# 18. Číselné soustavy a převody mezi nimi. Způsoby kódování čísel s pevnou a s pohyblivou řádovou tečkou. Kódování záporných čísel

## Číselné soustavy a převody mezi nimi

S číslicovými signály je třeba provádět nejenom logické ale i aritmetické operace. Abychom porozuměli těmto operacím, uvedeme zde stručné vysvětlení způsobu převodů mezi desítkovou, binární a hexadecimální soustavou, které se v číslicových obvodech používají nejčastěji. Libovolné kladné číslo F(Z) lze zapsat pomocí mnohočlenu ve tvaru:

$$F(Z) = \sum_{i=0}^{m-1} a_i Z^i$$

Kde F(Z) je číslo vyjádřené v číselné soustavě o základu Z, ai jsou číselné koeficienty, m je počet řádových míst. Pro jednoduchost v číselných soustavách zapisujeme pouze koeficienty ai. V praxi potom často řešíme problém, který koeficient vyjadřuje nejvyšší řád a který nejnižší. Zpravidla se řídíme konvencí, která říká, že nejvyšší řád je vlevo a nejnižší vpravo. U dvojkové soustavy nejvyšší řád označujeme jako MSB (Most Significant Bit) a nejnižší řád jako LSB (Least Significant Bit).

Tab. 1 : Vyjádření čísel od 0 do 15 v různých soustavách

| Číslo (dekadicky) | Základ 2 | Základ 8 | Základ 16 |
|---|---|---|---|
| 0 | 0000 | 00 | 0 |
| 1 | 0001 | 01 | 1 |
| 2 | 0010 | 02 | 2 |
| 3 | 0011 | 03 | 3 |
| 4 | 0100 | 04 | 4 |
| 5 | 0101 | 05 | 5 |
| 6 | 0110 | 06 | 6 |
| 7 | 0111 | 07 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |

**2–> 10**

Přepočet čísla z libovolné soustavy do soustavy se základem 10 provedeme dosazením do mnohočlenu, uvedeného výše.

Př. 3

***Převeďme číslo 11001(2) do dekadické soustavy***

*11001(2) = 1×24 +1×23 + 0×22 + 0×21 + 1×20 = 16 + 8 + 1 = 25(10).*

**10 -> 2**

Přepočet z desítkové soustavy do ostatních soustav se provádí pomocí algoritmu postupného dělení čísla základem nové soustavy.

Př. 4

***Převeďme číslo 25(10) do dvojkové soustavy.***

*Řešení:*

| | | | | |
|---|---|---|---|---|
| *25 : 2 = 12,* | *zbytek 1* | ... | $a_0$ | = 1 |
| *12 : 2 = 6,* | *zbytek 0* | ... | $a_1$ | = 0 |
| *6 : 2 = 3,* | *zbytek 0* | ... | $a_2$ | = 0 |
| *3 : 2 = 1,* | *zbytek 1* | ... | $a_3$ | = 1 |
| *1 : 2 = 0,* | *zbytek 1* | ... | $a_4$ | = 1 |

Při dělení získáváme zbytky, které reprezentují výsledné bity binárního čísla postupně od LSB. Postupné dělení provádíme tak dlouho, až dostaneme nulový výsledek částečného dělení. Výsledek: 25(10) = 11001(2).

**2 -> 8**

Přepočet mezi dvojkovou, osmičkovou a šestnáctkovou soustavou. Využíváme toho, že 8 i 16 jsou mocninou čísla 2. Z toho vyplývá, že jeden řád osmičkové soustavy je representován třemi místy dvojkovými, jedno místo šestnáctkové čtyřmi místy dvojkovými a opačně.

***Převeďte číslo 1010111111010001(2) do šestnáctkové a osmičkové soustavy.***
*Pro převod do šestnáctkové soustavy rozdělíme číslo na čtveřice od nejnižšího řádu a čtveřice nahradíme šestnáctkovou cifrou.*

| | | | | |
|---|---|---|---|---|
| *0001* | *0101* | *1111* | *1101* | *0001* |
| *1* | *5* | *F* | *D* | *1* |

*Výsledek: 1010111111010001(2) = 15FD1(16)*

**2 -> 16**

*Pro převod do osmičkové soustavy rozdělíme číslo na trojice od nejnižšího řádu a ty pak nahradíme osmičkovou cifrou*

| | | | | | |
|---|---|---|---|---|---|
| *010* | *101* | *111* | *111* | *010* | *001* |
| *2* | *5* | *7* | *7* | *2* | *1* |

*Výsledek:1010111111010001(2) = 257721(8).*

**Další možnosti kódování**

Doposud jsme hovořili o binární, oktalové, dekadické a hexadecimální číselné soustavě. Tyto soustavy spolu s pravidly, která říkají, jaké informace jsou přiřazeny jednotlivým číslům, nazýváme kódy. Příkladem kódu může být tzv. doplňkový kód, který jsme využili při odečítání binárních čísel.

## *Inverzní kód*

- kladná čísla jsou reprezentována normálním způsobem
- je-li číslo záporné provede se na klasickou reprezentaci čísla bitová negace
- existují dvě reprezentace nuly +0 a -0
- např.: $123_{(10)}$ $01111011_{(2)}$
  $-123_{(10)}$ $10000100_{(2)}$
  $0_{(10)}$ $00000000_{(2)}$
  $-0_{(10)}$ $11111111_{(2)}$

Pro snazší práci s čísly desítkové soustavy byl vytvořen BCD (Binary Coded Decimal) kód. Tento kód slouží k zobrazení tzv. desítkových číslic, tj. čísel 0, 1, ..., 9. Tento kód je 4bitový, neboť k zakódování každé z deseti číslic vyžaduje 4 bity. V Tab. 3 je tento kód uveden.

Tab. 3 Tabulka BCD kódu

| Znak | Obraz | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 |

V dalším textu popíšeme ještě kód Grayův. Tento kód má tu vlastnost, že sousední čísla se liší pouze v jednom bitu. Této výhodné vlastnosti je využito např. při sestavování Karnaughovy mapy v kapitole (3.1.3). Tabulka Grayova kódu pro čísla od 0 do 15 je uvedena v Tab. 4.

Tab. 4 Grayův kód

| Znak | Obraz | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 1 | 0 |
| 5 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 0 | 1 |
| 7 | 0 | 1 | 0 | 0 |
| 8 | 1 | 1 | 0 | 0 |
| 9 | 1 | 1 | 0 | 1 |
| 10 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 12 | 1 | 0 | 1 | 0 |
| 13 | 1 | 0 | 1 | 1 |
| 14 | 1 | 0 | 0 | 1 |
| 15 | 1 | 0 | 0 | 0 |

Kód 1 z N má tu vlastnost, že obraz je tvořen N-1 nulami a jednou jedničkou, která je na pozici dané vstupním znakem. Například číslo 3 se zobrazí v kódu 1 z 8 jako vektor 00010000. Kód se využívá například u dekodéru popsaného v Tab. 5.

Tab. 5 Kód 1 z N

| Znak | Obraz | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 |

Důležitou skupinou kódů jsou alfanumerické kódy. Tyto kódy zobrazují abecedně číslicovou informaci pomocí binárních čísel. Alfanumerické kódy se začaly používat pro přenos zpráv a byly pro tento účel normalizovány. Nejčastěji se používají kódy osmibitové, tzn. každý znak, který se přenáší, je kódován osmi bity. Velmi rozšířený je kód ASCII (viz Tab. 6), který je uzpůsoben k přenosu anglické abecedy a dalších běžných znaků, je nevhodný k přenosu znaků české abecedy. Z tohoto důvodu byl upraven pro přenos těchto českých znaků, čímž vznikl kód Kamenických, později LATIN 2.

## Způsoby kódování čísel s pevnou a s pohyblivou řádovou tečkou

**S pevnou**

V pevné řádové čárce bude číslo A rovno součtu součinů:

$$A = a_n 2^n + a_0 2^0 + a_{-1} 2^{-1} + \cdots + a_{-m} 2^{-m}$$

kde $a_i$ je binární číslice nabývající dvou hodnot: 0 a 1. Číslo bude vyjádřeno pomocí příslušného počtu binárních číslic $a_n$ … $a_0$, $a_{-1}$ … $a_{-m}$. Mezi řádem „0“ a „-1“ píšeme „desetinnou“ čárku. Záporná čísla zobrazujeme pomocí některého z těchto kódů: přímý kód, kód s posunutou nulou, kód jednotkového doplňku a kód dvojkového doplňku.

## Kódování záporných čísel

V případě, kdy chceme rozlišit kladná a záporná čísla ve dvojkové soustavě použijeme znaménko mínus stejně jako v soustavě dekadické. Pro kódování znaménka v počítači se používají různé způsoby. Nejčastěji jsou to tyto tři:

- Vyhrazení znaménkového bitu
- Přičtení konstanty
- Pomocí dvojkového doplňku

V případě, kdy jeden bit (většinou ten nejvýznamnější) představuje znaménko, ostatní bity představují absolutní hodnotu daného čísla. Konvence říká, že v případě, že znaménkový bit je roven jedné, jedná se o záporné číslo a je-li roven nule, jedná se o číslo kladné. Znaménkový bit nám snižuje rozsah čísel, která můžeme zakódovat daným počtem bitů. Např. pomocí osmi bitů můžeme vyjádřit čísla v rozsahu -127 až 127 (přičemž máme kladnou a zápornou nulu).

**S pohyblivou**

Pro zobrazení reálných nebo příliš velkých celých čísel je výhodné použít pohyblivou řádovou čárku. Potom číslo A bude ve tvaru $A = M.2^E$, kde M je mantisa a E exponent. Přesnost čísla A závisí na počtu číslic mantisy, rozsah zobrazení závisí na počtu číslic exponentu. Reálná čísla jsou zobrazována buď v jednoduché přesnosti pomocí 32 bitů (1 bit znaménko mantisy, 23 bitů mantisa v přímém kódu, 8 bitů exponent v kódu s posunutou nulou) anebo v dvojnásobné přesnosti 64 bitů (znaménko, 52 bitů mantisa, 11 bitů exponent). Pro vyčíslení výsledků aritmetických operací v pohyblivé řádové čárce jsou využity základní obvody, které zpracovávají aritmetické operace v pevné řádové čárce. Pro sčítání/odčítání je třeba srovnat exponenty a sečíst/odečíst mantisy. Pro násobení je třeba sečíst exponenty a vynásobit mantisy. Pro dělení je třeba odečíst exponenty a vydělit mantisy. Výsledek každé operace je třeba normalizovat, tj. vyjádřit takovým výrazem $M.2^E$, který zajistí na nejvýznamnějším bitu mantisy hodnotu log. 1.

### Přičtení konstanty

Při využití tohoto způsobu kódování záporných binárních čísel dochází k posouvání nuly tím, že ke každému číslu přičteme vhodnou konstantu. Např. pro čísla v rozsahu -127 až 128 bychom volili konstantu 127.

### Dvojkový doplněk

Dvojkový doplněk binárního čísla získáme jako jeho inverzi zvětšenou o jedničku. Kódování záporných čísel pomocí dvojkového doplňku nám umožňuje snadno provádět odčítání ve dvojkové soustavě (přičtením záporného čísla).

Rozsah čísel, který je možné pomocí dvojkového doplňku zakódovat je $(-2^{n-1})$ až $(2^{n-1} - 1)$, to znamená, že pomocí např. 4 bitů můžeme zakódovat čísla v rozsahu: -8 až 7, pomocí 8mi bitů čísla v rozsahu: -128 až 127 atd.

Tab. 2 Vyjádření čísel ve dvojkovém doplňku

| AA | BB |
|---|---|
| 0111 | 7 |
| 0110 | 6 |
| 0101 | 5 |
| 0100 | 4 |
| 0011 | 3 |
| 0010 | 2 |
| 0001 | 1 |
| 0000 | 0 |
| 1111 | -1 |
| 1110 | -2 |
| 1101 | -3 |
| 1100 | -4 |
| 1011 | -5 |
| 1010 | -6 |
| 1001 | -7 |
| 1000 | -8 |

Při převodu kladného čísla do dvojkového doplňku musíme nejprve kladné binární číslo doplnit zleva nulami na správný počet bitů (např. číslo 510 = 1012 doplníme na 01012). V případě že tak neučiníme, nedostaneme správnou hodnotu.

**Př. 2**

***Vypočtěte dvojkový doplněk k číslu 11102 =1410.***
*Číslo 11102 nejprve doplníme na potřebný počet bitů (pro vyjádření čísla -14 potřebujeme 5 bitů) a tím dostaneme 01110. Inverze tohoto čísla je 10001. Přičtením jednotky v LSB dostaneme číslo $10010_2 = -14_{10}$.*

**Operace ve dvojkové soustavě**

a) Sčítání: Při sčítání se příslušné koeficienty $a_i$ sčítají obdobně jako v desítkové soustavě:

00011011 (2)

00110010 (2)

01001101 (2)

b) Odčítání: Ručně provádíme odčítání v číselných soustavách tak, jak jsme zvyklí ze soustavy desítkové, respektujeme při tom změny řádu v dané soustavě.

01001011 (2)

-00110010 (2)

00011001 (2)

Výpočetní technika používá pro odčítání převodu na přičítání dvojkového doplňku menšitele. Dvojkový doplněk k menšiteli 00110010(2) získáme jako jeho inverzi (11001101) zvětšenou o 1 (11001110). Dvojkový doplněk reprezentuje záporné číslo o stejné absolutní hodnotě, jako bylo číslo původní. Dvojkový doplněk přičteme k menšenci:

01001011 (2)

11001110 (2)

00011001 (2)

Nula v řádu MSB signalizuje, že výsledek odečítání je kladné číslo. V případě, že výsledek odečítání by byl záporný, získali bychom výsledek s MSB rovným jedné. V případě, že bychom chtěli získat absolutní hodnotu tohoto záporného čísla, určili bychom ji jako dvojkový doplněk výsledku.

**Př. 6**

***Ve dvojkové soustavě proveďte operaci odečtení čísel 1-7.***
*Pro binární kódování čísel v rozsahu od –7 do +7 využijeme 4 bitů. Při využití dvojkového doplňku čísla 7 provedeme následující výpočet:*

*0001 (2)*
*1001 (2)*
*1010 (2)*

*Výsledek 1010 je vyjádřením čísla –6. Absolutní hodnotu získáme jako jeho inverzi (0101) a přičtení jednotky (0110).*