Tranzystor jest elementem półprzewodnikowym

- działanie - budowa złącza **p-n**,
- fizyka ciała stałego.

Służy do wzmacniania lub przełączania sygnałów

Elementy 3-końcówkowe, w których:

- przewodność między dwoma końcówkami zależy od liczby nośników ładunków znajdujących się między nimi,
- liczba nośników ładunków zależy od wartości napięcia doprowadzonego do elektrody sterującej zwanej bazą w tranzystorach bipolarnych lub bramką w tranzystorach polowych.

Podział

- **tranzystor bipolarny**
- **tranzystor polowy (unipolarny)**
  JFET oraz MOSFET,

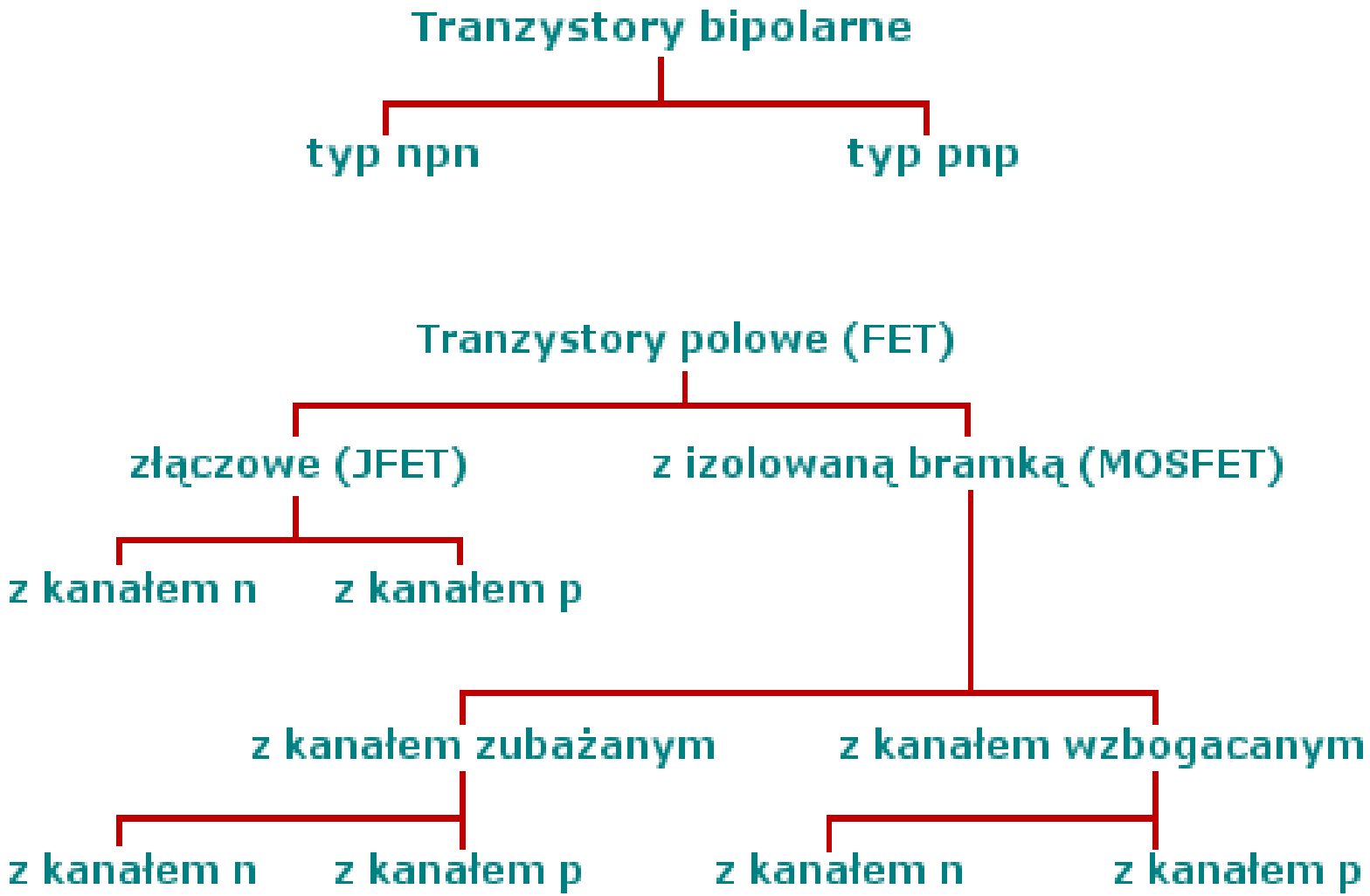
Tranzystory bipolarne można podzielić na:

- tranzystory npn
- tranzystory pnp,

Tranzystory polowe na:

- złączowe
- z izolowaną bramką.

Dalszy podział jest przedstawiony niżej.


Tranzystory bipolarne
typ npn
typ pnp
Tranzystory polowe (FET)
złączowe (JFET)
z izolowaną bramką (MOSFET)
z kanałem n
z kanałem p
z kanałem zubażanym
z kanałem wzbogacanym
z kanałem n
z kanałem p
z kanałem n
z kanałem p

Tranzystory bipolarne

Tranzystory bipolarne dzieli się na:

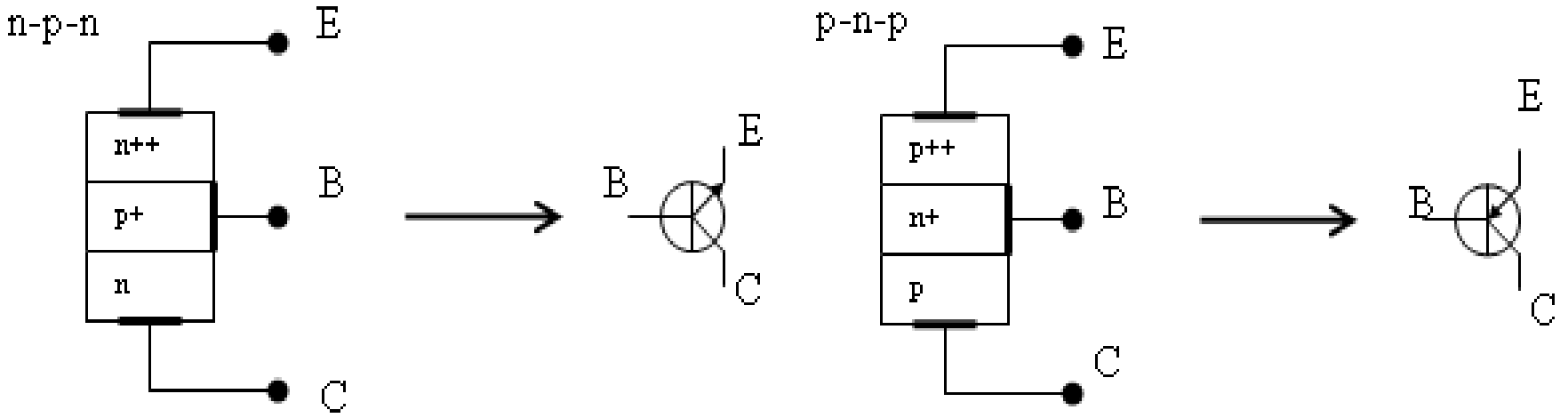
- krzemowe
- germanowe,

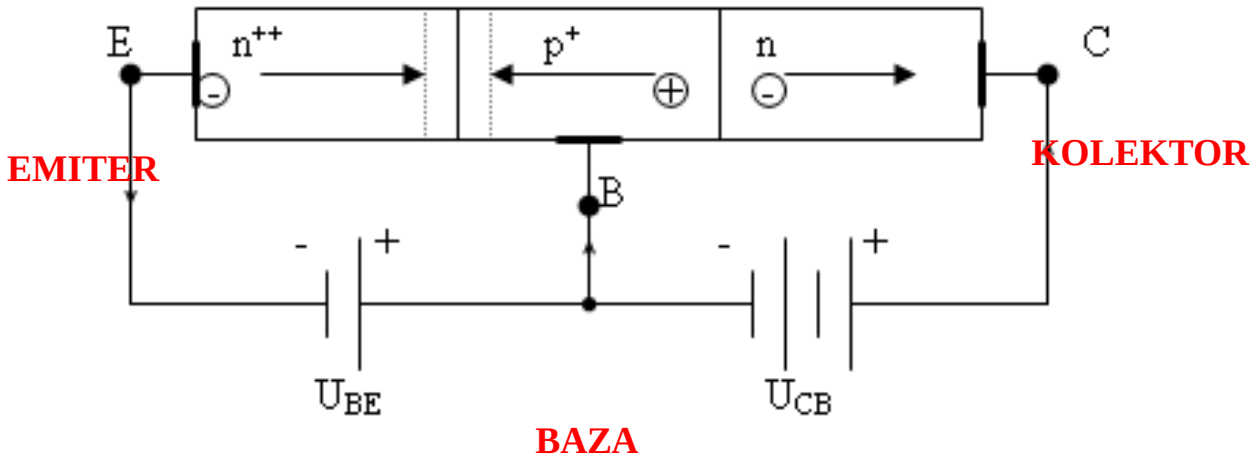
a każdy z nich może być typu

- npn
- lub pnp.

Tranzystor bipolarny...

Tranzystor bipolarny posiada dwa złącza p-n, wytworzone w jednej płytce półprzewodnikowej.
Ze względu na kolejność ułożenia warstw półprzewodnika rozróżniamy dwa typy tego tranzystora





Nośnikami ładunku elektrycznego są elektrony i dziury. Zasada działania tranzystora jako wzmacniacza:

Należy go odpowiednio spolaryzować

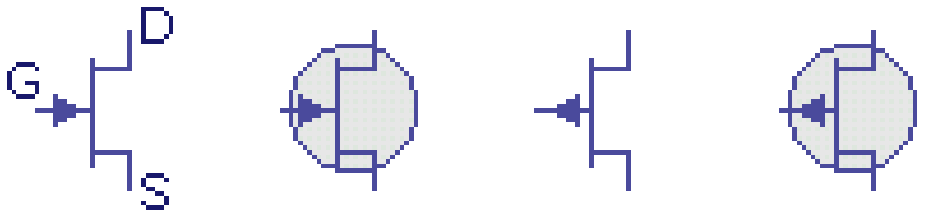
- złącze emiterowe w kierunku przewodzenia,
- złącze kolektorowe w kierunku wstecznym

Tranzystory polowe (unipolarne)

są również elementami półprzewodnikowymi lecz różnią się od bipolarnych tym, że są:

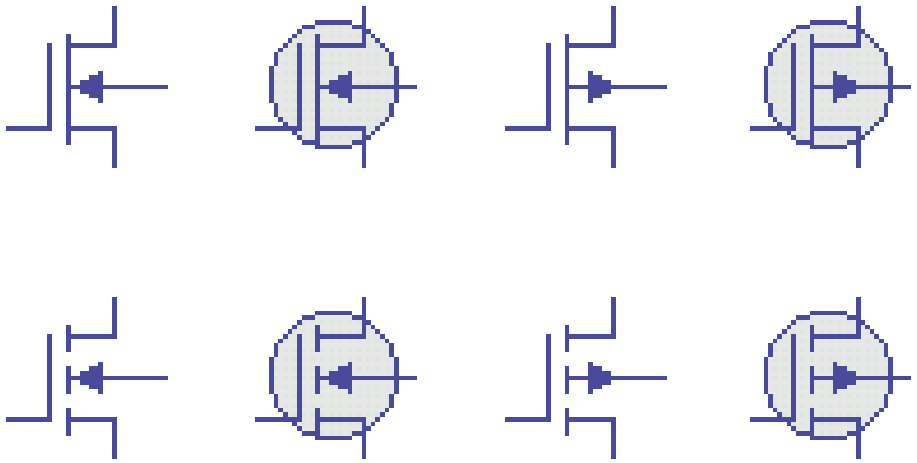
- sterowane polem elektrycznym
- nie pobierają mocy na wejściu.

JFET



złączowe

MOSFET



z izolowaną bramką

Nazwy poszczególnych elektrod to: D - dren, S - źródło, G - bramka.

Elektrody te spełniają podobne funkcje jak odpowiadające im elektrody w tranzystorze bipolarnym.

- kolektorowi C odpowiada dren D,
- emiterowi E odpowiada źródło S,
- bazie B odpowiada bramka G.

Działanie tranzystora polowego polega na sterowaniu przepływem prądu przez kanał za pomocą pola elektrycznego wytwarzanego przez napięcie doprowadzone do bramki.

Ponieważ w tranzystorze polowym nie ma żadnych przewodzących złączy więc do bramki nie wpływa ani z niej nie wypływa żaden prąd i jest to najważniejsza cecha tranzystorów polowych.

Z właściwości tej wynika duża wartość rezystancji wejściowej tranzystora polowego - w zastosowaniach takich jak przełączniki analogowe bardzo istotne.

Układy TTL

transistor – transistor logic

tranzystory bipolarne

Układy CMOS

polowe MOSFET

pary komplementarne:

- jeden tranzystor MOSn,
- jeden MOSp

Układy kombinacyjne i sekwencyjne

Istota techniki cyfrowej:

Wytwarzanie cyfrowych sygnałów wyjściowych jako odpowiedzi na cyfrowe sygnały wejściowe

Na przykład:

Sumator przetwarza doprowadzone do wejść dwie liczby 16-bitowe na 16-bitową sumę tych liczb oraz bit przeniesienia.

Wykonanie - jednostka arytmetyczna komputera.

Inne zadania:

- **porównanie dwóch liczb w celu sprawdzenia, która z nich jest większa**
- **porównanie zestawu sygnałów wejściowych z sygnałem pożądanym w celu sprawdzenia, czy podzespoły weszły w tryb normalnej pracy,**
- **dołączenia do liczby "bitu parzystości" tak, aby całkowita liczba jedynek w reprezentacji liczby była parzysta, na przykład przed transmisją przez łącze; następnie parzystość może być sprawdzona przy odbiorze, co daje prostą kontrolę poprawności transmisji,**
- **pobranie pewnych liczb binarnych i ich wyświetlenie,**

Sygnały (stany) wyjściowe są zdeterminowanymi funkcjami sygnałów (stanów) wejściowych

Zalicza się je do zadań "kombinacyjnych".

Wszystkie mogą być wykonane za pomocą urządzeń zwanych bramkami, które realizują działania algebry Boole`a w dziedzinie układów dwustanowych (binarnych).

Istnieje druga klasa zagadnień, które nie mogą być rozwiązane przez utworzenie kombinacyjnych funkcji tylko bieżących stanów wejść, lecz wymagają znajomości poprzednich stanów wejść.

Ich rozwiązanie wymaga zastosowania układów "sekwencyjnych".

Typowe zadania sekwencyjne to:

- zamiana szeregowego ciągu bitów (bity następują kolejno jeden po drugim) w równoległy zestaw bitów,
- zliczanie jedynek w danej sekwencji,
- rozpoznanie pewnego wzoru w sekwencji,
- wytworzenie jednego impulsu dla co czwartego impulsu wejściowego.

Do realizacji wszystkich wymienionych zadań konieczne jest zastosowanie jakiejś pamięci cyfrowej.

Podstawowym urządzeniem pamięciowym jest przerzutnik bistabilny (ang. *flip flop* lub *bistable multivibrator*)

Zaczniemy od bramek i układów kombinacyjnych ...

Bramka OR – suma logiczna

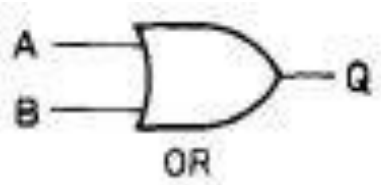
Wyjście bramki OR (czyli LUB) jest w stanie wysokim, jeżeli któreś z wejść (lub obydwa) jest w stanie wysokim

Narysowana bramka to 2-wejściowa bramka OR.

W przypadku ogólnym bramki mogą mieć dowolną liczbę wejść

Typowy układ scalony

- **cztery bramki 2-wejściowe,**
- **trzy bramki 3-wejściowe**
- **lub dwie bramki 4-wejściowe**



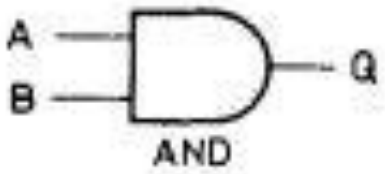
Tablica prawdy

Wejścia A	Wejścia B	Wyjście Q
0	0	0
0	1	1
1	0	1
1	1	1

Bramka AND – iloczyn logiczny

Wyjście bramki AND (czyli I) jest w stanie wysokim tylko wtedy, gdy obydwa wejścia są w stanie wysokim.

Na przykład 8-wejściowa bramka AND będzie miała wyjście w stanie wysokim tylko wtedy, gdy wszystkie wejścia będą w stanie wysokim.



Tablica prawdy

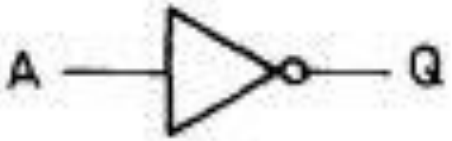
Wejścia		Wyjście
0	0	0
0	1	0
1	0	0
1	1	1

Inwerter (funkcja NOT)

Zmiana stanu logicznego na przeciwny (negowaniem stanu logicznego).

"bramka" o jednym wejściu

Zapis – A’ lub $\overline{A}$

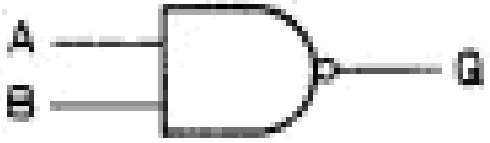


Inwerter

A	Q
0	1
1	0

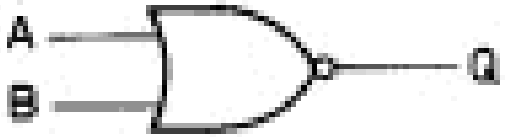
NAND i NOR

Funkcja NOT może być połączona z innymi funkcjami, tworząc NAND i NOR



NAND

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

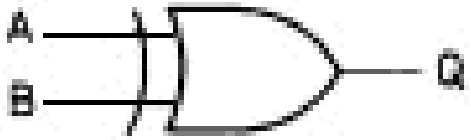


NOR

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR Exclusive-OR (XOR, czyli WYŁĄCZNE LUB)

Wyjście bramki XOR jest w stanie wysokim, jeżeli jedno albo drugie wejście jest w stanie wysokim (jest to zawsze funkcja dwóch zmiennych). Mówiąc inaczej, wyjście jest w stanie wysokim, jeżeli stany wejść są różne



XOR

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

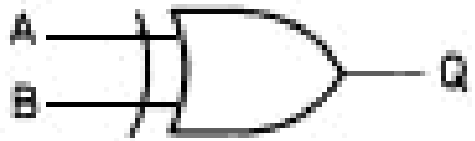
Kod Graya

Liczba	Liczba binarna	Kod Graya
0	0000	0000
1	0001	0001
2	0010	0011
3	0100	0010
4	0101	0110
5	0110	0111

itd.

kolejne liczby różni tylko jedna pozycja (jeden bit)

Zastosowanie XOR: równoległe konwertery kodów

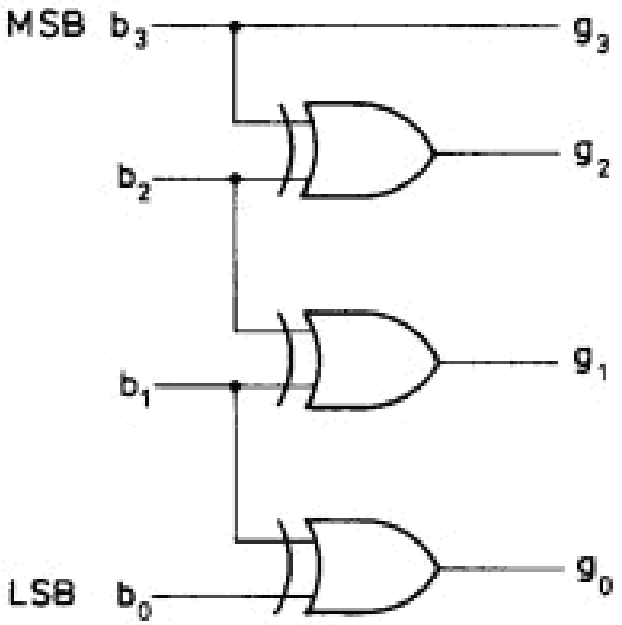


XOR

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

z binarnego na Graya

we



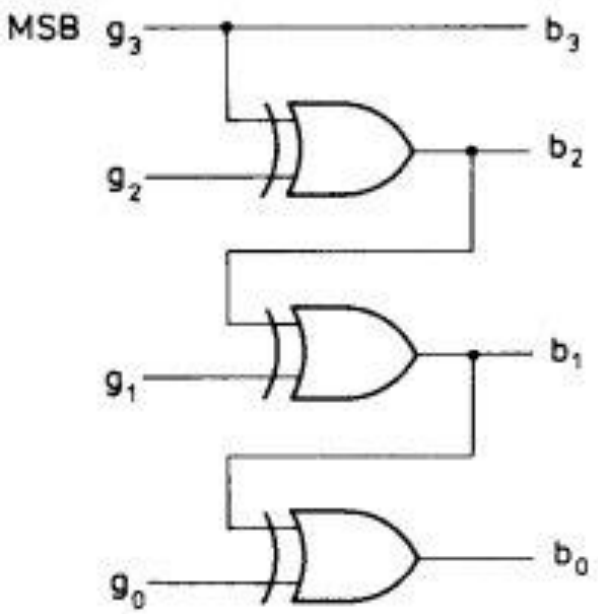
wy

np.

we	wy
0	0
0	0
1	1
0	1

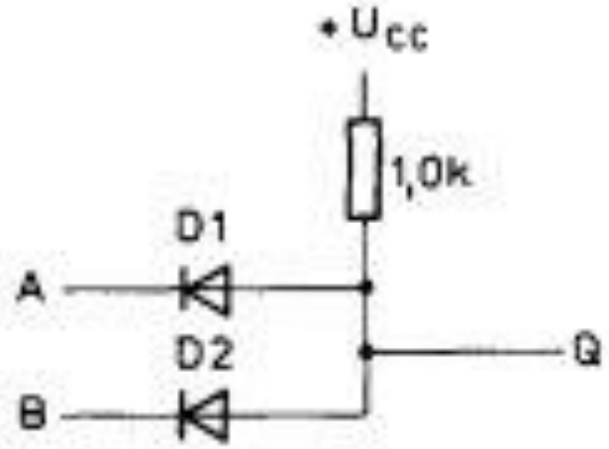
3 bramki XOR

z kodu Graya na binarny



Bramki z elementów dyskretnych

bramka AND wykonana z użyciem diod



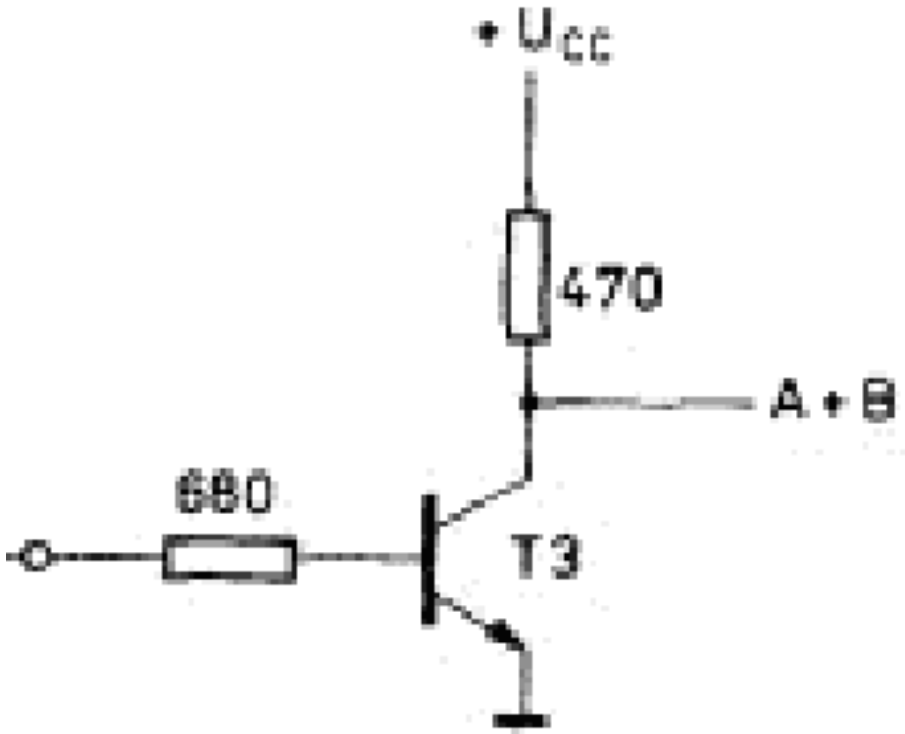
Wady:

- **poziom niski na wyjściu jest o spadek napięcia na diodzie wyższy od poziomu niskiego na wejściu. Nie pozwala to na łączenie szeregowe wielu takich układów!**
- **nie ma żadnego "wzmocnienia logicznego" (zdolności wyjścia do sterowania wielu wejść równocześnie)**
- **układ charakteryzuje się małą szybkością działania**

Zasada:

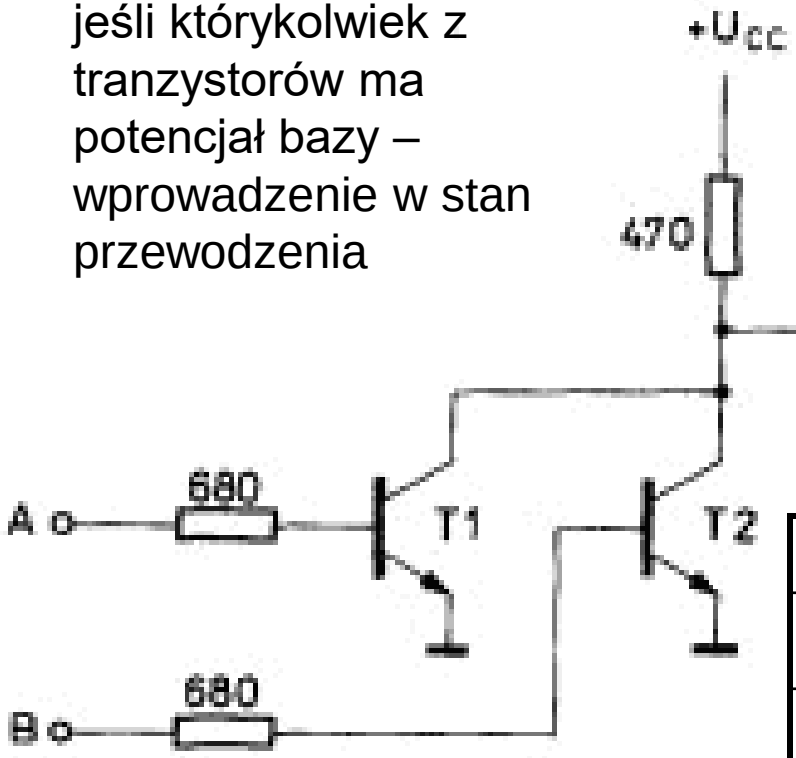
Trudno jest z elementów dyskretnych skonstruować układ logiczny działający równie dobrze, jak zbudowany z układów scalonych

BRAMKA NOT



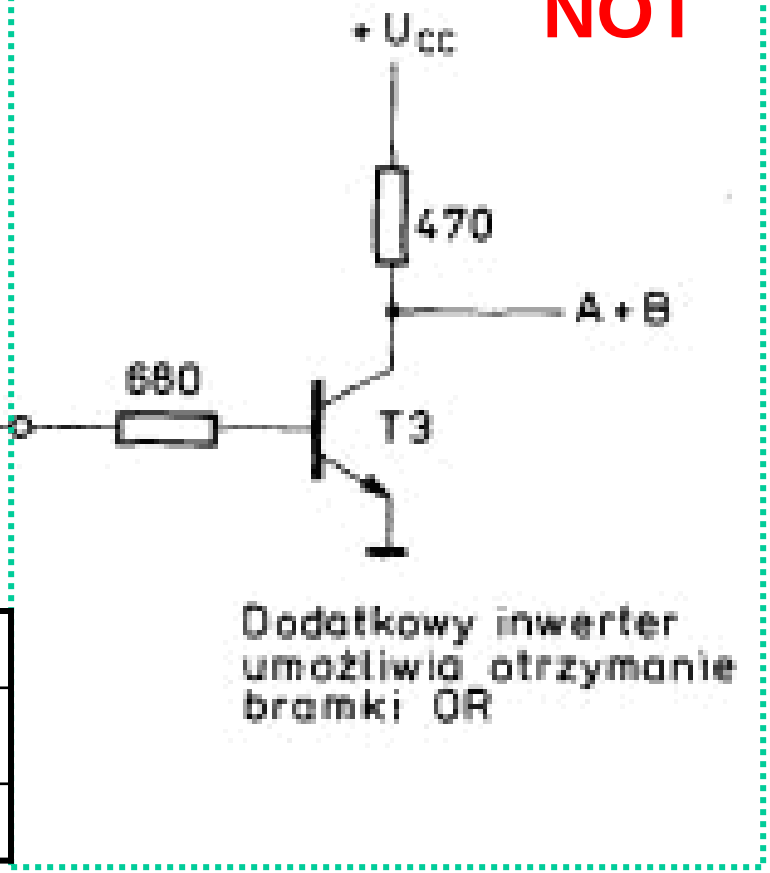
Najprostsza postać tranzystorowej bramki NOR

zwieranie do masy jeśli którykolwiek z tranzystorów ma potencjał bazy – wprowadzenie w stan przewodzenia



	0	1
0	1	0
1	0	0

NOT



OR

Odrobina arytmetyki komputerów

Kodowanie i działania na liczbach binarnych

Zapis pozycyjny

Układ dziesiętny

$$L_{10}=a_n a_{n-1}\ldots a_1,a_0,a_{-1}a_{-2}\ldots a_{-k}=\sum_{i=-k}^{i=n} a_i 10^i$$

Układ dowolny

$$L_p=a_n a_{n-1}\ldots a_1,a_0,a_{-1}a_{-2}\ldots a_{-k}=\sum_{i=-k}^{i=n} a_i p^i$$

NIEWYGODNY OBLICZENIOWO

ALGORYM ZAMIANY

Zamiana liczby dziesiętnej na binarną - 317,45

317 : 2 = 158 + r. 1
158 : 2 = 79 + r. 0
79 : 2 = 39 + r. 1
39 : 2 = 19 + r. 1
19 : 2 = 9 + r. 1
9 : 2 = 4 + r. 1
4 : 2 = 2 + r. 0
2 : 2 = 1 + r. 0
1 : 2 = 0 + r. 1

najmłodszy bit

100111101

0,45 · 2 = 0 | 0,9
0,9 · 2 = 1 | 0,8
0,8 · 2 = 1 | 0,6
0,6 · 2 = 1 | 0,2
0,2 · 2 = 0 | 0,4
0,4 · 2 = 0 | 0,8
0,8 · 2 = 1 | 0,6
0,6 · 2 = 1 | 0,2
itd. .....

01110011

Wynik 100111101,01110011

KODOWANIE

Kod – zestaw symboli przypisany danej informacji

Kodowanie – przypisywanie informacjom pewnych umownych symboli

KOD BCD

BCD (ang. *Binary Coded Decimal* czyli *liczby dziesiętne zakodowane binarnie*) - sposób zapisu liczb w komputerze. Polega na zapisaniu jednej cyfry dziesiętnej przy użyciu dokładnie czterech bitów.

Cyfry:

0	0000	**5**	0101
1	0001	**6**	0110
2	0010	**7**	0111
3	0011	**8**	1000
4	0100	**9**	1001

Przykład kodowania BCD (*BCD8421*):

$317_{10} = 100111101_2 = 0011 \quad 0001 \quad 0111$

3 1 7

Większość komputerów zapisuje liczby na ośmiu bitach (lub więcej)

Dwie możliwości zapisu BCD:

- wyzerowanie lub ustawienie najstarszych czterech bitów i zapisywanie cyfr na czterech najmłodszych bitach.
- zapis dwóch cyfr w każdym bajcie - tak zwane spakowane BCD.

INNE KODY

- **kod AIKENA**
- **kod JOHNSONA**
- **kod ze stałym indeksem**
- **kod Hamminga**
- **kod Graya**

Operacje arytmetyczne

Dodawanie

```
    1 0 1 0 0 1 0 1
+   0 1 1 1 1 0 0 1
-----------------------
  1 0 0 0 1 1 1 1 0
    1 1 1 0 0 0 0 1 przeniesienia
```

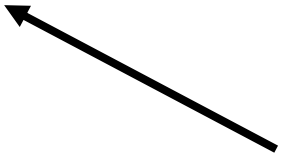
Zapis liczb binarnych:

Metoda ZM – znak-moduł

0, 1 0 1 0 0 1 0 1 **+165**

znak +

1, 1 0 1 0 0 1 0 1 **– 165**



znak –

Dodawanie łatwe – gorzej z odejmowaniem

Wprowadzono ułatwienie – **kod uzupełnieniowy do 2**

Zasada

- bit znaku zostawiamy bez zmian
- zamieniamy w liczbie 1 na 0, a 0 na 1
- do liczby dodajemy 1

Algorytmy dodawania i odejmowania są wówczas identyczne

Odejmowanie – przykład 165 - 121

		1	0	1	0	0	1	0	1	165
-		0	1	1	1	1	0	0	1	121
	0	0	0	1	0	1	1	0	0	44
	0	1	1	1	1	0	0	0		pożyczka

Odejmowanie w kodzie uzupełnieniowym do 2

ZM	0,	1	0	1	0	0	1	0	1	1 składnik dodatni
ZM	1,	0	1	1	1	1	0	0	1	2 składnik ujemny

	1,	1	0	0	0	0	1	1	0	2 skł. uzupełniamy do 2
									1	
	1,	1	0	0	0	0	1	1	1	2 skł. U2
	0,	1	0	1	0	0	1	0	1	pierwszy składnik
	0,0	0	1	0	1	1	0	0		dodajemy

Odejmowanie – przykład 121 - 165

		0	1	1	1	1	0	0	1	121
-		1	0	1	0	0	1	0	1	165
	?	1	1	0	1	0	1	0	0	212 (–256) = - 44 !!!
	1	0	0	0	0	1	0	0		pożyczka

ZM	0,	0	1	1	1	1	0	0	1	1 składnik dodatni
ZM	1,	1	0	1	0	0	1	0	1	2 składnik ujemny
	1,	0	1	0	1	1	0	1	0	2 skł. uzupełniamy do 2
									1	
	1,	0	1	0	1	1	0	1	1	2 skł. U2
	0,	0	1	1	1	1	0	0	1	pierwszy składnik
	1,	1	1	0	1	0	1	0	0	dodajemy
		0	1	1	1	1	0	1		przeniesienia
	1,	0	0	1	0	1	0	1	1	jeśli ujemne to U2
									1	
	1,	0	0	1	0	1	1	0	0	WYNIK – 44 w U2

Realizacja operacji matematycznych – układy logiczne

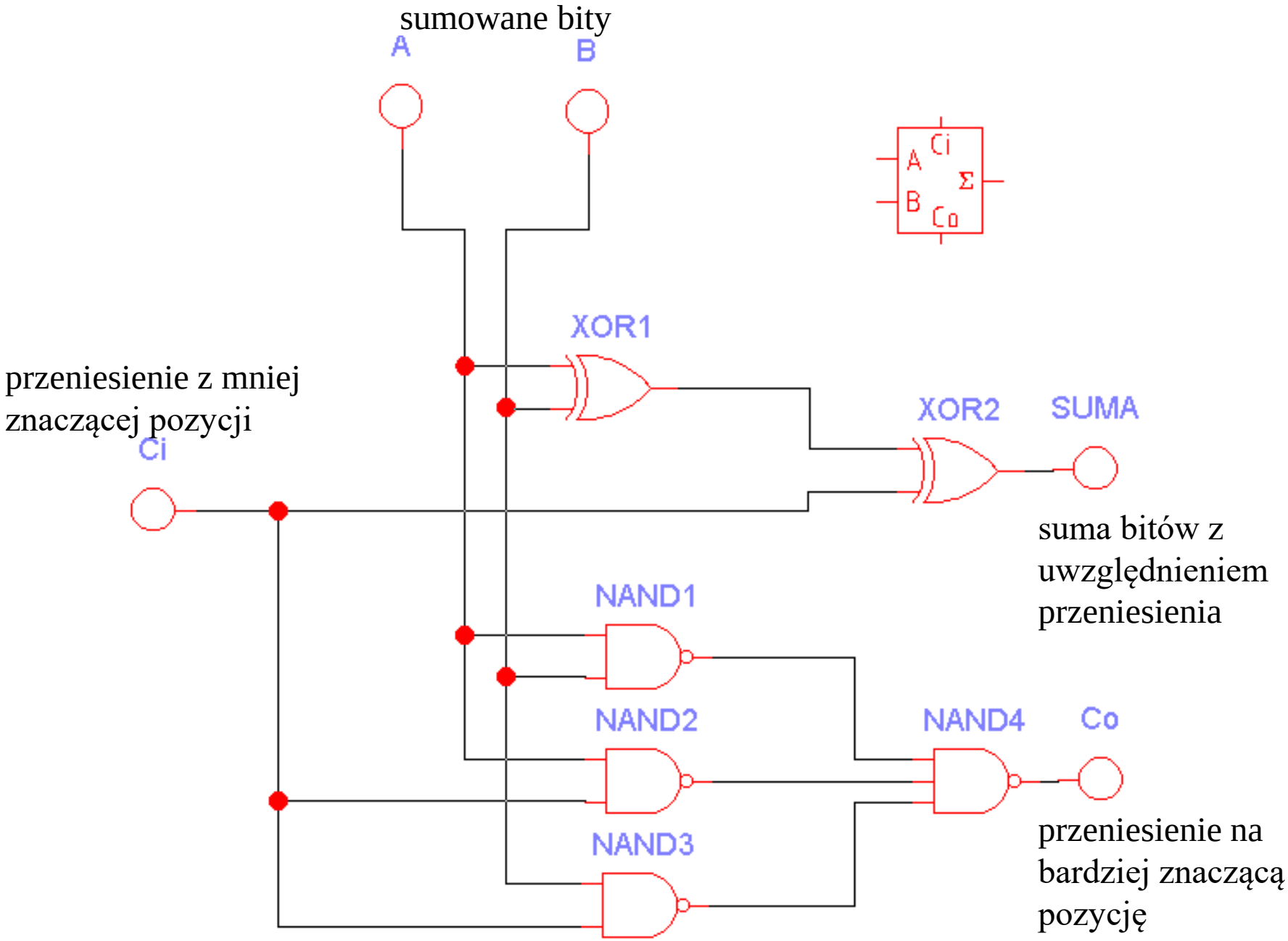
Algebra BOOLE'a

Analiza układów przełączających

- **tablice (siatki) Karnaugha**
- **metoda Quine'a - McCluskeya**

Optymalizacja (minimalizacja) układów

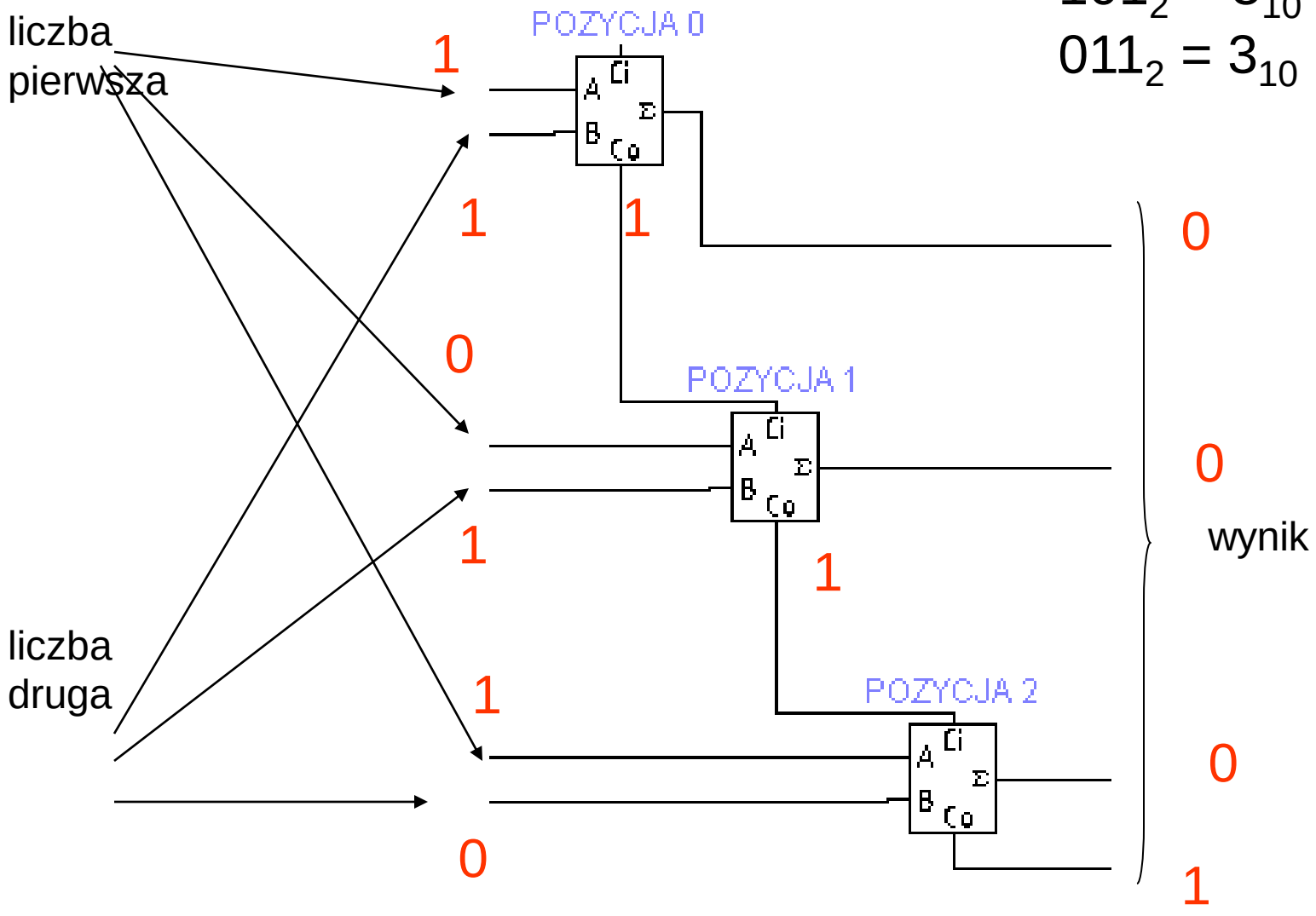
SUMATOR 1-bitowy



SUMATOR 2 liczb 3-bitowych

$101_2 = 5_{10}$

$011_2 = 3_{10}$



$1000_2 = 8_{10}$