

## Informace ve výpočetní technice

### Převody

DEC	BIN	HEX
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

$$73_{10} = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1001001_2$$

$$1100010_{10} = \boxed{1} \cdot 2^6 + \boxed{1} \cdot 2^5 + \boxed{0} \cdot 2^4 + \boxed{0} \cdot 2^3 + \boxed{0} \cdot 2^2 + \boxed{1} \cdot 2^1 + \boxed{0} \cdot 2^0$$

### Dvojková soustava

- Je číselná soustava, která používá jenom dvě číslice a to **0** a **1**.
- Dvojková soustava je číselná soustava se základem **2**, každá číslice odpovídá **n-té** mocnině čísla **2**, kde **n** je pozice určité číslice v zapsaném čísle.
- Takto zapsané číslo se nazývá binární číslo.

### Šestnáctková soustava

- Také jako *hexadecimální soustava*, tak se v ní zapisují pomocí číslic **0** až **9** a písmen **A-F**, přičemž písmena určují cifry s hodnotou **10–15**.
- V matematice se šestnáctková čísla označují dolním indexem **H**, **hex** nebo **16**
- Díky jednoduchému převodu ze šestnáctkové na dvojkovou soustavou a opačně, tak se hexadecimální zápis čísel často používá v oblasti IT (pro adresy v operační paměti)

### BCD

- Neboli *Binary Code Decima*, tak je binárně kódované decimální číslo.
- Ve výpočetní technice se často používá binární kódování desítkového čísla tak, že každá desítková číslice má přiřazen pevný počet bitů. Obvykle **4 Bity**, výjimečně **8 Bitů** a dříve dokonce i **6 Bitů**.
- Je to "oříznutý" kód hexadecimální soustavy v rozsahu **4 Bity**, kdy bitové kombinace, které neodpovídají číslicím **0-9**, tedy hodnoty **10-15**, jsou buď zakázány anebo mohou být použity pro určení znaménka, chyby, přetečení, nebo polohy desetinné čárky.

### Podpora ze strany CPU

- Více jádrový procesor je mikroprocesor, který na jednom čipu integruje více CPU.
- Obvykle jde o jádra, která jsou vzájemně programově kompatibilní a mohou tak snáze spolupracovat.
- Jejich společná činnost se nazývá multitasking nebo multithreading.
- Pro využití více jádrových procesorů je obvykle třeba podpora ze strany operačního systému. V současné době to však znamená podporu většiny systémů, včetně aktuálních systémů Microsoft Windows, Apple macOS, Linux a další.

### Zobrazení záporných čísel:

#### — Kódy —

**přímý kód** : rozsah zobrazení  $\langle -2^{n-1}+1, -0 \rangle$

pro  $n=8$   $\langle -127, -0 \rangle, \langle +0, +127 \rangle$

**inverzní kód** : inverze bitů (jedničkový doplněk)

**doplňkový kód** - operace dvojkový doplněk = inverze bitů a přičtení jedničky

rozsah zobrazení  $\langle -2^{n-1}, 2^{n-1}-1 \rangle$

pro  $n=8$   $\langle -128, 127 \rangle$

### Struktura procesorového systému z hlediska dat

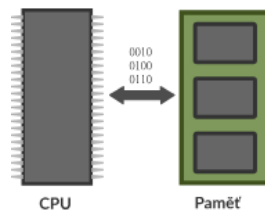
- Procesorový systém je klíčovou součástí počítače, která provádí všechny výpočetní operace a řídí běh programů. Je důležitá pro pochopení toho, jak jsou data ukládána, přenášena a zpracovávána v rámci počítačového systému.

#### — CPU —

- **Registry**: Procesor obsahuje registry, což jsou malé, velmi rychlé úložiště pro data, která jsou přímo dostupná procesoru.
- **Arithmetic Logic Unit: ALU** je část procesoru, která provádí výpočty a logické operace. Může sčítat, odčítat, násobit, dělit a ostatní.
- **Control Unit: Řadič procesoru** řídí tok instrukcí a dat mezi různými částmi procesoru a pamětí. Vysvětluje instrukce a řídí operace v ALU.

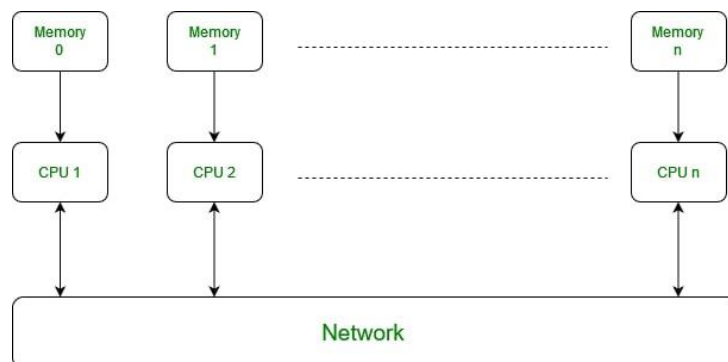
#### — Paměť —

- **Random Access Memory: RAM** je typ paměti, která umožňuje náhodný přístup k datům a instrukcím. To znamená, že procesor může číst a zapisovat data na libovolné místo v RAM.
- **Read Only Memory: ROM** je paměť, která obsahuje pevně zapsaná data a instrukce, která nelze měnit. Obsahuje základní instrukce pro inicializaci a spuštění počítače.



## Program jako data

- Programy jsou soubory instrukcí, které řídí chování počítače.
- Mohou být napsány v různých programovacích jazycích, jako je **Python, Java, C++**, a mnoho dalších.
- Pomocí programů můžeme vytvářet, upravovat a manipulovat s daty. Data jsou informace, které počítač zpracovává, ukládá a zobrazuje. Mohou to být texty, obrázky, zvuky, čísla, a mnoho dalšího.



## Data v paměti - endianita dat

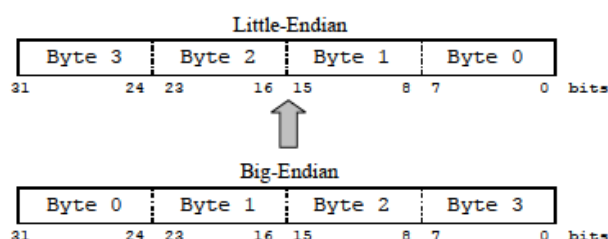
- *Endianita* je způsob, jakým jsou v počítači uložena více bajtová čísla v paměti.
- Existují dvě základní varianty: **big-endian** a **little-endian**.

### **Big-Endian** (*velké zakončení*):

- V tomto formátu je nejvíce významný bajt uložen na nejnižší paměťové adrese a nejméně významný bajt na nejvyšší paměťové adrese.

### **Little-Endian** (*malé zakončení*):

- U tohoto formátu je naopak nejméně významný bajt uložen na nejnižší paměťové adrese a nejvíce významný bajt na nejvyšší paměťové adrese.



## Základní datové typy

### — *Obecně* —

- Základními datovými typy je logická hodnota (pravda x nepravda), celé číslo, reálné číslo, znak a výčet prvků.
- Existuje také prázdný datový typ, který nezískává žádnou hodnotu. Složené datové typy vznikají skládáním výše uvedených jednoduchých typů.

### — *V programovacích jazycích* —

#### *Základní*

typ	velikost (bit)	popis
<a href="#">BOOL</a>	1	Boolean
<a href="#">BYTE (Byte)</a>	8	hexadecimální číslo / BCD
<a href="#">WORD (Word)</a>	16	binární zobrazení / hexadecimální číslo / BCD
<a href="#">DWORD (Double Word)</a>	32	binární zobrazení / hexadecimální číslo / BCD
<a href="#">INT (Integer)</a>	16	celé desítkové číslo se znaménkem
<a href="#">DINT (Double Integer)</a>	32	celé desítkové číslo se znaménkem
<a href="#">REAL (Reálné číslo)</a>	32	číslo s plovoucí desetinnou čárkou
<a href="#">TIME (IEC time)</a>	32	čas, integer se znaménkem, rozlišení 1 ms
<a href="#">DATE (IEC date)</a>	16	datum, rozlišení 1 den
<a href="#">TIME OF DAY (time)</a>	32	čas v rozsahu 1 den, rozlišení 1 ms
<a href="#">CHAR</a>	8	znak ASCII

#### *Komplexní*

typ	velikost	popis
<a href="#">DATE AND TIME</a>	64 bit	datum a čas
<a href="#">STRING</a>	254 Byte	textový řetězec
<a href="#">ARRAY</a>	*)	více rozměrová oblast dat (vektor, matice, ...) jednoho typu dat
<a href="#">STRUCT</a>	*)	strukturovaná oblast jednoho nebo různých typů dat

#### *Formální*

typ	velikost	popis
<a href="#">TIMER</a>	2 Byte	číslo časovače (integer)
<a href="#">COUNTER</a>	2 Byte	číslo čítače (integer)
<a href="#">BLOCK</a>	2 Byte	číslo bloku (programový nebo datový) (integer)
<a href="#">POINTER</a>	6 Byte	identifikátor proměnné a adresy, odkazuje na adresu proměnné
<a href="#">ANY</a>	10 Byte	pro použití, kdy je datový typ aktuálního parametru neznámý nebo lze použít libovolný typ
<a href="#">VARIANT</a>	proměnná	ukazatel na proměnné různých typů