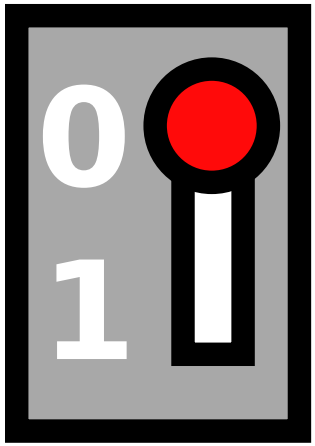
Technologie Informacyjne

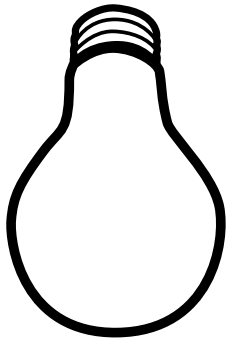
System binarny

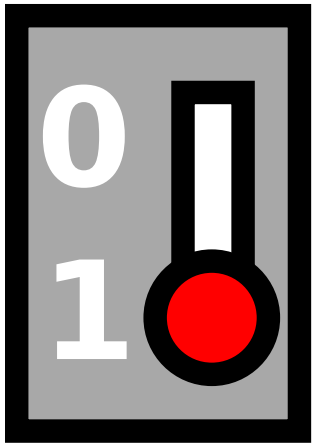
Adam Krasuski

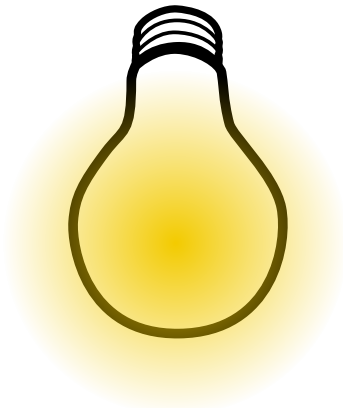
Szkoła Główna Służby Pożarniczej
Zakład Matematyki i Informatyki

October 25, 2019


0
1



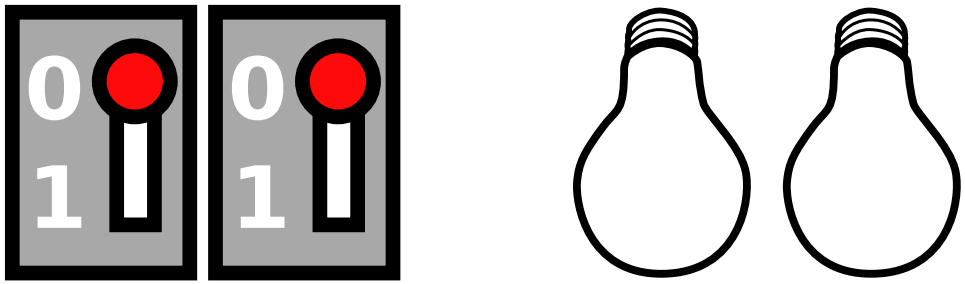


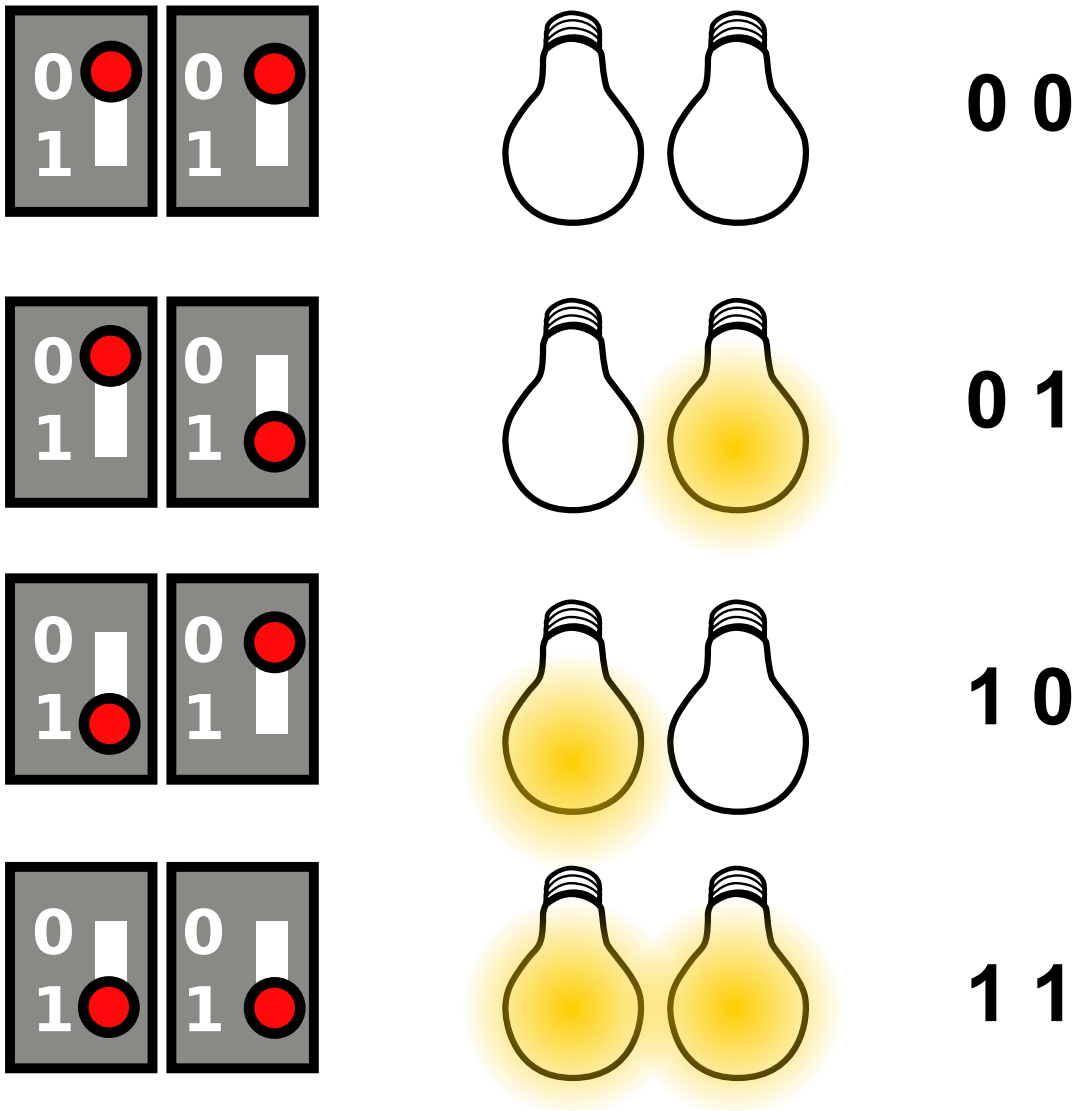


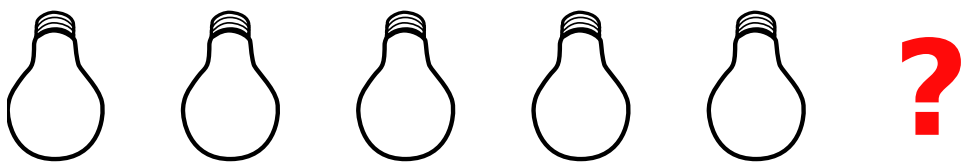
Zmiana stanu jest informacją (pojawiła się zmiana).
Brak zmian to brak informacji.
Podstawową jednostką informacji jest **bit** (binary digit).

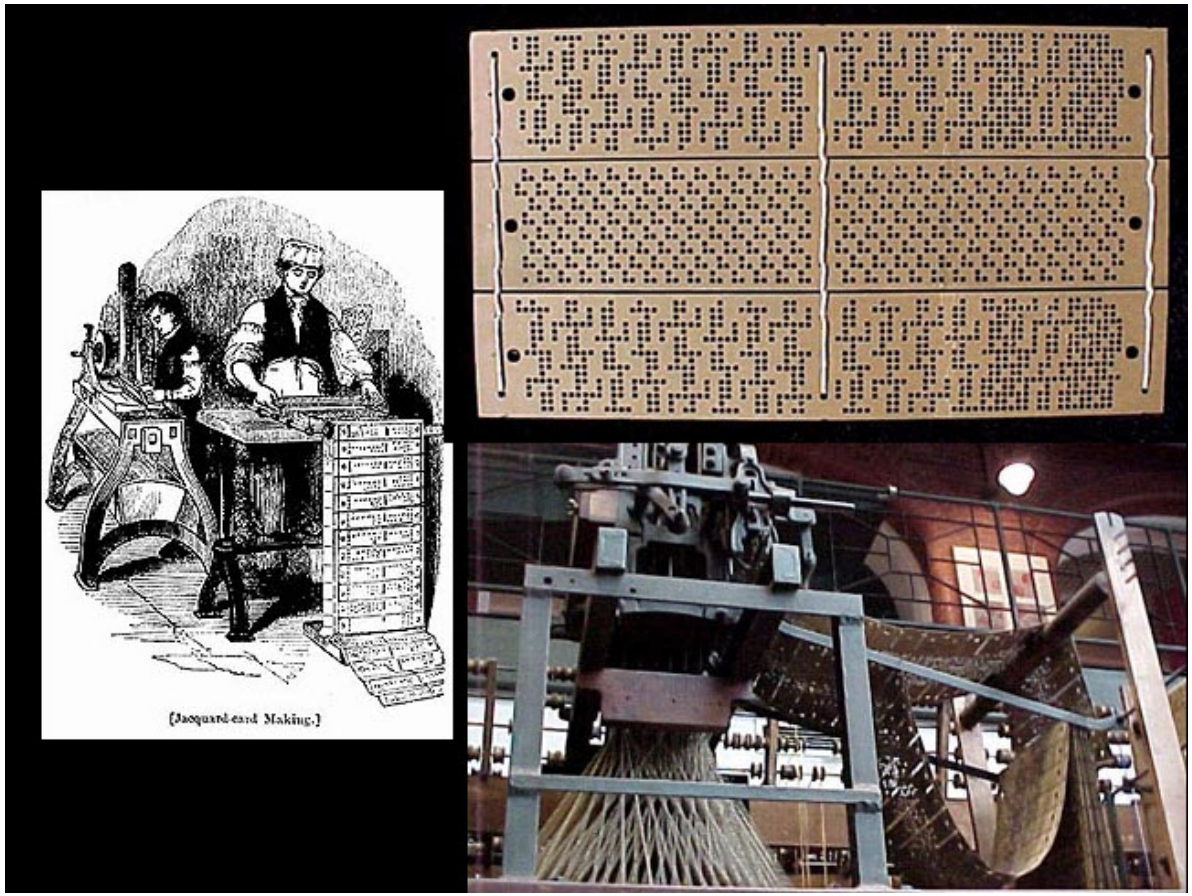
Bit - jednostka ilości informacji wystarczająca do zakomunikowania jednego z co najwyżej dwóch równo prawdopodobnych zdarzeń.

Ile możliwych stanów istnieje dla dwóch żarówek?




0
1
0
1
0 0
0
1
0
1
0 1
0
1
0
1
1 0
0
1
0
1
1 1





źródło: people.duke.edu

Ile różnych liczb można zapisać za pomocą trzech cyfr dziesiętnych XYZ?

0 0 0
0 0 1
0 0 2
...
...
...
9 9 7
9 9 8
9 9 9

Ile różnych liczb można zapisać za pomocą trzech cyfr dziesiętnych XYZ?

0 0 0
0 0 1
0 0 2
...
...
...
9 9 7
9 9 8
9 9 9

Odpowiedź: Wszystkie kolejne liczby od najmniejszej (000) do największej (999). W sumie 1000 liczb.

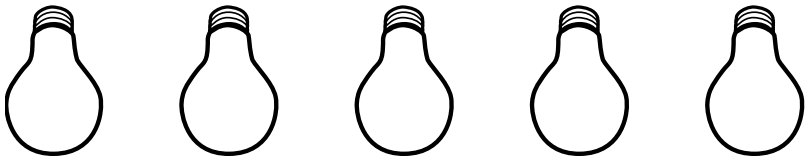
W dowolnym systemie liczbowym można przedstawić

$$P^n$$

P - podstawa systemu
n - ilość cyfr

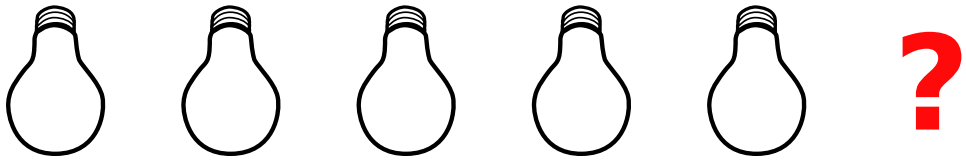
liczb/kombinacji.

P^n



00000=0
00001=1
00010=2
00011=3
00100=4
00101=5
00110=6
00111=7
01000=8
...
11110=30
11111=31

P^n



5 bitów pozwala na 2^5(32) kombinacji

Przykładowa liczba 907 w systemie dziesiętnym powstaje wg poniższego schematu:

$$\begin{matrix} 10^2 & 10^1 & 10^0 \\ 9 & 0 & 7 \end{matrix} = 9*100+0*10+7*1=907_{10}$$

Analogicznie, przykładowa liczba 1101 w systemie dwójkowym ma postać:

$$\begin{matrix} 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & 0 & 1 \end{matrix} = 1*8+1*4+0*2+1*1=13_{10}$$

Reprezentacja ułamków

10^2 10^1 10^0 10^{-1} 10^{-2}

9 0 7,2 3

2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} 2^{-4}

1 0 1,1 0 0 1

4 0 1 $\frac{1}{2}$ 0 0 $\frac{1}{16}$

W każdym systemie liczbowym możliwe jest przedstawienie dowolnej liczby.

Konwersja z systemu dziesiętnego do binarnego

121	1 (LSB)
60	0
30	0
15	1
7	1
3	1
1	1 (MSB)
0	

$$121_{10} = 1111001_2$$

System dwójkowy jest naturalnym systemem informatyki

Zalety

- łatwy zapis informacji za pomocą zjawisk elektrycznych, świetlnych, magnetycznych;
- łatwe pomiary stanów (odczyt);
- odporność na zakłócenia;

Wady

- długość zakodowanej informacji;
- przyzwyczajenia człowieka do systemu dziesiętnego.

Ważniejsze potęgi dwójki

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$

$2^8 = 256 = 1$ bajt

1bajt=8bitów

10101111

$2^{16} = 65.536$

$2^{24} = 16.777.216$

2^{10}bajtów=1kB **(1024)**

2^{20}bajtów=1MB **(1024*1024)**

2^{30}bajtów=1GB **(1024*1024*1024)**

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

1 0 1 0 1 1 0 0

8 bitów = 1 bajt

2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

1 0 1 0 0 1 0 0 1 0 1 0 0 1 0 1

16 bitów = 2 bajty

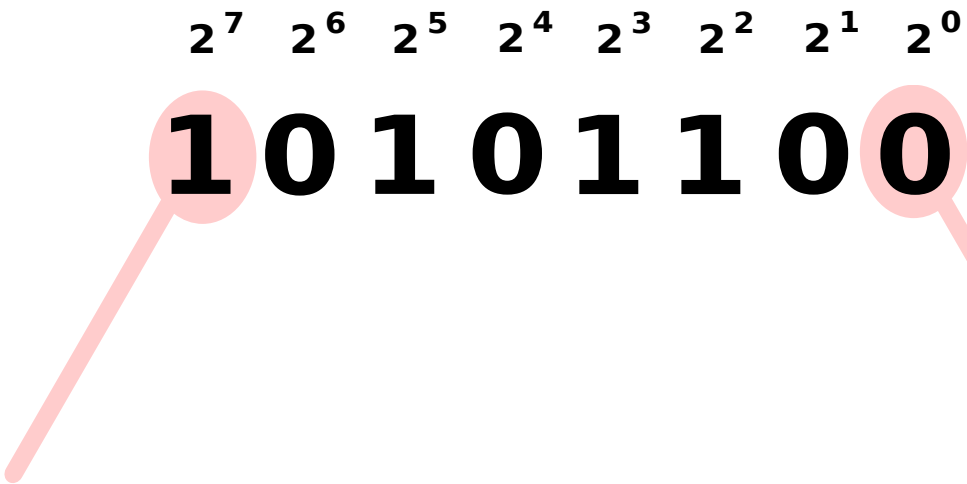
Przybliżanie potęg dwójki

Kolor 24 / 32 bitowy, dźwięk 16 bitowy

$$2^{24} = 2^{10} * 2^{10} * 2^{4} = 1024 * 1024 * 16 \approx 16\,000\,000$$

$$2^{32} \approx 1\,000\,000\,000 * 4$$

$$2^{16} \approx ?$$



Most Significant Bit /
Najbardziej znaczący bit

Least Significant Bit /
Najmniej znaczący bit

$$\overset{18}{0}\,1\,0\,1\,\overset{16}{1}\,\overset{15}{1}\,\overset{14}{1}\,\overset{13}{1}\,0\,1\,\overset{11}{0}\,1\,\overset{9}{1}\,\overset{8}{0}\,1\,\overset{6}{0}\,1\,\overset{4}{0}\,0\,\overset{1}{1}\,0_2 =$$

$$= 262144_{10}+65536_{10}+32768_{10}+16384_{10}+8192_{10}$$

$$+2048_{10}+512_{10}+256_{10}+64_{10}+16_{10}+2_{10}$$

$$= 387922_{10}$$

System szesnastkowy

Celem wprowadzenia systemu szesnastkowego (hexadecymalnego) jest skrócenie zapisu zero - jedynkowego.
System dziesiętny (decymalny) niezbyt nadaje się do tego celu.

100111111010011000011101

hex	bin	dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Każde 4 bity da się przedstawić za pomocą jednej cyfry szesnastkowej.

1001111110100110000 1101					
?	**F**	**D**	**3**	**0**	**D**

hex	bin	dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

16^5 16^4 16^3 16^2 16^1 16^0

4 F D 3 0 D

$$=4*16^5+15*16^4+13*16^3+$$

$$+3*16^2+0*16^1+13*16^0$$

hex	bin	dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

$$0\,1\,0\,1\,1\,1\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,0\,1\,0_2 =$$

$$= 262144_{10}+65536_{10}+32768_{10}+16384_{10}+8192_{10}$$

$$+2048_{10}+512_{10}+256_{10}+64_{10}+16_{10}+2_{10}$$

$$= 387922_{10}$$

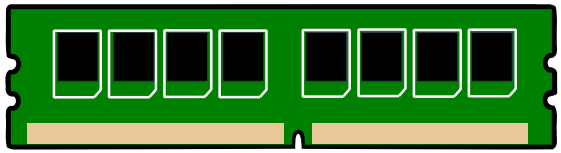
$$01011110101101010010_2$$

$$0101\ 1110\ 1011\ 0101\ 0010_2$$

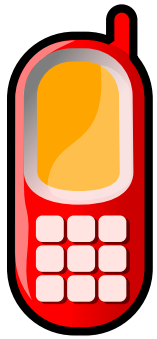
$$\mathbf{5\quad E\quad B\quad 5\quad 2}_{16}$$

$$5EB52_{16}$$

Kodowanie informacji



11010101
01010101
0101010

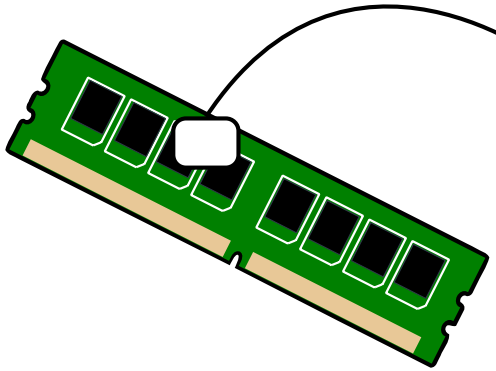


Kod ASCII

A 01000001
B 01000010

7 bitów

American Standard Code for Information Interchange
Kod przypisujący 7-bitowe (128 kombinacji) ciągi do znaków.

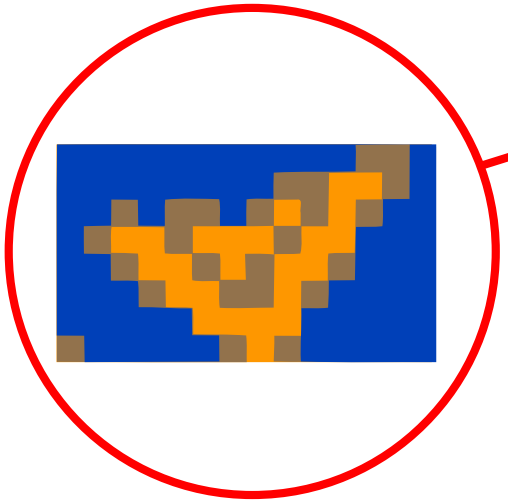


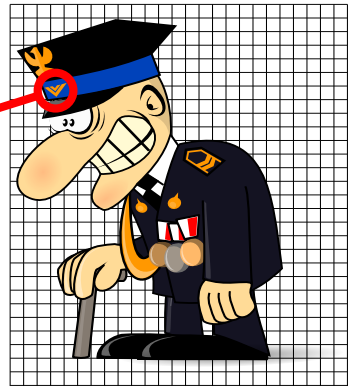
0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1
0	1	1	1	1	0	0	1	1	1	0	1	1	1	1	0	0	1
1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1

128 kombinacji wystarcza do zakodowania wszystkich liter i cyfr oraz kilkudziesięciu znaków drukowalnych (+ - =...) i niedrukowalnych znaków sterujących (np. nowy wiersz).

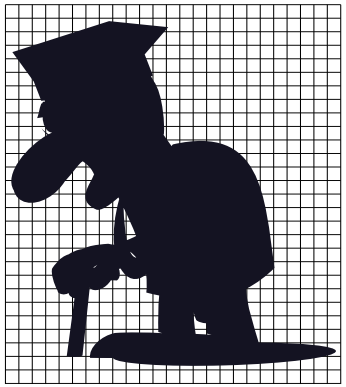
Rozbudowanie kodu do 8 bitów pozwala na przypisanie znaków narodowych (ąęäö...). Przykładowo Europa Centralna używa dla swoich alfabetów rozszerzenia iso-8859-2, a Europa Zachodnia iso-8895-1.

Kodowanie plików graficznych





24 bity / pixel (fullcolor)

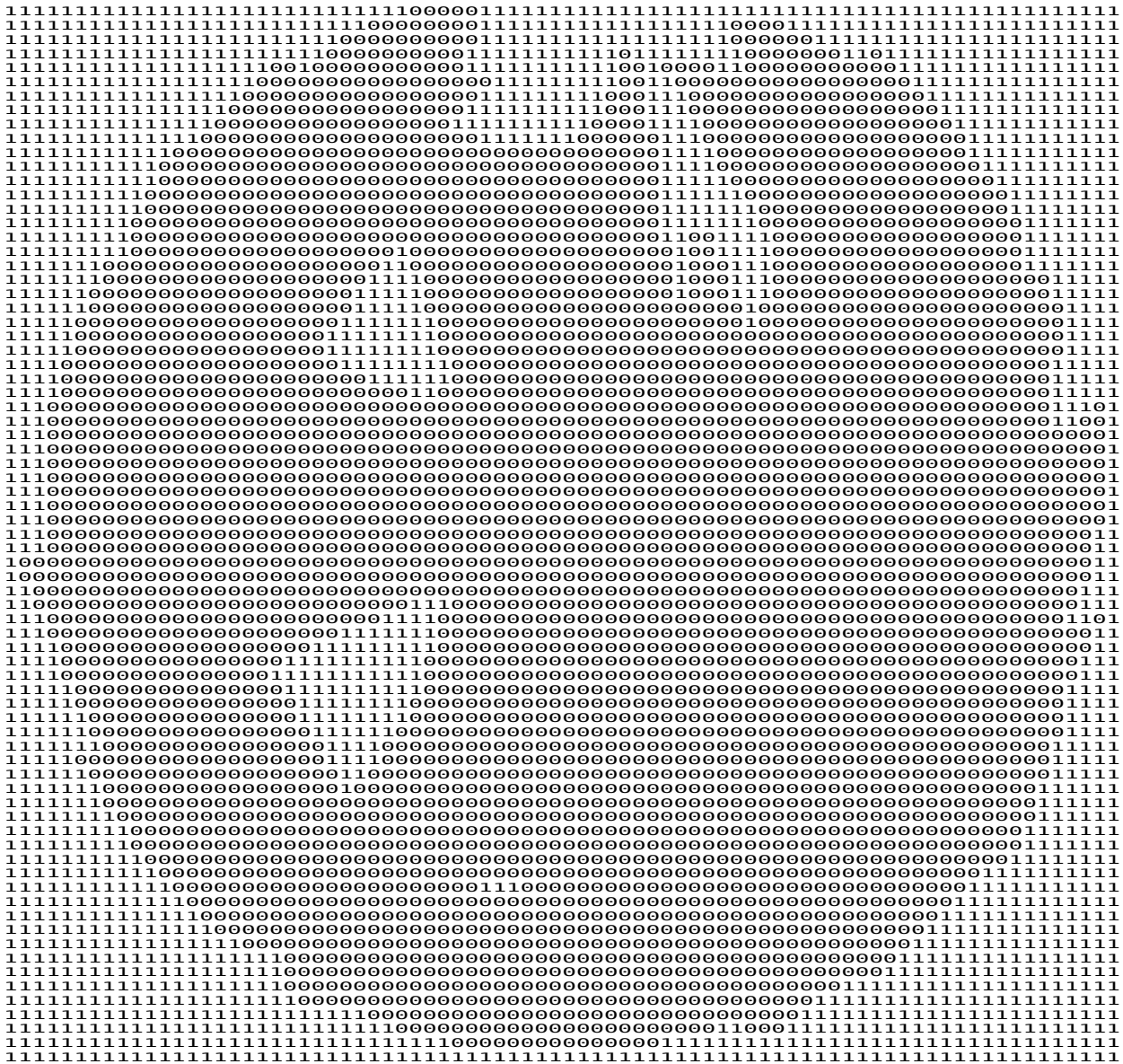


1 bit / pixel

Interpretacja zer i jedynek wynika z kontekstu

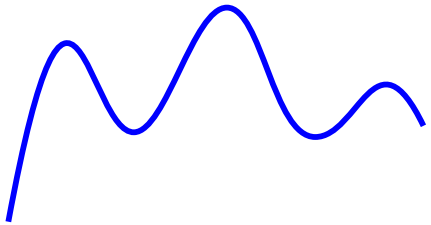
```
1010111010101
0001001010101
0010100101010
1001010101010
1010001010100
1010100101010
0101001011111
1101010101010
1010101010010
```

```
1010111010101
0001001010101
0010100101010
1001010101010
1010001010100
1010100101010
0101001011111
1101010101010
1010101010010
```



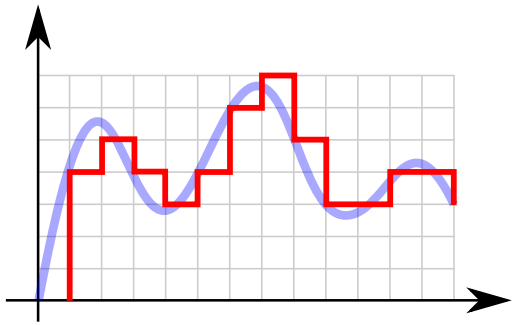
Cyfrowa reprezentacja danych

Sygnały występujące w naturze (np. światło, dźwięk, fala morska) są w zdecydowanej większości analogowe/ciągłe. Oznacza to, że nawet najmniejsza zmiana w sygnale jest zauważalna.



sygnał analogowy

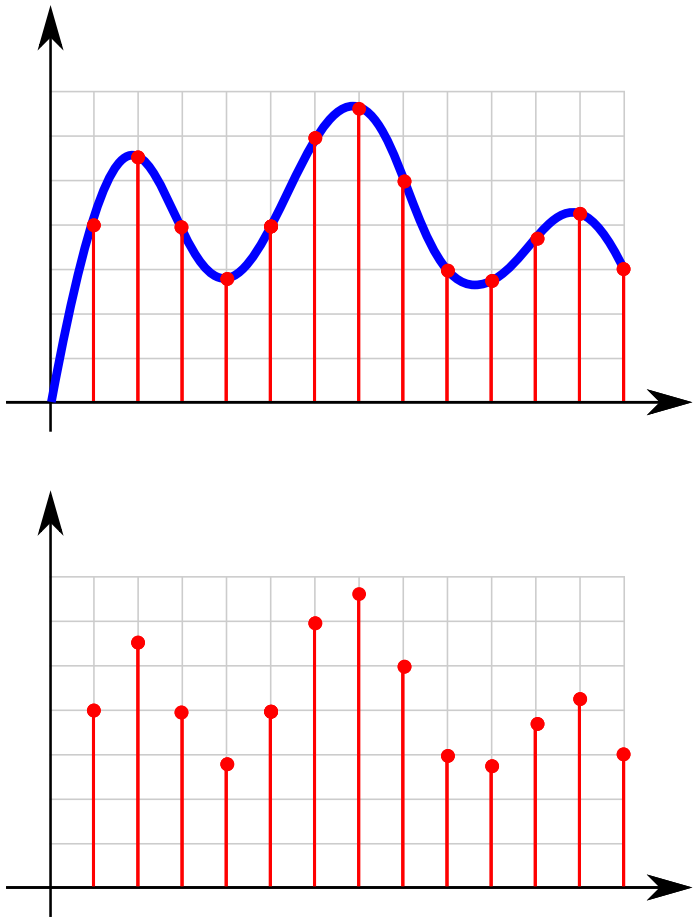
Komputer nie jest w stanie przechować nieskończonej ilości danych opisujących sygnały analogowe; nie jest też w stanie przetwarzać dowolnie długich liczb. W związku z tym sygnał należy poddać aproksymacji i zapisać jako skończony zbiór elementów.



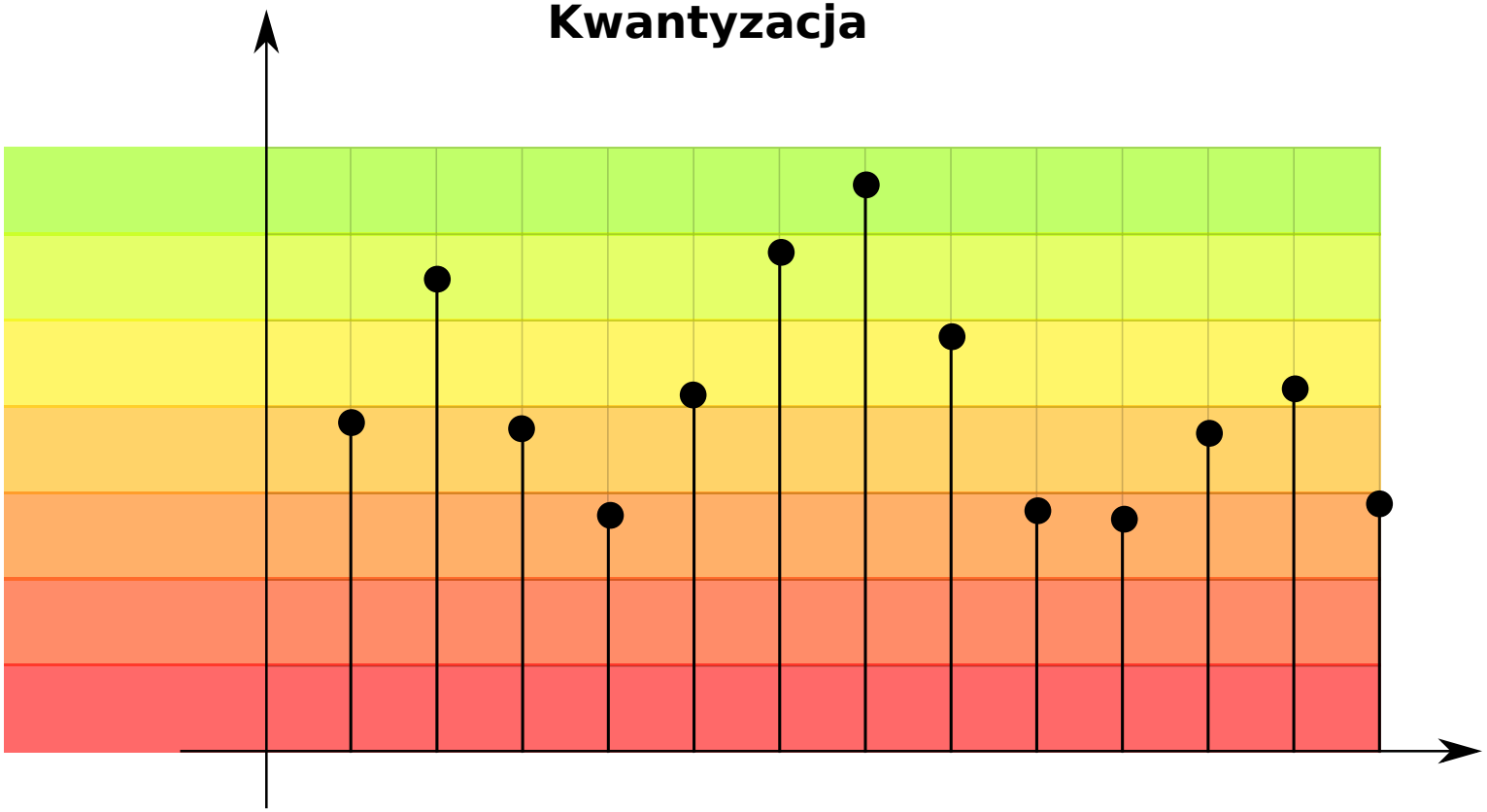
sygnał cyfrowy

Próbkowanie (dyskretyzacja)

Proces przetwarzania sygnału analogowego na sygnał dyskretny

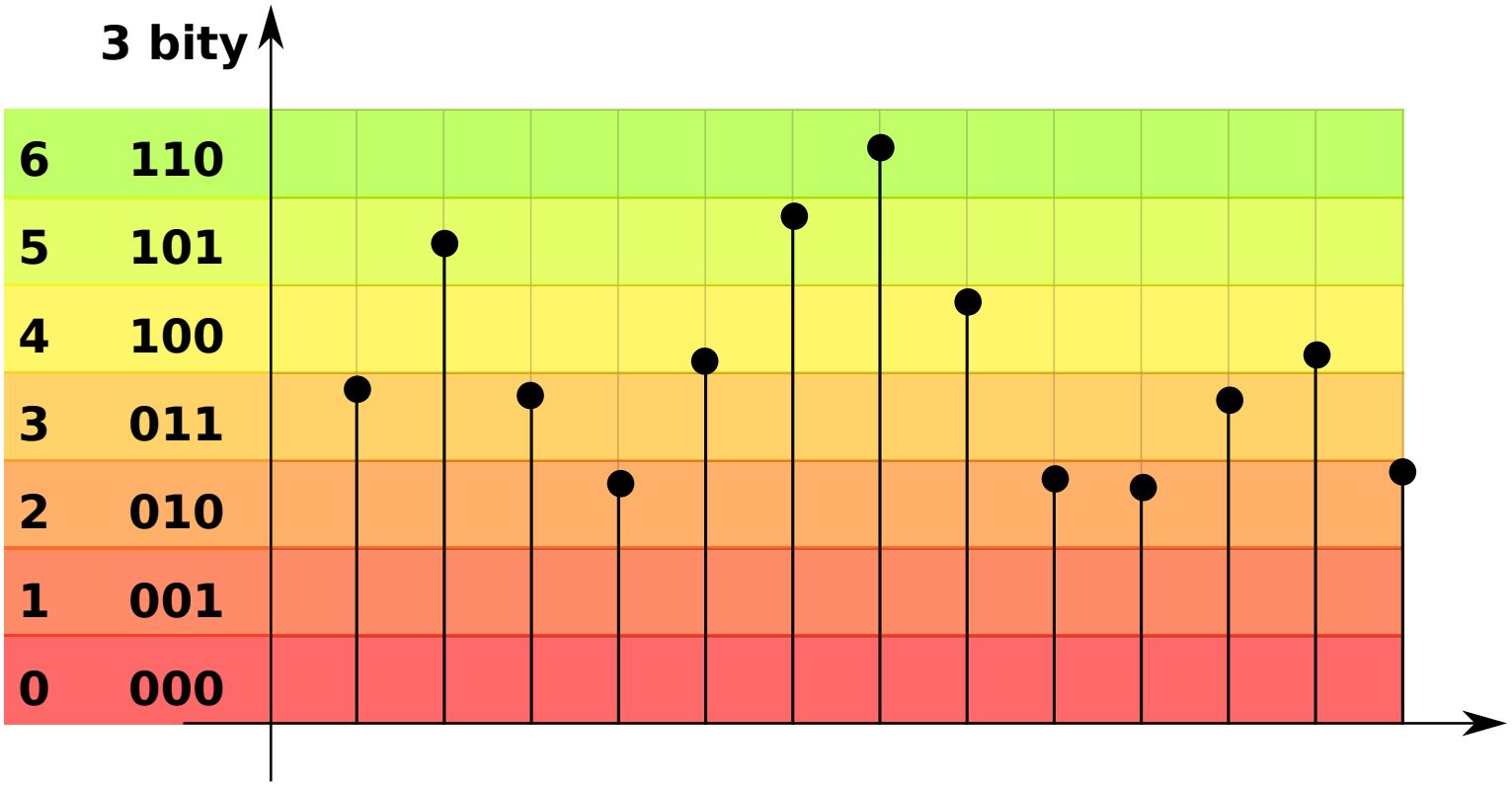


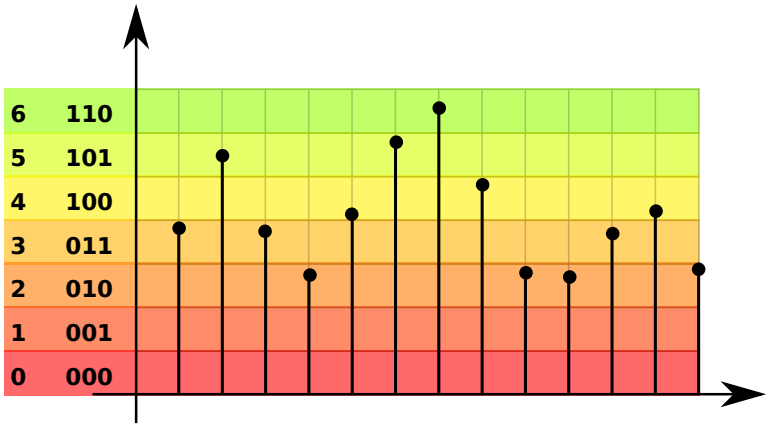
Kwantyzacja



Ile cyfr wystarczy do opisania 7 poziomów kwantyzacji?
a) w systemie dziesiętnym
b) w systemie binarnym

Kwantyzacja

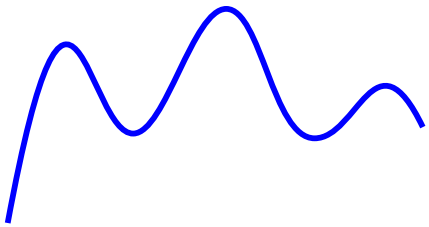


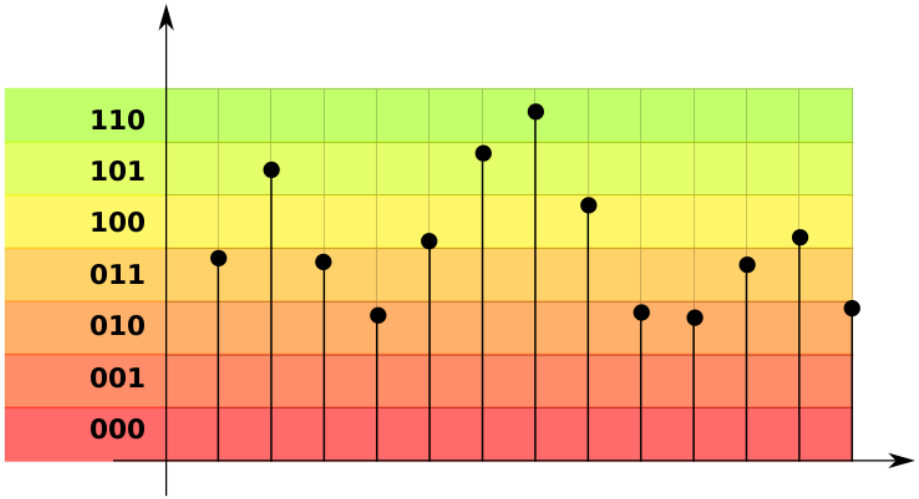


Sygnał cyfrowy

011 101 011 010 100 101 110 100 010 010 011 100 010

jest więc reprezentacją sygnału analogowego



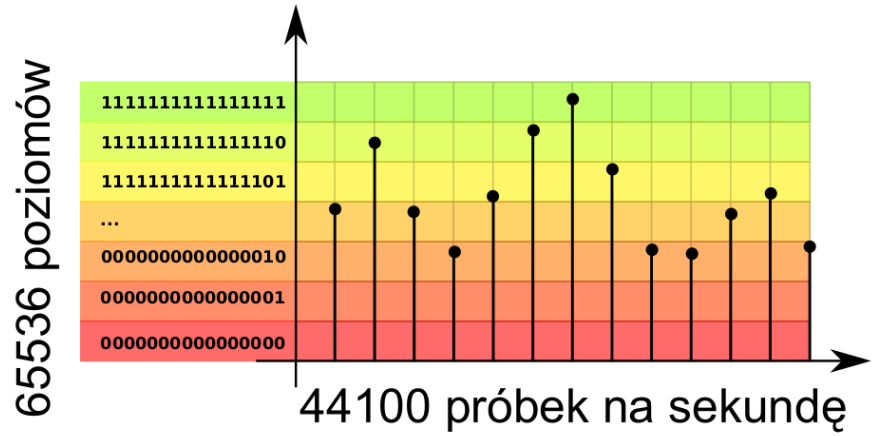


011 101 011 010 100 101 110 100 010 010 011 100 010

Jak poprawić rozdzielczość X?
Jak poprawić rozdzielczość Y?

Płyta Audio CD zawiera sygnał próbkowany z częstotliwością 44,1 kHz przy 16 bitach przeznaczonych na opis każdej próbki.

1 sekunda = 44100 próbek * 16 bitów * 2 kanały (stereo) = 1,4 Mbit



0101101010100010	1010101011100110	1010111110101
próbka 1	próbka 2	próbka 3

Kodowanie liczb ujemnych

$$6_{10} = 110$$

$$-6_{10} = ?$$

Kodowanie liczb ujemnych

$6_{10} = 110$

$-6_{10} = 1110$

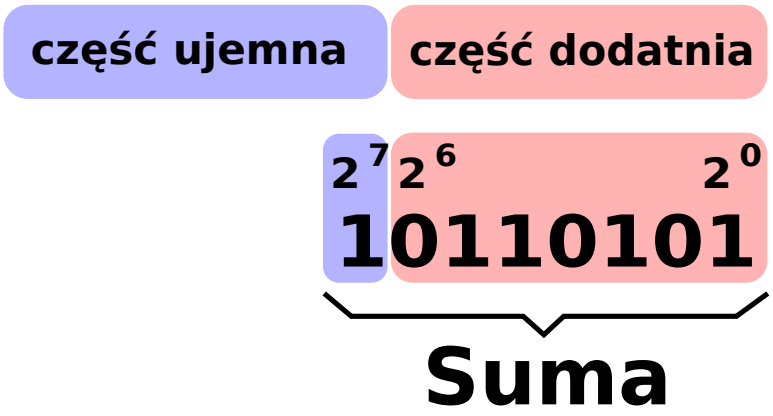
$6_{10} = 0110$

Kodowanie znak-moduł przypomina używany przez człowieka sposób kodowania liczb ujemnych

1 minus
0 plus

Przyjęcie takiego systemu zapisu liczb komplikuje operacje binarne dodawania i odejmowania. Wydzielenie i kontrolowanie znaku jest dodatkową operacją wykonywaną przez jednostkę arytmetyczno - logiczną (ALU).

KU2 Kod uzupełnieniowy do dwóch



Najmniejsza liczba

10000000=-128+0=-128

Największa liczba

01111111=-0+127=127

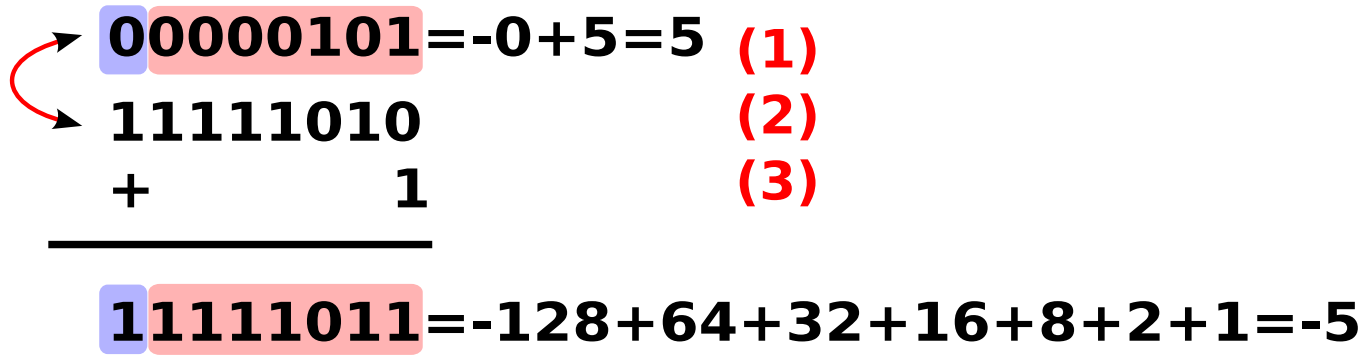
Jak zapisać 3?
Jak zapisać -1?

Jak zapisać -5?

KU2: Algorytm znajdujący liczbę przeciwną

Problem: wygenerować w KU2 liczbę -5 (przeciwna do +5)

1. Zapisać liczbę (+5);
2. Zamienić wszystkie 1/0 i 0/1;
3. Dodać 1.



Zalety KU2

- jedna reprezentacja zera (zamiast +/-0 w kodzie znak-moduł);
- nie ma potrzeby sprawdzania znaku liczby przy arytmetyce.

Dzięki powyższym zaletom i w efekcie prostszej implementacji, KU2 jest najpopularniejszą metodą kodowania liczb ujemnych.

Dodawanie dziesiętne / binarne

Dodawanie dziesiętne

Dodawane cyfry	$\sum$	Przenie-sienie
5 + 2	7	0
5 + 8	3	1

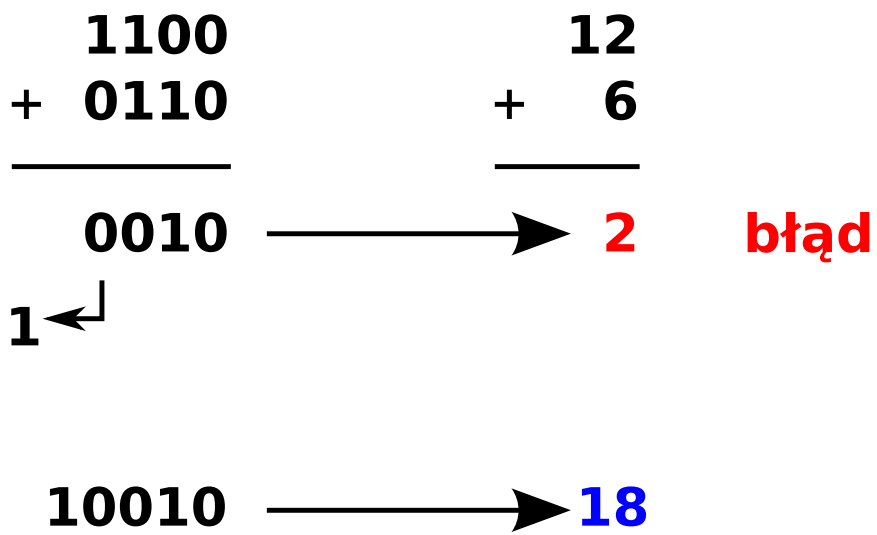
Dodawanie binarne

Dodawane cyfry	$\sum$	Przenie-sienie
0 + 0	0	0
0 + 1	1	0
1 + 0	1	0
1 + 1	0	1

```
    1        0111
+   1      + 0101
-------    -------
   10        1100
```

Nadmiar (overflow)

Układy realizujące operacje arytmetyczne muszą uwzględniać zakres argumentów dla tych operacji oraz zakres wyniku.



Mnożenie/dzielenie przez dwa

Przesuwanie cyfr w prawo powoduje podwajanie liczby. Przesuwanie jest dużo wydajniejsze od mnożenia / dzielenia i często jest stosowane przez programistów.

0011	3
0110	6
1100	12

W systemie	0074
dziesiętnym	0740
*10	7400

Dzielenie odbywa się na podobnej zasadzie - bity są przesuwane w prawo.

Mnożenie binarne

$$
\begin{array}{rr}
 & 1011 \\
\times & 101 \\
\hline
 & 1011 \\
 & 0000 \\
+ & 1011 \\
\hline
 & 110111
\end{array}
$$

Mnożenie odbywa się podobnie jak w systemie dziesiętnym z tą różnicą, że układy cyfrowe są przystosowane do sumowania tylko dwóch liczb (w przykładzie obok wystąpiły 3 liczby do zsumowania).

Problem rozwiązuje się w ten sposób, że każdy kolejny wynik mnożenia dodaje się na bieżąco do dotychczasowej sumy.

Sumowanie wymaga wcześniejszego wyrównania liczb.

Liczby zmiennopozycyjne

5 973 600 000 000 000 000 000 000 kg
(Masa Ziemii)

$5{,}9736 * 10^{24}$

5,9736 E+24

mantysa (precyzja) **cecha (wykładnik)**

Notacja naukowa pozwala na kodowanie bardzo dużych / małych liczb

Liczby zmiennopozycyjne

$5{,}625_{10}$

101,101

$0{,}101101 * 2^3$

Mantysa znormalizowana dla liczb binarnych należy do przedziału $<\frac{1}{2},1)$.

W praktyce oznacza to, że przecinek należy ustawić w taki sposób, aby liczba miała postać:

0,1xxxxxx...

Dzięki normalizacji zapis staje się jednoznaczny.

Liczby zmiennopozycyjne

znak	cecha / wykładnik	mantysa / precyzja
1	10010	1010110101

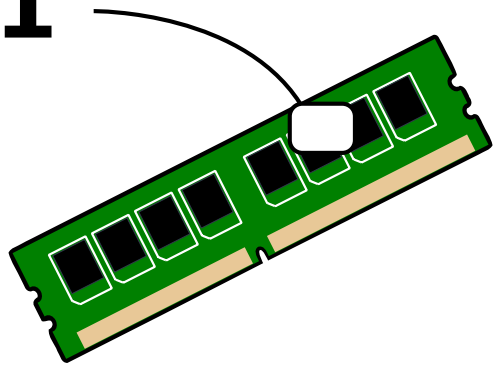
Standard IEEE 754

pojedyncza precyzja	1	8	23	(32 bity)
podwójna precyzja	1	11	52	(64 bity)

Liczby zmiennopozycyjne

Przykład

1000110110110101



Liczby zmiennopozycyjne

Przykład

1000110110110101

1000110110110101

- 0,110 110101 * 2 3 =

= 110,110101=6,828125 10