

Architektura počítačů

Úvod

http://d3s.mff.cuni.cz/teaching/computer_architecture/



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

Lubomír Bulej

bulej@d3s.mff.cuni.cz

Proč jsou počítače zajímavé?



- **Velmi dynamický obor**

- První digitální elektronické počítače kolem roku 1940
- O 60 let později počítače zcela běžné, dnes všudypřítomné
- Nové technologie jsou nahrazovány dříve, než stačí zestárnout

- **Neuvěřitelný vliv na každodenní život**

- Internet, vestavěné počítače, lidský genom, výpočetní chemie, ...
- Každé řádové snížení cen, zvýšení výkonu, zmenšení velikosti přináší nové možnosti



Co je vlastně počítač?



- **V dnešní době velmi široký pojem**

- Řada společných technologií, ale různé uspořádání podle požadavků a využití

- **Hlavní třídy**

- **Osobní počítače**

- Optimální poměr cena/výkon (tlak na vývoj)

- **Servery, mainframy, superpočítače**

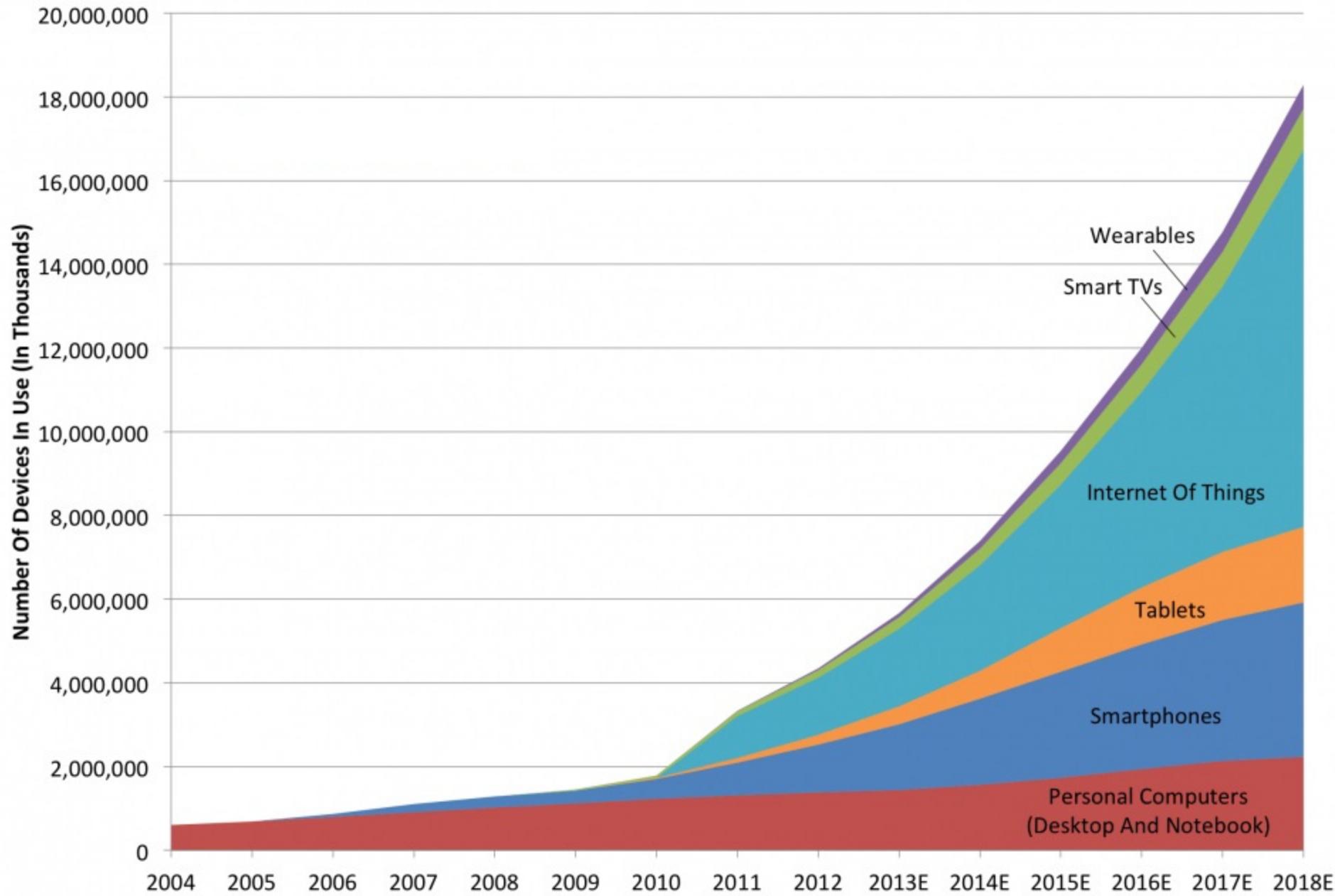
- Vyšší propustnost, spolehlivost, výpočetní výkon
 - Vědecké výpočty, velké množství požadavků

- **Vestavěné (embedded) počítače**

- Dnes nejrychleji rostoucí trh (nejen mobilní zařízení)
 - Omezené zdroje (paměť, výkon, energie, cena), speciální nároky (fyzická odolnost)



Global Internet Device Installed Base Forecast



Sálový počítač (1964)

• IBM System 360

- Použití integrovaných obvodů
- Revoluční prvky
 - Stavebnicová konstrukce
 - Jednotná struktura dat a instrukcí
 - Jednotný způsob připojování periferií
 - Ochrana dat v paměti
- Prvky architektury zachovány v mainframech až dodnes



[1]



Sálový počítač (2005)



- IBM System Z9-109 model S54

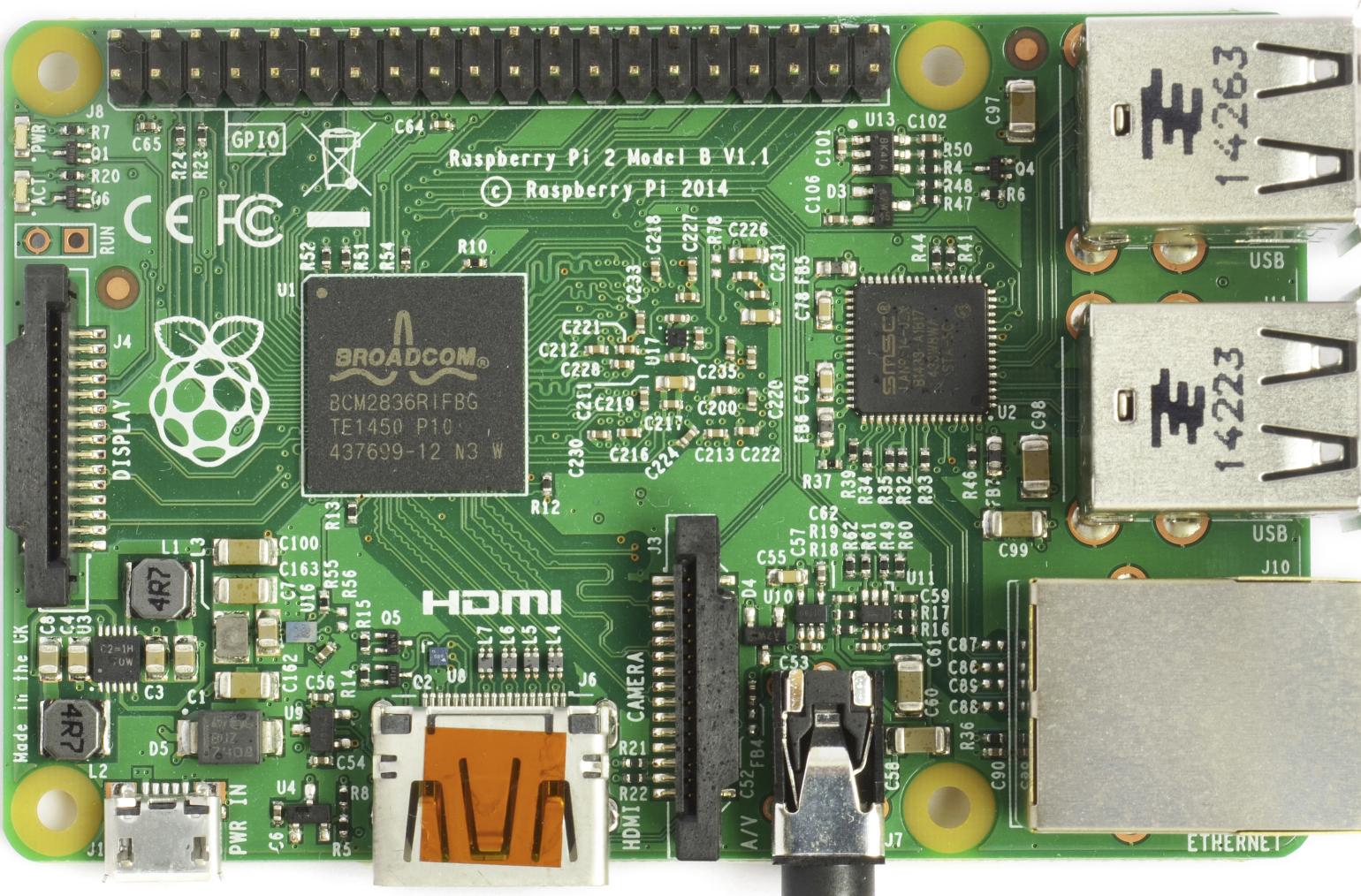
- 60 konfigurovatelných LPARS
- *Special-purpose processors*
- Paměť 512 GB
- 1 740 kg, 2,49 m², příkon 18,3 kW
- Dostupnost, propustnost, bezpečnost



[2]



Méně běžný osobní počítač



[3]



Běžný osobní počítač

• • •



[4]



Co je v krabici běžného počítače?



Základní deska

Procesor

Paměť (RAM, ROM)

Čipová sada

Základní I/O periferie

[4]



Co je v krabici běžného počítače?



Základní deska

Procesor

Paměť (RAM, ROM)

Čipová sada

Základní I/O periferie

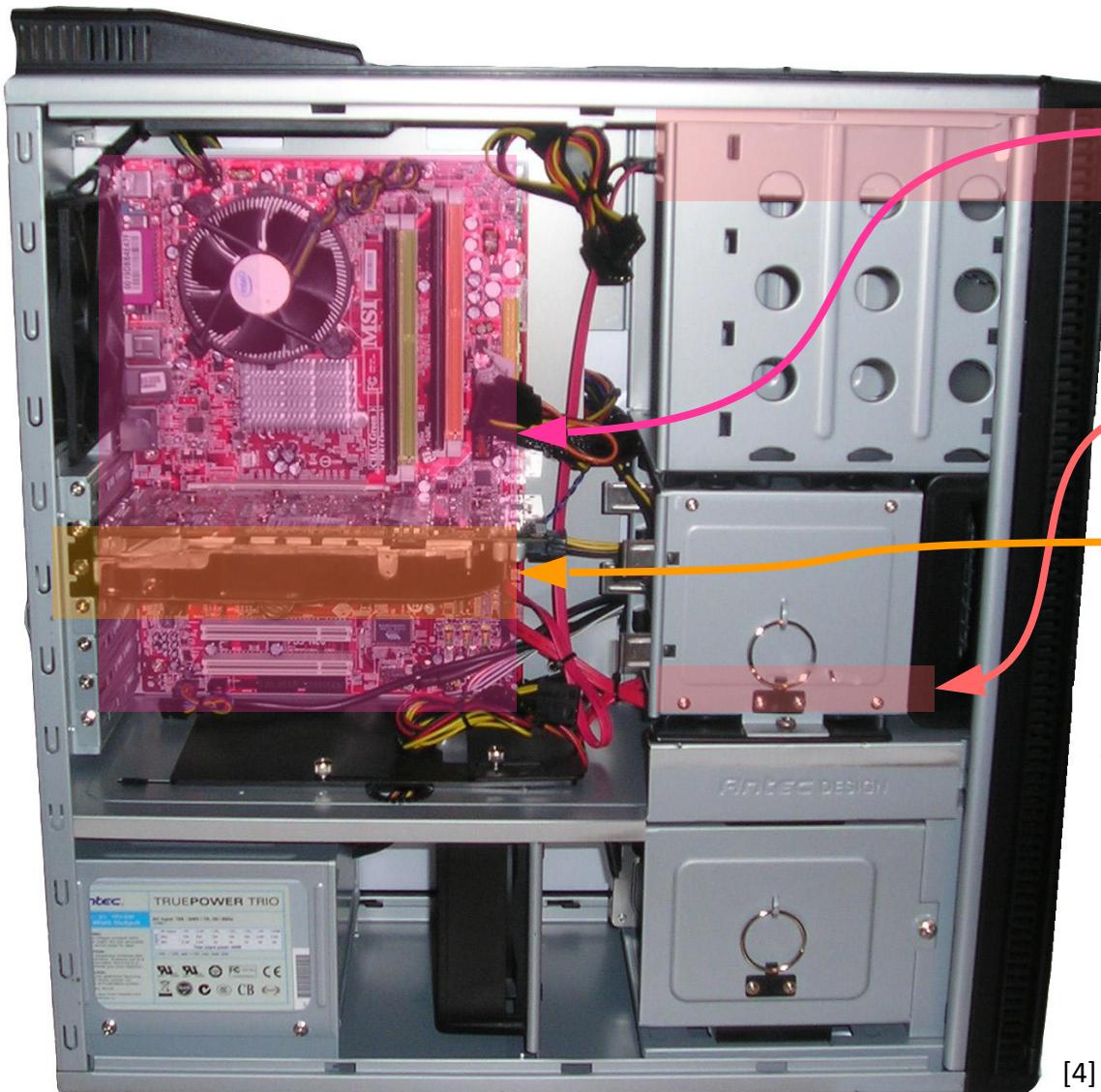
Optická mechanika

Pevný disk

[4]



Co je v krabici běžného počítače?



Základní deska

Procesor

Paměť (RAM, ROM)

Čipová sada

Základní I/O periferie

Optická mechanika

Pevný disk

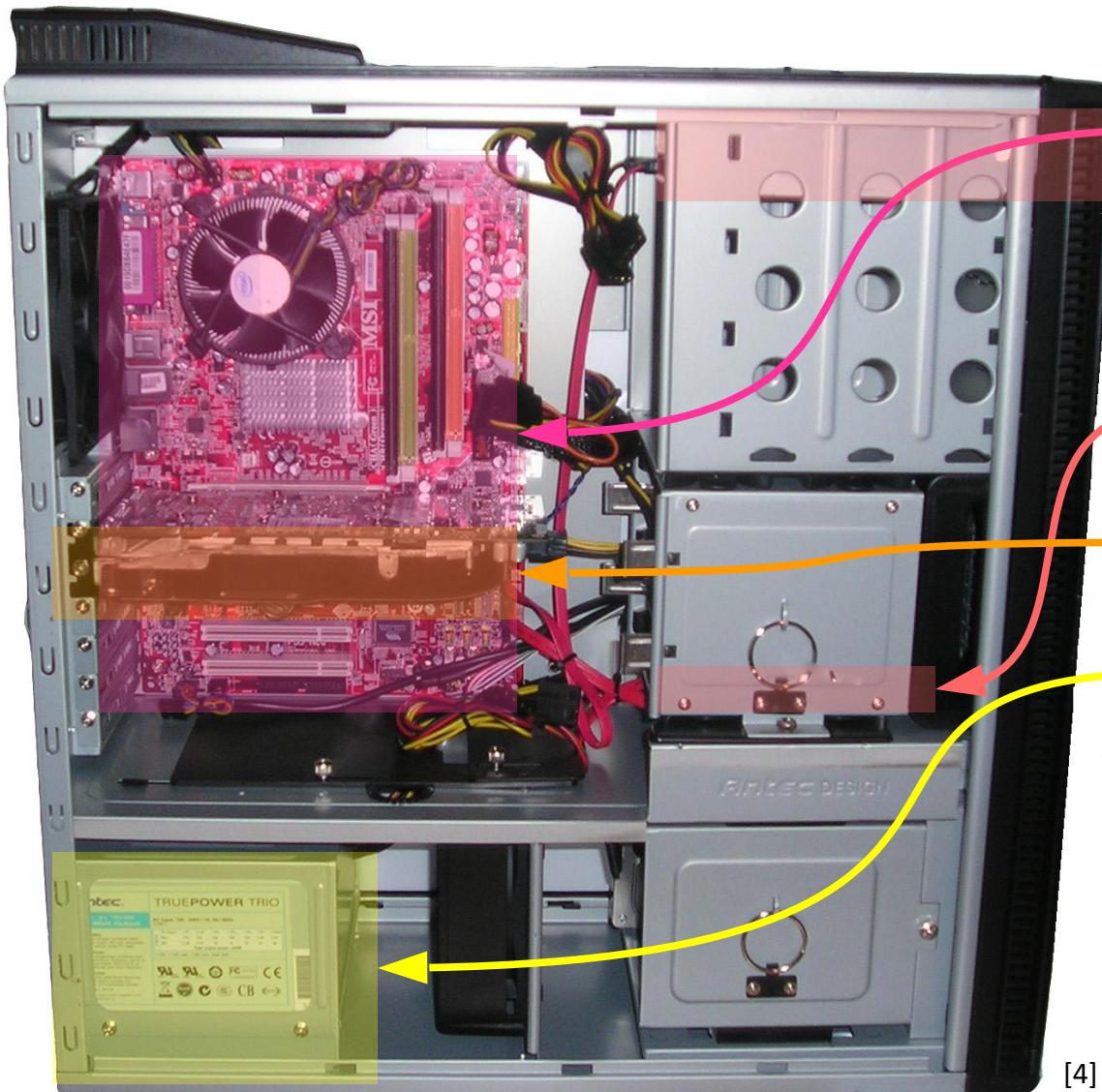
Rozšiřující karty

Grafická karta

[4]



Co je v krabici běžného počítače?



Základní deska

Procesor

Paměť (RAM, ROM)

Čipová sada

Základní I/O periferie

Optická mechanika

Pevný disk

Rozšiřující karty

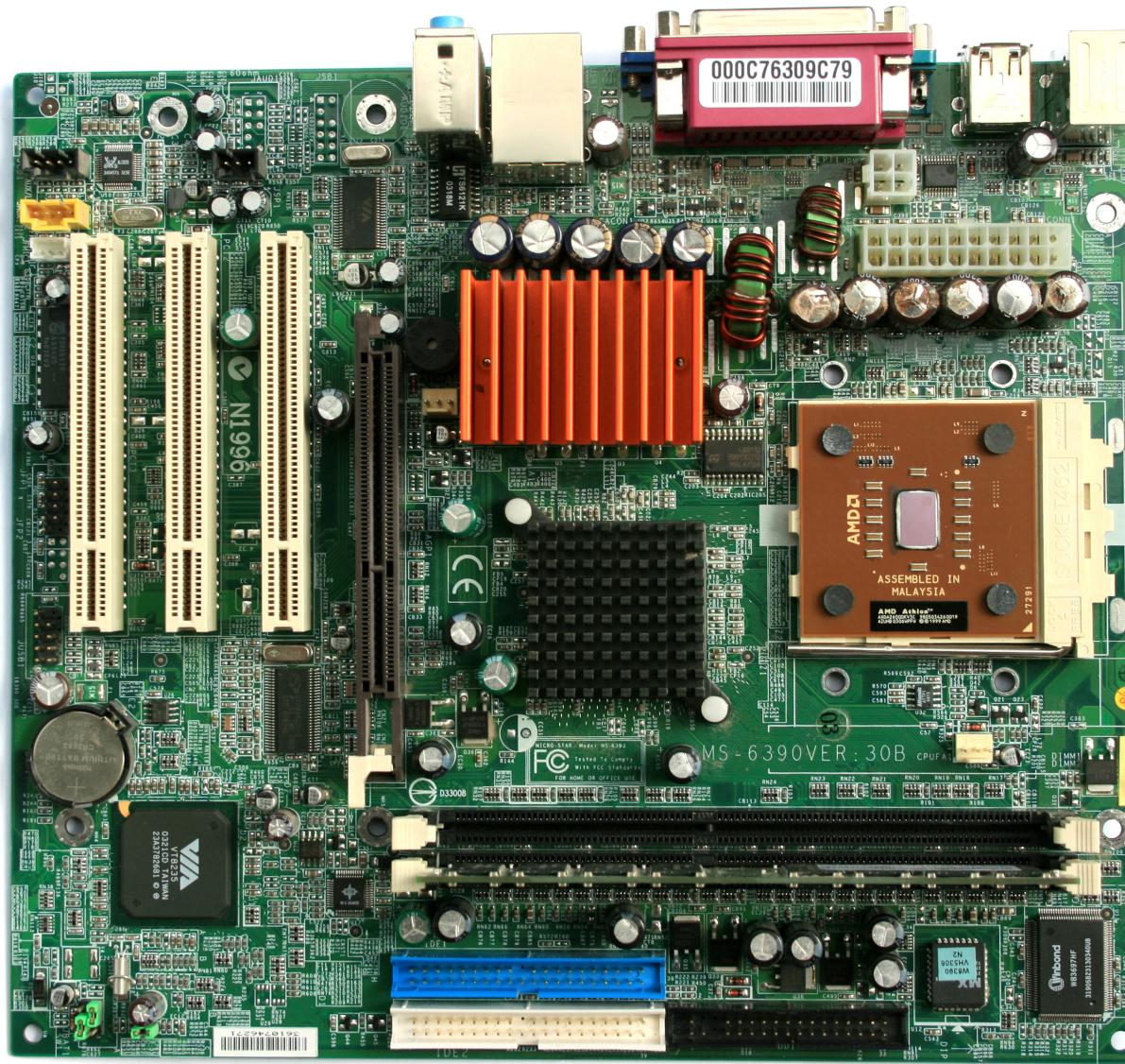
Grafická karta

Zdroj

[4]



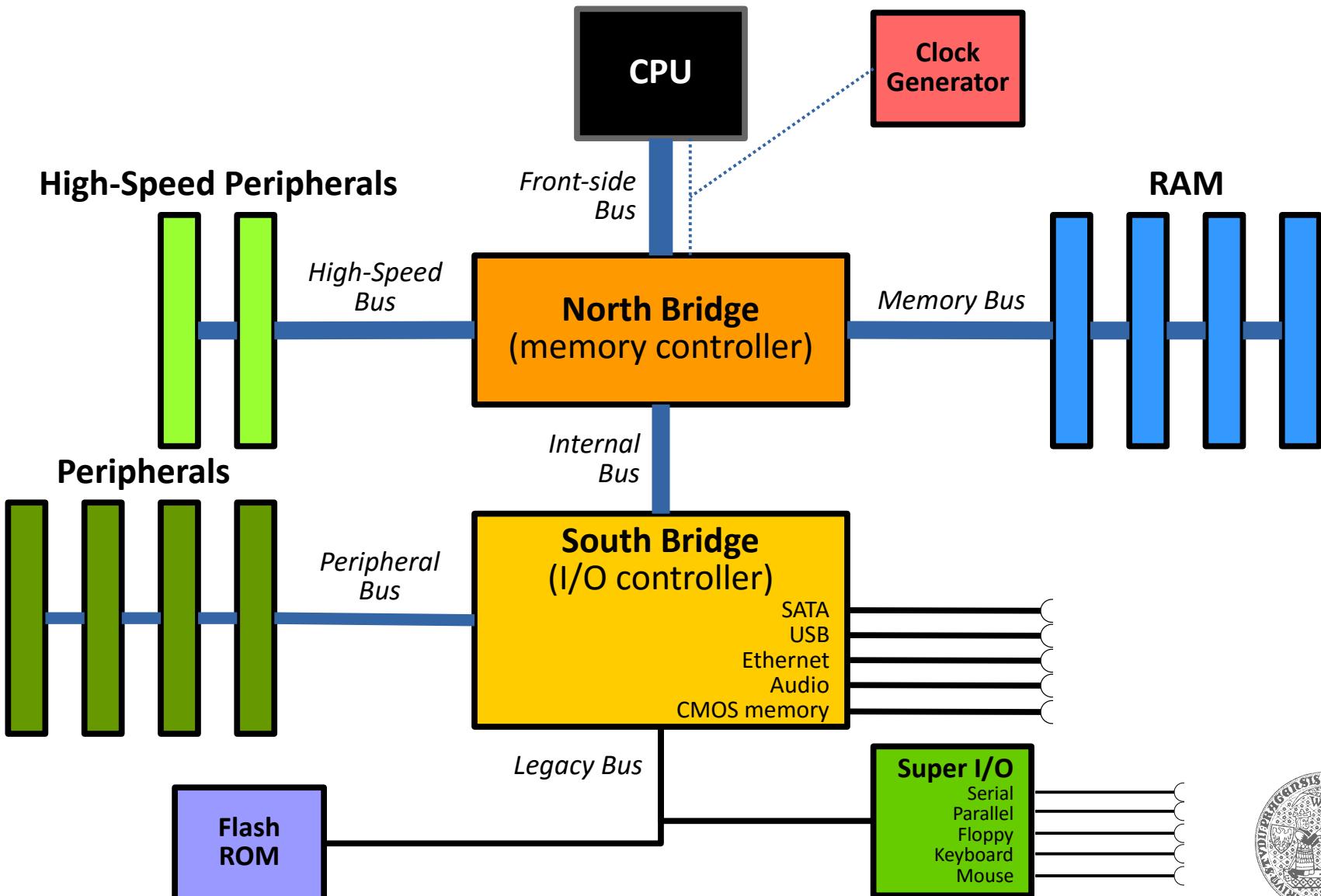
Základní deska



[5]



Základní deska (2)



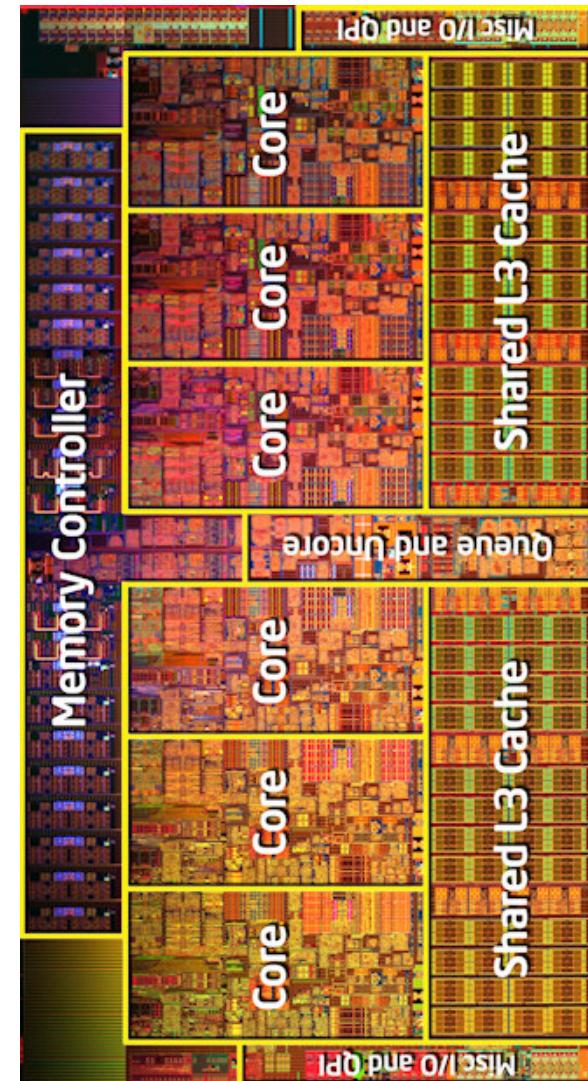
Procesor

• Součásti

- Datová cesta
(operace s daty)
- Řízení
(řadič datové cesty)
- Paměťové prvky
(soubor registrů a cache)

• Intel Core i7-980X

- 6 jader, 12 MB L3 cache,
taktovací frekvence 3,33 GHz
- 32 nm výrobní proces,
248 mm², 1,2 mld. tranzistorů

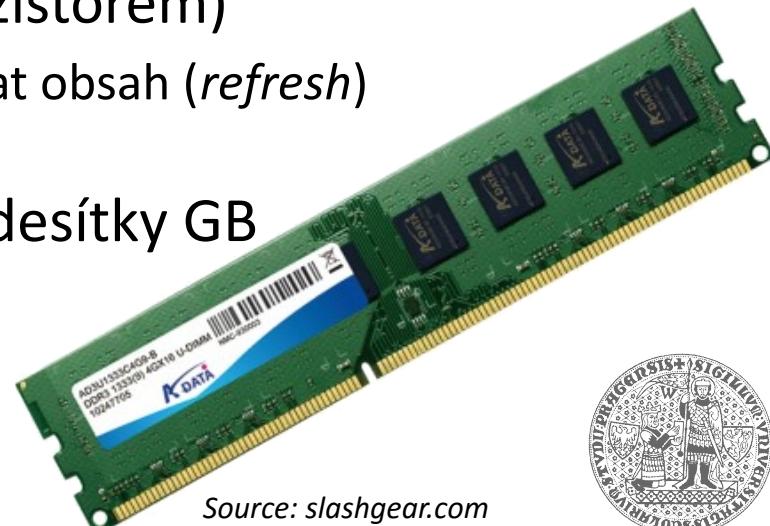


Source: intel.com



• Dočasná paměť

- Běžící programy a data, přímá adresace procesorem
- *Dynamic Random-Access Memory (DRAM)*
 - Konstantní doba přístupu (v řádu desítek ns)
 - Bity uloženy jako náboje v integrovaných kondenzátorech (spolu s tranzistorem)
 - Nutnost periodicky občerstovovat obsah (*refresh*)
 - Typická frekvence 16 Hz
 - Typická kapacita jednotky až desítky GB



● Dočasná paměť

■ *Static Random-Access Memory (SRAM)*

- Realizovaná pomocí dvoustavových klopných obvodů (obvykle 4 až 6 tranzistorů na bit)
 - Není potřeba periodické občerstvování
 - Výrazně vyšší rychlosť (jednotky ns), ale výrazně nižší hustota záznamu a vyšší cena
 - Při plné rychlosti výrazně vyšší spotřeba
- Cache procesoru
- Soubor registrů procesoru
- Interní paměť procesoru



Technologie procesorů a pamětí



● Polovodičový tranzistor

- Základní stavební prvek
 - Typicky použit jako diskrétní elektronický prvek (signálem řízený spínač), ne analogový prvek (zesilovač)

● Integrovaný obvod

- Kombinace více tranzistorů na jednom polovodičovém čipu
 - Včetně doplňkových elektronických součástek (kondenzátorů, rezistorů apod.)
- Lepší výrobní technologie → menší rozměry → vyšší stupeň integrace → vyšší rychlosť procesorů a vyšší kapacita pamětí

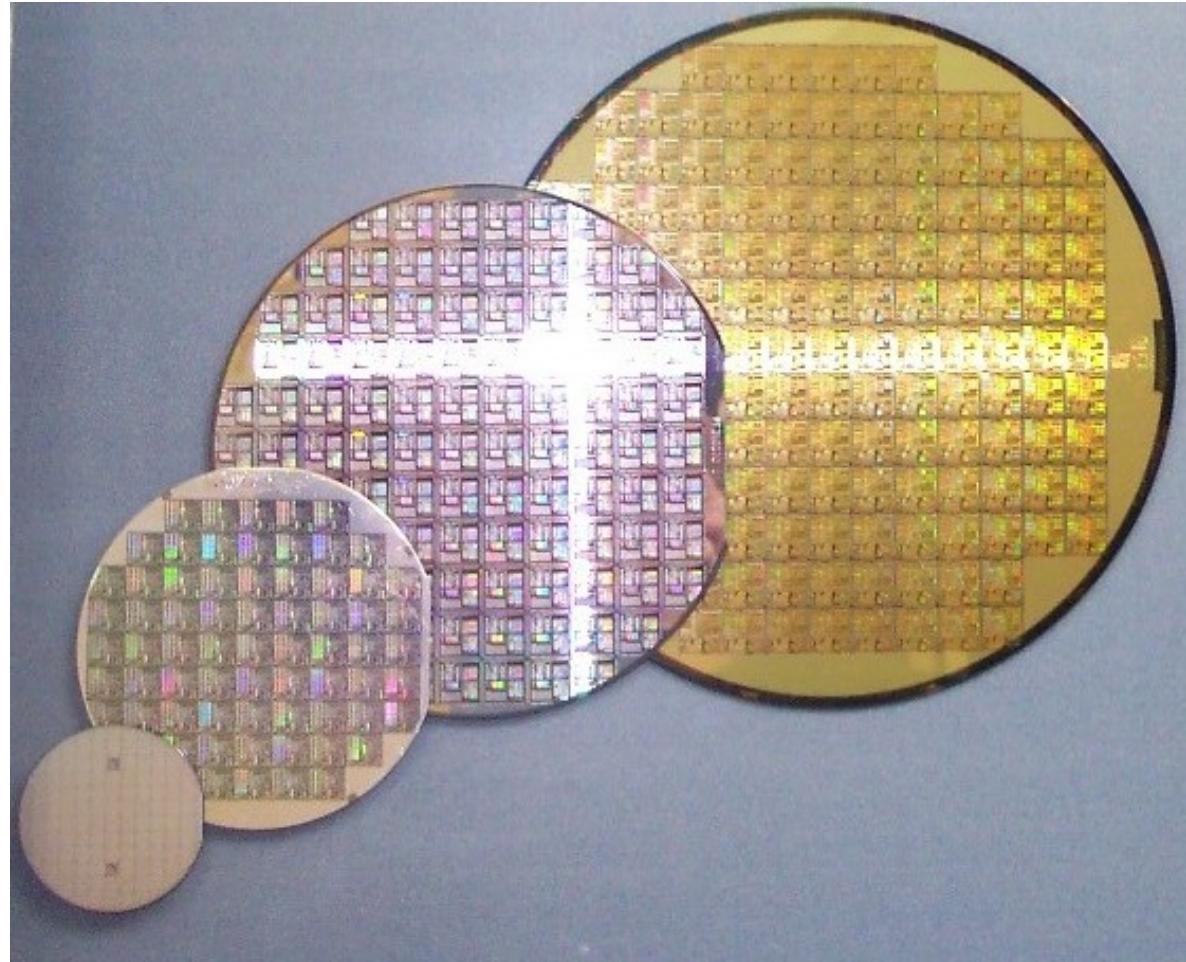


Technologie procesorů a pamětí (2)

...



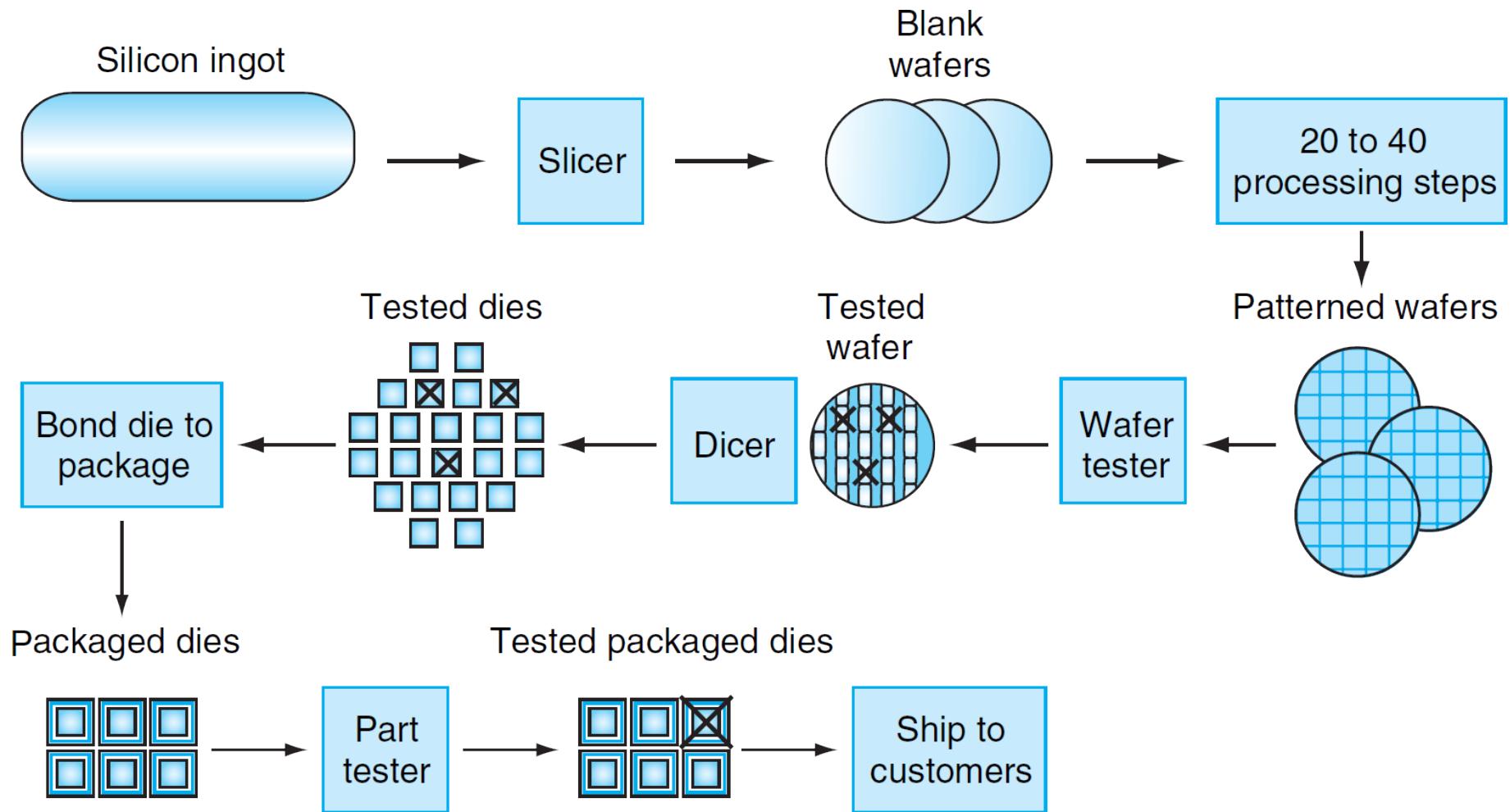
[6]



[7]



Technologie procesorů a pamětí (3)



Source: P&H



Sekundární paměť



● Permanentní paměť

- Data uchovávána i po odpojení napájení
- Datové soubory a spustitelné programy
- Není přímo adresovatelná procesorem ☠
(jedná se o vstupně/výstupní periferii,
vyžaduje programové řízení)

■ Pevné disky

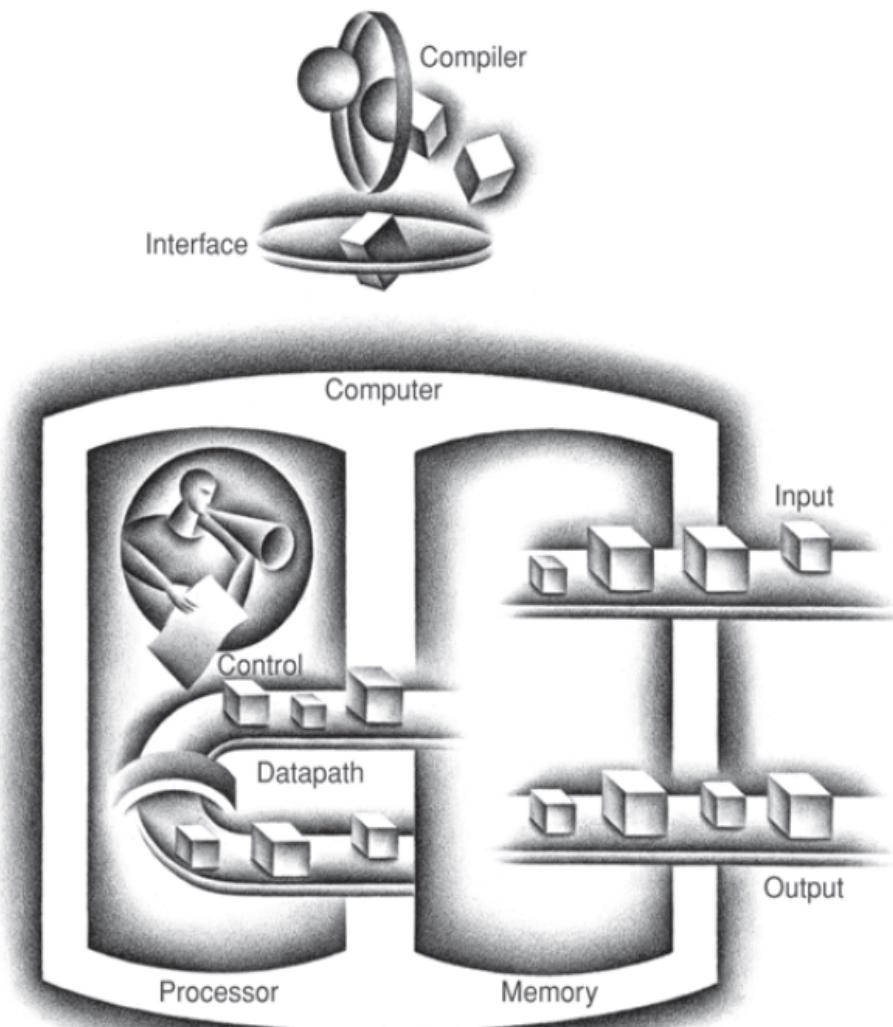
- Magnetické rotační médium
- Adresace po sektorech (512 B, 4 KB),
přístupová doba v řádu jednotek až desítek
ms (není konstantní)

■ Solid-State Drive (SSD), Flash paměť

- Nepohyblivá, tranzistorová permanentní
paměť (*floating-gate MOSFET*)
- Čtení po bitech, zápis po blocích
(asymetrické operace čtení a zápisu),
konstantní přístupová doba v řádech desítek
až stovek μ s



Základní organizace počítače



Zdroj: P&H

- **Počítač**

- vstup
- výstup
- paměť
- procesor
 - datová cesta
 - řízení

- **Nezávisí na technologii**

- pasuje na současné i minulé počítače



- **Vstupní zařízení**
 - klávesnice, myš, tablet, snímač otisků, joystick
- **Výstupní zařízení**
 - CRT monitor, LCD panel, grafická karta, tiskárna
- **Vstupně/výstupní zařízení**
 - síťová karta, pevný disk, zvuková karta, kamera, volant + pedály se zpětnou vazbou (force-feedback), ...

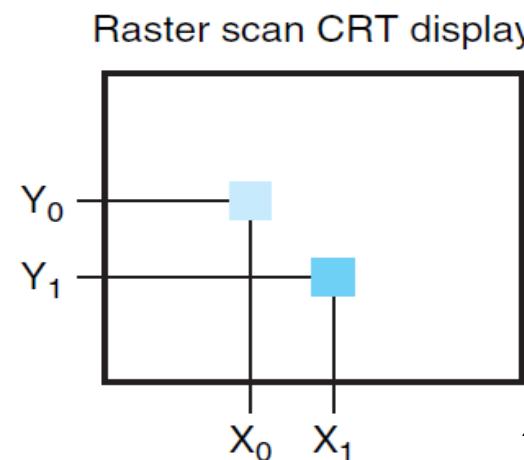
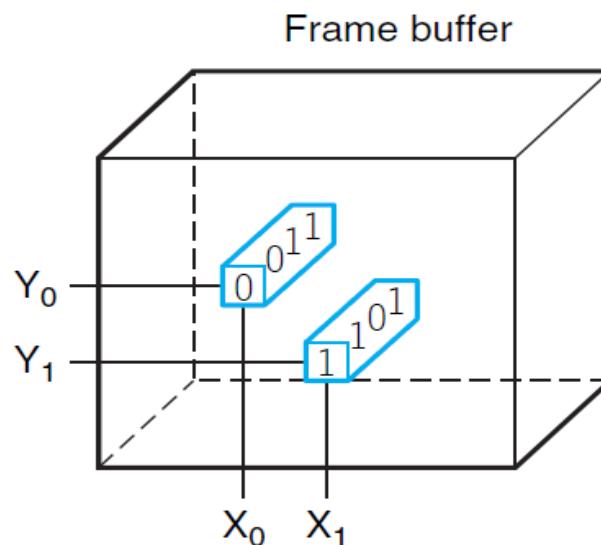


Grafický výstup na obrazovku

...

- **Paměť na grafické kartě (framebuffer)**

- každé místo v paměti (nebo shluk několika míst) odpovídá jednomu obrazovému bodu
- obsah místa v paměti reprezentuje barvu
- velikost místa určuje barevné rozlišení



Zdroj: P&H



Aneb ...

**Co nevidíte na
svém (oblíbeném)
programu?**



Od zapnutí počítače k běžící aplikaci



- **Firmware**

- BIOS (Basic Input/Output System)

- **Zavaděč operačního systému**

- Boot sector
 - Boot loader

- **Operační systém**

- **Uživatelské rozhraní**

- **Aplikace**



Stovky tisíc řádků kódu



● Aplikační software

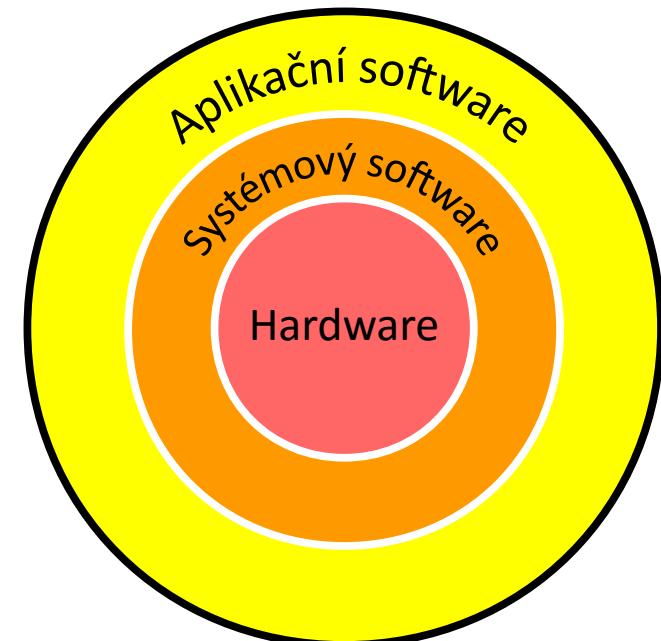
- Např. textový editor
- Knihovny pro uživatelské rozhraní

● Systémový software

- Operační systém
 - Vstupně/výstupní operace
 - Alokace paměti a úložného prostoru
 - Sdílení prostředků
- Firmware

● Hardware

- Procesor, paměť, periferie



Od uživatele k algoritmu

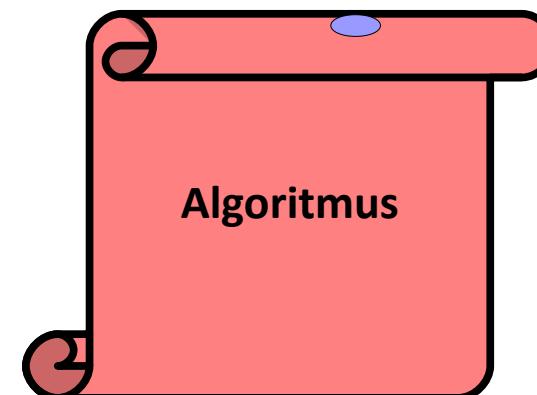


Od uživatele k algoritmu

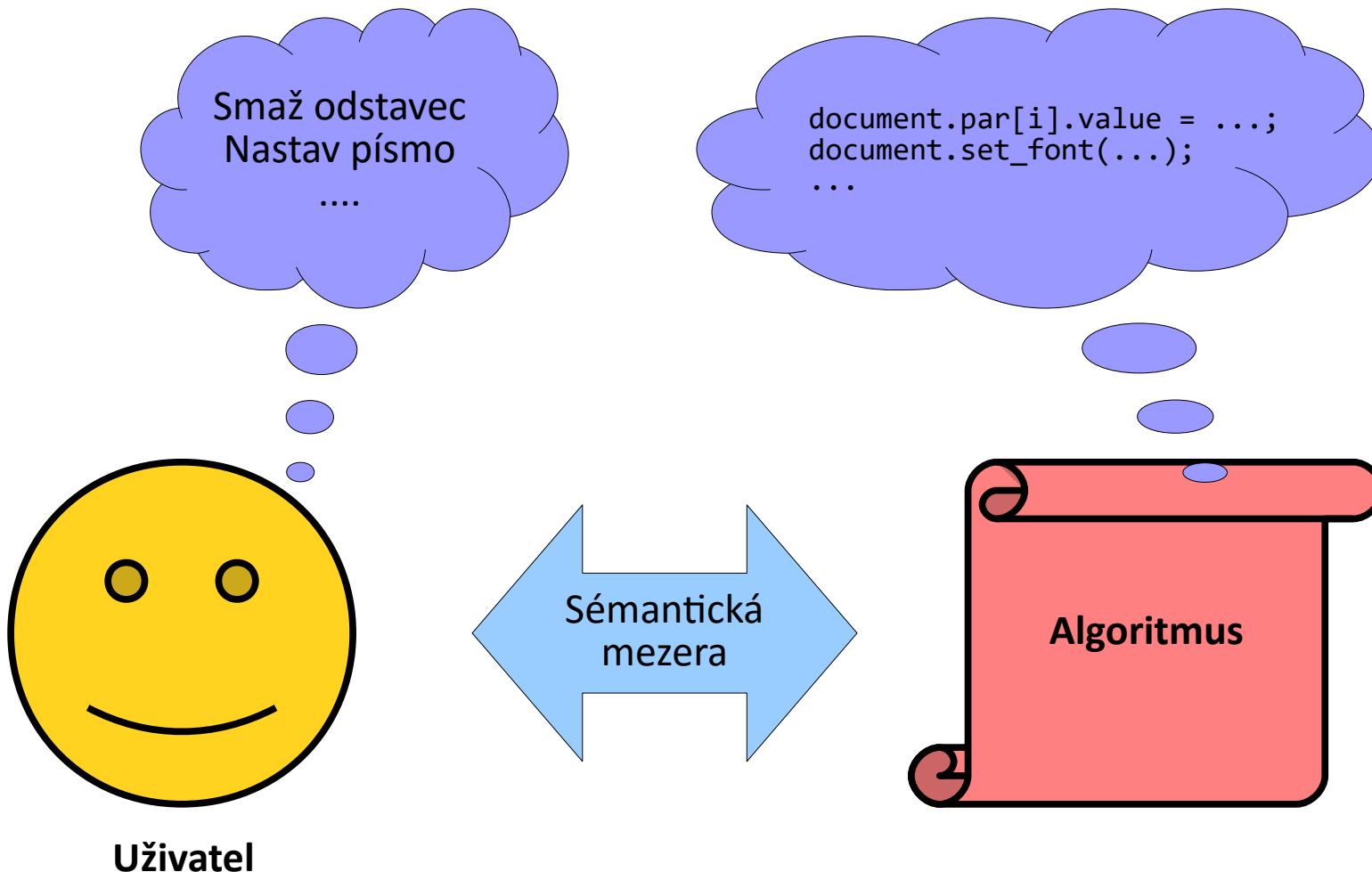


Smaž odstavec
Nastav písmo
....

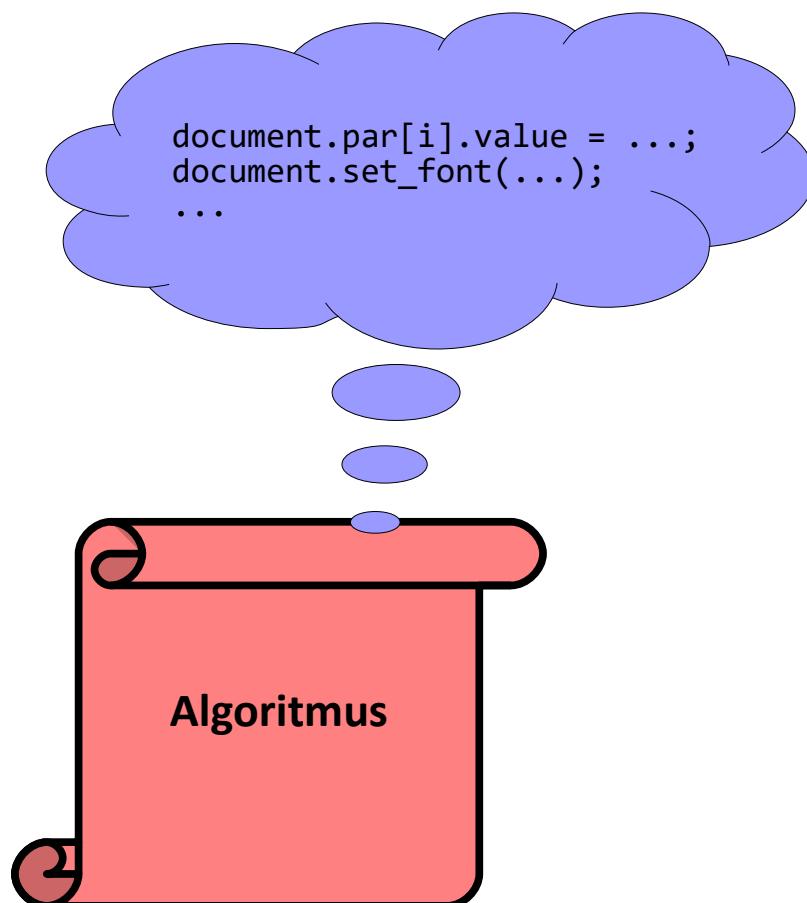
document.par[i].value = ...;
document.set_font(...);
...



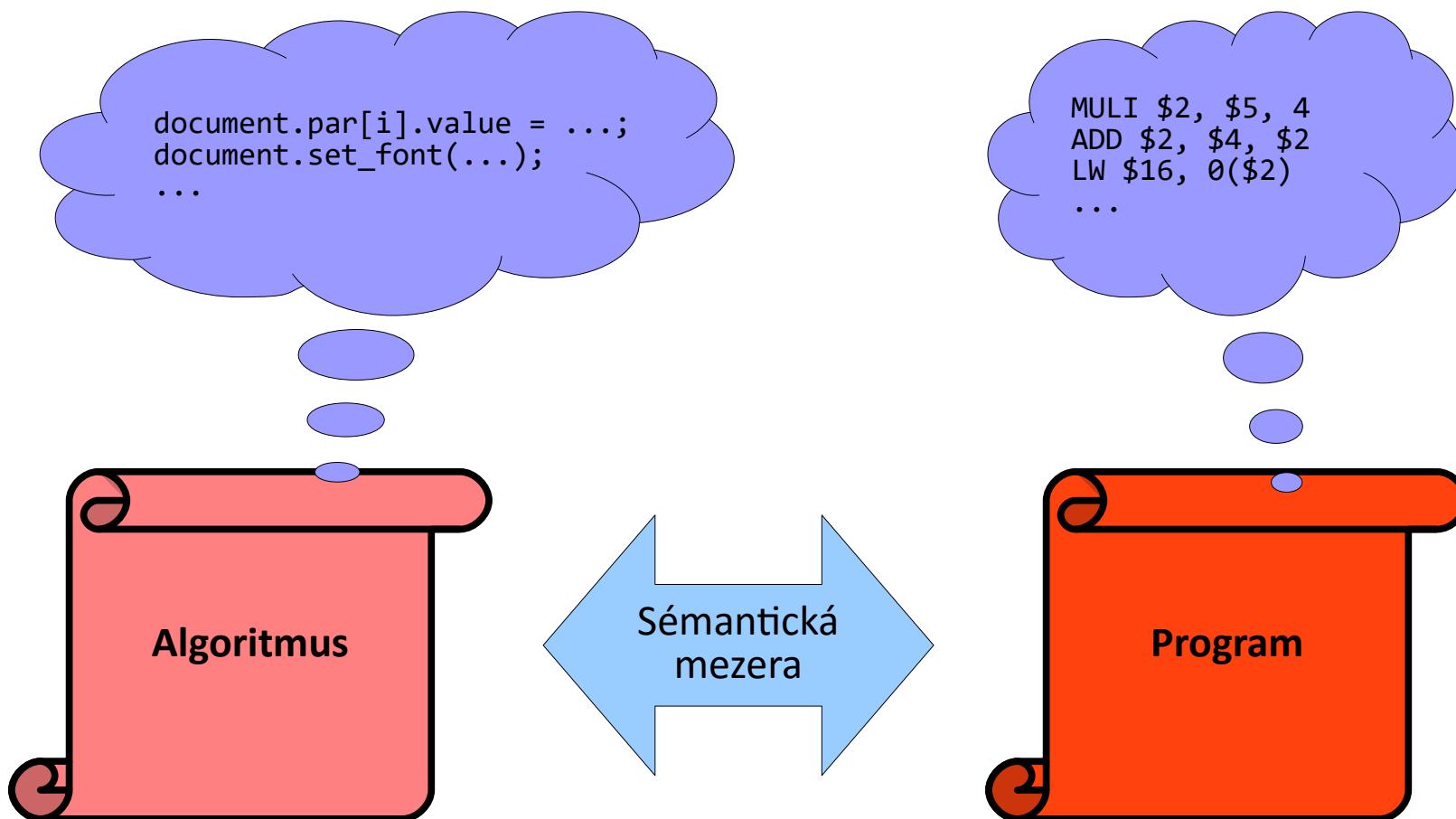
Od uživatele k algoritmu



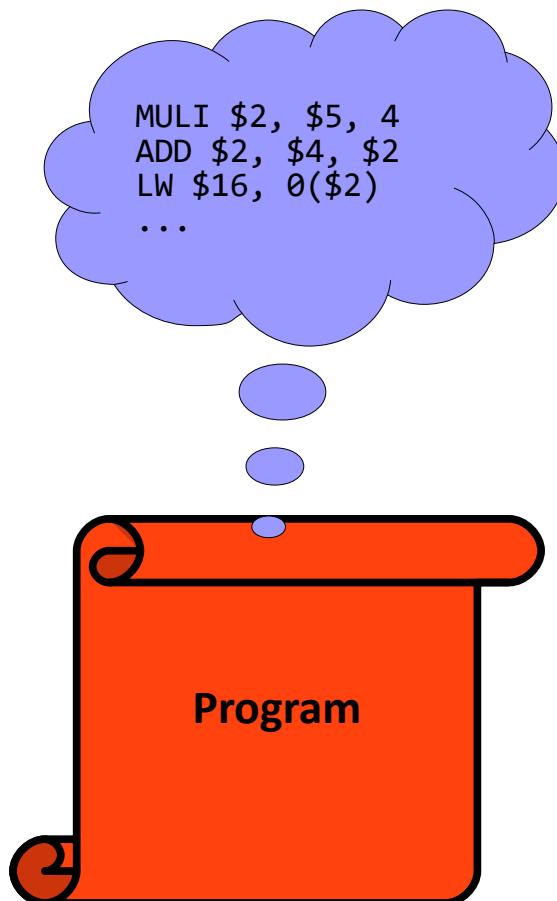
Od algoritmu k programu



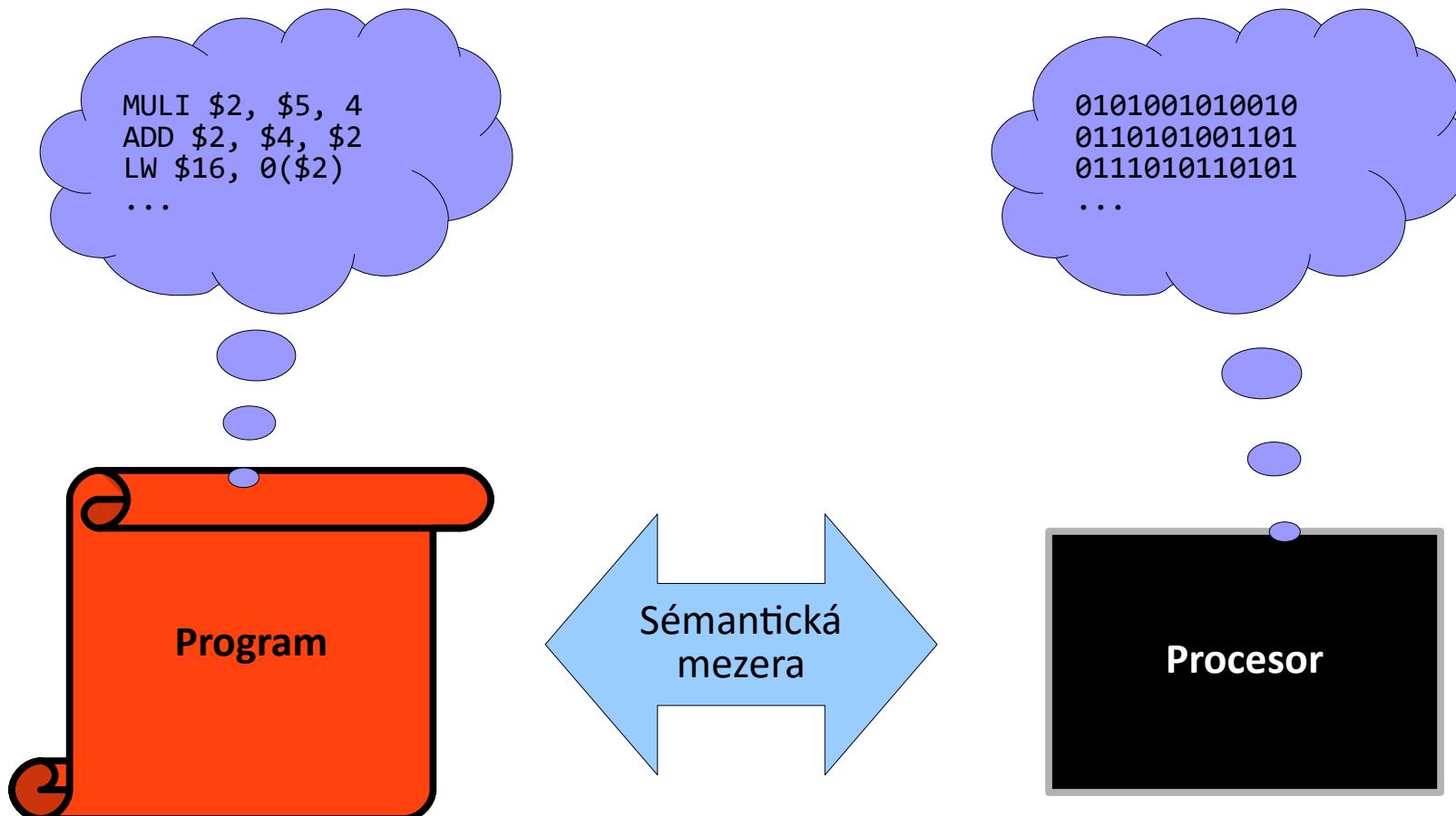
Od algoritmu k programu



Od programu ke strojovému kódu



Od programu ke strojovému kódu



Abstrakce





● Překonávání sémantických mezer

- Postup od konkrétního (technického) jazyka k abstraktnímu (obecnému) jazyku
- Ideálně se zachováním přesnosti, ale využití šířeji definovaných pojmu a „zapouzdření“ vnitřních detailů
 - Stručnější a kompaktnější vyjádření
- *„An abstraction is one thing that represents several real things equally well.“* (Edsger Dijkstra)



Abstrakce (2)



- **Vhodný jazyk pro danou úroveň abstrakce**

- Řídící logika procesoru
 - Pomocí jednotky ALU sečti hodnotu z registru A a hodnotu z registru B, výsledek ulož do registru C
- Strojový kód: instrukce (slova) nad abecedou {0, 1}
 - 1000110010100000
- Assembler: symbolický zápis instrukcí programu
 - add R2, R3, R1
- Vyšší programovací jazyk: symbolický zápis algoritmu
 - fruits := apples + oranges



Abstrakce (3)



● **Vhodný jazyk pro danou úroveň abstrakce**

- Překlad mezi jazyky = překonávání sémantické mezery
- Jazyk vyšší úrovně = vyšší produktivita člověka
 - Doménově-specifické jazyky
- Jazyk nižší úrovně = vyšší produktivita stroje
 - Strojový kód
- Překladač
 - Překlad z vyššího programovacího jazyka do jazyka nižší úrovně (často až do symbolického zápisu instrukcí konkrétního procesoru)
- Assembler
 - Překlad ze symbolického zápisu instrukcí do binárního kódu vykonatelného konkrétním procesorem



Příklad: Záměna k. a (k+1). prvku pole



- Vyšší programovací jazyk

```
void swap(unsigned int array[], unsigned int k) {  
    unsigned int old = array[k];  
    array[k] = array[k + 1];  
    array[k + 1] = old;  
}
```



Příklad: Záměna k. a (k+1). prvku pole (2)



- Symbolický zápis instrukcí pro MIPS

swap:

```
sll $a1, $a1, 2  
addu $a1, $a1, $a0  
lw $v0, 0($a1)  
lw $v1, 4($a1)  
sw $v1, 0($a1)  
sw $v0, 4($a1)  
jr $ra
```



Příklad: Záměna k. a (k+1). prvku pole (2)



- Symbolický zápis instrukcí pro SuperH

swap:

```
shll2 r5  
mov    r4,r1  
add    r5,r1  
mov.l  @r1,r2  
add    #4,r5  
add    r5,r4  
mov.l  @r4,r3  
mov.l  r3,@r1  
rts  
mov.l  r2,@r4
```



Příklad: Záměna k. a (k+1). prvku pole (3)



- Symbolický zápis instrukcí pro x86-64

swap:

```
movslq %esi, %rsi  
leaq    (%rdi, %rsi, 4), %rdx  
leaq    4(%rdi, %rsi, 4), %rax  
movl    (%rdx), %ecx  
movl    (%rax), %esi  
movl    %esi, (%rdx)  
movl    %ecx, (%rax)  
retq
```



Příklad: Záměna k. a (k+1). prvku pole (4)



- **Zápis ve strojovém kódu pro MIPS**

```
000000000000101001010001000000  
0000000101001000101000010001  
100011001010001000000000000000  
10001100101000110000000000000100  
10101100101000100000000000000100  
10101100101000110000000000000000  
0000011110000000000000001000
```



Příklad: Záměna k. a (k+1). prvku pole (4)



- **Zápis ve strojovém kódu pro SuperH**

```
0000100001000101  
0100001101100001  
0101110000110001  
0001001001100010  
0000010001110101  
0101110000110100  
0100001001100011  
0011001000100001  
0000101100000000  
0010001000100100
```



Příklad: Záměna k. a (k+1). prvku pole (5)



- **Zápis ve strojovém kódu pro x86-64**

010010000110011111110110

01001000100011010001010010110111

010010001000110101000100101101110000100

1000101100001010

1000101101110000

1000100101110010

1000100100001000

11000111



- **Opačný směr než abstrakce**

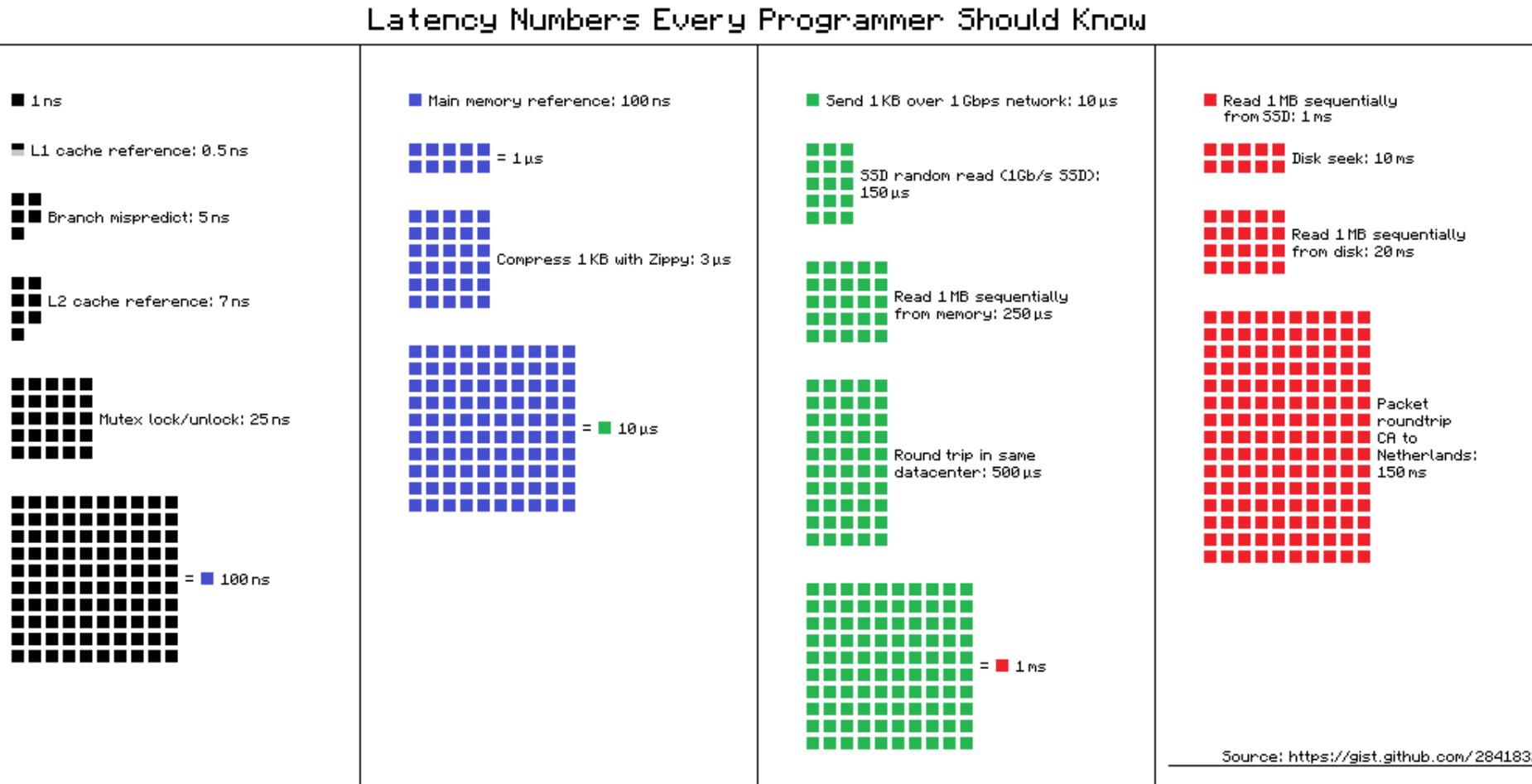
- Konkretizace
- Od architektury počítače ke konkrétnímu fyzickému počítači
- Vysokoúrovňový jazyk
 - Blokové schéma, logický popis obvodů
- Nízkoúrovňový jazyk
 - Schéma zapojení elektronických součástek, schéma polovodičových prvků na integrovaném obvodu
- „Strojový kód“
 - Samotná fyzická realizace počítače



Vrstvy abstrakce počítače



Pozor: abstrakce je jen nástroj!



Source: <https://qist.github.com/2841832>

