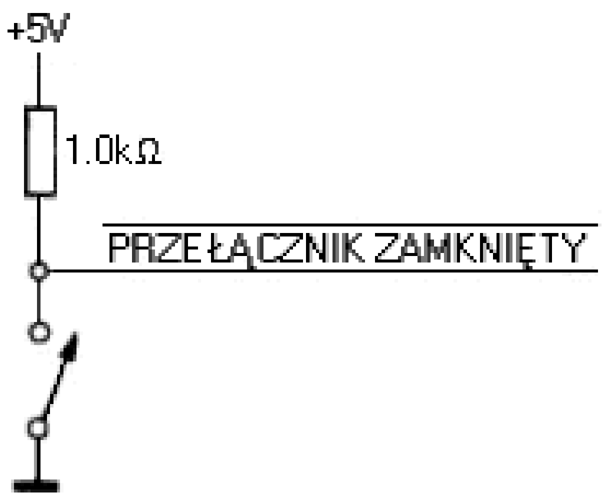
# PODSTAWY

Przez układy cyfrowe rozumiemy układy, w których w każdej chwili występują tylko dwa (zwykle) możliwe stany, np. tranzystor, jako element układu cyfrowego, może być albo w stanie nasycenia, albo w stanie przewodzenia. Z możliwości opisu stanów jako poziomów napięcia lub poziomów prądu zwykle wybiera się to pierwsze rozwiązanie. Mówi się wtedy o poziomie napięcia odpowiadającym stanowi wysokiemu, oznaczanemu symbolem H (ang. high), oraz o poziomie napięcia odpowiadającym stanowi niskiemu, oznaczanemu symbolem L (ang. low).

Te dwa stany mogą reprezentować wiele różnych "bitów" (cyfr dwójkowych) informacji, jak na przykład:
czy przełącznik jest otwarty czy zamknięty?
czy sygnał jest obecny czy nieobecny?
czy zaszło już pewne wydarzenie czy nie?
itd.

## Stan wysoki (H) i stan niski (L)

Można zdefiniować odpowiedniość między stanami: wysokim (H) i niskim (L) układu cyfrowego a elementami algebry Boole'a, określanymi jako "prawda" i "fałsz". Jeśli pojawiającemu się w pewnej linii układu stanowi H przypiszemy "prawdę", wówczas tę linię nazywamy *aktywną stanem wysokim* (lub *aktywną w stanie wysokim*), a przyjętą konwencję nazywamy *logiką dodatnią.*



**jednostka** - standard stosowany do zliczania oddzielnych elementów

**wielkość** - bezwzględna lub fizyczna ilość jednostek
**liczba** - symboliczny zapis przedstawiający wielkość
**system liczbowy** - sposób przedstawienia wielkości za pomocą zbioru znaków lub liczb; zero wskazuje brak jednostek, a inne symbole wskazują wielkości.
**podstawa systemu** - ilość symboli używanych w systemie
np. reprezentacja stałoprzecinkowa Liczby całkowitej bez znaku:

$$\text{Liczba} = a_k * p^k + a_{k-1} * p^{k-1} + .. + a_1 * p^1 + a_0 * p^0 = \sum_{i=0}^{k} a_i * p^i$$

p - podstawa systemu, p = 2, 10, 16, ..

$a_k$ - wartości kolejnych pozycji, $a_k$ = 0, .. , p- 1

k - numery kolejnych pozycji, k = 0, 1, 2, ..

## reprezentacja stałoprzecinkowa

- liczba pozycji przeznaczonych dla zakodowania części całkowitej i ułamkowej jest ustalona; bezwzględna dokładność reprezentacji liczb jest stała; między innymi:

• **system stałobazowy** (fixed-base system) - system pozycyjny, każdej pozycji przypisana jest waga (mnożnik), a zbiór dozwolonych cyfr i znaków jest identyczny dla każdej pozycji

• **system uzupełnieniowy** (radix-complement system) - modyfikacja systemu stałobazowego, w którym waga najbardziej znaczącej pozycji jest ujemna, w stosunku do wag pozostałych pozycji; reprezentacja liczb znakowych

## reprezentacja zmiennoprzecinkowa

-osobno kodowane są:

•**znacznik S** (Significand) - znak liczby (sign); np. S=0 dla liczb dodatnich, S=1 dla liczb ujemnych

•**część ułamkowa M** (fraction), dawniej mantysa (Mantissa); zwykle w kodzie uzupełnienia do 2 z przesunięciem o 127

•**wykładnik E** (Exponent) - określający relację między liczbą i jej znacznikiem w postaci potęgi bazy (radix) użytego systemu liczbowego; MSB=1 i nie jest pamiętany

# Wybrane systemy liczbowe

| system | podstawa systemu | symbole |
|---|---|---|
| **dziesiętny** (decimal) | p = 10 | $a_k$ = 0, 1, .. 8, 9<br><br>193 |
| **dwójkowy** (binary) | p = 2 | $a_k$ = 0, 1<br><br>1100 0001b |
| **szesnastkowy** (hexadecimal) | p = 16 | $a_k$ = 0, 1, .. 8, 9, A, B, C, D, E, F<br>0C1h |

## BINARNY SYSTEM CYFROWY

**Zamiast pojęcia cyfr i miejsc znaczących, w systemie binarnym używane jest pojęcie bitu zapożyczone z teorii informacji. Poszczególnym bitom odpowiadają różne wagi, zależnie od położenia bitu.**

**Bit położony najbardziej w prawo jest najmniej znaczący (LSB Least Signifiant Bit) i ma wagę $2^0 = 1$ (przy liczbie całkowitej).**

**Bit najbardziej z lewej MSB (Most sign. Bit) i ma wagę $2^{n-1}$, przy czym n – ilość bitów w liczbie.**

**Odczyt – sumowanie bitów, oznaczonych wartością 1:**

$$(11001)_2 = 2^4 + 2^3 + 2^0 = (25)_{10}$$

**Maksymalna liczba dziesiętną, jaką można zapisać przy n – bitach, jest równa $2^n – 1$.**

**Przekształcenie zapisu dziesiętnego w zapis binarny wymaga przedstawienia danej liczby w postaci sumy potęg liczny 2:**

$$a_{n-1}2^{n-1} + \ldots + a_2 2^2 + a_1 2^1 + a_0 2^0 + a_{-1}2^{-1} + \ldots + a_{-m}2^{-m}$$

**n – liczba bitów wyrażająca część całkowitą liczby,**
**m – liczba bitów wyrażająca część ułamkową,**
**$a_i$ – współczynniki rozkładu (0 lub 1)**

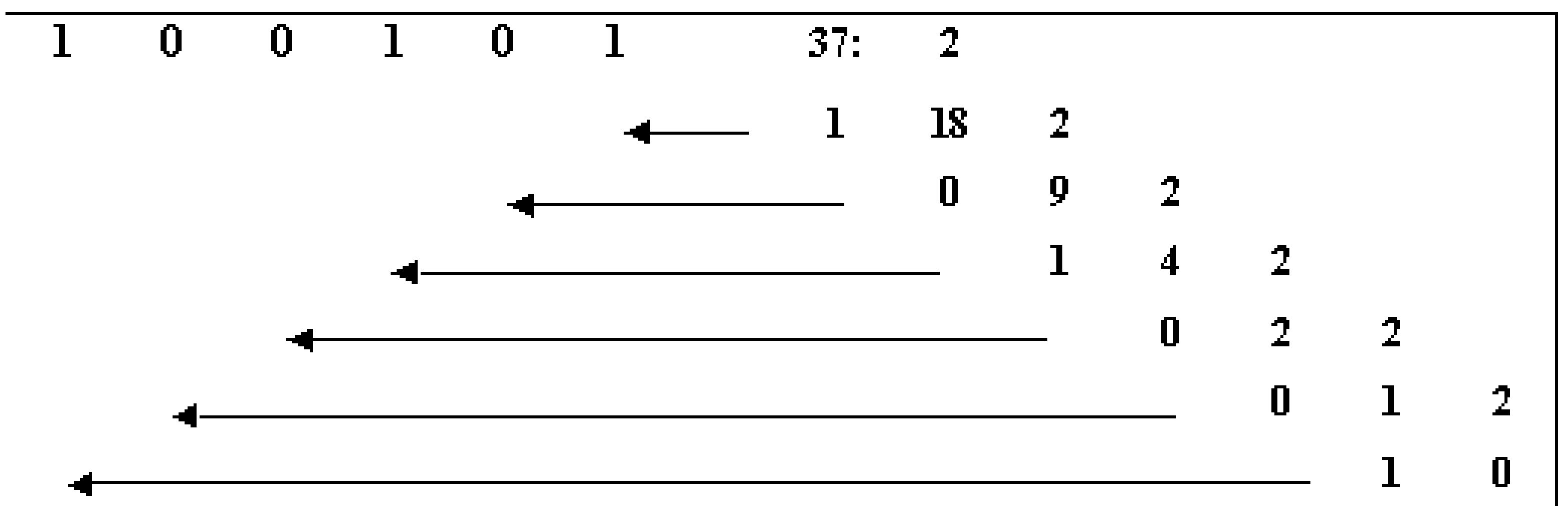
**Aby określić współczynniki rozkładu można stosować kolejne dzielenie części całkowitej przez 2 i mnożenia części ułamkowej przez 2.**

**Współczynniki części całkowitej są równe resztom z kolejnych operacji dzielenia, które kończą się po osiągnięciu wyniku równego 0.**

**Dla części ułamkowej współczynniki są równe części całkowitej kolejnego wyniku mnożenia. Przy każdorazowym wystąpieniu jedynki w części całkowitej jest ona odrzucana. (możliwe tylko w niektórych przypadkach w większości trzeba stosować przybliżenie – ograniczać ilość bitów po przecinku).**

**Przykład 37,625**

Część całkowita



Część ułamkowa

$$
\begin{array}{r}
0{,}625 \\
\times\ 2 \\
\hline
1{,}250 \\
\hline
0{,}25 \\
\times\ 2 \\
\hline
0{,}50 \\
\hline
0{,}50 \\
\times\ 2 \\
\hline
1{,}00
\end{array}
$$

.1 0 1

**Ostatecznie:** $(37{,}625)_{10} = (100101.101)_2$

# DZIAŁANIA ARYTMETYCZNE

Liczba

| dziesiętna | binarna |
|---|---|
| 0+0=0 | 0+0=0 |
| 0+1=1 | 0+1=1 |
| 1+1=2 | 1+1=10 |
| 1+1+1=3 | 1+1+1=11 |

**Dodawanie (przykład):**

**$(5)_{10} + (13)_{10}$**

$$
\begin{array}{cccccl}
 & & 1 & 0 & 1 & \\
+ & 1 & 1 & 0 & 1 & \\
\hline
1 & 0 & 0 & 1 & 0 & = (18)_{10}
\end{array}
$$

**Odejmowanie (przykład):**

$(12)_{10} – (7)_{10}$

| | | | | | |
|---|---|---|---|---|---|
| | 1 | 1 | 0 | 0 | |
| - | | 1 | 1 | 1 | |
| | 0 | 1 | 0 | 1 | $=(5)_{10}$ |

**Mnożenie (przykład):**

**$(7)_{10} \bullet (5)_{10}$**

$$
\begin{array}{cccccc}
 & & & 1 & 1 & 1 \\
 & & \times & 1 & 0 & 1 \\
\cline{2-6}
 & & & 1 & 1 & 1 \\
 & + & 0 & 0 & 0 & \\
 & 1 & 1 & 1 & & \\
\hline
1 & 0 & 0 & 0 & 1 & 1
\end{array}
\; = (35)_{10}
$$

## TOŻSAMOŚCI LOGICZNE

**$0+a = a+0 = a$**

**$1+a = a+1 = 1$**

**Prawo łączności iloczynu**

$$(a \bullet b) \bullet c = a \bullet (b \bullet c)$$

**Prawo łączności sumy**

$$(a+b)+c = a+(b+c)$$

**Prawo przemienności iloczynu**

$$a \bullet b = b \bullet a$$

**Prawo przemienności sumy**

$$a+b = b+a$$

**Prawo indempotentności iloczynu**

$$a \bullet a = a$$

**Prawo indempotentności sumy**

$$a+a = a$$

**Prawo rozdzielności iloczynu względem sumy**

$$a\bullet(b+c) = (a\bullet b)+(a\bullet c)$$

**Prawo rozdzielności sumy względem iloczynu**

$$a+(b\bullet c) = (a+b)\bullet(a+c)$$

**Prawa pochłaniania**

$$a(b+a) = a$$

$$a + b\bullet a = a$$

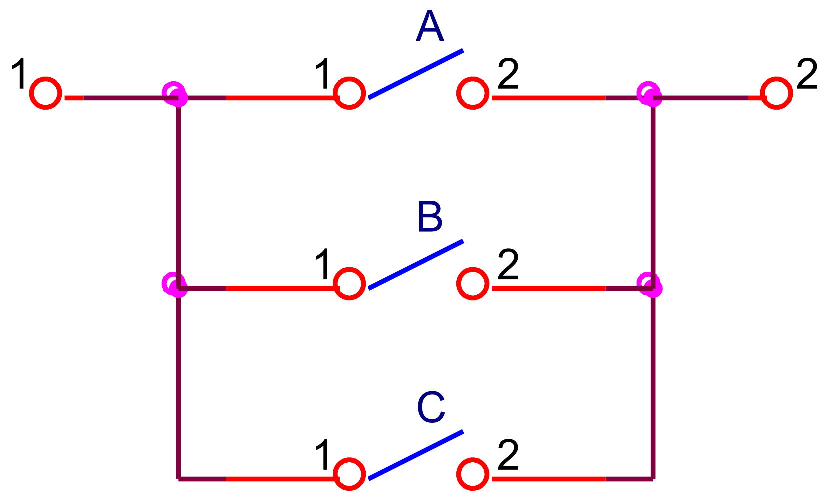
**wyprowadzenie przy użyciu powyższych wzorów:**

$$a\bullet(b+a) = (0+a)\bullet(b+a) = 0\bullet b + a = 0+a = a \qquad !$$

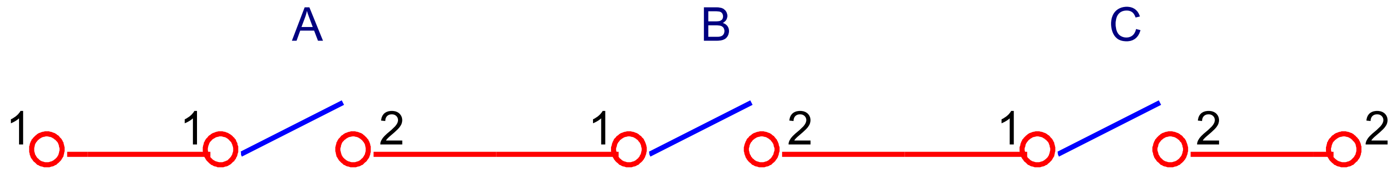
$$a+b\bullet a = 1\bullet a + b\bullet a = (1+b)\bullet a = 1\bullet a = a \qquad !$$

## INTERPRETACJA GRAFICZNA TOŻSAMOŚCI LOGICZNYCH:

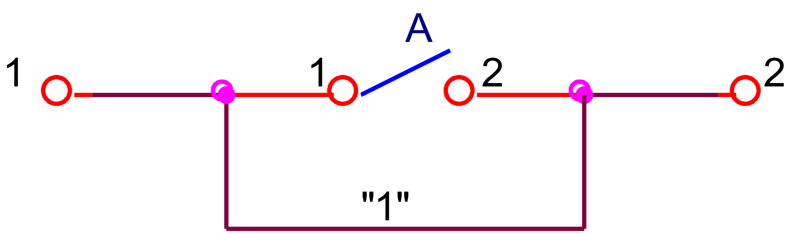
**A+B+C**




**A•B•C**

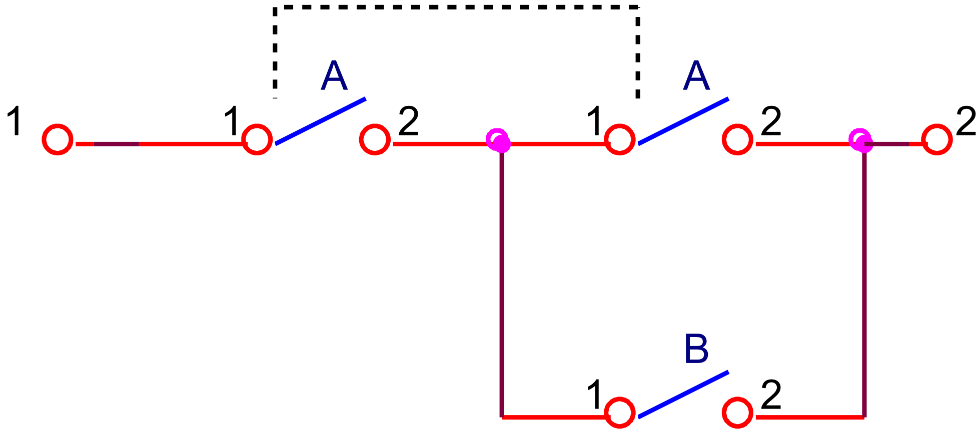


**A+1=1**

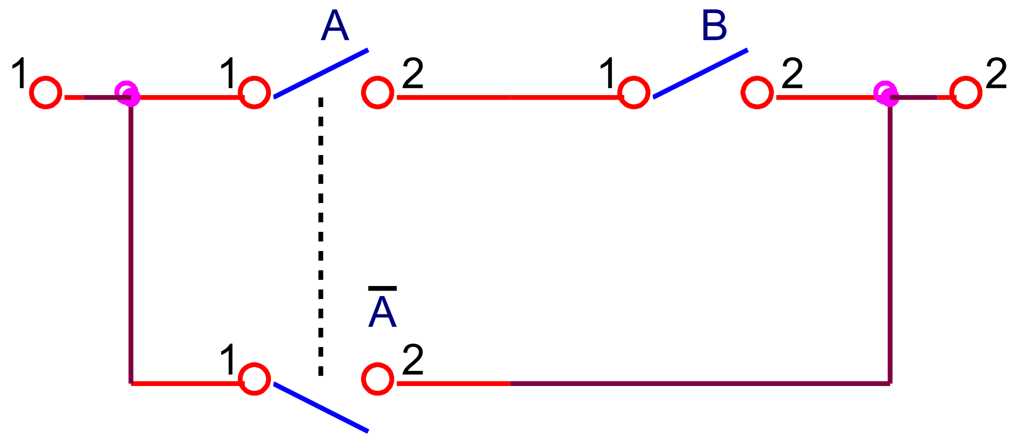



**A • 0 = 0**


**A (A + B) = A**



**/A + AB = /A + B**



**Kod** - zbiór symboli kodowych oraz reguła wzajemnego przyporządkowania informacji tym symbolom

**Kod dwójkowo-dziesiętny** (BCD - Binary Coded Decimal) - przyporządkowanie cyfrom dziesiętnym 4-bitowych kombinacji binarnych, najczęściej w systemie 8421, np.

1234 = 4D2 h = 0100 1101 0010 b = 0001 0010 0011 0100 BCD

**Kod Gray-a** - kod, którego słowa reprezentujące kolejne liczby różnią się wartością tylko jednego bitu (w kodzie binarnym w połowie zakresu zmieniane są wszystkie bity)

## Reprezentacje liczb ujemnych:

**kod znak-moduł** (sign-magnitude) - znak liczby kodowany jest dwuwartościowo na najbardziej znaczącej pozycji:
*wartość_ujemna liczby obliczana przez negację najbardziej znaczącego bitu:*

np. dla 8-bitowego zapisu binarnego liczbie +7 = 0000 0111b odpowiada liczba ujemna: - 7 = 1000 0111b

podwójna reprezentacja zera: +0 = 0000 0000b oraz - 0 = 1000 0000b

symetryczny zakres liczb, np. dla liczb 8-bitowych: - 127 = 1111 1111b .. +127 = 0111 1111b

**kod uzupełnienia do 1** (1’s complement) - kod uzupełnienia niepełny, oznaczany jako U1, najbardziej znacząca pozycja traktowana jest jako kod znaku;

wartość ujemna liczby obliczana jako:
*wartość_ujemna = not (wartość_dodatnia)*

np. dla 8-bitowego zapisu binarnego liczbie +7 = 0000 0111b odpowiada liczba ujemna: - 7 = not 7 = 1111 1000b

podwójna reprezentacja zera: +0 = 0000 0000b oraz - 0 = 1111 1111b

symetryczny zakres liczb, np. dla liczb 8-bitowych: - 127 = 1000 0000b .. +127 = 0111 1111b

**kod uzupełnienia do 2** (2’s complement) - kod uzupełnienia pełny, oznaczany jako U2, najbardziej znacząca pozycja traktowana jest jako kod znaku

wartość ujemna liczby obliczana jako:

*wartość_ujemna = Baza- wartość_dodatnia = not(wartość_dodatnia) +1*

np. dla 8-bitowego zapisu binarnego liczbie +7 = 0000 0111b odpowiada liczba ujemna: - 7 = 28 - 7 = 249 = 1111 1001b

pojedyncza reprezentacja zera: 0 = 0000 0000b

niesymetryczny zakres liczb, np. dla liczb 8-bitowych: - 128 = 1000 0000b .. +127 = 0111 1111b