Struktura a architektura počítačů

Katedra číslicového návrhu
Fakulta informačních technologií
České vysoké učení technické

Zobrazení dat v číslicovém počítači

BI-SAP, březen 2021

Obsah

- Poziční číselné soustavy a převody
  - Dvojková soustava, převod do desítkové
  - Šestnáctková soustava, převod do dvojkové
- Aritmetika (1)
  - Sčítání, odčítání
  - Násobení
- Řádová mřížka
- Zobrazení čísel se znaménkem (tedy i záporných)

Čerpáno z podkladů pro předmět z FELu Tomáše Brabce, Miroslava Skrbka, úpravy pro BI-SAP Hana Kubátová

Pluháček, A., „*Projektování logiky počítačů*,“ skripta, Praha, ČVUT, 2000, ISBN 80-01-02145-9

Poziční číselné soustavy

- Určeny bází (základem) z, $z \in N$, $z \geq 2$
- Soustava s bází z … z-adická
- Nejčastěji používané soustavy:

$z = 2$	dvojková (binární)
$z = 10$	desítková (dekadická)
$z = 16$	šestnáctková (hexadecimální)

Zápis čísla v z-adické soustavě

řádová čárka

$$A_z = (\underbrace{a_n\ a_{n-1} \ldots a_1\ a_0}_{\text{celá část}}\ ,\ \underbrace{a_{-1}\ a_{-2} \ldots a_{-m}}_{\text{zlomková část}})_z,\quad n, m \in N$$

základ soustavy

a_i … z-adická cifra (číslice) na pozici i

a_i … hodnota číslice a_i, $0 \leq \mathrm{a}_i < z$

i … řád číslice (řádové místo, pozice), určuje její váhu $v_i = z^i$

n … nejvyšší řád s nenulovou číslicí

-m … nejnižší řád s nenulovou číslicí

Hodnota čísla A_z :

$$\mathrm{A} = v(A_z) = \sum_{-m}^{n} \mathrm{a}_i \cdot v_i = \sum_{-m}^{n} \mathrm{a}_i \cdot \mathrm{z}^i$$

Dvojková soustava

- Základ (báze) soustavy $z = 2 \Rightarrow$ zápis čísla tvořen posloupností 0 a 1

Příklad

$$
\begin{array}{lcccccccc}
v_i & \ldots 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} \\
 & 1 & 0 & 0 & 1 & 1, & 1 & 0 & 1_2
\end{array}
$$

$$v(A) = 2^4+2+1+1/2+1/8 = \underbrace{19{,}625}$$

Toto je ekvivalentní zápis čísla A v desítkové soustavě.

- Určení hodnoty čísla ≈ převod do desítkové soust., tj. **Dvojková → Desítková**

Desítková → Dvojková (celá část)

- **Postupným dělením celé části číslem 2** (tj. základem dvojkové soustavy)

Př. Převeďte číslo 57_{10} do dvojkové soustavy.

$57_{10} \approx A_2$

$A_2 = 111001_2$

57 : 2 = 28	*zbytek*	1	… a_0
28 : 2 = 14	*zbytek*	0	
14 : 2 = 7	*zbytek*	0	
7 : 2 = 3	*zbytek*	1	
3 : 2 = 1	*zbytek*	1	
1 : 2 = 0	*zbytek*	1	… a_5

Pozn. Zápis čísla odpovídá posloupnosti zbytků brané v opačném pořadí.

Desítková → Dvojková (zlomková část)

- **Postupným násobením zlomkové části číslem 2** (tj. základem dvojkové soustavy)

Př. Převeďte číslo $0{,}65625_{10}$ do dvojkové soustavy.

$$0{,}65625_{10} \approx A_2$$

$$A_2 = 0{,}10101_2$$

$$\begin{array}{llll}
0{,}65625 & \cdot\, 2 = 1 & ,\ 3125 & \dots\ a_{-1} \\
0{,}3125 & \cdot\, 2 = 0 & ,\ 625 & \\
0{,}625 & \cdot\, 2 = 1 & ,\ 25 & \\
0{,}25 & \cdot\, 2 = 0 & ,\ 5 & \\
0{,}5 & \cdot\, 2 = 1 & ,\ 0 & \dots\ a_{-5}
\end{array}$$

Příklady:

1. $1101\ 0001{,}11_2$ → $209{,}75_{10}$
2. $111\ 1111_2$ → 127_{10}

   nebylo by lepší převést $1000\ 0000_2$??? .. 128 – 1 = 127

3. $1{,}011001_2$ → $1{,}390625_{10}$
4. $147{,}15625_{10}$ → $1001\ 0011{,}0010\ 1_2$
5. $1345{,}125_{10}$ → $101\ 0100\ 0001{,}001_2$

Přesnost, zobrazitelnost

$0{,}1_{10} \rightarrow 0{,}000\,11\,00\,11\,00\,11\,00\,\ldots\ldots_2$

0,1 . 2 = 0,2

0,2 . 2 = 0,4

0,4 . 2 = 0,8

0,8 . 2 = 1,6

0,6 . 2 = 1,2

0,2 . 2 = 0,4

0,4 . 2 = 0,8

0,8 . 2 = 1,6

0,6 . 2 = 1,2

....

Mocniny dvou

n	2^n	Dec.
0	2^0	1
1	2^1	2
2	2^2	4
3	2^3	8
4	2^4	16
5	2^5	32
6	2^6	64
7	2^7	128

n	2^n	Dec.
8	2^8	256
9	2^9	512
10	2^{10}	1 024
11	2^{11}	2 048
12	2^{12}	4 096
13	2^{13}	8 192
14	2^{14}	16 384
15	2^{15}	32 768
16	2^{16}	65 536

n	2^n	Dec.
20	2^{20}	1 M
30	2^{30}	1 G
32	2^{32}	4 G
40	2^{40}	1 T
-1	2^{-1}	0,5
-2	2^{-2}	0,25
-3	2^{-3}	0,125
-4	2^{-4}	0,0625

Toto je důležité!

Šestnáctková soustava

- Zápis čísla tvořen ciframi 0..9 a A..F

Hex.	Dec.	Bin.
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

Hex.	Dec.	Bin.
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Toto se hodí znát zpaměti!

Dvojková ←→ Šestnáctková

- Jsou to příbuzné soustavy, tj. $z_{16} = 16 = 2^4 = {z_2}^4$
- $\Rightarrow$ Jedna cifra v z_{16} odpovídá čtyřem cifrám v z_2
  $\Rightarrow$ **Mezi zápisy v soustavách z_{16} a z_2 je pouze formální rozdíl.**

Př. Převeďte čísla mezi příbuznými soustavami:

a) $100\ 1101{,}0101\ 1_2$

$$\underbrace{0100}_{4}\ \underbrace{1101}_{D}\ ,\ \underbrace{0101}_{5}\ \underbrace{1000}_{8_{16}}{}_2$$

b) $734{,}051_{16}$

$$0111\ 0011\ 0100{,}0000\ 0101\ 0001_2$$

Příklady

$1\ 0110\ 1011,0101\ 11_2 \rightarrow 16B,5C_{16}$

1. $111\ 0101\ 1101\ 0100_2 \rightarrow 75D4_{16}$
2. $0,0011\ 0101\ 1100\ 1_2 \rightarrow 0,35C8_{16}$

3. $12A5F,1_{16} \rightarrow 1\ 0010\ 1010\ 0101\ 1111,0001_2$
4. $F563D,8_{16} \rightarrow 1111\ 0101\ 0110\ 0011\ 1101,1_2$
5. $0,98736_{16} \rightarrow 0,1001\ 1000\ 0111\ 0011\ 0110_2$

Sčítání ve dvojkové soustavě

- Základem je součet dvou 1-ciferných čísel

+	0	1
0	0	1
1	1	$^{1}0$

Přenos *do vyššího řádu.*

Př. Sečtěte čísla 0101_2 a 1110_2.

$$
\begin{array}{rr}
 & 0\,1\,0\,1 \\
+ & 1\,1\,1\,0 \\
\hline
 & {}^{1}1\,{}^{1}0\,{}^{0}0\,{}^{0}1\,1
\end{array}
$$

Přenos z řádu i *se sčítá s ciframi v řádu* (i+1).

Pozn. Součtem dvou *N*-ciferných čísel může vzniknout (*N*+1)-ciferné číslo.

Násobení ve dvojkové soustavě

- Základem je součin dvou 1-ciferných čísel

×	0	1
0	0	0
1	0	1

- Více-ciferné násobení se převádí na sčítání

Př. Vynásobte čísla 1110_2 a 101_2.

```
        1 1 1 0
  ×       1 0 1
  -------------
        1 1 1 0     ... 1 × (1 1 1 0)
  +   0 0 0 0       ... 0 × (1 1 1 0)
  + 1 1 1 0         ... 1 × (1 1 1 0)
  -------------
  1 0 0 0 1 1 0
```

Pozn. Součinem *N*- a *M*-ciferného čísla může vzniknout (*N*+*M*)-ciferné číslo.

Řádová mřížka

- **Řádová mřížka určuje formát zobrazitelných čísel** na počítači (tj. definuje nejvyšší řád *n* a nejnižší řád *-m*)

Příklad:

n = 3, -m = 0 n = 0, -m = -3

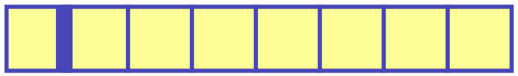
- Základní vlastnosti:
  - **Délka ř.m.** (l) – počet řádů obsažených v ř.m.
  - **Jednotka ř.m.** (ε) – nejmenší číslo zobrazitelné v ř.m. (*nezáporné!*)
  - **Modul ř.m.** (M) – nejmenší číslo, které již v ř.m. zobrazitelné není

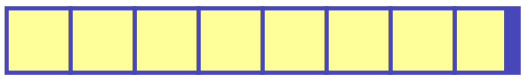
M = 1 0000_2, ε = 1

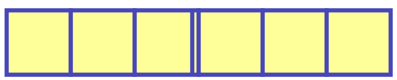
M = $10{,}000_2$, ε = 0,001

Vlastnosti ř.m.

- Určete vlastnosti následujících řádových mřížek (z = 2):

a)  $l = 8$, $M = 2_{10}$, $\varepsilon = (2^{-7})_{10}$

b)  $l = 8$, $M = (2^{8})_{10}$, $\varepsilon = 1$

c)  $l = 6$, $M = (2^{3})_{10}$, $\varepsilon = (2^{-3})_{10}$

- obecně, tj. v závislosti na ***n*** a ***–m***:

$$l = n + m + 1, \quad M = z^{n+1}, \quad \varepsilon = z^{-m}$$

Odečítání ve dvojkové soustavě

- Odčítání ≈ přičítání opačného čísla, vždy v rámci řádové mřížky, tedy modulu M:

Př. Určete rozdíl čísel $10_{10} - 6_{10}$ (ve dvojkové soustavě).

(Odečtení 6_{10} je totéž jako přičtení 26_{10} modulo 32)

Všimněte si volby ř.m. Obě čísla v ní musí být správně zobrazena!

$6_{10} = 00110_2$ → $-6_{10} = 11010_2$

$+10_{10} = 01010_2$

$4_{10} = 00100_2$

	1	1	0	1	0
	0	1	0	1	0
1	0	0	1	0	0

znaménko výsledku

tento přenos ignorujeme (prozatím)

pro M=16:
7-4= 3 ....7+(16-4) ... $7+12/_{M16} = 19/_{M16} = 3$

Úloha: Odečtěte ve dvojkové soustavě.

- Převeďte čísla do dvojkové soustavy (je-li to nutné) a spočítejte jejich rozdíl.

odečtení 10 je totéž jako přičtení 22?
22 + 6 = 28 ... ale má to být -4

1. $6_{10} - 10_{10} = 11\ 100_2$
2. $7_{10} - 7_{10} = 0_2$
3. $1001_2 - 0110_2 = 0\ 0011_2$
4. $F1_{16} - 3_{16} = 0\ 1110\ 1110_2$

Problémy:
velikost řádové mřížky, určení a zobrazení správného výsledku, jestliže používáme jen nezáporná čísla ... jak zobrazíme ta záporná?

Detekce (ne)správného výsledku

Záleží na použitém kódu pro zobrazení čísel se znaménkem a velikosti řádové mřížky, tzn. modulu.

Sčítání a odčítání:

Příklady:

$M=10000_2$ tzn.: zobrazíme 16 čísel, 0 až 15

$(12+7)_{10}$ ..1100 + 0111 = **1**0011 … 3_{10} nebo 19_{10} ?

$(12-7)_{10}$ ..1100 - 0111 = 1100 + 1000+1= **1**0101 …5_{10} nebo 21_{10} ?

1 …. přenos

Odčítání pro nezáporná čísla

pozorování na příkladu M=1000, ε=1:

B=101 $\overline{B}$=010

$B + \overline{B} = 111 = 1000 - 1 = M - 1$

$-B = \overline{B} + 1 - M$

$\boxed{A - B = A + \overline{B} + 1 - M}$

Abychom dostali správný výsledek, musíme mít možnost odečíst modul. Musí vyjít přenos !!

Příklad:

M=10000_2…$(12-7)_{10}$ ..1100 – 0111 ..1100 + 1000+1= **1**0101 …5_{10}

Zobrazení čísel se znaménkem

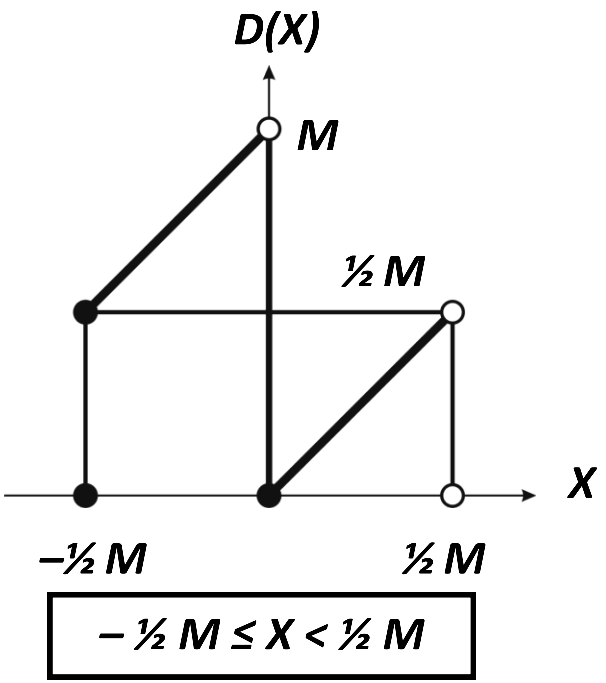
(tedy kladných i záporných)

- Standardní polyadické soustavy ⇒ pouze nezáporná čísla
- Zobrazení záporných čísel ⇒ **číselné kódy**
  - popisují transformaci z omezené množiny celých čísel do omezené množiny nezáporných čísel
- Nejpoužívanější číselné kódy:
  - **přímý (znaménko a absolutní hodnota – sign-magnitude)**
  - **aditivní (s posunutou nulou – biased)**
  - **doplňkový (pro dvojkovou soustavu - 2´s complement)**
  - (inverzní)

Doplňkový kód

Definice:

$$\mathcal{D}(X) = \begin{cases} X, & \text{je-li } X >= 0 \\ M + X, & \text{je-li } X < 0 \end{cases}$$



Příklad – napsat všechna 3 bitová čísla
(M = 1000, ε = 1, l = 3)

X	$D(X)$		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
-4	1	0	0
-3	1	0	1
-2	1	1	0
-1	1	1	1

Znaménko je určeno prvním bitem zleva,
ale tento bit je organickou součástí obrazu !!!

Př. Obrazy čísel +5 a –5 ($z = 2$, $M = 10000_2$, 16_{10}).

nejvyšší bit představuje znaménko

$\mathcal{D}(5) = 5_{10} = 101_2$ $\quad +101_2 \xrightarrow{\mathcal{D}}$ 0 1 0 1

$\mathcal{D}(-5) = 16_{10} + (-5_{10}) = 11_{10} = 1011_2$ $\quad -101_2 \xrightarrow{\mathcal{D}}$ 1 0 1 1

Algoritmus určení obrazu záporného čísla (ve dvojkové soustavě):

1. Zapíšeme číslo X_2 do řádové mřížky.
2. Invertujeme všechny bity.
3. Přičteme jedničku.

Lze rychleji:
Zprava opisuj 0 až do první 1, tu také opiš.
Další bity invertuj.

$5_{10} = 101_2$

0 1 0 1 *... 1. zápis v ř.m.*

1 0 1 0 *... 2. inverze bitů*

+ 0 0 0 1 *... 3. přičtení jedničky*

1 0 1 1 ... $\mathcal{D}(-5_{10})$

Př. Obraz čísla –5 ($z = 2$, $M = 16$).

Doplňkový kód - pokračování

- Obraz záporného čísla X je doplňkem jeho hodnoty do modulu M řádové mřížky

- Příklady:

M = 10 000$_{10}$, ε = 1: M - 25 = 9 975

$-25_{10} \xrightarrow{\mathcal{D}}$ | 9 | 9 | 7 | 5 |

$+0{,}05_{10} \xrightarrow{\mathcal{D}}$ | 0 | 0 | 5 | 0 |

M = 10,000$_{10}$, ε = 0,001:
(1)0 + 0,05 = 0,050
pro – 0,05: 10 – 0,05 = 9,95

M = 1 0000$_2$, ε = 1

$+101_2 \xrightarrow{\mathcal{D}}$ | 0 | 1 | 0 | 1 |

$-0{,}11_2 \xrightarrow{\mathcal{D}}$ | 1 | 0 | 1 | 0 |

M = 10,000$_2$, ε = 0,001:

```
  10,000
–  0,110
--------
+  1,010
```

Sčítání a odčítání v doplňkovém kódu

Příklady:

$M=10000_2$ tzn. zobrazíme 16 čísel, ale nyní v rozsahu -8 až 7

$(7 - 4)_{10}$ .. 7+(-4) ... 0111 + 1100 = **1**0011 ... $\cancel{3}_{10}$ nebo 19_{10} ?

$(4 + 7)_{10}$ .. 0100 + 0111 = 0100 + 0111= 1011 ...$\cancel{11}_{10}$ nebo -5_{10} ?

V podstatě jen sčítáme ....

Ale co teď s přenosem a jak poznáme, že výsledek je správně?

Detekce správného výsledku

Mějme M = 10000 a vyzkoušejme všechny možnosti součtů:

lze zobrazit: celá čísla od – 8 (M-8 = 8 ... 1000) do 7 (0111)

$M = 1\ 0000_2,\ \varepsilon = 1$

Snadný převod na opačné číslo: Zprava opisuj nuly až do první jedničky, tu opiš a další bity invertuj.

viz příklad ze 3. přednášky

Detekce správného výsledku

Mějme M = 10000 a vyzkoušejme všechny možnosti součtů:

lze zobrazit: celá čísla od – 8 (M-8 = 8 … 1000) do 7 (0111)

$M = 1\,0000_2$, $\varepsilon = 1$

1. sčítáme malá kladná: 3+4=7

1

```
      0011
    + 0100
přenosy: 0 0 0 0
 (+)  0111  .... 7₁₀
```

Snadný převod na opačné číslo: Zprava opisuj nuly až do první jedničky, tu opiš a další bity invertuj.

Detekce správného výsledku

Mějme M = 10000 a vyzkoušejme všechny možnosti součtů:

lze zobrazit: celá čísla od – 8 (M-8 = 8 … 1000) do 7 (0111)

$M = 1\ 0000_2,\ \varepsilon = 1$

1. sčítáme malá kladná: 3+4=7

2. sčítáme větší kladná: 5 + 4 = 9
   (správný výsledek je mimo mřížku)

1

```
      0011
  +   0100
přenosy: 0 0 0 0
(+)   0111  .... 7₁₀
```

2

```
      0101
  +   0100
přenosy: 0 1 0 0
(-)   1001  .... -7₁₀   ??
```

Snadný převod na opačné číslo: Zprava opisuj nuly až do první jedničky, tu opiš a další bity invertuj.

Detekce správného výsledku

Mějme M = 10000 a vyzkoušejme všechny možnosti součtů:

lze zobrazit: celá čísla od – 8 (M-8 = 8 … 1000) do 7 (0111)

$M = 1\,0000_2,\ \varepsilon = 1$

1. sčítáme malá kladná: 3+4=7

2. sčítáme větší kladná: 5 + 4 = 9
   (správný výsledek je mimo mřížku)

3. kladné + záporné:
   a) 3 + (-2) = 1, b) 3 + (-8) = -5
   -2 … 16–2 =14 … 1110 -8 … 1000

1

```
         0011
    +    0100
přenosy: 0 0 0 0
(+)      0111 .... 7₁₀
```

2

```
         0101
    +    0100
přenosy: 0 1 0 0
(-)      1001 .... -7₁₀   ??
```

3a

```
         0011
    +    1110
přenosy: 1 1 1 0
(+)      0001 .... 1₁₀
```

3b

```
         0011
    +    1000
přenosy: 0 0 0 0
(-)      1011 ... -5₁₀
```

Snadný převod na opačné číslo: Zprava opisuj nuly až do první jedničky, tu opiš a další bity invertuj.

Detekce správného výsledku

Mějme M = 10000 a vyzkoušejme všechny možnosti součtů:

lze zobrazit: celá čísla od – 8 (M-8 = 8 … 1000) do 7 (0111)

$M = 1\,0000_2$, $\varepsilon = 1$

1. sčítáme malá kladná: 3+4=7

2. sčítáme větší kladná: 5 + 4 = 9
   (správný výsledek je mimo mřížku)

1	
	0011
+	0100
přenosy: 0 0 0 0	
(+)	0111 .... 7_{10}

2	
	0101
+	0100
přenosy: 0 1 0 0	
(-)	1001 .... -7_{10} ??

3. kladné + záporné:
   a) 3 + (-2) = 1, b) 3 + (-8) = -5
   -2 … 16–2 =14 … 1110 -8 … 1000

3a	
	0011
+	1110
přenosy: 1 1 1 0	
(+)	0001 .... 1_{10}

3b	
	0011
+	1000
přenosy: 0 0 0 0	
(-)	1011 ... -5_{10}

4. sčítáme „malá“ záporná:
   (-3) + (-1) = (-4)
   -3 … 16–3 =13 … 1101 -1 … 1111

4	
	1101
+	1111
přenosy: 1 1 1 1	
(-)	1100 .... -4_{10}

Snadný převod na opačné číslo: Zprava opisuj nuly až do první jedničky, tu opiš a další bity invertuj.

Detekce správného výsledku

Mějme M = 10000 a vyzkoušejme všechny možnosti součtů:

lze zobrazit: celá čísla od – 8 (M-8 = 8 … 1000) do 7 (0111)

$M = 1\ 0000_2$, $\varepsilon = 1$

1. sčítáme malá kladná: 3+4=7

1
0011
+ 0100
přenosy: 0 0 0 0
(+) 0111 …. 7_{10}

2. sčítáme větší kladná: 5 + 4 = 9
   (správný výsledek je mimo mřížku)

2
0101
+ 0100
přenosy: 0 1 0 0
(-) 1001 …. -7_{10} ??

3. kladné + záporné:
   a) 3 + (-2) = 1, b) 3 + (-8) = -5
   -2 … 16–2 =14 … 1110 -8 … 1000

3a
0011
+ 1110
přenosy: 1 1 1 0
(+) 0001 …. 1_{10}

3b
0011
+ 1000
přenosy: 0 0 0 0
(-) 1011 … -5_{10}

4. sčítáme „malá“ záporná:
   (-3) + (-1) = (-4)
   -3 … 16–3 =13 … 1101 -1 … 1111

4
1101
+ 1111
přenosy: 1 1 1 1
(-) 1100 …. -4_{10}

5. sčítáme „větší“ záporná
   -3 … 16–3 =13 … 1101 -8 … 1000

5
1101
+ 1000
přenosy: 1 0 0 0
(+) 0101 …. 5_{10} ??

Snadný převod na opačné číslo: Zprava opisuj nuly až do první jedničky, tu opiš a další bity invertuj.

Sčítání a odčítání

		$D(A) + D(B)$	$D(A + B)$
1	$A \geq 0 \; B \geq 0$	$A + B$	$A + B$
2	$A \geq 0 \; B < 0$ $A < 0 \; B \geq 0$	$A + B + M$	$A + B$ $A + B + M$
3	$A < 0 \; B < 0$	$A + B + M + M$	$A + B + M$

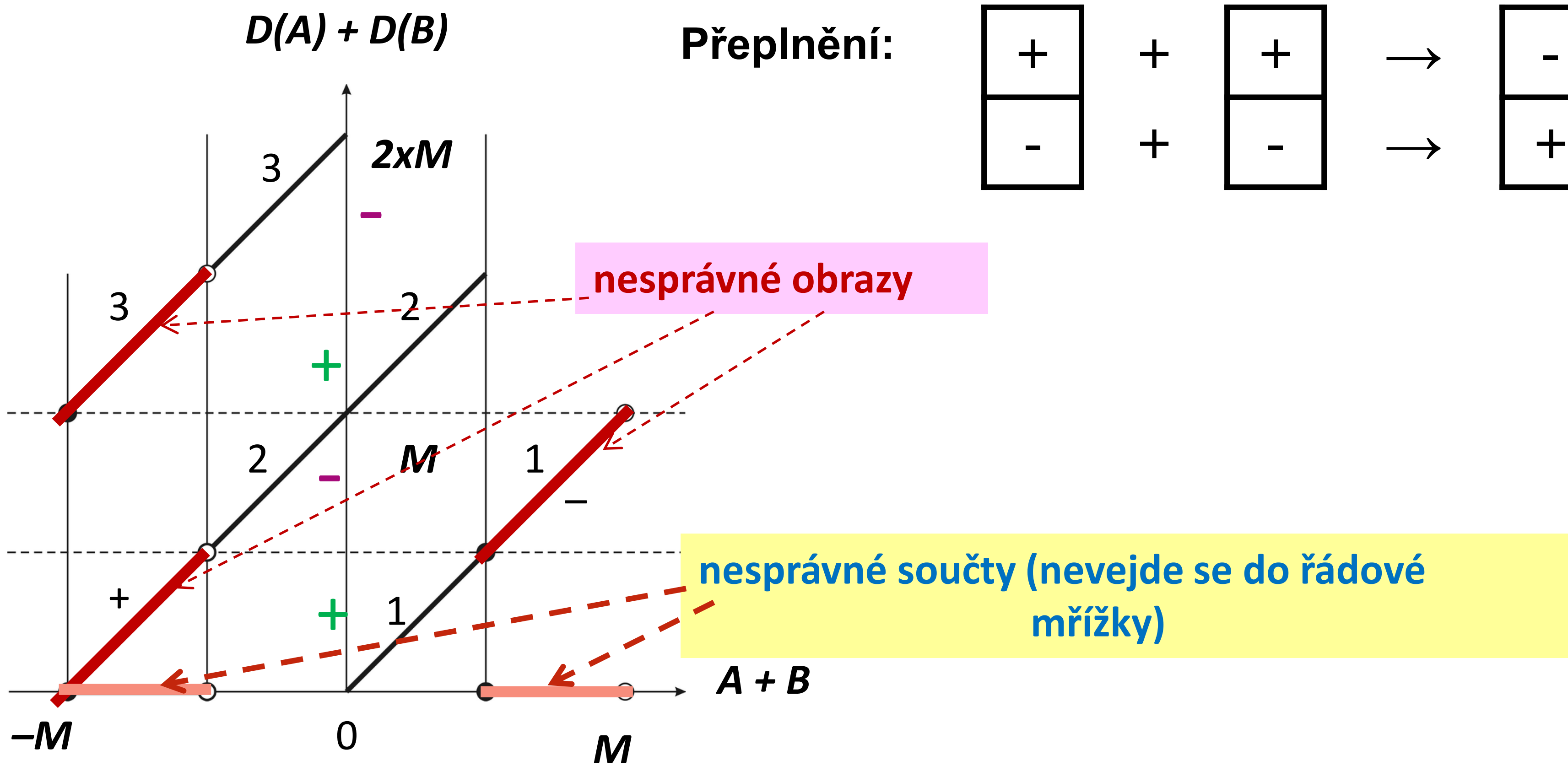
$$D(A + B) = \begin{cases} D(A) + D(B) \\ D(A) + D(B) - M \end{cases}$$

Sečtou se obrazy a ignoruje se přenos !!!

Odčítání = přičtení opačného čísla
→ detekce nesprávného výsledku stejná jako u sčítání přeplnění/přeteční/overflow

Přeplnění

Přeplnění (overflow) není přenos (carry) !!!!!

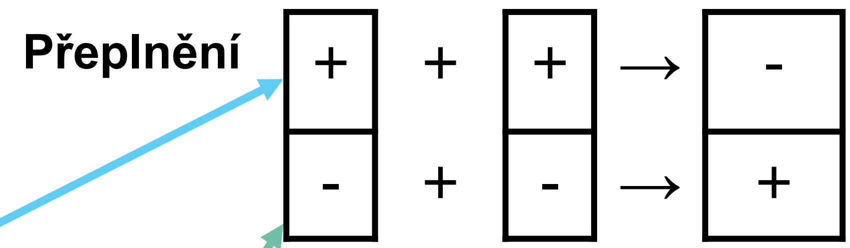


Přeplnění:

+	+	+	→	-
-	+	-	→	+

Přeplnění … podle bloku sčítačky pro nejvyšší řád

a	b	p	q	s
0	0	0	0	0
0	**0**	1	0	**1**
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	**1**	0	1	**0**
1	1	1	1	1



Přeplnění … podle bloku sčítačky pro nejvyšší řád

a	b	p	q	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Přeplnění

+	+	+	→	-
-	+	-	→	+

Odčítání

Příklad pro 3 bitová **nezáporná** čísla (opakování):

$B=101 \quad \overline{B}=010$

$B + \overline{B} = 111 = 1000 - 1 = M - 1$

$-B = \overline{B} + 1 - M$

$\boxed{A - B = A + \overline{B} + 1 - M}$

Správný výsledek = musím mít možnost odečíst modul, Musí vyjít přenos !!

Ale v **doplňkovém** kódu:

$A - B = A + (-B)$

$\mathcal{D}(B) + \mathcal{D}(-B) = B + (-B) + M = M$

$\mathcal{D}(-B) = M - \mathcal{D}(B)$

$\mathcal{D}(-B) = \overline{\mathcal{D}(B)} + 1$

$A - B = \mathcal{D}(A) + \overline{\mathcal{D}(B)} + 1$

detekce přeplnění je stejná jako u sčítání

Doplňkový kód pro desítkovou soustavu

- 10´s complement

Příklad: 3 místná desítková čísla:

$$M = 1000_{10}$$

znaménko je určeno první číslicí zleva:

0 – 4 .... + (kladná čísla)
5 – 9 .... – (záporná čísla)

X	D(X)	X	D(X)
0	000	–500	500
1	001	–499	501
…	…	…	…
499	499	–1	999

$D(X) + D(-X) = 1000 = 999 + 1$

$D(-X) = 999 - D(X) + 1$

označme: $\overline{a} = 9 - a$

$D(X) = 499 \rightarrow D(-X) = \overline{499} + 1$

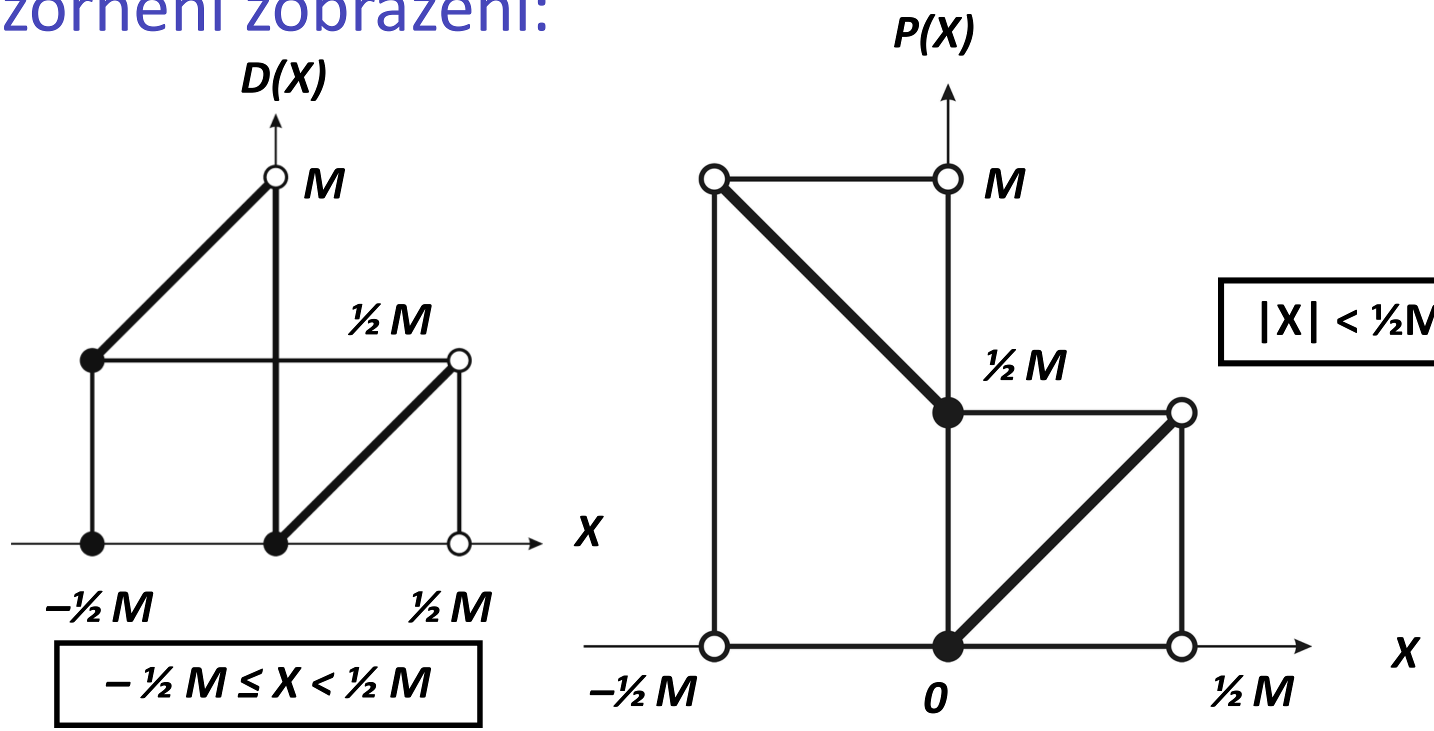
$D(-X) = 500 + 1 = 501$

Přímý kód

- Nejvyšší řád ř.m. představuje znaménko, zbytek ř.m. je absolutní hodnota
- Znaménko je reprezentováno číslicí: +…0, -…1

+/-	absolutní hodnota

- Znázornění zobrazení:



Příklady – přímý kód

M = 1000 ... tzn. 3bitová čísla

X	***P(X)***			
+0	0	0	0	←kladná nula
+1	0	0	1	
+2	0	1	0	
+3	0	1	1	
-0	1	0	0	←záporná nula
-1	1	0	1	
-2	1	1	0	
-3	1	1	1	

$-25_{10} \xrightarrow{\mathcal{P}}$ 1 0 2 5

$+101_{2} \xrightarrow{\mathcal{P}}$ 0 1 0 1

$+0{,}05_{10} \xrightarrow{\mathcal{P}}$ 0 0 5 0

$-0{,}11_{2} \xrightarrow{\mathcal{P}}$ 1 1 1 0

Sčítání a odčítání

- Pracujeme zvlášť se znaménkem a absolutní hodnotou
- Absolutní hodnota je nezáporné číslo

Příklad pro 3 bitová nezáporná čísla, viz opět:

$B=101 \quad \overline{B}=010$

$B + \overline{B} = 111 = 1000 - 1 = M - 1$

$-B = \overline{B} + 1 - M$

$$\boxed{A - B = A + \overline{B} + 1 - M}$$

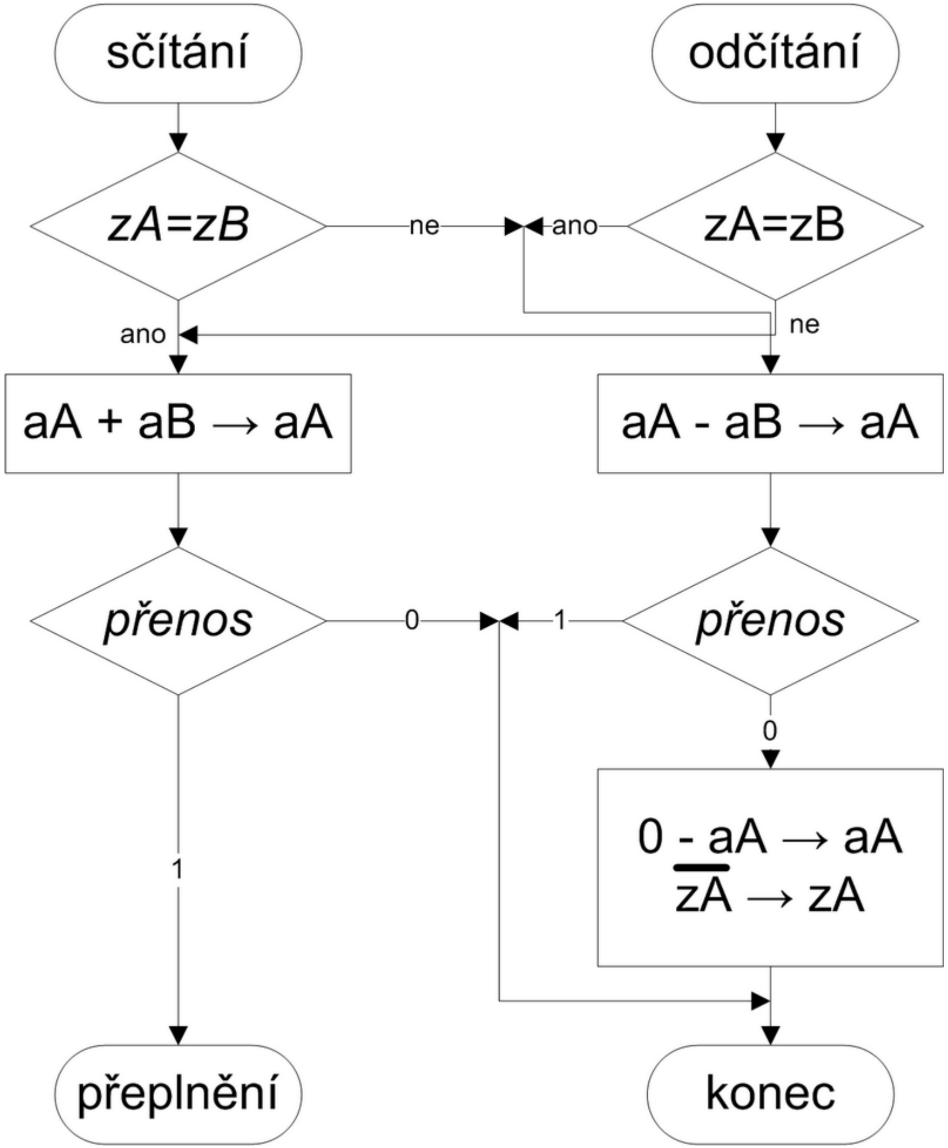
Algoritmus sčítání a odčítání

- A + B, A – B, výsledek ulož do A
- kde

  A ~ (zA, aA),

  B ~ (zB, aB)
- z – znaménko,

  a – absolutní hodnota

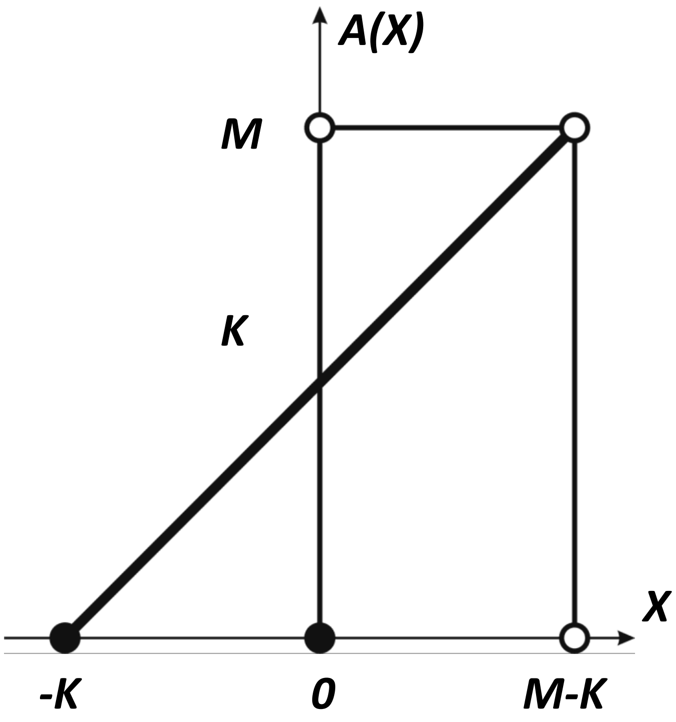


Realizace je složitější než pro doplňkový kód (jiná detekce nesprávného výsledku pro sčítání a odčítání + následná úprava → proto se používá doplňkový kód

Aditivní kód

- Též označovaný jako „kód s posunutou nulou“
- Formální definice: $\mathcal{A}(X) = X + K \quad \text{pro} \ -K \leq X < M - K$
- K: vhodná konstanta,
  často se volí:

  $K = ½ M$



Příklady – aditivní kód

$-25_{10} \xrightarrow[K=5000]{\mathcal{A}}$ | 4 | 9 | 7 | 5 |

$+101_2 \xrightarrow[K=1000_2]{\mathcal{A}}$ | 1 | 1 | 0 | 1 |

$+0{,}05_{10} \xrightarrow[K=1{,}000]{\mathcal{A}}$ | 1 | 0 | 5 | 0 |

$-0{,}11_2 \xrightarrow[K=1{,}000_2]{\mathcal{A}}$ | 0 | 0 | 1 | 0 |

Převod aditivní ↔ doplňkový
Přičtení aditivní konstanty (+ K)

$\mathcal{A}(5)$: 1101
+ K: 1000

$\mathcal{D}(5)$: 0101
+K: 1101

$\mathcal{A}(-0{,}11)$: 0,010
+ K: 1,000

$\mathcal{D}(-0{,}75)$: 1,010 ..... – 0,110
+ K: 1,000

0,010