

13. Jednočipové počítače, mikrokontrolery

Created	@April 26, 2025 7:11 PM
Tags	Done
Kdo vypracoval	Luky

- PC (procesor computer)
- (Jednočipové počítače – 1 čip)

Počítač

- **CPU (central processor unit):** Hlavní procesor, který vykonává instrukce a provádí výpočty.
 - Jakýsi mozek počítače, řídí jeho ostatní části a jejich podíl na společném výpočetním díle celého systému. Hlavními výrobci procesorů do počítačů jsou společnosti intel a AMD.
 - Základní stavební jednotkou procesoru je tranzistor zapojený jako spínač. Jeho úkolem je měnit stav mezi logickou nulou a jedničkou. Jeden samotný tranzistor by příliš výpočtů realizovat nemohl, proto jich v dnešních procesorech nalezneme miliardy, díky čemuž je možné provádět i nejsložitější operace.
 - Interval mezi dvěma pulsy z oscilátoru se nazývá frekvenční cyklus. V takovém čase je procesor schopen své tranzistory přepnout do jedničky a poté zpět do nuly. O změně stavu z nuly do jedničky referujeme jako o náběžné hraně a o změně jedničky na nulu jako o hraně sestupné.
- **RAM (random acces memory):** Elektronická polovodičová dočasná paměť, která uchovává data a instrukce pro aktuální operace/programy. (zjednodušeně číslované buňky z nichž každá obsahuje hodnotu)
 - po odpojení se maže
 - počet zápisu a čtení není omezený
 - **volatilní** – při vypnutí napájení se informace smaže; takto se chovají polovodičové paměti RAM
 - **nevolutilní** – informace vydrží vypnutí napájení; tuto vlastnost mají magnetické paměti
- **ROM (read only memory):** Trvalá paměť pro základní systémová data (např. BIOS). Její obsah je dán už při výrobě a není závislý na napájení (nemaže se po odpojení napájení)
 - pro uložení firmwaru (softwaru)
 - dříve více častý, dnes už se používá jen zřídka (z důvodu nemožnosti aktualizace softwaru, který obsaoval a taky protože RAMky jsou dnes mnohem rychleší)

- **GPU (graphics processing unit)**: Grafická jednotka pro zpracování obrazu, promítnutí do monitoru, může být ale využit i jako k jiným výpočtům (kryptoměny)
 - **NPU: (Neural Processing Unit)** je specializovaný procesor navržený k urychlení výpočtů spojených s umělou inteligencí (AI), zejména pro zpracování neuronových sítí a úloh strojového učení. (může se skládat z více CPU a GPU)
 - **I/O (input/output)**: Výměna dat mezi pc a externími zařízeními (myš, monitor, disk, klávesnice...)
-

Jak funguje procesor:

1. Načtení instrukce

Procesor načte instrukci z paměti podle adresy uložené v adresovém registru.

2. Dekódování instrukce

Řídicí jednotka zjistí, co instrukce znamená a jaký typ operace se má vykonat.

3. Vykonání instrukce

Podle typu instrukce se provede jedna nebo více následujících činností:

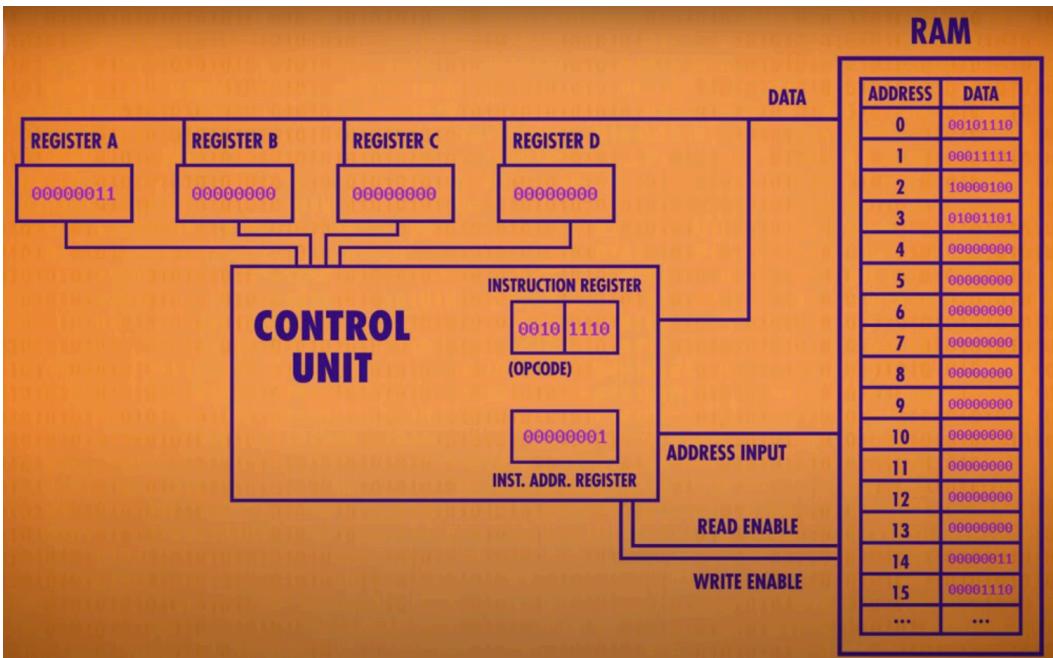
- **Přenos dat** – například mezi registry nebo mezi pamětí a registrem.
- **Aritmetická operace** – například sčítání, odčítání.
- **Logická operace** – například porovnání hodnot, AND, OR, NOT.
- **Ovládání toku programu** – podmíněné nebo nepodmíněné skoky, smyčky.

4. Zvýšení adresového registru

Po dokončení instrukce se zvýší hodnota adresového registru, aby ukazoval na další instrukci.

5. Opakování celého cyklu

- **Registr** – malá, rychlá paměť pro dočasně uchování dat.
- **Instrukční registr** – uchovává právě vykonávanou instrukci.
- **Adresový registr** – uchovává adresu v paměti, ze které se čte/zapisuje.
- **Aritmeticko-logická jednotka (ALU)** – provádí výpočty a logické operace.

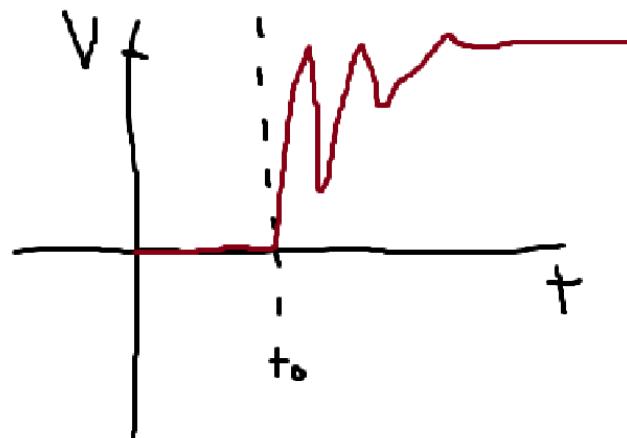


Připojení procesoru k napájení

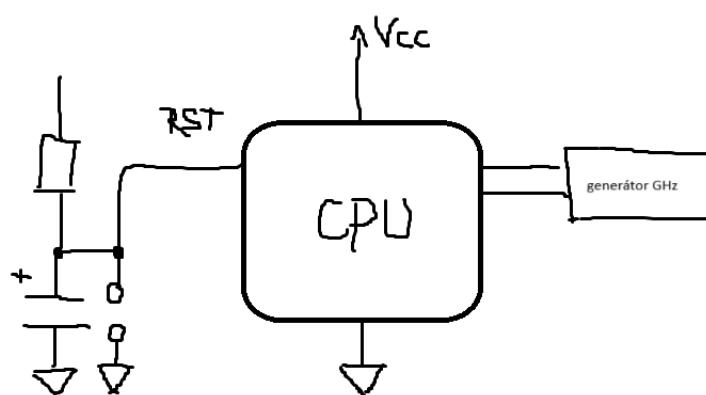
Při zapnutí napájení mikrokontroléru (např. raspberka nebo arduina) nemusí být napětí (Vcc) ihned stabilní – často dochází ke kmitání a přechodovým jevům. Pokud by se procesor spustil hned, mohlo by dojít k nesprávné inicializaci nebo poškození. Proto se používá resetovací pin (RST), který zajistí, že se procesor spustí až tehdy, když je napětí stabilní.

Zapojuje se kondenzátor a rezistor, kde je čas tau jejich součin a k uklidnění dojde po 3 až 5 tau. (časová hodnota přechodového jevu).

- Vstup generátoru hodin
- Vcc a GND
- Vystup Reset pin - Pin, který obstarává správné zapnutí procesoru i při zákmitech v připojení Vcc
 - Časová konstanta $T = R*C$ (Tau = odpor * farady)
 - Vyjadřuje čas, který bude CPU čekat než se zapne



Zákmity na V_{cc} při připojení



Von Neumann architektura

- Společná paměť pro data a instrukce

Výhody:

- Jednodušší architektura
- Flexibilnější programování
- Možnost modifikace programu za běhu

Nevýhody

- Rychlosť zpracování instrukcí je podstatně vyšší než komunikace s pamětí → tzv. *memory wall*

Harvardská architektura

- Oddělená paměť pro data a instrukce

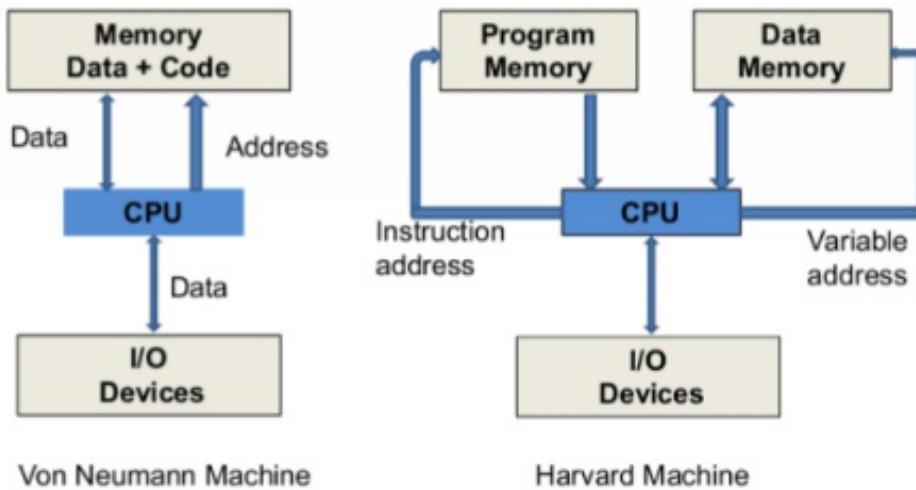
Výhody

- Vyšší bezpečnost (nelze snadno modifikovat kód)
- Možnost současného přístupu k datům a instrukcím
- Rychlejší zpracování díky paralelnímu přístupu

Nevýhody

- Vyžaduje dvě samostatné paměťové cesty
- Paměť určená pro instrukce nemůže být využita pro data a naopak, což může vést k nevyužitému prostoru.

Hlavní rozdíl spočívá v organizaci paměti a způsobu přístupu k datům a instrukcím, což ovlivňuje výkon, bezpečnost a flexibilitu systému.



Instrukční sady:

- **CISC (Complex Instruction Set Computer):** Nabízí širokou škálu komplexních instrukcí, které mohou provádět složité operace, ale s nižší rychlostí. (levnější)
 - používá se v zařízeních kde je potřeba velký výpočetní výkon (stolní pc, servery...)
- **RISC (Reduced Instruction Set Computer):** Poskytuje menší počet jednodušších instrukcí, které mohou být prováděny rychleji a efektivněji. (dražší)
 - díky své nízké spotřebě energie se používá u telefonů

Obvyklou chybou je domněnka, že procesory CISC mají více strojových instrukcí, než procesory RISC. Ve skutečnosti nejde o absolutní počet, ale o počet různých druhů operací, které procesor sám přímo umí vykonat na hardwarové úrovni (tj. již z výroby). Procesor CISC tak může například paradoxně obsahovat pouze jednu strojovou instrukci pro danou operaci (logické op.) zatímco procesor RISC může tuto operaci obsahovat jako několik strojových instrukcí, které stejnou operaci umí provést nad různými registry.

Jak funguje procesor (MOS 6502)

16-bit adresní prostor

(Data rozdělena na 2 na 16 Byte, tedy po 4 v šestnáctkové soustavě. Po resetu se podívá na adresy FFFC a FFFD. Z každé vezme 8 bit, získaná data dá do registru x nebo y, které má na práci a dá je na adresní piny A0 až A15 a tam je podívá, najde další bit, třeba ze samých 0 a od něj jede dále. Procesor má příkazy na 3 Byte, tak má příkaz třeba 5 bit a zbylá tři označí důležitou hodnotu (ACCUMULÁTOR, registr x/y), pomocí dalších dvou Bytů dá data, která tam chce nahrát.)

- Důležité adresy:

- FFFD a FFFC - po resetu se podívá na adresu a uloží **bootovací vektor**
- **Bootovací vektor** (reset vector) = výchozí **místo**, kde CPU hledá **nové instrukce**
- Akumulátor = drží data, se kterými pracuje procesor
- Programový čítač ví kde je **bootovací vektor** (je přímo napojený na paměť)
 - Vezme **první půlku** bootovacího vektoru a narve je do **x**
 - Vezme **druhou půlku** bootovacího vektoru a narve ji do **y**
 - Prohodí se **x** a **y** a načtou se
 - Narve je do programového čítače (i pro druhou adresu)
- Celkový **počet základních instrukcí je malý kvůli variacím v posledních číslech adresy**

[Video na téma](#)

<https://cs.wikipedia.org/wiki/ROM>, <https://cs.wikipedia.org/wiki/RAM>,
https://cs.wikipedia.org/wiki/Centr%C3%A1ln%C3%AD_procesorov%C3%A1_jednotka,
<https://cs.wikipedia.org/wiki/GPU>, <https://www.ibm.com/think/topics/neural-processing-unit>,
<https://cs.wikipedia.org/wiki/Vstup/v%C3%BDstup>,
https://cs.wikipedia.org/wiki/Von_Neumannova_architektura,
https://cs.wikipedia.org/wiki/Harvardsk%C3%A1_architektura,
https://cs.wikipedia.org/wiki/Jedno%C4%8Dipov%C3%BD_po%C4%8D%C3%ADta%C4%8D

[Jirka \(1\)](#)

[Martin \(Staré\) \(1\)](#)

[Merge vereze done](#)