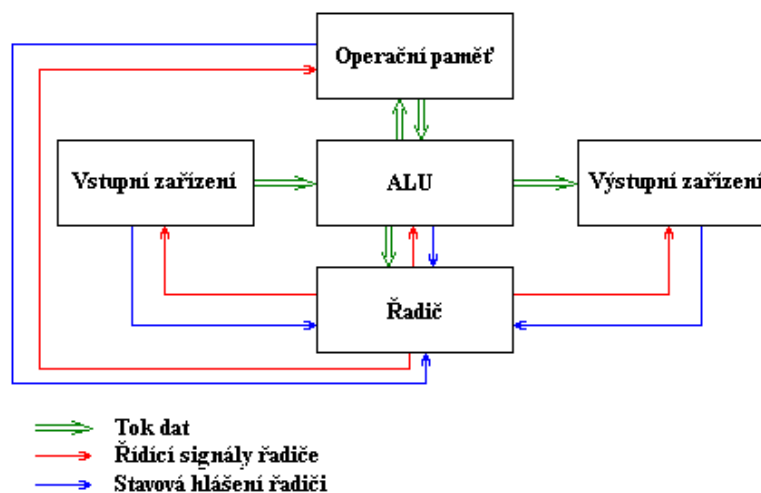


#### 4. Von Neumanova architektura, Harvardská architektura počítače, architektura RISC a CISC, taxonomie sběrnic, paralelní, sériový, synchronní, asynchronní přenos dat.

HARDWARE A APLIKAČNÍ SOFTWARE

---

##### Von Neumanova architektura



**Vstupy/Výstupy** – periferie

**ALU** – kombinační obvod, nejdůležitější operace

**Operacní paměť (OP)** – paměťový blok, Bytově Adresovatelná

**Řadič** – sekvenční obvod, generuje řídicí signály (**ŘS**) pro ostatní bloky, řídí přenosy po sběrnicích

**Data** (operandy ALU) jsou uložena rovněž v OP, z výpisu OP nelze jednoznačně určit co jsou instrukce a co data.

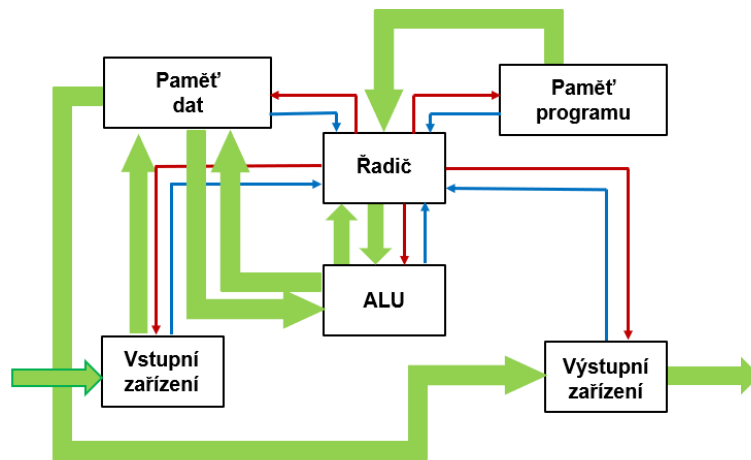
NOP (0x0), černý bod obrázku (0x0)

Počítač čte instrukce z OP, řadič je dekóduje (rozpozná co se má udělat) a generuje příslušné ŘS.

Stavy jednotlivých bloků jsou zasílány zpět řadiči pomocí stavových hlášení.

Instrukce a data jsou uložena v operační paměti. Nelze odlišit data a instrukce. Počítač může zpracovat data jako instrukci, a naopak instrukci jako data.

## Harvardská architektura



**Paměť programu** – typ flash, instrukce programu (a také konstanty v programu), jsou uchovány i v době vypnutí

**Paměť dat** – typ statická RAM, data se ztratí vypnutím

Každá paměť může mít jinou velikost nejmenší adresovací jednotky.

Program nemůže přepsat sám sebe.

Dvě sběrnice umožňují jednoduchý paralelismus, kdy lze přistupovat pro instrukce i data současně.

Současné mikrokontrolery, program je typicky uložen v paměti programu typu FLASH.

## Architektura RISC a CISC

### RISC

= Redukovaná instrukční sada / Reduced Instruction Set Computer (podle zásady „V jednoduchosti je síla“)

Využití standardizovaného OS (UNIX, Linux) umožnilo využít rozdílné architektury a jejich rozvoj, celkové rozšíření a zlevnění.

Tyto změny umožnili využití architektury RISC (r. 1980) – využití paralelního zpracování instrukcí (zřetězení, efektivnější organizace paměti cache).

Otevřená instrukční sada RISC-V je vyvíjena od roku 2010 na Kalifornské univerzitě v Berkeley. Navržen pro široké použití od vestavěných systémů přes mobilní telefony až po cloudové počítače (s důrazem na výkon i na spotřebu).

### Instrukce

- Jen jednoduché
- Typicky kódovány stejným počtem bitů
- Typicky vykonány v jednom, nebo několika málo taktech hodinového signálu
- Instrukce mají pevnou délku a jednotný formát který vymezuje význam jednotlivých bitů.
- Je použit vyšší počet registrů propojených přímo s ALU. Například nemůžeme násobit jednotlivé buňky paměti, ale základními instrukcemi si postupně obsah paměti přesunout do registrů, v registrech provést násobení a poté výsledek zapsat na požadovanou adresu

## Pozitiva

- Jednoduchost – malé množství instrukcí
- Jednoduchý dekodér instrukcí => rychlé dekódování instrukcí
- Umožňuje proudové zpracování instrukcí
- Rychlý obvodový řadič

ATmega328 - Arduino

## CISC

= počítač s rozsáhlým souborem instrukcí/Complex Instruction Set Computer

### Instrukční sada obsahuje

- Složité instrukce (např. kopírování bloku dat paměti) i jednoduché instrukce
- Instrukcí je hodně
- Typicky různá délka instrukcí (co do zakódování, tak i trvání)

### Původní snaha

- urychlit vykonávání instrukcí realizovat stále složitější instrukce pro jejich
- samostatné provedení (přesun bloku paměti). Složitější instrukce ale nejsou používány tak často.

## Pozitiva

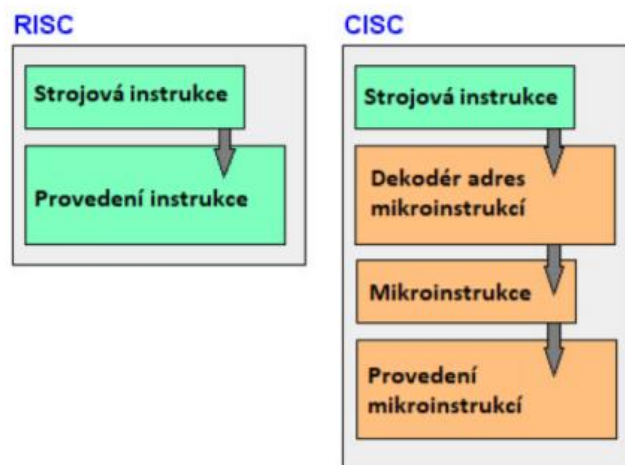
- Snížená četnost načítání instrukcí (snaha načíst jenom jednu, zbytek provede mikrořadič)
- Možnost vícenásobného využití funkčních jednotek v různých fázích vykonání instrukce
- Přítomnost mikroprogramového řadiče dává možnost změnit instrukční repertoár

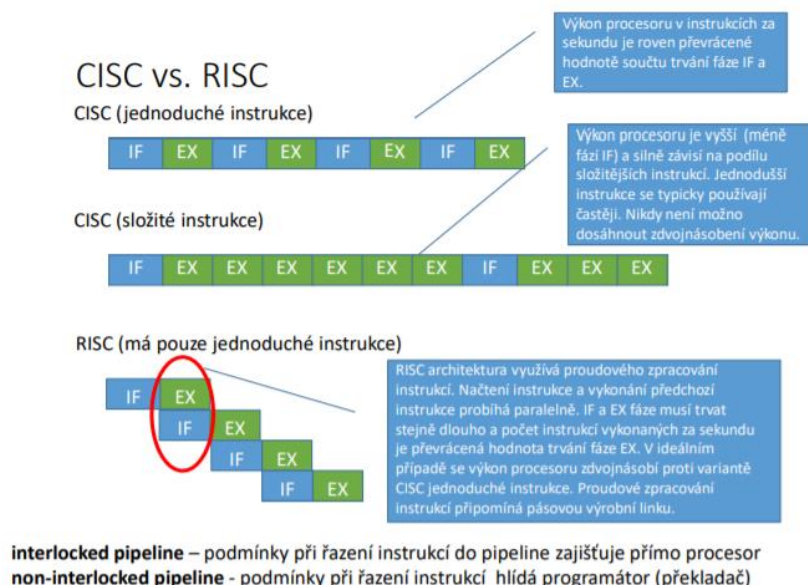
## Negativa

- Složité instrukce jsou specializované, nutnost různých variant, aby skládačka byla úplná
- Velký počet instrukcí => složitý dekodér instrukcí => dekódování jednoduchých a obvykle
- nejčtetnějších instrukcí (např. sčítání) trvá dlouho
- Nutnost mikroprogramovatelných řadičů
- Instrukce typicky trvají různě dlouho, těžko se zavádí proudové zpracování

CISC - 8051

## Porovnání





U mobilních aplikací se využívají architektury RISC, kde dominuje architektura ARM (Advanced RISC Machine), AVR (Alf and Vegard's RISC procesor) a MIPS (Microprocessor without Interlocked Pipeline Stages).

	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

## Taxonomie sběrnic

### Účelu:

#### Adresová sběrnice

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

#### Datová sběrnice

- 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 sběrnice

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

#### Systémová sběrnice

- Sběrnice pro přenos dat mezi procesorem, pamětí, periferiemi.
- Typicky zahrnuje adresovou, datovou a řídicí sběrnici, ale může se jednat i jednu sběrnici (např. PCI) jejichž protokol.

- Implementuje přenos adresy, dat a realizaci čtecích a zápisových cyklů do paměti a periférií.
- 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**

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

#### **Procesorová sběrnice**

- HyperTransport, DMI

#### **Paměťová sběrnice**

- Sběrnice mezi dříve severním můstkem a pamětí, dnes spíše mezi procesorem a pamětí (zahrnuje přenos adresy a dat).

#### **Synchronizace:**

- Náběžný hrana clock
- Sestupná hrana clock

#### **Směru přenosu dat:**

##### **Jednosměrný**

- Typicky adresová sběrnice.
- 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).

##### **Obousměrná**

- Přenos jedním a druhým směrem se multiplexuje v čase.
- Přenos jedním směrem nemůže probíhat současně s přenosem v druhém směru (přenosy 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.

#### **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 nestejných délek meandry).

##### **Sériový**

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

##### **Synchronní**

- Příjímáč i vysílač jsou řízeny zdrojem stejného hodinového signálu.

**Asynchronní**

- Synchronizace přijímače a vysílače se stále obnovuje při přenosu každého slova.
- Pomalejší přenos.
- Musí být zajištěno normalizované propojení obou zařízení.
- Rychlost přenosu dat je proměnlivá.