

# Řízení experimentu počítačem F3300

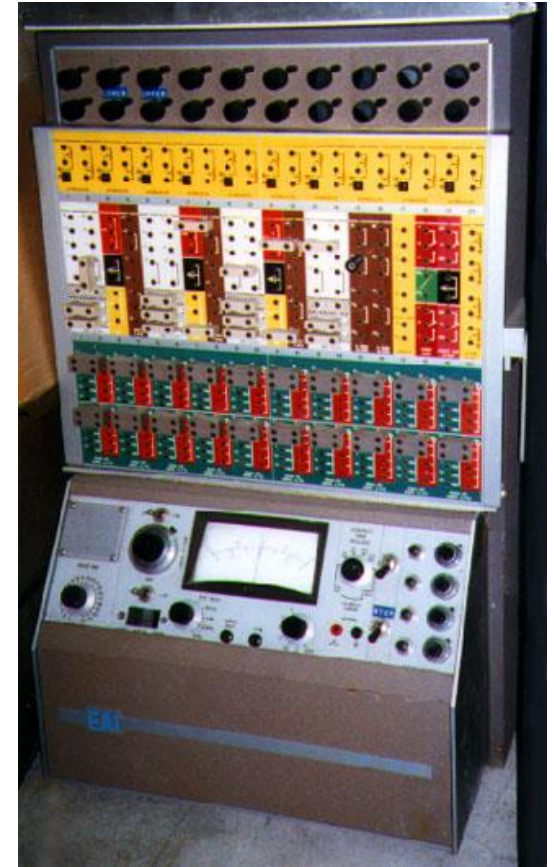
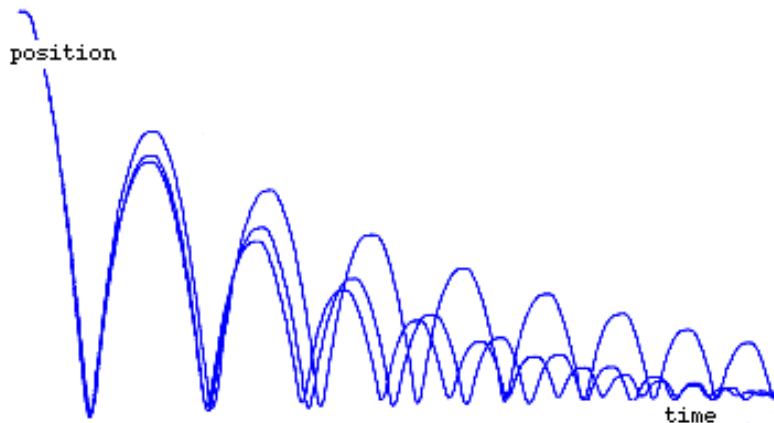
Počítač

# Počítač

- analogový
- číslicový (digitální)

# Elektronický analogový počítač

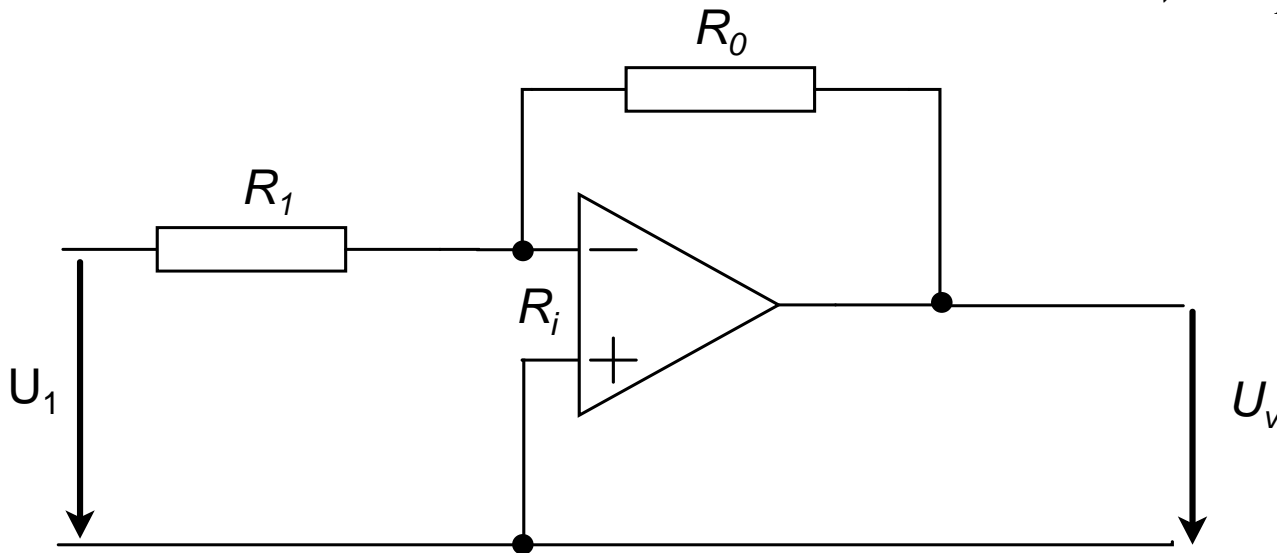
- Elektronický analogový počítač byl v principu stovebnice složená se základních obvodů pro analogové zpracování elektrických signálů – operační zesilovače, funkční měniče, komparátory, spínače
- Studované problémy byly nejprve vyřešeny matematicky (tj. byla nalezena např. rovnice, popisující řešení)
- Vyřešit problém pomocí AP znamenalo sestavit obvod, který byl popsán stejnou rovnicí a tedy simuloval studovaný děj.
- Typicky: hledaná funkce  $f(x)$  byla určena měřením závislosti napětí na čase  $U(t)$



# Operační zesilovač

- širokopásmový zesilovač se stejnosměrným vstupem, s velkým vstupním odporem  $R_i$  řádu stovky  $k\Omega$  až několika  $M\Omega$ , s výstupním odporem řádu  $100\ \Omega$  a velkým zesílením větším než  $10^4$ .
- V ideálním případě  $R_i \rightarrow \infty$ , a pak

$$U_v = -\frac{R_0}{R_1} U_1$$

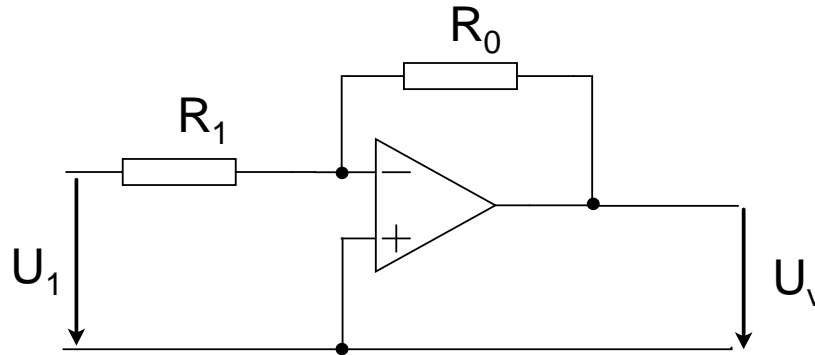


# Některé operace

- násobení konstantou

$$y = k \cdot x$$

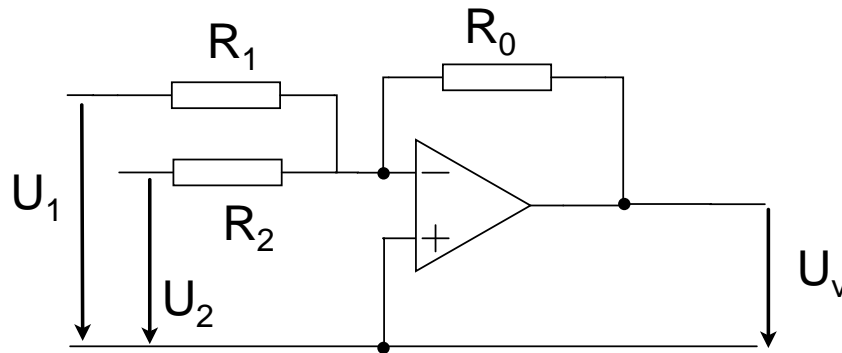
$$U_v = -\frac{R_0}{R_1} U_1$$



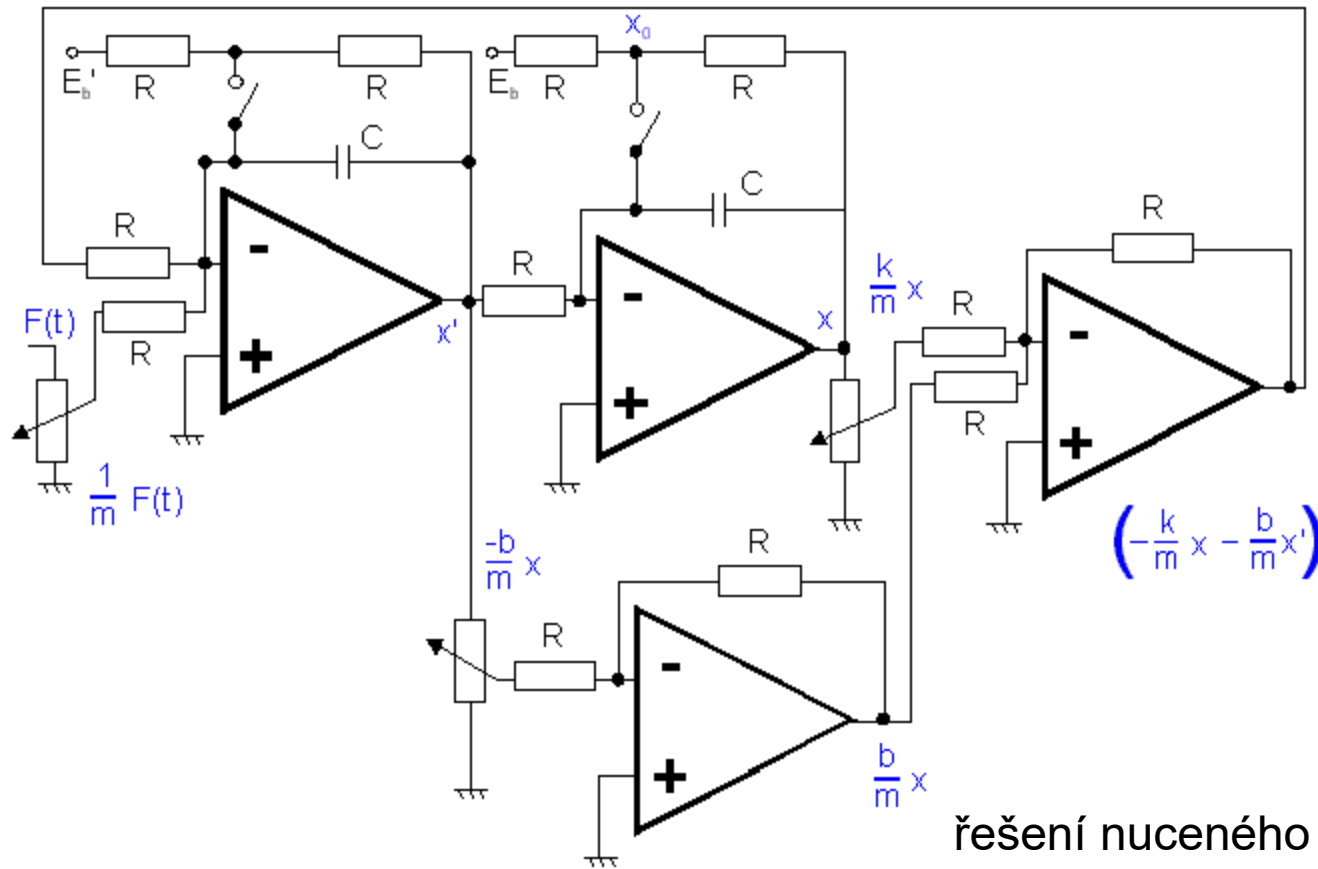
- sčítání dvou čísel

$$y = x_1 + x_2$$

$$U_v = -\left(\frac{R_0}{R_1} U_1 + \frac{R_0}{R_2} U_2\right)$$



# Příklad analogového počítače



řešení nuceného kmitavého pohybu

$$\frac{d^2 x(t)}{dt^2} + \frac{b}{m} \frac{dx(t)}{dt} + \frac{k}{m} x(t) = \frac{F(t)}{m}$$

# Nevýhody analogového počítače

- malá přesnost při provádění výpočtů, korektnost modelu, neznámý vliv okolí, systematické a náhodné chyby měření
- počítač je stavebnice, v určitém zapojení řeší zcela konkrétní problém – jde spíše o speciální kalkulátor
- neuniverzálnost – úlohy nematematického charakteru se provádějí „těžce“
- **Ale:** nastavenou činnost jednoduchého charakteru vykonává bez nutnosti užít DT → dnešní použití je např. předzpracování signálu

# Diskretizace

- Původně analogová veličina (s oborem reálných čísel) je uměle omezena na konečnou množinu diskrétních hodnot.
- Interval mezi hodnotami (krok) je zvolen tak, aby se jednotlivé hodnoty daly snadno odlišit.
- Diskrétními hodnotami se kódují jednotlivé číslice, nutnost řešení vztahu mezi řády.

## Logika TTL:

0 repr. napětí 0 – 0,8 V;

1 repr. napětí 2 – 5 V.





# Dvojková soustava

- Je snadnější zkonstruovat elektronické prvky, u kterých budou dobře rozlišeny dva stavy než deset. Každá číslice může proto nabývat jen dvou hodnot.
- Desítková soustava reprezentuje čísla v násobcích 10, číslice 0 – 9. Dvojková soustava reprezentuje čísla v násobcích 2, číslice 0 – 1.

$$235_{10} = 2 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0$$

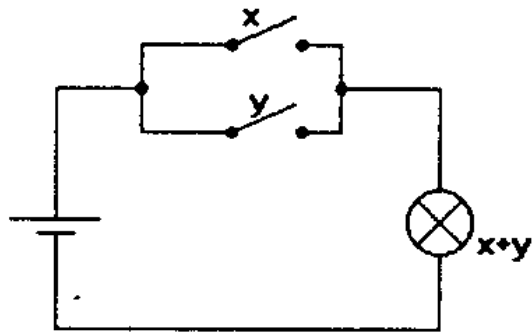
$$110_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6_{10}.$$

Sčítání binárních čísel probíhá podobně jako v desítkové soustavě, k přenosu do vyššího řádu však dochází, získá-li součet cifer hodnotu 2.

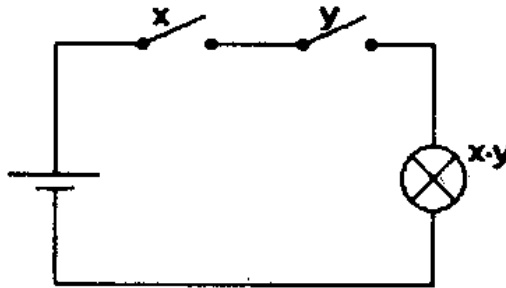
- Binární číslice (resp. místo, kde je číslice uložena) je tzv. bit (binary digit).

# Boolova algebra

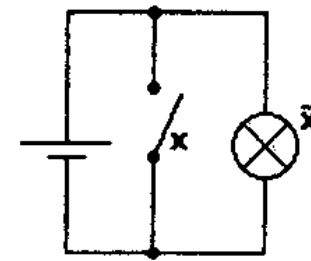
- dvě hodnoty: 1 „pravda“, 0 „nepravda“
- tři základní operace: logický součet  $+$ , logický součin  $\cdot$ , negace  $\bar{\phantom{x}}$



součet



součin



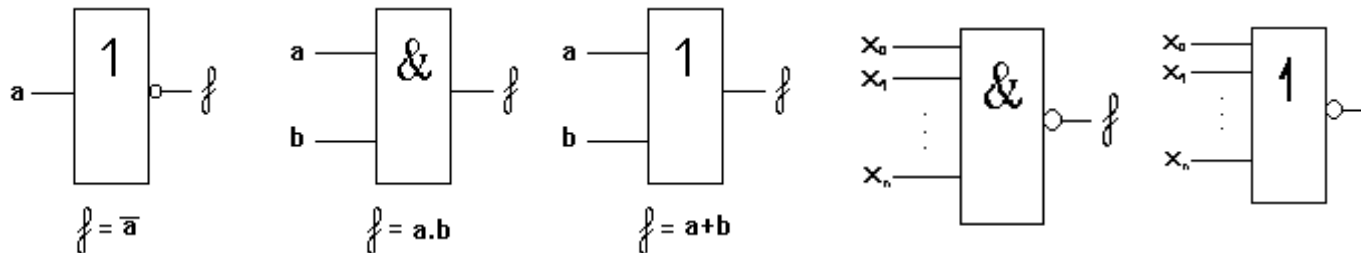
negace

naivní realizace logických operací

# Boolova algebra

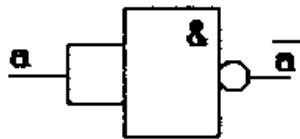
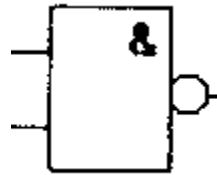
- dvě hodnoty: 1 „pravda“, 0 „nepravda“
- tři základní operace: negace  $\bar{\phantom{a}}$ , logický součin  $\cdot$ , logický součet  $+$

| a | b | $\bar{a}$<br>NOT | $a \cdot b$<br>AND | $a + b$<br>OR | $\overline{a \cdot b}$<br>NAND | $\overline{a + b}$<br>NOR | $a \oplus b$<br>XOR |
|---|---|------------------|--------------------|---------------|--------------------------------|---------------------------|---------------------|
| 0 | 0 | 1                | 0                  | 0             | 1                              | 1                         | 0                   |
| 0 | 1 | 1                | 0                  | 1             | 1                              | 0                         | 1                   |
| 1 | 0 | 0                | 0                  | 1             | 1                              | 0                         | 1                   |
| 1 | 1 | 0                | 1                  | 1             | 0                              | 0                         | 0                   |

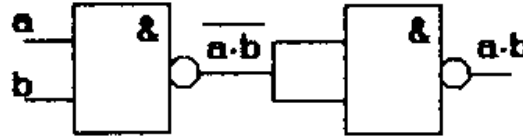


# Shefferova algebra

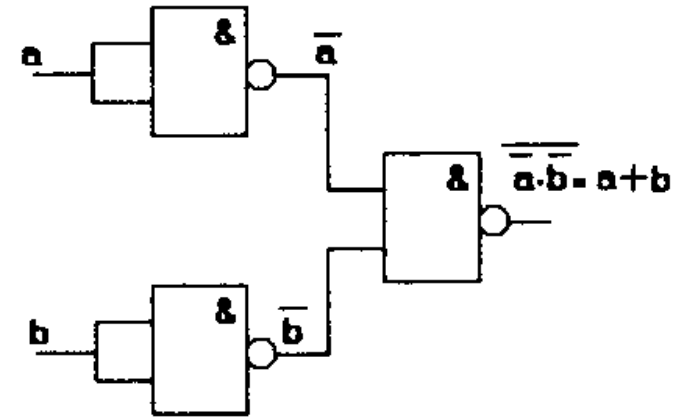
- všechny operace lze vytvořit pomocí jediné funkce (hradla) NAND



negace



součin



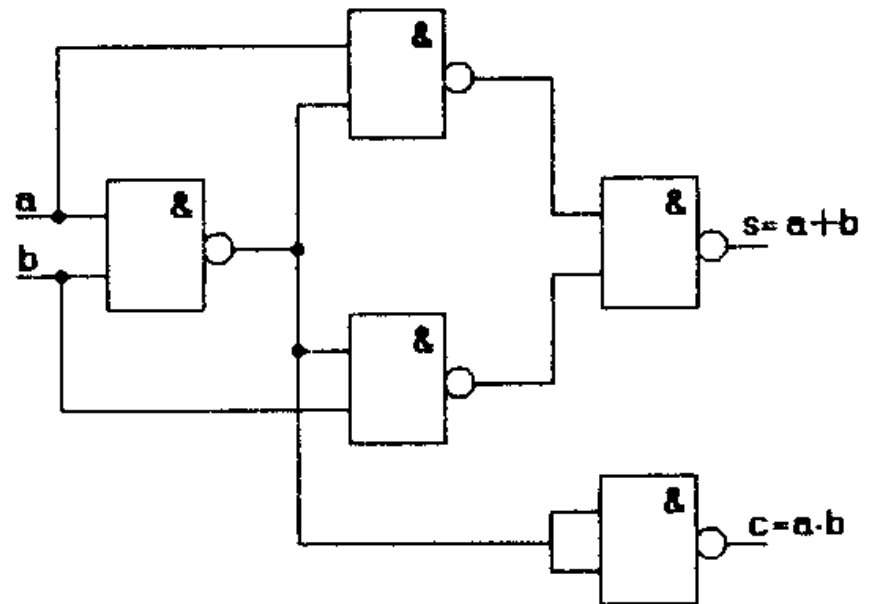
součet

- Pierceova algebra – využívá funkci NOR

# Polosčítačka

- logický součet nemá vlastnosti aritmetického součtu
- tuto vlastnost má operace XOR, vyžaduje však doplnění o určení, zda došlo k přenosu jedničky do vyššího řádu

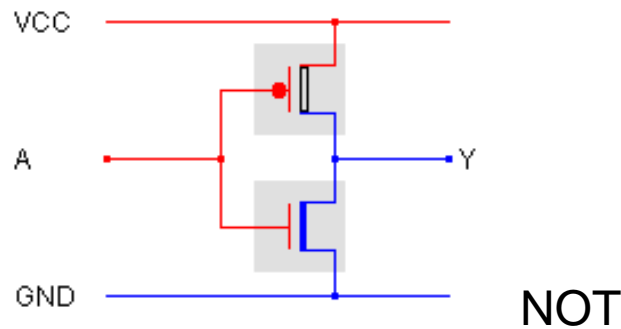
| a | b | $a \oplus b$<br>XOR | $a \cdot b$ |
|---|---|---------------------|-------------|
| 0 | 0 | 0                   | 0           |
| 0 | 1 | 1                   | 0           |
| 1 | 0 | 1                   | 0           |
| 1 | 1 | 0                   | 1           |



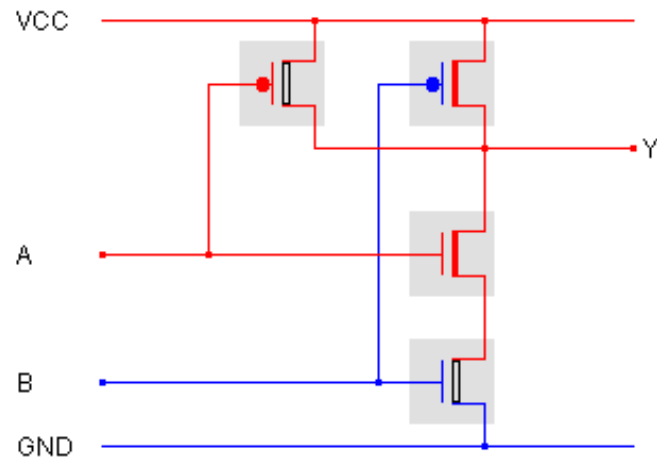
realizace pomocí hradel NAND

# Realizace hradel

- 0. generace: relé
- 1. – 3. generace: elektronky, tranzistory, integrované obvody
- Př. unipolární tranzistor technologie CMOS (*Complementary Metal-Oxide Semiconductor*)



Colors: [VCC] [GND] [Z (floating)] [Short-circuit]



| B | A | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Colors: [VCC] [GND] [Z (floating)] [Short-circuit]

# Počítač x samočinný kalkulátor x kontroler

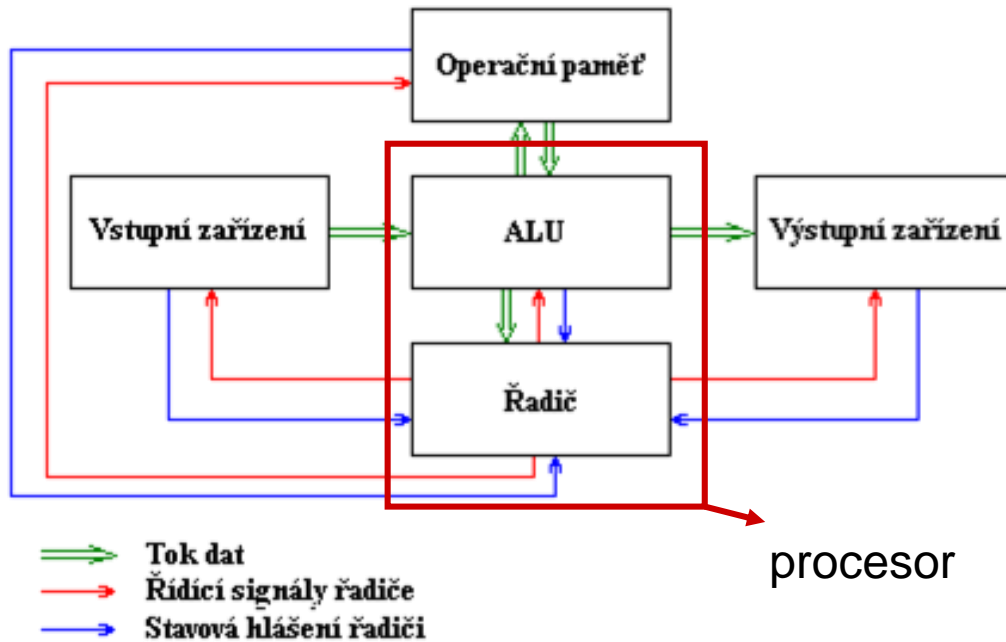


- Kalkulátor je výpočetní stroj s pevně stanoveným programem činnosti. Ten je „zadrátován“ v přístroji a jeho změna je obtížná, ne-li nemožná.
- Počítač je univerzální výpočetní stroj, řízený vkládaným programem. Program je načítán spolu se vstupními daty.
- Kontroler je výpočetní stroj, u kterého se změna úlohy častone dělá. provádí změnou firmware. Firmware se mění při upgradu.

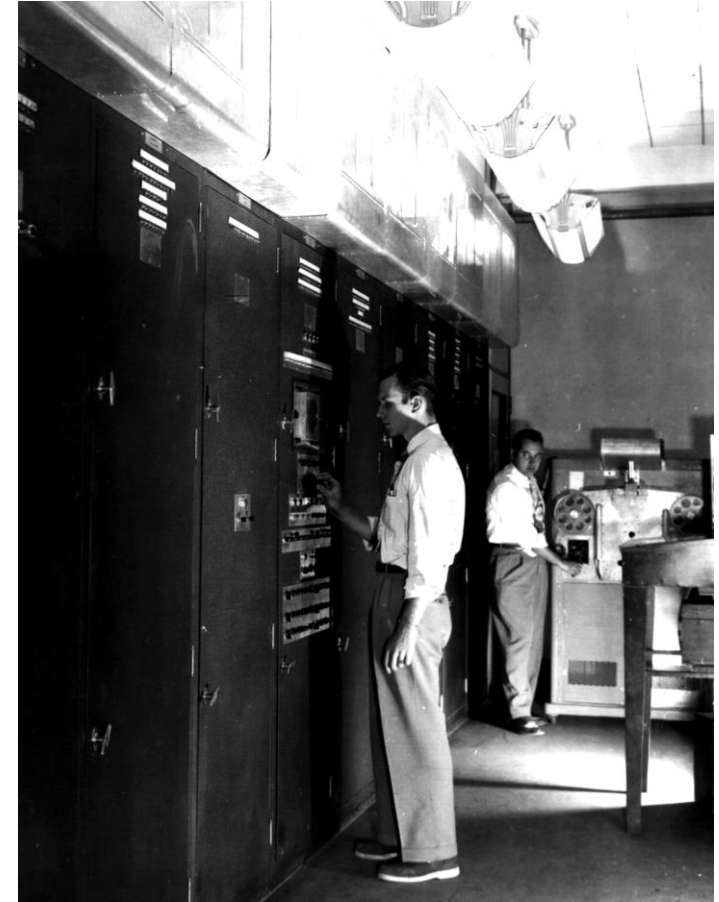
Charles Babbage – návrh tzv. *Analytical Engine* 1834-1871

# von Neumannova architektura

- Program je umístěn ve stejné paměti jako data.



zdroj Pelikán: Architektura osobních počítačů



1. umístění programu a dat do paměti přes vstupní zařízení
2. řadič interpretuje program, výpočet s daty v paměti provádí ALU
3. výstup výsledků přes výstupní zařízení



# von Neumannova architektura

- Struktura počítače je nezávislá na typu řešené úlohy, počítač se programuje obsahem paměti.
- Následující krok počítače je závislý na kroku předchozím.
- Instrukce a operandy (tj. data) jsou v téže paměti.
- Paměť je rozdělena do buněk stejné velikosti, jejich pořadová čísla se využívají jako adresy.
- Program je tvořen posloupností instrukcí, ty se vykonávají jednotlivě v pořadí, v jakém jsou zapsány do paměti.
- Změna pořadí provádění instrukcí se provede instrukcí podmíněného nebo nepodmíněného skoku.
- Pro reprezentaci instrukcí, čísel, adres a znaků se používá dvojková číselná soustava.

# von Neumannova architektura

Rozdíly vůči dnešku:

- multitasking
- paralelismus, multiprocessing
- interaktivní režim
- V/V zařízení
- OS x aplikační software

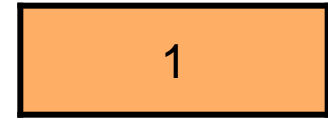
# Harvardská architektura

- podle počítače Mark 1 (Harvard University, 1944)
- oddělená paměť pro program a data (von Neumannova arch. používá společnou paměť pro program i data)
- výhody: vyšší rychlost při načítání, různé druhy paměti (typ, šířka slova), bezpečnost
- časté využití u mikrokontrolerů (PIC, Atmel-AVR)
- prvky i u PC (rozdělené cache paměti L1 na čipu procesoru)

# RAM

## Random Access Machine

- výpočetní model (model počítače na úrovni assembleru)



počítadlo instrukcí



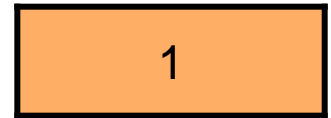
READ číslo | adresa vstupního registru  
STORE adresa pracovního registru  
LOAD číslo | adresa pracovního registru  
ADD číslo | adresa pracovního registru  
SUB číslo | adresa pracovního registru  
JUMP číslo instrukce  
JPOS číslo instrukce  
JZERO číslo instrukce  
JNEG číslo instrukce  
HALT

program (konečná posloupnost instrukcí)



# RAM

vstupní registry



počítadlo instrukcí

```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



pracovní registry

# RAM

vstupní registry

|   |   |   |   |     |
|---|---|---|---|-----|
| 5 | 3 | 0 | 0 | ... |
|---|---|---|---|-----|

|   |
|---|
| 2 |
|---|

počítadlo instrukcí

```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```

|   |         |
|---|---------|
| 5 | stradač |
| 0 |         |
| 0 |         |
| 0 |         |

pracovní registry

# RAM

vstupní registry

|   |   |   |   |     |
|---|---|---|---|-----|
| 5 | 3 | 0 | 0 | ... |
|---|---|---|---|-----|

3

počítadlo instrukcí

```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```

|   |         |
|---|---------|
| 5 | stradač |
| 0 |         |
| 0 |         |
| 0 |         |

pracovní registry

# RAM

vstupní registry



počítadlo instrukcí

```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



pracovní registry



# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



počítadlo instrukcí



pracovní registry

# RAM

vstupní registry

|   |   |   |   |     |
|---|---|---|---|-----|
| 5 | 3 | 0 | 0 | ... |
|---|---|---|---|-----|

6

počítadlo instrukcí

```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```

|   |         |
|---|---------|
| 3 | stradač |
| 5 |         |
| 0 |         |
| 0 |         |

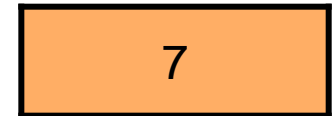
pracovní registry

# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



počítadlo instrukcí



pracovní registry

# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



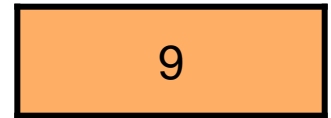
počítadlo instrukcí



pracovní registry

# RAM

vstupní registry



počítadlo instrukcí

```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



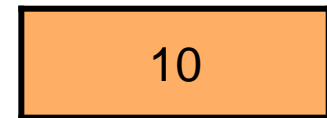
pracovní registry

# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



počítadlo instrukcí



pracovní registry

# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



počítadlo instrukcí



pracovní registry

# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```

12

počítadlo instrukcí



pracovní registry



# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



počítadlo instrukcí



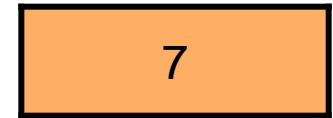
pracovní registry

# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```



počítadlo instrukcí



pracovní registry

# RAM

vstupní registry



```
READ 1
JZERO 13
STORE 1
READ 2
JZERO 13
STORE 2
LOAD 3
ADD 1
STORE 3
LOAD 2
SUB =1
JPOS 6
LOAD 3
HALT
```

*jako výsledek je chápán obsah střadače  
(nebo i obsah pracovních registrů)*

14

počítadlo instrukcí

15 stradač

5

1

15

pracovní registry

# Mikroprocesor 8080

Osmibitový procesor

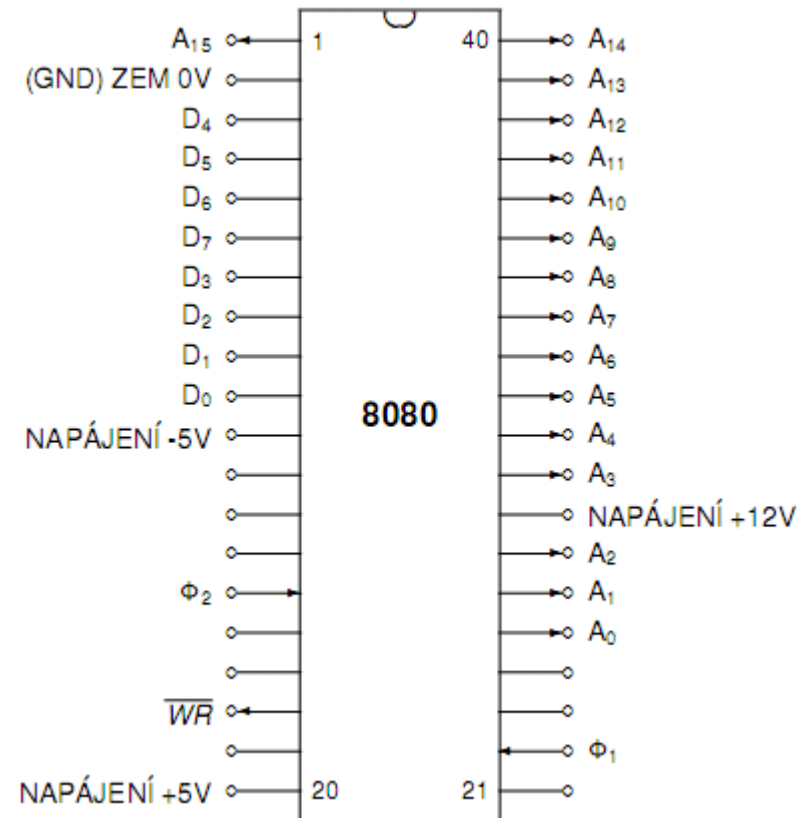
- data 8 bit, adresa 16 bit

PC (Program Counter) 16 bit

- registr s adresou prováděné instrukce

A (Accumulator, střadač) 8 bit

- registr, nad kterým se provádějí operace



# Instrukce 8080

## Příklady instrukcí

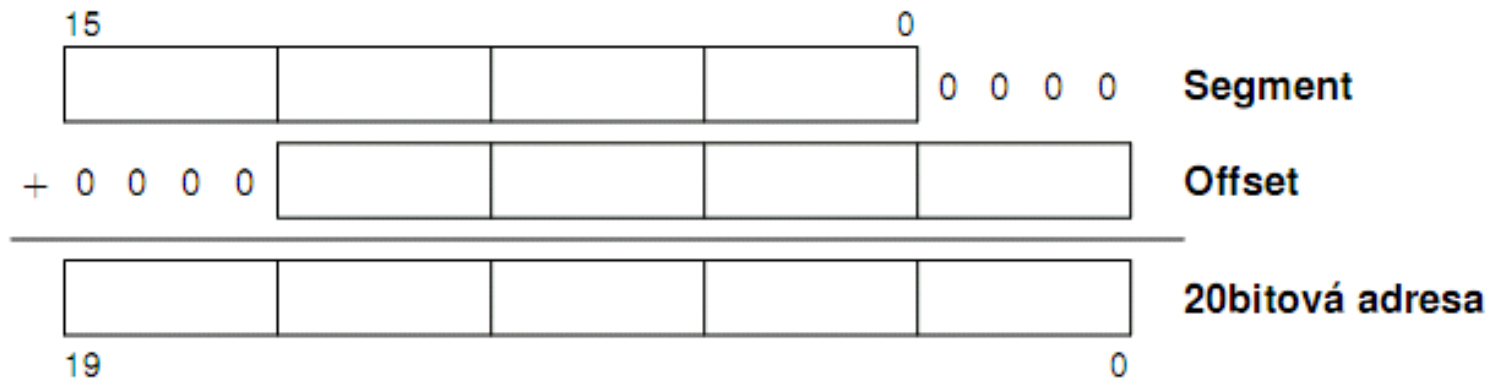
- LDA (Load A Direct) 3Ah <nižší bajt> <vyšší bajt>
- STA (Store A Direct) 32h <nižší bajt> <vyšší bajt>
- OUT
- IN
  
- JMP (Jump Unconditional) C3h <nižší bajt> <vyšší bajt>
- JZ (Jump Zero) CA <nižší bajt> <vyšší bajt>
  
- ADD registr
- MOV cíl. registr zdroj. registr

# Mikroprocesor 8080 - opcodes

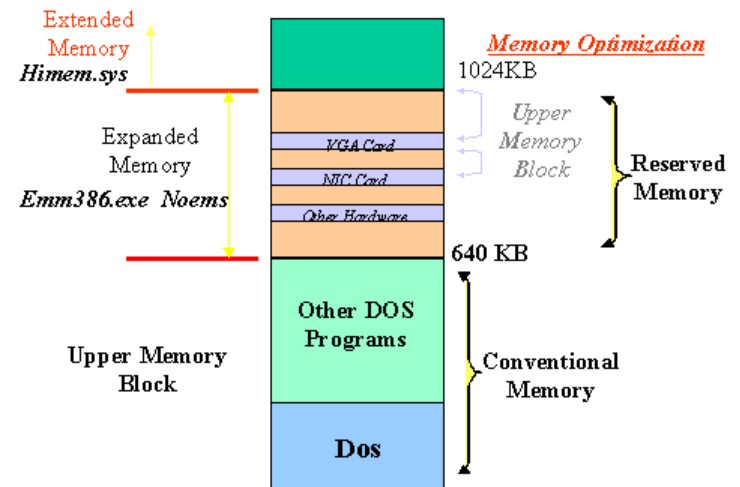
|    | x0                        | x1                           | x2                        | x3                        | x4                         | x5                         | x6                         | x7                        | x8                        | x9                        | xA                        | xB                        | xC                        | xD                        | xE                         | xF                        |  |
|----|---------------------------|------------------------------|---------------------------|---------------------------|----------------------------|----------------------------|----------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|----------------------------|---------------------------|--|
| 0x | NOP<br>1 4<br>---         | LXI B,d16<br>3 10<br>---     | STAX B<br>1 7<br>---      | INX B<br>1 5<br>---       | INR B<br>1 5<br>S Z A P -  | DCR B<br>1 5<br>S Z A P -  | MVI B,d8<br>2 7<br>---     | RLC<br>1 4<br>---         |                           | DAD B<br>1 10<br>---      | LDAX B<br>1 7<br>---      | DCX B<br>1 5<br>---       | INR C<br>1 5<br>S Z A P - | DCR C<br>1 5<br>S Z A P - | MVI C,d8<br>2 7<br>---     | RRC<br>1 4<br>---         |  |
| 1x |                           | LXI D,d16<br>3 10<br>---     | STAX D<br>1 7<br>---      | INX D<br>1 5<br>---       | INR D<br>1 5<br>S Z A P -  | DCR D<br>1 5<br>S Z A P -  | MVI D,d8<br>2 7<br>---     | RAL<br>1 4<br>---         |                           | DAD D<br>1 10<br>---      | LDAX D<br>1 7<br>---      | DCX D<br>1 5<br>---       | INR E<br>1 5<br>S Z A P - | DCR E<br>1 5<br>S Z A P - | MVI E,d8<br>2 7<br>---     | RAR<br>1 4<br>---         |  |
| 2x |                           | LXI H,d16<br>3 10<br>---     | SHLD a16<br>3 16<br>---   | INX H<br>1 5<br>---       | INR H<br>1 5<br>S Z A P -  | DCR H<br>1 5<br>S Z A P -  | MVI H,d8<br>2 7<br>---     | DAA<br>1 4<br>S Z A P C   |                           | DAD H<br>1 10<br>---      | LHLD a16<br>3 16<br>---   | DCX H<br>1 5<br>---       | INR L<br>1 5<br>S Z A P - | DCR L<br>1 5<br>S Z A P - | MVI L,d8<br>2 7<br>---     | CMA<br>1 4<br>---         |  |
| 3x |                           | LXI SP,d16<br>3 10<br>---    | STA a16<br>3 13<br>---    | INX SP<br>1 5<br>---      | INR M<br>1 10<br>S Z A P - | DCR M<br>1 10<br>S Z A P - | MVI M,d8<br>2 10<br>---    | STC<br>1 4<br>---         |                           | DAD SP<br>1 10<br>---     | LDA a16<br>3 13<br>---    | DCX SP<br>1 5<br>---      | INR A<br>1 5<br>S Z A P - | DCR A<br>1 5<br>S Z A P - | MVI A,d8<br>2 7<br>---     | CMC<br>1 4<br>---         |  |
| 4x | MOV B,B<br>1 5<br>---     | MOV B,C<br>1 5<br>---        | MOV B,D<br>1 5<br>---     | MOV B,E<br>1 5<br>---     | MOV B,H<br>1 5<br>---      | MOV B,L<br>1 5<br>---      | MOV B,M<br>1 7<br>---      | MOV B,A<br>1 5<br>---     | MOV C,B<br>1 5<br>---     | MOV C,C<br>1 5<br>---     | MOV C,D<br>1 5<br>---     | MOV C,E<br>1 5<br>---     | MOV C,H<br>1 5<br>---     | MOV C,L<br>1 5<br>---     | MOV C,M<br>1 7<br>---      | MOV C,A<br>1 5<br>---     |  |
| 5x | MOV D,B<br>1 5<br>---     | MOV D,C<br>1 5<br>---        | MOV D,D<br>1 5<br>---     | MOV D,E<br>1 5<br>---     | MOV D,H<br>1 5<br>---      | MOV D,L<br>1 5<br>---      | MOV D,M<br>1 7<br>---      | MOV D,A<br>1 5<br>---     | MOV E,B<br>1 5<br>---     | MOV E,C<br>1 5<br>---     | MOV E,D<br>1 5<br>---     | MOV E,E<br>1 5<br>---     | MOV E,H<br>1 5<br>---     | MOV E,L<br>1 5<br>---     | MOV E,M<br>1 7<br>---      | MOV E,A<br>1 5<br>---     |  |
| 6x | MOV H,B<br>1 5<br>---     | MOV H,C<br>1 5<br>---        | MOV H,D<br>1 5<br>---     | MOV H,E<br>1 5<br>---     | MOV H,H<br>1 5<br>---      | MOV H,L<br>1 5<br>---      | MOV H,M<br>1 7<br>---      | MOV H,A<br>1 5<br>---     | MOV L,B<br>1 5<br>---     | MOV L,C<br>1 5<br>---     | MOV L,D<br>1 5<br>---     | MOV L,E<br>1 5<br>---     | MOV L,H<br>1 5<br>---     | MOV L,L<br>1 5<br>---     | MOV L,M<br>1 7<br>---      | MOV L,A<br>1 5<br>---     |  |
| 7x | MOV M,B<br>1 7<br>---     | MOV M,C<br>1 7<br>---        | MOV M,D<br>1 7<br>---     | MOV M,E<br>1 7<br>---     | MOV M,H<br>1 7<br>---      | MOV M,L<br>1 7<br>---      | HLT<br>1 7<br>---          | MOV M,A<br>1 7<br>---     | MOV A,B<br>1 5<br>---     | MOV A,C<br>1 5<br>---     | MOV A,D<br>1 5<br>---     | MOV A,E<br>1 5<br>---     | MOV A,H<br>1 5<br>---     | MOV A,L<br>1 5<br>---     | MOV A,M<br>1 7<br>---      | MOV A,A<br>1 5<br>---     |  |
| 8x | ADD B<br>1 4<br>S Z A P C | ADD C<br>1 4<br>S Z A P C    | ADD D<br>1 4<br>S Z A P C | ADD E<br>1 4<br>S Z A P C | ADD H<br>1 4<br>S Z A P C  | ADD L<br>1 4<br>S Z A P C  | ADD M<br>1 7<br>S Z A P C  | ADD A<br>1 4<br>S Z A P C | ADC B<br>1 4<br>S Z A P C | ADC C<br>1 4<br>S Z A P C | ADC D<br>1 4<br>S Z A P C | ADC E<br>1 4<br>S Z A P C | ADC H<br>1 4<br>S Z A P C | ADC L<br>1 4<br>S Z A P C | ADC M<br>1 7<br>S Z A P C  | ADC A<br>1 4<br>S Z A P C |  |
| 9x | SUB B<br>1 4<br>S Z A P C | SUB C<br>1 4<br>S Z A P C    | SUB D<br>1 4<br>S Z A P C | SUB E<br>1 4<br>S Z A P C | SUB H<br>1 4<br>S Z A P C  | SUB L<br>1 4<br>S Z A P C  | SUB M<br>1 7<br>S Z A P C  | SUB A<br>1 4<br>S Z A P C | SBB B<br>1 4<br>S Z A P C | SBB C<br>1 4<br>S Z A P C | SBB D<br>1 4<br>S Z A P C | SBB E<br>1 4<br>S Z A P C | SBB H<br>1 4<br>S Z A P C | SBB L<br>1 4<br>S Z A P C | SBB M<br>1 7<br>S Z A P C  | SBB A<br>1 4<br>S Z A P C |  |
| Ax | ANA B<br>1 4<br>S Z A P C | ANA C<br>1 4<br>S Z A P C    | ANA D<br>1 4<br>S Z A P C | ANA E<br>1 4<br>S Z A P C | ANA H<br>1 4<br>S Z A P C  | ANA L<br>1 4<br>S Z A P C  | ANA M<br>1 7<br>S Z A P C  | ANA A<br>1 4<br>S Z A P C | XRA B<br>1 4<br>S Z A P C | XRA C<br>1 4<br>S Z A P C | XRA D<br>1 4<br>S Z A P C | XRA E<br>1 4<br>S Z A P C | XRA H<br>1 4<br>S Z A P C | XRA L<br>1 4<br>S Z A P C | XRA M<br>1 7<br>S Z A P C  | XRA A<br>1 4<br>S Z A P C |  |
| Bx | ORA B<br>1 4<br>S Z A P C | ORA C<br>1 4<br>S Z A P C    | ORA D<br>1 4<br>S Z A P C | ORA E<br>1 4<br>S Z A P C | ORA H<br>1 4<br>S Z A P C  | ORA L<br>1 4<br>S Z A P C  | ORA M<br>1 7<br>S Z A P C  | ORA A<br>1 4<br>S Z A P C | CMP B<br>1 4<br>S Z A P C | CMP C<br>1 4<br>S Z A P C | CMP D<br>1 4<br>S Z A P C | CMP E<br>1 4<br>S Z A P C | CMP H<br>1 4<br>S Z A P C | CMP L<br>1 4<br>S Z A P C | CMP M<br>1 7<br>S Z A P C  | CMP A<br>1 4<br>S Z A P C |  |
| Cx | RNZ<br>1 11/5<br>---      | POP B<br>1 10<br>---         | JNZ a16<br>3 10<br>---    | JMP a16<br>3 10<br>---    | CNZ a16<br>3 17/11<br>---  | PUSH B<br>1 11<br>---      | ADI d8<br>2 7<br>S Z A P C | RST 0<br>1 11<br>---      | RZ<br>1 11/5<br>---       | RET<br>1 10<br>---        | JZ a16<br>3 10<br>---     |                           | CZ a16<br>3 17/11<br>---  | CALL a16<br>3 17<br>---   | ACI d8<br>2 7<br>S Z A P C | RST 1<br>1 11<br>---      |  |
| Dx | RNC<br>1 11/5<br>---      | POP D<br>1 10<br>---         | JNC a16<br>3 10<br>---    | OUT d8<br>2 10<br>---     | CNC a16<br>3 17/11<br>---  | PUSH D<br>1 11<br>---      | SUI d8<br>2 7<br>S Z A P C | RST 2<br>1 11<br>---      | RC<br>1 11/5<br>---       |                           | JC a16<br>3 10<br>---     | IN d8<br>2 10<br>---      | CC a16<br>3 17/11<br>---  |                           | SBI d8<br>2 7<br>S Z A P C | RST 3<br>1 11<br>---      |  |
| Ex | RPO<br>1 11/5<br>---      | POP H<br>1 10<br>---         | JPO a16<br>3 10<br>---    | XTHL<br>1 18<br>---       | CPO a16<br>3 17/11<br>---  | PUSH H<br>1 11<br>---      | ANI d8<br>2 7<br>S Z A P C | RST 4<br>1 11<br>---      | RPE<br>1 11/5<br>---      | PCHL<br>1 5<br>---        | JPE a16<br>3 10<br>---    | XCHG<br>1 5<br>---        | CPE a16<br>3 17/11<br>--- |                           | XRI d8<br>2 7<br>S Z A P C | RST 5<br>1 11<br>---      |  |
| Fx | RP<br>1 11/5<br>---       | POP PSW<br>1 10<br>S Z A P C | JP a16<br>3 10<br>---     | DI<br>1 4<br>---          | CP a16<br>3 17/11<br>---   | PUSH PSW<br>1 11<br>---    | ORI d8<br>2 7<br>S Z A P C | RST 6<br>1 11<br>---      | RM<br>1 11/5<br>---       | SPHL<br>1 5<br>---        | JM a16<br>3 10<br>---     | EI<br>1 4<br>---          | CM a16<br>3 17/11<br>---  |                           | CPI d8<br>2 7<br>S Z A P C | RST 7<br>1 11<br>---      |  |

# Segmentace paměti 8086

- Čtyři registry: CS, DS, SS, ES



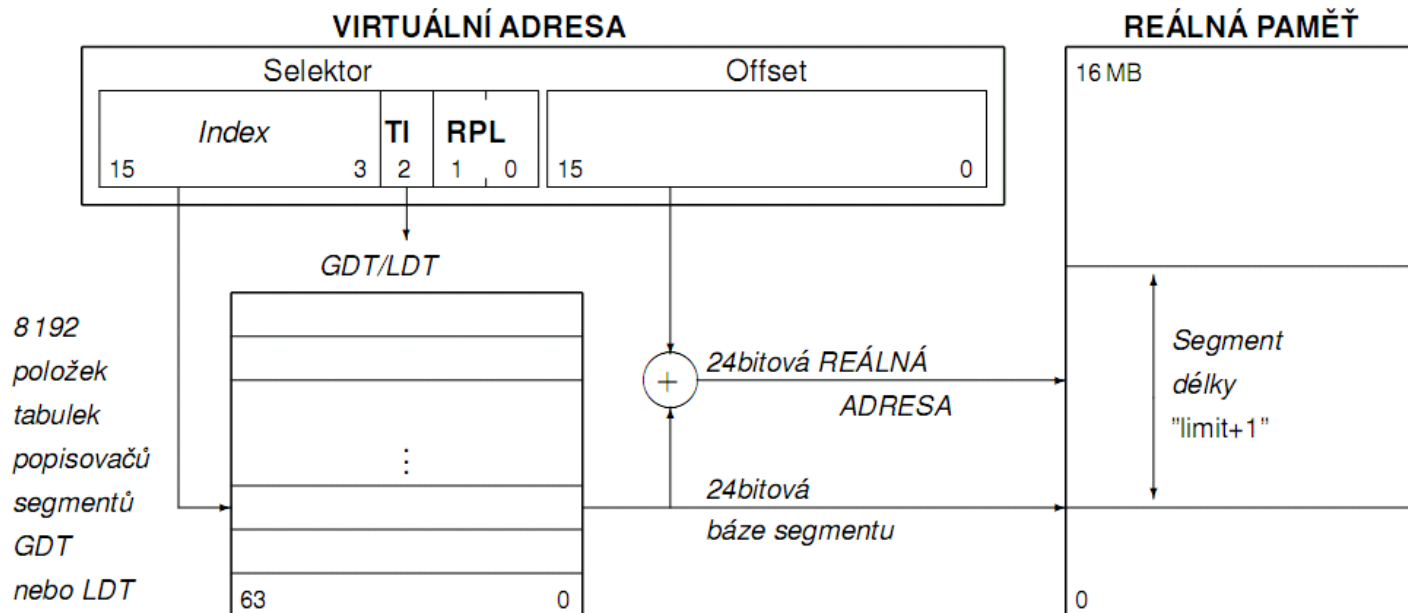
- Adresa ve tvaru segment:offset



# 80286 (základ PC AT)

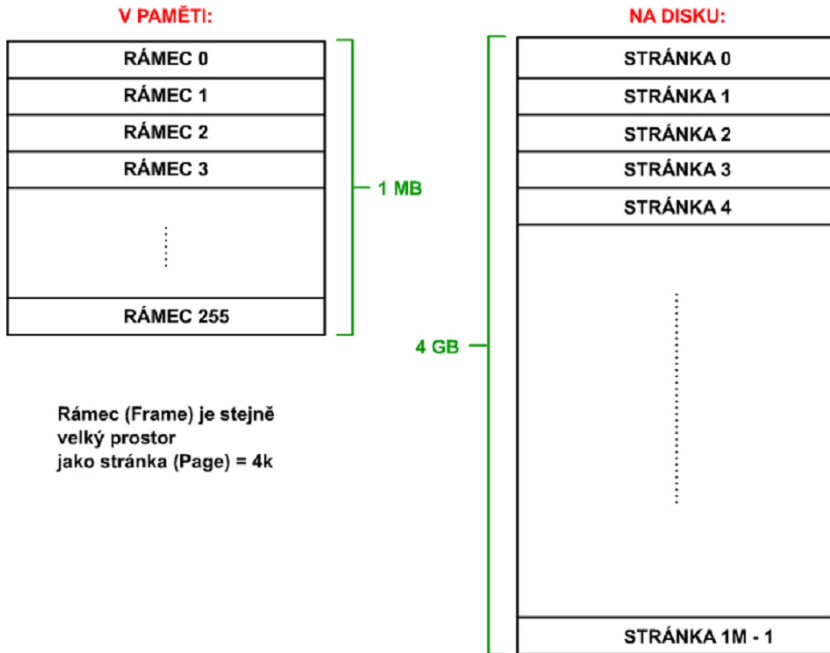
Dva režimy

- Reálný (jako u 8086)
- Chráněný režim (podpora multiprogramování, ochrana paměti, jiná adresace, přístupová práva)
- Globální/lokální adresový prostor

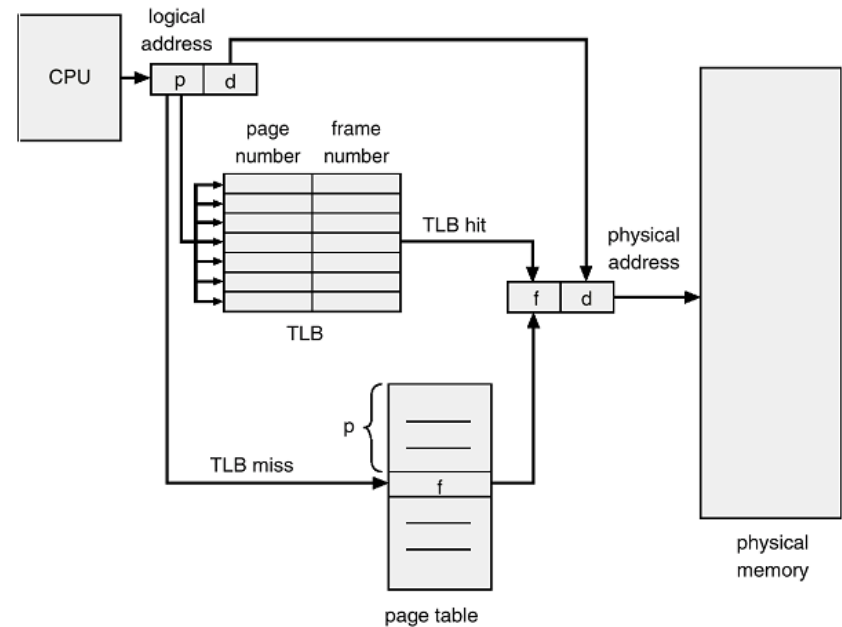




# Virtualizace paměti



Rámeč (Frame) je stejně velký prostor jako stránka (Page) = 4k

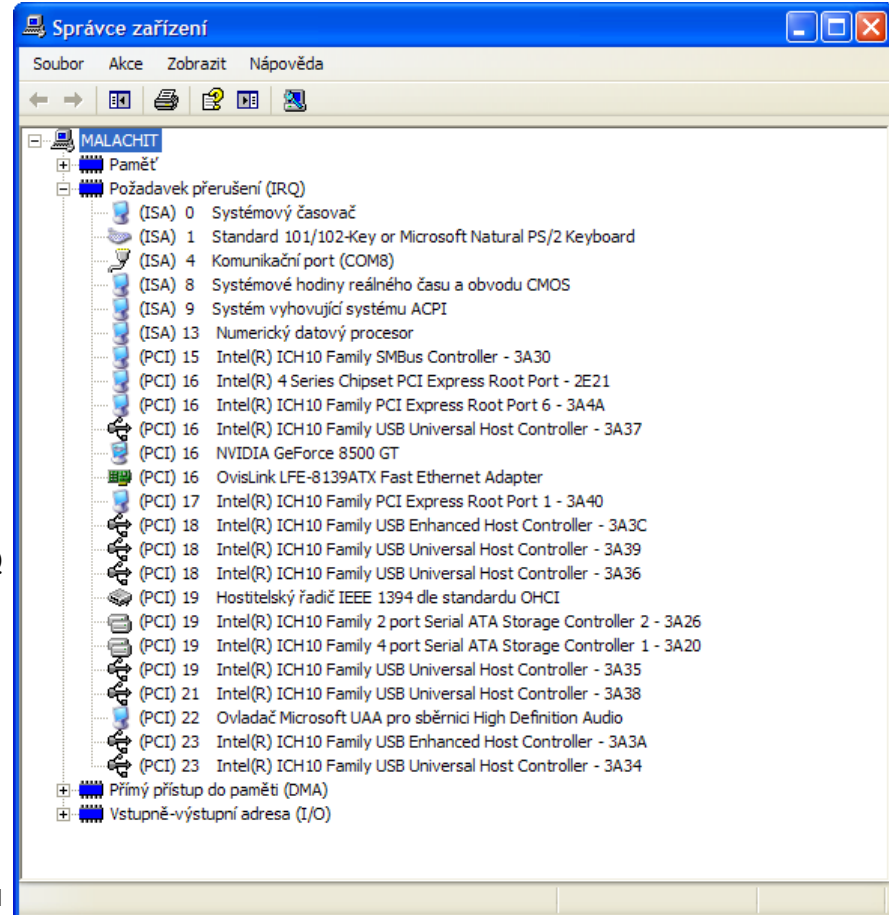


# Přerušení (IRQ = Interrupt Request)

- je potřeba reagovat na hardwarové události, spouštět služby OS či obsluhovat více procesů
- **zdroje přerušení (x86):**
  - vnější
    - signál NMI – katastrofické stavy (např. chyba parity v paměti)
    - signál INTR (interrupt) maskovatelný signál, vnější zdroje IRQ přes řadič přerušení
  - vnitřní
    - softwarové generování přerušení – instrukce INT
    - dělení nulou
- **průběh:**
  1. Přerušení prováděného programu
  2. Úklid kontextu do zásobníku (registry)
  3. Provedení rutiny
  4. Obnovení kontextu, pokračování
- **Ne vždy lze přerušení provést**
  - povolení/zakázání přerušení – instrukce STI, CLI, nastavují příznak IF (Interrupt Flag)
  - Kritická sekce

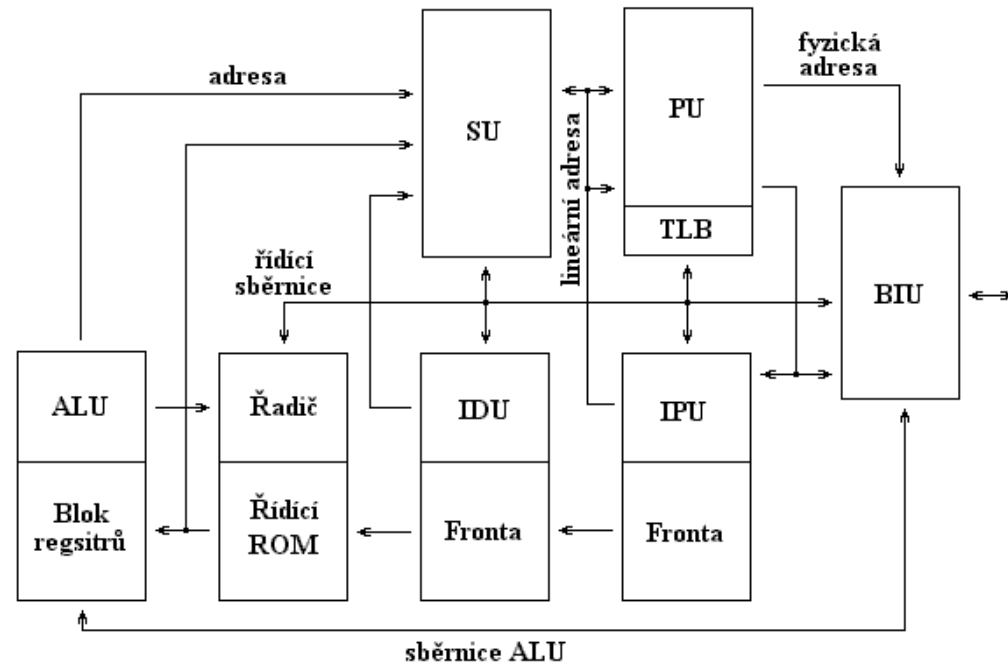
# Přerušeni

|        |                     |                                       |
|--------|---------------------|---------------------------------------|
| Linka  | PC/XT               | PC/AT                                 |
| NMI    | Chyba parity        | Chyba parity                          |
| IRQ 0  | Systémový časovač   | Systémový časovač                     |
| IRQ 1  | Klávesnice          | Klávesnice                            |
| IRQ 2  | Grafický adaptér    | Propojení řadičů přerušeni            |
| IRQ 3  | Sériový port 2      | Sériový port 2                        |
| IRQ 4  | Sériový port 1      | Sériový port 1                        |
| IRQ 5  | Řadič pevného disku | Paralelní port 2                      |
| IRQ 6  | Řadič disket        | Řadič disket                          |
| IRQ 8  | -----               | Hodiny reálného času                  |
| IRQ 9  | -----               | Volný, softwarově přeměrován na IRQ 2 |
| IRQ 10 | -----               | Volný                                 |
| IRQ 11 | -----               | Volný                                 |
| IRQ 12 | -----               | Volný                                 |
| IRQ 13 | -----               | Numerický koprocesor                  |
| IRQ 14 | -----               | Řadič pevného disku                   |
| IRQ 15 | -----               | Volný                                 |



# Mikroprocesor 80386

- **BIU** (Bus Interface Unit) – styk se sběrnici, pro přenos dat mezi procesorem a okolím
- **IPU** (Instruction Prefetch Unit – naplňuje frontu instrukcí pro IDU po vybrání instrukce z fronty žádá přes BIU její doplnění z adresy v paměti
- **IDU** (Instruction Decode Unit) – dekóduje instrukci, zjišťuje její délku a převádí do fronty ve formátu pro ALU
- **ALU** (Arithmetic - Logic Unit), provádí aritmetické a logické operace, přes BIU zapisuje na výstup

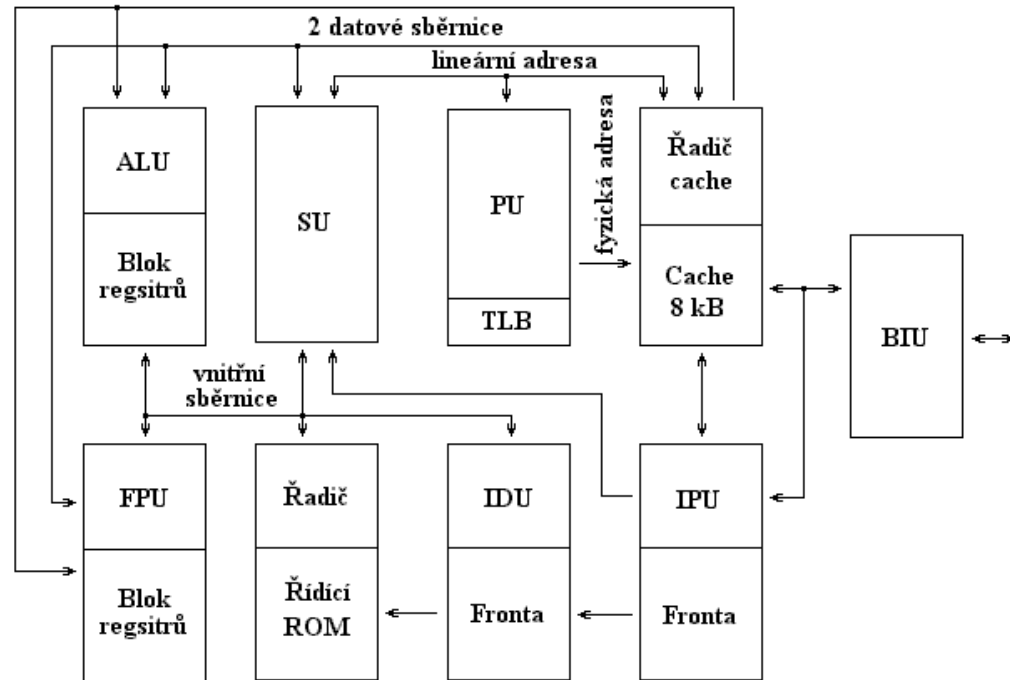


- **SU** (Segmentation Unit) – zajišťuje segmentaci paměti
- **PU** (Paging Unit) – jednotka pro stránkování paměti
- **TLB** (Translation Lookaside Buffer)



# Mikroprocesor 80486

- Numerický koprocesor na čipu (FPU)
- Interní cache paměť 8kB
- (externí už u 386)
- Skalární procesor (má jednu frontu pro zřetězené provádění instrukcí - pipelining)



## Pipelining

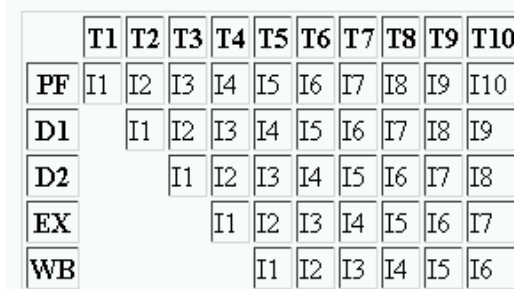
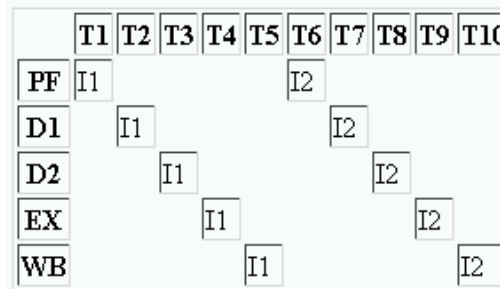
**PF** (Prefetch)

**D1** (Decode<sub>1</sub>) (délka a typ)

**D2** (Decode<sub>2</sub>) (výpočet adresy)

**EX** (Execution)

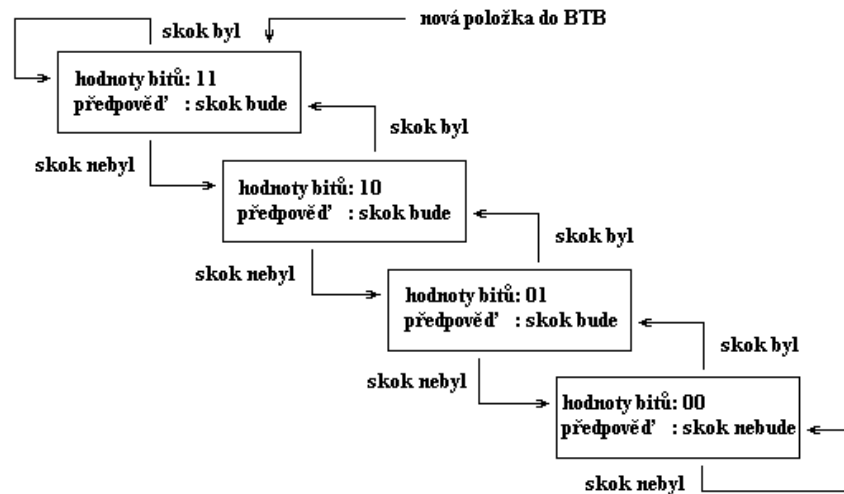
**WB** (Write Back)



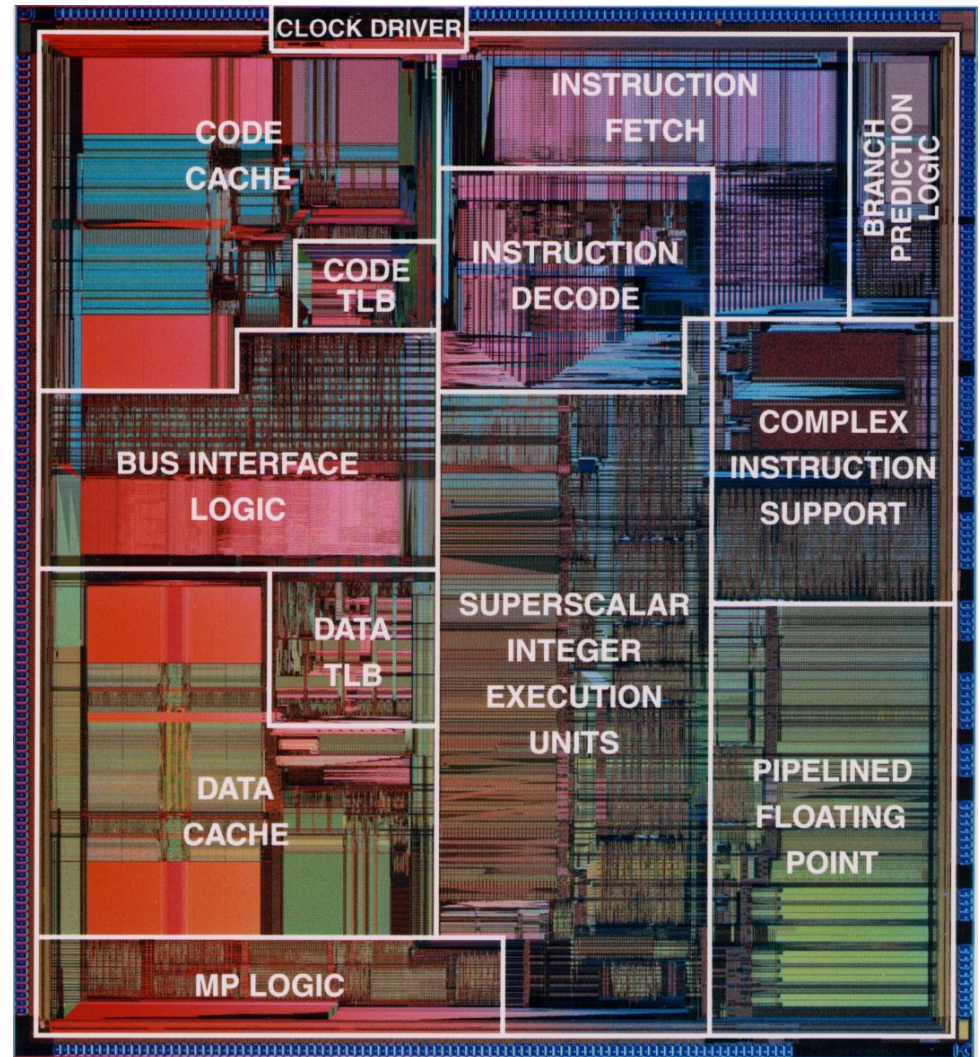
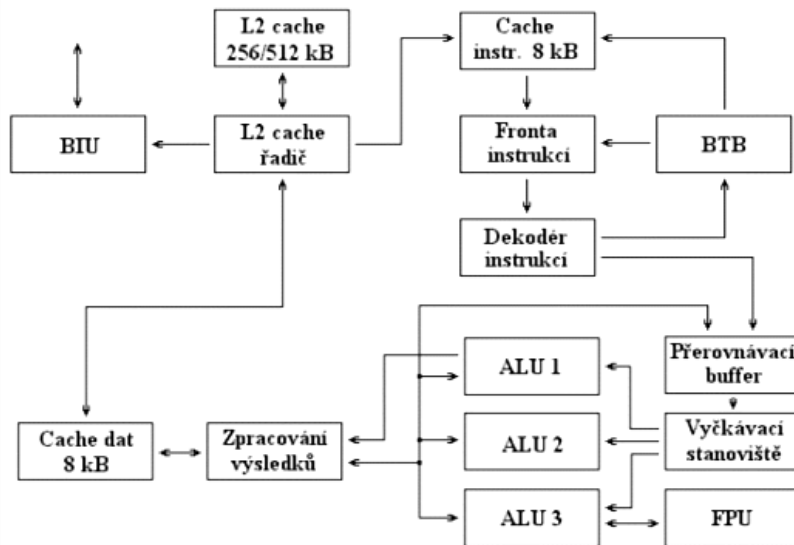
# Mikroprocesor Pentium

- Superskalární architektura – dvě fronty U, V
- BTB
- rozdělení cache paměti na data + instrukce

- **Pentium Pro**
- Spekulativní provádění
- Přejmenování registrů
- Mikrooperace
- Integrace cache paměti do pouzdra (není na čipu)



# Mikroprocesor Pentium Pro



# Pentium 4, Pentium D...

Hyper-threading technology

- paralelismus na úrovni vláknů s efektivním využitím zdrojů
- dynamické přizpůsobování frekvence a napájecího napětí

Intel Virtualization Technology

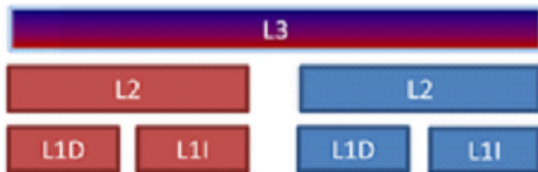
# Pentium M, Core Duo...



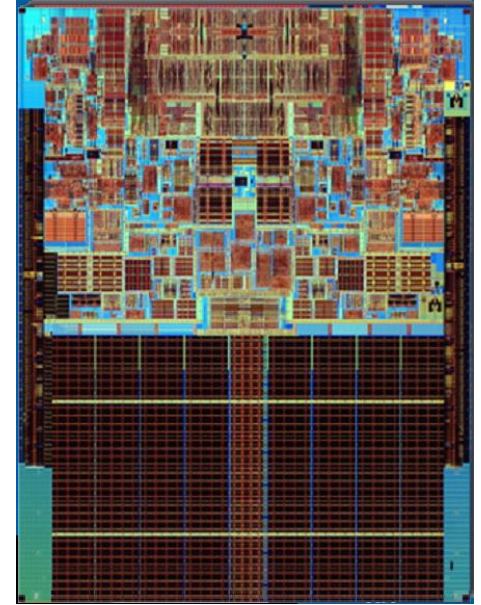
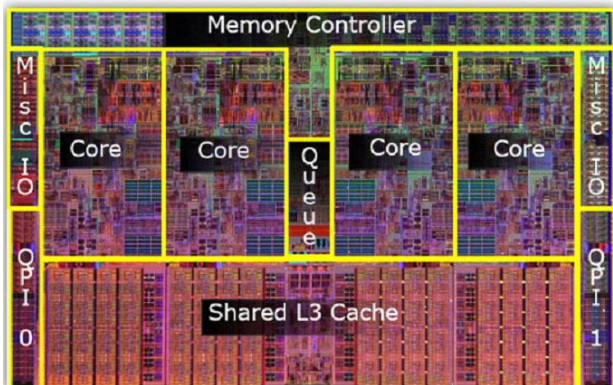
# A další...

## Core i7

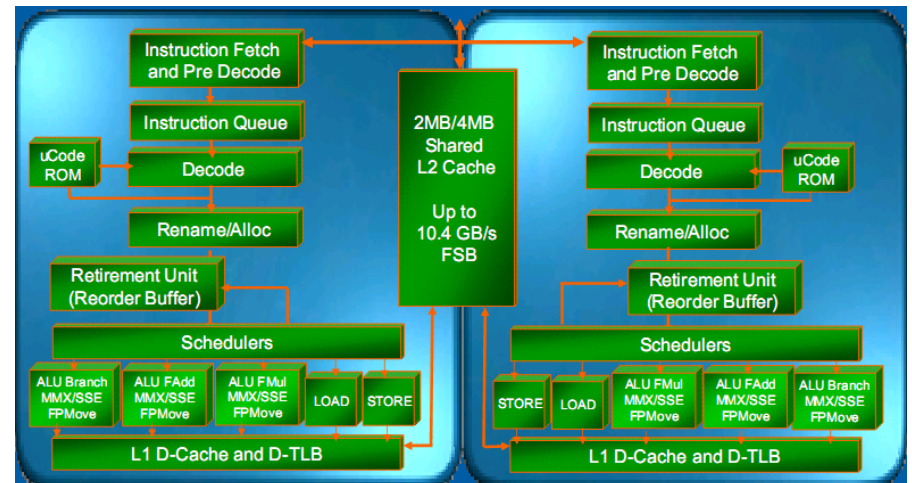
- 45nm, všechna jádra na jedné křemíkové desce
- integrovaný řadič DDR3
- Quick Path Interface místo FSB – pro graf. karty
- 1,65 V
- Turbo Mode, možnost vypnutí nepoužívaných jader



Nehalem



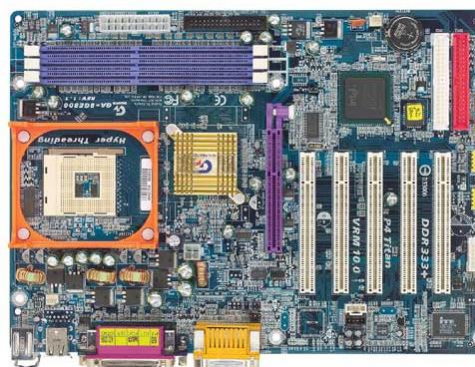
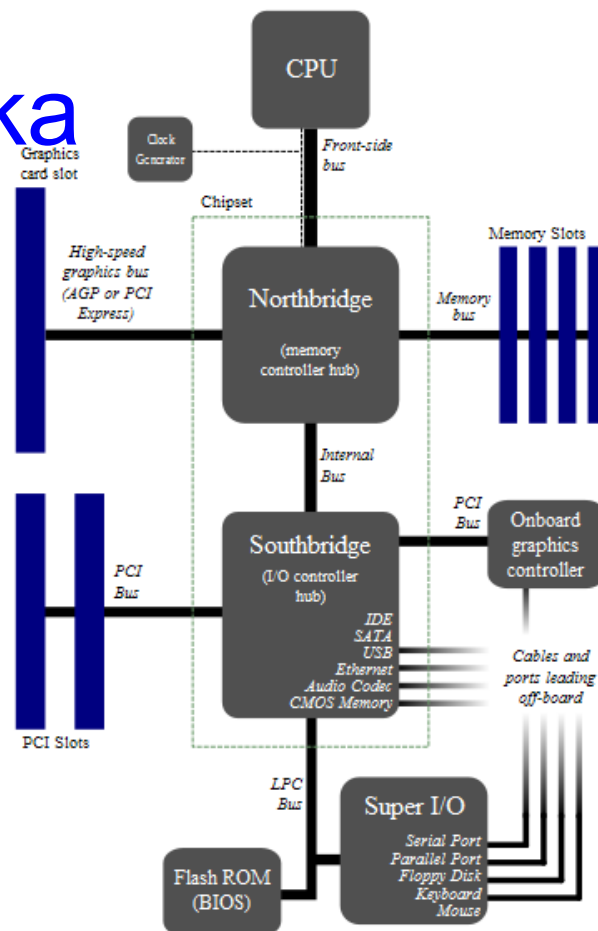
Core 2 Duo



# Základní deska

## Základní deska

- Mikroprocesor(y)
- „Vnitřní“ paměť RAM (příp. cache paměť)
- Obvody čipové sady
- Rozšiřující sběrnice (ISA, PCI, AGP, PCI Express)
- Sloty pro připojení rozšiřujících karet
- CMOS paměť
- Hodiny reálného času
- Akumulátor zálohující CMOS paměť
- Vstup / výstupní rozhraní LPT, RS-232, USB,
- Klávesnice, myš
- Rozhraní pevných disků (EIDE, SATA)
- Integrace grafické karty, zvukové karty, síťové karty, WIFI, infraportu



# Sběrnice

## sběrnice v PC

- PC Bus (8bit)
- ISA (16bit)
- VL-Bus (32bit)
- PCI (32/64bit)
- PCI-Express (sériová)

## Plug and Play – úkoly:

- identifikace desky
- zjištění, které prostředky (IRQ, DMA, V/V, RAM a ROM adresy) deska potřebuje
- zjištění, které prostředky deska může použít
- nastavení prostředků tak, aby nedošlo ke kolizi
- vyhledání a konfigurace ovladače

## Piny sběrnice PC-Bus

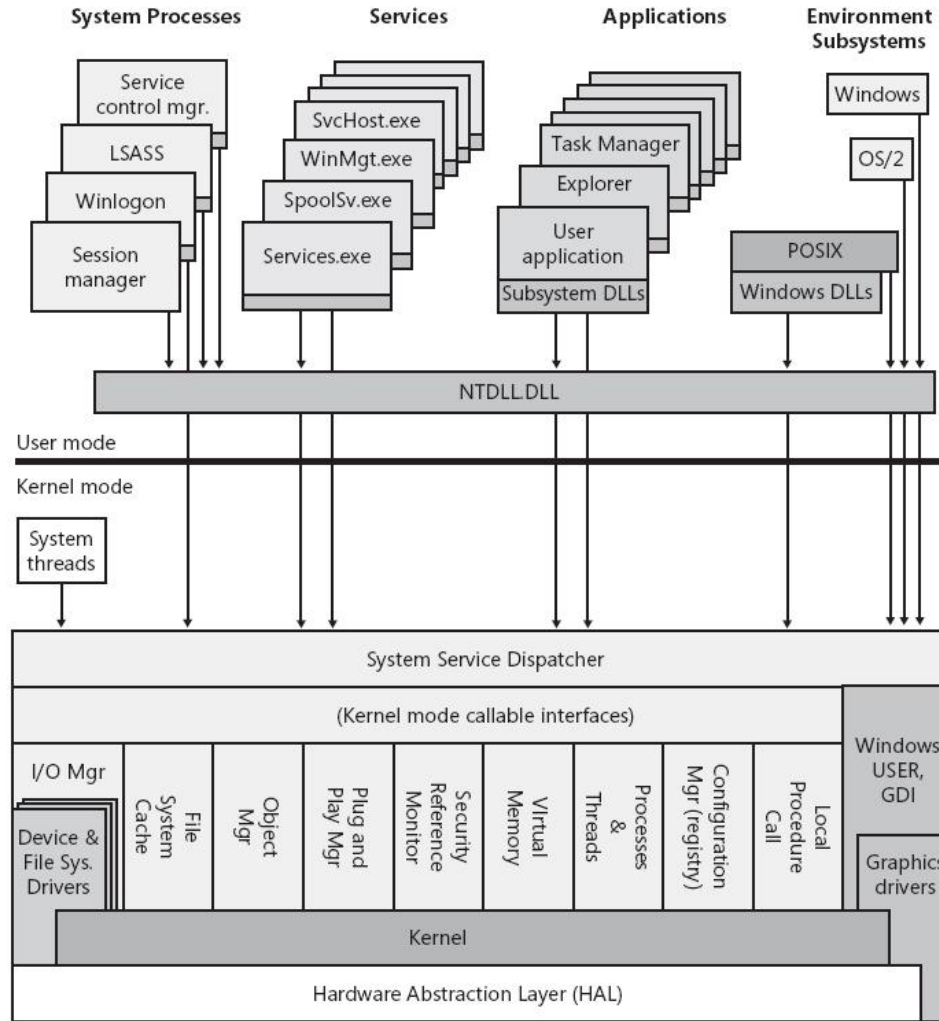
- datové (8)
- adresové (20)
- napájení, zem
- čtení x zápis, paměť x V/V z.
- řídicí signály
- IRQ
- DMA

# Operační systém

OS je program.

- Řízení provádění (uživatelských) programů
  - správa procesů a vláken, přidělování času procesoru
  - časovač (preemptivní x kooperativní multitasking)  
časování časově-kritických aplikací
- Správa systémových prostředků a jejich přidělování
  - správa paměti
    - přidělování, ochrana,
  - správa IO zařízení
    - zpracování přerušení

# Operační systém Windows (NT→)



Hardware interfaces (buses, I/O devices, interrupts, interval timers, DMA, memory cache control, etc.)

# Operační systém – ovladače

*User-mode drivers* – hlavně rozhraní mezi Win32 aplikací a kernel-mode drivery nebo komponentami OS (př. Vista – drivery tiskáren)

*Kernel-mode drivers* – provádí se v kernel módu jako část *Executive*. Jsou vrstvené.

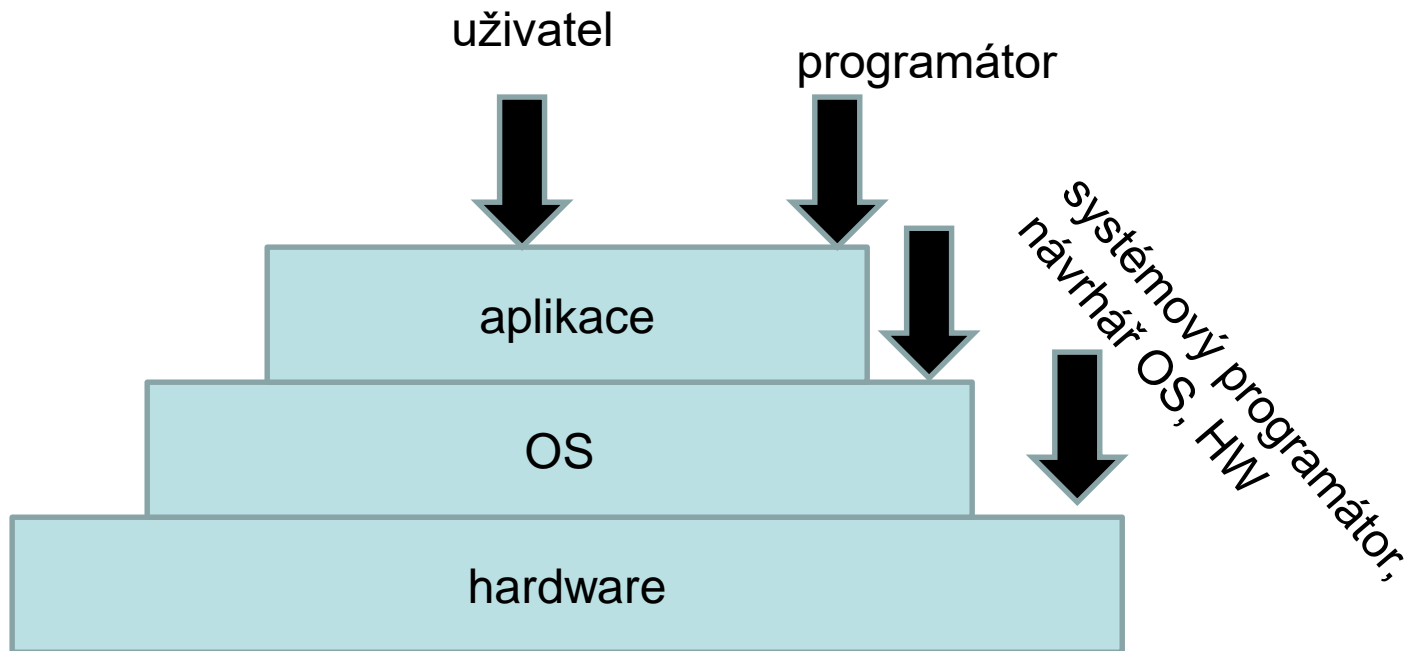
Vývoj ovladačů vyžaduje specializované znalosti.

Certifikace ovladačů, 64bitové systémy

Např. Windows Driver Kit (knihovny, nástroje, nápověda)



# Operační systém - úrovně



# Odkazy

## **Předměty na MU**

- **FI PB151 Výpočetní systémy**
- **FI PB152 Operační systémy**
- **FI PV094 Technické vybavení počítačů**