



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PSK2-20

Název školy:	Vyšší odborná škola a Střední průmyslová škola, Božetěchova 3
Autor:	Ing. Marek Nožka
Anotace:	Způsoby kódování celých a reálných čísel
Vzdělávací oblast:	Informační a komunikační technologie
Předmět:	Počítačové sítě a komunikační technika (PSK)
Tematická oblast:	Vrstvy protokolu TCP/IP
Výsledky vzdělávání:	Žák vytvoří dvojkový doplněk, popisuje výhody a nevýhody pevné a plovoucí řadové čírký
Klíčová slova:	dvojkový doplněk, pevná a plovoucí řadová čárka, BCD
Druh učebního materiálu:	Video-prezentace, Online vzdělávací materiál
Typ vzdělávání:	Střední vzdělávání, 3. ročník, technické lyceum
Ověřeno:	VOŠ a SPŠE Olomouc; Třída: 3L
Zdroj:	Vlastní poznámky, Wikipedia, Wikimedia Commons

Kódování čísel

Přímý kód

- Slouží k uchování **nezáporných celých čísel**
- Jedná se o pouhý převod čísla do dvojkové soustavy => získáme řadu nul a jedniček

Výhoda:

maximální jednoduchost

Nevýhoda:

neumožňuje uchování záporných čísel

- V praxi tento kód v řadě případů dostačuje např. počítadlo opakování cyklu apod.

Doplňkový kód

- Slouží pro uchovávání celých čísel včetně znaménka.
- Nejvyšší významový bit (**MSB**) má funkci tzv. znaménkového bitu
 - 0 = kladné číslo
 - 1 = záporné číslo
- Vždy musíme znát celkový počet bitů, který máme k dispozici pro uchování čísla.
- Kladná čísla se uchovávají stejně jako u přímého kódu, tj. ve dvojkové soustavě.
- Pro kladná čísla je doplňkový kód kompatibilní s přímým kódem.
- Záporná čísla se uchovávají v podobě tzv. dvojkového doplňku.

Dvojkový doplněk

1. Kladné číslo vyjádříme ve dvojkové soustavě.
2. Všechny bity negujeme.
3. Přičteme jedna

6	=	0110
negace		1001
	+	0001

		1010

Pokud postup opakujeme, tj.

1. Všechny bity negujeme.
2. Přičteme jedna

		1010
negace		0101
	+	0001

6	=	0110

... dostaneme opět původní číslo.

Převodní tabulka pro 4-bitový doplňkový kód

0000	=	0
0001	=	1
0010	=	2
0011	=	3

0100	=	4
0101	=	5
0110	=	6
0111	=	7

1000	=	-8
1001	=	-7
1010	=	-6
1011	=	-5

1100	= -4
1101	= -3
1110	= -2
1111	= -1

- Výhodou tohoto řešení je možnost provádění operací součtu a rozdílu bez ohledu na znaménka operandů či výsledku -- výsledek vždy vyjde správně vč. znaménka

$$4 + (-6) = 4 - 6$$

0100
+ 1010

1110 = -2

$$7 + (-6) = -6 + 7$$

1010
+ 0111

10001 = 1 (berme v úvahu jen 5 bitů)

BCD kód

- BCD kóduje celé číslo v desítkové soustavě přímo po jednotlivých číslicích
- Každé číslici v desítkové soustavě odpovídají 4 bity

$$3805_{10} = (0011\ 1000\ 0000\ 0101)_{BCD}$$

- Při zpětném převodu je nutné hlídat, aby každá čtveřice bitů odpovídala platnému kódu BCD
 - Např. bity 1101 (=13) nepatří do kódu BCD tj. není to desítková číslice
 - Ne každá posloupnost nul a jedniček odpovídá nějakým datům v BCD kódu.
- V BCD nelze uchovávat záporná čísla.
- BCD není vhodný pro počítání (někdy se dělají tzv. BCD korekce)
 - např. $9 + 2 = 1001 + 0010 = 1011$... nepatří do BCD !!!
- Vhodné pro uložení čísel, s nimiž se neprovádějí žádné výpočty.

Pevná řadová čárka

- <http://www.root.cz/clanky/fixed-point-arithmetic/>
- Slouží k uchování reálných (desetiných) čísel.
- Čísla jsou v paměti uchovávána jakoby se jednalo o celé číslo,

přičemž v určitém místě se předpokládá desetinná čárka.

- Je stanoven pevný počet bitů celé části a pevný počet bitů desetinné části čísla.

Např. pro číslo 11001, 011 se uloží do 16 bitů

0001 1001 0110 0000 (uprostřed je pomyslná desetinná čárka)

výhoda:

Snadné výpočty +,- (stejně jako s celými čísly => vysoká rychlost výpočtů).

nevýhoda:

V praxi malý rozsah a málo desetinných míst => málo používaný.

Plovoucí řadová čárka

- Pohyblivá řadová čárka
- Slouží k uchování reálných (desetinných) čísel.
- Číslo se uchovává ve tvaru obdobném tzv. vědeckému zápisu:

- $2500 = 2.5 \cdot 10^3$
- $0.00001589 = 1.589 \cdot 10^{-5}$

- Číslo se uchovává ve tvaru:

$$X = M_x \cdot 2^{E_x}$$

- M_x -- tzv. **mantisa**, před desetinnou čárkou se předpokládá vždy jedno číslo (doplňkový kód).
- E_x -- **exponent** je to vždy kladné nebo záporné celé číslo.
- Ukládá se také znaménkový bit.
- Někdy se ukládá jen desetinná část mantisy, protože v binární soustavě je před des. čárkou (téměř) vždy 1
 - ušetří se 1 bit => vyšší přesnost, o 1 bit delší mantisa
 - problém s uložením nuly -- ukládá se jako $1.00 * 2^{-MAX}$
 - nejedná se o pravou nulu ale o nejmenší možné číslo 0, 000. . . 0001
 - kladná nula vs. záporná nula

výhoda:

Možnost uložení velmi velkých i velmi malých čísel.

nevýhoda:

Náročnější implementace matematických operací.