Otázka 11 - Y36SAP

Zadání

Zobrazení dat a operace. Číselné soustavy. Sčítání, odčítání, posuvy, násobení a dělení ve dvojkové soustavě a zapojení příslušných obvodů. Zobrazení čísel se znaménkem a operace s nimi. Pohyblivá řádová čárka. Alfanumerické kódy. (Y36SAP)

Slovníček pojmů

- **poziční číselná soustava** - hodnota čísla je určena pozicí znaků - vetšina číselných soustav
- **nepoziční číslená soustava** - hodnota není určena pozicí - např. římské číslice
- **základ číselné soustavy** - u pozičních soustav, označuje počet hodnot které můžou být na jedné pozici, u dvojkové soustavy (základ *z=2*) znamená že na jedné pozici můžou být pouze 2 hodnoty: 0 a 1
- **číselný kód** - transformace celého čísla na nezáporná čísla (způsob reprezentace celých čísel pomocí nezáporných čísel)
- **příbuzné soustavy** - soustavy se základy z_1 a z_2 jsou příbuzné pokud existuje celé číslo k takové, že platí $z_1 = z_2^k$ (např. dvojková, osmičková a šestnáctková soustava)

Zobrazení dat a operace

Zobrazení číselných dat v paměti

- v pevné řádové čárce - fixed point, celá čísla (int, byte)
- v pohyblivé řádové čárce - floating point, racionální čísla (real, float, double)
- bez znaménka (unsigned), se znaménkem (signed)
- různá délka nebo rozsah hodnot (short int, int, long int, byte)

(*zdroj*: poslední slide z první přednášky z roku 2007)

Operace

- aritmetické operace - sčítání, odčítání, násobení, dělení
- bitové operace - posuv (aritmetický, logický, rotace)
- operace v řádové mřížce - zaokruhlení (nahoru, dolů, preference sudé číslice, preference většího čísla)

Číselné soustavy

Číselná soustava je soustava znaků a pravidel pro zobrazení čísel.

- **Polyadické** (poziční) - určené svým základem - bází z (např. desítková soustava,

soustava pro vyjádření času)

- **Nepolyadické (nepoziční) - smíšený základ, nepoužívají se ve výpočetní technice (např. římské číslice)**

Zápis čísla o základu z

$$A_z = \left(a_n \, a_{n-1} \cdots a_1 \, a_0 \, , \, a_{-1} \, a_{-2} \cdots a_{-m} \right)_z$$

- a_n **- symbol na nejvyšší pozici (nejvyšší řád)**
- a_{-m} **- symbol na nejnižší pozici (nejnižší řád)**
- z **- základ soustavy**

jeho hodnota se spočítá $\sum_{-m}^{n} a_i z^i$

Příklad

Hodnota čísla v osmičkové soustavě $154,2_8$

$$\begin{array}{ccccccccc} 1 & & 5 & & 4 & , & 2 & & \\ 1\times8^2 & + & 5\times8^1 & + & 4\times8^0 & + & 2\times8^{-1} & = & 108,25 \end{array}$$

Převody číselných soustav

Desítková na dvojkovou (celá část)

Převod se dělá postupných dělením základem soustavy (2).

Příklad převod čísla 57

$$57_{10} \approx A_2 \qquad A_2 = 111001_2 \leftarrow \left\{ \begin{array}{rl} 57 : 2 = 28 & \text{zbytek} \quad 1 \ \ldots a_0 \\ 28 : 2 = 14 & \text{zbytek} \quad 0 \\ 14 : 2 = \ \ 7 & \text{zbytek} \quad 0 \\ 7 : 2 = \ \ 3 & \text{zbytek} \quad 1 \\ 3 : 2 = \ \ 1 & \text{zbytek} \quad 1 \\ 1 : 2 = \ \ 0 & \text{zbytek} \quad 1 \ \ldots a_5 \end{array} \right.$$

Desítková na dvojkovou (zlomková část)

Převod se dělá postupných násobením základem soustavy (2).

Příklad převod čísla 0,65625

$$0{,}65625_{10} \approx A_2 \qquad \left\{ \begin{array}{llll} 0{,}65625 & \cdot 2 = 1 & , 3125 & \dots a_{-1} \\ 0{,}3125 & \cdot 2 = 0 & , 625 & \\ 0{,}625 & \cdot 2 = 1 & , 25 & \\ 0{,}25 & \cdot 2 = 0 & , 5 & \\ 0{,}5 & \cdot 2 = 1 & , 0 & \dots a_{-5} \end{array} \right.$$

$$A_2 = 0{,}10101_2 \longleftarrow$$

Šestnáctková soustava

Dec.	Hex.	Bin.	Oct.		Dec.	Hex.	Bin.	Oct.
0	0	0000	0		8	8	1000	10
1	1	0001	1		9	9	1001	11
2	2	0010	2		10	A	1010	12
3	3	0011	3		11	B	1011	13
4	4	0100	4		12	C	1100	14
5	5	0101	5		13	D	1101	15
6	6	0110	6		14	E	1110	16
7	7	0111	7		15	F	1111	17

Mezi dvojkovou soustavou a šestnáctkovou se převádí tak, že jeden symbol ze šestnáctkové připadá na 4 symboly z dvojkové soustavy. Stejně se převádí mezi dvojkovou a osmičkovou, pouze jeden symbol osmičkové připadá pouze na tři symboly dvojkové soustavy. Podobně by se prováděl převod mezi dalšími příbuznými soustavami.

Nejčastěji používané soustavy

- dvojková (binární) z=2 {0, 1}
- osmičková (oktalová) z=8 {0, 1, 2, 3, 4, 5, 6, 7}
- desítková (dekadická) z=10 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- šestnáctková (hexadecimální) z=16 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Řádová mřížka

Určuje formát zobrazitelných čísel, maximální velikost a přesnost. Definuje nejvyšší (n) a nejnižší (m) řád.

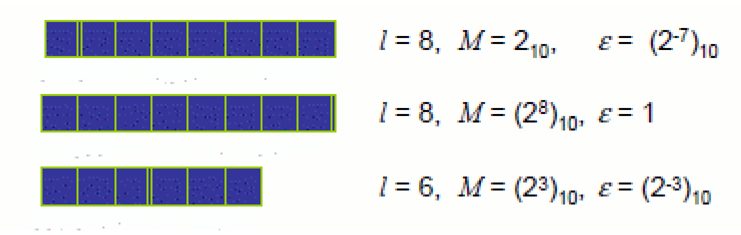
Příklad

n=3, -m=0 | 3 | 2 | 1 | 0 |

n=0, -m=-3 | 0 | -1 | -2 | -3 |

vlastnosti řádových mřížek

- délka ř.m. - počet řádů ve mřížce $l = n + m + 1$. Pozor - kladné m znamená záporné řády ve mřížce
- jednotka ř.m. - nejmenší hodnota kterou řádová mřížka umí zobrazit $\varepsilon = z^{-m}$
- modul ř.m. - nejmenší hodnota kterou řádová mřížka už nemůže zobrazit, tedy tak velká že už se nevejde do ř.m. $M = z^{n+1}$

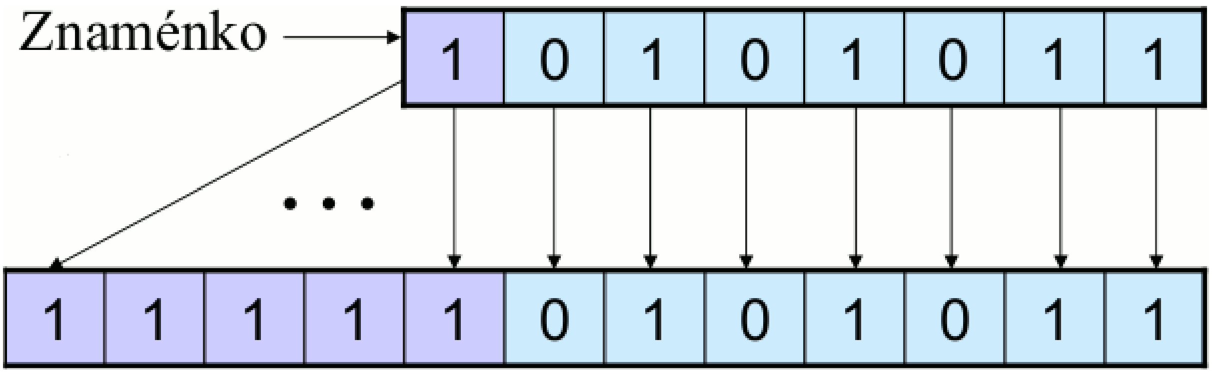


Zobrazení čísel se znaménkem

Standardní číselné soustavy nemají znaménko. Řešením je použít číselný kód.

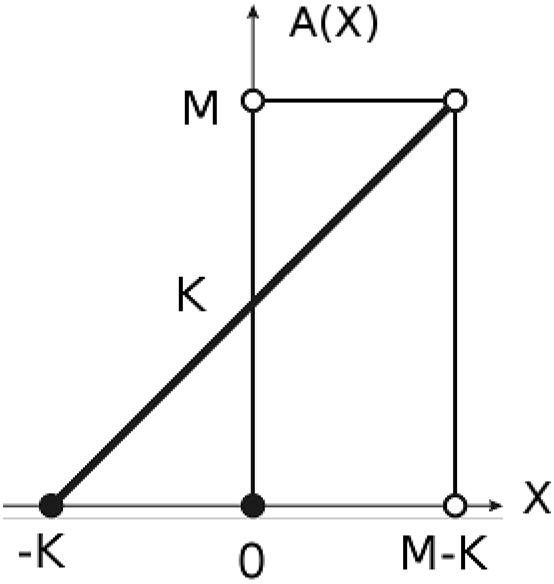
Nejpoužívanější číselné kódy

- ***přímý kód***
  - **znaménko je přidáno dodatečně, např. -10,+10 (1. bit zleva je pro znaménko a pak je číslo)**
  - **umožňuje zobrazit rozsah $<-2^{n-1}, 2^{n-1}>$, kde n je velikost řádové mřížky**
  - **existuje kladná a záporná nula**
- ***aditivní kód***
  - **s posunutou nulou, např. 8→0, 0→8**
  - **umožňuje zobrazit rozsah $<-2^{n}+k, 2^{n-1}+k>$, kde n je velikost řádové mřížky a k je posunutí**
- ***doplňkový kód***
  - **mapuje záporná čísla do horní poloviny rozsahu zobrazitelných čísel**
  - **zachovává pravidla pro sčítání a násobení**
  - **znaménko je možné poznat podle 1. bitu zleva**
  - **umožňuje zobrazit rozsah $<-2^{n}, 2^{n-1}-1>$, kde n je velikost řádové mřížky. (Pozn. Vojtech Kral - řekl bych, že rozsah je $<-2^{n-1}, 2^{n-1}-1>$ s tím, že n je počet bitů čísla, aby se to nepletlo se zmíněným n v souvislosti s formátem mřížky)**
  - **znaménkové rozšíření - při zvětšení velikosti řádové mřížky se do horních bitů zkopíruje nejvyšší bit, zachovává hodnotu čísla**



Aditivní kód

Číslo je uloženo posunuto o nějakou konstantu do kladných čísel, zpravidla je tato konstanta



1/2 modulu.

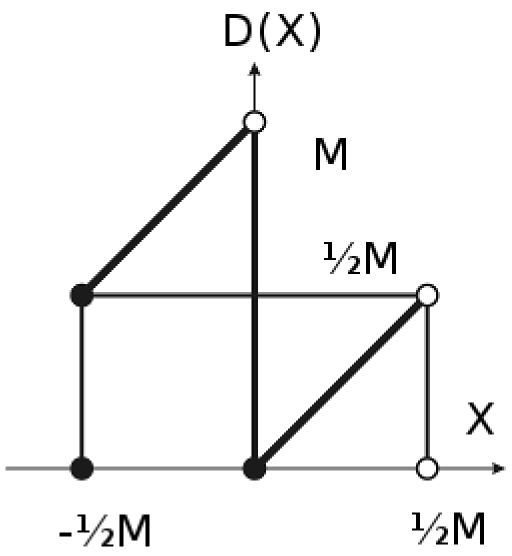
Příklad v desítkové soustavě. Mřížka velikosti 2 ⇒ modul = 100 ⇒ konstanta je 50.

- Číslo 0 je ve skutečnosti -50 (0-50)
- číslo 50 je ve skutečnosti 0 (50-50)
- číslo 99 je ve skutečnosti 49 (99-50)

Doplňkový kód

Příklad zápisů čísel v doplňkovém kódu v mřížce velikosti 3

X	
0	000
1	001
2	010
3	011
-4	100
-3	101
-2	110
-1	111



algoritmus změny znaménka v doplňkovém kódu

1. inverze všech bitů
2. přičtení jedičky

tento algoritmus jde také použít pro získání hodnoty záporného čísla nebo získání zápisu záporného čísla z kladného

příklad

zobrazení čísla 5 do záporných hodnot (v mřízce velikosti 4)

0101	výchozí číslo
1010	inverze bitů
1011	přičtení jedničky

Bitové operace

Posuv

Přesunutí bitů ve slově. Posuv vlevo odpovídá násobení dvěma, posuv vpravo dělení dvěma.

- logický posuv - na uvolněná místa jsou vloženy 0, v javě operátory » a «
- aritmetický posuv - v javě operátory »> a «, posuv vpravo nemění znaménko
- cyklický - na uvolněné místo jsou uložené přeteklé bity

logický posuv vlevo

```
0101 1101 -> 1011 1010
```

aritmetický posuv vpravo

```
1101 1001 -> 1110 1100
```

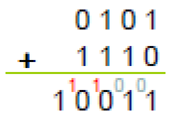
cyklický posuv vlevo

```
1101 1001 -> 1011 0011
```

Realizace aritmetických operací

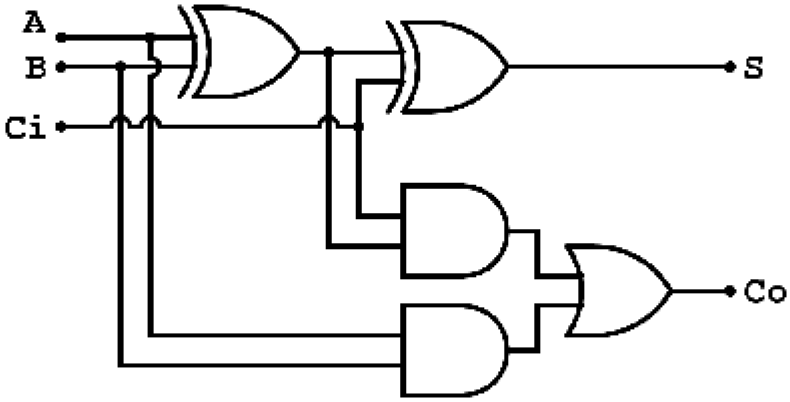
Sčítání

Sčítání funguje stejně jako jsme se učili na základní škole sčítat „pod sebe“. Akorát s tím rozdílem že v binární soustavě je 1+1 = 0 a do dalšího řádu jde další jednička.

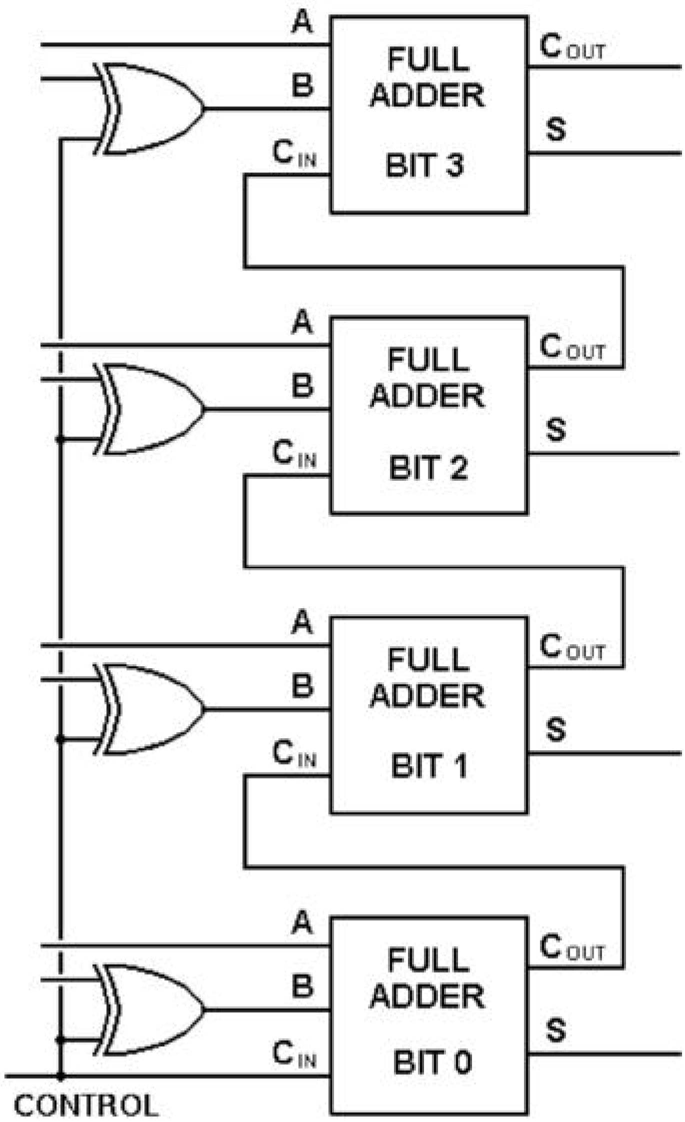


Zapojení obvodu

Zapojení jednobitové sčítačky.
A, B ... vstupy
Ci ..... přenos z nižšího bitu
Co ..... přenos (např. 1 + 1 = 0 přenos 1 - všechny čísla jsou binární)
S ...... výsledný součet



Vícebitová sčítačka se vytvoří zřetězením více jednobitových sčítaček (místo full adder si představ předchozí obrázek).



Násobení

Násobení funguje také jak jsme se učili na základní škole s desítkovou soustavou.

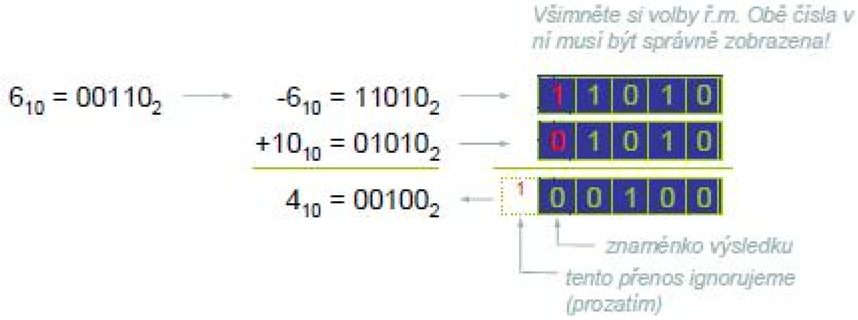
```
     1110
  ×   101
  -------
     1110
+   0000
+ 1110
  -------
  1000110
```

Odečítání v doplňkovém kódu

Odečítání čísla v doplňkovém kódu ⇔ přičítání záporného čísla

Příklad

10 - 6 ⇔ 10 + (-6)



Přenos do vyšších řádů neříká nic o tom jestli číslo při výpočtu přeteklo. Přetečení detekujeme pokud při sčítání dvou kladných čísel výjde záporné, nebo při sčítání dvou záporných čísel výjde kladné.

Dělení ve dvojkové soustavě

Na n-adickou soustavu můžeme přenést veškeré naše znalosti z desítkové soustavy. Např. v teorii čísel kritéria dělitelnosti v desítkové soustavě pro číslo devět (tedy n-1) záleží na ciferném součtu, i stejnou analogii má kritérium dělitelnosti číslem n+1 (analogie kritéria dělitelnosti jedenácti) atd.

Věnujme naši pozornost nyní dělení se zbytkem a odčítání v n-adické soustavě (tedy i ve dvojkové soustavě). Analogii si vezmeme z desítkové soustavy. Rozhodně dělení je složitější operace, než-li odčítání, proto začneme odčítáním. Pro odčítání v n-adické soustavě trošku upravíme známý algoritmus odčítání z desítkové soustavy, a to tak, že budeme místo odčítání přičítat takzvaný (p-1) doplněk. Postup si vysvětlíme na příkladu v desítkové soustavě:

```
         337
 - 218
-------
         337
 + 781  (devítkový doplněk čísla 218)
-------
  1118  a první cifru přeneseme k poslední
   119 = 337 - 218.
```

Jiný příklad:

```
         140
 -  84
-------
         140
 +  15  (devítkový doplněk čísla 84)
-------
   155
   a první cifru přičteme k poslední
   56 = 140 - 84.
```

Pro aritmetické operace s dvojkovou soustavou budeme potřebovat sčítalku, násobilku, odčítalku a dělitelku. Sčítání, ani násobení ve dvojkové soustavě, nedělá potíže (malá násobilka):

```
       + | 0 |  1               * | 0 | 1
       -----------              ---------
       0 | 0 |  1               0 | 0 | 0
       -----------              ---------
       1 | 1 | 10               1 | 0 | 1
```

Odtud i plyne algoritmus odčítání ve dvojkové soustavě: menšitel 'převrátíme', tj. zaměníme nulu za jedničku a naopak. Čísla sečteme a přeneseme jednotku největšího řádu k cifře nultého řádu. Následuje příklad ve dvojkové soustavě:

```
            110111010001
     -     10111010
    ---------------
            110111010001
     +     01000101    doplněk menšitele
    ---------------
      111000010110    součet, a přeneseme první jedničku k poslední
    ---------------
       11000010110
     +           1
    ---------------
       11000010111 = 110111010001 - 10111010.
```

Při operaci dělení se uplatňuje jak sčítání a odčítání, tak i násobení. Opět si pomůžeme analogií s desítkovou soustavou a postup vysvětlíme na příkladu dělení ve dvojkové soustavě:

```
      110111010001 : 10111010 = 1
     - 10111010
    ---------------
      11011101
     + 01000101  (doplněk)
    ---------------
     100100010  (přenos)
    ---------------
      00100010
     +        1
    ---------------
      00100011  (a sepíšeme další cifru a nuly vynecháváme)
        1000110
```

a pokračujeme dále:

```
            110111010001 : 10111010 = 10101
        1000110
        10001100
      - 10111010
      ------------
       + 01000101 (doplněk)
      ------------
               10111001 (přenos)
      ------------
                0111001
       +        1
      ------------
                0111010 a sepíšeme další cifru...
          01110100 , tj.
           1110100 protože 1110100 < 10111010, dostáváme nulkrát a sepíšeme poslední cifru
                 11101001
        - 10111010
       ------------
               + 01000101 (doplněk), tj.
             ------------
           11010001
         + 01000101
       ------------
          100010110 a přeneseme první cifru:
       ------------
                 00010110
         +        1
       ------------
              10111
```

tedy 110111010001 : 10111010 = 10101 (zb. 10111).

(Petr Vaněk (Vanekp4): v tomhle příkladu je chyba na čtvrtém řádku. 10001100 < 10111010, takže je nelze odečíst, místo toho je třeba sepsat další nulu. Správný výsledek je tedy 10011. Desítkově je tento příklad 3537 : 186 = 19)

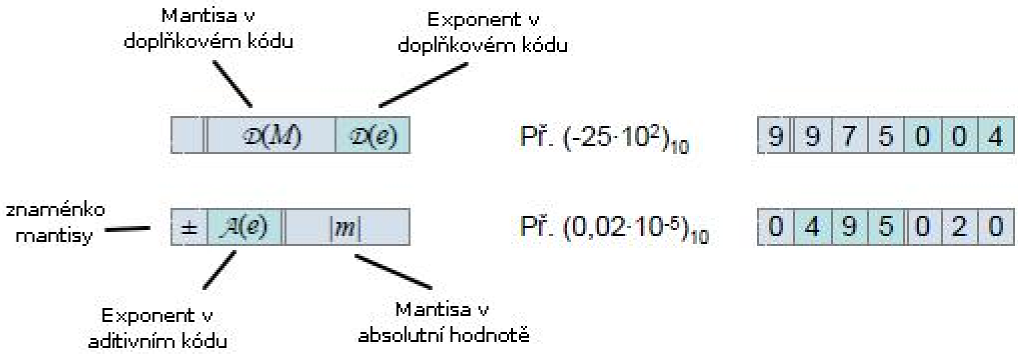
strucny popis delenia [http://www.helpwithpcs.com/courses/multiplying-dividing-binary-numbers.htm] doplnkovy link pre pochopenie priebehu delenia [http://www.mathsisfun.com

/long_division3.html]

Pohyblivá řádová čárka

Kvůli zvětšení rozsahu a přesnosti čísel byla vytvořena čísla s pohyblivou řádovou čárkou. Číslo se skládá ze dvou částí, z mantisy = informace o „hodnotě“ čísla. Její velikost určuje přesnost čísla, tedy na kolik míst je přesné. Dále číslo má exponent, který určuje pozici řádové čárky (desetinné).

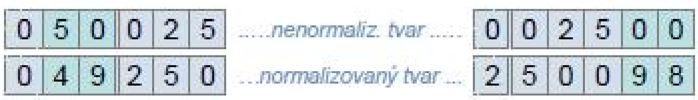
Příklad řádových mřížek



Normalizovaný tvar

- zjednodušuje aritmetické operace
- tvar kdy nejde mantisu posunout víc doleva, tedy takový kde je exponent co nejnižší

Příklad zápisu čísla $0{,}025_{10}$



První číslo je zápis $0{,}25 \times 10^{-1}$ (exponent je v aditivním kódu s konstantou 50). Druhé číslo je $2{,}5 \times 10^{-2}$ (exponent je v doplňkovém kódu).

Alfanumerické kódy

Používané například jako metoda kódování textu. Rozlišujeme znakovou sadu a kódování. Znaková sada (např. unicode) definuje seznam znaků. Kódování určuje numerický kód pro jednotlivé znaky znakové sady. Nemusí definovat kód pro všechny znaky. Pokud není definovaná endianita (pořadí bytů ve slově), tak se u více bajtových kódování používá BOM (byte order mark). Je poslán jako první znak textu a nemá jiný význam než určení pořadí bytů v jednom znaku.

- ASCII 7 bit - anglická abeceda + speciální symboly
- ASCII 8 bit - různá kódování - české : windows-1250, iso-8859-2, a další už historické
- UTF 8
  - variabilní počet bajtů na znak 1 až 4
  - je ASCII kompatibilní (znaky definované v ASCII jsou kódované 7 bity - 0XXX

XXXX)
- určení počtu bytů na znak - 110X XXXX(2 byty), 1110 XXXX(3 byty), 1111 0XXX(4 byty), 10XX XXXX(2 a další byte)
- používá se pro přenos dat mezi počítači
- BOM - 0xEF 0xBB 0xBF
- UTF 16 (UCS 2)
  - 2 bajty na znak
  - používá se pro vnitřní reprezentaci textu v programu
  - není vhodné pro přenos dat mezi počítači (definuje speciální symboly, které se na jednotlivých platformách můžou lišit. např. konce řádků, oddělovač souborů, ukončovací 0 atd.)
  - BOM - 0xFE 0xFF

užitečné odkazy

- Poziční číselné soustavy [http://cs.wikipedia.org/wiki/Poziční_číselná_soustava]
- Data, jejich zobrazení a zpracování. [http://service.felk.cvut.cz/courses/Y36SAP/pdf-web/sap-5-data.pdf]
- Aritmetické operace. Pohyblivá řádová čárka. [http://service.felk.cvut.cz/courses/Y36SAP/pdf-web/sap-6-oper.pdf]

Zdroje

slidy k Y36SAP a X36SKD
http://www.play-hookey.com/digital/images/faddsub-4b.gif [http://www.play-hookey.com/digital/images/faddsub-4b.gif]
http://www.doc.ic.ac.uk/~ih/teaching/lectures/comparch/logic/adder/ [http://www.doc.ic.ac.uk/~ih/teaching/lectures/comparch/logic/adder/]

spolecne/spol11.txt · Poslední úprava: 2010/06/15 08:34 autor: Strider