1. ZOBRAZENÍ INFORMACÍ V POČÍTAČI

Tato příručka se zabývá zobrazením informací v počítačích.

1.1. Základní pojmy a jejich vysvětlení

poziční soustava
číselná soustava, ve které je každé pozici přiřazena určitá váha

z
základ soustavy. V této knize nejčastěji dvě.

n
řád čísla před řádovou čárkou

m
řád čísla za řádovou čárkou

bit
binary digit (binární číslice) čili jeden řád dvojkového čísla. Vyjadřuje právě jeden ze dvou stavů: 0 nebo 1

slabika = byte
skupina osmi bitů

slovo
skupina slabik. Pokud je procesor 16bitový, pak slovo jsou dvě slabiky; je-li procesor 32bitový, pak slovo jsou čtyři slabiky atp.

bitová mřížka
znázornění slabiky či slova graficky:

a_n	a_{n-1}	…	a_1

nebo ekvivalentní zjednodušenou formou:

$[a_n a_{n-1} \ldots a_2 a_1]$

nebo vzorcem

$$x = \sum_{i=1}^{n} a_i \cdot z^{i-1}$$

prefix &H
určuje, že se jedná o číslo v šestnáctkové soustavě

prefix &O
určuje, že se jedná o číslo v osmičkové soustavě

uložitelná a zobrazitelná čísla
Platí pro reálná čísla – u celých čísel pojem zobrazitelná a uložitelná splývá. Tyto dva pojmy byly zavedeny profesorem Koníčkem. Uložitelná čísla jsou podmnožinou zobrazitelných. Více v kapitole o reálných číslech.

přetečení – overflow
výsledek přesahuje možnosti největšího uložitelného čísla.

podtečení
výsledek je sice nenulový, avšak menší, než je nejmenší uložitelné číslo.

2. POLYADICKÉ SOUSTAVY

Mimo polyadických soustav ještě existují soustavy zbytkových tříd. Ale těmi se zde zabývat nebudu.

Číselné soustavy se skládají z:

- množiny čísel (číslice)
- množiny slov (číselné obrazy), znaky slov jsou číslice

Racionální čísla (záporná i nezáporná) se vyjadřují:

$$A = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z^1 + a_0 z^0 , a_{-1} z^{-1} + \ldots + a_{-m} z^{-m}$$ Rovnice 2.1

nebo stručněji

$$A = a_n a_{n-1} \cdots a_1 a_0 , a_{-1} a_{-2} \cdots a_{-m}$$ Rovnice 2.2

kde číslice mohou nabývat hodnot:

$$a_i = 0,1,2,\ldots,z-1$$ Rovnice 2.3

kde: A = dané číslo; a_i = číslice daného řádu (číslo celé); z = základ; n, m = mocnina, řád. Připustíme-li růst m do nekonečna, rozšíříme definici na iracionální čísla.

(i) Vzorec pro výpočet počtu míst

Čím větší je základ soustavy, tím větší počet kódových znaků se používá, ale zase je třeba méně řádů. Pokud tedy

$$a_2^n = a_1^m$$ Rovnice 2.4

pak platí (a_1 a a_2 jsou základy navzájem různých soustav n a m; x je základ logaritmu – volí se většinou 10)

$$\frac{n}{m} = \frac{\log_x \cdot a_1}{\log_x \cdot a_2}$$ Rovnice 2.5

2.2. Rozšiřující a zobecňující zásady pro počítače

- iracionální čísla se vyjadřují stejně jako racionální. Ovšem, protože mají neukončený rozvoj (což u počítačů nepřipadá v úvahu) je třeba je aproximovat na daný počet míst
- záporná čísla se vyjadřují znaménkem -. U počítačů jde o tzv. přímý kód
- komplexní čísla se vyjadřují pomocí programovacích technik
- u dvojkové soustavy lze pomocí základu z = -2 vyjádřit bez znaménka čísla záporná i nezáporná. Algoritmy jsou ovšem složité.

2.3. Používané číselné soustavy

2.3.a) Poziční dvojková soustava

Obsahuje tyto znaky pro číslice 0; 1. Každé dvojkové číslo má průměrně 3,3krát více míst než stejné číslo vyjádřené v desítkové soustavě. Nehodí se sice pro ruční výpočty, ale je ideální pro strojové zpracování.

2.3.b) Dvojkově-desítková soustava

Každý řád desítkového čísla se vyjadřuje tetrádou dvojkového pozičního kódu.

2.3.c) Poziční osmičková soustava

Obsahuje tyto znaky pro číslice 0; 1; 2; 3; 4; 5; 6; 7. Tato soustava se rovněž používá především v číslicové technice, ale je používána lidmi, protože ve dvojkové soustavě se lidem špatně pracuje.

2.3.d) Poziční desítková soustava

Obsahuje tyto znaky pro číslice 0; 1; 2; 3; 4; 5; 6; 7; 8; 9. Desítková soustava je používána lidmi, protože je lidem nejbližší a vžitá.

2.3.e) Poziční šestnáctková soustava

Obsahuje tyto znaky pro číslice 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; A; B; C; D; E; F. Jinak o ní platí totéž co platí o osmičkové číselné soustavě.

2.4. Polyadické soustavy s relativními číslicemi

Lze použít pouze na z > 2.

pro z = 3: $-\frac{z-1}{2},-\frac{z-1}{2}+1,\ldots 0,\ldots,+\frac{z-1}{2}$ [1] Rovnice 2.6

pro z = sudé a z > 3: $-\frac{z}{2},-\frac{z}{2}+1,\ldots,0,\ldots,+\frac{z}{2}$ [2] Rovnice 2.7

což například znamená, že číslo 15 lze v desítkové soustavě vyjádřit dvěma způsoby:

jako +1 +5 nebo jako +2 -5 (nejednoznačnost vyjádření čísel). Stejně tak lze zavést redundantní počet číslic – například u desítkové soustavy se zavede číslice 10, takže číslo 20 lze napsat jako 20 nebo jako (1)(10).

2.4.a) Výhody relativních číslic:

- jednotným způsobem se vyjadřují jak čísla nezáporná, tak i záporná, z čehož vyplývá jednotný algoritmus výpočtu
- znaménko čísla je určeno znaménkem nejvýše (vlevo) postavené nenulové číslice
- znaménko čísla se obrátí obrácením znamének všech jeho číslic

2.5. Polyadické soustavy o několika základech

Jedná se například o časovou soustavu. Problémy se řeší vhodným programováním.

$A = a_n z_n z_{n-1} \cdots z_1 + \ldots + a_i z_i \cdots z_1 + \ldots + a_1 z_1 + a_0$ Rovnice 2.8

kde

---

[1] Pro z=3 se jedná o třístavový (trojhodnotový) systém: ano, ne, nevím.

[2] Tzn. že desítková soustava má tyto číslice:-5,-4,-3-2-1,0,1,2,3,4,5

$a_i = 0,1,\ldots,z_{i+1}-1$

3. PŘEVODY MEZI ČÍSELNÝMI SOUSTAVAMI

3.1. Z jakékoli do desítkové

3.1.a) Ruční způsob

$$A_z = a_i z^i + a_{i-1} z^{i-1} + \ldots + a_1 z^1 + a_0 z^0 \qquad \text{Rovnice 3.1}$$

kde: A = převáděné číslo; z je základ převáděné číselné soustavy; i je řád

3.1.b) Strojový způsob převodu dvojkového čísla s řádovou čárkou před první platnou číslicí

Dvojkové číslo se převede do dvojkově kódované desítkové soustavy (např. kód 8421). Z tohoto kódu se poté převede číslo do kódu 1 z 10 pomocí dekodéru.

Vlastní převod dvojkového čísla do kódu 8421 probíhá takto:

Číslo v dvojkové soustavě s řádovou čárkou před první platnou číslicí se postupně násobí deseti (ve skutečnosti se násobí zlomkem 10/16 a potom násobíme výsledek 16). Při každém násobení vznikne před řádovou čárkou obraz desítkové číslice převodu v kódu 8421, počínaje první číslicí za řádovou čárkou. Násobení čísla zlomkem 10/16 se rozkládá na tři části:

1. násobení čísla 8/16=1/2, což se provede posuvem čísla o jedno místo vpravo
2. násobení čísla zlomkem 2/16=1/8, což se provede posuvem o tři místa vpravo
3. sečtení takto vzniklých násobků

Příklad

Převeďte číslo 0,011 (=3/8=0,375) do desítkové soustavy

posuv o jedno místo vpravo	0,001100
posuv o tři místa vpravo	0,000011
součet	0,001111
posuv o čtyři místa vlevo	0011,110000
posuv o jedno místo vpravo	0,011000
posuv o tři místa vpravo	0,000110
součet	0,011110
posuv o čtyři místa vlevo	0111,100000
posuv o jedno místo vpravo	0,010000
posuv o tři místa vpravo	0,000100
součet	0,010100
posuv o čtyři místa vlevo	0101,000000

3.2. Z desítkové do soustav s jiným základem[3]

Nejčastěji se převádí do soustavy dvojkové, osmičkové či šestnáctkové. Ale uvedený algoritmus platí pro všechny polyadické soustavy. Přesnost zobrazení závisí na počtu výsledných řádů.

3.2.a) Ruční způsob

(i) Celé číslo

1. Metoda postupného dělení pro přechod ze základu D na B: původní číslo se rozkládá odečítáním zmenšujících se mocnin základu B, přičemž se hledá mocnina čísla B rovná převáděnému číslu nebo menší.
2. Desítkové číslo dělíme základem soustavy, do které je číslo převáděno a zbytky po dělení tvoří obraz čísla v dané soustavě počínaje nejnižší platnou číslicí.

$$Y_z = \left(\frac{Q_{n-1}}{z}\right) + \ldots + \left(\frac{Q}{z}\right), \text{ kde } \frac{Q_{n-1}}{z} = Q_n + R_n \qquad \text{Rovnice 3.2}$$

z = základ soustavy; Q je desítkové převáděné číslo

Číslo s řádovou čárkou před první platnou číslicí

Desítkové číslo postupně násobíme základem soustavy, do které je číslo převáděno. Obraz čísla v dané soustavě počínaje nejvyšší platnou číslicí vpravo od řádové čárky tvoří číslice přenosu, které se objeví vlevo od řádové čárky. Přesnost čísla pak záleží na počtu řádů.

3.2.b) Strojový způsob převodu celého čísla do dvojkové soustavy

Desítkové číslo v kódu 1 z 10 se pomocí dekodéru převede do dvojkově kódované desítkové soustavy (např. do kódu 8421). Pak se nejvyšší desítková číslice v kódu 8421 umístí jako čtveřice nejnižších řádů dvojkového obrazu převáděného čísla. Ta se násobí deseti (ve dvojkové soustavě) a přičte se další nižší číslice převáděného čísla.

Příklad

Převeďte číslo 325 do dvojkové soustavy

přičíst tři	0011
násobit deseti	11110
přičíst dvě	100000
násobit deseti	101000000
přičíst pět a je tu výsledek	101000101

[3] Pokud chcete převádět z jiné než desítkové soustavy pak musíte dělitel převézt do dané soustavy např: 1357_8 do šestnáctkové ($16_{10} = 20_8$): $1357_8 : 20_8$

3.3. Z dvojkové do osmičkové a naopak

Dvojkové číslo rozdělíme vpravo a vlevo od řádové čárky na trojmístné skupiny (triády) a každou skupinu vyjádříme osmičkovou číslicí (0 až 7).

Do dvojkové z osmičkové se převádí jednoduchým vyjádřením každé osmičkové číslice dvojkovým číslem o třech bitech.

3.4. Z dvojkové do šestnáctkové a naopak

Dvojkové číslo rozdělíme vpravo a vlevo od řádové čárky na čtyřmístné skupiny (tetrády) a každou skupinu vyjádříme šestnáctkovou číslicí.

Do dvojkové ze šestnáctkové se převádí jednoduchým vyjádřením každé šestnáctkové číslice dvojkovým číslem o čtyřech bitech.

3.5. Váhové kódy

Nejčastější metoda pro tvorbu kódu desítkových číslic je v tom, že každému místu v kombinaci dvojkových číslic přiřadíme určitou **váhu** – pak se jedná o tzv. poziční číselnou soustavu. Celková hodnota kódované desítkové číslice je pak daná součtem těch vah, na jejichž místě jsou jedničky.

$a_i = v_i \cdot x_i$, kde v_i je váha itého členu; x_i je itý člen kódu (0 nebo 1)

tabulka 1 Příklady nejpoužívanějších váhových kódů desítkových číslic

D	BCD (8421)	BCD+3	(84-2-1)
0	0000	0011	0000
1	0001	0100	0111
2	0010	0101	0110
3	0011	0110	0101
4	0100	0111	0100
5	0101	1000	1011
6	0110	1001	1010
7	0111	1010	1001
8	1000	1011	1000
9	1001	1100	1111

3.6. Doplňkový kód[4]

Doplňkový kód číslice je takový kód, u něhož invertováním dvojkových číslic (tj. záměnou nul za jedničky a naopak) získáme doplněk původní číslice do devíti. Doplňkové kódy jsou důležité při počítání se zápornými čísly.

Inverzní vlastnost kódů desítkových číslic (tvoření devítkových doplňků) je výhodná při sčítání desítkových číslic. Sečteme-li totiž odpovídající bity dvou číslic, jejichž součet je větší než deset, vznikne samočinně přenos do vyššího řádu.

3.7. Neváhové kódy

Jsou to kódy, u nichž nelze určit hodnotu pomocí vzorce pro váhové kódy. Právě tyto kódy se používají pro bezpečnostní a samoopravné kódy. Některé z těchto kódů mají sice váhy (lze je vytvářet různě), ale výsledek není pro všechny kombinace váhový!

V následující tabulce uvádím nejběžnější neváhové kódy. Jako vzor je uveden i 2z5, který patří mezi mnoho kódů *pzn.*

tabulka 2 Některé neváhové kódy

D	2z5 (vždy dvě jedničky)	Gray ův kód	Gray ův kód+ 3	Aiken ův kód (2421)
0	01010	0000	0010	0000
1	01100	0001	0110	0001
2	00011	0011	0111	0010
3	01001	0010	0101	0011
4	00110	0110	0100	0100
5	00101	0111	1100	1011
6	10001	0101	1101	1100
7	10100	0100	1111	1101
8	10010	1100	1110	1110
9	11000	1101	1010	1111

[4] Je zajímavé, že vlastnost kódu tvořit doplňkový kód pouhým invertováním dvojkových číslic existuje tehdy, když každá číslice a její doplněk jsou v tabulce kombinací (tabulka 1) uloženy souměrně podle středu tabulky.

3.8. Kontrola přenosu dat

Pro správné a bezchybné zpracování informací v počítači je důležité kontrolovat správnost přenosu. Ta se zajišťuje použitím redundantních (nadbytečných) kódů. Podle počtu nadbytečných bitů pak vznikají bezpečnostní nebo samoopravné kódy.

Při studiu vlastností a sestavování bezpečnostních kódů je výhodné zobrazit příslušné kódové kombinace *n*bitového kódu graficky na tzv. **jednotkové n-rozměrné krychli nazývané též Hammingův prostor**, která má jednotkovou vzdálenost mezi vrcholy (pohybujeme-li se po hraně krychle). **Kódové kombinace** chápeme jako souřadnice jednotlivých vrcholů krychle (nepoužíváme však samozřejmě všechny vrcholy). **Kódová vzdálenost** je nejmenší vzdálenost mezi využitými vrcholy, tj. počet hran, po kterých musíme přejít mezi využitými vrcholy.

3.8.a) Bezpečnostní kódy

Bezpečnostní kódy mohou chybu indikovat (kódy zjišťující chybu), nebo indikovat a opravit (samoopravné kódy)

Kódy zjišťující chybu

Kódy zjišťující chybu jsou takové, které umožňují pouze zjistit chybu při přenosu. Nelze však už zjistit, který bit je chybný.

Aby byl kód zjišťující chybu s možností zjištění jedné chyby musí být kódová vzdálenost dvě. Obecně platí, že pro zjištění *k* chyb je třeba aby kódová vzdálenost byla *k+1*.

Příklady vytváření kódů zjišťujících chybu:

- kód 2 z 5: každá desítková číslice obsahuje dvě jedničky
- parita sudá: přidáme paritní bit, tak aby počet jedniček byl sudý
- parita lichá: přidáme paritní bit, tak aby počet jedniček byl lichý

Samoopravné kódy

Samoopravné kódy jsou takové kódy, které nejenže zjistí, že došlo k chybě při přenosu, ale také umožňují opravu alespoň jedné chyby.

Aby byl kód samoopravný platí, že pro opravu *k* chyb je třeba aby kódová vzdálenost byla *2k+1*.

Příklady vytváření samoopravných kódů

- vytvoření příčné parity pro řádky a podélné parity pro sloupce. Chyba je v průsečíku.
- součtem modulo
- Hammingovy kódy

3.9. Kódy se změnou v jednom řádu

U těchto kódů požadujeme, aby kterékoli dvě kódové kombinace se vzájemně lišily pouze v jednom bitu. Tento druh kódů lze také vyšetřovat na *n*-rozměrné jednotkové krychli. Čára spojující vrcholy vede po hranách krychle a každým vrcholem prochází jen jednou. Lze sestrojit velké množství těchto kódů, jako příklad se uvádí například Grayův kód.

4. DVOJKOVÁ SOUSTAVA

4.1. Organizace dat a kódování informací v počítači

Nejprve je třeba si uvědomit, že při počítání s čísly vlastně pracujeme s jejich reálnou reprezentací. Počítání je vlastně nejvyšší forma abstrakce.

4.1.a) Možné definice kódování

1. Kód je přepis informací (čísel, písmen a jiných znaků či symbolů dané abecedy) pomocí kódových kombinací podle určitých pravidel.
2. Předpis, jak k sobě přiřadit prvky dvou množin

Fyzikálním nositelem informace je signál. Dnes nejobvyklejší forma signálu je dvoustavový signál (logická 0 a logická 1 – ale samozřejmě by šlo nahradit tyto dvě hodnoty libovolnými jinými symboly). Proto se používá dvojková soustava (přesněji řečeno dvojkově kódovaná desítková soustava) ke znázorňování čísel v počítačích.

Počet různých informací (laicky řečeno číslic) uložitelných ve slově o *n* bitech je dán výrazem 2^n. Je tedy konečný a omezený délkou slova.

4.1.b) Kódování desítkových číslic v počítačích – dekadické uložení

Protože počítače pracují ve dvojkové soustavě je potřeba převádět kód z většinou desítkové soustavy do dvojkové a naopak. Desítkové číslice můžeme vyjádřit různými kombinacemi čtyř nebo více dvojkových číslic. Pro zobrazení desítkového čísla potřebujeme alespoň čtyřbitový kód – 2^4 , který umožňuje šestnáct kombinací (což dostačuje pro deset číslic). Pro zobrazení *n* místného kódovaného desítkového čísla potřebujeme *4n* bitů, což je asi o 20 % více než pro vyjádření téhož čísla ve dvojkové soustavě.

4.1.c) Způsob interpretace slova v počítačích

Pro jednoduchost můžeme znázorňovat obecnou hodnotu dvojkového čísla bitovou grafickou mřížkou:

a_n	a_{n-1}					a_2	a_1

nebo její zjednodušenou fomou:

$[a_n a_{n-1} \ldots\ldots a_2 a_1]$, kde můžete oddělovat skupiny číslic mezerami pro lepší čtení.

Nejčastěji se dvojkové číslo rozdělí na trojice (čtveřice) bitů a každá tato skupina se převede do osmičkové (šestnáctkové) soustavy.

Co však dané dvojkové číslo vyjadřuje lze zodpovědět pouze na základě přesné znalosti toho, jak interpretovat jeho obsah.

4.2. Zobrazení alfanumerických znaků

Musíme si stanovit nějakou abecedu – uspořádanou množinu přípustných znaků. Tyto abecedy se nazývají tabulky znaků a obsahují znaky abecedy, mezeru, číslice a

speciální znaky. Každý znak je určen jednoznačně svým číselným kódem a jeho pořadí je určeno pořadovým číslem v tabulce znaků. Jsou dva typy:

1. Osmibitové vyjádření – slabika – pro 256 kombinací (ASCII tabulka). Většina kódových tabulek má prvních 128 znaků společných, dalších 128 znaků bývá rozmístěno podle národních zvyklostí.
2. Šestnáctibitové vyjádření. Protože osmibitové vyjádření nevyhovovalo, bylo Microsoftem navrženo znázorňování pomocí 16bitového čísla. Tím se vlastně do jedné kódové tabulky vejdou znaky všech národních abeced (65536 kombinací!).

Pro tabulky znaků platí některá pravidla, jež musí být dodržována ve všech kódových tabulkách:

- prvních 31 znaků jsou řídící
- mezera je nejmenší znak
- 0 < 1 … < 9
- A < B … < Z

4.3. Zobrazení logických hodnot

Logickými údaji rozumíme takové údaje, jež nabývají pouze dvou hodnot 1 (true, ano, on) nebo 0 (false, ne, off). Tady však dochází k plýtvání místem, protože k znázornění tohoto dvouhodnotového údaje stačí jeden bit, ovšem k ukládání údajů v počítači je nejmenší jednotka slabika. Logická operace se tedy provádí se všemi bity slabiky.

Pokud je potřeba pracovat s jednotlivými bity slabiky, pak používáme tzv. masku pro extrahování určitého bitu slabiky.

4.4. Zobrazení celých čísel

Počet míst ve dvojkově zobrazeném čísle v počítači je pevně stanoven, a tak určuje konečnost množiny zobrazitelných celých čísel. Pokud dojde k operaci, při které výsledek přesáhne zobrazitelný rozsah nazýváme to přetečením (overflow).
K překonání těchto problémů se používá několik triků:

1. lepší vyjádření vzorce např.: místo (ab)/c je lepší a(b/c)
2. používá se místo jednoho slova slov dvou – a to nejen pro výsledek, ale i pro operandy
3. nebo počítání v reálných číslech

V počítačové celočíselné aritmetice platí komutativní zákony pro sčítání a násobení, ale kvůli možnosti přeplnění neplatí asociativní ani distributivní zákon.

4.5. Zobrazení nezáporných celých čísel

Nezáporné celé číslo můžeme vyjádřit:

$$A = \sum_{i=1}^{n} a_i \cdot z^{i-1}$$ Rovnice 4.1

Nejmenší zobrazitelné číslo je 0 a největší zobrazitelné číslo je z^n-1. Zobrazitelných je tedy z^n čísel (hodnot) z intervalu <0, z^n-1>.

4.6. Zobrazení záporných celých čísel

4.6.a) Kód s posunutou nulou o K

$$A = \sum_{i=1}^{n} a_i \cdot z^{i-1} - K$$ Rovnice 4.2

Můžeme zobrazit libovolné celé číslo z intervalu <-K, z^n-1-K>. Abychom získali téměř shodný počet kladných i záporných čísel volíme K=z^{n-1} (rozdělíme interval na dvě poloviny). Po dosazení tohoto K získáme interval <-z^{n-1}, z^{n-1}-1> a po úpravě 4.2 nám vznikne:

$$A = (a_{n-1}) \cdot z^{n-1} + \sum_{i=1}^{n-1} a_i \cdot z^{i-1}$$ Rovnice 4.3

4.6.b) Přímý kód

Záporné číslo se zobrazí v absolutní hodnotě a se znaménkem. Znaménko se zobrazuje v tzv. *znaménkovém bitu* (0 = kladné; 1 = záporné).

$$A = (1 - z\, a_n) \sum_{i=1}^{n-1} a_i \cdot z^{i-1}$$ Rovnice 4.4

nebo stručněji

$$A = \pm \sum_{i=1}^{n-1} a_i \cdot z^{i-1}$$ Rovnice 4.5

Rozsah zobrazitelných čísel je dán intervalem <1-2^{n-1}, 2^{n-1}-1>, který vykazuje naprostou symetrii v počtu zobrazitelných záporných i kladných čísel, ale obsahuje 2^n-1 čísel – a to proto, že nula může být znázorněna dvěma způsoby: [00…00] nebo [10…00], tedy jako kladná i záporná. Tento kód se nepoužívá v počítačích, protože by se musela zavést mimo sčítačky i odčítačka z důvodů komplikované logiky sčítání a odčítání.

4.6.c) Nepřímý kód

Nepřímý kód má dvě varianty – inverzní a doplňkový kód – jež se liší pouze způsobem zobrazení záporných čísel. Pro nezáporná čísla je zobrazení shodné pro tyto obě varianty i pro přímý kód. První bit vlevo určuje u obou variant znaménko čísla – stejně jako u přímého kódu.

(i) Varianta 1 - doplňkový kód (používaný v počítačích)

Doplňkový kód čísla

$$A = \sum_{i=1}^{n-1} a_i \cdot z^{i-1} - a_n \cdot z^{n-1}$$ Rovnice 4.6

zjednodušeně

$$(-a) = z^n - a$$ Rovnice 4.7

kde z = základ; *n* = řády

Kód dvojkového záporného čísla vytvoříme takto:

1. číslo v jeho absolutní hodnotě vyjádříme zápisem o n-1 číslicích
2. na znaménkovém místě vlevo od řádové čárky napíšeme 0
3. u čísla invertujeme číslice (změníme 0 na 1 a naopak)
4. k poslednímu nejnižšímu řádu přičteme jedničku a provedeme nezbytné přenosy
5. nastane-li přenos do znaménkového místa, počítáme s ním jako s dvojkovou číslicí. Pokud vzniknou dvě číslice před řádovou čárkou, pak se první zleva zahodí.

Je třeba dbát, aby součet byl ve své absolutní hodnotě menší než jedna a nenastalo přetečení. Je-li výsledek záporný, pak je samozřejmě znázorněn v doplňkovém kódu. Nezáporná čísla se v něm zobrazí beze změny.

(ii) Varianta 2 - inverzní kód[5]

Inverzní kód čísla a je

$$A = \sum_{i=1}^{n-1} a_i \cdot z^{i-1} - a_n \left(z^{n-1} - 1\right)$$ Rovnice 4.8

zjednodušeně

$$(-a)' = z^n - 1 - a$$ Rovnice 4.9

Kód dvojkového záporného čísla vytvoříme takto:

1. číslo v jeho absolutní hodnotě vyjádříme zápisem o n-1 číslicích
2. na znaménkovém místě vlevo od řádové čárky napíšeme 0
3. u čísla invertujeme číslice (změníme 0 na 1 a naopak)
4. nastane-li přenos do znaménkového místa, neztrácí se, jako u doplňkového kódu, ale přičítá se k poslednímu řádu čísla vpravo (tzv. kruhový přenos).

Rozdíl proti doplňkovému kódu je tedy v tom, že se nepřičítá jednička po inverzi a tak je možno vytvořit inverzní kód jednoduššími obvody. Inverzní kód je podobný kódu přímému – nula může být znázorněna dvěma způsoby: [00…00] nebo [11…11], tedy jako kladná i záporná.

Je třeba dbát, aby součet byl ve své absolutní hodnotě menší než jedna a nenastalo přetečení. Je-li výsledek záporný, pak je samozřejmě znázorněn v doplňkovém kódu.

tabulka 3 Stručný přehled rozsahů zobrazitelných čísel u jednotlivých kódů

kód	min	max	rozsah pro n=32
bez znaménka	0	2^n-1	<0, 4 294 967 295>
posunutá nula	-2^{n-1}	$2^{n-1}-1$	<-2 147 483 148, 2 147 483 147>
přímý kód	$-(2^{n-1}-1)$	$2^{n-1}-1$	<-2 147 483 147, 2 147 483 147>

[5] Nula může být znázorněna jako kladné číslo samými nulami, nebo jako záporné číslo samými jedničkami. Obojí způsob však vede ke správnému výsledku.

inverzní kód	$-(2^{n-1}-1)$	$2^{n-1}-1$	<-2 147 483 147, 2 147 483 147>
doplňkový kód	-2^{n-1}	$2^{n-1}-1$	<-2 147 483 148, 2 147 483 147>

4.7. Zobrazení reálných čísel

Z hlediska zobrazení na počítači nemá smysl rozlišovat racionální a reálná čísla – nepracujeme s ničím jiným než s racionálními aproximacemi reálných čísel. Iracionální čísla totiž nelze vyjádřit konečným zápisem v žádné polyadické soustavě a tak buď je aproximujeme nebo označujeme dohodnutými symboly (π, e, apod.) nebo používáme obecně platných vlastností reálných funkcí (cos π=1, e^0=1 apod.).

4.7.a) Zobrazení necelých reálných čísel

Necelá část desetinného čísla se znázorňuje stejně jako celé číslo, kde nejvyšší řád a_n je znaménkový a ostatní řády již určují vlastní číslo. Uložitelná část čísla nemůže tedy přesáhnout hodnotu 1.

$[a_n, a_{n-1|} \ldots| a_1]$

Takto lze znázornit tuto množinu racionálních čísel:

$Q = \{k2^{-(n-1)} : k=1-2^{(n-1)}, 2-2^{(n-1)}, \ldots, -1, 0, 1, \ldots, 2^{(n-1)}-2, 2^{(n-1)}\}$ Rovnice 4.10

kde

základní krok $\varepsilon = 2^{-(n-1)}$

$k \in \langle 1-2^{(n-1)}, 2^{(n-1)}-1\rangle$

tedy

$Q = \{k \cdot 2^{-(n-1)} : k \in I\}$ Rovnice 4.11

Q je množina uložitelných (závisí na kroku) racionálních čísel v daném kódu

I je množina zobrazitelných celých čísel

Z výše uvedeného vyplývá, že z reálného intervalu (-1,1) lze zobrazit jen prvky množiny Q. Tuto situaci řešíme tak, že každé číslo z množiny Q považujeme za reprezentanta (aproximaci) reálných čísel z určité podmnožiny. Vlastní aproximace se provádí různými způsoby

Podmnožinu R reálných čísel, pro která je definována aproximace čísly množiny Q nazýváme zobrazitelnými reálnými čísly v daném kódu s touto aproximací.

Z výše uvedeného vyplývá, že zobrazitelná čísla splývají s množinou R a uložitelná čísla splývají s množinou Q. Ovšem nezapomínejte, že R je určeno také způsobem aproximace!

Zobrazení čísel se skládá ze dvou částí:

- aproximace reálného čísla x nějakým racionálním uložitelným číslem y ∈ Q
- vlastní zakódování čísla y bitovou mřížkou

V počítačové reálné aritmetice platí komutativní zákony pro sčítání a násobení, ale kvůli možnosti přeplnění neplatí asociativní ani distributivní zákon.

4.7.b) Zobrazení v pevné řádové čárce

Pro aritmetiky s pevnou řádovou čárkou platí předpoklad stejného umístění řádové čárky. Také se počítá na povolený počet platných míst. Protože však čísla mají ve skutečnosti umístěnu řádovou čárku různě, pak se musí převádět u operací sčítání a odčítání do stejného tvaru.

Umístíme-li myšlenou řádovou čárku před q-té místo zprava, bude pro množinu uložitelných čísel platit:

$$Q = \{k \cdot 2^{-q} : k \in I\}$$ Rovnice 4.12

kde velikost základního kroku v mřížce je $\varepsilon = 2^{-m}$.

Tento způsob vyjádření reálných čísel je velmi pracný (převod čísel na číslo s řádovou čárkou na stejném místě), takže se ve skutečnosti nepoužívá.

4.7.c) Zobrazení s pohyblivou řádovou čárkou – semilogaritmický zápis

Zobrazení v pohyblivé řádové čárce odstraňuje problémy s délkou čísel zobrazených v pevné řádové čárce.

Nejjednodušší způsob a zároveň také nejpraktičtější je rozdělit číslo na mantisu a exponent (tak jak se to provádí v inženýrských výpočtech – tzv. semilogaritmický tvar např.: $8{,}7 \cdot 10^5$) do uspořádané dvojice čísel (m, e).

$$A = m \bullet z^e$$ Rovnice 4.13

kde m je mantisa, e je celé číslo – tzv. exponent a z je základ soustavy.

Protože stejné číslo lze vyjádřit různými způsoby, klade se na hodnotu mantisy následující omezení:

$$z^{-1} \leq |m| < 1$$ Rovnice 4.14

tzv. normalizovanost mantisy.

Mantisa se ukládá jako pravý zlomek, exponent je nejčastěji uložen jako celé číslo s posunutou nulou (v tomto pořadí). První bit mantisy vždy určuje znaménko čísla.

Mantisa i exponent se společně ukládají do jednoho slova, které je určitým způsobem rozděleno. To je jednoduchá přesnost. Pro dvojnásobnou přesnost se používá slov dvou (slovo navíc se použije pro mantisu).

V případě semilogaritmického tvaru může často docházet k podtečení.

Protože mantisa má pevně stanovený počet míst, a navíc při převodu na stejné exponenty dochází ke zmenšení přesnosti, tak neplatí ani asociativní ani distributivní zákon.
Čísla necelá s malým počtem desetinných míst (0,1; 0,2; atp.) mají neukončený desetinný rozvoj ve dvojkovém tvaru a tak dochází k jejich aproximaci, čímž se výpočet zatíží chybou již na začátku. Snažte se tedy omezit počet kroků výpočtu.

Při sčítání či odčítání s čísly v pohyblivé řádové čárce se musí nejprve sjednotit exponenty. Poté se počítá stejně jako v případě operací s pevnou řádovou čárkou. Výsledek se upraví posuvem mantisy vlevo tak, aby za řádovou čárkou byla první platná číslice. K exponentu se potom přičte číslo, které vyjadřuje počet míst, o kolik se posunula mantisa. Tomuto se říká normalizace čísla do semilogaritmického tvaru.

Při násobení se exponenty čísel sečtou a mantisy se znásobí.

Při dělení se exponenty čísel odčítají a mantisy se dělí. Je třeba aby mantisa dělitele byla větší než mantisa dělence.

4.8. Číselné operace ve dvojkové soustavě

Protože je poměrně složité přímo odčítat čísla ve dvojkové soustavě, tak se provádí zobrazení záporných čísel tak, aby bylo možné převést odčítání na sčítání.

Základní číselné operace, jako je sčítání a odčítání, se provádí *sčítačkou* a základní operace násobení a dělení se provádí pomocí sčítačky a posuvů.

Složitější operace se provádí pomocí algoritmů, které opět používají jen sčítání a posuv.

4.8.a) Sčítání

Při operaci sčítání musí mít součet rezervován o jeden řád více než sčítanci. Ve skutečnosti u počítačů se vyhrazuje součtu stejné místo jako sčítancům.

U celočíselné aritmetiky a reálné aritmetiky necelých čísel může dojít k přeplnění.

tabulka 4 Sčítání dvojkových číslic včetně přenosů

A_i	B_i	P_{i-1}	S	P
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

4.8.b) Odečítání

Odečítání se pomocí inverzního nebo doplňkového kódu převádí na sčítání.

4.8.c) Násobení

Při násobení dvou *n*místných čísel dostaneme součin, který má *2n* míst. Násobení se provádí stejně jako u desítkové soustavy postupnými součty a posuvy. Znaménko součinu se dostane součtem znaménkových míst násobence a násobitele na jednomístné sčítačce.

U celočíselné aritmetiky může dojít k přeplnění. U reálné aritmetiky necelých čísel k přeplnění nedochází, avšak dochází ke ztrátě přesnosti aproximací.

4.8.d) Dělení

U dělení může dojít ke třem případům, a to jak u celočíselné aritmetiky, tak u reálné aritmetiky necelých čísel:

1. výsledek je uložitelný, což bývá málo pravděpodobné
2. výsledek je zobrazitelný, ale ne uložitelný. V tomto případě se aproximuje
3. výsledek není zobrazitelný, pak se jedná o přeplnění.

U dělení je situace složitější: například co bude výsledkem celočíselného podílu čísel 5a -2? Proto se zavádí dvě důležité operace (b≠0):

(i) div – operace celočíselného dělení:

$$a\,div\,b = \mathrm{sgn}\left(\frac{a}{b}\right)\cdot\left\|\frac{a}{b}\right\| \qquad \text{Rovnice 4.15}$$

(ii) mod – operace zbytku po dělení celých čísel:

$$a \bmod b = a - (a\,div\,b)\cdot b \qquad \text{Rovnice 4.16}$$

a pro necelá reálná čísla

$$a \bmod b = (a - (a\,div\,b)\cdot b)\cdot 2^{n-1} \qquad \text{Rovnice 4.17}$$

kde funkce sgn(x) je definána následovně:

pro x > 0 je 1; pro x =0 je 0; pro x < 0 je -1

1. Pomocí zvláštního podprogramu
2. převod A : B na A . 1/B
3. při dělení odčítáme dělitele od dělence a po každém odčítání posuneme dělence o jedno místo vlevo. Když dostaneme kladný rozdíl, je číslice podílu 1 počínaje první číslicí vpravo od řádové čárky. V odčítání pokračujeme tak dlouho, až dostaneme záporný rozdíl a potom číslice podílu je 0. V následujícím kroku dělitele přičítáme a dostaneme-li kladný výsledek, znovu dělitele odčítáme. Nulu při odčítání považujeme za kladný výsledek. Znaménko podílu určíme odečtením znaménka dělitele od znaménka dělence. Musí se provádět pro čísla s řádovou čárkou před první platnou číslicí a vyžaduje se aby dělenec i dělitel byl menší než jedna.

4.9. Operace ve dvojkově kódované desítkové soustavě

4.9.a) sčítání

Pokud si zkusíte sčítat čísla v dvojkově kódované desítkové soustavě, pak zjistíte, že výsledek neodpovídá skutečnosti. Musí se provádět korekce součtu pro součet větší než 9:

- u součtů v rozmezí 10 – 15 je třeba od výsledku odečíst 10 a současně vytvořit desítkový přenos do vyššího řádu.
- u součtů v rozmezí 16 – 19 je třeba kromě vytvoření desítkového přenosu do vyššího řádu přičíst k součtu šest.

V počítači se provádí pomocí korekčních obvodů.

4.9.b) odčítání

Používá se inverzní vlastnost desítkového kódu (pokud kód nemá tuto vlastnost – např. 8421, pak se vytváří speciální obvod, který devítkový doplněk vytvoří). Záporné číslo se tedy zobrazí pomocí devítkového doplňku. Znaménko plus se označuje nulou a znaménko minus se označuje devítkou. Při sčítání u přenosů na nejvyšším řádu se provádí kruhový přenos.

4.9.c) násobení

Násobení se provádí stejně jako u násobení u dvojkové soustavy.

4.9.d) dělení

Odčítání se provádí stejně jako u odčítání ve dvojkové soustavě, ale záporná čísla se zobrazují v devítkovém doplňku.

5. LITERATURA, ZDROJE

1. J. Kolář, K. Müller, F. Plášil: Počítače a programování. Skriptum ČVUT FEL, Praha, 1983
2. K. Krištoufek, K. Kabeš: Stroje na zpracování informací. SNTL, Praha, 1975
3. J. Blatný, K. Krištoufek, Z. Pokorný, J. Kolenička: Číslicové počítače. SNTL, Praha, 1982
4. L. Gvozdjak, L. Hanulová, A. Jankovičová, L. Molnár: Počítače a programovanie. ALFA, Bratislava, 1985