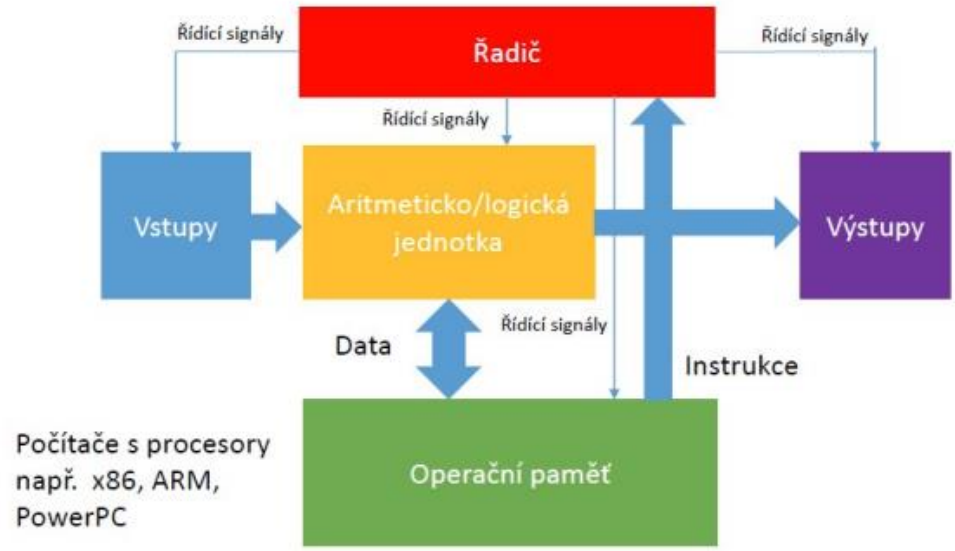
Von Neumannova architektura, Harvardská architektura počítače, architektura RISC a CISC, instrukční cyklus, zřetězení instrukcí, taxonomie sběrnic, paralelní, sériový, synchronní, asynchronní přenos dat, otevřený kolektor, třístavový budič

Von Neumannova architektura



Řadič

- Sekvenční obvod
- Generuje **řídící signály pro ostatní bloky, které nazpět zasílají stavové hlášení**
- Řídí přenosy po sběrnicích
- Načítá instrukce z operační paměti a řadič je dekóduje a generuje příslušné řídící signály

ALU (Aritmeticko-logická jednotka)

- Kombinační obvod
- Činnost je ovlivňována pomocí příznaků (Flag)
- Podle šířky ALU se udává, kolika bitový systém je
- Zde probíhají nejdůležitější operace
    - AND, OR, XOR, NOT, sčítaní, odčítání, dělení, násobení, posun vpravo/vlevo

Operační paměť

- Z výpisu operační paměti nelze přesně určit, co jsou instrukce a co data

Data

- Jsou uložená v operační paměti
- Počítač může zpracovat data jako instrukce a naopak (nevýhoda)

Vstupní zařízeními

- Zařízení určená pro vstup programu a dat

Výstupní zařízení

- Zařízení určená pro výstup výsledků, které program zpracoval

Princip

- Do operační paměti se pomocí vstupních zařízení přes ALU umístí program, který bude provádět výpočet
- Stejným způsobem se do operační paměti umístí data, která bude program zpracovávat
- Proběhne vlastní výpočet, jehož jednotlivé kroky provádí ALU, která je řízena radičem (stejně jako všechny ostatní moduly)
- Mezivýsledky jsou ukládány do operační paměti
- Po skončení výpočtu jsou výsledky poslány přes ALU na výstupní zařízení

Harvardská architektura

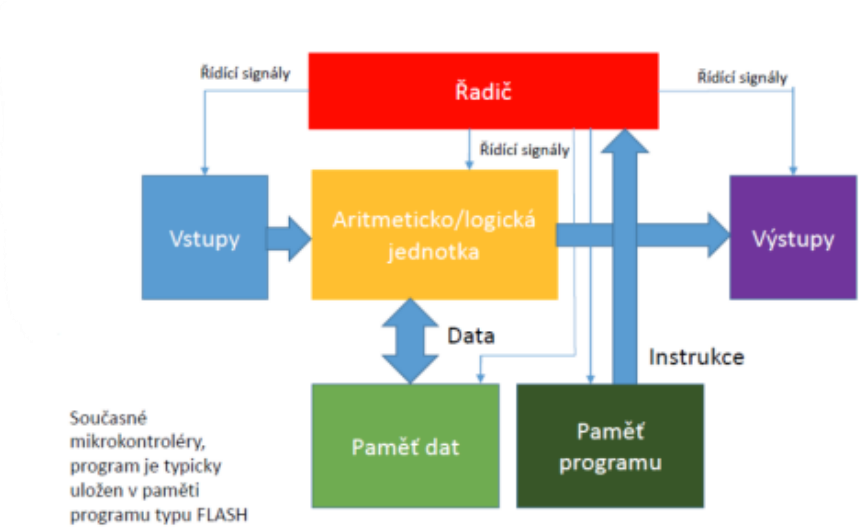
- Řeší nedostatek Von Neumannově architektury (instrukce a data v operační paměti)
- **Operační paměť rozdělena do dvou dalších bloků**
- Lze přistupovat pro instrukce i data současně
- Každá paměť může mít jinou velikost
- Program nemůže přepsat sám sebe

Paměť dat

- Typ statická RAM
- Data se ztratí vypnutím

Paměť programu

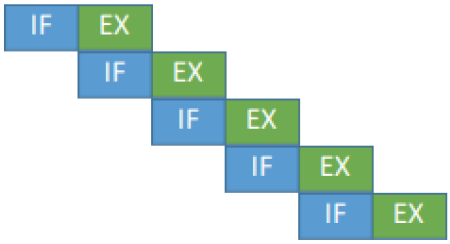
- Typ flash
- Instrukce programu a konstanty
- Uchováno i v době vypnutí



Instrukční sady

RISC (Reduced instruction Set Computer)

- **Jednoduché instrukce**
- Typicky vykonány v jednom, nebo několika málo taktech hodinového signálu
- Instrukce mají pevnou délku a jednotný formát, který vymezuje význam jednotlivých bitů
- Vyšší počet registrů propojených přímo s ALU
- Například nemůžeme násobit jednotlivé buňky paměti, ale základními instrukcemi si postupně obsah pamětí přesunout do registrů, v registrech provést násobení a poté výsledek zapsat na požadovanou adresu
- Proudové zpracování instrukcí
- **Načtení a vykonání instrukce probíhá paralelně**
- Používá se u mobilních aplikací, kde dominuje architektura ARM, AVR, MIPS



- IF a EX trvají musí trvat stejně dlouho
- V ideálním případě je RISC 2x rychlejší než CISC

Výhody

- Jednoduchost – malé množství instrukcí, rychlé dekódování
- Rychlý obvodový řadič

CISC (Complex Instruction Set Computer)

- Složité i jednoduché instrukce
- Instrukcí je hodně
- Různá délka instrukcí

Jednoduché instrukce

- Výkon procesoru v instrukcích za sekundu = převrácená hodnota součtu trvání fáze IF a EX

Složité instrukce

- Výkon procesoru je vyšší (méně fází IF), závisí na podílu složitějších instrukcí
- Jednodušší instrukce se typicky používají častěji

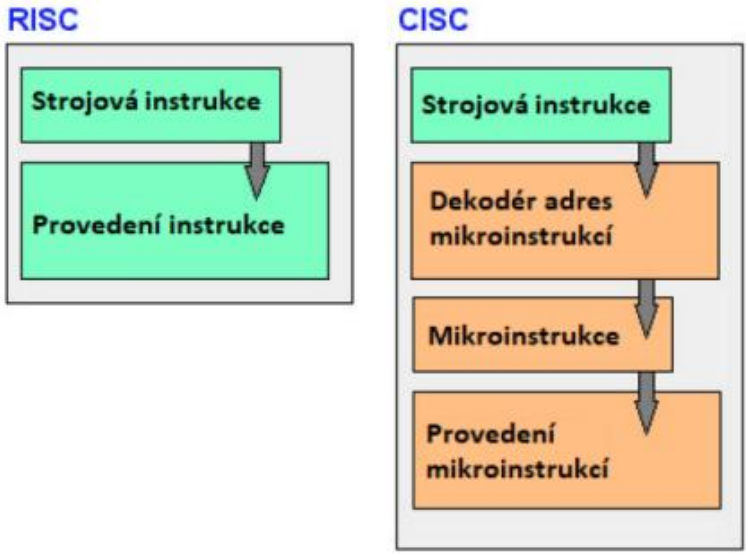
Výhody

- Snížená četnost načítání instrukcí – IF (snaha načíst jenom jednu, zbytek provede mikrořadič)
- Díky přítomnosti mikroprogramového řadiče lze změnit instrukční repertoár
- Možnost vícenásobného využití funkčních jednotek v různých fázích vykonání instrukce

Nevýhody

- Nutnost mikroprogramovatelných řadičů
- Velký počet instrukcí -> složitý dekodér instrukcí -> dekódování jednoduchých a obvykle nejčetnějších instrukcí trvá dlouho (např. sčítání)
- Instrukce trvají různě dlouho, těžko se zavádí proudové zpracování

Porovnání RISC a CISC



	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.

Instrukce

- **Nedělitelný** pokyn k vykonání činnosti

Strojový cyklus

- Čas potřebný k zápisu (čtení) slova z paměti
- Provedení **elementární** operace s registrem

Instrukční cyklus

- Čas potřebný pro výběr a provedení instrukce
  - IF (Instruction Fetch) – **načtení instrukce**
  - ID (Intruction Decode) – **dekódování instrukce**
  - OF (Operand Fetch) – **načtení operandů**
  - EX (Execute)- **vykonání instrukce**
  - Interrupt detection – **test žádosti o přerušení**
- Instrukce nemusí nutně využívat všech prostředků najednou

Zřetězení instrukcí

- Vykonání více instrukcí současně
- Načítání další instrukce, zatímco nějaká se dokončuje
- Nese to s sebou nějaké nevýhody – vzájemné ovlivňování apod.

Taxonomie sběrnic

Podle účelu

Adresová

- Slouží pro **přenos adresy** mezi procesorem, pamětí a ostatními částmi systému

Datová

- Slouží pro **přenos dat** mezi procesorem, pamětí a ostatními částmi systému
- Za datovou sběrnici obecně můžeme pokládat jakoukoliv sběrnici, po které se přenášejí data

Řídící

- Slouží pro **přenos řídících signálů** jako jsou například signály read (RD), write (WR), byte enable (BE)

Systémová

- Sběrnice pro přenos dat mezi procesorem, pamětí, periferiemi
- Typicky zahrnuje **adresovou, datovou a řídící sběrnici**, ale může se jednat i jednu sběrnici (např. PCI) jejíchž protokol implementuje přenos adresy, dat a realizaci čtecích a zápisových cyklů do paměti a periferií
- Transakce na systémové sběrnici jsou přímo vyvolány instrukcemi pro zápis/čtení paměti a ve stupně/výstupním adresním prostoru
- Příklady: PCI, PCIe, HyperTransport, DMI (Direct Media Interface)

Periferní

- Sběrnice mezi řadičem periferních sběrnic a periferiemi na dané sběrnici
- USB, SATA, SAS, SCSI, SMBus

Podle synchronizace

Synchronní

- je **synchronizován odděleným synchronizačním signálem**
- jsou přenášeny celé bloky dat
- datové bity dat následují těsně po sobě, bez jakýchkoli časových odstupů, a nejsou prokládány žádnými start či stop-bity (mohou však být doplněny jedním paritním bitem)
- Typickým příkladem sériová linka WAN (DCE nastavení clock_rate, DTE detect_clock)

Asynchronní

- jednotlivé znaky jsou přenášeny s libovolnými časovými odstupy mezi sebou přenos začíná **start-bitem** (začátek bloku dat + synchronizace)
- pak následuje přenos znaku, který může mít na konci paritní bit datový blok je zakončen **stop-bitem**
- Typickým příkladem komunikace na sériovém portu (nastavení bitrate, databits, stopbit, parity, Flow contrl u Hyperterminálu, PuTTy)

Podle směru přenosu dat

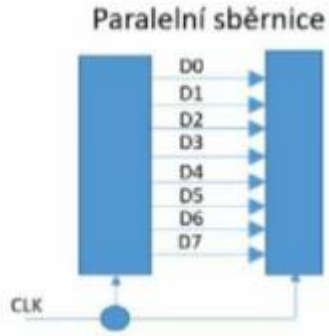
Jednosměrná

- **Dvě jednosměrné sběrnice** ale **v opačném směru** realizuje **full duplexní přenos** dat mezi dvěma body (např. cache)
- Umožňuje přenos dat z jednoho místa na více míst současně (broadcast)
- Typicky adresová sběrnice

Obousměrná (arbirtrace)

- Přenos jedním a druhým směrem se multiplexuje v čase
- Přenos jedním směrem nemůže probíhat současně s přenosem v druhém směru (přenos y procesoru do periferie, nebo naopak, např. čtení nebo zápis do periferie)
- Mluvíme často o tzv. **half duplexu**
- Data můžeme současně přenášet na více míst
- Pokud je třeba přenášet data z více míst propojených sběrnicí, musí se vyloučit kolize.
- Ze kterého místa se budou v dané okamžiku přenášet data rozhoduje proces zvaný arbitrace sběrnice

Podle způsobu přenosu dat



Paralelní

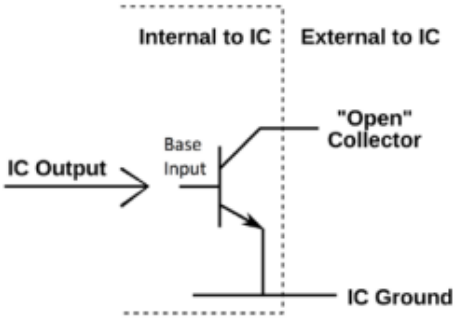
- Přenos dat probíhá paralelně po více vodičích např. 32 bitů
- Data musí **dorazit do cíle současně**.
- Při dnešních rychlostech přenosu hraje roli délka jednotlivých vodičů (kompenzace nestejné délky meandry)

Sériová

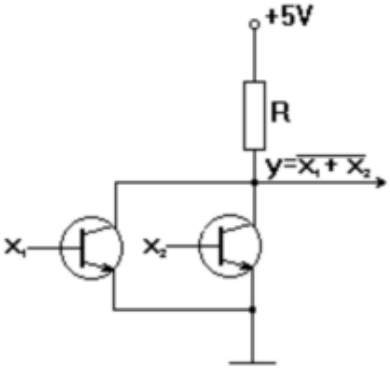
- Přenos dat probíhá **postupně**
- Přenos bitů je rozložen v čase
- Jednotlivé bity jsou tedy přenášeny jeden za druhým v pravidelných časových intervalech

Otevřený kolektor

- Tranzistorový obvod
- **Umožňuje komunikaci mezi různými obvody**, zejména mezi obvody s různými napájecími úrovněmi
- NOR



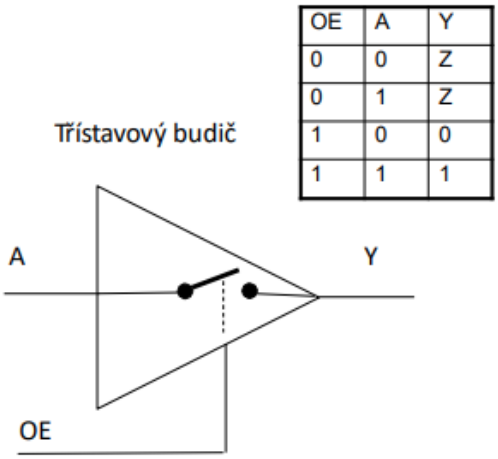
x_1	x_2	NOR
0	0	1
0	1	0
1	0	0
1	1	0



Třístavový budič

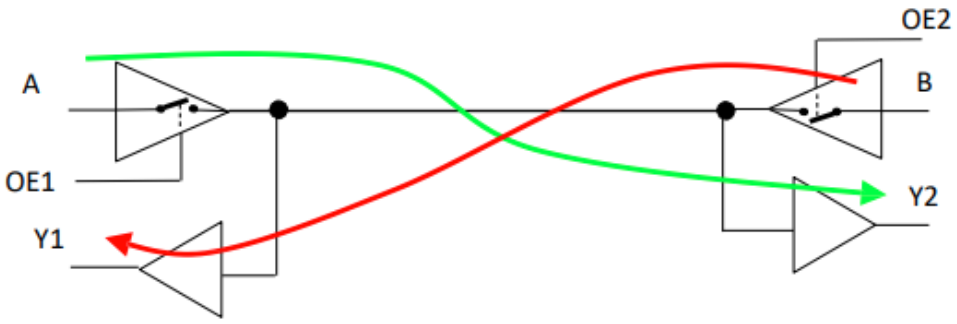
- **Umožňuje připojenému zařízení komunikovat s dalšími zařízeními na sběrnici**
- Nedochází k narušení datových signálů
- *Oboustranná směrnice
- Tři stavy
    - Logická nula (LOW)
    - Logická jednička (HIGH)
    - High-impedance (Hi-Z) – nevynucuje žádnou logickou odpověď

OE	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1



Pokud OE=1, pak Y kopíruje A

Pokud OE=0, budič se odpojí a na výstupu žádná logická úroveň



Přenos zleva doprava: OE1=1, OE2=0 (musí být! nula, jinak zkrat), Y2 má stejnou hodnotu jako A

Přenos z prava doleva: OE1=0 (musí být! nula, jinak zkrat), OE2=1, Y1 má stejnou hodnotu jako B