Układy kombinacyjne

Układy kombinacyjne są to układy cyfrowe, których stany wyjść są zawsze jednoznacznie określone przez stany wejść.

Oznacza to, że doprowadzając na wejścia tych układów określoną kombinację sygnałów binarnych, otrzymujemy na wyjściach taką samą odpowiedz jak w chwilach poprzednich, gdy ta sama kombinacja była doprowadzana.

Jest to niezależne od tego co działo się z tymi układami wcześniej.

Można zatem stwierdzić, że są to układy bez pamięci.

Bramki logiczne

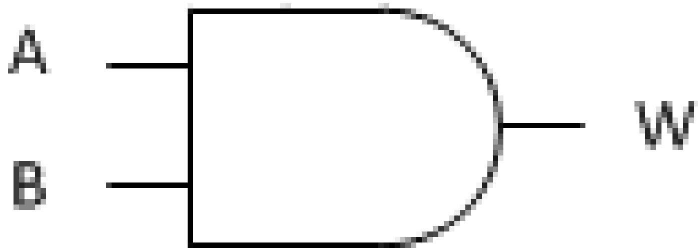
Bramki logiczne są podstawowymi przedstawicielami układów kombinacyjnych

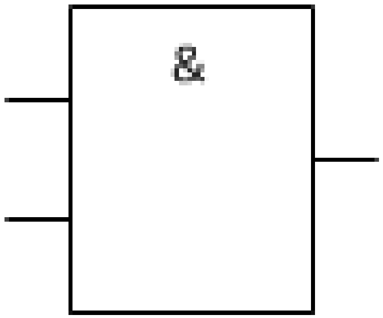
Symbole graficzne bramek logicznych są dwojakie:

- o kształtach zróżnicowanych (wprowadzone przez normy amerykańskie i szeroko stosowane w przemyśle)
- o kształtach prostokątnych (zalecane przez IEC - *Międzynarodową Komisję Elektrotechniczną*)

AND

iloczyn logiczny





równanie logiczne: $\mathbf{W = A \cdot B}$

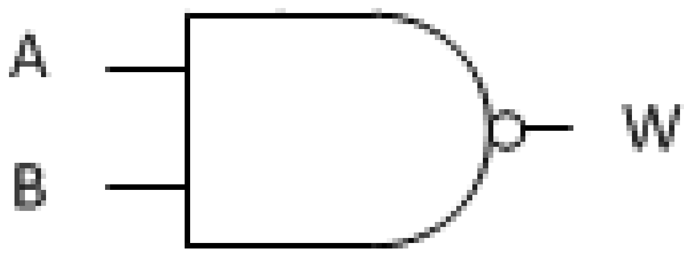
tabela prawdy

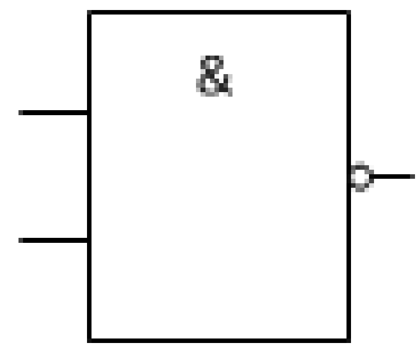
A	B	W
0	0	0
0	1	0
1	0	0
1	1	1

znak „· ”
tu oznacza
<u>iloczyn logiczny</u>
inaczej „∧”
<u>koniunkcję</u>

NAND

negacja iloczynu logicznego





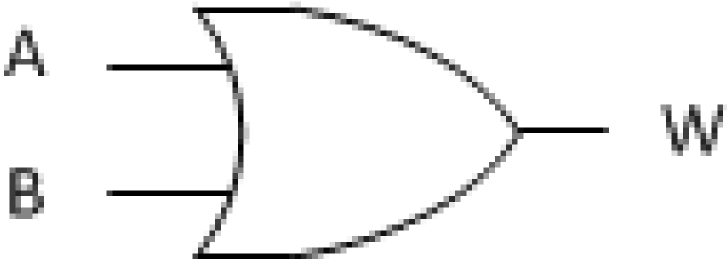
równanie logiczne: $W = \overline{A \cdot B}$

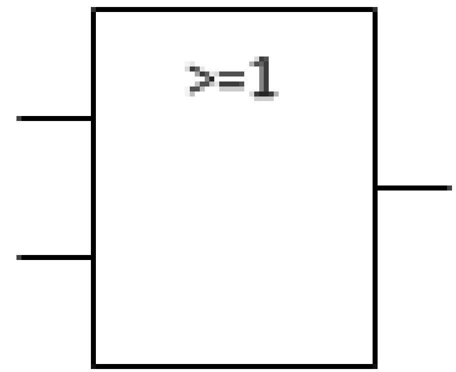
tabela prawdy

A	B	W
0	0	1
0	1	1
1	0	1
1	1	0

OR

suma logiczna





równanie logiczne: W = A + B

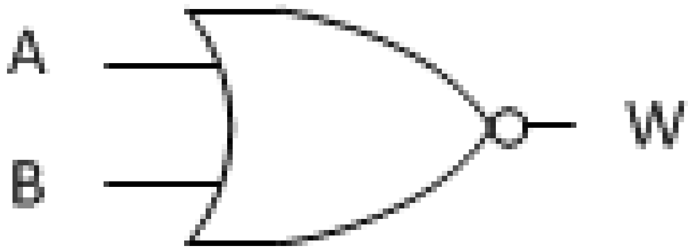
tabela prawdy

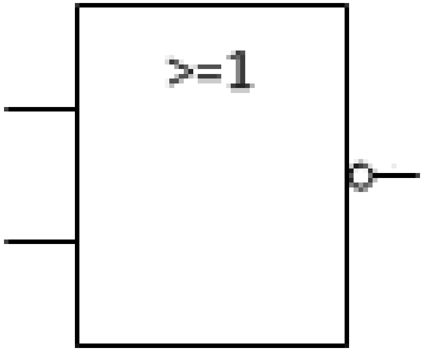
A	B	W
0	0	0
0	1	1
1	0	1
1	1	1

znak „**+**” tu oznacza <u>sumę logiczną</u> inaczej „**v**” <u>alternatywę</u>

NOR

negacja sumy logicznej





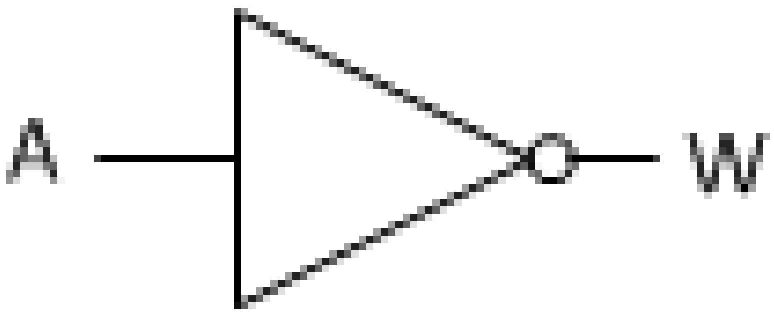
równanie logiczne: $W = \overline{A + B}$

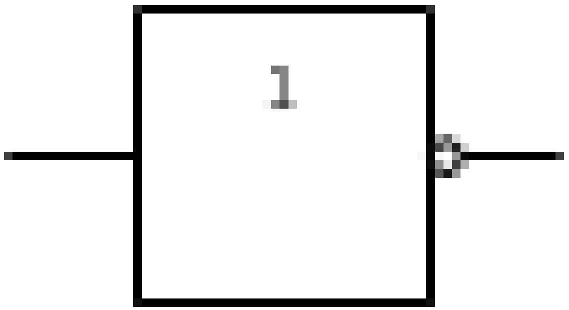
tabela prawdy

A	B	W
0	0	1
0	1	0
1	0	0
1	1	0

NOT

negacja, funktor NOT





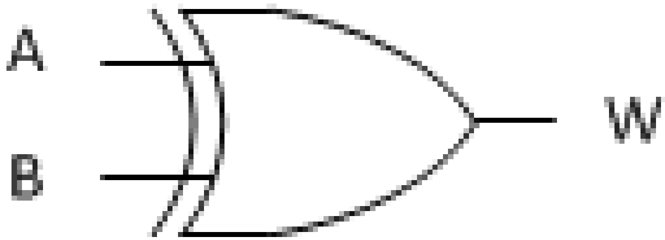
równanie logiczne: $W = \overline{A}$

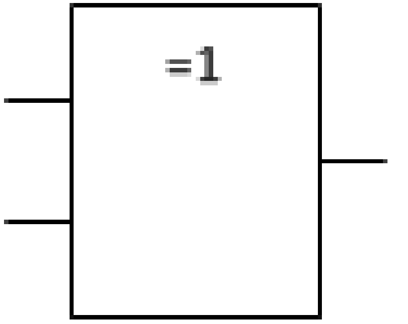
tabela prawdy

A	W
0	1
1	0

XOR, EXOR

Exclusive OR



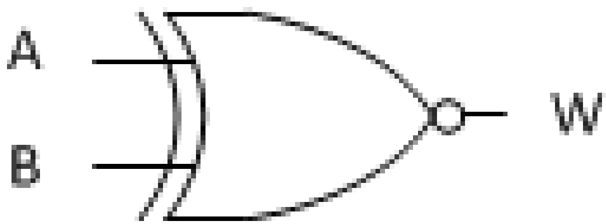


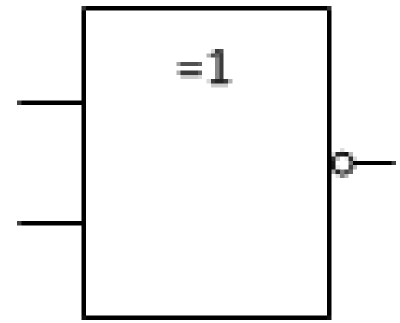
równanie logiczne: $W = A \oplus B = \overline{A}\,B + A\,\overline{B}$

tabela prawdy

A	B	W
0	0	0
0	1	1
1	0	1
1	1	0

XNOR, EXNOR
Exclusive NOR





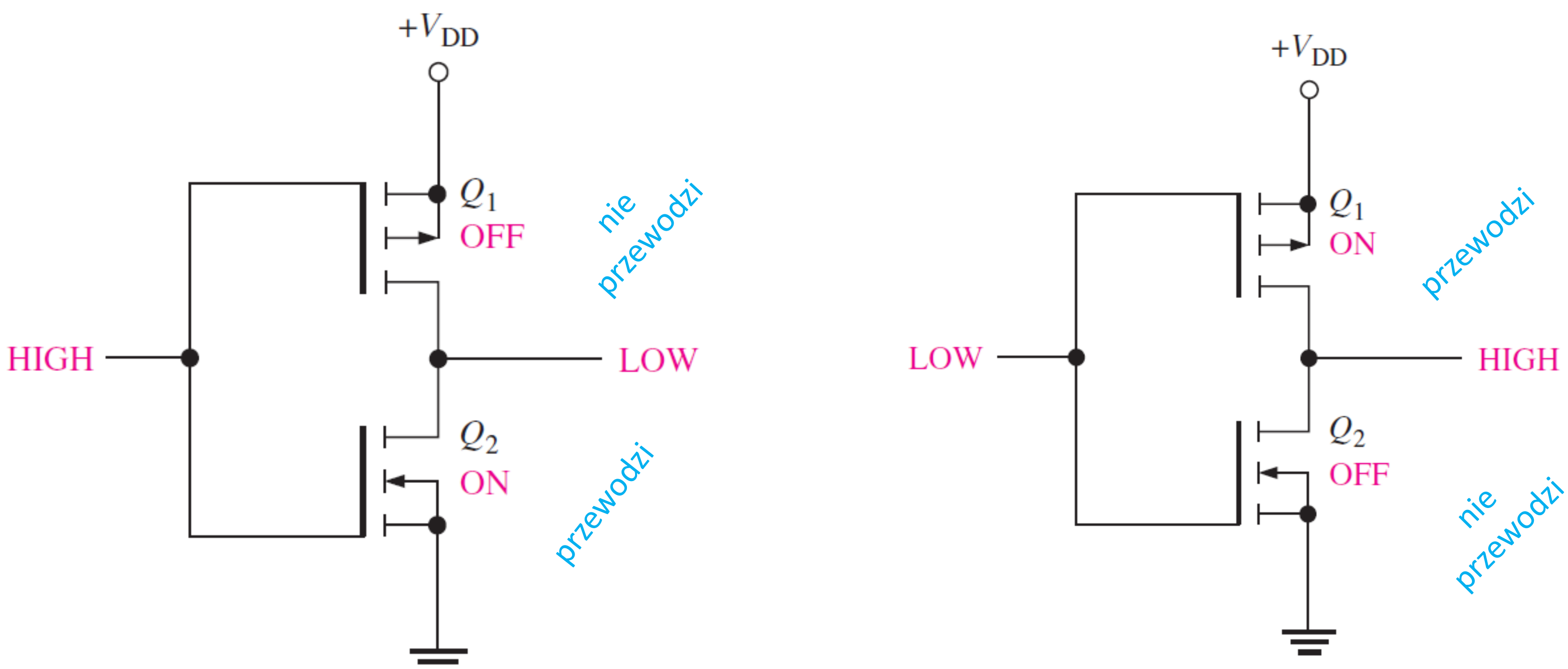
równanie logiczne: $W = \overline{A \oplus B} = \overline{A}\,\overline{B} + A\,B$

tabela prawdy

A	B	W
0	0	1
0	1	0
1	0	0
1	1	1

PRZYKŁAD REALIZACJI BRAMEK LOGICZNYCH

bramka NOT w technice CMOS



inwerter CMOS z zaznaczeniem stanów tranzystorów

Komparatory

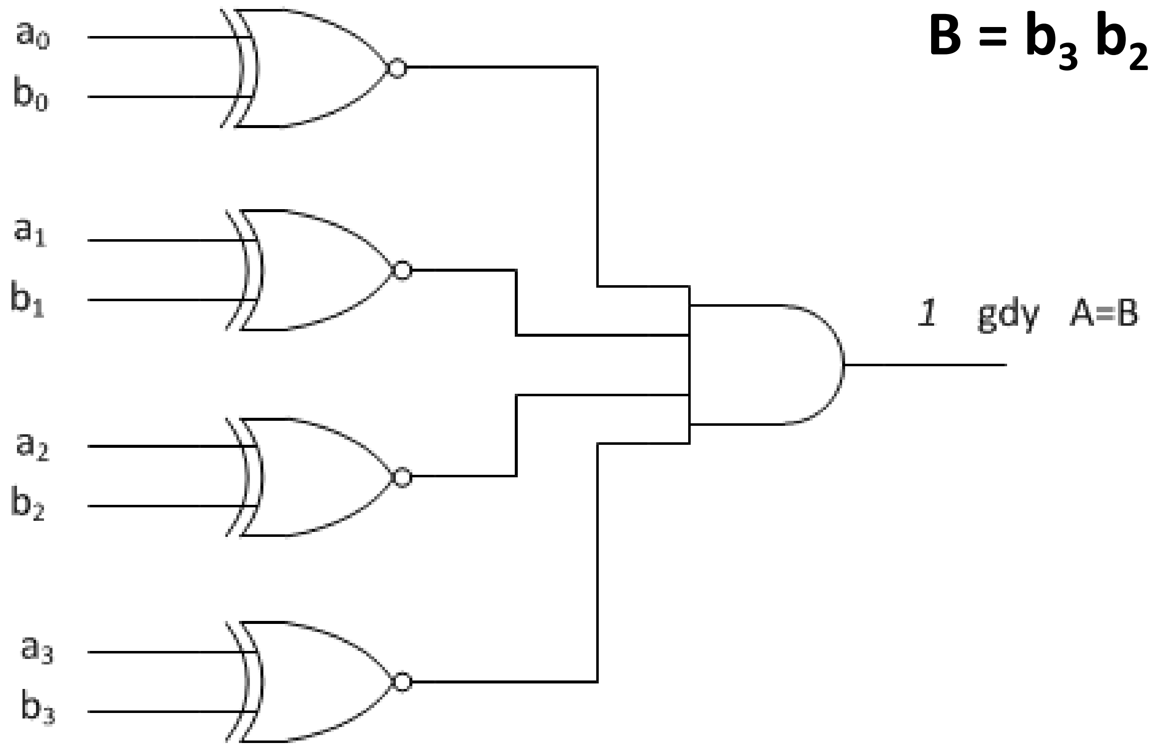
Komparatory są to cyfrowe układy kombinacyjne, które porównują ze sobą dwie liczby.

Najważniejsze kryteria porównania dwóch liczb A i B to:
A>B, A=B i A<B

komparator identyczności (określa równość dwóch liczb)

$A = a_3\ a_2\ a_1\ a_0$

$B = b_3\ b_2\ b_1\ b_0$



AND dwuwejściowy

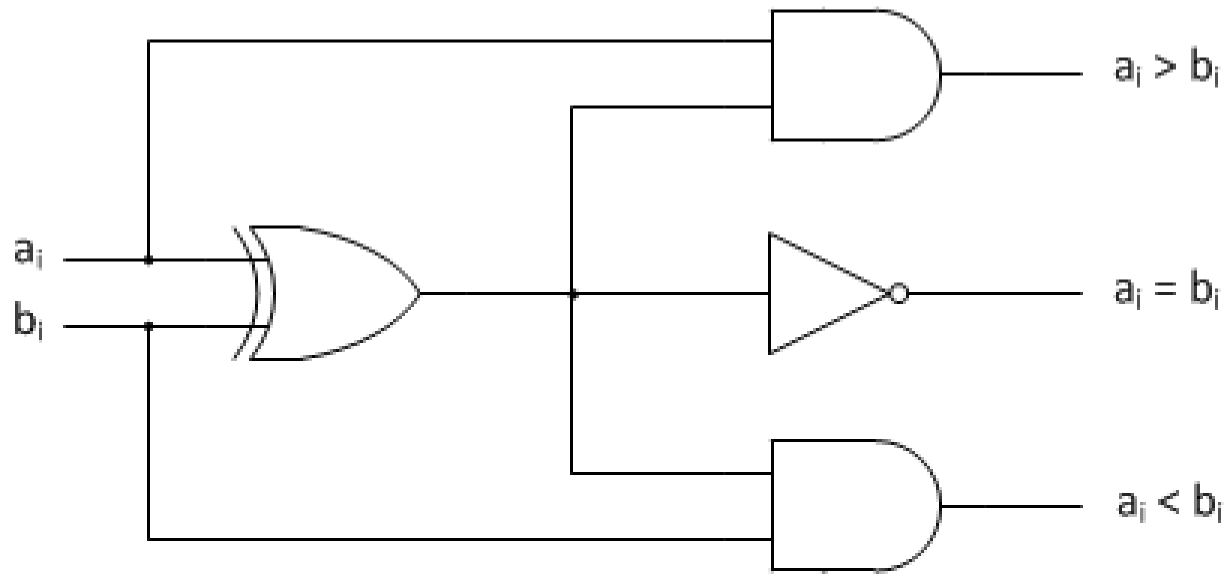
A	B	W
0	0	0
0	1	0
1	0	0
1	1	1

EXNOR

A	B	W
0	0	1
0	1	0
1	0	0
1	1	1

$$\prod_{i=0}^{i=3} \overline{a_i \oplus b_i} = \begin{cases} 1 & gdy\ liczby\ są\ identyczne \\ 0 & gdy\ liczby\ są\ różne \end{cases}$$

jednobitowy komparator wielkości (pokazuje: A=B, A<B, A>B)



EXOR

A	B	W
0	0	0
0	1	1
1	0	1
1	1	0

a_i	b_i	$a_i = b_i$	$a_i > b_i$	$a_i < b_i$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

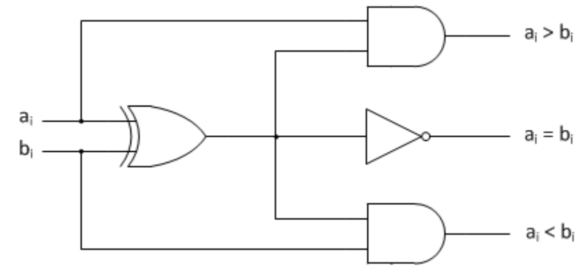
Aby porównać dwie liczby czterobitowe
zapisane w naturalnym kodzie dwójkowym

$$\mathbf{A = a_3\ a_2\ a_1\ a_0}$$

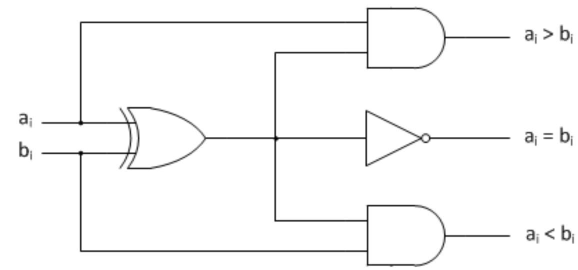
$$\mathbf{B = b_3\ b_2\ b_1\ b_0}$$

trzeba mieć cztery komparatory

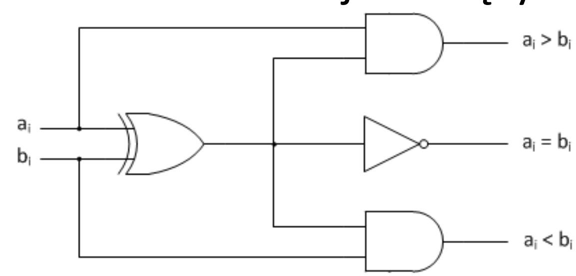
dla bitów najbardziej znaczących **a_3 b_3**



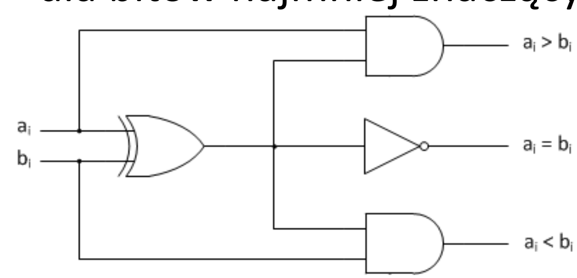
dla bitów mniej znaczących **a_2 b_2**



dla bitów mniej znaczących **a_1 b_1**



dla bitów najmniej znaczących **a_0 b_0**



Działanie komparator wielkości
w procesie porównywania kolejnych bitów dwóch liczb

Niech liczby $A = a_3\ a_2\ a_1\ a_0$ i $B = b_3\ b_2\ b_1\ b_0$ będą zapisane w naturalnym kodzie dwójkowym.
Porównuje się w pierwszej kolejności bity najbardziej znaczące (najstarsze, o największej wadze) i wtedy to, gdy:
$a_3>b_3$ to już wiadomo, że $A>B$ i następuje koniec procesu porównywania
$a_3<b_3$ to już wiadomo, że $A<B$ i następuje koniec procesu porównywania
$a_3=b_3$ przechodzi się do porównywania kolejnych młodszych bitów
i wtedy to, gdy:
$a_2>b_2$ to już wiadomo, że $A>B$ i następuje koniec procesu porównywania
$a_2<b_2$ to już wiadomo, że $A<B$ i następuje koniec procesu porównywania
$a_2=b_2$ przechodzi się do porównywania kolejnych młodszych bitów
i wtedy to, gdy:
$a_1>b_1$ to już wiadomo, że $A>B$ i następuje koniec procesu porównywania
$a_1<b_1$ to już wiadomo, że $A<B$ i następuje koniec procesu porównywania
$a_1=b_1$ porównuje się ostanie najmłodsze bity
i wtedy, gdy:
$a_0>b_0$ to $A>B$
$a_0<b_0$ to $A<B$
$a_0=b_0$ to $A=B$

Sumatory

Sumatory są cyfrowe układy kombinacyjne realizujące operację arytmetycznego sumowania liczb binarnych.

operacja matematyczna sumowania

w systemie dziesiętnym

$$
\begin{array}{r}
22 \\
+\ 28 \\
\hline
50
\end{array}
$$

waga 10^1

waga 10^0

w systemie dwójkowym

$$
\begin{array}{r}
10110 \\
+\ 11100 \\
\hline
110010
\end{array}
$$

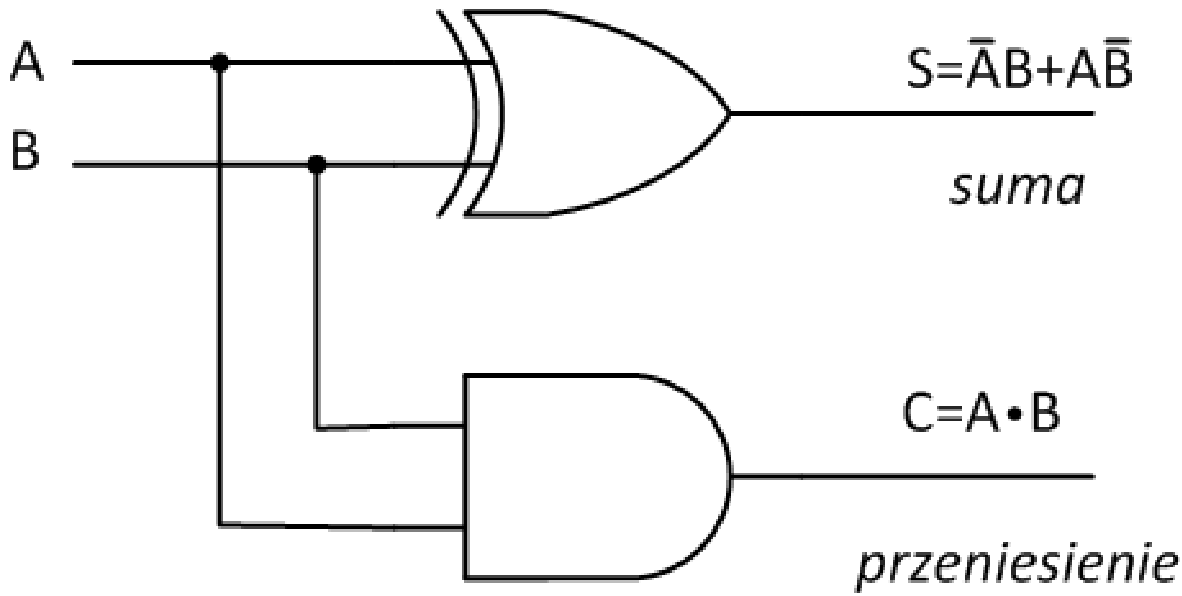
waga 2^5

....

waga 2^0

Reguły dodawania w obu systemach te same.
W pierwszej kolejności dodajemy liczby najmniej znaczące. Jeżeli wynik jest mniejszy od podstawy systemy zapisujemy go poniżej. Jeżeli wynik dodawania jest większy lub równy podstawie systemu dokonywane jest przeniesienie . Przeniesienie dodawane jest do pary cyfr na kolejnej wyższej pozycji wagowej ... itd.

półsumator

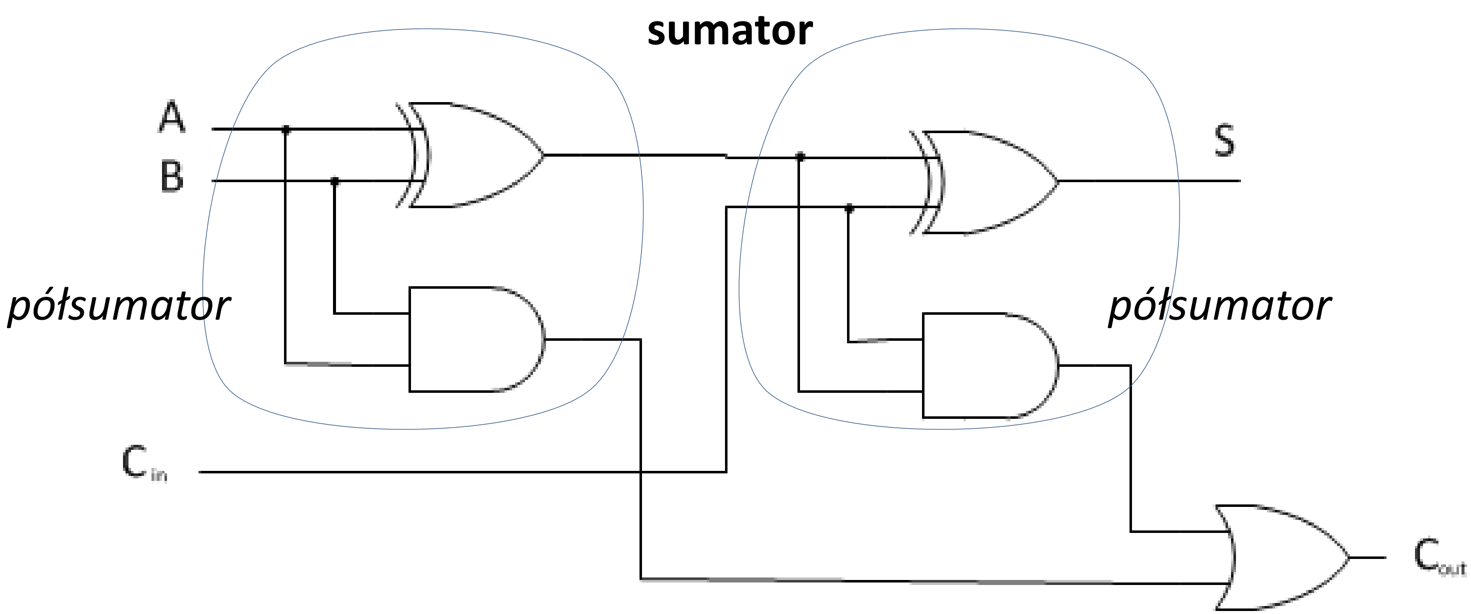


EXOR

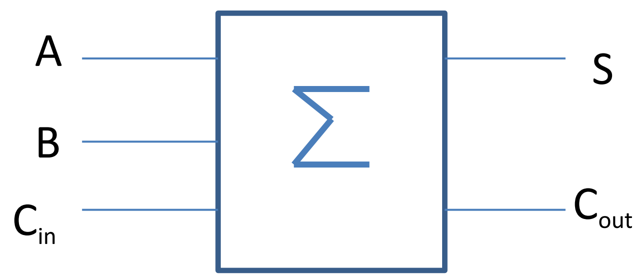
A	B	W
0	0	0
0	1	1
1	0	1
1	1	0

tabela prawdy

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



dwa półsumatory z dodatkową bramką OR tworzą sumator



symbol sumatora

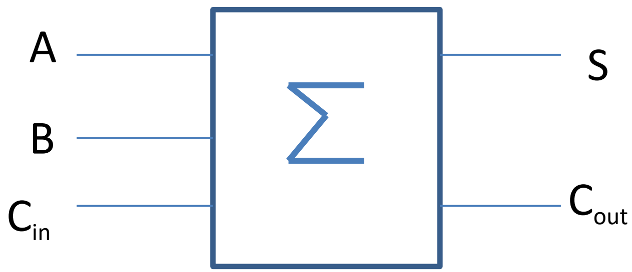
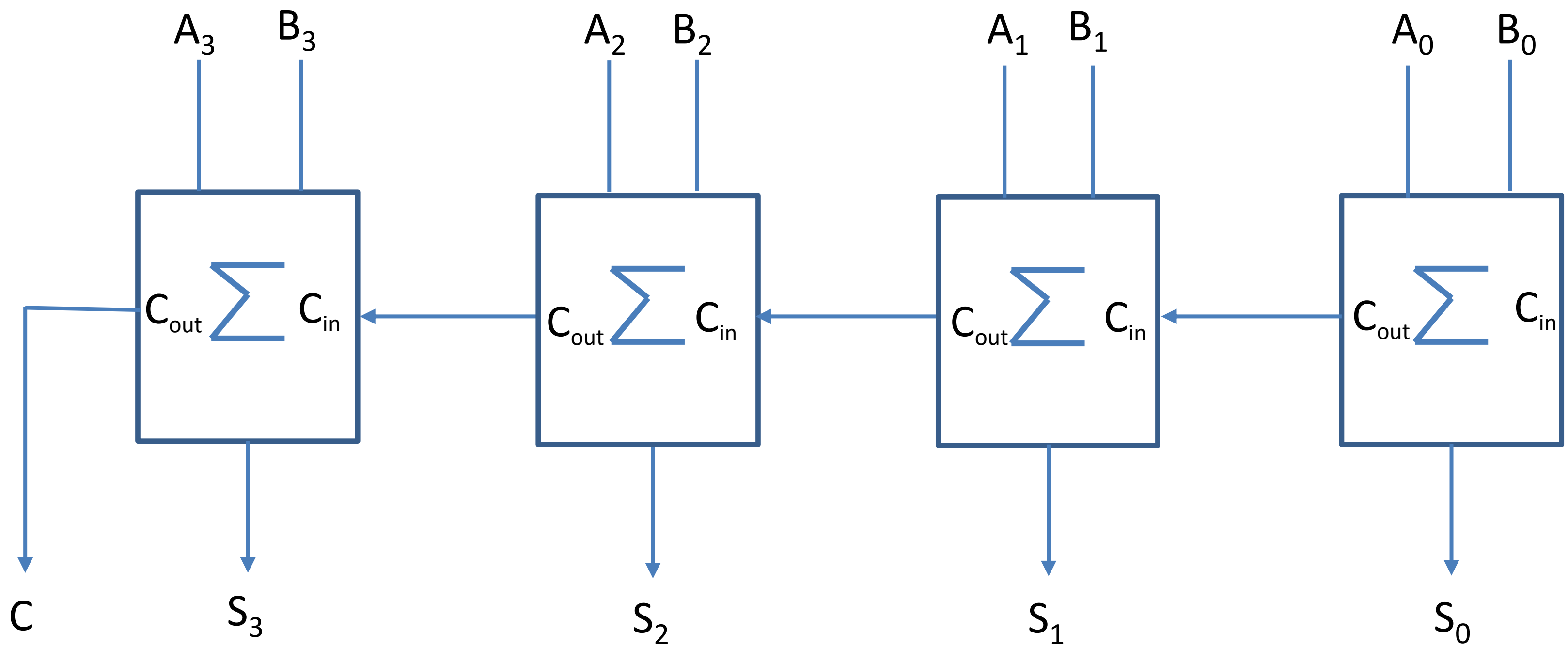


tabela prawdy

A	B	C in	S	C out
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

czterobitowy sumator równoległy z przeniesieniem szeregowym



sumowanie dwóch liczb czterobitowych $A_3A_2A_1A_0$ i $B_3B_2B_1B_0$

otrzymujemy w wyniku liczbę C $S_3S_2S_1S_0$ gdzie piąty bit C jest ewentualnym przeniesieniem z sumowania bitów najstarszych (bitów o największej wadze)

Układy kombinacyjne

KONIEC