

## 2. INFORMACE VE VÝPOČETNÍ TECHNICE

**Co je informace** – to co můžeme zachytit svými smysly, nazýváme údaje informace je potom část údajů, která je pro příjemce pochopitelná. Informace tedy slouží ke zpracování, skladování nebo přenášení dat.

K tomu, aby informace mohly být zpracovány počítačem, musí být převedeny na data. Toto se provádí kódováním (převod informace do dat).

### **Číselné soustavy :**

Dvojková soustava (binární) -je polyadická soustava o základu  $N=2$

Používají se pouze 2 číslice 0 a 1 přičemž 0/1 znamená u tranzistoru zapnuto/vypnuto

Jakékoliv číslo můžeme vyjádřit tak, že ho rozložíme na součet postupných mocnin se základem 2 tzv. na čísla :  $2^0=1; 2^1=2; 2^2=4$  atd. Pokud se při rozkladu příslušná mocnina v daném řádu vyskytuje , zapisujeme ji jako 1 , chybí-li píšeme 0.

Šestnáctková soustava (hexadecimální) – polyadická číselná soustava o základu  $N=16$  .

Používá šestnáct číslic :0-9-A-F .

V některých případech se používá hex místo binárky kvůli tomu, že by byla binárka příliš dlouhá a tedy nepraktická .

Desítková soustava ( BCD) - polyadická číselná soustava o základu  $N=10$  nejpoužívanější vzhledem k počtu prstů na ruce.

Jakékoliv přirozené číslo je možno tedy vyjádřit jako polynomiální mocniny základny koeficientu:

$$C \cdot N^2 + C \cdot N^1 + C \cdot N^0 = N$$

Př:

Desítková:  $38956 \Rightarrow 3 \cdot 10^4 + 8 \cdot 10^3 + 9 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0$   
C4 C3 C2 C1 C0

### **Převod :**

Převedení z desítkové do jiné soustavy spočívá v 2 krocích:

- 1) Vztít nejvyšší mocninu základny (N) menší než dané číslo
- 2) Odečtu mocninu daného čísla a s rozdílem opakuji tento postup od 1)

Na příkladu:

$$27_{(10)} \Rightarrow {}^{(2)}11011 \Rightarrow \begin{array}{ccccc} 27-16=\underline{11} & ; & 11-8=\underline{3} & ; & 3-4=\underline{X} & ; & 3-2=\underline{1} & ; & 1-1=\underline{0} \\ & & 1 & & 1 & & 0 & & 1 & & 1 \end{array}$$

Převody pro ostatní soustavy:

$$11110_{(2)} \Rightarrow {}^{(10)}30 \Rightarrow \begin{array}{cccc} 2^1 & + & 2^2 & + & 2^3 & + & 2^4 = \underline{30} \\ \text{C3} & & \text{C2} & & \text{C1} & & \text{C0} \end{array}$$

$$0101\ 0001\ 1110_{(2)} \Rightarrow {}^{(16)}51\text{E} \Rightarrow 0101 \rightarrow 5 ; 0001 \rightarrow 1 ; 1110 \rightarrow \text{E}$$

$$61\text{A}_{(16)} \Rightarrow {}^{(2)}011000011010 \Rightarrow 6 \rightarrow 0110 ; 1 \rightarrow 0001 ; \text{A} \rightarrow 1010$$

### **Sčítání/odčítání v binárce :** $0111+1100 = \underline{10011}$ ; $1011110-0111000 = \underline{100110}$

Postup je prakticky stejný jako u desítkové soustavy u sčítání pokud máme dvě 1 přidáme jedničku do dalšího řádu u odečítání když máme v prvním řádku 0 a v druhém 1 odebíráme jedničku od dalšího řádu.

**Endianita** - je způsob uložení čísel v operační paměti počítače, který definuje, v jakém pořadí se uloží jednotlivé bajty číselného datového typu. Jde tedy o to, v jakém pořadí jsou v operační paměti uloženy jednotlivé řády čísel, které zabírají více než jeden bajt.

- **Little-endian** - paměťové místo s nejnižší adresou uloží nejméně významný bajt (LSB) a za něj se ukládají ostatní bajty až po nejvíce významný bajt (MSB)
- **Big-endian** - paměťové místo s nejnižší adresou uloží nejvíce významný bajt (MSB) a za něj se ukládají ostatní bajty až po nejméně významný bajt (LSB) na konci
- **Middle-endian** - užívají složitější způsob pro určení pořadí jednotlivých bajtů, který je dán kombinací obou výše zmíněných způsobů.

### **Metody zabezpečení programů a dat proti chybám**

Při přenosu dat může dojít k chybám, způsobených závadou nebo vnějšími vlivy. Pokud dojde k přerušení linky a počítač přestane dostávat data chyba se vlastně odhalí sama. Horší je, když počítač omylem přijme log.0 místo log.1.

Počítač by nepoznal, že nastala chyba a dále by pracoval se špatnými daty. Proto se používá různých zabezpečovacích prostředků. Jednou z metod zabezpečení je použití zabezpečovacích kódů. To znamená, že k

datům samotným se přidá další informace, která umožní příjemci zjistit, jestli jsou došlá data v pořádku.

#### **Paritní bit:**

-Patří k nejjednodušším způsobům detekce chyb. Dokáže chyby pouze nalézt. Paritní bit je Bit přidaný k řetězci přenášených dat binárního kódu, který má zajistit, že celkový počet bitů v řetězci je sudý či lichý. Existují dvě varianty: sudý paritní bit a lichý paritní bit. Lichá parita znamená lichý počet jedničkových bitů ve slově (i s paritním bitem), sudá parita sudý počet jedničkových bitů ve slově. Nevýhodou je, nastane-li chyba na více bitech najednou, v tom případě parita neodhalí chybu. Nebo pokud je chybně přijat paritní bit, mohou být správná data vyhodnocena jako chybná.

#### **CRC:**

-zde se pomocí tzv. generujícího polynomu z přenášených dat vypočte určitým způsobem zabezpečující kód. Ten se pak připojí za přenášená data. Výhodou je, že výpočet kódu se může provádět průběžně a data se mohou mezitím vysílat. Zabezpečující kód se připojí až za vysílaná data a nestává se jejich součástí. Metoda výpočtu CRC je založena na matematické teorii polynomů. Z toho plyne i výše uvedený pojem "generující polynom".

CRC se používá velmi často, protože jde o metodu poměrně jednoduchou, ale značně efektivní.