

Číselné soustavy, binární aritmetické operace, převody mezi soustavami (10, 2, 16), doplňkový a adiktivní kód, zobrazení čísla bez a se znaménkem, přetečení

Číselné soustavy

Nepoziční

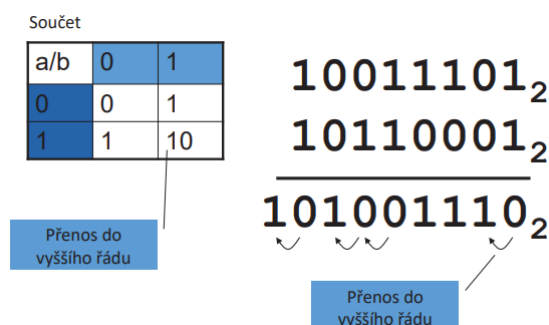
- Zářezky – II (2); III (3); IIII (4); IIIII (5)
- Římské číslice – I (1); V (5); X (10); L (50); C (100); D (500); M (1000)

Poziční

- Dvojková (binární)
 - Každá číslice odpovídá na n-té mocnině čísla dvě, kde n je pozice dané číslice v zapsaném čísle
 - Logické stavy – zapnuto: 1; vypnuto: 0
 - Digitální elektronické obvody
- Osmičková (oktalová)
- Desítková (dekadická)
- Dvanáctková (tucet)
 - Málo používaná
- Šestnáctková (hexadecimální)
 - Použity číslice 1-9 a A-F
 - Např. pro zápis barvy, registrů
- Šedesátková
 - K měření času

Binární aritmetické operace

Sčítání



Odčítání

$$\begin{array}{r} 0111 \\ -0010 \\ \hline 0111 \\ +1101 \text{ - Inverzní číslo} \\ \hline 10100 \\ \text{ } \rightarrow +1 \\ \hline 0101 \end{array}$$

Násobení

- Stejně jako pro desítkovou soustavu

$$\begin{array}{r} 1001_2 \\ 110_2 \\ \hline 0000 \\ 1001 \\ 1001 \\ \hline 110110_2 \end{array}$$

Dělení

- Stejně jako pro desítkovou soustavu
- Kolikrát se dělitel vejde do dělece (ANO/NE)

$$\begin{array}{r} 10101_2 : 110_2 = 011_2 \\ 1010 \\ -110 \\ \hline 1001 \\ -110 \\ \hline 011 \end{array}$$

Převody mezi soustavami

Převod z dvojkové do desítkové

- Definice jednotlivých vah v binárním čísle a následné sčítání

Příklad

$$10100110b = 166d$$

binární číslo	1	0	1	0	0	1	1	0
pozice čísla	7	6	5	4	3	2	1	0
mocnina čísla dvě	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
hodnoty mocnin (váhy)	128	64	32	16	8	4	2	1
krát binární číslice	x 1	x 0	x 1	x 0	x 0	x 1	x 1	x 0
výsledek součinu	128	0	32	0	0	4	2	0

Převod z desítkové do dvojkové

- Opakované dělení číslem dvě (se zbytkem), pokud zbyde jednička, zapisování zprava doleva

Příklad

```
52:2 = 26 → zbytek: 0 lsb
26:2 = 13 → zbytek: 0
13:2 = 6 → zbytek: 1
6:2 = 3 → zbytek: 0
3:2 = 1 → zbytek: 1
1:2 = 0 → zbytek: 1 msb
```

Výsledek: 110100

Převod z desítkové do šestnáctkové (a naopak)

- Funguje stejně jako dvojkové
- Při převodu z desítkové do šestnáctkové dělím 16
- Při převodu z šestnáctkové do desítkové definuju váhy 16

Převod z dvojkové do šestnáctkové

- Dvojkové číslo rozdělím po 4bitech a každou čtveřici převedu zvlášť (na desítkové, potom na šestnáctkové, např. 1011 -> 11 -> B)

Příklad

```
0111 1011 | 7 11 | 7B(16)
             | → 7 B |
```

Doplňkový kód (dvojková doplněk)

- Zjednodušuje konstrukci ALU díky tomu, že sčítání a odečítání provádí pro čísla se znaménkem stejně jako pro čísla bez znaménka
- Díky dvojkovému doplňku dokážeme zobrazit záporné číslo
- Násobení čísel v doplňkovém kódu – převést na kladná, vynásobit a pak podle znaménka výsledek upravit.

```
11101100 (-20)
00010011 negace (1->0 a 0->1, po bitech)
00010100 +1 => 20
```

Adiktivní kód

- Reprezentace celých čísel
- Převod mezi doplňkovým a adiktivní spočívá v nejvýznamnějším bitu

Posun (offset)

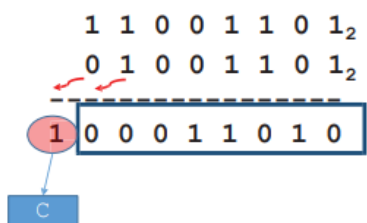
- takový, aby nejnižší číslo odpovídalo nule a nejvyšší číslo odpovídalo největšímu číslu
- pro datové typy o n bitech obvykle 2^{n-1}

Desítkový zápis	Aditivní kódování s posunem 8	Dvojkový doplněk
7	1111	0111
6	1110	0110
5	1101	0101
4	1100	0100
3	1011	0011
2	1010	0010
1	1001	0001
0	1000	0000
-1	0111	1111
-2	0110	1110
-3	0101	1101
-4	0100	1100
-5	0011	1011
-6	0010	1010
-7	0001	1001
-8	0000	1000

Přetečení (carry a overflow)

Celá kladná čísla

- Jev, který nastane, pokud nelze výsledek operace vyjádřit v daném číselném formátu
- Pokud přeteče výsledek operace přes nejvyšší bit, nastaví se příznak přenosu (carry)



Čísla ve dvojkovém kódu

- **Výsledek je nesprávný**
- Pokud přeteče výsledek operace přes nejvyšší bit, nastaví se příznak přenosu (overflow)

Nenastane přetečení (overflow):

$$\begin{array}{r} 01001101_2 \\ + 00001101_2 \\ \hline 10101101_2 \end{array} \quad \begin{array}{l} 77 \\ 13 \\ 90 \end{array}$$

Nastane přetečení (overflow):

$$\begin{array}{r} 01001101_2 \\ + 01001111_2 \\ \hline 10011101 \end{array} \quad \begin{array}{l} 77 \\ 79 \\ 157 \end{array}$$