

10. Zobrazenie informácie

- **Pomenujte základné typy informácií v PC**
- **Popíšte základné spôsoby kódovania rôznych typov informácií**

Údaje (informácie), ktoré program spracováva môžu byť **znaky**, **čísla** alebo **logické hodnoty**.

Nakoľko počítač pracuje len s **0** a **1** (binárna sústava) je nutné údaje kódovať

1) **Znak** (reťazec) - text. **Používajú sa rôzne kódy**, napr.:

- a) **ASCII** (*American Standard Code for Information Interchange*, doslovne *Americký štandardný kód pre výmenu informácií*) je kódovací systém znakov anglickej abecedy, číslíc, neviditeľných znakov, iných špeciálnych znakov a riadiacich kódov slúžiacich k riadeniu. Najskôr bol 7 bitový, čo umožňovalo kódovať 128 znakov, neskôr bol rozšírený na 8 bitov, čo je 256 znakov. Prvých 128 znakov je rovnakých a ďalších 128 znakov sú znaky pre národné potreby.
- b) **UNICODE** - toto kódovanie používa 16 a 24 bitov (2 či 3 bajty) na zakódovanie jedného znaku, čo umožňuje zakódovať cca 1,1 milióna možných znakov. (Na znak, na ktorý predtým stačil 8b, je potrebné viac 16 alebo 24b),
- c) **UTF-8** (*8-bitový Unicode Transformation Format*), používa 8 až 32 bitov na zapísanie znaku podľa toho, o aký znak sa jedná. Napríklad, len jeden UTF-8 bajt (8 bitov) je potrebný na zakódovanie všetkých 128 US-ASCII znakov, ďalšie znaky potrebujú viac bitov (bajtov).

To, koľko bajtov zaberá daný kód znaku je dané povinnými bitmi pre každú dĺžku znaku:

Dĺžka kódu Povinné bity

1 bajt **0xxxxxxx**

2 bajty **110xxxxx 10xxxxxx**

3 bajty **1110xxxx 10xxxxxx 10xxxxxx**

4 bajty **11110xxx 10xxxxxx 10xxxxxx 10xxxxxx**

d) **UTF-16, ..**

2) **Číslo** - **rôzne kódy**, napr.:

- a) **binárny kód** – desiatkové čísla sú prevedené do dvojkovej sústavy,
- b) **BCD kód** – každá desiatková číslica je zakódovaný štvoricou dvojkových číslic.

pr: $133_{10} = 10000101_2 = 0001\ 0011\ 0011_{BCD}$

Pozrime sa na oba typy kódov z hľadiska výhod/nevýhod:

Binárny: ++ ľahšia hardwarová realizácia, menšie požiadavky na pamäť,
-- nutnosť prevodu kódov,

BCD: ++ netreba prevody kódov,
-- ťažšia hardwarová realizácia, vyššie požiadavky na pamäť.

3) **logické** (boolovské) hodnoty - netreba kódovať.

Popíšte zobrazenie čísel v pevnej a pohyblivej rádovej čiarkke:

Reálne čísla sú čísla, ktoré majú aj necelú zlomkovú časť (pri desiatkových číslach hovoríme o desatinnej časti), t.j. čísla, ktoré vieme vyjadriť zlomkom.

Reálne čísla sa v počítači zobrazujú 2 spôsobmi:

1. s pevnou rádovou čiarkou
2. s pohyblivou (plávajúcou) rádovou čiarkou

Reálne čísla s pevnou rádovou čiarkou

V dvojkovej sústave sa dajú zobraziť aj necelé čísla rovnako ako v desiatkovej sústave:

$$27.43_{10} = 2 \cdot 10^1 + 7 \cdot 10^0 + 4 \cdot 10^{-1} + 3 \cdot 10^{-2}$$

$$1010.011_2 = 2^3 + 2^1 + 2^{-2} + 2^{-3} = 8 + 1 + 0.25 + 0.125 = 10.375_{10}$$

Pri zobrazení čísel v pevnej rádovej čiarkke určitý počet bitov je vyhradených pre celú časť čísla a určitý počet bitov pre zlomkovú časť čísla. Najviac významový bit je znamienkový. Napr.: pri 16 bitovom čísle, môže byť 8 bitov vyhradené pre celú časť a 8 bitov pre zlomkovú časť za rádovou čiarkou.



Takto zobrazené čísla majú **menší rozsah zobraziteľných hodnôt**, ale majú väčšiu rozlišovaciu schopnosť - presnosť. Tým, že majú pevne umiestnenú rádovú čiarku, môže pri matematických operáciách ľahko dôjsť k pretečeniu. Ak dôjde k pretečeniu za rádovou čiarkou a dôjde k zaokrúhleniu, to je vo väčšine prípadov akceptovateľné. Ak dôjde k pretečeniu v celej časti, výsledok je nepoužiteľný. Z toho jasne vyplýva, že čísla v pevnej rádovej čiarkke nie sú vhodné pre vedecké výpočty. Majú menej problémov so zaokrúhľovaním, preto sa niekedy používajú na výpočty s peniazmi, kde malý rozsah až tak nevádi.

Príklad:

Číslo s pevnou čiarkou je zobrazené v 8 bitovom zápise **01010010** s 3 bitmi pred rádovou čiarkou a 5 bitmi za ňou.

$$10.1001_2 = 2^1 + 2^{-1} + 2^{-4} = 2 + 0.5 + 0.0625 = 2.5625_{10}$$

Reálne čísla s pohyblivou (plávajúcou) rádovou čiarkou

Číslo 2308 sa dá zapísať aj ako $23.08 \cdot 10^2$ alebo $0.002308 \cdot 10^6$. Prednosť má ale zápis $2.308 \cdot 10^3$, ktorý má pred desatinnou bodkou práve jednu nenulovú číslicu. Nazývame to **normalizovaný tvar čísla**.

V počítačoch sa používa základ 2. Pri zobrazení čísel v pohyblivej rádovej čiarky sa teda čísla zobrazujú v tvare: **$x = \text{mantisa} * \text{základ}^{\text{exponent}}$** a teda je potrebné uložiť

2 hodnoty: mantisu a exponent, z ktorých obe môžu byť aj kladné aj záporné. Základ je vždy rovnaký.

Norma IEEE 754 - najrozšírenejší tvar čísla v pohyblivej rádovej čiarky. Väčšina procesorov používa práve tento tvar. Normalizované číslo má pred rádovou čiarkou nenulovú cifru. Ale v dvojkovej sústave **jediná nenulová cifra je 1**. Preto si ju nemusíme pamätať a ušetríme jeden bit. **$x =$**

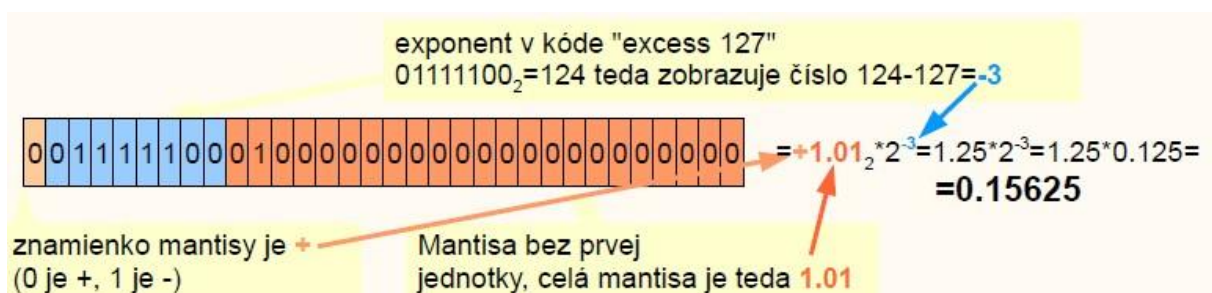
$1.\text{mantisa} * \text{základ}^{\text{exponent}}$

Binary32 (single precision) –formát jednoduchšej presnosti:

1 bit znamienko mantisy, 8 bitov exponent v kóde "excess 127" a 23 bitov absolútna hodnota mantisy bez prvej jednotky, teda s pevnou rádovou čiarkou pred prvou cifrou.

Pr.1

Prevod čísla z IEEE 754:



Pr.2

Prevod čísla z IEEE 754: **11000001001011000000000000000000**

Z = 1 (mínus)

EXP = **1000 0010**₂ = 130₁₀ v kóde "excess 127" -> $130 - 127 = 3$

Mantisa s prvou jednotkou = 1,01011

Výsledok = **-1.01011**₂ * 2³ = - (2⁰ + 2⁻² + 2⁻⁴ + 2⁻⁵) * 2³ = - (1 + 0,25 + 0,0625 + 0,03125) * 8 = -1,34375 * 8 = **-10,75**

Pr.3

Prevod čísla **-10,75** do IEEE 754 s presnosťou 32 bitov *Riešenie:*

$-10,75_{10} = -1010,11_2 = -1,01011 \cdot 2^3$ -> normalizácia mantisy t.j. posun rádovej čiarky o 3 miesta doľava

$Z = 1$ (mínus)

$EXP = 3 + 127 = 130_{10} = 1000\ 0010_2$ (8 bitov)

Mantisa = **0101 1000 0000 0000 0000 0000** (bez prvej jednotky – 23 bitov)

Výsledok v IEEE 754: **11000001001011000000000000000000**

Definujte IK a DK:

Inverzný kód

Kladné čísla sa zobrazujú ako v priamom kóde a začínajú vždy 0. Záporné čísla sa zobrazujú ako jednotkový doplnok čísla t.j. všetky bity čísla sa obrátia na opačné a začínajú vždy 1. Prvý bit teda hovorí o znamienku a vstupuje do výpočtu. Odčítanie v inverznom kóde vieme realizovať pomocou pričítania záporného čísla.

Podobne ako v priamom kóde máme **dve nuly** (-0) s kódom 1111 1111 a (+0) s kódom 0000 0000. Rozsah platných čísiel inverzného 8 bitového kódu je $(-127)_{10}$ s kódom 1000 0000 až $(+127)_{10}$ s kódom 0111 1111 t.j. $\langle -127; 127 \rangle$

$$+7_{10} = 0111_{IK}$$

$$-7_{10} = 1000_{IK}$$

Ak pri sčítaní 2 čísel v inverznom kóde vznikne **prenos z najvyššieho rádu, pripočíta sa k najnižšiemu rádu.**

Doplnkový kód

Kladné čísla sa zobrazujú ako v priamom kóde a začínajú vždy 0. Záporné čísla sa zobrazujú ako (jednotkový doplnok čísla +1) nazývaný aj dvojkový doplnok t.j. (inverzný kód + 1) a začínajú vždy 1. Prvý bit hovorí o znamienku a vstupuje do výpočtu. Operáciu odčítania, tak ako pri inverznom kóde vieme realizovať pomocou pričítania záporného čísla. Výhodou doplnkového kódu je, že má **len 1 nulu** a to 0000 0000.

Rozsah platných čísiel doplnkového 8 bitového kódu je $(-128)_{10}$ s kódom 1000 0000 až $(+127)_{10}$ s kódom 0111 1111 t.j. $\langle -128; 127 \rangle$.

$$+7_{10} = 0111_{DK}$$

$$-7_{10} = 1001_{DK}$$

Odlicanie cek Inwency "rod":

$34 \rightarrow 100010_{TK} \rightarrow 100010_{IK}$
 $-14 \quad -010001_{PK}$
 $\hline 14$

1010000
 $\xrightarrow{+1}$
 $010001_{IK} \rightarrow 14$

Wzrost 1 nie odpow!

$14 \rightarrow 010001_{PK} \xrightarrow{norm. 1} 010001_{IK}$
 $-34 \quad -100010_{PK} \xrightarrow{inw. 2} 011101_{IK}$
 $\hline -14$

$101110_{IK} \xrightarrow{16} 010001_L = -14$

$14 < 34$ (wynik jest IK)

$$\begin{array}{r}
 22 \rightarrow 10110_{PK} \rightarrow 110110_{DK} \\
 -12 \quad 10001_{PK} \rightarrow 01111_{DK} \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 01110_{IK} \\
 +1 \\
 \hline
 01111_{DK} \uparrow \\
 \boxed{DK = IK + 1}
 \end{array}$$

$$\begin{array}{r}
 12 \rightarrow 10001_{PK} \rightarrow 10001_{DK} \\
 -22 \quad -10110_{PK} \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 01001_{IK} \\
 +1 \\
 \hline
 01010_{DK}
 \end{array}$$

$$\begin{array}{r}
 11011_{DK} \\
 -1 \\
 \hline
 11010_{IK} \Rightarrow -0101 \Rightarrow -5_{10}
 \end{array}$$

Enamienkoy bit

Uved'te možnosti zobrazovania signálov osciloskopom:

Pri AO používame **CRT obrazovku**, zatiaľ čo pri DO používame **modernejšie displeje** ako napr.: **LCD, TFT**. Pri **viackanálových osciloskopoch** si vieme vybrať, že či chceme zobraziť **jeden kanál** alebo, či chceme ich zobraziť **viacero naraz**, alebo či ich chceme **aritmeticky sčítať** alebo **odčítať do jedného signálu**.

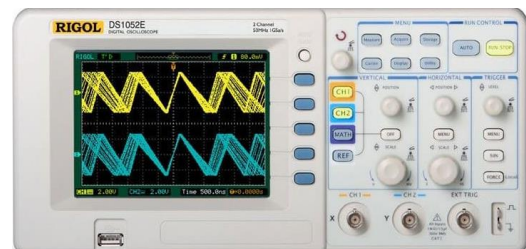
Ďalej pri viackanálových môžeme vyradiť **časovú základňu** (z Y-T na X-Y) a **nahradiť ju druhým kanálom**. Ďalej pri viackanálových metóda **CHOP** a **ALT** (viz. **elektronický prepínač** – 14.).

Pri **AO** je to voľne kmitajúca (AUTO/NORM) / spúšťaná (Single/Os-Z (externá)) časová základňa.

Pri **DO** je to možnosť zobrazovania jednorazových dejov, zobrazenie dopočítaných parametrov, zobrazenie signálu pred vznikom synchronizačnej jednotky, zobrazenie kurzorov, vlastné zobrazenie – rastrové alebo vektorové, zobrazovacia frekvencia (FPS displeja), aliasing, metódy vzorkovania....

OS-Y
OS-X

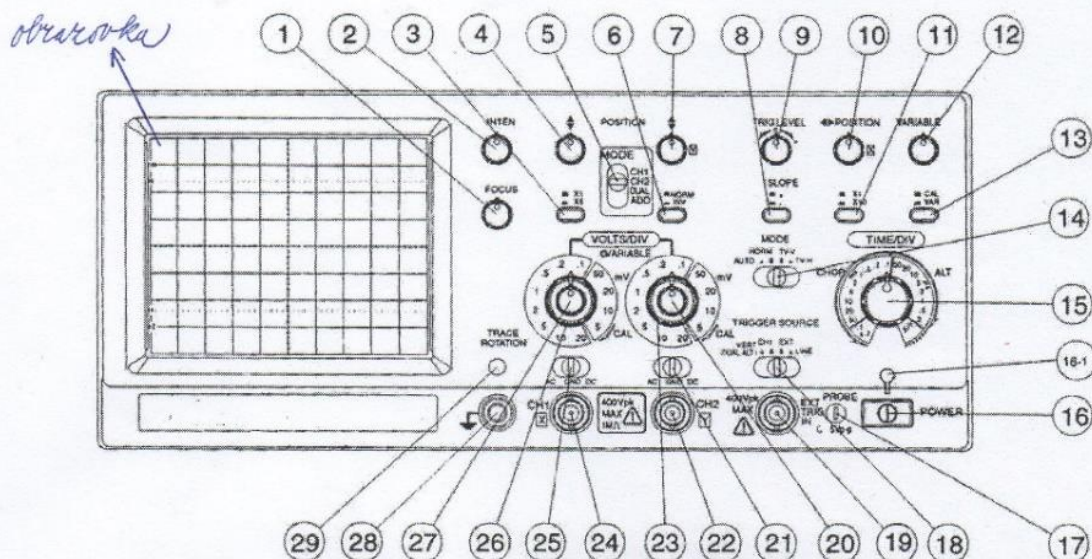
| <i>Analógový osciloskop</i> | <i>Digitálny osciloskop</i> |
|---|--|
| NORM = spúšťanie ČZ, je spustená s príchodom vst. signálu, dovtedy je obrazovka tmavá | SINGLE = jednorázové spustenie ČZ |
| GND, DC, AC, ČZ (X-Y; Y-T); ms/dielik, ROLL – zprava doľava (rolovanie priebehu, nečaká sa na zosynchronizovanie a akvizíciu, vhodné pre nízke frekvencie) | |



OSCILOSKOP:

4.C

Analogové osc. → pracujú v reálnom čase (nemajú oneskorenie)
 Digitálne osc. → majú malé oneskorenie, dokážu jednoducho spolupracovať s PC a inými digitálnymi zariadeniami



1. ostrosť (hrúbka čiar, zaostrenie obrázku)
2. jas
3. zväčšenie signálu 5x
4. a 7. posun v vertikálne nastavenie signálu
5. prepínač kanálov
6. inverzia signálu o 180°
8. prepínač + a - polohy zachytávacieho
9. nastavenie spúšťania čí sign.
10. horizontálne nastavenie signálu
11. rozťahnutie čí 10-násobne
12. variable čí
13. prepínač medzi čí a jej kalibráciou
14. funkcia spúšťacia čí
15. nastavenie čí
16. svetový vypínač
17. vstup pre sondu
18. voľba spúšťacieho zdroja
19. vstupný prúd pre externé zariadenie
- 20, 21. variabilná vstupná citlivosť
- 21, 25. voľba exponencie vstupu CH1 / CH2
- 22, 24. konektory kanál 1/2
23. vstupná citlivosť v diel
26. 1
28. mzmnenie
29. nastavenie vodivosti signálu