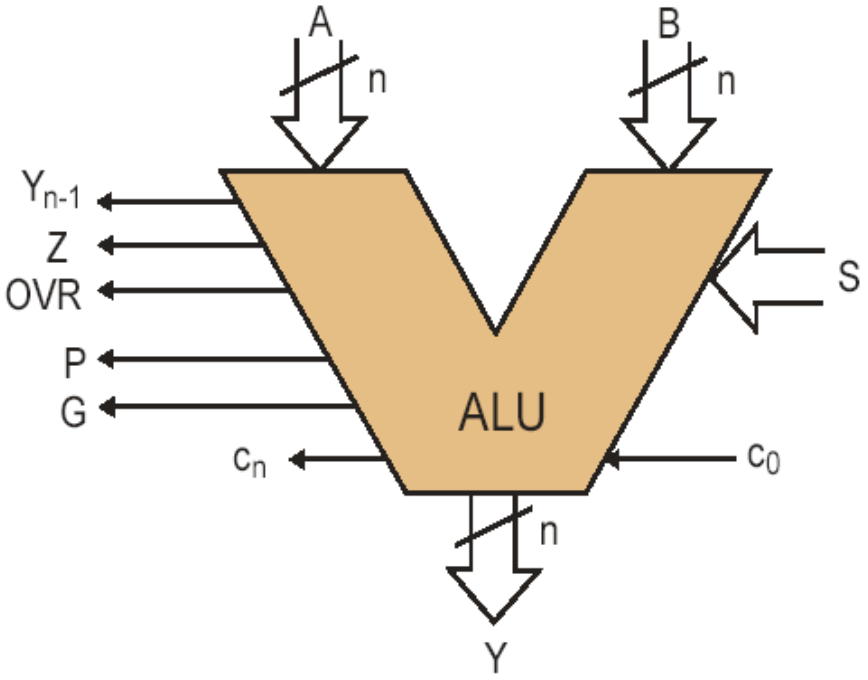
ZASTOSOWANIA - ALU

Jednostka arytmetyczno-logiczna

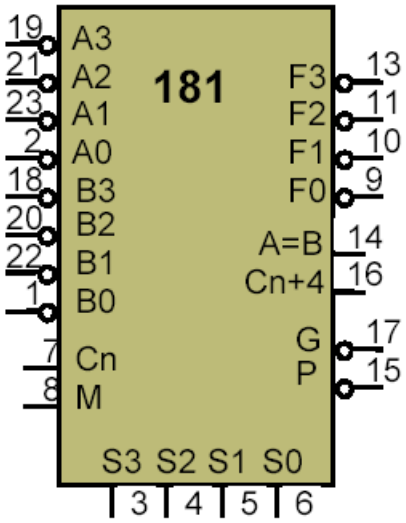
Arytmometr (układ wykonawczy: mikrokontrolera, procesora sygnałowego)

Inne układy arytmetyczne:

układy mnożące
układy kryptograficzne



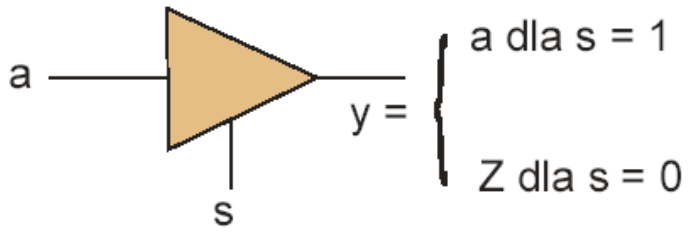
...są budowane z sumatorów



	M = 1	M = 0	
	Funkcje logiczne	Funkcje arytmetyczne	
S3 S2 S1 S0		$C_n = 0$	$C_n = 1$
0 0 0 0	$\overline{A}$	A - 1	A
0 0 0 1	$\overline{A \wedge B}$	AB - 1	AB
0 0 1 0	$\overline{A} + B$	$A\overline{B}$ -1	$A\overline{B}$
0 0 1 1	1	-1	0
0 1 0 0	$\overline{A \vee B}$	$A + (A \vee \overline{B})$	$A + (A \vee \overline{B}) + 1$
0 1 0 1	$\overline{B}$	$AB + (A \vee \overline{B})$	$AB + (A \vee \overline{B}) + 1$
0 1 1 0	$\overline{A \oplus B}$	A - B - 1	$(A \vee \overline{B}) + 1$
0 1 1 1	$A + \overline{B}$	$A + \overline{B}$	A - B
1 0 0 0	$\overline{A}B$	$A + (A \vee B)$	$(A \vee \overline{B}) + 1$
1 0 0 1	$A \oplus B$	A + B	$A + (A \vee B) + 1$
1 0 1 0	B	$A\overline{B} + (A \vee B)$	$A\overline{B} + (A \vee B) + 1$
1 0 1 1	$A \vee B$	$A \vee B$	$(A \vee B) + 1$
1 1 0 0	0	A	A + A + 1
1 1 0 1	$A\overline{B}$	AB + A	AB + A + 1
1 1 1 0	AB	$A\overline{B} + A$	$A\overline{B} + A + 1$
1 1 1 1	A	A	A + 1

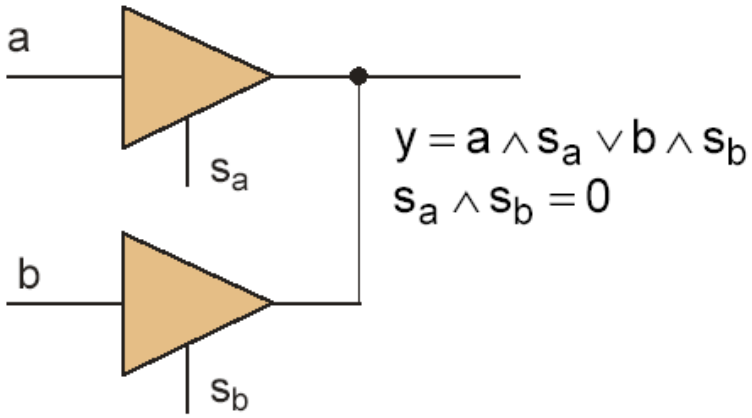
MAGISTRALE (SZYNY)

Budowane z elementów trójstanowych



Z - stan wysokiej
impedancji wyjściowej

Bramka trójstanowa



Łączenie bramek

Szyna zbudowana z bram trójstanowych sterowanych dekoderem