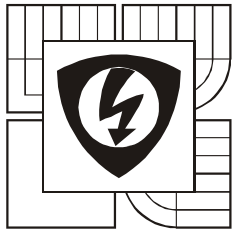
Kvantování a formáty čísel



Kurz: Signálové procesory

Autor: Petr Sysel

Lektor: Petr Sysel

Vytvoření této videopřednášky bylo podpořeno projektem č. CZ.1.07/2.2.00/28.0098
Evropského sociálního fondu a státním rozpočtem České republiky.

Obsah přednášky

Kvantování signálu

- Při kvantování je signál spojitý v hodnotách vyjádřen pomocí konečného počtu bitů N_b,
- signál pak může nabývat pouze 2^{N_b} kvantovacích hladin,
- vzdálenost mezi mezi hladinami udává kvantovací krok q roven váze nejnižšího bitu (nejčastěji $q = 2^{-N_\mathrm{b}+1}$),
- při usekávání (truncation) je hodnota signálu vyjádřena vždy nejbližší nižší hladinou

$$x_\mathrm{T}(t) = \lfloor x(t) \rfloor \,,$$

- při zaokrouhlování (rounding) je hodnota signálu vyjádřena vždy nejbližší hladinou

$$x_\mathrm{R}(t) = \left\lfloor x(t) + \frac{q}{2} \right\rfloor .$$

Vznik kvantovacího šumu

- Rozdíl mezi kvantovaným signálem $x_Q(t)$ a původním signálem $x(t)$ se označuje jako kvantovací šum

$$e_Q = X_Q - X,$$

- rozsah kvantovacího šumu je v případě usekávání roven

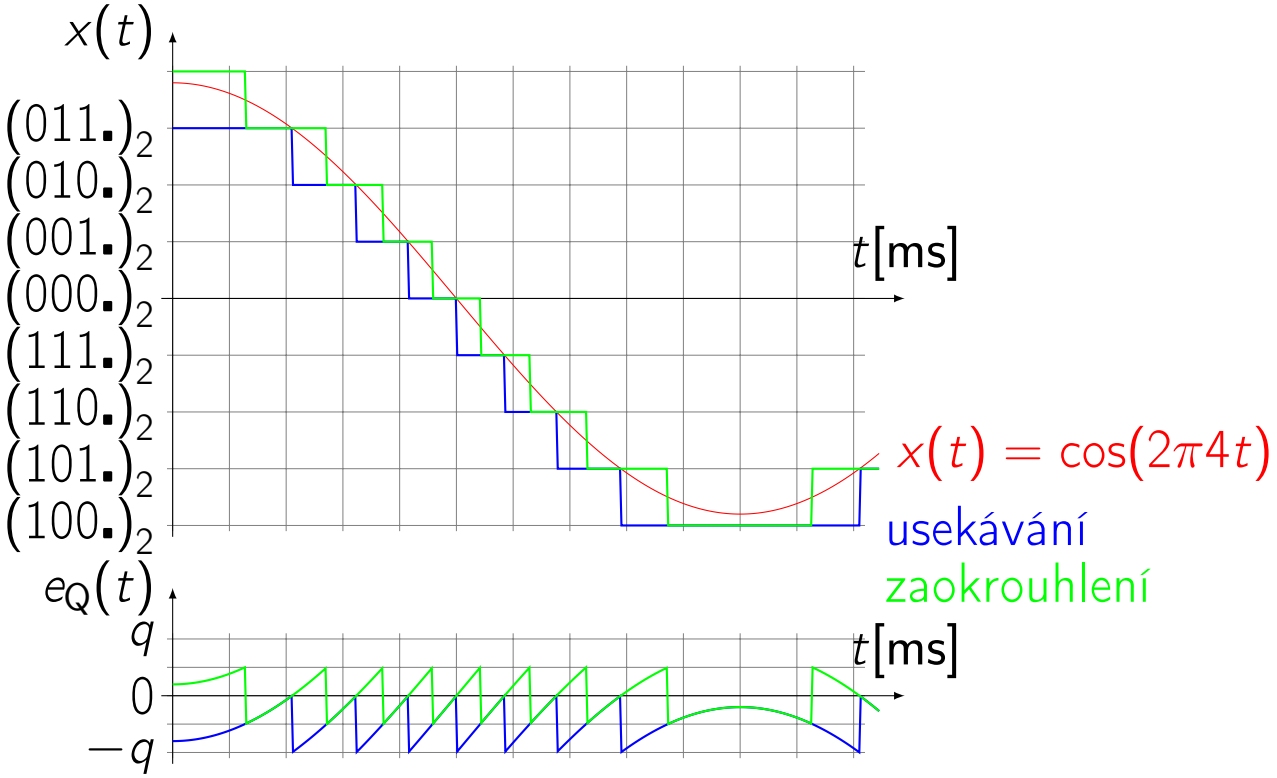
$$-q < e_T \leq 0,$$

- rozsah kvantovacího šumu je v případě zaokrouhlování roven

$$-\frac{q}{2} < e_R \leq \frac{q}{2},$$

- charakter kvantovacího šumu závisí i na tvaru kvantovaného signálu,
- pro zjednodušení se uvažuje, že kvantovací šum nabývá hodnot podle rovnoměrného rozdělení pravděpodobnosti (všechny hodnoty jsou stejně pravděpodobné).

Kvantování signálu



Vznik kvantovacího šumu

- Střední hodnota kvantovací chyby je rovna:

$$\mu_{\mathrm{Q}} = \int e_{\mathrm{Q}} p(e_{\mathrm{Q}}) \mathrm{d}e_{\mathrm{Q}}$$

- v případě usekávání je střední hodnota $\mu_{\mathrm{T}} = -\frac{q}{2}$,
- v případě zaokrouhlení je střední hodnota $\mu_{\mathrm{R}} = 0$.
- Rozptyl kvantovacího šumu je roven:

$$\sigma_{\mathrm{Q}}^2 = \int (e_{\mathrm{Q}} - \mu_{\mathrm{Q}})^2 \, p(e_{\mathrm{Q}}) \mathrm{d}e_{\mathrm{Q}} = \frac{q^2}{12} = \sigma_{\mathrm{T}}^2 = \sigma_{\mathrm{R}}^2,$$

- V případě zaokrouhlování je potom poměr signálu od kvantovacího šumu ($SQNR$) roven:

$$SQNR = 10 \log_{10} \frac{\sigma_x^2}{\sigma_{\mathrm{R}}^2} = 10 \log_{10} \frac{\sigma_x^2}{\frac{q^2}{12}} = 6{,}02 N_{\mathrm{b}} + 10{,}79 + 10 \log_{10} \sigma_x^2.$$

Kvantování vstupního signálu

- V případě usekávání bude poměr ještě menší, protože energie šumu je větší díky nenulové střední hodnotě,
- poměr signálu od kvantovacího šumu vzrůstá s počtem bitů,
- poměr signálu od kvantovacího šumu vzrůstá s rozptylem (energií) vstupního signálu,
- rozptyl hodnot vstupního signálu je však omezen rozsahem čísel,
- navíc u některých algoritmů může mít výstupní signál větší rozptyl než vstupní,
- v takovém případě je nutné vstupní signál zeslabit vážením (scaling) koeficientem $\beta < 1$,
- potom

$$SNR = 10 \log_{10} \frac{\beta^2 \sigma_x^2}{\sigma_R^2} = 10{,}79 + 6{,}02N + 10 \log_{10} \sigma_x^2 + 20 \log_{10} \beta.$$

Důležitost volby formátu

Je i dnes nutné řešit formát čísel?

- z důvodu ceny?

fixed point		floating point	
TMS320C6421	$10.10	TMS320C6720	$7.09
TMS320C6416	$205.67	TMS320C6713	$19.20

- z důvodu přesnosti?

	fixed point	floating point
dynamický rozsah [dB]	90 - 186	1 530 - 12 282
SQNR [dB]	90 - 186	144 - 318

- z důvodu výpočetního výkonu?

fixed point		floating point	
TMS320C6421	400 MHz	TMS320C6720	350 MHz
TMS320C6416	1 000 MHz	TMS320C6713	300 MHz

Pohyblivá řádová čárka

- Formát v pevné řádové čárce má omezenou přesnost, ale především malý dynamický rozsah,
- proto se používá i formát v pohyblivé řádové čárce (floating point), kdy číslo je vyjádřeno pomocí mantisy a exponentu

$$mantisa \cdot 2^{exponent},$$

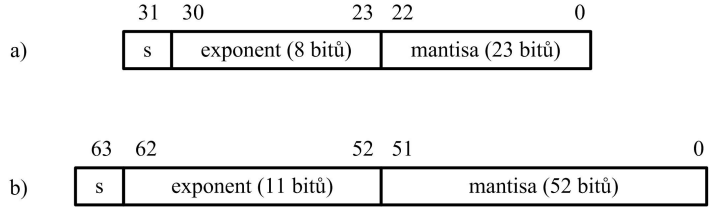
- standard IEEE 754 definuje několik formátů uložení čísel s pohyblivou řádovou čárkou, z nichž nejpoužívanější jsou

formát	znaménko	exponent	mantisa	celkem
základní přesnost	1	8	23	32
dvojinásobná přesnost	1	11	52	64

- pro zvýšení přesnosti je každé číslo normalizováno tak, aby hodnota mantisy byla v rozsahu $1 \leq mantisa < 2$, nejvyšší bit však není vyjádřen (tzv. skrytá jednička).

Pohyblivá řádová čárka

Pro vyjádření se používá formát IEEE754.



- mantisa je vyjádřena přímým kódem,
- exponent je povýšen o hodnotu 127, resp. 1023, a poté vyjádřen také přímým kódem.

a	b
základní přesnost	dvojnásobná přesnost
$y = -1^s \cdot 2^{(exp-127)} \cdot 1,mantisa$	$y = -1^s \cdot 2^{(exp-1023)} \cdot 1,mantisa$

Příklady čísel v pohyblivé řádové čárce

- Číslo 0,75 normalizujeme vynásobením 2:
  znaménko $+$ 0,
  exponent -1 01111110,
  mantisa 1.5 .10000000000000000000000,
- číslo $-123{,}745 \cdot 10^{-6}$ normalizujeme vynásobením 2^{13}:
  znaménko $-$ 1,
  exponent -13 01110011,
  mantisa 1,01371904 .00000011100000110001011,
- po zpětném převodu dostaneme číslo
  $123{,}744990560226142406463623046 88 \cdot 10^{-6}$,
- číslo $6564 \cdot 10^{9}$ normalizujeme vynásobením 2^{-42}:
  znaménko $+$ 0,
  exponent 42 10101001,
  mantisa 1,49248080 ... .01111110000100110011100,

Porovnání single a double

hodnota	Matlab	single	double
maximální kladné číslo	realmax	1.7014e+38	1.7977e+308
minimální kladné číslo	realmin	1.1755e-38	2.2251e-308
kvantovací krok (přesnost)	eps(x)	1.1921e-07	2.2204e-16

Možné případy vyjádření v základní přesnosti

Znaménko	Exponent	Mantisa	Význam
0	$0 < exp < 255$	> 0	normalizované kladné číslo
1	$0 < exp < 255$	> 0	normalizované záporné číslo
0	0	> 0	denormalizované kladné číslo
1	0	> 0	denormalizované záporné číslo
0	0	0	kladná nula
1	0	0	záporná nula
0	255	0	kladné nekonečno
1	255	0	záporné nekonečno
0	255	> 0	NaN – Not a Number
1	255	> 0	NaN – Not a Number

Dynamický rozsah

- Označme počet bitů mantisy N_m a počet bitů exponentu N_e,
- nejmenší vyjádřitelné kladné číslo

$$x_{min} = 2^{-2^{N_\mathrm{e}}},$$

- největší vyjádřitelné kladné číslo

$$x_{max} = (2 - 2^{N_\mathrm{m}}) \cdot 2^{2^{N_\mathrm{e}}},$$

- dynamický rozsah definovaný jako podíl největšího ku nejmenšímu kladnému číslu

$$R_{FLP} = 20 \log \frac{x_{max}}{x_{min}} = 20 \log \frac{(2 - 2^{N_\mathrm{m}}) \cdot 2^{2^{N_\mathrm{e}}}}{2^{-2^{N_\mathrm{e}}}} \cong 6 \cdot (2^{N_\mathrm{e}+1} - 1)[\mathrm{dB}].$$

Přesnost

- Naproti tomu přesnost ve smyslu poměr podílu vzdálenosti mezi dvěma nejbližšími čísly ku nižšímu z nich je dána především počtem bitů mantisy,
- uvažujme číslo $1.0000\ldots0000 \cdot 2^{exp}$,
- nejbližší vyšší číslo bude $1.0000\ldots0001 \cdot 2^{exp}$,
- rozdíl mezi nimi (dalo by se říct kvantovací krok) bude $0.0000\ldots0001 \cdot 2^{exp}$,
- potom poměr bude dán

$$\begin{aligned} 20\log_{10}\frac{0.000\ldots00001 \cdot 2^{exp}}{1.0000\ldots0000 \cdot 2^{exp}} &= 20\log_{10} 0.0000\ldots0001 \\ &= 20\log_{10} 2^{-N_\mathrm{m}} \approx 6 \cdot N_\mathrm{m}, \end{aligned}$$

- a tento poměr bude přibližně konstantní.

Operace v pohyblivé řádové čárce

- Sčítání
  - sjednotíme exponent obou činitelů tak, že číslo s menším exponentem posuneme vpravo o rozdíl exponentů,
  - provedeme součet mantis,
  - výsledek normalizujeme.
- Odčítání
  - sjednotíme exponent obou činitelů tak, že číslo s menším exponentem posuneme vpravo o rozdíl exponentů,
  - provedeme rozdíl mantis,
  - výsledek normalizujeme.

Operace v pohyblivé řádové čárce

- Násobení
  - sečteme exponenty obou činitelů,
  - provedeme vynásobení mantis,
  - výsledek normalizujeme.
- Dělení
  - odečteme exponenty obou činitelů,
  - provedeme dělení mantis,
  - výsledek normalizujeme.

Pevná řádová čárka

- V případě většiny signálových procesorů jsou čísla vyjádřena v pevné řádové čárce,
- číslo je vyjádřeno na pevný počet bitů,
- řádová čárka je pevně umístěna a neposouvá se,
- v pevné řádové čárce lze rozlišit čísla:
  - celá (s těmito pracují procesory pro všeobecné použití),
  - smíšená (jako výsledky některých operací),
  - zlomková (používají se v signálových procesorech),
- označme N_b^+ počet bitů pro celou část a N_b^- počet bitů pro zlomkovou část, počet bitů $N_\mathrm{b} = N_\mathrm{b}^+ + N_\mathrm{b}^-$,
- potom číslo takto vyjádřené bude

$$x = \sum_{k=-N_\mathrm{b}^-}^{N_\mathrm{b}^+} b_i \cdot 2^k.$$

Pevná řádová čárka

- Příklady vyjádření $N_{\mathrm{b}}^{+} = 3, N_{\mathrm{b}}^{-} = 4$ $(Q3.4)$:

$$\begin{array}{lll} 2{,}4375 & 010.0111 & 2{,}4375, \\ 0{,}3128 & 000.0101 & 0{,}3125, \\ 9{,}8456 & 111.1101 & 7{,}8125. \end{array}$$

Pevná řádová čárka

- Omezme se pouze na zlomková čísla s délkou zlomkové části $N_\mathrm{b}^- = N_\mathrm{b}, N_\mathrm{b}^+ = 0$,
- nejmenší a největší vyjádřitelné kladné číslo bude

$$x_\mathrm{min} = 2^{-N_\mathrm{b}^-}, x_\mathrm{max} = 1 - 2^{-N_\mathrm{b}^-}$$

- poměr obou čísel udává relativní rozsah vyjadřitelných čísel – dynamický rozsah

$$R_\mathrm{FXD} = 20 \log \frac{x_\mathrm{max}}{x_\mathrm{min}} = 20 \log \frac{1 - 2^{-N_\mathrm{b}^-}}{2^{-N_\mathrm{b}^-}} = 20 \log 2^{N_\mathrm{b}^-} = N_\mathrm{b}^- 20 \log 2,$$
$$R_\mathrm{FXD} \approx 6 \cdot N_\mathrm{b}^- \,[\mathrm{dB}].$$

Pevná řádová čárka

- Poměr signálu ku kvantovacímu šumu $SQNR \approx 6 \cdot N_\mathrm{b}$ [dB], kde N_b je délka slova, – nepřímo vyjadřuje přesnost zpracování,
- dynamický rozsah $R = 20 \cdot \log_{10} \frac{x_{max}}{x_{min}} \approx 6 \cdot N_\mathrm{b}$ [dB], kde x_{max} je maximální a x_{min} je minimální vyjádřitelné kladné číslo,
- dynamický rozsah použitého formátu musí odpovídat dynamickému rozsahu zpracovávaného signálu – jinak dojde ke zkreslení,

Pevná řádová čárka

- Převod celého čísla:
  1. provedeme celočíselné dělení čísla 2,
  2. zapíšeme zbytek po dělení (0 nebo 1),
  3. pokud výsledek dělení je větší než 0, tak binární číslo posuneme o 1 bit vpravo a pokračujeme bodem 1.
- převod zlomkového čísla:
  1. převáděné číslo vynásobíme 2,
  2. pokud je výsledek násobení větší než 1, zapíšeme 1 a 1 odečteme,
  3. pokud je výsledek násobení menší než 1, zapíšeme 0,
  4. pokračujeme bodem 1 dokud není výsledkem 0 nebo nedosáhneme zadaného počtu bitů.

Rozdíly v operacích `int` × `frac`

	integer	fractional
megaAVR	`mul`	`fmul`
DSP56300	`mpy`	`smpy`
TMS320C6000	`mpy`	`smpy`

Příklad násobení:

celá čísla			zlomková čísla		
7		0111	0.111		0,875
× 5		× 0101	× 0.101		× 0,625
35	=	0010 0011	0.010 0011	≠	0,546875

Vyjádření záporných čísel

- V případě čísel se znaménkem se před nejvyšší bit MSB formátu přidává znaménkový bit,
- znaménkový bit vždy vyjadřuje znaménko čísla:
  - 0 – kladné číslo,
  - 1 – záporné číslo,
- kladné číslo je tak vyjádřeno stejně jako předtím,
- pro vyjádření záporného čísla se pak používají formáty:
  - přímý kód – záporná čísla se liší pouze ve znaménkovém bitu,
    - číslo převedeme jako číslo kladné a nastavíme znaménkový bit.
  - jednotkový doplněk – záporná čísla jsou vyjádřena doplňkem do 1,
    - číslo převedeme jako číslo kladné a invertujeme všechny bity.
  - dvojkový doplněk – záporná čísla jsou vyjádřena doplňkem do 2,
    - číslo převedeme jako číslo kladné,
    - invertujeme všechny bity,
    - k nejnižšímu bitu připočteme 1.

Vyjádření záporných čísel

- Přímý kód – číslo je vyjádřeno znaménkem plus zlomková část:

$$+0{,}8125 \iff 0.1101,$$
$$-0{,}8125 \iff 1.1101,$$

- jednotkový doplněk (inverzní kód) – záporné číslo je vyjádřeno znaménkem a zlomková část je invertována:

$$+0{,}8125 \iff 0.1101,$$
$$-0{,}8125 \iff 1.0010,$$

- dvojkový doplněk (doplňkový kód) – záporné číslo je vyjádřeno doplňkem do dvojky (lze provézt invertováním všech bitů a přičtením 1 k nejnižšímu bitu):

$$+0{,}8125 \iff 0.1101,$$
$$-0{,}8125 \iff 1.0011.$$

Výhody a nevýhody vyjádření

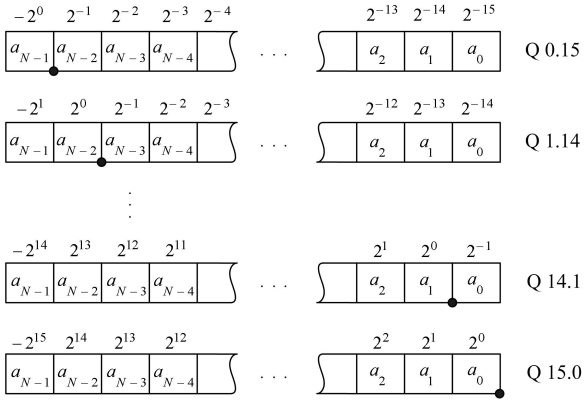
- Přímý kód:
- inverzní kód:
  - existují dvě nuly (0.0000, 1.0000),
  - symetrický rozsah $(-(1-2^{-b});(1-2^{-b}))$,
  - v případě některých operací je nutné provádět korekce výsledku podle znaménka operandů,
- dvojkový doplněk:
  - nesymetrický rozsah $(-1;1-2^{-b})$,
  - existuje pouze jedna nula (0.0000),
  - může dojít ke krátkodobému přetečení a přesto bude výsledek správně.

Výhody a nevýhody vyjádření

- Příklad ošetření krátkodobého přetečení:

	přímý kód		dvojkový doplněk	
0,6875	0.1011	0,6875	0.1011	0,6875
0,5625	0.1001	0,5625	0.1001	0,5625
1,2500	1.0100	−0,2500	1.0100	−0,7500
−0,3750	1.0110	−0,3750	1.1010	−0,3750
0,8750	1.1010	−0,6250	0.1110	0,8750

Příklad formátů v pevné řádové čárce se znaménkem



Převod zlomkového čísla ve formátu $\mathrm{Q}0.n$ na celé číslo ve formátu $\mathrm{Q}n.0$ je proveden vynásobením hodnotou 2^n. Obrácený převod násobením hodnotou 2^{-n}.

Sčítání a odečítání

Sčítání:

- v přímém kódu je nutné testovat znaménkový bit, přičtení záporného čísla je realizováno odečtením jeho kladného ekvivalentu,
- v doplňkovém kódu je nutné v případě přenosu mimo slovo přičíst 1,
- v dvojkovém doplňku není nutná žádná korekce.

Odčítání:

- v přímém kódu je nutné testovat znaménkový bit, odečtení záporného čísla je realizováno přičtením jeho kladného ekvivalentu,
- v doplňkovém kódu je nutné v případě přenosu mimo slovo odečíst 1,
- v dvojkovém doplňku není nutná žádná korekce.

Při sčítání nebo odčítání dvou čísel stejného znaménka může dojít k tomu, že výsledek nelze zobrazit v daném formátu. Tento jev se označuje jako *přetečení*. Při sčítání nebo odčítání čísel s různým znaménkem přetečení nemůže vzniknout.

Násobení v pevné řádové čárce

Při násobení v pevné řádové čárce má výsledek dvojnásobný počet bitů.

Násobení v pevné řádové čárce:

- v přímém kódu násobíme čísla bez znaménka a znaménko výsledku nastavíme podle znamének činitelů – pokud jsou shodná, je znaménko výsledku +, pokud jsou různá, je znaménko výsledku −.
- v doplňkovém kódu jsou záporná čísla nejprve negována a potom násobena jako čísla celá, v případě násobení dvou čísel s různým znaménkem musí být výsledek negován,
- v dvojkovém doplňku je nutné v případě násobení záporným číslem provést korekci – od horního slova odečtu druhého činitele, pokud oba činitelé mají záporné znaménko, musí se odečíst obě.

Chyby způsobené omezení délky slova lze omezit, pokud použijeme zlomkových čísel.

Násobení zlomkových čísel v pevné řádové čárce

Násobení zlomkových čísel v pevné řádové čárce:

- Násobení probíhá jako v případě čísel celých,
- po násobení je nutné provést bitový posun vlevo se saturací.
- Takto je možné realizovat násobení zlomkových čísel i na procesorech, které nepodporují uložení zlomkových čísel.

Při násobení zlomkových čísel v dvojkovém doplňku může dojít k přetečení, pokud násobíme dvě nejmenší čísla $-1 \cdot -1$.

Operace dělení

- Dělení je nejsložitější operace a proto se jí snažíme vyhnout,
- je nutné ho realizovat postupným odečítáním dělitele od dělence,
- velká většina signálových procesorů má sice pro dělení instrukci, ale její náročnost je několikanásobná:
  např. u 56F8367 trvá je pro dělení 32 bitů / 16 bitů opakovat instrukci `div` 16 krát (16 hodinových cyklů).
- v případě procesorů TMS320C6000 se pro iteraci používá instrukce `SUBC` – podmíněné odečítání (subtract conditional).

Operace dělení

- Dělíme 11 ($(1011)_2$) hodnotou 3 ($(0011)_2$),
- v každé iteraci se pokusíme odečíst dělitel posunutý o 3 bity vlevo
  - pokud bude rozdíl menší než 0, pak se k výsledku zpátky připočte dělenec a součet se posune vlevo o 1 bit,
  - pokud bude rozdíl větší nebo roven 0, pak se výsledek posune vlevo a zprava se nasune 1,
- u dělení se znaménkem se podělí absolutní hodnoty a znaménko se nastaví podle operandů.

	0000	1011	
$-$	0001	1000	
$=$	1111	0011	< 0
	0001	0110	
$-$	0001	1000	
$=$	1111	1110	< 0
	0010	1100	
$-$	0001	1000	
$=$	0001	0100	≥ 0
	0010	1001	
$-$	0001	1000	
$=$	0001	0001	≥ 0
	0010	0011	
	zb.	výsl.	

Přetečení

Přetečení znamená, že výsledek leží mimo rozsah použitého vyjádření čísel.

Příklad přetečení v dvojkovém doplňku:

5		0	1	0	1	-5		1	0	1	1
4		0	1	0	0	-4		1	1	0	0
9	$\neq$	1	0	0	1	-9	$\neq$	0	1	1	1

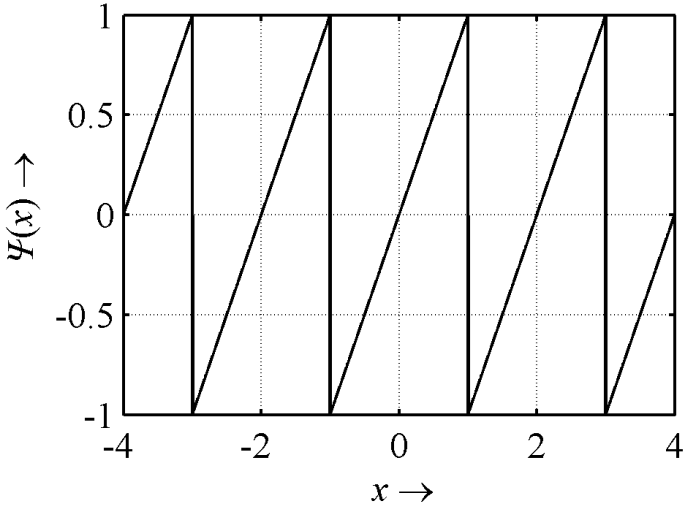
Řešení přetečení:

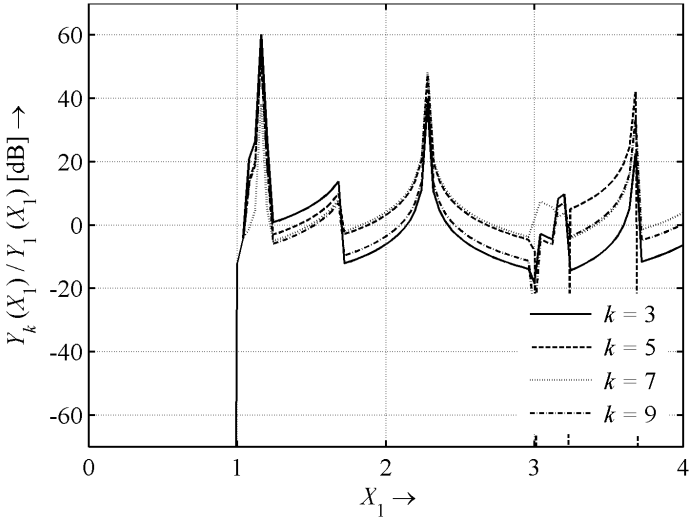
1. číslo po přetečení vynulujeme,
   - vynulování způsobí velkou chybu,
2. číslo po přetečení nahradíme největším kladným číslem nebo nejmenším záporným číslem podle znaménka,
   - chyba je mnohem menší,
   - používá se mnohem častěji.

Pro ošetření krátkodobého přetečení mají střadače (akumulátory) signálových procesorů rozšiřující část zvyšující rozsah zobrazitelných čísel.

Neošetřené přetečení (wrap around)

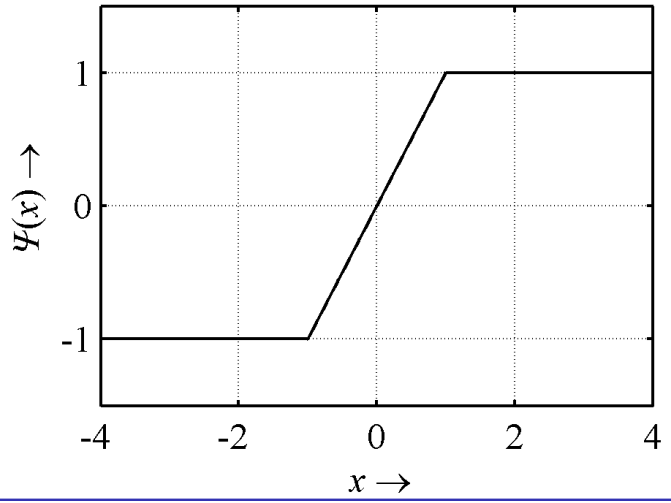
- Pokud hodnota překročí největší vyjádřitelné číslo, dojde k přetečení,
- podobně pokud hodnota je menší než nejmenší vyjadřitelné číslo,
- převodní charakteristika neošetřeného přetečení má tvar pily,
- v signálu vzniknou vyšší harmonické, jejichž úroveň může být značná.

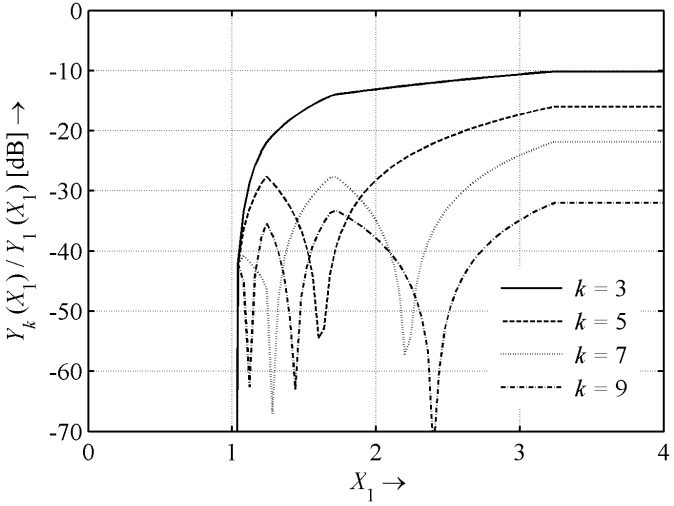




Ošetřené přetečení (saturation)

- Pro ošetření přetečení se používá saturace,
- číslo je číslo větší než největší vyjádřitelné číslo, je jím nahrazeno,
- podobně pokud je číslo menší než nejmenší vyjadřitelné číslo, je jím nahrazeno,
- i v tomto případě vzniknou vyšší harmonické, ale mají mnohem menší úroveň.





Zaokrouhlení zlomkového čísla

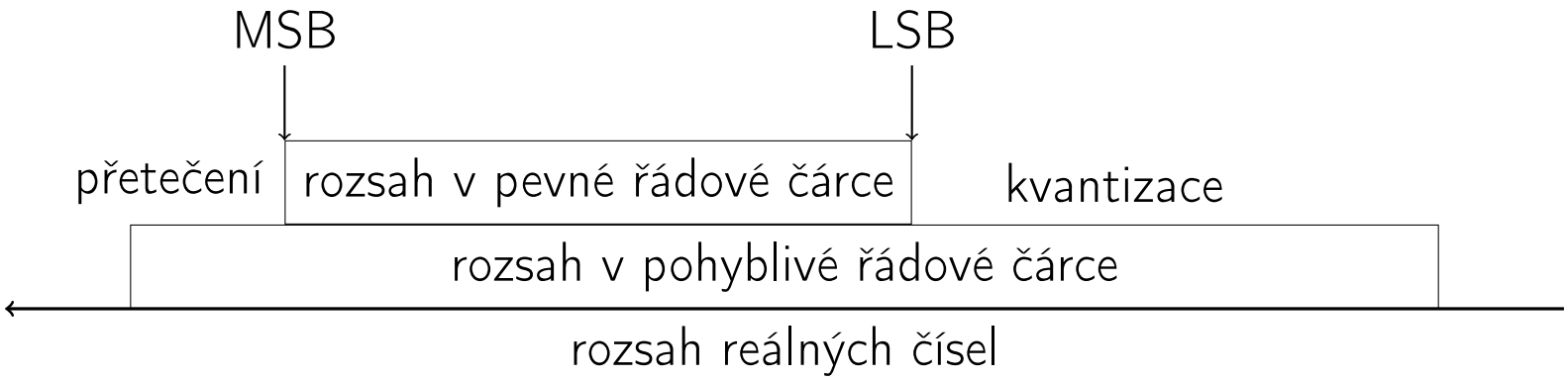
- Useknutí
  - spodní slovo vynulujeme, ponecháme pouze horní slovo.
- Zaokrouhlení dvojkového doplňku
  - k nejvýznamnějšímu bitu spodního slova přičteme 1 a poté spodní slovo vynulujeme,
  - dochází k zaokrouhlování jedním směrem a celkový výsledek by mohl být vychýlený,
  - častěji se používá konvergentní zaokrouhlování.
- Konvergentní zaokrouhlení
  - zaokrouhlení k nejbližšímu sudému číslu,
  - k nejvyššímu bitu spodního slova připočteme 1 a poté spodní slovo vynulujeme,
  - pokud je ve spodním slově nastavený pouze nejvýznamnější bit (ostatní jsou nulové), pak 1 přičteme jen v tom případě, že nejnižší bit horního slova je také nastavený,
  - pokud je ve spodním slově nastavený pouze nejvýznamnější bit (ostatní jsou nulové) a nejméně významný bit horního slova je nulový, pak 1 nepřičítáme.

Zaokrouhlení záporných čísel ve dvojkovém doplňku

Pozor na jednoduchá makra: `#define FLOAT2FRAC(x) ((int)(32768*(x)+0.5))` Příklad zaokrouhlení:

-0,34375	-0,34375
0.010 1	0.010 1
doplněk	zaokrouhlení
1.101 0	0.000 1
0.000 1	0.011 0
1.101 1	doplněk
zaokrouhlení	1.100 1
0.000 1	0.000 1
1.110	1.101
-0,25	-0,375

Rozsah různých formátů



MSB váha nejvýznamnějšího bitu (Most Significant Bit),

LSB váha nejméně významného bitu (Least Significant Bit).

Formáty vyjádření čísel

Důležité vlastnosti formátů:

- pevná řádová čárka
  - menší přesnost, menší rozsah,
  - jednodušší aritmetická logická jednotka,
  - horší návaznost na vyšší programovací jazyky,
  - levnější (v dnešní době je spíše výhodou vyšší výpočetní výkon).
  - MOTOROLA DSP56000, DSP56300, Texas Instruments TMS320C5510, TMS320C6416,...
- pohyblivá řádová čárka
  - větší přesnost,
  - složitější aritmetická logická jednotka,
  - lepší návaznost na vyšší programovací jazyky,
  - dražší (v dnešní době je nevýhodou spíše menší výpočetní výkon),
  - MOTOROLA DSP96000, Texas Instruments TMS320C6711,...

Srovnání pevné řádové čárky a plovoucí řádové čárky

Formát	Rozsah čísel	Dynamický rozsah	Přesnost
Pevná řádová čárka – 16 bitů			
Celé BZ	0 až 65 535	96 dB	1
Celé SZ	−32 768 až 32 767	90 dB	1
Zlomkové BZ	0 až 0,99998474	96 dB	2^{-16}
Zlomkové SZ	−1 až 0,99998474	90 dB	2^{-15}
Plovoucí řádová čárka			
Základní přesnost	$1{,}18 \cdot 10^{-38}$ až $3{,}4 \cdot 10^{38}$	1 530 dB	2^{-23}
Dvojitá přesnost	$2^{-1\,022}$ až $2^{1\,024}$	12 282 dB	2^{-52}

BZ – bez znaménka
SZ – se znaménka

Požadovaný dynamický rozsah a přesnost

typ signálu	dynamický rozsah	počet bitů
řeč	50 – 60 dB	>13
hudba	> 110 dB	>16 (24)