Číselné soustavy

Mikroprocesorová technika a embedded systémy

doc. Ing. Tomáš Frýza, Ph.D.

září 2018

Obsah přednášky

Obsah přednášky

Používané číselné soustavy

- Dekadická (desítková) soustava:
  - **základ** soustavy: 10
  - využívá **symboly**: 0, 1, 2, . . ., 9
  - **označení** soustavy pomocí sufixu $1\,100_{10}$
  - kladný celočíselný **rozsah** hodnot: 0 až $10^n - 1$, kde n je počet použitých symbolů
- Hexadecimální (šestnáctková) soustava se především využívá pro přehlednější zápis binárních čísel:
  - **základ** soustavy: 16
  - využívá **symboly**: 0, 1, 2, . . ., 9, a, b, . . ., f
  - **označení** soustavy pomocí sufixu $2ba6_{16}$, 4f6H, prefixů 0xd2, $25a4
  - kladný **rozsah** hodnot: 0 až $16^n - 1$
- Binární (dvojková) soustava se používá v číslicové a mikroprocesorové technice k reprezentaci hodnot:
  - **základ** soustavy: 2
  - využívá **symboly**: 0, 1
  - **označení** soustavy pomocí sufixu 101_2, nebo prefixu 0b1010
  - kladný **rozsah** hodnot: 0 až $2^n - 1$
- *Pozn.: Osmičková (oktalová) soustava:*
  - **základ** *soustavy: 8*
  - *využívá* **symboly***: 0, 1, . . ., 7*
  - **označení** *soustavy pomocí sufixu* 1100_8
  - *kladný* **rozsah** *hodnot: 0 až* $8^n - 1$

Binární formáty s řádovou čárkou

- V binární soustavě jsou hodnoty reprezentovány ve tvaru s tzv. pevnou řádovou čárkou (Fixed-point), tj. pozice binární čárky je pevně dána a je neměnná:
  - celá čísla neznaménková (unsigned, řádová čárka úplně vpravo)
  - celá čísla znaménková (signed)
  - zlomkový tvar (fractional)
  - ostatní formáty (definuji si pozici tečky libovolně)
- nebo s plovoucí řádovou čárkou (Floating-point, IEEE 754):
  - jednoduchá/základní přesnost (single precision)
  - dvojitá přesnost (double precision)
  - rozšířená přesnost (extended precision)

- Pro 8bitovou reprezentaci uvažujme $\mathbf{d} = (d_7, d_6, \ldots, d_1, d_0)$:
  - bit d_7 označuje *most significant bit* (MSB)
  - bit d_0 označuje *least significant bit* (LSB)
- Každá pozice má příslušnou váhu (mocninu dvou) v závislosti na pozici od řádové čárky: "jednotky": 2^0, "desítky": 2^1, ...

Example

Převod kladného čísla $1011{,}101_2$ rozkladem dle příslušných mocnin.

Řešení

$(1{\cdot}2^3) + (0{\cdot}2^2) + (1{\cdot}2^1) + (1{\cdot}2^0) + (1{\cdot}2^{-1}) + (0{\cdot}2^{-2}) + (1{\cdot}2^{-3}) = 8 + 0 + 2 + 1 + 0{,}5 + 0 + 0{,}125 = 11{,}625_{10}$

Převod mezi číselnými soustavami

Table: Převod hodnot mezi dekadickou, binární a hexadecimální soustavou

Dekadická	Hexadecimální	Binární
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001

Dekadická	Hexadecimální	Binární
10	a	1010
11	b	1011
12	c	1100
13	d	1101
14	e	1110
15	f	1111
16	10	10000
17	11	10001
18	12	10010
...	...	...

- Pro převod hodnoty ze soustavy se základem X do jiné se základem Y lze použít metody postupného dělení (continuous dividing) a násobení (continuous multiplying):
  - dělení hodnot **nalevo** od řádové čárky hodnotou Y
  - násobení hodnot **napravo** od řádové čárky hodnotou Y

Převod celých čísel z dekadické do binární soustavy

- *Pro převod hodnoty ze soustavy se základem X do jiné se základem Y lze použít metody postupného dělení (continuous dividing) a násobení (continuous multiplying):*
  - *dělení hodnot* **nalevo** *od řádové čárky hodnotou Y*
  - *násobení hodnot* **napravo** *od řádové čárky hodnotou Y*

Example

Převeďte dekadickou hodnotu 25 do binární soustavy.

Řešení

Metoda postupného dělení hodnot nalevo od řádové čárky základem binární soustavy.

25	:	2	=	12,	zbytek **1**	2^0	LSB
12	:	2	=	6,	zbytek **0**	2^1	
6	:	2	=	3,	zbytek **0**	2^2	
3	:	2	=	1,	zbytek **1**	2^3	
1	:	2	=	0,	zbytek **1**	2^4	MSB

Výsledek: 11001_2.

Pozn.: Algoritmus je ukončen po dosažení hodnoty 0

Převod desetinné části z dekadické do binární soustavy

- *Pro převod hodnoty ze soustavy se základem X do jiné se základem Y lze použít metody postupného dělení (continuous dividing) a násobení (continuous multiplying):*
  - *dělení hodnot* **nalevo** *od řádové čárky hodnotou Y*
  - *násobení hodnot* **napravo** *od řádové čárky hodnotou Y*

Example

Převeďte dekadickou hodnotu 0,375 do binární soustavy.

Řešení

Metoda postupného násobení hodnot napravo od řádové čárky základem binární soustavy.

$$
\begin{array}{rcrcll}
0{,}375 & \cdot & 2 & = & \mathbf{0},75 & 2^{-1} \quad \text{první bit po řádové čárce} \\
0{,}75 & \cdot & 2 & = & \mathbf{1},5 & 2^{-2} \\
0{,}5 & \cdot & 2 & = & \mathbf{1},0 & 2^{-3} \quad \text{poslední bit}
\end{array}
$$

Výsledek: $0{,}011_2$.

Pozn.: Algoritmus lze ukončit pokud je desetinná část rovna 0, příp. po dosažení požadované bitové přesnosti

Rychlý převod celých dekadických čísel do binární soustavy

- Podstatou je rozklad dekadické hodnoty na mocniny dvou, tj. na váhy jednotlivých bitů

Example

Převeďte dekadickou hodnotu 53 do binární soustavy pomocí rychlé metody.

Řešení

	53	
−	32	tj. 2^5
	21	
−	16	tj. 2^4
	5	
−	4	tj. 2^2
	1	
−	1	tj. 2^0
	0	→ ukončení algoritmu

$2^5 + 2^4 + 2^2 + 2^0 = \mathbf{1}\cdot 2^5 + \mathbf{1}\cdot 2^4 + \mathbf{0}\cdot 2^3 + \mathbf{1}\cdot 2^2 + \mathbf{0}\cdot 2^1 + \mathbf{1}\cdot 2^0$

Výsledek: 110101_2

Vzájemný převod mezi hexadecimální a dekadickou soustavou

- Algoritmus převodu z hexadecimální do dekadické soustavy:
  - Jednotlivé symboly lze váhovat obdobně jako u binární soustavy nebo použít algoritmy postupného dělení/násobení

Example

Rozklad čísla 356_{16}.

Řešení

$(3 \cdot 16^2) + (5 \cdot 16^1) + (6 \cdot 16^0) = 768 + 80 + 6 = 854_{10}$

Example

Rozklad čísla $2af_{16}$

Řešení

$(2 \cdot 16^2) + (10 \cdot 16^1) + (15 \cdot 16^0) = 512 + 160 + 15 = 687_{10}$

Vzájemný převod mezi hex. a dekadickou soustavou

- Algoritmus převodu z dekadické do hexadecimální soustavy:
  - Obdobný postup jako u převodu z dekadické do binární soustavy, tj. rozdělení čísla na celou a "desetinnou" část.

(a) Celá část se dělí základem soustavy (tj. 16) a zbytek po dělení udává hodnoty od "desetinné" čárky směrem k MSB

(b) Desetinná část se násobí základem soustavy a přenos udává hodnotu od "desetinné" čárky směrem k LSB

Example

Převeďte hodnotu $423{,}34_{10}$ do hexadecimální soustavy.

Řešení

(a) Metoda postupného dělení základem hexadecimální soustavy.

423	:	*16*	=	*26,*	*zbytek* **7**	16^0	*LSB*
26	:	*16*	=	*1,*	*zbytek* **10**	16^1	
1	:	*16*	=	*0,*	*zbytek* **1**	16^2	*MSB*

Výsledek: $1a7_{16}$

Vzájemný převod mezi hex. a dekadickou soustavou

Řešení

(b) Metoda postupného násobení desetinné části základem hexadecimální soustavy.

0,34	$\cdot$	*16*	$=$	**5**,*44*	16^{-1}	*první symbol po řádové čárce*
0,44	$\cdot$	*16*	$=$	**7**,*04*	16^{-2}	
0,04	$\cdot$	*16*	$=$	**0**,*64*	16^{-3}	
0,64	$\cdot$	*16*	$=$	**10**,*24*	16^{-4}	
0,24	$\cdot$	*16*	$=$	**3**,*84*	16^{-5}	

...

Lze pokračovat, dokud není desetinná část násobení rovna nule, příp. po dosažení požadované bitové přesnosti.

Výsledek přibližně: $1a7{,}570a3_{16}$

Vzájemný převod mezi hexadecimální a binární soustavou – rychlý postup

- Algoritmus převodu z hexadecimální do binární soustavy: Každý symbol v hexadecimální soustavě se převádí odděleně a představuje právě čtveřici bitů

Example

Převeďte hodnotu −6f,aH do binární soustavy.

Řešení

$-6f{,}a_{16} = -\ 0110\quad 1111\ ,\ 1010_2$

- Algoritmus převodu z binární do hexadecimální soustavy: Binární číslo se rozdělí na čtveřice bitů (tj. na nibly) a ty se převádí nezávisle na ostatních

Example

Převeďte binární hodnotu 0b11,000011001 do hex. soustavy.

Řešení

$11{,}000011001_2 = 0011\ ,\ 0000\quad 1100\quad 1000_2 = 3{,}0c8_{16}$

Základní početní operace s neznaménkovými binárními čísly, příklady na tabuli

- Součet
- Součin
- Mocniny dvou (bitový posuv doleva, doprava)

Obsah přednášky

Vyjádření záporných čísel ve fixed-point

- V číslicových systémech s pevnou řádovou čárkou se k vyjádření záporných celých čísel používá doplňkový kód ve tvaru dvojkového doplňku (Two's Complement)
- **Dvojkový doplněk** reprezentuje číslo opačné k dané hodnotě, tj. -5 k hodnotě 5, ale také $+3$ k hodnotě -3
- U znaménkových čísel informuje zda je hodnota kladná nebo záporná, bit na pozici MSB:
  - 0 $\Leftrightarrow$ kladná hodnota; 1 $\Leftrightarrow$ záporná hodnota

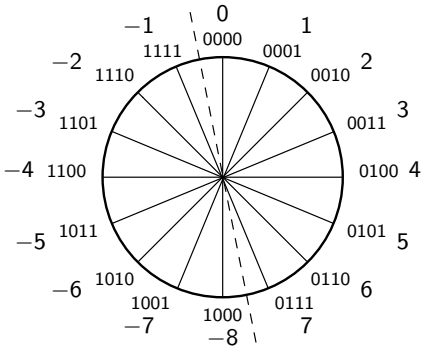

Figure: Grafická interpretace rozsahu binárních (znaménkových) hodnot

Dvojkový doplněk

- Algoritmus pro vytvoření dvojkového doplňku:
  (a) negace všech bitů (tj. tvorba jednotkového doplňku)
  (b) přičíst 1

Example

Doplňkový dvojkový kód k $-2 = ?$

Řešení

(a) $+2$: *0010* $\rightarrow$ *(negace) 1101*

(b) *1101* $+$ *0001* $=$ *1110*: -2

- Rychlá metoda tvorby dvojkového doplňku:
  (1) opisovat všechny nulové bity od LSB směrem k MSB
  (2) první jedničkový bit opsat
  (3) zbývající bity negovat

Početní operace se znaménkovými čísly

- Aritmetické bitové operace pro záporná čísla ve dvojkovém doplňku platí stejně jako pro kladná čísla.
- Sčítání znaménkových hodnot:

$$\begin{array}{rr} 01001 & (9) \\ +11110 & (-2) \\ \hline 00111 & (7) \end{array}$$

- Odečítání znaménkových čísel lze vyjádřit pomocí součtu čísla opačného:
  (1) k menšiteli vytvořit dvojkový doplněk
  (2) tento doplněk přičíst k menšenci

Př.: $9 - 4 = 9 + (-4) = ?$

00100 $(+4) \longrightarrow$ negace:

$$\begin{array}{r} 11011 \\ +00001 \\ \hline (-4)\ 11100 \end{array} \qquad \begin{array}{r} 01001 \\ +11100 \\ \hline 00101 \end{array} \qquad \begin{array}{r} 9 \\ +(-4) \\ \hline 5,\ \mathrm{C}{=}1 \end{array}$$

Rekapitulace (ne)znaménkových celých čísel pro fixed-point

- Neznaménková celá čísla (unsigned) reprezentují hodnoty od 0 do $2^n - 1$, kde n udává počet bitů
- Znaménková celá čísla (signed) jsou v číslicové technice vyjádřena ve tvaru dvojkového doplňku a reprezentují hodnoty od -2^{n-1} až $2^{n-1} - 1$
- Při použití celočíselného typu (signed, unsigned) mohou nastat dva problémy:
  - přetečení při součtu (výsledek je obecně vyjádřen $n+1$ bity)
  - přetečení při násobení (výsledek je obecně vyjádřen $2 \cdot n$ bity)

Odstranění problémů s přetečením u (ne)znaménkových čísel

- Potřebu většího počtu bitů pro vyjádření výsledku než mají operandy aritmetických operací je možné odstranit:
  (1) saturací výsledku, tj. omezení/oříznutí na maximální počet bitů
  (2) použitím zvýšené přesnosti pro výsledek, tj. větší počet bitů
  (3) použitím zlomkového formátu
  (4) použitím formátu plovoucí řádové čárky

(1a) Saturace výsledků u neznaménkových čísel:
- pokud $A \cdot B \leq 2^n - 1$, pak výsledek $= A \cdot B$
- pokud $A \cdot B > 2^n - 1$, pak výsledek $= 2^n - 1$

(1b) Saturace výsledků u znaménkových čísel:
- pokud $-2^{n-1} \leq A \cdot B \leq 2^{n-1} - 1$, pak výsledek $= A \cdot B$
- pokud $A \cdot B > 2^{n-1} - 1$, pak výsledek $= 2^{n-1} - 1$
- pokud $-2^{n-1} > A \cdot B$, pak výsledek $= -2^{n-1}$

Odstranění problémů s přetečením u (ne)znaménkových čísel

(2) Rozšíření počtu bitů pro reprezentaci početně korektního výsledku aritmetické operace přináší několik nevýhod:
- zvýšení paměťové náročnosti (použití většího počtu registrů/paměťových pozic)
- v případě, že výsledek aritmetické operace je následně operandem další operace, je nutné data upravit do původního počtu bitů
- následné snížení počtu bitů je nutné také v případě vyslání vypočtené hodnoty do D/A převodníku

(3) Použití zlomkového tvaru (formátu) zabraňuje přetečení při násobení, protože platí $|A \cdot B| < min(|A|, |B|)$

Fixed-point: Zlomkový tvar

- Převod hodnoty zlomkového tvaru do dekadické soustavy; 32bitové vyjádření $\mathbf{d} = (d_{31}, d_{30}, \ldots, d_0)$:

$$v = -d_{31} \cdot 2^0 + \sum_{n=1}^{31} d_{31-n} \cdot 2^{-n}$$

- tj. znaménkový bit (zde d_{31}) má váhu -1, ostatní bity mají váhu záporných mocnin: $1/2$, $1/4$, $1/8$, ...
- Zlomkový tvar nabývá hodnot $< -1, 1 - 2^{-(n-1)} >$

  pro $n = 8$ je $MAX = 0,9921875$

  pro $n = 16$ je $MAX = 0,99996948242188$

Formáty s pevnou řádovou čárkou

Figure: Nejpoužívanější formáty s pevnou řádovou čárkou

Obsah přednášky

Číselná reprezentace s plovoucí řádovou čárkou

- Pro reprezentaci desetinných čísel a pro zvýšení rozsahu je použita reprezentace s plovoucí řádovou čárkou (Floating-point)
- Základní formáty dle standardu ANSI/IEEE 754:
  - jednoduchá přesnost (Single precision, SP): 32 bitů
  - dvojitá přesnost (Double precision, DP): 64 bitů
  - rozšířená přesnost (Extended precision, EP): 80 bitů
- Formát IEEE floating-point reprezentuje běžné hodnoty, NaN (not a number) a nekonečno
- Formát binárního čísla se skládá ze tří částí:
  - znaménkový bit
  - exponent
  - mantisa

Floating-point, jednoduchá přesnost

- Znaménkový bit $= 1 \Leftrightarrow$ záporné číslo:
  - záporná čísla tedy nejsou ve floating-point vyjádřena ve dvojkovém doplňku; čísla opačná se liší pouze znaménkovým bitem
- Mantisa (fraction):
  - reprezentuje nejdůležitější bity kódovaného čísla
  - hodnota je normována $\Rightarrow$ desetinná čárka je přesunuta za první nenulovou číslicí (v binární soustavě za první jedničku)
- Exponent:
  - může vyjádřit jak kladný, tak i záporný exponent
  - k dané hodnotě je vždy přičtena konstanta, tzv. bias $2^{nexp-1} - 1$, kde $nexp$ je počet bitů pro vyjádření exponentu, proto je výsledná hodnota exponentu ve formátu floating-point vždy kladná. Bias pro SP: 127_{10}

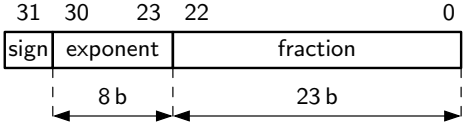

Figure: Struktura formátu s plovoucí řádovou čárkou, jednoduchá přesnost

Floating-point, jednoduchá přesnost

Example

Vyjádřete desetinné číslo $+6{,}625_{10}$ ve tvaru s plovoucí řádovou čárkou, jednoduchá přesnost.

Řešení

(1) *Přepsat dané číslo do bin. soust.:* $6{,}625 = 110{,}101_2$

(2) *Normovaná podoba čísla:* $110{,}101 \rightarrow 1{,}10101 \cdot 2^2$

(3) *23bitová mantisa (hodnoty nejdůležitějších bitů za první jedničkou, doplněných nulami):*
1010 1000 0000 0000 0000 000

(4) *Exponent:* $2 + 127$ *(bias pro SP)* $= 129_{10} = 1000\ 0001_2$

(5) *Znaménkový bit:* $0 \Leftrightarrow$ *kladné číslo*

Výsledek: $0100\ 0000\ 1101\ 0100\ 0000\ 0000\ 0000\ 0000_2$

Floating-point, jednoduchá přesnost

Example

Podle standardu IEEE 754 přepište číslo 0100 1111 1001 0101 0000 0010 1111 1001 z plovoucí řádové čárky do dekadické podoby.

Řešení

(1) *Znaménkový bit: 0 ⇔ kladné číslo*

(2) *Exponent:* $1001\ 1111_2 - 127_{10} = 159 - 127 = 32$

(3) $1{,}0010\ 1010\ 0000\ 0101\ 1111\ 001 \cdot 2^{32}$ *(tj. přesun binární čárky o 32 pozic doprava; ekvivalent zápisu z desítkové soustavy:* $1{,}5 \cdot 10^4 = 15\,000$*)*

1 0010 1010 0000 0101 1111 0010 0000 0000

Výsledek: $5\,000\,000\,000_{10}$

Pozn.: Pro srovnání: ve dvojkovém doplňku (fixed-point) lze pomocí stejného počtu bitů (tj. 32) vyjádřit hodnoty od $-2\,147\,483\,648$ do $2\,147\,483\,647$

Pozn.: Rozsah hodnot single-precision leží v intervalu cca $\pm 3{,}4 \cdot 10^{38}$

Floating-point, dvojitá přesnost

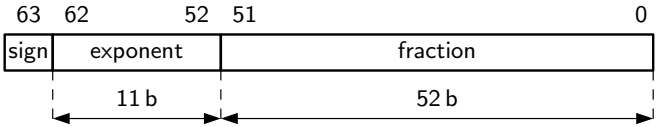

Figure: Struktura formátu s plovoucí řádovou čárkou, dvojitá přesnost

- Formát s dvojitou přesností (double precision, 64 bitů) se skládá ze tří částí:
  - znaménkový bit (63)
  - 11bitový exponent (62:52)
  - 52bitová mantisa (51:0)
- Reálná hodnota je vyjádřena vztahem $(-1)^{sign} \cdot 2^{exp-bias} \cdot 1, mantisa$, kde posunutí $bias = 2^{nexp-1} - 1$, tj. pro double precision $bias$=1 023 (pro SP: 127)
- Rozsah hodnot double-precision leží v intervalu cca $\pm$1,8·10^{308}

Floating-point, dvojitá přesnost

Example

Podle standardu IEEE 754 přepište číslo z plovoucí řádové čárky do dekadické podoby

- 1
- 100 0001 1001
- 1101 0110 1111 0011 0100 0101 0100 0000 0000 0000 0000 0000 0000

Řešení

(1) *Záporné číslo*

(2) *Exponent:* $exp{=}1\,049{-}1\,023{=}26$

(3) *Mantisa: 1,1101 0110 1111 0011 0100 0101 0100* $\ldots \cdot 2^{exp}$

Výsledek: $-123\,456\,789$

Floating-point, dvojitá přesnost

Example

Vyjádřete desetinné číslo +255,96875 ve tvaru s plovoucí řádovou čárkou (dvojitá přesnost).

Řešení

(1) *Kladné číslo:* $sign$*=0*

(2) *Přepsat hodnotu do binární soustavy: 255,96875 = 0b1111 1111,1111 1*

(3) *Normovaná podoba: 1,111 1111 1111 1* $\cdot 2^7$

(4) *Exponent:* exp*=7 + 1 023 = 1 030 = 0b100 0000 0110*

Výsledek:

- *0*
- *100 0000 0110*
- *111 1111 1111 1 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000*

Speciální hodnoty ve formátu floating-point

Table: Některé speciální hodnoty formátu floating-point

Hodnota	**Znaménko**	**Exponent**	**Mantisa**
"Kladná" nula	0	0	0
"Záporná" nula	1	0	0
Nekonečno	1 nebo 0	0b1111_1111	0
NaN (not a number)	1 nebo 0	0b1111_1111	$\neq 0$

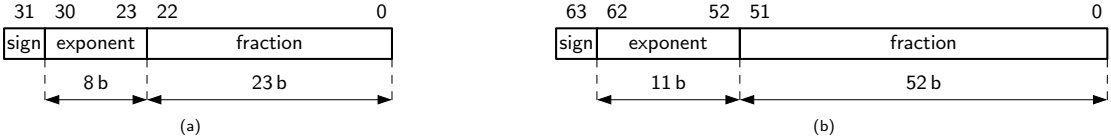

Figure: Struktura formátů s plovoucí řádovou čárkou: (a) single, (b) double precision