



# Architektura komputerów

## Wykład 4

Jan Kazimirski



# Reprezentacja danych



## Plan wykładu

- Systemy liczbowe
- Zapis dwójkowy liczb całkowitych
- Działania arytmetyczne
- Liczby rzeczywiste
- Znaki i łańcuchy znaków



# Zapis dziesiętny – liczby całkowite

- Symbole: 0,1,2,3,4,5,6,7,8,9
- Zapis pozycyjny – pozycja określa wagę
- Wagi – kolejne potęgi liczby 10.
- Przykład:

$$123 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$$



# Zapis dziesiętny - ułamki

- Rozwinięcie koncepcji dla liczb całkowitych
- Po przecinku kolejne pozycje – wagi liczby 10 z ujemnymi wykładnikami, tzn  $10^{-1}$ ,  $10^{-2}$  itd.
- Przykład:
$$0.25 = 2 \cdot 10^{-1} + 5 \cdot 10^{-2}$$
$$= 2 \cdot \frac{1}{10} + 5 \cdot \frac{1}{100}$$
- UWAGA! Nie wszystkie ułamki można przedstawić dokładnie (np.  $\frac{1}{3}$ )



# System dwójkowy

- Analogiczny do systemu dziesiętnego, ale podstawą są potęgi 2.
- Dwa symbole – 0,1
- Kolejne pozycje:  $2^0=1$ ,  $2^1=2$ ,  $2^2=4$ , itd.
- Przykład:

$$1011B = 1*8+0*4+1*2+1*1 = 11$$



## Konwersja dec $\leftrightarrow$ bin

- Z systemu dziesiętnego na binarny – metoda kolejnych podziałów na 2
- Z systemu binarnego na dziesiętny – wartości 1 należy przemnożyć przez odpowiadające im wagi i wyniki zsumować



## System ósemkowy

- System dwójkowy jest niewygodny – długie ciągi znaków
- System ósemkowy (oktalny) – podstawa 8, symbole: 0,1,2,3,4,5,6,7
- Często stosowany w informatyce – lepiej „pasuje” do zapisu dwójkowego.
- Łatwa konwersja  $\text{bin} \leftrightarrow \text{oct}$  – 3 bity = jeden symbol oktalny





# System szesnastkowy (heksadecymalny)

- Podstawa – 16, symbole: 0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F
- Często stosowany w informatyce i programowaniu
- Krótki zapis (liczba 8-bitowa – 2 znaki hex)
- Łatwa konwersja z systemu dwójkowego – 4-bity – jeden symbol hex



## Zapis liczb ze znakiem

- Najbardziej typowe:
  - Znak-moduł – jeden z bitów określa znak liczby.
  - Uzupelnienie do 2 (U2) – najbardziej znaczący bit ma ujemną wagą
- Liczby typu int w komputerach implementowane są w zapisie U2 ze względu na łatwą arytmetykę



# Liczby ze znakiem – uzupełnienie do dwóch

- Algorytm zamiany liczby na przeciwną:
  - Negujemy wszystkie bity
  - Dodajemy 1
- Nie występuje problem dwóch zer (+0 i -0), obecny przy zapisie znak-moduł.
- Naturalne operacje dodawania i odejmowania



## Zapis BCD

- Używany do zapisu liczb dziesiętnych całkowitych
- Cyfra dziesiętna kodowana jest na 4-bitach
- Kod „rozrzutny”, dosyć wolne obliczenia
- Stosowany do zapisywania wielocyfrowych liczb całkowitych
- Stosowany czasami w systemach wbudowanych.



# Własności liczb dwójkowych

- Liczby parzyste mają najmniej istotny bit = 0, liczby nieparzyste – 1
- Jeżeli N najmniej istotnych bitów = 0, to liczba dzieli się przez  $2^N$  bez reszty.
- Jeżeli tylko bit N-ty = 1 to wartość liczby =  $2^N$
- Przesunięcie wszystkich bitów w lewo odpowiada mnożeniu przez 2



# Własności liczb dwójkowych

- Dla liczby bez znaku przesunięcie wszystkich bitów w prawo odpowiada dzieleniu przez dwa
- Wynik dodawania liczb N-bitowych można zapisać na N+1 bitach
- Wynik mnożenia liczb N-bitowych może wymagać  $2*N$  bitów
- Jeżeli N-bitowa liczba składa się z samych jedynek to jej wartość =  $2^N - 1$ .



# Własności liczb dwójkowych

- Zwiększenie o 1 największej (w danym formacie) liczby bez znaku da 0 – **przepełnienie**
- Zmniejszenie o 1 liczby 0 zapisanej jako liczba bez znaku da największą liczbę dającą się zapisać w danym formacie - **zawinięcie**



# Dodawanie i odejmowanie liczb dwójkowych

- Dodawanie – bit po bicie według poniższych reguł:
  - $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=0$  z przeniesieniem (P)
  - $0+0+P=1$ ,  $0+1+P=1+0+P=0$  z P
  - $1+1+P=1$  z P
- Odejmowanie – analogicznie, ale z „pożyczką”
- W przypadku liczb U2 można dodać liczbę przeciwną





# Mnożenie i dzielenie liczb dwójkowych

- Można realizować jako wielokrotne dodawanie
- Sposób „naturalny” - tak jak mnożenie „ręczne” liczb dziesiętnych – reguły:  $0*0=0$ ,  $0*1=1*0=0$ ,  $1*1=1$
- Zmodyfikowane algorytmy mnożenia – np. algorytm Bootha
- Dzielenie - metoda naturalna – sprawdzanie ile razy dzielnik mieści się w dzielnej.



## Operacje logiczne

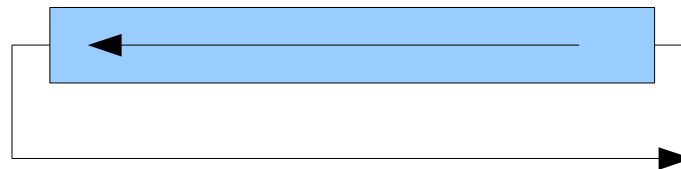
- Operacje na bitach: AND, OR, XOR, NOT
- Operacje na ciągach bitów – wykonujemy operację na odpowiednich bitach operandów
- Zastosowanie operacji bitowych
  - Testowanie bitu
  - Maskowanie bitów
  - Szyfrowanie (XOR)

# Przesunięcia i rotacje

- Przesunięcie w lewo (prawo) – odpowiada mnożeniu (dzieleniu) przez dwa.
- Przesunięcia i rotacje ułatwiają operacje na bitach



Przesunięcie



Rotacja



## Dwójkowy zapis ułamków

- Zapis naturalny – ujemne potęgi liczby dwa
- Zapis naturalny jest niewygodny przy operowaniu małymi i dużymi liczbami jednocześnie – wąski zakres (liczba bitów)
- Komputery do obliczeń stosują zapis **zmiennoprzecinkowy (zmiennopozycyjny)**



## Zapis zmiennopozycyjny

- Liczba składa się z:
  - Bitu znaku
  - Bity mantysy
  - Bity wykładnika
- Analogia do zapisu dziesiętnego w notacji naukowej:  
**-1.343567e-25**



# Zapis zmiennopozycyjny c.d.

- Zapis zmiennopozycyjny pozwala zapisać zarówno bardzo małe, jak i bardzo duże liczby na tej samej liczbie bitów.
- Przykład:  
$$1.0000e-100$$
$$1.0000e+100$$
- UWAGA na błędy (utrata dokładności)!



# Formaty IEEE

- Format zmiennopozycyjny pojedynczej precyzji (float) – 32 bity  
*bit znaku, 8 bitów wykładnik, 23(24) bity mantysa*
- Format zmiennopozycyjny podwójnej precyzji (double) – 64 bity  
*bit znaku, 11 bitów wykładnik, 52(53) bity mantysa*
- Format zmiennopozycyjny zwiększonej precyzji – 80 bitów  
*bit znaku, 15 bitów wykładnik, 64 bity mantysa*



# Parametry formatów IEEE

Parametr	float	double
Długość słowa	32	64
Długość wykładnika	8	11
Długość mantysy	23	52
Najmniejsza wartość	1,00E-038	1,00E-308
Największa wartość	1,00E+038	1,00E+308





## Norma IEEE 754

- Definicje formatów liczb
- Arytmetyka nieskończoności
- Symbol NaN (Not a Number) do określenia sytuacji gdy wynik jest nieznan
- Definicje operacji arytmetycznych
- Definicje wyjątków



## IEEE 754 - operacje

- Podstawowe operacje arytmetyczne
- Skalowanie i kwantyzacja
- Manipulowanie znakiem (np. abs)
- Operacje porównania
- Testowanie i klasyfikacja (np. NaN)
- Testowanie i ustawianie flag
- Inne ...



## IEEE 754 - wyjątki

- Nieprawidłowa operacja  
Np. Pierwiastek z liczby ujemnej.
- Dzielenie przez zero
- Nadmiar (overflow)  
Rezultat zbyt duży by go poprawnie zapisać
- Niedomiar (underflow)  
Rezultat zbyt mały by go poprawnie zapisać
- Wynik niedokładny  
Obliczenia prowadzą do utraty dokładności.



## Przykładowe działania

- Dodawanie i odejmowanie
  - Sprawdzenie znaków i ew. zmiana znaku odjemnika
  - Sprawdzenie, czy jeden z argumentów nie jest zerem
  - Wyrównanie wykładników
  - Sumowanie mantys z uwzględnieniem znaku
  - Normalizacja wyniku
- Mnożenie i dzielenie
  - Sprawdzenie czy jeden z argumentów nie jest zerem
  - Dodawanie (lub odejmowanie) wykładników
  - Działanie na mantysach z uwzględnieniem znaku



# Koszt obliczeniowy działań

Operacja: CPU:	+,- (int)	+,- (float)	* (int)	* (float)	/ (int)	/ (float)
Intel 386/387	2	23-34	9-38	27-35	43	89
Intel i486	1	10	12-42	11	43	35 (62)
Pentium	1	3	10	3	46	19 (33)
PowerPC	1	3	4	3	20	18 (31)
MIPS R4x00	1	4	10	7 (8)	69	23 (36)

Źródło: GAME DEVELOPER FEBRUARY/MARCH 1996, str 19-24



## Kodowanie znaków

- Komputer operuje danymi liczbowymi – inne symbole muszą być **kodowane**
- Kodowania znaków:
  - ASCII
  - Rozszerzone kody ASCII
  - Zestaw znaków EBCDIC (IBM)
  - Zestawy dwubajtowe
  - Unicode



## Kody ASCII

- Standard wymiany danych między komputerami
- Zestaw 128 znaków kodowanych na 7 bitach
- Cztery grupy po 32 znaki
  - Znaki kontrolne (np. sterowanie drukarką)
  - Znaki przestankowe, symbole i cyfry
  - Wielkie litery (+symbole specjalne)
  - Małe litery (+symbole specjalne i znak delete)



## Unicode

- Rozszerzony standard kodowania znaków
- Znosi ograniczenia stosowane przez dotychczasowe kodowania (strony kodowe)
- Definiuje przestrzeń numeracyjną dla poszczególnych grup znaków
- **Nie definiuje samego kodowania!**
- Metody kodowania np. UTF-8, UTF-16, UTF-32 i inne





## Łańcuchy znakowe

- Implementowane zwykle jako tablice znaków
- Dwa sposoby oznaczania długości łańcucha znaków
  - Znak kończący (null-terminated string – C/C++)  
*Nieograniczona długość, wolne operacje, ryzyko przepełnienia*
  - Łańcuch poprzedzony długością (Pascal)  
*Ograniczona długość, szybkie operacje.*



## Złożone struktury danych

- Rekordy, struktury
- Rekordy z wariantami (unie)
- Tablice
- Listy (jedno, dwukierunkowe)
- Mapy
- Drzewa



# Big Endian vs. Little Endian

- Liczby wielobajtowe mogą być zapisywane na dwa sposoby:
  - Big endian – najbardziej znaczący bajt jest na początku (IBM 370, SPARC)
  - Little endian – najbardziej znaczący bajt jest na końcu (x86)
- Format zapisu utrudnia przenoszenie danych binarnych między platformami oraz przekazywanie ich w internecie (big endian)



## Podsumowanie

- Komputery stosują binarny system liczenia
- Inne typy danych wymagają konwersji na system dwójkowy.
- Liczby rzeczywiste zapisywane są w formacie zmiennoprzecinkowym
- Znaki i łańcuchy znakowe zapisywane są w postaci odpowiednich kodów
- W programach stosuje się również bardziej zaawansowane struktury danych