

Principy počítačů

Historie a vývoj výpočetní techniky

Předchůdci

- Abbakus (podobný princip jako počítadlo),
- mechanické kalkulátory (da Vinci, Schickard, Pascal, von Leibnitz, de Colmar),
- Babbageův analytický stroj (počítač poháněný párou),
- logaritmické pravítka, elektromechanické kalkulátory.

I. Generace

- vývoj začal během 2.SV
- neexistoval jednotný software – každý počítač – vlastní program zapsaný ve strojovém kódu,
- elektronky => velký rozměr počítačů,
- I/O – mechanická záznamová média – páska, děrovací štítky,
- náročné na údržbu,
- většina dat uložena na přenos. Médiiích
- hlavní paměť – pevné nevyměnitelné disky (40 až 50 kb),
- počítač mohla využívat pouze jedna osoba,
- výkon 10³ - 10⁴ oper/sec

III. Generace

- tranzistory – velké množství tepla – časté poškození
- 1958 - Jack St. Clair Kilby – integrovaný obvod
- velké zvýšení výkonu, zmenšení rozměrů
- konec dř. štítků, pásek, mag. jader, nástup paměťových disků
- zlepšení výstupu – LED diody, obrazovky
- vylepšení programovacích jazyků
- 1. osobní počítače a sálové superpočítače
- výkon až 10⁶ oper/sec

V. Generace

- matice mikroprocesorů
- transputery (TRANZISTOR a COMPUTER - kompletní počítač na čipu, který tvoří základní stavební jednotkou procesorového

II. Generace

- 3 nejnaléhavější problémy - zvýšení rychlosti provádění operací, zvýšení kapacity paměti a zvýšení rychlosti přenosu dat na vstupu a výstupu
- vyřešeno s příchodem polovodičů (1948, tranzistor, dioda)
- od roku 1956 používány v počítačích
- dochází ke zmenšování počítačů
- velký vývoj paměti – magnetická jádra, magnetické pásy, první diskové paměti
- 1. univerzální vyšší programovací jazyky
- výkon kolem 10⁵ oper/sec

IV. Generace

- 1968 – vylepšení integrovaných obvodů – miniaturizace
- mikroprocesory – 1971
- nová paměťová média – FDD, HDD, CD, DVD
- začátek masové výroby osobních počítačů určených pro použití v kanceláři i v rodinách
- neustálý vývoj nových technologií => zvyšování výkonu, miniaturizace, snížení pořizovacích nákladů
- dynamické paměti, BIOS
- výkon přes 10⁶ oper/sec
- Intel 4040 (1969,IBM,USA), Intel 8080(1972,USA, v ČSSR MHB8080A), Sinclair ZX Spectrum (GB, 1982), 80386SX (1988,USA), Pentium (1996,USA) systému)
- optické prvky od roku 1990
- výkon až 10⁹ oper/sec
- vstup a výstup povelů v lidské řeči

Číselné soustavy

Typy soustav	
Nepoziční soustava	Poziční soustavy
- pořadí nerozhoduje - chybí znaky pro záporná čísla - dlouhý zápis - původně římská čísla (později částečně poziční)	- dnes všechny běžně využívané - praktické

Běžně používané soustavy:

- 10 soustava
- 60 soustava – minuty, stupně
- 12 soustava – hodiny, měsíce
- 7 soustava – dny v týdnu

Desítková soustava – rozlišení 10-ti cifer 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- váha cifry je dána pozicí v čísle, 1869 chápeme jako:

$$1 \cdot 10^3 + 8 \cdot 10^2 + 6 \cdot 10^1 + 9 \cdot 10^0$$

- v číselné soustavě s obecným základem z lze číslo zapsat:

$$X = a_n \cdot z^n + a_{n-1} \cdot z^{n-1} + \dots + a_2 \cdot z^2 + a_1 \cdot z^1 + a_0 \cdot z^0$$

- zapisujeme: $a_n, a_{n-1}, \dots, a_2, a_1, a_0$

- základ z může být libovolný – nutnost použití právě z cifer

Binární soustava - pro počítačové systémy je však vhodnější dvojková (binární) soustava rozlišující dva stavy – 0 a 1 představující hodnotu elektrického napětí (0 V nebo > 0 V). Jednotkou informace je **bit**.

1 bit	1b		
1 0 1 0	4 b = tetráda	NIBL	
0 1 1 1 0 1 0 1	8 b = 1 Byte	1B	

zápis binárních hodnot: 0101 1100 1001 0111

Octalová (osmičková) soustava

- 3 binární bity = 1 octalová cifra
- rozsah 000 - 111
- cifry 0-7 -> tedy osm cifer ($2^3 = 8$)

Hexadecimální (šestnáctková) soustava

- 4 bity = 1 hexadecimální cifra
- 16 cifer ($2^4 = 16$)
- 1B = 2 hexadecimální cifry

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Převod mezi číselnými soustavami - princip je poměrně jednoduchý. Číslo, které chceme převést, dělíme neustále dvojkou, až dojdeme k nule, přičemž si zapisujeme zbytky po celočíselném dělení. Pokud chceme převést číslo do jiné soustavy, například do šestnáctkové, budeme dělit šestnáctkou. Pokud do šestkové, dělíme šestkou. Takže v praxi to bude vypadat takto:

$(120)_{10} = (???)_2$

120 : 2 = 60 → 0 (zbytek)
 60 : 2 = 30 → 0
 30 : 2 = 15 → 0
 15 : 2 = 7 → 1
 7 : 2 = 3 → 1
 3 : 2 = 1 → 1
 1 : 2 = 0 → 1

výsledek zapisujeme ↑

Výsledek jsou právě ty zbytky, ale pozor na to, je tady drobná zrada, musíte brát zbytky odspodu, nikoliv svrchu. Takže výsledek bude číslo **1111000**.

Zkouška: $1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 64 + 32 + 16 + 8 = 120$

$(130)_{10} = (???)_7$

130 : 7 = 18 → 4 (zbytek)
 18 : 7 = 2 → 4
 2 → 2

výsledek zapisujeme ↑

Výsledek je **244**.

Zkouška: $2 \cdot 7^2 + 4 \cdot 7^1 + 4 \cdot 7^0 = 98 + 28 + 4 = 130$

$(11\ 0101\ 0100)_2 = (???)_{16}$

0011 0101 0100
 3 5 4

Dvojkové číslo rozdělíme zprava na čtveřice. Pokud první čtveřice není úplná, doplníme zleva nulami. Převedeme jednotlivé čtveřice do šestnáctkové soustavy. Složíme výsledek z dílčích výsledků.

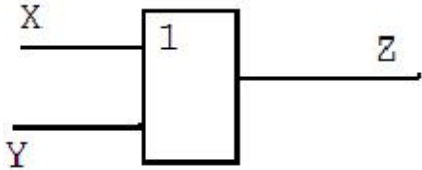
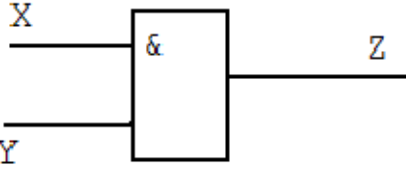
$(11\ 0101\ 0100)_2 = (354)_{16}$

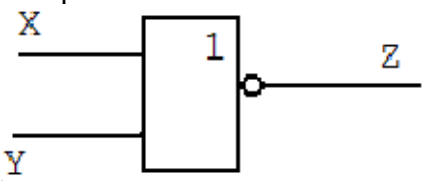
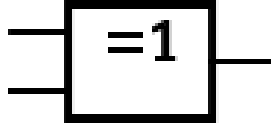
Zkouška:

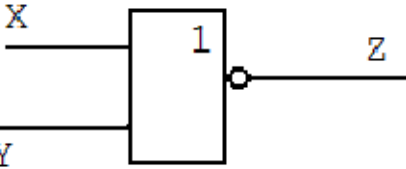
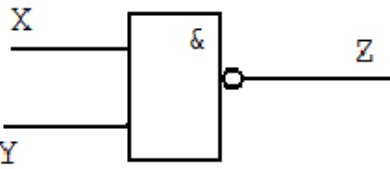
$(11\ 0101\ 0100)_2 = 1 \cdot 2^9 + 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 512 + 256 + 64 + 16 + 4 = (852)_{10}$
 $(354)_{16} = 3 \cdot 16^2 + 5 \cdot 16^1 + 4 \cdot 16^0 = (852)_{10}$

Booleovská algebra a logika

- zabývá se logickými operacemi ve dvojkové soustavě
- v praxi jsou tyto operace vykonávány na hradlech
- popř. klopné obvody s několika stabilními a nestabilními stavy
- dnes spíše integrované obvody, které nabízejí více funkcí \

<p>Logický součet – OR $X \vee Y = Z; X + Y = Z$</p> <p>Pokud platí jeden ze vstupů X a Y, pak platí výstup Z</p> <div style="display: flex; align-items: center; justify-content: center;">  <table border="1" style="margin-left: 20px;"> <thead> <tr><th>X</th><th>Y</th><th>Z</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> </div>	X	Y	Z	0	0	0	0	1	1	1	0	1	1	1	1	<p>Logický součin – AND $X \wedge Y = Z; X * Y = Z$</p> <p>Pokud platí oba vstupy X a Y, pak platí výstup Z</p> <div style="display: flex; align-items: center; justify-content: center;">  <table border="1" style="margin-left: 20px;"> <thead> <tr><th>X</th><th>Y</th><th>Z</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> </div>	X	Y	Z	0	0	0	0	1	0	1	0	0	1	1	1
X	Y	Z																													
0	0	0																													
0	1	1																													
1	0	1																													
1	1	1																													
X	Y	Z																													
0	0	0																													
0	1	0																													
1	0	0																													
1	1	1																													

<p>Negace</p> <p>Pokud platí vstup X, pak výstup Y neplatí</p> 	$\neg X = Y; X = \neg Y$	<p>XOR</p> <p>Výstupem je exkluzivní logický součet vstupů. Výstup je log.1 tehdy a jen tehdy pokud se hodnoty vstupů liší.</p> 	$X \text{ xor } Y = Z$ <table border="1" style="margin: auto;"> <thead> <tr> <th>X</th> <th>Y</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	X	Y	Z	0	0	0	1	0	1	0	1	1	1	1	0
X	Y	Z																
0	0	0																
1	0	1																
0	1	1																
1	1	0																

<p>Negovaný logický součet – NOR</p> <p>Pokud neplatí ani jeden ze vstupů X a Y, pak platí výstup Z</p> 	$\overline{X+Y} = Z$ <table border="1" style="margin: auto;"> <thead> <tr> <th>X</th> <th>Y</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	X	Y	Z	0	0	1	0	1	0	1	0	0	1	1	0	<p>Negovaný logický součin – NAND</p> <p>Pokud zároveň neplatí vstupy X a Y, pak platí výstup Z</p> 	$\overline{X*Y} = Z$ <table border="1" style="margin: auto;"> <thead> <tr> <th>X</th> <th>Y</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	X	Y	Z	0	0	1	0	1	1	1	0	1	1	1	0
X	Y	Z																															
0	0	1																															
0	1	0																															
1	0	0																															
1	1	0																															
X	Y	Z																															
0	0	1																															
0	1	1																															
1	0	1																															
1	1	0																															

Axiomy:

- **komutativa** (nezávislost na pořadí operandů): $X \text{ and } Y = Y \text{ and } X$
- **distributiva** (možnost různé distribuce funkce přes jinou): $(X \text{ and } Y) \text{ and } Z = X \text{ and } (Y \text{ and } Z)$
- **neutralita**: $X \text{ and } 1 = X; X \text{ or } 0 = X$
- **komplementarita**

Modulace signálu

Ideálním průběhem analogového signálu je spojitá hladká křivka, nejčastěji sinusovka. Ideálním průběhem digitálního signálu jsou pak obdélníkové pulsy. Ale bohužel na žádném přenosovém kanále se nepodaří přenést ideální signál. Vždy je více či méně degradován (znehodnocován) "šumem" a "útlumem".

Útlum: při přenosu signálu po přenosovém kanále (mediu) vždy dochází k energetickým ztrátám. Amplituda přenášeného signálu tedy slábne úměrně ke vzdálenosti.

Šum: jedná se o vznik nežádoucích, slabších či silnějších, signálů jiné frekvence než je přenášený signál. Skládáním těchto signálů s přenášeným dochází k "interferenci", tvar přenášeného signálu se

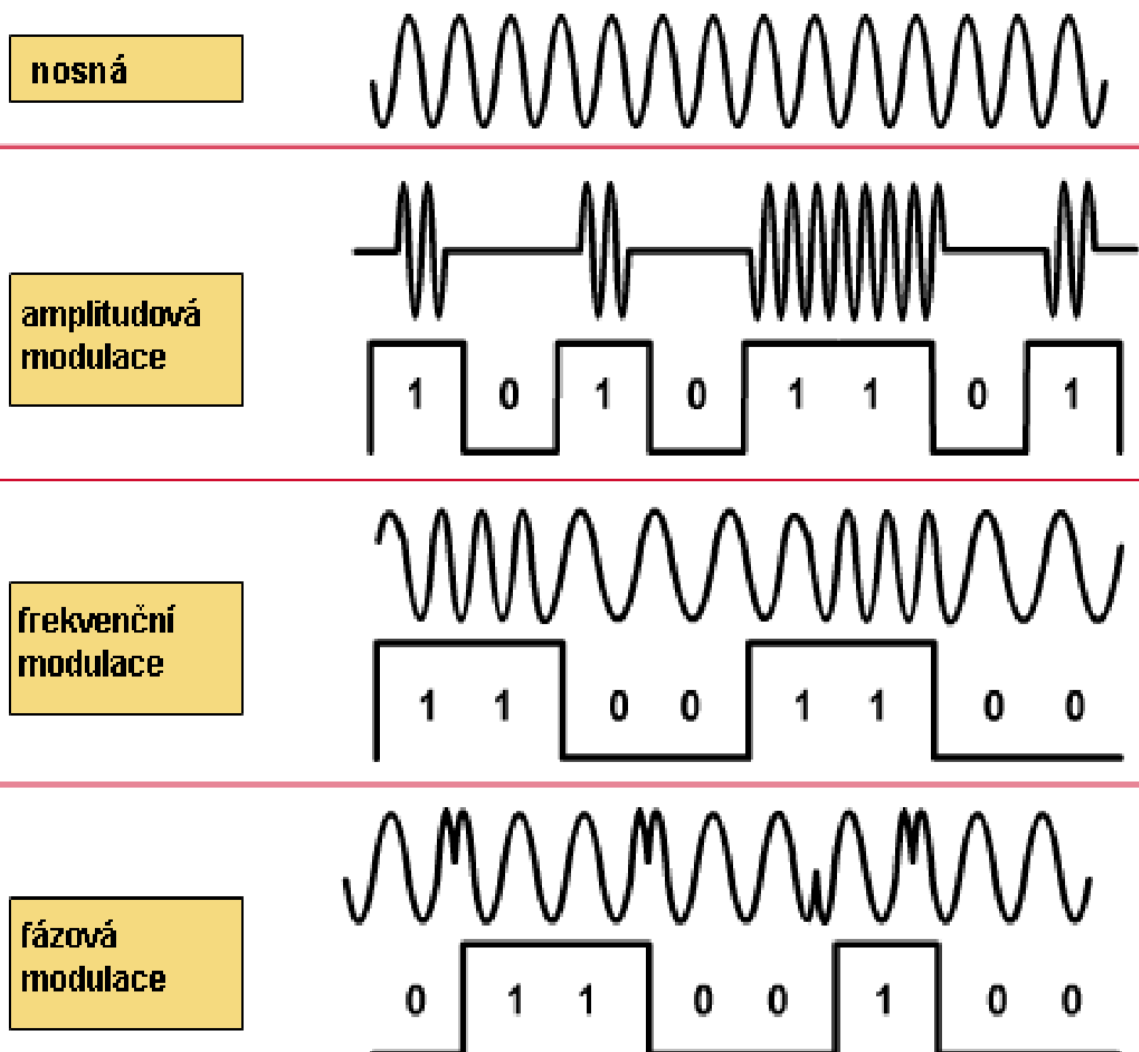
mění.

Příčina vznik rušivých signálů - šumů – je závislá na na nosném mediu. Například při přenosu signálu elektrickým obvodem - vodičem - může docházet k elektrické magnetické indukci od souběžného elektrického vedení . Rušivé signály můžeme eliminovat vhodnou volbou kabelu (např. kroucená dvoulinka), stíněním a zemněním, apod.

Zařízení, které provádí modulaci se nazývá modulátor. Opakem modulace je demodulace, kterou provádí demodulátor.

Známe různé druhy modulace:

- *Amplitudová modulace* - podle změn modulovaného signálu se mění amplituda nosného signálu. Nevýhodou tohoto způsobu je především situace, kdy je modulovaný signál nulový. Pak je i nosný signál nulový!
- *Frekvenční modulace* - podle změn modulovaného signálu se mění frekvence nosného signálu. Nosný signál má stále stejnou amplitudu. Oproti amplitudové modulaci tedy nemůže dojít k záměně výpadku vysílače s nulovým signálem. Velké změny nosného kmitočtu však někdy mohou činit problémy s naladěním přijímače.
- *Fázová modulace* - podle změn modulovaného signálu se mění fáze (dochází k fázovému posunu) nosného signálu. Eliminuje nevýhody obou předešlých typů, ale je složitá na realizaci.



Pojem KÓDOVÁNÍ:

Tento pojem nabývá smyslu právě při přenosu digitálního signálu, který nese pouze dva stavy - hodnoty "0" a "1". V různých elektronických obvodech jsou tyto dva stavy reprezentovány různými hodnotami. Např. v TTL logice je "0" reprezentována hodnotou 0V a "1" hodnotou 5V. Přenášíme-li takovýto digitální signál pomocí modulace, tak například amplitudová modulace při přenosu řady signálu s hodnotou "0" by prakticky nepřenášela nic. Proto digitální signál kódujeme.

Různé druhy kódování digitálního signálu:

