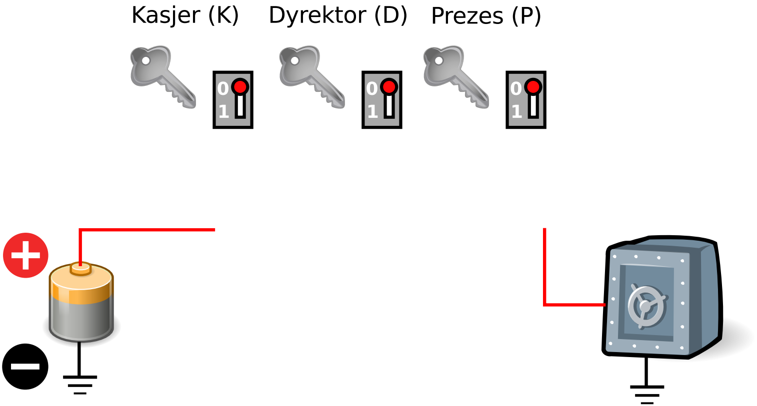
Technologie Informacyjne

Układy przełączające

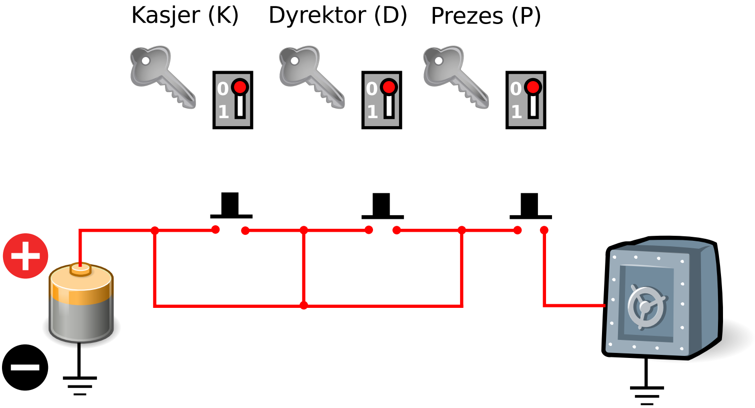
Adam Krasuski

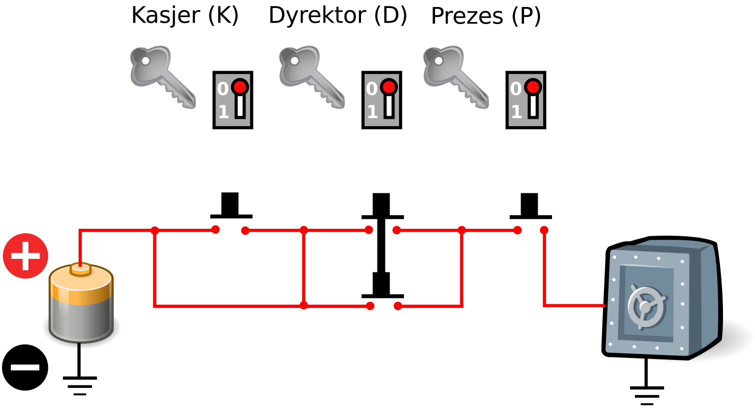
Szkoła Główna Służby Pożarniczej
Zakład Informatyki i Łączności

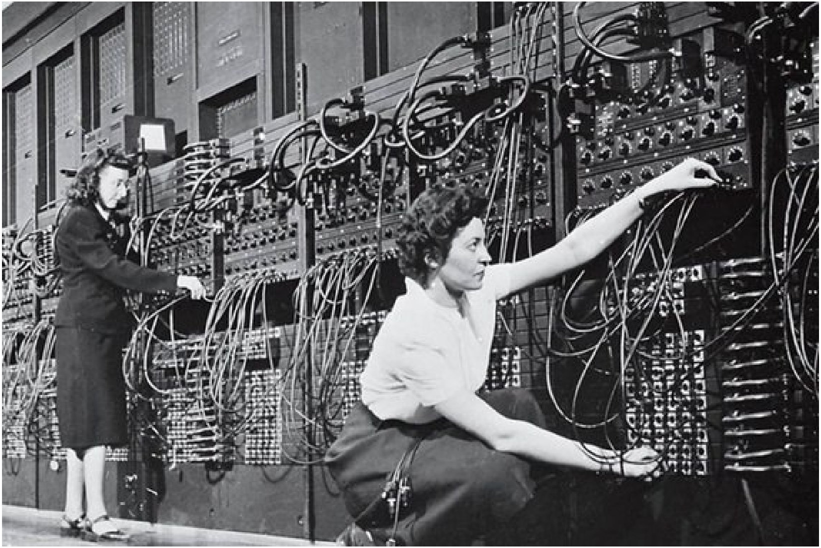
March 6, 2017



Kasjer (K)
Dyrektor (D)
Prezes (P)
0
1
0
1
0
1

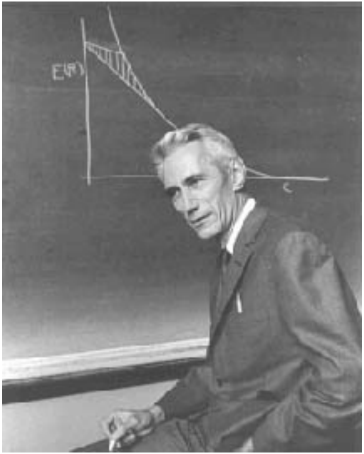

Kasjer (K)
Dyrektor (D)
Prezes (P)
0
1
0
1
0
1


Kasjer (K)
Dyrektor (D)
Prezes (P)
0
1
0
1
0
1



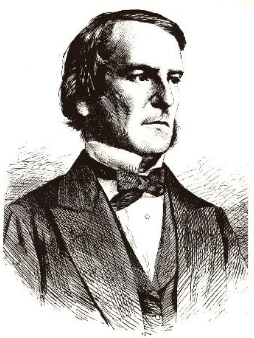
Claude Shannon

W 1938 r. Claude E. Shannon zadaptował algebrę Boole'a do opisu obwodów elektronicznych.



Claude Elwood Shannon (1916 - 2001) Amerykański matematyk i inżynier, profesor MIT.

Algebra Boole'a



George Boole (1815 – 1864)
Angielski matematyk, filozof i logik

W 1854 roku George Boole zaproponował dwuelementowy system algebraiczny z pogranicza filozofii i logiki matematycznej.

Algebra Boole'a

Podstawy algebry Boole'a

0, fałsz, off
1, prawda, on

Negacja $\sim$

$\overline{1} = 0$
$\overline{0} = 1$

Suma $\vee$

$0 + 0 = 0$
$1 + 0 = 1$
$0 + 1 = 1$
$1 + 1 = 1$

Iloczyn $\wedge$

$0 \cdot 0 = 0$
$1 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 1 = 1$

Aksjomaty

Twierdzenia

Aksjomaty algebry Boole'a

A1

$x = 0 \iff x \neq 1$
$x = 1 \iff x \neq 0$

A2

if $\mathrm{x} = 0 \implies \bar{x} = 1$
if $\mathrm{x} = 1 \implies \bar{x} = 0$

A3

$0 \cdot 0 = 0$
$1 + 1 = 1$

A4

$1 \cdot 1 = 1$
$0 + 0 = 0$

A5

$0 \cdot 1 = 1 \cdot 0 = 0$
$1 + 0 = 0 + 1 = 1$

Aksjomaty definiują algebrę przełączania

Twierdzenia algebry Boole'a

T1

$x + 0 = x$

$x \cdot 1 = x$

T2

$x + 1 = 1$

$x \cdot 0 = 0$

T3

$x + x = x$

$x \cdot x = x$

T4

$\bar{\bar{x}} = x$

T5

$x + \bar{x} = 1$

$x \cdot \bar{x} = 0$

Twierdzenia algebry Boole'a

T6

$x + y = y + x$

$x \cdot 1 = y \cdot x$

T9

$x + x \cdot y = x$

$x \cdot (x + y) = x$

T7

$(x + y) + z = x + (y + x)$

$(x \cdot y) \cdot z = x \cdot (y \cdot z)$

T10

$x \cdot y + x \cdot \bar{y} = x$

$(x + y) \cdot (x + \bar{y}) = x$

T8

$x \cdot (y + z) = x \cdot y + x \cdot z$

$x + (y \cdot z) = x \cdot y + x \cdot z$

T11

$x \cdot y + \bar{x} \cdot z + y \cdot z = x \cdot y + \bar{x} \cdot z$

$(x + y) \cdot (\bar{x} + z) \cdot (x + y) = (x + y) \cdot (\bar{x} + z)$

Dowód twierdzenia 8

$$x+(y*z) = (x + y) * (x + z)$$

Tabela prawdy

	x	y	z	x+(y*z)	(x+y)*(x+z)
1	0	0	0	0	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	1	1	1	1
5	1	0	0	1	1
6	1	0	1	1	1
7	1	1	0	1	1
8	1	1	1	1	1

Twierdzenia algebry Boole'a

T12

$x + x + x + \ldots + x = x$

$x \cdot x \cdot x \cdot \ldots \cdot x = x$

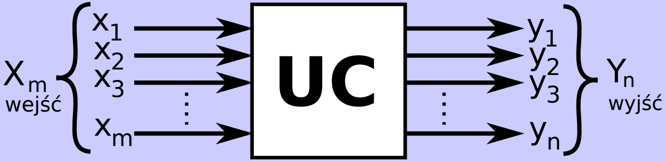
T13 (Prawo de Morgana)

$\overline{x_1 \cdot x_2 \cdot \ldots \cdot x_n} = \bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_n$

$\overline{x_1 + x_2 + \ldots + x_n} = \bar{x}_1 \cdot \bar{x}_2 \cdot \ldots \cdot \bar{x}_n$

Opis układów przełączających

Układem przełączającym (układem cyfrowym) nazywamy układ do przetwarzania sygnałów dwuwartościowych (binarnych).



Każde wejście i wyjście przyjmuje tylko jedną z dwóch wartości: 0 lub 1

Ciąg sygnałów wejściowych $X=(x_1, x_2,..., x_n)$ określa tzw. stan wejść, natomiast ciąg $Y=(y_1, y_2,..., y_n)$ stan wyjść automatu

Podział układów przełączających

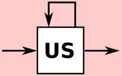
Układy kombinacyjne - stan wyjść układu w dowolnej chwili t zależy **wyłącznie** od obecnego stanu wejść układu, tzn. jest jednoznacznie określony przez aktualny stan wejść:

$Y_i = f(X_i)$ gdzie i=1, 2, 3... sekwencja stanów

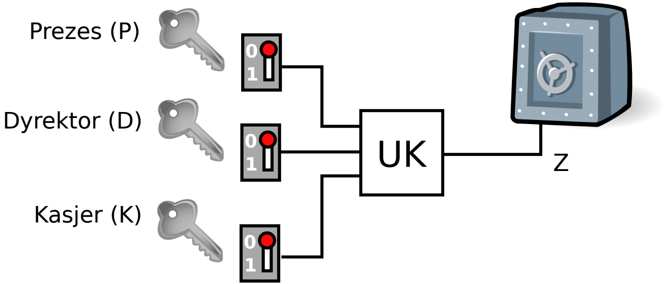
Układy sekwencyjne - stan wyjść zależy nie tylko od obecnego stanu wejść ale także od poprzednich, tzn. na stan wyjść wpływa również historia działania układu:

$Y_i = f(X_i, X_{i-1},\ldots)$

Układy sekwencyjne muszą dysponować pamięcią.



Układy złożone - połączenie układów kombinacyjnych i sekwencyjnych.


Prezes (P)
Dyrektor (D)
Kasjer (K)
0
1
0
1
0
1
UK
Z

Funkcja przełączająca

Funkcja przełączająca przyporządkowuje kolejnym kombinacjom wartości zmiennych wejściowych X odpowiednie wartości zmiennych wyjściowych Y. Najprostszym opisem funkcji jest tablica.

Przykłady:

a)

x_1	x_2	y_1
0	0	1
0	1	1
1	0	0
1	1	0

b)

Dwie zmienne wyjściowe

x_1	x_2	x_3	y_1	y_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

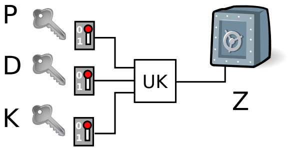
Intuicyjnie:

Wyjście układu a) będzie aktywne (jedynka)jeżeli zajdzie jedna z sytuacji:

x_1 będzie 0 **i** x_2 będzie 0

lub

x_1 będzie 0 **i** x_2 będzie 1



Tablica prawdy

	p	d	k	z		
1	0	0	0	0		
2	0	0	1	0		
3	0	1	0	0		
4	0	1	1	0		
5	1	0	0	1	$\longrightarrow p*\bar{d}*\bar{k}$	(1*1*1)
6	1	0	1	0		
7	1	1	0	1	$\longrightarrow p*d*\bar{k}$	(1*1*1)
8	1	1	1	1	$\longrightarrow p*d*k$	(1*1*1)

Postać kanoniczna
określająca jedynki funkcji

$$z = p*\bar{d}*\bar{k} + p*d*\bar{k} + p*d*k = p(\bar{d}*\bar{k} + d*\bar{k} + d*k) = p(\bar{k} + dk)$$

Po minimalizacji: $\mathbf{z = p\bar{k} + pdk}$

Metoda tablic Karnaugh

Tablica prawdy

	p	d	k	z
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	1

zmiana kolejności zmiennych p,d,k (Kod Graya)

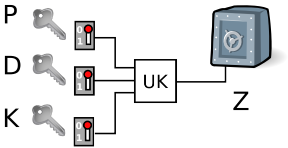
Tablica Karnaugh

p \ dk	00	01	11	10
0	0	0	0	0
1	1	0	1	1

$z = p*\bar{d}*\bar{k} + p*d*\bar{k} + p*d*k$
+ minimalizacja

Zakreślenie i szybka minimalizacja

Metoda tablic Karnaugh



p \ dk	00	01	11	10
0	0	0	0	0
1	1	0	1	1

$$T5: \; x + \bar{x} = 1$$

$$pdk + pd\bar{k} = pd(k+\bar{k})$$

$$p\overline{dk} + pd\bar{k} = p\bar{k}(\bar{d}+d)$$

Po minimalizacji: $z = pd + p\bar{k}$

Opis metody:
Dążąc do uzyskania minimalnej liczby obszarów należy zakreślać prostokąty 2^n elementowe i wypisywać tylko zmienne, których wartość nie zmienia się.

Metoda tablic Karnaugh

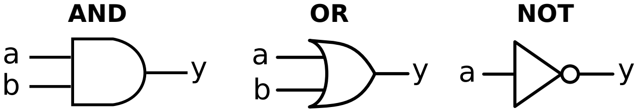
Przykład: Tablica Karnaugh dla czterech zmiennych

$a_1 a_2 \backslash a_3 a_4$	00	01	11	10
00	1	1	1	0
01	1	1	1	0
11	0	0	1	0
10	0	0	1	0

Opis metody:
Dążąc do uzyskania minimalnej liczby obszarów należy zakreślać prostokąty 2^n elementowe i wypisywać tylko zmienne, których wartość nie zmienia się.

Bramki logiczne

$$z = pd + p\bar{k}$$

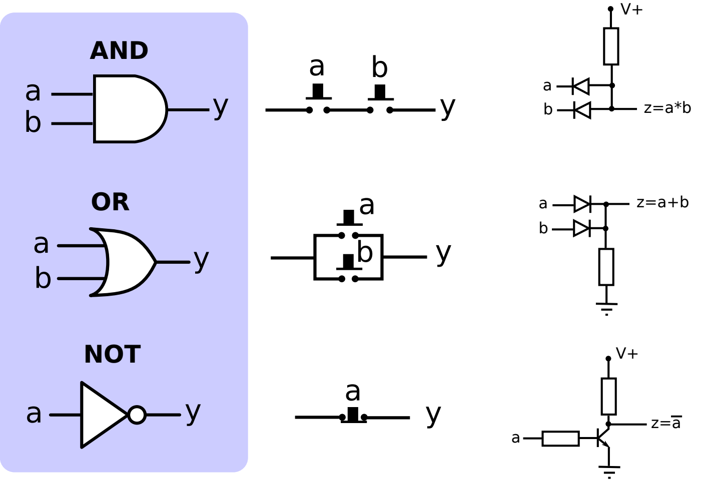


a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

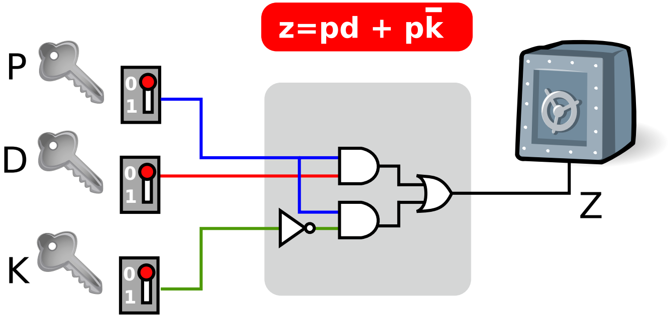
a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

a	y
0	1
1	0

Bramki logiczne

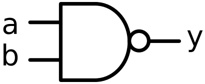


Realizacja układu



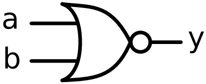
Inne bramki logiczne

NAND



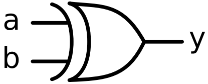
a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

NOR



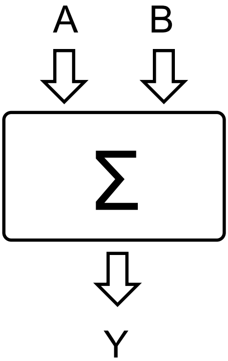
a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

EXOR



a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Sumator dwóch liczb w kodzie NKB

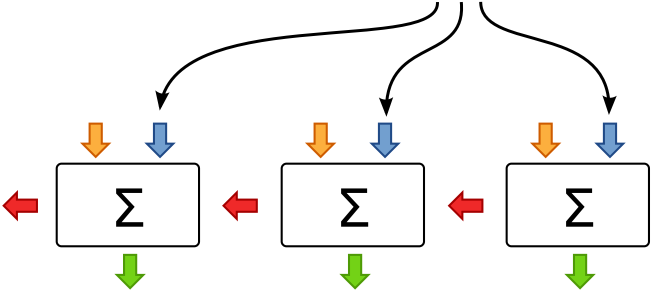


Sumator dwóch liczb w kodzie NKB

Przykład: Obliczyć 7+2

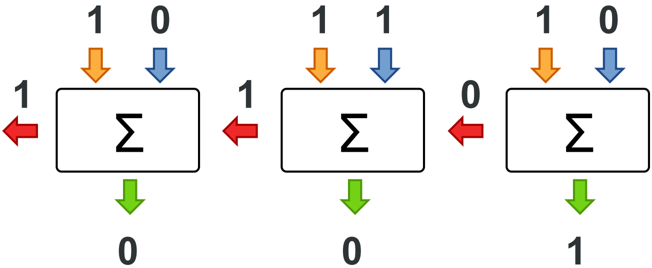
7 + 2

111 + 010



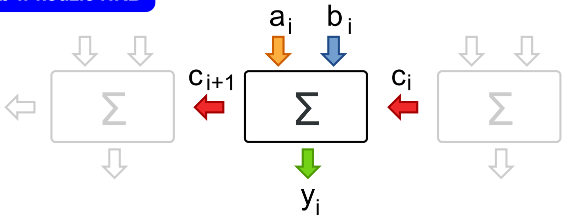
Sumator dwóch liczb w kodzie NKB

Przykład: Obliczyć 7+2



1001=9

Sumator dwóch liczb w kodzie NKB



Tablica prawdy dla c_{i+1}

	a_i	b_i	c_i	c_{i+1}
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

Tablica prawdy dla y_i

	a_i	b_i	c_i	y_i
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

Tablice Karnaugh dla sumatora

Tablica prawdy dla y_i

	a_i	b_i	c_i	y_i
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

$c_i \backslash a_i b_i$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

(bramka mod2)

$$y_i = \bar{a}_i b_i \bar{c}_i + a_i \bar{b}_i \bar{c}_i + \bar{a}_i \bar{b}_i c_i + a_i b_i c_i$$

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

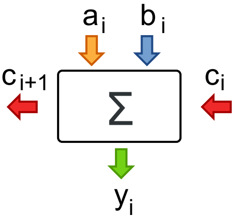
Tablica prawdy dla c_{i+1}

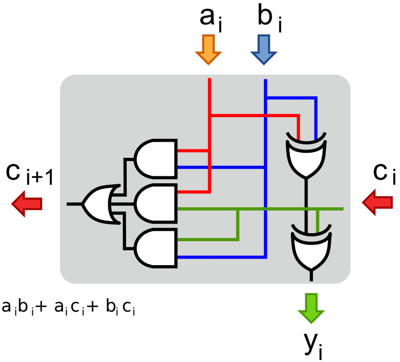
	a_i	b_i	c_i	c_{i+1}
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

$c_i \backslash a_i b_i$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$c_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

Realizacja sumatora na bramkach



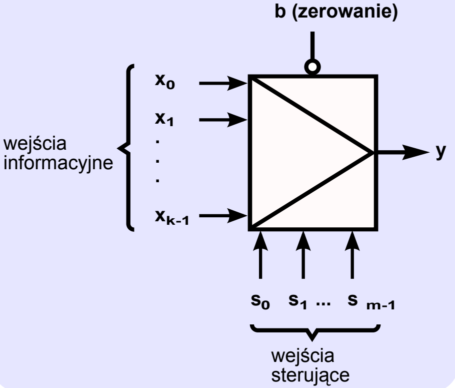


$$c_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

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

Układy kombinacyjne: Multiplekser

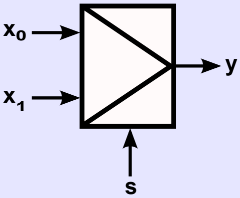
Schemat układu



Poprzez sygnał sterujący wybierane jest wejście informacyjne, z którego sygnał zostanie przeniesiony na wyjście

Układy kombinacyjne: Multiplekser dwuwejściowy

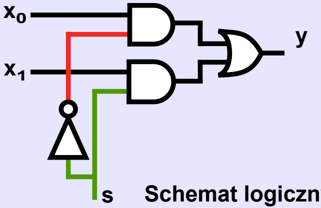
Układ



s	y
0	x_0
1	x_1

Funkcja opisująca układ

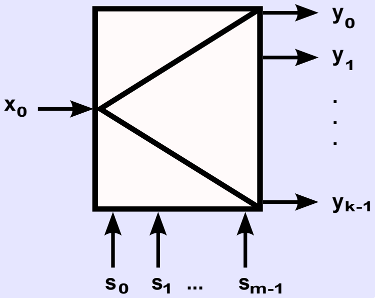
$$y = x_0\bar{s} + x_1 s$$



Schemat logiczny

Układy kombinacyjne: Demultiplekser

Schemat układu

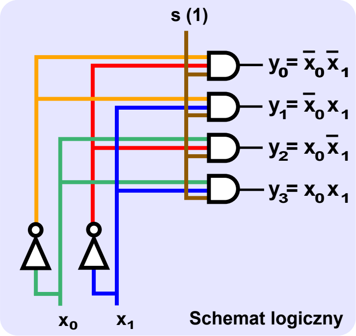


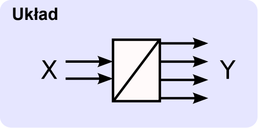
Demultiplekser jest układem o działaniu przeciwnym do multipleksera.

Sygnał wejściowy podany na wejście x_0 pojawia się na wyjściu y_i wybranym przez wektor adresowy S_i

Układy kombinacyjne: Dekoder

Dekoder jest układem służącym do konwersji kodów





Kod 1 z n

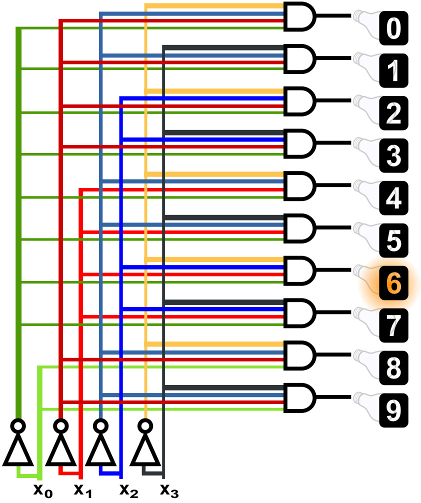
x_0	x_1	Y
		$y_3 y_2 y_1 y_0$
0	0	0001
0	1	0010
1	0	0100
1	1	1000

Przykład: Dekoder BCD (Binary Coded Decimal)

Problem:

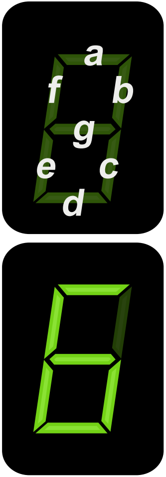
Jaką wartość dziesiętną ma ciąg BCD 0110?

	x_0	x_1	x_2	x_3	Y
0	0	0	0	0	0000000001
1	0	0	0	1	0000000010
2	0	0	1	0	0000000100
3	0	0	1	1	0000001000
4	0	1	0	0	0000010000
5	0	1	0	1	0000100000
6	0	1	1	0	0001000000
7	0	1	1	1	0010000000
8	1	0	0	0	0100000000
9	1	0	0	1	1000000000



Przykład: Dekoder BCD dla wyświetlacza siedmiosegmentowego

	X_0	X_1	X_2	X_3	Y abcdefg
0	0	0	0	0	1111110
1	0	0	0	1	0110000
2	0	0	1	0	1101101
3	0	0	1	1	1111001
4	0	1	0	0	0010011
5	0	1	0	1	1011011
6	0	1	1	0	1011111
7	0	1	1	1	1110000
8	1	0	0	0	1111111
9	1	0	0	1	1111011



Przykład: Dekoder BCD dla wyświetlacza siedmiosegmentowego

	X_0	X_1	X_2	X_3	Y abcdefg
0	0	0	0	0	1111110
1	0	0	0	1	0110000
2	0	0	1	0	1101101
3	0	0	1	1	1111001
4	0	1	0	0	0010011
5	0	1	0	1	1011011
6	0	1	1	0	1011111
7	0	1	1	1	1110000
8	1	0	0	0	1111111
9	1	0	0	1	1111011

a

CD \ AB	00	01	11	10
00	1	0	-	1
01	0	1	-	1
11	1	1	-	-
10	1	1	-	-

$a=A+C+\overline{BD}+BD$

b

CD \ AB	00	01	11	10
00	1	0	-	1
01	1	0	-	1
11	1	1	-	-
10	1	0	-	-

$b=\overline{B}+CD$

c

CD \ AB	00	01	11	10
00	1	1	-	1
01	1	1	-	1
11	1	1	-	-
10	0	1	-	-

$c=\overline{C}+B+D$

d

CD \ AB	00	01	11	10
00	1	0	-	1
01	0	1	-	1
11	1	0	-	-
10	1	1	-	-

$d=A+C\overline{D}+B\overline{C}D+\overline{B}\overline{D}+\overline{B}C$

e

CD \ AB	00	01	11	10
00	1	0	-	1
01	0	0	-	0
11	0	0	-	-
10	1	1	-	-

$e=C\overline{D}+\overline{B}\overline{D}$

f

CD \ AB	00	01	11	10
00	1	1	-	1
01	0	1	-	1
11	0	0	-	-
10	0	1	-	-

$f=A+B\overline{C}+\overline{CD}+B\overline{D}$

g

CD \ AB	00	01	11	10
00	0	1	-	1
01	0	1	-	1
11	1	0	-	-
10	1	1	-	-

$g=A+B\overline{D}+B\overline{C}+\overline{B}C$

Dążąc do uzyskania minimalnej liczby obszarów należy zakreślać prostokąty 2^n elementowe i wypisywać tylko zmienne, których wartość nie zmienia się.

Przykład: Dekoder BCD dla wyświetlacza siedmiosegmentowego

	A	B	C	D	Y abcdefg
0	0	0	0	0	1111110
1	0	0	0	1	0110000
2	0	0	1	0	1101101
3	0	0	1	1	1111001
4	0	1	0	0	0010011
5	0	1	0	1	1011011
6	0	1	1	0	1011111
7	0	1	1	1	1110000
8	1	0	0	0	1111111
9	1	0	0	1	1111011

$a=A+C+\overline{B}\overline{D}+BD$

$b=\overline{B}+CD$

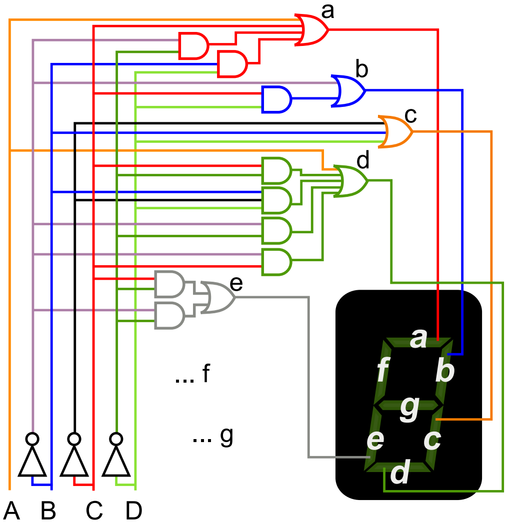
$c=\overline{C}+B+D$

$d=A+C\overline{D}+B\overline{C}D+\overline{B}\overline{D}+\overline{B}C$

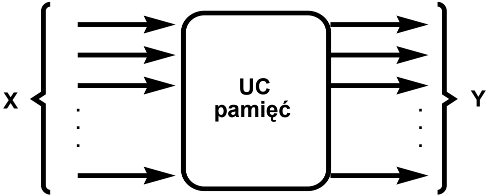
$e=C\overline{D}+\overline{B}\overline{D}$

$f=A+B\overline{C}+\overline{C}\overline{D}+B\overline{D}$

$g=A+B\overline{D}+B\overline{C}+\overline{B}C$



Układy sekwencyjne



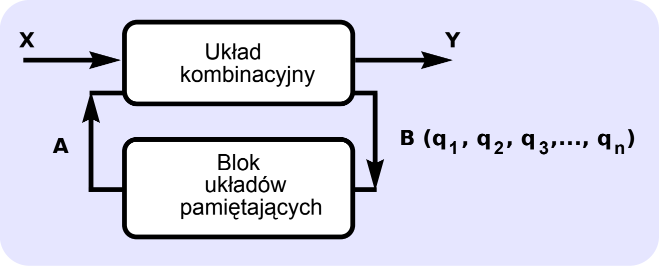
Pamięć jest podstawową cechą układu sekwencyjnego.

Obok stanów wejść X i stanów wyjść Y istnieje zbiór stanów pamięci A.

$A=(Q_1, Q_2, Q_3, \dots, Q_n)$ gdzie Q_i są elementarnymi stanami pamięci.

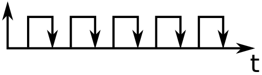
Układy sekwencyjne

Ogólna analiza automatów sekwencyjnych prowadzi do schematu blokowego, w którym wyodrębnia się blok pamiętający.



A - stan pamięci
B = $(q_1, q_2, q_3, \ldots, q_n)$ stan wzbudzeń elementów pamięci

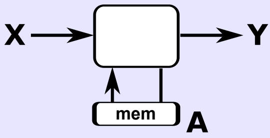
Zegar



Układy sekwencyjne

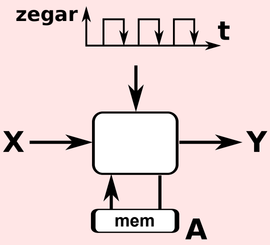
Układy asynchroniczne

Zmiany A oraz Y pojawiają się po zmianie X



Układy synchroniczne

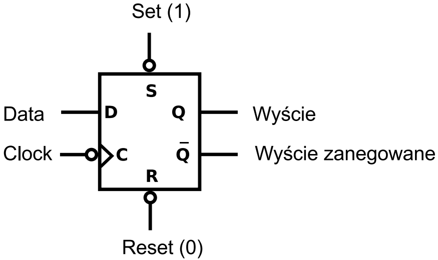
Zmiany A oraz Y pojawiają się po taktach zegara

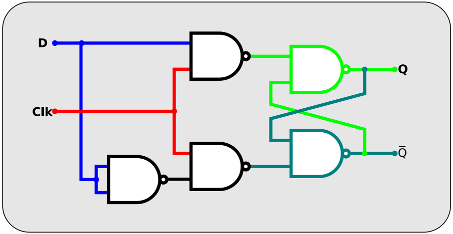


Synchroniczne układy sekwencyjne: Przerzutnik typu D (Delay)

Podstawowym elementem bloku pamięciowego jest przerzutnik

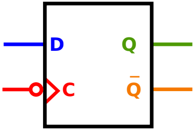
Przerzutnik pozwala na zapamiętanie 1 bitu

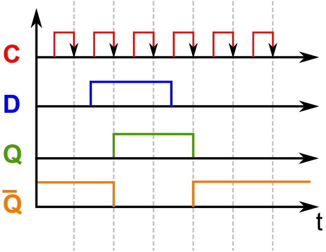




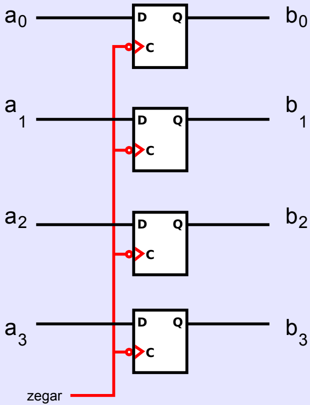
Synchroniczne układy sekwencyjne: Przerzutnik typu D (Delay)

Impuls zegara zmienia stan przerzutnika ale z przesunięciem czasowym





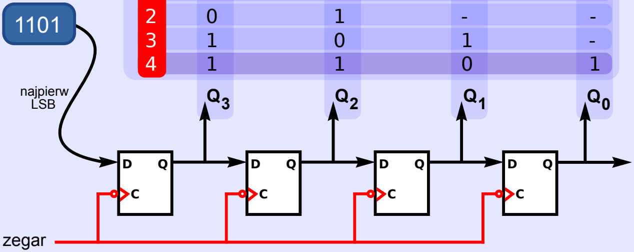
Synchroniczne układy sekwencyjne: Rejestr równoległy



Synchroniczne układy sekwencyjne: Rejestr przesuwny

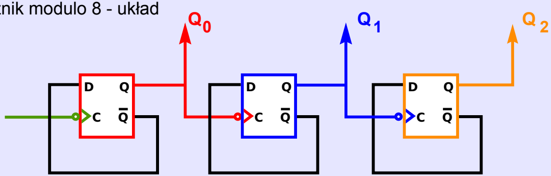
We Szeregowe, Wy Równoległe

takt zegara	Q_3	Q_2	Q_1	Q_0
1	1	-	-	-
2	0	1	-	-
3	1	0	1	-
4	1	1	0	1



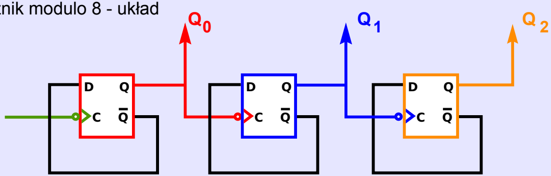
Synchroniczne układy sekwencyjne: Licznik binarny

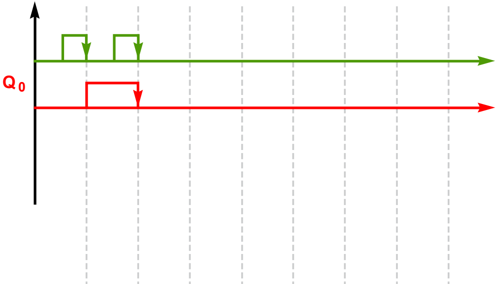
Licznik modulo 8 - układ



Synchroniczne układy sekwencyjne: Licznik binarny

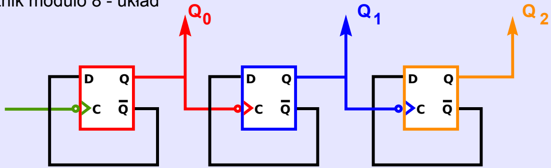
Licznik modulo 8 - układ

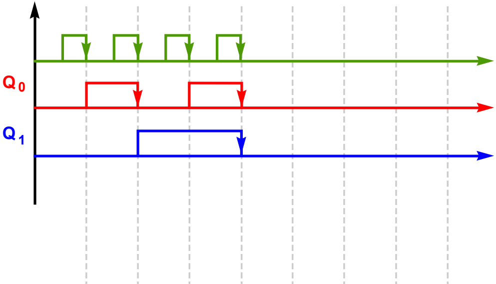




Synchroniczne układy sekwencyjne: Licznik binarny

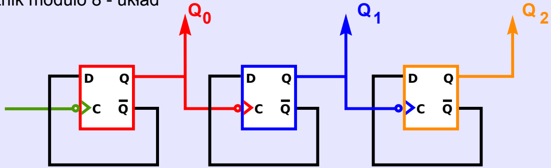
Licznik modulo 8 - układ

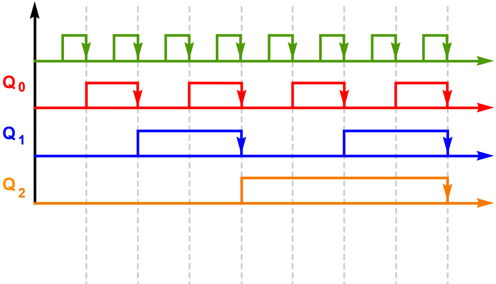




Synchroniczne układy sekwencyjne: Licznik binarny

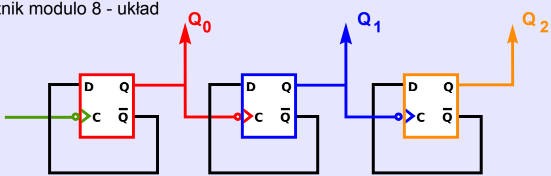
Licznik modulo 8 - układ

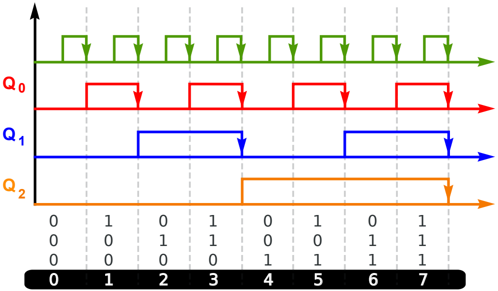




Synchroniczne układy sekwencyjne: Licznik binarny

Licznik modulo 8 - układ





Intel 4004 Architecture