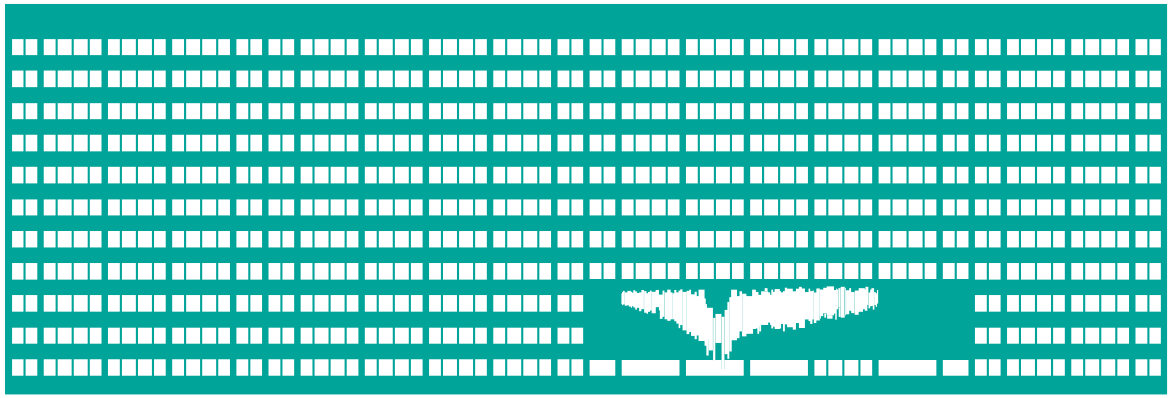
VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



www.vsb.cz

# Základy číslicových systémů

## 4. PŘEDNÁŠKA

# ČÍSELNÉ SOUSTAVY 2
# REÁLNÁ ČÍSLA

Převody do dekadické a z dekadické soustavy

Vztah mezi soustavami (2,8,16)

# Vyjádření přirozeného čísla polynomem – opak.

- Obecně lze číslo $(\alpha_{n-1}\alpha_{n-2}\cdots\alpha_1\alpha_0)_B$ zapsat polynomem

$$N_B = a_{n-1}\cdot B^{n-1} + \ \ldots + a_i \cdot B^i + \ \ldots + a_1 \cdot B^1 + a_0 \cdot B^0$$

kde

$B$ je základ číselné soustavy, angl. base, radix (přirozené číslo větší než 1),

$a$ je koeficient (přirozené číslo z intervalu 0 až $B$-1),

$i$ je pozice koeficientu, zapisuje se dekadicky, jako exponent prvku $B$ - řád koeficientu (celé číslo),

$B^i$ je váha, je dána pozicí a je vyjádřena mocninou základu,

$n$ je počet celočíselných míst (číslo $N_B$ je $n$-ciferné číslo)

$\alpha_i$ znaky číselné soustavy o základu $B$, reprezentující čísla $a_i$

# Vyjádření reálného čísla polynomem

- Reálné číslo $(\alpha_{n-1}\alpha_{n-2}\cdots\alpha_1\alpha_0,\alpha_{-1}\cdots\alpha_m)_B$ lze zapsat

$$N_B = a_{n-1}\cdot B^{n-1} + \ \ldots + a_i \cdot B^i + a_0 \cdot B^0 + a_{-1}\cdot B^{-1} + \ \ldots + a_{-m}\cdot B^{-m}$$

kde

- $B$ je základ číselné soustavy, angl. base, radix (přirozené číslo větší než 1),
- $a$ je koeficient (přirozené číslo z intervalu 0 až $B$-1),
- $i$ je pozice koeficientu, zapisuje se dekadicky, jako exponent prvku $B$ - řád koeficientu (celé číslo),
- $B^i$ je váha, je dána pozicí a je vyjádřena mocninou základu,
- $n$ je počet celočíselných míst
- $m$ je počet míst zlomkové části čísla
- $\alpha_i$ znaky číselné soustavy o základu $B$, reprezentující čísla $a_i$

# Převody do dekadické soustavy

- Převod celých i reálných kladných čísel – polynomem

- Převod celých kladných čísel (Hornerův algoritmus)

$$0 \cdot B + a_{n-1} = S_{n-1}$$
$$S_{n-1} \cdot B + a_{n-2} = S_{n-2}$$
$$\vdots$$
$$S_1 \cdot B + a_0 = S_0 = N_{10}$$

# Př. 4.1

## Převod $2 \rightarrow 10$, celé číslo

- $N = (01010011)_2$
- Polynom

  $1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^1 + 1 \cdot 2^0 = 64 + 16 + 2 + 1 = 83$
- Hornerův algoritmus

$$0 \cdot 2 + 1 = 1$$
$$1 \cdot 2 + 0 = 2$$
$$2 \cdot 2 + 1 = 5$$
$$5 \cdot 2 + 0 = 10$$
$$10 \cdot 2 + 0 = 20$$
$$20 \cdot 2 + 1 = 41$$
$$41 \cdot 2 + 1 = 83$$

- Číslo $(00010011)_2$ má dekadickou hodnotu $83$.

# Př. 4.2

## Převod 5 → 10, celé číslo

- $N = (1244)_5$
- Polynom

$$1 \cdot 5^3 + 2 \cdot 5^2 + 4 \cdot 5^1 + 4 \cdot 5^0 =$$
$$= 1 \cdot 125 + 2 \cdot 25 + 4 \cdot 5 + 4 \cdot 1 = 199$$

- Hornerův algoritmus

$$0 \cdot 5 + 1 = 1$$
$$1 \cdot 5 + 2 = 7$$
$$7 \cdot 5 + 4 = 39$$
$$39 \cdot 5 + 4 = 199$$

- Číslo $(1244)_5$ má dekadickou hodnotu $199$.

# Př. 4.3

## Převod 5 → 10, reálné číslo

- $N = (34{,}411)_5$
- Polynom
  $3 \cdot 5^1 + 4 \cdot 5^0 + 4 \cdot 5^{-1} + 1 \cdot 5^{-2} + 1 \cdot 5^{-3} =$
  $= 3 \cdot 5 + 4 \cdot 1 + 4 \cdot 0{,}2 + 1 \cdot 0{,}04 + 1 \cdot 0{,}008 = 19{,}848$

- Číslo $(34{,}411)_5$ má dekadickou hodnotu $19{,}848$.

# Přímé převody (2, 8, 16)

| $a_7$ | $a_6$ | $a_5$ | $a_4$ | $a_3$ | $a_2$ | $a_1$ | $a_0$ | $a_{-1}$ | $a_{-2}$ | $a_{-3}$ | $a_{-4}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0,5 | 0,25 | 0,125 | 0,0625 |
| | | | | | | | | | | | |
| | | | | | | | ← | → | | | |

- Přímé převody čísel mezi binární, oktální a hexadecimální soustavou (1. přednáška)
- $2 \leftrightarrow 8$ po třech bitech
- $2 \leftrightarrow 16$ po čtyřech bitech

# Vztah mezi soustavami (2,8,16)

$$N_2 = a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \ldots + a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

$$N_2 = \frac{(a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \ldots + a_3 \cdot 2^3)}{2^3} \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

$$N_2 = (a_{n-1} \cdot 2^{n-4} + a_{n-2} \cdot 2^{n-5} + \ldots + a_3) \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

$$N_2 = P^0 \cdot 2^3 + Z^0 \quad \Rightarrow \quad Z^0 = a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

$$N_2 = P^0 \cdot 2^4 + Z^0 \quad \Rightarrow \quad Z^0 = a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

# Př. 4.4

**Převod** $10 \rightarrow 2, 8, 16$
**reálné číslo**

| $a_7$ | $a_6$ | $a_5$ | $a_4$ | $a_3$ | $a_2$ | $a_1$ | $a_0$ | $a_{-1}$ | $a_{-2}$ | $a_{-3}$ | $a_{-4}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0,5 | 0,25 | 0,125 | 0,0625 |
| | | | | | | | | | | | |
| | | | | | | | ← | → | | | |

- Zadáno: $N = (140{,}5625)_{10}$
- $10 \leftrightarrow 2$ přímým převodem $(1000\ 1100{,}1001)_2$
- $2 \leftrightarrow 8$ po třech bitech $(214{,}44)_8$
- $2 \leftrightarrow 16$ po čtyřech bitech $(8C, 9)_{16}$

# Převody z dekadické soustavy

- Převod celých kladných čísel
  - Metoda postupného odečítání vah
  - Metoda postupného dělení základem
- Převod čísel kladných desetinných
  - Metoda postupného odečítání vah
  - Metoda postupného násobení základem

- V dalším
  $B_1 = 10$,
  $B_2$ nový základ (základ soustavy, do které převádíme)

# Př. 4.5

## Převod $10 \rightarrow 8$, celé číslo

- Metodou postupného odečítání vah převeďte do oktální soustavy číslo $N = (187)_{10}$
- Výsledek: $(187)_{10} = (273)_8$

| Váha | Rozdíl | Koeficient |
|---|---|---|
| $8^2 = 64$ | $187 - 64 = 123$ | $a_2 = 1$ |
| | $123 - 64 = 59$ | $\boldsymbol{a_2 = 2}$ |
| $8^1 = 8$ | $59 - 8 = 51$ | $a_1 = 1$ |
| | $51 - 8 = 43$ | $a_1 = 2$ |
| | $43 - 8 = 35$ | $a_1 = 3$ |
| | $35 - 8 = 27$ | $a_1 = 4$ |
| | $27 - 8 = 19$ | $a_1 = 5$ |
| | $19 - 8 = 11$ | $a_1 = 6$ |
| | $11 - 8 = 3$ | $\boldsymbol{a_1 = 7}$ |
| $8^0 = 1$ | $3 - 1 = 2$ | $a_0 = 1$ |
| | $2 - 1 = 1$ | $a_0 = 2$ |
| | $1 - 1 = 0$ | $\boldsymbol{a_0 = 3}$ |

# TABULE

# Metoda postupného odečítání vah 1/2

- Metoda spočívá v hledání koeficientů $a_{n-1}, a_{n-2}, \cdots, a_1, a_0$ postupným odčítáním zmenšujících se vah:

  Od čísla $N_{B1}$ odečteme váhu $B_2^{n-1}$ a dostaneme zbytek $^1N_{B1}$.

  - Když $^1N_{B1} \geq B_2^{n-1}$, potom $a_{n-1} = a_{n-1} + 1$ a znova odečítáme váhu.
  - Když $^1N_{B1} < B_2^{n-1}$, potom $a_{n-1} = a_{n-1} + 1$ a přejdeme k hledání koeficientu $a_{n-2}$.

- Aplikace - příklad 4.5

| Váha | Rozdíl | Koeficient |
|---|---|---|
| $8^2 = 64$ | $187 - 64 = 123$ | $a_2 = 1$ |
| | $123 - 64 = 59$ | $\boldsymbol{a_2 = 2}$ |
| $8^1 = 8$ | $59 - 8 = 51$ | $a_1 = 1$ |
| | $51 - 8 = 43$ | $a_1 = 2$ |
| | $43 - 8 = 35$ | $a_1 = 3$ |
| | $35 - 8 = 27$ | $a_1 = 4$ |
| | $27 - 8 = 19$ | $a_1 = 5$ |
| | $19 - 8 = 11$ | $a_1 = 6$ |
| | $11 - 8 = 3$ | $\boldsymbol{a_1 = 7}$ |
| $8^0 = 1$ | $3 - 1 = 2$ | $a_0 = 1$ |
| | $2 - 1 = 1$ | $a_0 = 2$ |
| | $1 - 1 = 0$ | $\boldsymbol{a_0 = 3}$ |

TEORIE PRO ZÁJEMCE

# Metoda postupného odečítání vah 2/2

- Od čísla ${}^{1}N_{B1}$ odečteme váhu $B_2^{n-2}$ a dostaneme zbytek ${}^{2}N_{B1}$ atd. viz předchozí bod.

- Algoritmus končí stanovením koeficientu $a_0$.

- Aplikace - příklad 4.5

| Váha | Rozdíl | Koeficient |
|---|---|---|
| $8^{\mathbf{2}} = 64$ | $187 - 64 = 123$ | $a_2 = 1$ |
| | $123 - 64 = 59$ | $\boldsymbol{a_2 = 2}$ |
| $8^{\mathbf{1}} = 8$ | $59 - 8 = 51$ | $a_1 = 1$ |
| | $51 - 8 = 43$ | $a_1 = 2$ |
| | $43 - 8 = 35$ | $a_1 = 3$ |
| | $35 - 8 = 27$ | $a_1 = 4$ |
| | $27 - 8 = 19$ | $a_1 = 5$ |
| | $19 - 8 = 11$ | $a_1 = 6$ |
| | $11 - 8 = 3$ | $\boldsymbol{a_1 = 7}$ |
| $8^{\mathbf{0}} = 1$ | $3 - 1 = 2$ | $a_0 = 1$ |
| | $2 - 1 = 1$ | $a_0 = 2$ |
| | $1 - 1 = 0$ | $\boldsymbol{a_0 = 3}$ |

# Metoda postupného dělení základem 1/3

- Metoda spočívá v hledání koeficientů $a_{n-1}, a_{n-2}, \cdots, a_1, a_0$ postupným dělením novým základem.

1. Hledání koeficientu $a_0$. Jestliže výraz (1) budeme dělit základem $B_2$, výsledkem bude podíl $P^0$ a zbytek $Z^0$, pro který bude platit $Z^0 < B_2$, přičemž $Z^0 = a_0$.

$$N_{B2} = a_{n-1} \cdot B_2^{n-1} + a_{n-2} \cdot B_2^{n-2} + \ \dots + a_1 \cdot B_2^1 + a_0 \cdot B_2^0 \qquad (1)$$

$$N_{B2} = \frac{(a_{n-1} \cdot B_2^{n-1} + a_{n-2} \cdot B_2^{n-2} + \ \dots + a_1 \cdot B_2^1)}{B_2^1} \cdot B_2^1 + a_0 \cdot B_2^0$$

$$N_{B2} = (a_{n-1} \cdot B_2^{n-2} + a_{n-2} \cdot 2^{n-3} + \ \dots + a_1 \cdot B_2^0) \cdot B_2^1 + a_0 \cdot B_2^0$$

$$N_{B2} = P^0 \cdot B_2^1 + Z^0 \quad \Rightarrow \quad Z^0 = a_0 \cdot B_2^0 = a_0$$

# Metoda postupného dělení základem 2/3

2. Hledání koeficientu $a_1$.
   Jestliže výraz $P^0$ budeme dělit základem $B_2$, výsledkem bude podíl $P^1$a zbytek $Z^1$, pro který bude platit $Z^1 < B_2$, přičemž $Z^1 = a_1$.

$$P^0 = a_{n-1} \cdot B_2^{n-2} + a_{n-2} \cdot B_2^{n-3} + \cdots + a_2 \cdot B_2^1 + a_1 \cdot B_2^0$$
$$P^0 = \frac{a_{n-1} \cdot B_n^{n-2} + a_{n-2} \cdot B_2^{n-3} + \cdots + a_2 \cdot B_2^1}{B_2^1} \cdot B_2^1 + a_1 \cdot B_2^0$$
$$P^0 = (a_{n-1} \cdot B_n^{n-3} + a_{n-2} \cdot B_2^{n-4} + \cdots + a_2 \cdot B_2^0) \cdot B_2^1 + a_1 \cdot B_2^0$$
$$P^0 = P^1 \cdot B_2^1 + Z^1 \Rightarrow Z^1 = a^1 \cdot B_2^0 = a^1$$

# Metoda postupného dělení základem 3/3

3. Dělením dalších podílů základem $B_2$ získáme postupně koeficienty hledaného čísla v pořadí od nejnižší k nejvyšší váze.
   Po opakovaném dělení nám nakonec musí vyjít nula, musíme tedy dělit i čísla, která jsou menší než dělitel.

| Dílčí podíl $P^n$ | Zbytek $Z^n$ | Koeficient $a_n$ |
|---|---|---|
| $N_B : B_2 = P^0$ | $Z^0$ | $a_0$ |
| $P^0 : B_2 = P^1$ | $Z^1$ | $a_1$ |
| $P^1 : B_2 = P^2$ | $Z^2$ | $a_2$ |

# Metoda postupného násobení základem 1

- Metoda spočívá v hledání koeficientů $a_{-1}, a_{-2}, \cdots, a_{-m+1}, a_{-m}$ postupným násobením základem.
- Hledání koeficientu $a_{-1}$.
  Jestliže výraz $(2)$ budeme násobit základem $B$, výsledek bude ve tvaru $a_{-1} + S^1$, kde $S^1$ je dílčí součin polynomu.

$$N_B = a_{-1} \cdot B^{-1} + a_{-2} \cdot B^{-2} + \cdots + a_{-m+1} \cdot B^{-m+1} + a_{-m} \cdot B^{-m} \qquad (2)$$

$$N_B = 0, a_{-1} a_{-2} \cdots a_{-m+1} a_{-m}$$

$$N_B \cdot B = a_{-1} + a_{-2} \cdot B^{-1} + \cdots + a_{-m+1} \cdot B^{-m+2} + a_{-m} \cdot B^{-m+1} =$$

$$= a_{-1} + S^1$$

# Metoda postupného násobení základem 2

2. Hledání koeficientu $a_{-2}$ .
   Jestliže výraz $S^1$ budeme násobit základem $B$, výsledek bude ve tvaru $a_{-2} + S^2$.
3. Opakujeme do zadané délky polynomu nebo stanovené přesnosti zobrazení v počtu bitů.

| Součin | Koeficient |
|---|---|
| $N_B \cdot B = a_{-1} + S^1$ | $a_{-1}$ |
| $S^1 \cdot B = a_{-2} + S^2$ | $a_{-2}$ |
| $S^2 \cdot B = a_{-3} + S^3$ | $a_{-3}$ |

# ZÁKLADNÍ POJMY ČÍSLICOVÝCH SYSTÉMŮ

Bit, byte, oktet, nibble, word, bit/s

Binární předpony (prefixy)

# BIT

- **Bit** - základní a současně nejmenší jednotka dat používaná v informatice a telekomunikacích.
- Může nabývat pouze jednu ze dvou hodnot, nejběžněji se pro znázornění používá 0 a 1.
- Značí se malým písmenem b, např. 16 b nebo 16 bit.
- V praxi se používá jako základní jednotka kapacity paměti, tzn. jednotka množství informace, která může být v jednom okamžiku v paměti uložena.

# BYTE, OKTET

- **Byte**
  - jednotka množství dat,
  - uspořádaná $n$-tice osmi bitů.
- Do jednoho bajtu je možno uložit $2^n$ různých hodnot.
- Byte jako číslo může nabývat hodnot:
  - dekadicky 0 až 255,
  - binárně 0000 0000 až 1111 1111,
  - hexadecimálně $00_h$ až $FF_h$
- **Oktet** - posloupnost osmi bitů, používáno v telekomunikacích.

# NIBBLE, WORD

- **Nibble**
  - uspořádaná $n$-tice čtyř bitů,
  - byte má dva nibbly.

- Nibble jako číslo může nabývat hodnot:
  - dekadicky 0 až 15,
  - binárně 0000 až 1111,
  - hexadecimálně $0_h$ až $F_h$

- **Word** – skupina bytů, velikost může být 16, 32 nebo 64 bitů podle architektury procesoru.

# BIT ZA SEKUNDU

- **Bit za sekundu** (b/s, bit/s, nebo angl. bps = bits per seconds)
  - jednotka rychlosti přenosu dat,

- Přenosová rychlost
  - udává, jaký objem informace se přenese za jednotku času (kolik bitů informace je přeneseno za jednu sekundu),
  - při rychlosti jednoho bitu za sekundu je každou sekundu přenesen právě jeden bit dat,
  - ke stažení souboru velikosti 1 MB za jednu sekundu se potřebuje připojení 8 Mb/s (1 bit = 1/8 byte).

# Binární předpony

- Předpony jednotek vyjadřujících násobek mocniny dvou

| Binární předpony (prefixy) dle IEC 60027-2 | | | | |
|---|---|---|---|---|
| $10^k$ | $2^n$ | Znak | Název | Hodnota |
| $10^3$ | $2^{10}$ | Ki | kibi | 1 024 |
| $10^6$ | $2^{20}$ | Mi | mebi | 1 048 576 |
| $10^9$ | $2^{30}$ | Gi | gibi | 1 073 741 824 |
| $10^{12}$ | $2^{40}$ | Ti | tebi | 1 099 511 627 776 |
| Poznámka: $10^k$ není rovno $2^n$, je to jen nejblíže odpovídající mocnina. | | | | |

- Kapacita polovodičových pamětí - binární předpony (IEC).
- Kapacita pevných disků, rychlost přenosu dat - obvykle dekadické předpony (SI).

# Př. 4.6 Převody - binární prefixy

- Převeďte na základní jednotky:

  $2\ \text{Kib} = 2^1 \cdot 2^{10}\ \text{b} = 2^{11}\ \text{b} = 2048\ \text{b}$

  $2\ \text{kb} = 2 \cdot 10^3\ \text{b} = 2\ 000\ \text{b}$

  $64\ \text{Mib} = 2^6 \cdot 2^{20}\ \text{b} = 2^{26}\ \text{b} = 67\ 108\ 900\ \text{b}$

  $64\ \text{Mb} = 64 \cdot 10^6\ \text{b} = 64\ 000\ 000\ \text{b}$

  $128\ \text{GiB} = 2^7 \cdot 2^{30}\ \text{B} = 2^{37}\ \text{B} = 137\ 439\ 000\ 000\ \text{B}$

  $128\ \text{GB} = 2^7 \cdot 10^9\ \text{B} = 128\ 000\ 000\ 000\ \text{B}$

- Převeďte na vhodné násobné jednotky:

  $2^{27}\ \text{bit} = 2^{27}\ \text{b} = 2^7 \cdot 2^{20}\ \text{b} = 2^7\ \text{Mib} = 128\ \text{Mib}$

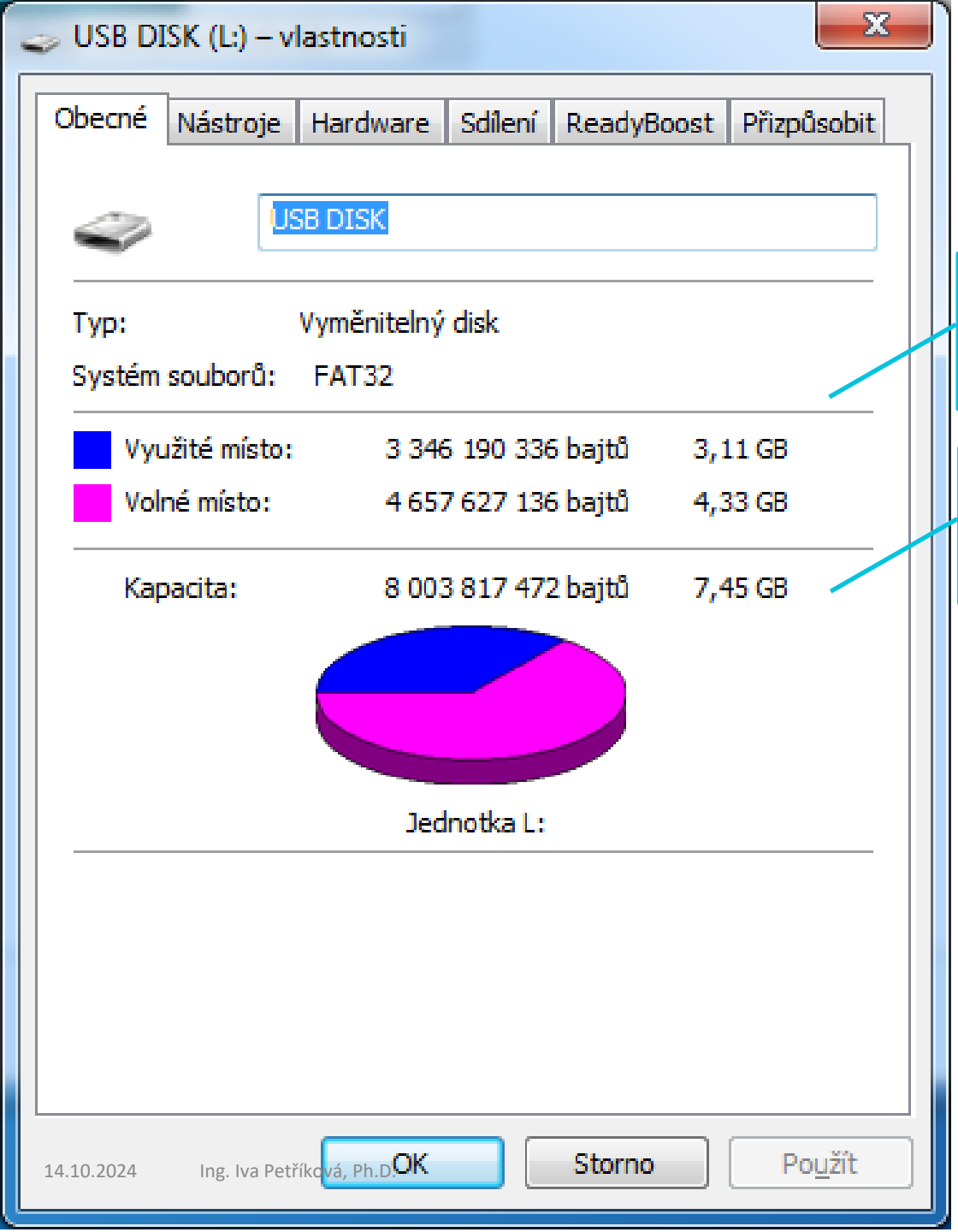
  $2^{16}\ \text{Byte} = 2^{16}\ \text{B} = 2^6 \cdot 2^{10}\ \text{B} = 2^6\ \text{KiB} = 64\ \text{KiB}$

  $2^{38}\ \text{Kib} = 2^{38} \cdot 2^{10}\ \text{b} = 2^8 \cdot 2^{40}\ \text{b} = 2^8\ \text{Tib} = 256\ \text{Tib}$

- Převeďte na bity:

  $1\ \text{KiB} = 8\ \text{Kib} = 8 \cdot 2^{10}\ \text{b} = 2^3 \cdot 2^{10}\ \text{b} = 2^{13}\ \text{b}$

  $1\ \text{MiB} = 8\ \text{Mib} = 8 \cdot 2^{20}\ \text{b} = 2^3 \cdot 2^{20}\ \text{b} = 2^{23}\ \text{b}$




píše se GB,
myslí se tím GiB

$7{,}45 \cdot 2^{30}\ \mathrm{B} = 8 \cdot 10^{9}\ \mathrm{B}$

7,45 GiB = 8 GB

# ENDIANITA

MSB, LSB a bitová pozice

Endianita

Little-endian

Big-endian

# MSB, LSB a bitová pozice

- MSB - nejvýznamnější bit (Most Significant Bit)
  - Termín se používá pro bit s nejvyšší hodnotou v binárním vyjádření čísla; v obvyklém dvojkovém zápisu jde o bit nejvíce vlevo.
- LSB – nejméně významný bit (Least Significant Bit)
  - Bit s váhou $2^0$.
- Bitová pozice
  - označuje bity in byte, word, vector…
  - v programovacích jazycích je definován rozsah a směr
  - index může být v rozsahu 0 – n, 1 – n, n – 0
- MSB a LSB se někdy spojuje s bytem.

# Př. 4.7

## MSB, LSB a bitová pozice

- Číslo 139
  - Obvykle:
  - Změna pozice
  - Změna váhy
  - Změna váhy i pozice

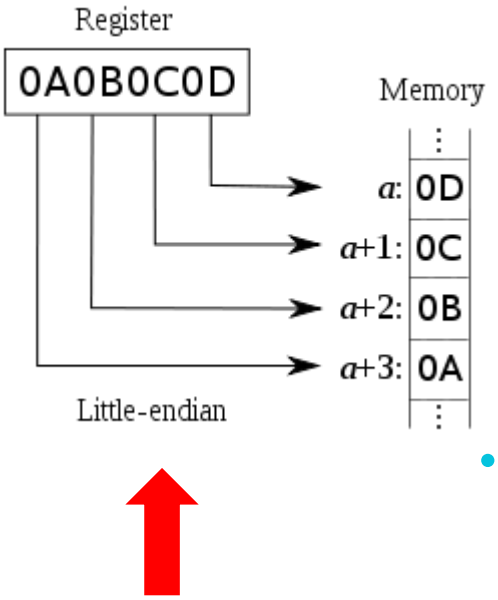
| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| MSB | | | | | | | LSB |
| 0 | | | | | | | 7 |
| **1** | **0** | **0** | **0** | **1** | **0** | **1** | **1** |
| MSB | | | | | | | LSB |
| 7 | | | | | | | 0 |
| **1** | **1** | **0** | **1** | **0** | **0** | **0** | **1** |
| LSB | | | | | | | MSB |
| 0 | | | | | | | 7 |
| **1** | **1** | **0** | **1** | **0** | **0** | **0** | **1** |
| LSB | | | | | | | MSB |

# Endianita – pořadí bytů

- Informace o $n$ bitech je rozdělena do samostatně adresovatelných atomických elementů. Atomický element většinou bývá byte, může být i slovo.
- **Endianita** definuje pořadí atomických elementů uvnitř rozsáhlé datové struktury (uložení v paměti, v souboru, datový proud při přenosu).
- Určuje, který atomický element je na nižší adrese nebo který bude přenášen jako první.
- S endianitou se setkáváme u ukládaní FP čísel, u formátů UTF-16 a UTF-32, které definují Unicodovou pozici, 64 bit integer, pořadí bytů při přenosu.

# Little-endian

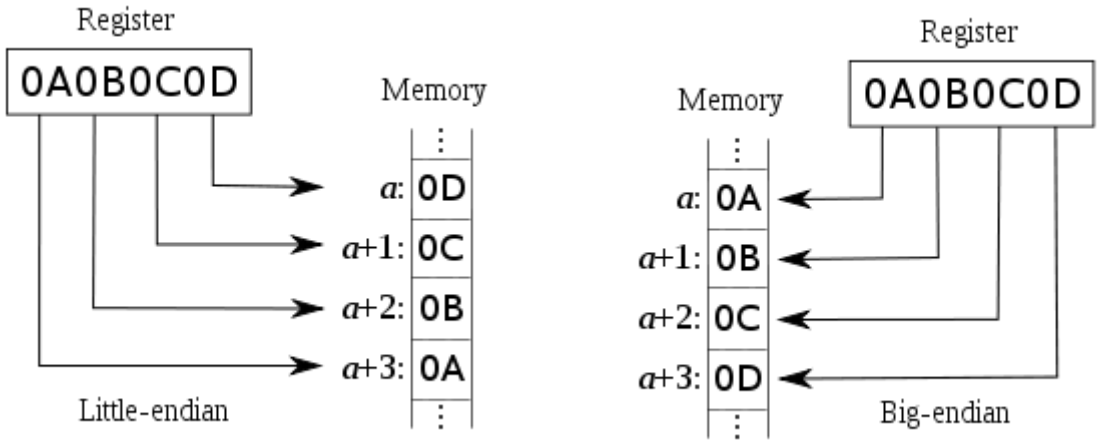
- Na paměťové místo s **nejnižší** adresou se uloží **nejméně** významný bajt.
- Za něj se ukládají ostatní bajty až po nejvíce významný bajt.




- Architektury uplatňující princip little-endian - MOS Technology 6502, Intel x86 a DEC VAX.

# Big-endian

- Na paměťové místo s nejnižší adresou se uloží nejvíce významný bajt.
- Za něj se ukládají ostatní bajty až po nejméně významný bajt.




- Architektury uplatňující princip big-endian Motorola 68000, SPARC a System/370.

# Př. 4.8 Little-endian

- Zapište do paměti na adresu $a_n$ 32-bitové číslo 0x235F2D3C podle principu little-endian. Do nepoužitých adres pište nuly.
  - Při ukládání do paměti po bytech:

| | $a_n$ | $a_{n+1}$ | $a_{n+2}$ | $a_{n+3}$ | |
|---|---|---|---|---|---|
| | 3C | 2D | 5F | 23 | |
| | LSB | | | MSB | |

  - Při ukládání do paměti po slovech:

| | $a_n$ | $a_{n+1}$ | $a_{n+2}$ | $a_{n+3}$ | |
|---|---|---|---|---|---|
| | 2D3C | 235F | 0000 | 0000 | |
| | LSB | MSB | | | |

# Př. 4.9 Big-endian

- Zapište do paměti na adresu $a_n$ 32-bitové číslo 0x235F2D3C podle principu big-endian. Do nepoužitých adres pište nuly.
  - při ukládání do paměti po bytech:

| | $a_n$ | $a_{n+1}$ | $a_{n+2}$ | $a_{n+3}$ | |
|---|---|---|---|---|---|
| | 23 | 5F | 2D | 3C | |
| | MSB | | | LSB | |

  - při ukládání do paměti po slovech:

| | $a_n$ | $a_{n+1}$ | $a_{n+2}$ | $a_{n+3}$ | |
|---|---|---|---|---|---|
| | 235F | 2D3C | 0000 | 0000 | |
| | MSB | LSB | | | |

# BITOVÉ OPERACE 1

Bitové operace - NOT, AND, OR, XOR, posuny

# Bitové operace

- Provádějí se pouze s **celočíselnými** hodnotami, které se rozloží do jednotlivých bitů.
- Bitová operace (anglicky bitwise operation) je operace, která aplikuje určitou logickou operaci na celé vektory bitů.

# Bitové operátory

- **Unární:** NOT - bitová negace (bitový doplněk)
  - provádí negaci každého bitu v bitovém řetězci jediného operandu.
- **Binární**: AND - bitový součin, OR - bitový součet, XOR - bitová nonekvivalence
  - provádí **binární** operaci s každým párem odpovídajících si bitů. Výsledek je umístěn do pozice stejného binárního řádu výsledku.
- **Posuny**

# Logické a bitové operátory v jazyku C

## Logické operátory

| | |
|---|---|
| ! | NOT<br>negace |
| && | AND<br>logický součin |
| \|\| | OR<br>logický součet |

K bitovým operátorům existují jejich verze kombinující v sobě přiřazení:
&=  |=  ^=  <<=  >>=

## Bitové operátory

| | |
|---|---|
| ~ | bitový NOT<br>jednotkový doplněk |
| & | bitový AND<br>logický součin bit po bitu |
| \| | bitový OR<br>logický součet bit po bitu |
| ^ | bitový XOR<br>nonekvivalence |
| << | bitový posun vlevo |
| >> | bitový posun vpravo |

# Bitová negace

- Provádí negaci každého bitu v bitovém řetězci. Po provedení operace je hodnota každého bitu opačná.
- Využití:
  - jednotkový doplněk,
  - dvojkový doplněk NOT (A) + 1.
- Protože jsou negovány všechny bity čísla, včetně počátečních nul, znamená to, že pro operandy se stejnou hodnotou, ale různým typem (char, short, int, long int), bude výsledná hodnota různá.
- V jazyku C je pro bitový NOT používán znak ~ (logický NOT používá znak **!**).

# Př. 4.10. Bitová negace

- Je zadané dekadické číslo A = 22. Určete bitovou negaci čísla A na osmi a na šestnácti bitech (binárně, hexadecimálně a dekadicky).

**TABULE**

# Bitový součet

- Provádí s každým párem bitů operaci OR
  - hodnota bitu na dané pozici ve výsledku je 1, pokud alespoň jeden bit na téže pozici má hodnotu 1, nebo je výsledek 0, pokud oba dva bity mají hodnotu 0.
- Využití:
  - Nastavení bitů, nastavování příznaků (bity prvního operandu jsou nastaveny na těch pozicích, kde je v druhém operandu, masce, hodnota 1).
  - Přičítání vah
- V jazyku C je pro bitový součet OR používán znak | (logický součet OR používá ||).

# Př. 4.11. Bitový součet

- Jsou zadaná dvě čísla A = 6 a B = 4. Určete výsledek bitového součtu na čtyřech a na osmi bitech (binárně, hexadecimálně a dekadicky).

**TABULE**

# Př. 4.12 Nastavení bitů v registru

- Je dán registr podle obrázku:

| 7(15) | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| MSB | | | | | | | LSB |

TABULE

- V registru jsou nastaveny bity na pozicích odpovídajících lichým číslům. Registr lze číst a zapisovat jako integer číslo. Stanovte hodnotu masky pro nastavení nejméně významného bitu pro osmibitový a šestnáctibitový registr a stanovte hodnotu registru po nastavení bitu.

# Bitový součin

- Provádí s každým párem bitů operaci AND.
  - hodnota bitu na dané pozici ve výsledku je 1,
    pokud jsou hodnoty bitů na téže pozici v obou operandech 1,
    jinak je hodnota bitu na dané pozici ve výsledku 0.
- Využití:
  - nulování bitů (bity prvního operandu jsou nulovány na těch pozicích, kde je v druhém operandu, masce, hodnota 0),
  - testování bitů.
- V jazyku C je pro bitový součin AND používán znak ampersand & (logický součin AND používá &&).

# Př. 4.13 Bitový součin

- Jsou zadaná dvě čísla A = 12 a B = 6. Určete výsledek operace bitového součinu na čtyřech a na osmi bitech (binárně, hexadecimálně a dekadicky).

**TABULE**

# Př. 4.14 Nulování bitů v registru

- Je dán registr podle obrázku:

| 0 | | | | | | | 7(15) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MSB | | | | | | | LSB |

TABULE

- V registru jsou nastaveny všechny bity. Registr lze číst a zapisovat jako integer číslo.
  Stanovte hodnotu masky pro nulování tří bitů na nejnižší pozici pro osmibitový a šestnáctibitový registr a stanovte hodnotu registru po nulování.

# XOR – bitová neekvivalence

- Provádí s každým párem bitů operaci XOR
  - hodnota bitu na dané pozici je ve výsledku 1, pokud jsou hodnoty na téže pozici různé, jinak je výsledek 0.
- Využití:
  - Invertování bitů (bity prvního operandu jsou invertovány na těch pozicích, kde je v druhém operandu, masce, hodnota 1).
  - Jednotkový doplněk (druhý operand má ve všech bitech jedničky; operace je ekvivalentní k operaci bitové negace).
- V jazyku C je pro bitovou neekvivalenci používán znak „stříška“ ^ .

# Př. 4.15 Invertování bitů v registru

- Je dán registr podle obrázku:

| 7(15) | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MSB | | | | | | | LSB |

TABULE

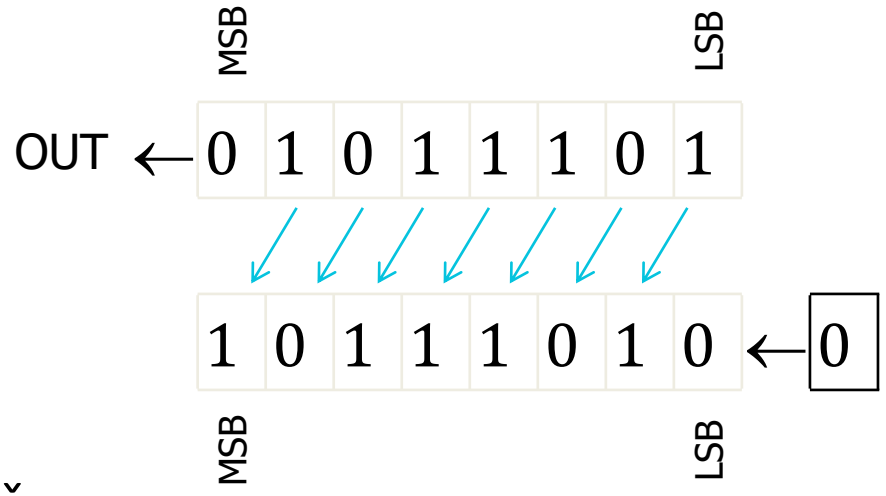
- V registru jsou nastaveny všechny bity. Registr lze číst a zapisovat jako integer číslo.
  Stanovte hodnotu masky pro invertování bitů na pozici 0 a 2 pro osmibitový a šestnáctibitový registr a stanovte hodnotu registru po invertování.

# Bitové posuny

- Při bitovém posunu vlevo (vpravo) se jednotlivé bity posouvají vlevo (vpravo), tedy do pozice s binárně vyšším (nižším) řádem.
- Uvedené pravidlo neplatí pro posun vpravo hodnoty celočíselného typu se znaménkem. V takovém případě se nejvyšší znaménkový bit zachovává.

# Bitový posun doleva

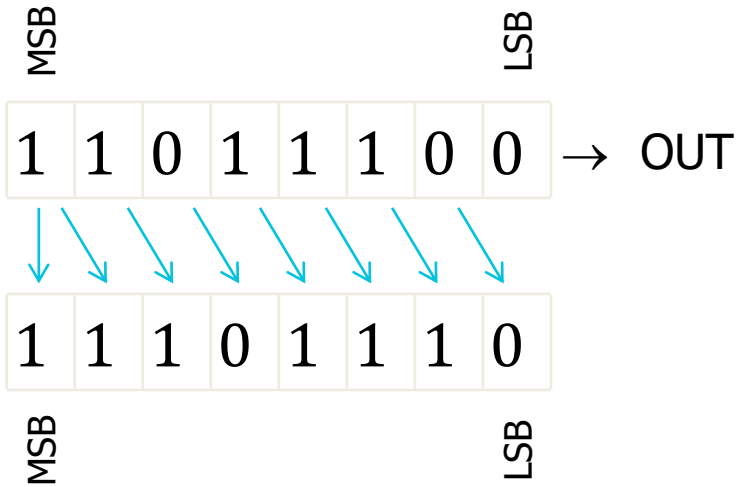
- Při posunu se bity nejvíce nalevo ztrácejí a zprava jsou uvolněná místa doplněna nulami.
- Bitový posun čísla doleva o jeden bit je stejný, jako bychom číslo vynásobili dvěma.




- Bitový posun čísla doleva o $n$ bitů je stejný, jako bychom číslo vynásobili $2^n$.
- Jazyk C: $x << n$ (posun doleva všech bitů levého operandu $x$ o počet míst $n$, který udává hodnota pravého operandu).

# Bitový posun doprava

- Bity nejvíce napravo se ztrácejí a zleva jsou doplňovány nulou u neznaménkových typů, nebo znaménkovým bitem u typů znaménkových.
- Bitový posun čísla doprava o jeden bit je stejný, jako bychom číslo dělili dvěma.




- Bitový posun čísla doprava o $n$ bitů je stejný jako bychom číslo dělili $2^n$.
- Jazyk C: $x >> n$ (posun doprava všech bitů levého operandu $x$ o počet míst $n$, který udává hodnota pravého operandu).

# Bitové rotace

- Rotace jsou takové posuvy, u kterých se vysunuté bity opět do místa postupně opět vsunou z druhé strany, tj. bity jsou v paměťovém místě rotovány.

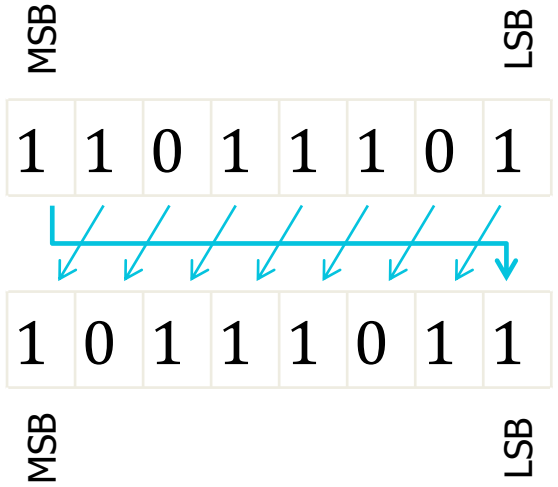
  (Při instrukcích posuvů jsou bity, které jsou z místa vysunuty ven, ztraceny.)

- V jazyku C neexistuje operátor rotace.

# Bitová rotace (NO carry)

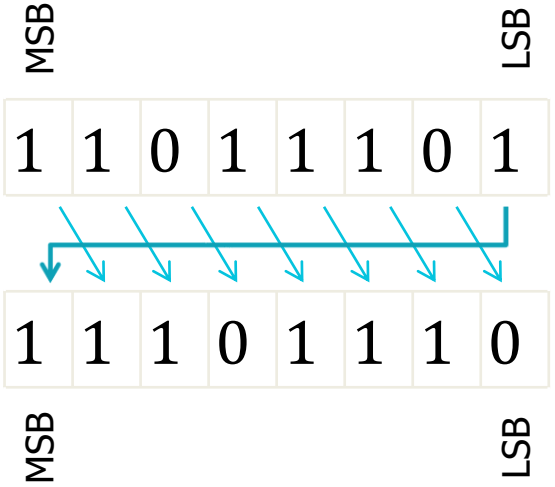
## Doleva

- Bity se přesunou o jednu pozici výše, do nejnižšího bitu se uloží vysunutý bit.




## Doprava

- Bity se přesunou o jednu pozici níže, do nejvyššího bitu se uloží vysunutý bit.



# Použité zdroje


- ZDRÁLEK, Jaroslav, CHMELÍKOVÁ, Zdeňka. 2017. Číslicové systémy 1.
- DIVIŠ, Zdeněk, CHMELÍKOVÁ, Zdeňka a ZDRÁLEK, Jaroslav. 2008. Logické obvody. Ostrava : VŠB - Technická univerzita Ostrava, 2008. str. 152. Skripta. ISBN 978-80-248-1724-8.
- Přispěvatelé Wikipedie, Bit [online], Wikipedie: Otevřená encyklopedie, c2022, Datum poslední revize 15. 09. 2022, 10:10 UTC, [citováno 9. 10. 2022] https://cs.wikipedia.org/w/index.php?title=Bit&oldid=21680017
- Wikipedia contributors. (2018, February 26). Endianness. In Wikipedia, The Free Encyclopedia. Retrieved 07:48, March 7, 2018, from https://en.wikipedia.org/w/index.php?title=Endianness&oldid=827746821

# Použité zdroje


- Přispěvatelé Wikipedie, *Přenosová rychlost* [online], Wikipedie: Otevřená encyklopedie, c2022, Datum poslední revize 23. 06. 2022, 19:24 UTC, [citováno 9. 10. 2022] https://cs.wikipedia.org/w/index.php?title=P%C5%99enosov%C3%A1_rychlost&oldid=21411228
- Přispěvatelé Wikipedie, *Bitová operace* [online], Wikipedie: Otevřená encyklopedie, c2021, Datum poslední revize 5. 08. 2021, 11:09 UTC, [citováno 9. 10. 2022] <https://cs.wikipedia.org/w/index.php?title=Bitov%C3%A1_operace&oldid=20292702>

# Děkuji za pozornost

**Ing. Iva Petříková, Ph.D.**

+420 597 325 840
**Iva.petrikova@vsb.cz**

**www.vsb.cz**
**https://comtech.vsb.cz/**