

Reprezentacja liczb w różnych systemach

Waga jest wartością przyporządkowaną pozycji cyfry w liczbie. W systemie dziesiętnym wagi są kolejnymi (od ostatniej cyfry) potęgami liczby 10. Ponieważ podstawą tego systemu jest liczba 10, posiada on 10 cyfr: 0,1,2,3,4,5,6,7,8,9. Najmłodsza (ostatnia) cyfra danej liczby to mnożnik liczby 10^0 , przedostatnia cyfra to mnożnik liczby 10^1 itd. Przykładowo, liczbę 859 w kodzie dziesiętnym można przedstawić następująco:

$$(859)_{10} = (8 \cdot 10^2 + 5 \cdot 10^1 + 9 \cdot 10^0)_{10}$$

Kod binarny (dwójkowy) [NKB, ang. NBC]

W kodzie binarnym podstawę stanowi liczba 2, a zatem wagi są kolejnymi potęgami liczby 2, a cały system wykorzystuje tylko 2 cyfry: 0 i 1. Poszczególne pozycje liczby dwójkowej (zapisanej w kodzie binarnym) nazywamy bitami. Bit określa także najmniejszą porcję informacji – może on przyjąć wartość 0 lub 1 (tak, nie; włączony, wyłączony). Osiem bitów stanowi bajt. Przykładowo liczbę 101011 zapisaną w kodzie binarnym można zapisać [dla ułatwienia obliczenia przedstawione od ostatniej cyfry]:

$$(101011)_2 = (1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0)_{10} = (32 + 8 + 2 + 1)_{10} = (43)_{10}$$

gdzie dla 1 bajtu (8 bitów):

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

Za pomocą słowa o długości k bitów zapisać można wartość dziesiętną A:

$$0 \leq A \leq 2^k - 1$$

Aby zapisać liczbę dziesiętną w postaci binarnej, zapisujemy kolejne reszty z dzielenia tej liczby przez 2, począwszy od ostatniej. Przykładowo, liczba $(79)_{10}$:

$$\begin{array}{l} 79 : 2 = 39 \text{ R } 1 \\ 39 : 2 = 19 \text{ R } 1 \\ 19 : 2 = 9 \text{ R } 1 \\ 9 : 2 = 4 \text{ R } 1 \\ 4 : 2 = 2 \text{ R } 0 \\ 2 : 2 = 1 \text{ R } 0 \\ 1 : 2 = 0 \text{ R } 1 \end{array}$$

A zatem:

$$(79)_{10} = (0100 \ 1111)_2$$

Kod heksadecymalny (szesnastkowy)

Wagi są tu kolejnymi potęgami liczby 16. Cyfr musi być więc aż 16, przyjmują one wartości od 0 do 15 i zapisane są one za pomocą następujących znaków: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Każdą cyfrę kodu szesnastkowego można zamienić na cztery cyfry kodu binarnego.

kod dziesiętny	kod binarny	kod heksadecymalny
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
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Kod heksadecymalny doskonale nadaje się do opisu układów logicznych, gdzie długość ciągu zero-jedynkowego (długość wektora informacji cyfrowej) wynosi jeden i więcej bajtów. Przeliczanie liczb dziesiętnych na szesnastkowe przebiega tak samo jak w przypadku liczb binarnych, za wyjątkiem dzielnika, którym jest liczba 16 (podstawa systemu). Przykładowo, liczba $(249)_{10}$:

$$\begin{aligned} 249 &: 16 = 15 \text{ R } 9 \\ 15 &: 16 = 0 \text{ R } 15 \end{aligned}$$

A zatem:

$$(249)_{10} = (\text{F9})_{16}$$

Największą zaletą kodu heksadecymalnego jest łatwa konwersja liczb z tego kodu na kod binarny i z kodu binarnego na kod heksadecymalny. Jeżeli liczba w kodzie binarnym zapisana jest na bajcie (8 bitach), to zapiszemy ją za pomocą dwóch cyfr kodu szesnastkowego (po cyfrze na każde 4 bity), np.:

$$(249)_{10} = (\text{F } 9)_{16} = (1111 \ 1001)_2$$

Kod ósemkowy

Wagi w kodzie ósemkowym są kolejnymi potęgami liczby 8. Przeliczenia pomiędzy zapisem ósemkowym a dziesiętnym przebiegają analogicznie do systemów wymienionych wcześniej.

Stałopozycyjna reprezentacja liczb

Do reprezentacji liczb całkowitych stosowane są kody:

- zapis znak-moduł
- zapis U1
- zapis U2

1. Zapis **znak-moduł** ZM (ang. sign-magnitude) utworzono przez dodanie jednego bitu do zapisu NKB. Bit ten, dodawany przed najbardziej znaczącą pozycją słowa, jest bitem znaku. Wartość tego bitu równa 0 oznacza liczbę dodatnią, a 1 liczbę ujemną. Dalsze bity w zapisie NKB oznaczają moduł tej liczby. Dla słów 16-bitowych można przedstawić liczby z zakresu $-2^{15} + 1 \leq A \leq +2^{15} - 1$, ponieważ największą liczbą zakodowaną na 15 bitach jest $2^{15} - 1$.

2. **U1** to tzw. zapis uzupełnień do 1 (ang. 1s complement). W zapisie tym najbardziej znaczący bit jest również bitem znaku (0 – liczba dodatnia, 1 – ujemna), ale w zależności od jego wartości dalsze bity zapisu mają różne znaczenie. Jeśli bit znaku jest 0 (liczba dodatnia) to dalsze bity są reprezentacją liczby dodatniej w kodzie NKB. Natomiast gdy bit znaku jest 1 (liczba ujemna), to dalsze bity reprezentują moduł liczby ujemnej w taki sposób, że zanegowane ich wartości odpowiadają modułowi tej liczby w kodzie NKB. Zakres liczb tego zapisu jest taki sam jak dla kodu ZM.

3. **U2** to tzw. zapis uzupełnień do 2 (ang. 2s complement). Zapis ten różni się od U1 tylko dla liczb ujemnych. Moduł liczby ujemnej w kodzie U2 jest obliczany w taki sposób, że do zanegowanych pozycji słowa jest arytmetycznie dodawana jedynka i dopiero tak utworzone słowo odpowiada w kodzie NKB modułowi tej liczby.