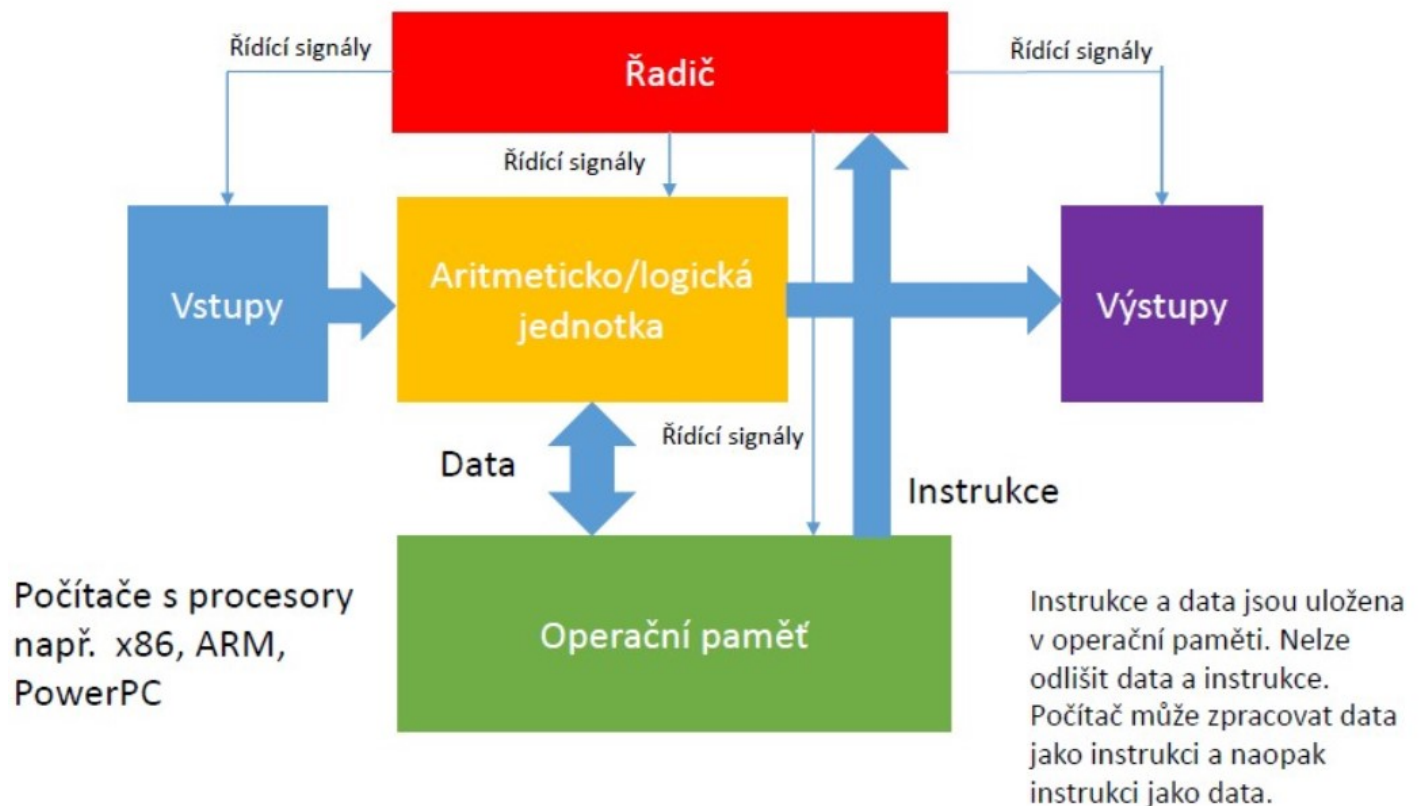


Architektura procesorů

Od automatu k RISC architektuře

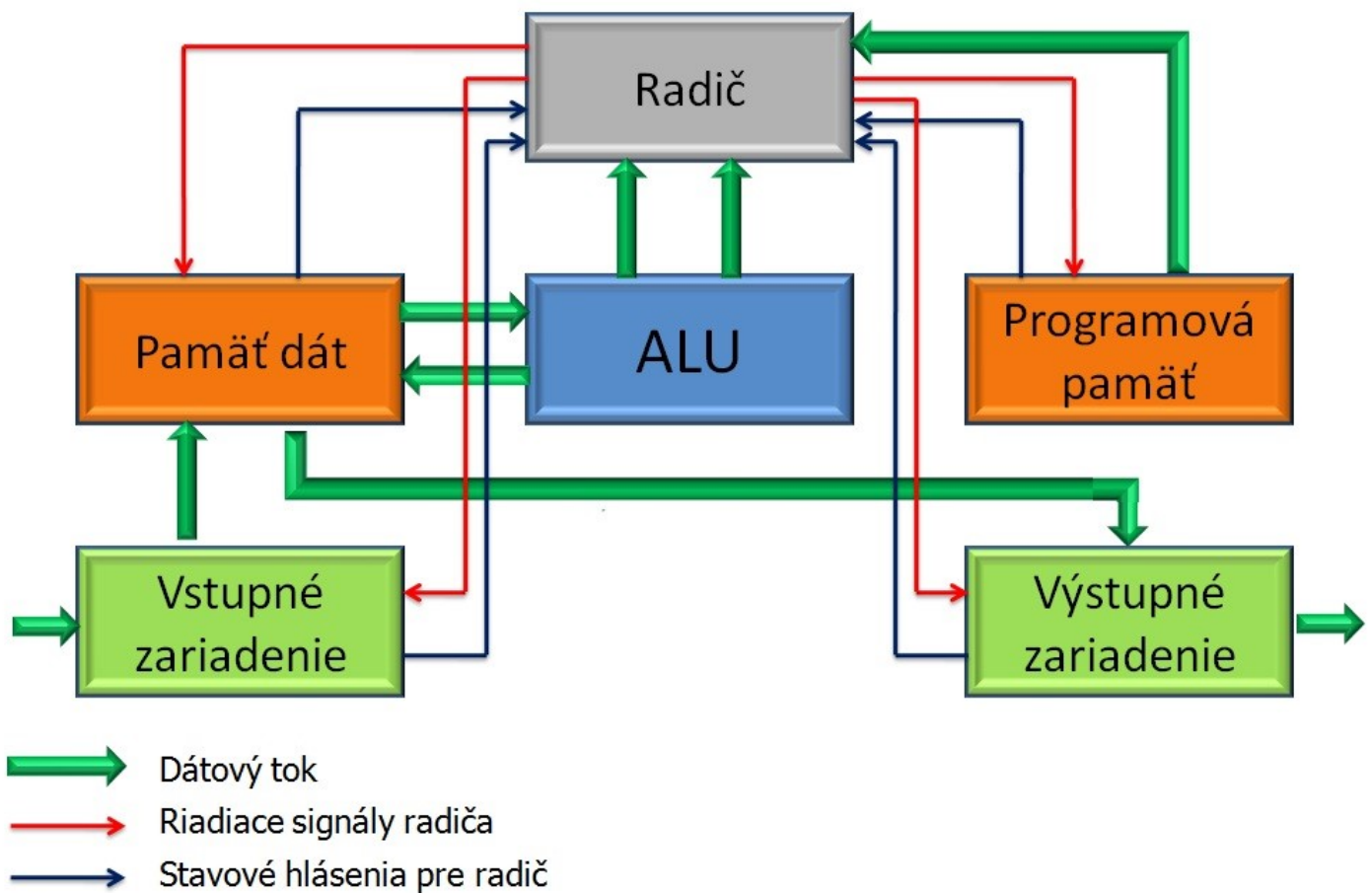
- Z počátku byly počítače specializované na určité výpočty a daly se přeprogramovat pouze změnou zapojení - automaty
- Koncept Johna Von Neumanna položil základ pro univerzální programovatelné stroje. Díky tomuto konceptu vznikl procesor, který vykonává zadaný program

Von Neumannova architektura



- Nejvyšší úroveň abstrakce
- **vstupy/výstupy** - periférie
- **ALU** - kombinační obvod pro nejdůležitější operace
- **Řadič** - sekvenční obvod, který generuje řídicí signály pro ostatní bloky
 - řídí přenosy po sběrnicích
 - dekoduje instrukce a generuje příslušné řídicí signály
 - data jsou rovněž uložena v operační paměti a nelze rozpoznat zda jsou to data nebo instrukce

Harvardská architektura



Paměť programu

- typ flash
- instrukce a konstanty v programu jsou uchovány i době vypnutí

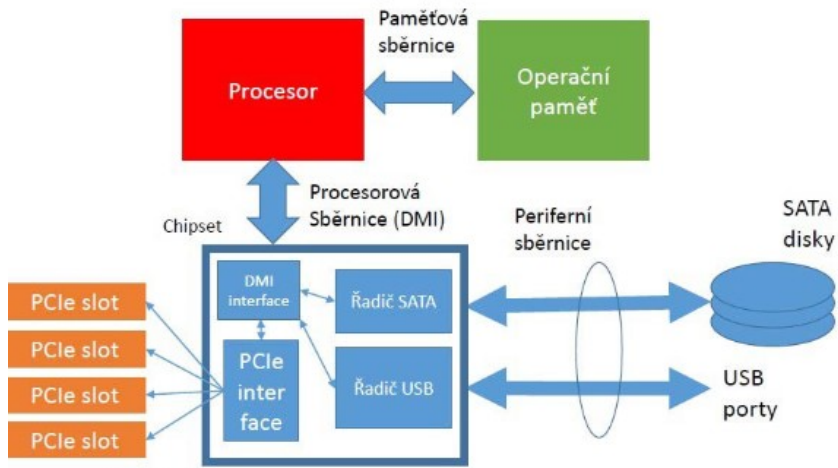
Paměť dat

- statická RAM
- data se ztratí po vypnutí

obecně

- Každá paměť může mít jinou velikost nejmenší adresovací jednotky
- program nemůže přepsat sám sebe
- Dvě sběrnice umožňují jednoduchý paralelizmus kdy lze přistupovat pro instrukce i data současně
- příklad ESP82668

Architektura současného počítače



Různé sběrnice podle účelu

- obsahují adresy, data, řídicí a stavové signály, jsou specificky definované. mají různé protokoly a časování

Procesor implementuje stále větší část

- integruje řadič operační paměti - dříve north bridge chipsetu základní desky

Paměťová sběrnice

- Direct media interface
- Chipset zajišťuje komunikaci mezi periferiemi a procesorem pomocí rozšiřujících a periferních sběrnic

Registry procesoru

- udržují stav procesoru
- slouží k odkládání mezivýsledků
- tvoří operandy aritmetických a logických operací
- typické registry
 - program counter
 - instrukční registr
 - stavový registr
 - stack pointer
 - General purpose registers

Řadič procesoru

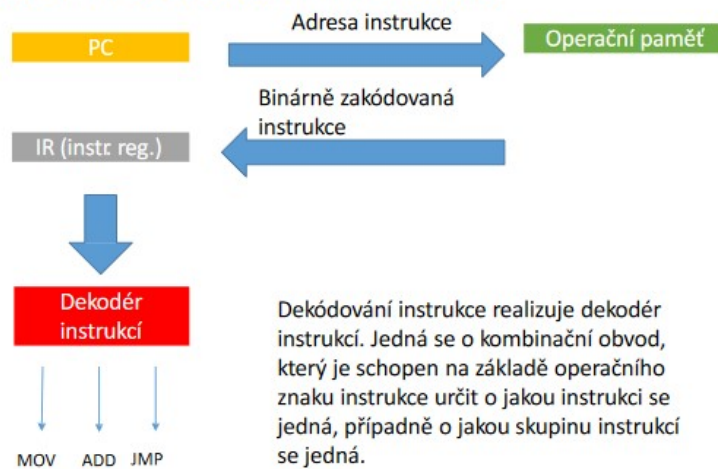
- Realizuje instrukční cyklus
- Řídí vykonávání dílčích operací v rámci instrukčního cyklu
- generuje řídicí signály
- Reaguje na stavové signály
 - Přerušení
- Realizace
 - Obvodový řadič
 - konečný automat
 - D-flip flop + kombinační logika
 - Mikroprogramový řadič
 - realizuje složitější instrukce

- má paměť pro uložení mikroinstrukcí
- instrukce procesoru je realizována vykonáním sady mikroinstrukcí

Instrukční cyklus počítače

- IF
 - načtení instrukce
- ID
 - dekódování instrukce
- OF
 - načtení operandů
- EX
 - vykonání instrukce
- WB
 - zapsání výsledku
- Interrupt detection

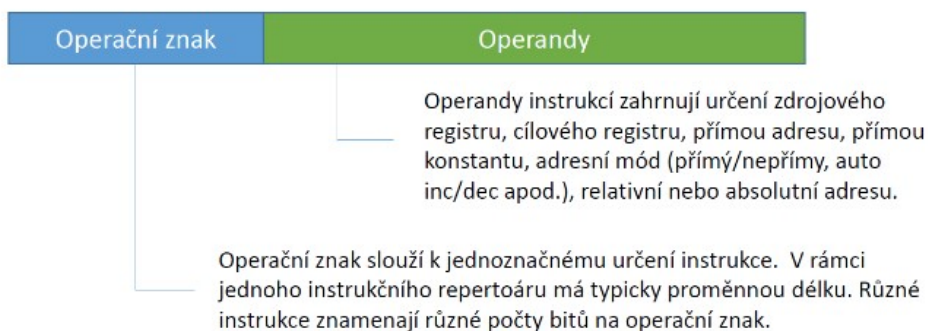
Načtení a dekódování instrukce



- přerušení

Instrukce ve strojovém kódu

- instrukce s pevnou délkou
 - instrukční repertoár může mít delší instrukce, ale ty jsou minoritně zastoupeny
- instrukce s proměnnou délkou
 - délka se mění například od jednoho do několika bytů



Průběh instrukce

- Načtení operandu
 - přesun operandu z registrů na vstupy ALU
- Vykonání instrukce
 - provedení všech operací
- Zapsání výsledku
 - zapsání do registrů nebo paměti
 - U instrukce ADD se jedná o zápis výstupu ALU do registrů nebo paměti

typy instrukcí

- Aritmetické
- logické
- posuvy
- skokové
- přesuny

Instruction set architecture

- tvoří rozhraní mezi hardware a software
- má zásadní vliv na architekturu procesoru
- Zahrnuje
 - Registry procesoru
 - instrukční sadu
 - kódování instrukcí do binární podoby
 - adresuje prostory
 - výjimky

CISC

- Complex instruction set computer
- počítač s rozsáhlým souborem instrukcí

Instrukční sada obsahuje

- složité instrukce i jednoduché
 - kopírování bloku dat do paměti
- typicky různá délka instrukcí
- Původní snaha: urychlit vykonávání instrukcí realizovat je stále složitější
- **Pozitiva**
 - snížena četnost načítání instrukcí
 - snaha načíst pouze jednou
 - možnost vícenásobného využití funkčních jednotek v různých fázích vykonávání instrukce
 - Přítomnost mikroprogramového řadiče dává možnost změnit repertoár
- **Negativa**
 - Složité instrukce jsou specializované
 - mnoho variant
 - velký počet instrukcí
 - složitý dekodér

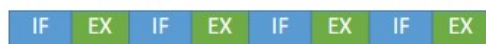
- Nutnost mikroprogramovatelných řadičů
- instrukce trvají různě dlouho

RICS

- Reduced instruction set computer
- v jednoduchosti je síla
- kódování stejným počtem bitů
- vykonáno v jednom nebo několika málo taktech hodinového signálu
- Nemůžeme dělat složité instrukce ale pomocí jednoduchých dosáhnout stejného výsledku

CISC vs. RISC

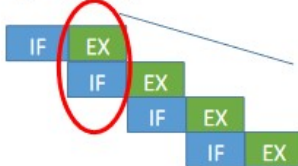
CISC (jednoduché instrukce)



CISC (složité instrukce)



RISC (má pouze jednoduché instrukce)



RISC architektura využívá proudového zpracování instrukcí. Načtení instrukce a vykonání předchozí instrukce probíhá paralelně. IF a EX fáze musí trvat stejně dlouho a počet instrukcí vykonaných za sekundu je převrácená hodnota trvání fáze EX. V ideálním případě se výkon procesoru zdvojnásobí proti variantě CISC jednoduché instrukce. Proudové zpracování instrukcí připomíná pásovou výrobní linku.

Výkon procesoru v instrukcích za sekundu je roven převrácené hodnotě součtu trvání fáze IF a EX.

Výkon procesoru je vyšší (méně fází IF) a silně závisí na podílu složitějších instrukcí. Jednodušší instrukce se typicky používají častěji. Nikdy není možno dosáhnout zdvojnásobení výkonu.

interlocked pipeline – podmínky při řazení instrukcí do pipeline zajišťuje přímo procesor

non-interlocked pipeline - podmínky při řazení instrukcí hlídá programátor (překladač)

Porovnání vlastností CISC a RISC

	CISC	RISC
Časová složitost instrukcí:	může probíhat mnoho hodinových cyklů	většina trvá jeden hodinový cyklus
Práce s pamětí:	jednoduchá	složitější
Instrukce:	komplexní (například více operandů než dva)	primitivní standardizované instrukce
Počet instrukcí	průměrně 100-200 i více	většinou méně než 100
Instrukce, které mohou přistupovat do paměti:	Load a Store	téměř všechny.

Sběrnice

- Sběrnice je skupina vodičů, které propojují dvě a více zařízení
- skupina vodičů tvoří určitý logický celek sloužící k jednomu účelu
- u sběrnic musí být definováno CLOCK signál případně protokol, kterého jsou data přenášeno (např. hrana clock signálu)
- V daném okamžiku může být připojeno pouze jedno výstupní zařízení

Taxonomie sběrnice

účelu

- adresová
 - přenos adresy mezi procesorem, pamětí a ostatními částmi systému
- datová
 - přenos dat mezi -||-
 - za datovou sběrnici můžeme pokládat za jakoukoliv sběrnici, po které se přenášejí data
- řídicí sběrnice
 - slouží pro přenos řídicích signálů
 - READ
 - WRITE
 - BYTE ENABLE
- systémová sběrnice
 - sběrnice pro přenos dat mezi procesorem, pamětí a periferiemi
 - typicky zahrnuje adresovou, datovou a řídicí sběrnici, ale může se jednat i jednu sběrnici jejíž protokol implementuje přenos adresy dat a realizaci čtecích a zapisovacích cyklů do paměti a periferií
 - Transakce na systémové sběrnici jsou přímo vyvolány instrukcemi pro zápis/čtení paměti a ve vstupně/výstupním adresním prostoru
 - příklady
 - PCI, PCIe, HyperTransport, DMI
- periferní sběrnice
 - sběrnice mezi řadičem periferních sběrnic a periferiemi na dané sběrnici
 - příklady
 - USB, SATA, SAS, SCSI, SMBus

přenos dat

- Jednosměrná
 - typicky adresová sběrnice
 - dvě jednosměrné sběrnice ale v opačném směru realizuje full duplexní přenos dat mezi dvěma body
 - cache
 - Umožňuje broadcast
- obousměrná
 - přenos jedním a druhým směrem se multiplexuje v čase
 - může probíhat pouze přenos jedním směrem zároveň
 - half duplex
 - data můžeme přenášet na více míst
 - pokud je třeba data z více míst propojených sběrnicí, musí se vyloučit kolize
 - o tom rozhoduje arbitrace sběrnice

Způsob přenosu dat

- Paralelní sběrnice
 - přenos dat probíhá paralelně po více vodičích
 - data ale musí dorazit současně

- při dnešních rychlostech přenosu hraje roli délka vodičů
- kompenzace meandrem
- Seriová sběrnice
 - přenos dat probíhá postupně
 - přenos bitů je roložen v čase
 - Jednotlivé bity jsou přenášeny jeden za druhým v intervalech

