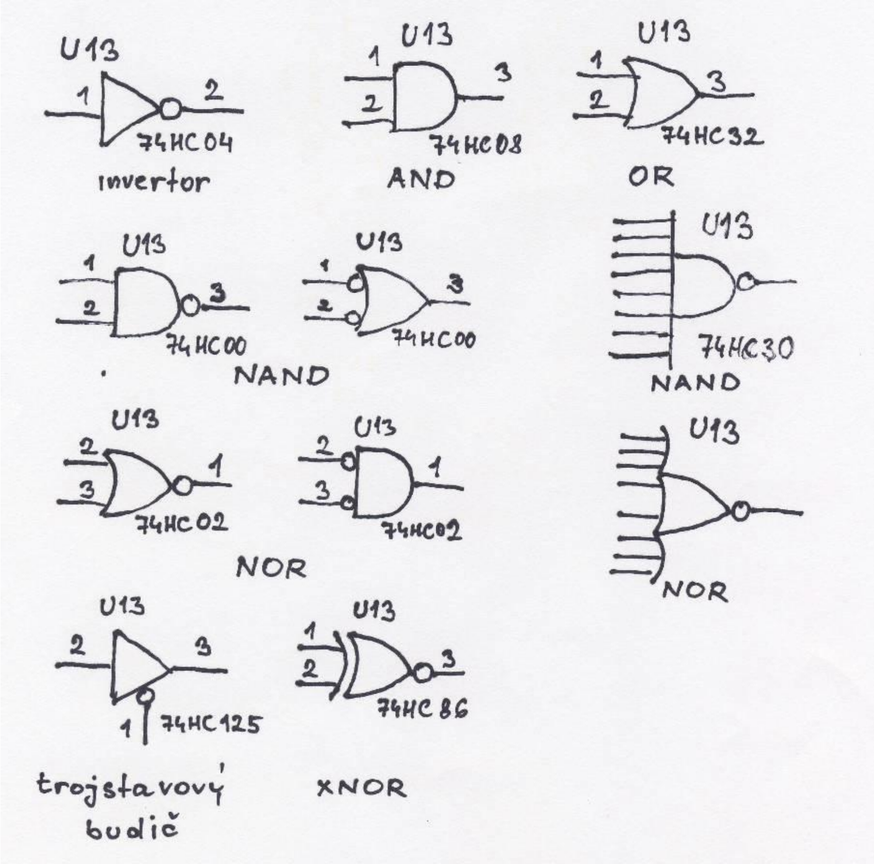
19. február 2015

1. prednáška
ČÍSLICOVÉ POČÍTAČE

Jana Milanová

Fakulta riadenia a informatiky,
Katedra technickej kybernetiky

- Americké symboly kombinačných logických členov



Logické systémy

- disjunktívna forma – „1“
- konjunktívna forma – „0“
- negácia $\bar{a}$
- logický súčin a . b
- logický súčet a + b
- de Morganove pravidlá
  - $\overline{a + b} = \bar{a} \cdot \bar{b}$
  - $\overline{a \cdot b} = \bar{a} + \bar{b}$

BOOLEOVA ALGEBRA

- a + (b + c) = (a + b) + c <- asociatívny zákon
- a . b = b . a <- komutatívny zákon
- a . 0 = 0
- a + 1 = 1
- a + a = a
- a . a = a
- $\bar{\bar{a}} = a$
- $a + \bar{a} = 1$
- $a . \bar{a} = 0$
- $a \oplus a = 0$

Reprezentácia dát v číslicovom počítači

- prečo číslicový počítač?
  - počítač, ktorý spracováva dáta obsahujúce 2 presne rozlíšiteľné hodnoty informácie – log. 0 (false) a log. 1(true), združením dostávame väčšiu hodnotu informácie,
  - výhoda – oba stavy ľahko rozpoznateľné,
  - pri zakódovaní väčšieho množstva rôznych slov budú slová dlhšie – menšia nevýhoda ako nejednoznačné alebo príliš komplikované rozpoznanie.

Reprezentácia dát v číslicovom počítači

- binárna sústava (dvojková) –
  - pozičná číselná (polyadická) sústava – cifra je daná hodnotou a pozíciou v zápise čísla

  (nepozičná sústava – rímska, hodnota je určená konfiguráciou znakov;

  XIV – 14

  XII – 12)
  - prvé mechanické počítače využívali desiatkovú sústavu – 10 prstov,
  - teraz výpočty v dvojkovom kóde, výsledky môžu byť zobrazené v tvare čitateľnejšom ľuďom,
  - jednotlivé miesta – bity (b), 8 bitov – 1 bajt(B),

Reprezentácia dát v číslicovom počítači

- reprezentácia čísel v počítači je závislá na druhu čísla,
- základné druhy čísel:
  - prirodzené – patrí sem aj 0,
  - celé,
  - desatinné (reálne)
- prevod z desiatkovej sústavy do dvojkovej
- prevod z dvojkovej do hexadecimálnej (0-9,A-F; pričom 0xA = 10_{10}),
  - napr. 0b 1110 0101
0x E 5

- $345{,}67_{10} = 3.10^2 + 4.\ 10^1 + 5.\ 10^0 + 6.10^{-1}+7.10^{-2}$
- $101{,}01_2 = 1*2^2 + 0*2^1 + 1*2^0 + 0*2^{-1} + 1*2^{-2}$
- $0xA7 = 10.16^1 + 7.16^0$

REPREZENTÁCIA DÁT V ČÍSLICOVOM POČÍTAČI

- príklad:
  - prevod z desiatkovej do dvojkovej sústavy:
    - $25_{10} : 2 = 12$ zv. **1**

      $12_{10} : 2 = 6$ zv. **0**

      $6_{10} : 2 = 3$ zv. **0**

      $3_{10} : 2 = 1$ zv. **1**

      $1_{10} : 2 = 0$ zv. **1**

    $\mathbf{25_{10} = 11001_2}$ (opačné poradie)
  - prevod z dvojkovej sústavy do šestnástkovej:
    - $11001_2 = 0001\ \ 1001_2$

      $1.2^0 \quad 1.2^3 + 1.2^0$

      0x 1 9

    $\mathbf{11001_2 = 0x19}$

PRIRODZENÉ ČÍSLA V BINÁRNEJ SÚSTAVE

- na ich zobrazenie sa využíva **prirodzený dvojkový kód**,
- n- bitové číslo má rozsah <0 , 2^n – 1>, t. j. je možné do neho uložiť n rôznych hodnôt (napr. pre 8 bitové číslo je rozsah 0 .. 255 – t.j. 256 rôznych hodnôt),
- pre prácu s číslami v dvojkovom kóde platia pravidlá:

0+0 = 0	0-0 = 0	0.0=0	0/1=0
0+1 = 1	1-0 = 1	0.1=0	1/1=1
1+1 = 10	10-1=1	1.1=1	0/0, 1/0 nie je definované

PRIRODZENÉ ČÍSLA V BINÁRNEJ SÚSTAVE

- základné matematické operácie v binárnej sústave:

sčítanie:	33	100001
	27	011011
	60	111100
násobenie:	33	100001
	. 27	011011
	231	100001
	66	100001
	891	100001
		100001
		1101111011

odčítanie:	33	100001
	- 27	011011
	6	000110
delenie:	33:3 = 11	100001:11 = 01011
	03	100
		100
		11

- delenie je možné vykonať pomocou operácie odčítania,

DELENIE PRIRODZENÝCH ČÍSEL V BINÁRNEJ SÚSTAVE

- príklad:

10-tková → 33:3 = 11
03

2-ková → 100001:11 = 01011
100
100
11

- delenec v dvojkovej sústave má 6 bitov a deliteľ 2 bity,
- postup:

1. začíname od bitu delenca s najvyššou váhou,
2. zoberieme toľko bitov delenca, ako je počet bitov v deliteli (t.j.10_2) – toto číslo nazveme **z**,
3. ak je číslo **z** väčšie alebo rovné ako deliteľ, do výsledku sa zapíše 1, inak sa zapíše 0,
4. ak bola pripísaná do výsledku 1, odpočítame od čísla **z** deliteľa (z= z - deliteľ),
5. do **z** pridáme z delenca ďalší bit, ktorý bude v čísle **z** na pozícii 2^0 - dostávame nové číslo **z,**
6. opakujeme kroky 3. až 5. dovtedy, kým už delenec neobsahuje ďalší bit, ktorý by bolo možné do čísla **z** pridať,
7. číslo, ktoré zostalo v **z** je zvyšok po delení (v našom prípade 0),

CELÉ ČÍSLA V BINÁRNEJ SÚSTAVE

- potrebné rozlíšiť či je číslo kladné alebo záporné,
- viacero kódov, ktoré sa líšia hlave v spôsobe zobrazenia záporných čísel:
  - priamy kód,
  - inverzný kód,
  - doplnkový kód,
  - predpätý kód,
- napr. 36-27 = 36+(-27)

CELÉ ČÍSLA – PRIAMY KÓD

- pre ľudí najprirodzenejší,
- kladné a záporné čísla sú zobrazené rovnako, rozdiel je len v znamienku,
- znamienko je umiestnené v najvyššom bite,
  - kladné číslo –log. 0,
  - záporné číslo– log. 1,
- n – bitové číslo má rozsah $<-2^{n-1}+1, 2^{n-1}-1>$ (8-bitové číslo- rozsah <-127, 127>) -> číslo 0 je reprezentované v tomto prípade dvoma spôsobmi – „kladná nula“ (00000000), „záporná nula“(10000000),
- matematické operácie sa uskutočňujú s absolútnymi hodnotami a znamienko sa určí samostatne,

CELÉ ČÍSLA – INVERZNÝ KÓD

- $-33_{10} = -00100001_2$ = 11111111-11111111-00100001

= 11111111-00100001-(100000000-1)

= 11011110+1-100000000

číslo sme dostali negáciou (inverziou) každého bitu pôvodného čísla – získali sme **jednotkový doplnok**,

- ak sa na reprezentáciu záporných čísel používa jednotkový doplnok – hovoríme o **inverznom kóde**,
- n – bitové číslo má rozsah <$-2^{n-1}+1$, $2^{n-1}-1$> (8-bitové číslo- rozsah <-127 , 127>) -> číslo 0 je v inverznom kóde reprezentované 2x,
- aj v tomto prípade majú záporné čísla najvyšší bit rovný 1, **nejde však o znamienkový bit**

33	100001	00100001
- 27	- 011011	11100100
6		100000101

k výsledku potrebné +1 – 100000000 => jednotku v najvyššom bite pripočítame k najnižšiemu bitu – tzv. **kruhový prenos**

```
  100000101
- 100000000
+         1
        110  = 6₁₀
```

- pri použití inverzného kódu je možné vykonať operáciu odčítania pomocou sčítania,

CELÉ ČÍSLA – DOPLNKOVÝ KÓD

- odvodený od inverzného kódu,
- okrem negácie všetkých bitov sa k negovanému číslu pripočíta 1 – vzniká dvojkový doplnok,
- v doplnkovom kóde „nula“ už nie je reprezentovaná dvakrát,
- n – bitové číslo má rozsah $<-2^{n-1},2^{n-1}-1>$ (8-bitové číslo- rozsah <-128,127>),
- zjednodušenie odčítania – vo výsledku sa iba ignoruje najvyšší bit

33	100001	00100001
- 27	- 011011	11100101
6		~~1~~00000110

Celé čísla – predpätý kód

- ku každému číslu sa pripočíta posunutie,
- ak je posunutie 4, bude číslo 3 zobrazené pomocou kódu 111 = 7 (4+3)
- v tomto prípade je možné aj záporné čísla zobrazovať ako kladné (napr. pri 8-bitovom čísle a posunutí 128 bude číslo -128 zobrazené ako 0, 0 bude 128, a 127 bude 255)
- výhodou je jednoduché porovnanie veľkosti dvoch čísel bez ohľadu na znamienko,

Reálne čísla

- v číslicových počítačoch sú reprezentované dvomi spôsobmi:
  - s pevnou rádovou (desatinnou) čiarkou,
  - s pohyblivou rádovou čiarkou,
- dôležité vedieť:
  - rozsah zobrazenia,
  - presnosť zobrazenia,

REÁLNE ČÍSLA – PEVNÁ RÁDOVÁ ČIARKA

- počet rádov čísla pred aj za čiarkou je presne daný, napr. XXX,YYYY – nevýhoda, pretože ľahko môže dôjsť k pretečeniu výsledku matematickej operácie cez rozsah zobrazenia,
- ak dôjde k pretečeniu za rádovou čiarkou, dôjde ku zaokrúhleniu čísla – v niektorých prípadoch toto zaokrúhlenie je akceptovateľné; horšie, ak dôjde k pretečeniu pred rádovou čiarkou,
- k pretečeniu najčastejšie dochádza pri:
  - násobenie čísel väčších ako 1 (v absolútnej hodnote) – možné ošetrenie také, že všetky čísla upravíme tak, aby boli v absolútnej hodnote menšie ako 1 – tzv. **normalizácia údajov** (napr. 120 m = 0,12 km),
  - viacnásobné sčitovanie,
  - delenie číslom menším ako 1 (v absolútnej hodnote),
- spôsoby umiestnenia rádovej čiarky:
  - umiestnenie za najmenej významnou číslicou – celé čísla v rozsahu $<-2^{n-1},2^{n-1}-1>$; k zaokrúhľovaniu dochádza len pri delení,
  - umiestnenie pred najvýznamnejšou číslicou – čísla v absolútnej hodnote menšie ako 1, rozsah hodnôt $<-1+2^{-(n-1)},1-2^{-(n-1)}>$; čísla menšie ako $2^{-(n-1)}$ sa zobrazia ako 0 – tzv**. strojová nula** (napr. ak n= 3: rozsah $<-\frac{7}{8}, \frac{7}{8}>$, číslo menšie ako $\frac{1}{8} \stackrel{\text{def}}{=} 0$)
- výhoda - jednoduchosť práce s číslami - rovnaká náročnosť ako pri celých číslach,
- pevná rádová čiarka sa používa v niektorých špeciálnych typoch procesorov, napr. v starších druhoch digitálnych signálnych procesorov (DSP). V novších sa už používa pohyblivá rádová čiarka (väčšia náročnosť implementácie už nepredstavuje významnejšie skomplikovanie, a teda zdraženie procesora).

REÁLNE ČÍSLA – POHYBLIVÁ RÁDOVÁ ČIARKA

- značné zväčšenie rozsahu, avšak zhoršenie presnosti zobrazenia,
- čísla v pohyblivej rádovej čiarke sú v skutočnosti reprezentované dvojicou čísel v pevnej rádovej čiarke:
- $x = m.z^e$,

kde m je **mantisa**, z je **základ** (v číslicových počítačoch 2) a e je **exponent**,

- kvôli problémom s pretečením býva mantisa normalizovaná napr. tak, že platí:

$2 > |m| \geq 1$, pričom samotná jednotka sa neukladá; ukladá sa len desatinná časť mantisy,

- napr. :
  - 8-bitové číslo v pevnej rádovej čiarke v tvare ±XX,YYYYY má menší rozsah ako 8-bitové číslo v pohyblivej rádovej čiarke v tvare ±mm±eeee, avšak dosahuje vyššiu presnosť,

Desiatkové čísla v číslicovom počítači

- v niektorých procesoroch (napr. procesory architektúry x86) existuje možnosť pracovať priamo s desiatkovými číslami, teda bez prevodu do dvojkovej sústavy, do dvojkového kódu sú prevedené len jednotlivé číslice- **BCD kód** (Binary Coded Decimal number)
- na zakódovanie jednej desiatkovej číslice sa zvyčajne používajú 4 bity, napr. 239_{10} = 0010 0011 1001
- matematické operácie v BCD - podobne ako pri priamom kóde, len je potrebné robiť desiatkovú korekciu – t.j. pripočítanie čísla 6 (rozdiel medzi základom hexadecimálnej a desiatkovej sústavy) k tým častiam výsledku, kde dôjde k prekročeniu čísla 9,
- napr.:

```
33     0011  0011
27     0010  0111
-----------------
60     0101  1010
             0110   <- korekcia
-----------------
       0110  0000
```

ZNAKY V ČÍSLICOVÝCH POČÍTAČOCH

- číslice (0-9), písmená (a-z, A-Z), špeciálne znaky (?,#,CR,LF...) sú v číslicovom počítači zakódované pomocou binárneho kódu.
- priradenie určitého kódu danému znaku sa nazýva kódovanie,
- jeden z prvých, ktorý sa stále používa je **ASCII** (American Standard Code for Information Interchange). ASCII kód je **7-bitový,** t.j. dokáže rozlíšiť 128 znakov. Ôsmy bit sa používal ako paritný. ASCII kód obsahuje len základné písmená (bez diakritiky), číslice a nejaké špeciálne znaky. Bez problémov je v ňom možné písať len v latinčine, swahilčine, havajčine a americkej angličtine,
- neskôr sa ASCII kód rozšíril na 8-bitový a vznikla sada kódov označovaných ako ISO-8859. Spodných 128 znakov je rovnakých ako v ASCII a horných 128 znakov sa líši podľa verzie kódu. ISO-8859-1 – západná Európa, 2 – východná (aj Slovensko), 3 – južná, 4 – severná Európa, 5 – cyrilika, 6 – arabské znaky, 7 – grécke, 8 – hebrejské (židovské), atď.,
- v súčasnosti sa stále viac presadzuje používanie kódu ISO-10464 resp. UNICODE, tieto dva štandardy sú prakticky totožné. Unicode je multibajtový kód (na zakódovanie znaku sa používa zvyčajne 2 B, ale môže sa aj viac); znak je možné v Unicode zakódovať rozličným spôsobom. Napr. 'é' má svoj vlastný kód, ale môže byť zapísané aj ako 'e' za ktorým nasleduje '´' (dĺžeň).

Ďakujem za pozornosť

Použité materiály:
Peter Gubiš – Číslicové počítače (podporné učebné texty)
Ondrej Karpiš – Prednášky k predmetu Číslicové počítače