Przedmiot: Urządzenia techniki komputerowej
Nauczyciel: Mirosław Ruciński

Temat: **Systemy zapisu liczb.**

Cele kształcenia: Zapoznanie z systemami zapisu liczb: dziesiętny, dwójkowy, ósemkowy, szesnastkowy. Zdobycie umiejętności wykonywania działań na liczbach w różnych systemach.

Zagadnienia:

Systemy liczbowa – pozycyjne, addytywne (niepozycyjne).
Pozycyjne systemy liczbowe.
System dziesiętny (decymalny).
System dwójkowy (binarny).
System szesnastkowy (heksadecymalny).
System ósemkowy (oktalny).

Systemy liczbowe– pozycyjne, addytywne (niepozycyjne)

a) pozycyjne - rzeczywista wartość cyfry w zapisach liczbowych jest uzależniona od pozycji, jaką ta cyfra zajmuje w liczbie. Systemami pozycyjnymi są m.in. dziesiętny, dwójkowy, ósemkowy, szesnastkowy.

b) addytywne (niepozycyjne) - W addytywnych systemach liczbowych wartość przedstawionej liczby jest sumą wartości jej znaków cyfrowych. Przykładem systemu addytywnego jednocześnie najbardziej znanym jest rzymski system liczbowy, w którym to poszczególne znaki mają swoją stałą wartość. Cyfry systemu rzymskiego to: I (1), V (5), X (10), L (50),C (100),D (500),M (1000).
W celu obliczenia wartości liczby w dowolnym systemie pozycyjnym należy pomnożyć poszczególne cyfry liczby przez potęgi podstawy systemu.

Ogólnie wartość reprezentowaną przez symbol liczby zapisujemy następująco:

$$\mathbf{c_n*p^n+c_{n-1}*p^{n-1}+...+ c_2*p^2+ c_1*p^1+ c_0*p^0}$$

gdzie: $\mathbf{c_n, c_{n-1},..., c_0}$ - cyfra systemu pozycyjnego **p** - podstawa systemu n - liczba cyfr w ciągu.

System dziesiętny (decymalny)

- 10 cyfr: 0,1,2,...,9
- p = 10
- zapis liczby: pozycja setek, pozycja dziesiątek, pozycja jedynek

$$543 = 5*100+4*10+3*1$$

$$\mathbf{5_2 4_1 3_{0D} = 5*10^2+4*10^1+3*10^0}$$

System dwójkowy (binarny)

- 2 cyfry: 0,1
- p = 2
- zapis liczby:

$$10101_{B} = 1*2^4+0*2^3+1*2^2+0*2^1+1*2^0=$$

$$1*16+0*8+1*4+0*2+1*1 = 21_D$$

Aby dokonać zamiany liczby dziesiętnej na postać binarną, należy wykonać cykliczne dzielenie z resztą.

21:2=10	r=1
10:2=5	r=0
5:2=2	r=1
2:2=1	r=0
1:2=0	r=1

21_D=10101_B

System szesnastkowy (heksadecymalny)

- 16 cyfr: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- p = 16
- zapis liczby:

$$11_{16} = 1 * 16^1 + 1 * 16^0 = 17_{10}$$

$$C9_{16} = 12 * 16^1 + 9 * 16^0 = 201_{10}$$

$$4F_{16} = 4 * 16^1 + 15 * 16^0 = 79_{10}$$

Aby dokonać zamiany liczby dziesiętnej na postać heksadecymalną, należy wykonać cykliczne dzielenie z resztą.

1221:16=76	r=5	↑
76:16=4	r=12(C)	
4:16=0	r=4	

$1221_D = 4C5_H$

System ósemkowy (oktalny)

- 8 cyfr: 0,1,2,3,4,5,6,7
- p = 8
- zapis liczby:

$$112_8=1*8^2+1*8^1+2*8^0 =74_{10}$$

Aby dokonać zamiany liczby dziesiętnej na postać oktalną, należy wykonać cykliczne dzielenie z resztą.

74:8=9	r=2	↑
9:8=1	r=1	
1:0=0	r=1	

$74_D = 112_O$

Temat: <u>Podstawowe operacje arytmetyczne na liczbach binarnych.</u>

Cele kształcenia: Zapoznanie z podstawowymi działaniami arytmetycznymi w systemie binarnym. Zdobycie umiejętności wykonywania działań na liczbach w różnych systemach liczbowych.

Zagadnienia:

Liczby binarne umożliwiają wykonywanie operacji arytmetycznych (ang. Arithmetic operations on binary numbers). Arytmetyką liczb binarnych rządzą pewne zasady, tzw tabliczki: dodawania, odejmowania, mnożenia i dzielenia.

Dodawanie liczb binarnych
Odejmowanie liczb binarnych.
Mnożenie liczb binarnych.
Dzielenie liczb binarnych.
Zamiany różnych systemów liczbowych.

Dodawanie liczb binarnych - opiera się na prostej tabliczce dodawania, w której reprezentowane są cztery sumy cząstkowe:

0+0=0
0+1=1
1+0=1
1+1=0 i 1 dalej

Przykład dodawania liczb binarnych: 1101_B i 1011_B

	0	0	1	
1+				
	1+1	**1**+1	**0**+1	**1**
+	**1**	**0**	**1**	**1**
1	**1**	**0**	**0**	**0**

Wynik: **11000_B**

Odejmowanie liczb binarnych – tabliczka odejmowania

0-0=0
1-0=1
1-1=0
0-1=1 i pożyczka

Przykład odejmowania liczb binarnych: 1101 i 1011

		0		
	1	**1**-1	**0**	**1**
-	**1**	**0**	**1**	**1**
	0	**0**	**1**	**0**

Wynik: **0010_B**

Mnożenie liczb binarnych – tabliczka mnożenia.

0*0=0
1*0=0
0*1=0
1*1=1

Przykład mnożenia liczb binarnych: 1010 i 1101

				1	**0**	**1**	**0**
			*	**1**	**1**	**0**	**1**
				1	0	1	0
			0	0	0	0	
		1+1	0+1	1	0		
	1+1	0	1	0			
+							
1	**0**	**0**	**0**	**0**	**0**	**1**	**0**

Wynik: **$\mathbf{10000010_B}$**

Dzielenie liczb binarnych - jedną z metod wykonywania ilorazu liczb binarnych jest cykliczne odejmowanie odpowiednio przesuwanego dzielnika do dzielnej.

		1	1	0			
	1	**1**	**0**	**1**	**:**	**1**	**0**
-	1	0	↓	↓			
	0	1	0				
-		1	0				
	0	0	0	1			
			1	0			
reszta	0	0	0	1			

13_D:2_D=6 i reszty 1_D

1. Dzielenie zaczyna się od podstawienia dzielnika pod dzielną, począwszy od jej najstarszej cyfry (lewa strona). Następnie sprawdza się, czy dzielnik można odjąć od fragmentu dzielnej. Jeżeli tak, to w wyniku wprowadza się jedynkę – w kolumnie nad najmłodszą cyfrą dzielnika (prawa strona).
2. Następnie odejmuje się cyfry i uzupełnia powstałą liczbę znakami przepisanymi z oryginalnej dzielnej (czarna strzałki). Cyfry przepisujemy przy każdorazowym odejmowaniu dzielnika od dzielnej, aż do uzyskania wyniku.
3. Jeżeli dzielnik nie da się odjąć do fragmentu dzielnej, w wynik wprowadza się zero, a dzielną przepisuje się bez zmian. Cały proces powtarza się aż do momentu uzyskania ostatniej cyfry w wyniku.
4. Jeżeli ostatnie odejmowanie nie może być wykonane lub z ostatniej różnicy nie wychodzą zera, przepisana dzielna lub powstała z różnicy liczba stanowi resztę z dzielenia.

Zamiany różnych systemów liczbowych

Konwersja dwójkowo - ósemkowa

Do konwersji dwójkowo ósemkowej pomocna jest tabela, w której wartości cyfr ósemkowych wyrażone są w kodzie binarnym.

0	1	2	3	4	5	6	7
000	001	010	011	100	101	110	111

Zasada konwersji dwójkowo ósemkowej jest następująca. Liczbę binarną rozdzielamy na grupy 3 bitowe idąc od strony prawej do lewej. Jeśli w ostatniej grupie jest mniej bitów, to brakujące bity uzupełniamy zerami. Teraz każdą zl trzy-bitowych grup zastępujemy cyfrą ósemkową zgodnie z tabelką konwersji. W wyniku otrzymujemy liczbę ósemkową o identycznej wartości jak wyjściowa liczba binarna. Konwersja w drugą stronę jest analogiczna. Każdą cyfrę ósemkową zastępujemy grupą 3 bitów według tabelki konwersji. Grupy łączymy w jedną liczbę binarną.

$$101001111_{(2)} = (101)\ (001)\ (111)$$

Konwersja dwójkowo - szesnastkowa

Do konwersji dwójkowo szesnastkowej pomocna jest tabela, w której wartości cyfr szesnastkowych wyrażone są w kodzie binarnym.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Liczbę dwójkową dzielimy na grupy cztero-bitowe idąc od strony prawej do lewej. Jeśli w ostatniej grupie jest mniej bitów, to brakujące bity wypełniamy zerami. Następnie każdą grupę bitów zastępujemy jedną cyfrą szesnastkową zgodnie z tabelką konwersji. Konwersja w drugą stronę jest analogiczna. Każdą cyfrę szesnastkową zastępujemy grupą 4 bitów według tabelki konwersji. Grupy łączymy w jedną liczbę binarną.

$$100110101001111_{(2)}= (0100)\ (1101)\ (0100)\ (1111)$$

W celu szybkiego przekształcania liczb binarnych na postać dziesiętną dobrze jest zapamiętać krotności poszczególnych wag system binarnego.

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1

$$101001111_{(2)} = (101)\ (001)\ (111) = 517_8$$

$$100110101001111_{(2)} = (0100)\ (1101)\ (0100)\ (1111) = 4D4F$$

Temat: <u>Zapis liczby binarnej ze znakiem.</u>

Cele kształcenia: Zapoznanie z zasadami zapisywania liczb ze znakiem w systemie dwójkowym. Wykonywanie działań na liczbach ze znakiem w systemie dwójkowym.

Zagadnienia: W systemie dziesiętnym liczby ujemne opatrzone są specjalnym znakiem graficznym (-) do zapisu liczb binarnych ze znakiem opracowano kilka metod**.**

Metoda znak-moduł (ZM)
Metoda uzupełnień do 1 (U1)
Metoda uzupełnień do 2 (U2)

Metoda znak-moduł (ZM)
W metodzie znak-moduł zastosowano prosty zabieg kodowania znaku za pomocą najstarszej cyfry w liczbie binarnej. Najstarszą cyfrę określa się jako znak, pozostałe cyfry są modułem reprezentującym daną liczbę binarną:

Znak	Moduł
a_{n-1}	$a_{n-2} \ldots a_1\ a_0$

W celu obliczenia wartości naturalnej liczby binarnej ze znakiem należy posłużyć się następującym wzorem:

$$a_{n-1}\ a_{n-2} \ldots a_2\ a_1\ a_0 = (1-2*a_{n-1})* \sum_{i=0}^{n-2} ai2^i$$

Przykład: $\mathbf{0111_{(Z-M)} = 7_D \quad 1111_{(Z-M)} = -7_D}$

$$\mathbf{0111_{(Z-M)} = 0\ 1_2\ 1_1\ 1_0 = (1-2*0)*(1*2^2+1*2^1+1*2^0) = 1*(4+2+1) = 7_D}$$
$$\mathbf{1111_{(Z-M)} = 1\ \ 1_2\ \ 1_1\ 1_0 = (1-2*1)*(1*2^2+1*2^1+1*2^0) = -1*(4+2+1) = -7_D}$$

Metoda uzupełnień do 1 (U1)

Zapis liczb ujemnych – system uzupełnień do jedynki (U1)

Liczby ujemne mają zamienione wszystkie bity na przeciwne.

00001100b = +12
11110011b = -12

Najbardziej znaczący bit (pierwszy z lewej) oznacza znak liczby: 0 –liczba dodatnia, 1 – liczba ujemna

Uwaga 1: Liczba zero może mieć znak

00000000b = +0
11111111b = -0

Uwaga 2: Zakres liczb ulega zmianie z 0..255 na -127...-0,+0,...+127

Metoda uzupełnień do 2 (U2)

Zapis liczb ujemnych – system uzupełnień do dwóch (U2)

Liczbę ujemną zapisuje się w systemie U2 zapisując jej wartość bezwzględną w postaci binarnej, po czym zamieniając wszystkie bity na przeciwne (U1) oraz dodając liczbę „1”

przykład: zapis liczby –12 w systemie U2:

- zapis binarny wartości bezwzględnej liczby (bez znaku)
- zamiana wszystkich bitów na przeciwne (U1)
- dodanie liczby 1 (00000001b)

1 +12 (dec) = 00001100b
2 11110011b
3 +00000001b
==========
11110100b = -12

Liczba ujemna -12

1. -12(Dec)= 11110100b
2. 00001011b
3. +00000001b
==========
00001100b = +12

Sprawdzenie czy +12 + (-12) = 0 ?

```
+12 (dec) = 00001100b
-12 (dec) = 11110100b
    +       ============
            1 00000000b = 0
```

Temat: <u>Reprezentacja stało i zmiennopozycyjna.</u>

Cele kształcenia: Poznanie reprezentacje stało i zmiennopozycyjnych. Charakteryzowanie reprezentacji stało i zmiennopozycyjnych.

Zagadnienia: Podobnie jak w systemie dziesiętnym liczby binarne mogą być zapisane w postaci ułamkowej. Zapis liczb z przecinkiem może przyjąć postać stało lub zmiennoprzecinkową.

Liczby stałoprzecinkowe (stałopozycyjne).
Liczby zmiennoprzecinkowe (zmiennopozycyjne).

Liczby stałoprzecinkowe (stałopozycyjne) – pozycja przecinka ustalana jest w zależności od wymaganej dokładności. Binarną liczbę stałoprzecinkową można potraktować, jako złożenie dwóch części liczby całkowitej oraz ułamkowej rozdzielonych przecinkiem.

Część całkowita	Część ułamkowa
10110011,	0101

Zapis 8.8 oznacza 8 bitów części całkowitej (w tym bit znaku) i 8 bitów części ułamkowej

Nr bitu	7	6	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8
Bit	+1	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
znaku wartość	-1	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256

Tabela. Wartości (wagi) bitów w zapisie binarnym liczb rzeczywistych stałoprzecinkowych (słowo 16 bitowe, zapis 8.8)
- zakres: -128.00000000 ... +127.99609375

Przykład 1 (zapis 8.8)
Nr 7654 321 0 1234 5678
Liczba 00010010 10100001 b reprezentuje wartość

$$\mathbf{1*2^{4} + 1*2^{1} + 1*2^{-1} + 1*2^{-3} + 1*2^{-8} = 16 + 2 + 1/2 + 1/8 + 1/256 = 18.62890625}$$

Liczby zmiennoprzecinkowe (zmiennopozycyjne) – umożliwiają obsługę większego zakresu liczb, jednak kosztem wolniejszego przetwarzania i mniejszej dokładności. Termin „zmiennoprzecinkowe” oznacza, że **nie istnieje stała liczba cyfr przed przecinkiem i po przecinku.**
Ze względu na wygodę operowania na takich liczbach, ***przyjmuje się ograniczony zakres na mantysę i cechę.*** Powoduje to, że reprezentacja liczby rzeczywistej jest tylko przybliżona, a jedna liczba zmiennoprzecinkowa może reprezentować różne liczby rzeczywiste z pewnego zakresu.

Podstawa matematyczna

Wartość liczby zmiennoprzecinkowej jest obliczana według wzoru:

$$x = S \cdot M \cdot B^{E}$$

gdzie:
S (ang. sign) – znak liczby, 1 lub -1
M (ang. mantissa) – znormalizowana ***mantysa***, liczba ułamkowa.
B (ang. base) – podstawa systemu liczbowego (2 dla systemów komputerowych binarnych).
E (ang. exponent) – wykładnik, ***cecha***, liczba całkowita.
Mantysa jest znormalizowana, tj. należy do przedziału $[1, B)$(przedział prawostronnie otwarty!). Jeżeli M jest stałe, a E zmienia się, wówczas przesunięciu ulega przecinek – stąd właśnie pochodzi nazwa tej reprezentacji.
Zarówno dla mantysy jak i wykładnika liczba cyfr jest z góry ustalona. Zatem dana liczba jest reprezentowana z pewną skończoną dokładnością i należy do skończonego zbioru wartości.

Przykład reprezentacji
Przyjmijmy, że ***B=10***, liczba cyfr dziesiętnych przeznaczonych na mantysę wynosi 4, natomiast na wykładnik 2. Chcemy zapisać wartość 60,89523.
Liczba 60,89523 odpowiada M=60,89523, E=0.

Normalizacja mantysy.
Mantysa nie należy do przedziału [1,10), należy przesuwać przecinek w lewo aż będzie do niej należała. Przesuwanie przecinka w lewo wiąże się ze zwiększaniem wykładnika (cechy)?
M=6,089523, E=1.

Odcięcie i zaokrąglenie mantysy.
po odcięciu: 6,089,
po zaokrągleniu: 6,090. **Wynik:** $6{,}090 \bullet 10^{1}$ = 6,090 E 1

Przykład dla liczby mniejszej od 1: 0,0000125.
M= 0,0000125, E = 0.

Po normalizacji

M = 1,25, E = -5.
Liczba cyfr znaczących jest mniejsza od dostępnej, więc nie jest potrzebne zaokrąglanie.

Wynik: $1{,}250 \bullet 10^{-5}$ = 1,25 E -5

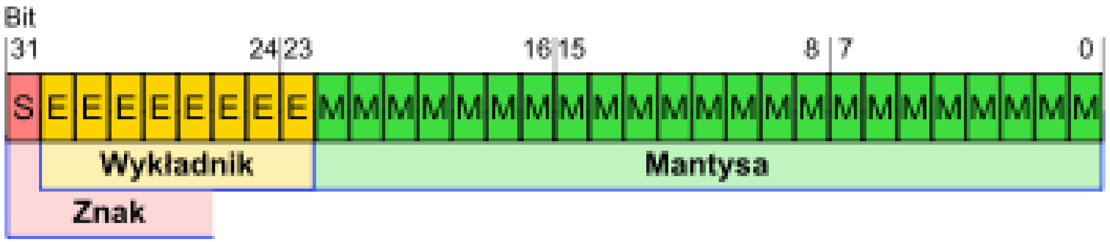
<u>Podstawa systemu w reprezentacji liczby w systemie dwójkowym wynosi 2, stąd liczba zmiennopozycyjne ma wartość:</u>

$$x = S * M * 2^E$$

W reprezentacji konkretnej liczby na mantysę i wykładnik (w tym i znak wykładnika) są przeznaczone odpowiednie liczby bitów. ***Im dłuższa mantysa tym większa dokładność . Im dłuższy wykładnik, tym większy przedział liczb, które można reprezentować.***

Liczbę można zapisać w postaci zmiennopozycyjnej w różny sposób. Ustala się znormalizowaną postać mantysy, należy ona do przedziału [1, B), czyli spełnia nierówność B > I M I > 1. Położenie przecinka w mantysie nie jest ustalone i może się dowolnie zmieniać. Jeżeli zmieni się wykładnik, przecinek przesuwa się.

Implementacje sprzętowe



Reprezentacja zmiennoprzecinkowa IEEE-754 single

W implementacjach sprzętowych liczby zmiennoprzecinkowe wyraża się liczbami dwójkowymi (B=2). Ma to następujące zalety:
Mantysa należy do przedziału [1,2), jest więc postaci 1.xxxxx.... (x – bit o dowolnej wartości). Ponieważ część całkowita jest znana, i równa zawsze 1, nie jest zapamiętywana, co daje dodatkowy bit na część ułamkową.

Ponieważ znak liczby jest zapamiętywany na jednym bicie, otrzymanie modułu i wartości przeciwnej wymaga, odpowiednio, wyzerowania tego bitu (logiczna operacja AND), lub zmiany na wartość przeciwną (logiczna operacja XOR).

W celu ujednolicenia zasad operacji na liczbach zmiennoprzecinkowych na różnych platformach sprzętowych, opracowano standard IEEE 754, w oparciu o który realizuje się obecnie wszystkie implementacje sprzętowe liczb zmiennoprzecinkowych.

Definiuje on dwie klasy liczb:
pojedynczej precyzji (ang. single) - 32 bity (1bit znak, 8 bitów wykładnik, 23 bity mantysa).
podwójnej precyzji (ang. double) – 64 bity (1bit znak, 11 bitów wykładnik, 52 bity mantysa).

Literatura:

Urządzenia techniki komputerowej – Tomasz Kowalski

Wikipedia- wolna encyklopedia internetowa

Strona internetowa:

http://www.math.edu.pl/systemy-liczbowe

Opracował Mirosław Ruciński
e-mail: nauczyciel.zsen@gmail.com