**BI-SPOL-28 Architektura číslicového počítače, instrukční cyklus počítače, základní třídy souborů instrukcí (ISA). Paměťový subsystém počítače, paměťová hierarchie, cache**

BI-SAP

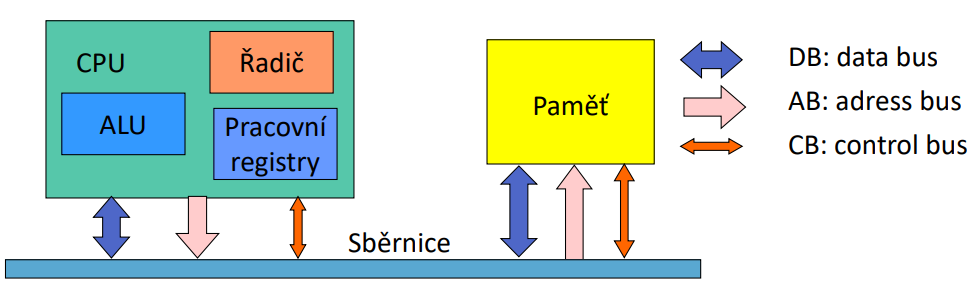
### Architektura číslicového počítače

**Základní části počítače**:

* CPU (ALU + řadič)
* vstupní a výstupní zařízení
* hlavní paměť (často mimo procesor)

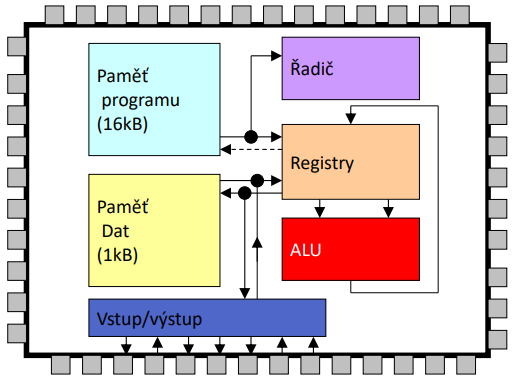
**Procesor (CPU)** musí obsahovat:

* aritmeticko-logickou jednotku (ALU)
* řadič (řídí všechny jednotky, vysílá jim řídící signály po CB)
* registry (pro dočasné uchování dat, instrukcí a pomocných proměnných)



**Von Neumannova architektura**

* paměť společná pro instrukce i data



**Harwardská architektura**

* využívanější (používá se v menších počítačích současnosti)
* paměť pro data a instrukce je oddělená

### Instrukční cyklus počítače

**Instrukce**

* příkaz zakǒdovaný jako číslo
* co se má provést
* s čím (operandy)
* kam se má uložit výsledek
* kde se má pokračovat
* nejnižší úroveň se kterou může programátor pracovat
* aritmetické (ADD, SUB), řídící (JMP), uložení do paměti..

Činnost procesoru: sekvenční zpracování instrukcí v nekonečné smyčce podle instrukčního cyklu.

**Dělení instrukcí** (podle počtu adres instrukcí):



* 1 adresová (ADD R1)
* 2 adresová (ADD R1, R2)
  + výsledek se ukládá na místo 1. operandu
* 3 adresová (ADD R1, R2, R3)

### Základní třídy souborů instrukcí (ISA)

**ISA ( = Instruction Set Architecture)**

* zahrnuje: typy a formáty instrukcí, instrukční soubor

datové typy, kódovaní a reprezentace, způsob uložení dat v paměti

módy adresování paměti, přístup do paměti, mimořádné stavy

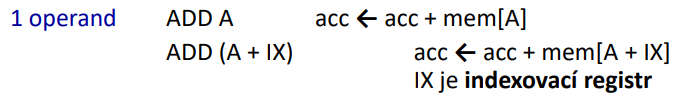
* umožňuje definici rozhraní mezi HW a SW
* standardizuje instrukce, bitové vzory strojového jazyka
* abstrakce (výhoda – různé implementace stejné architektury)
* obecný popis organizačních, funkčních a provozních principů procesoru

**Adresace operandů**

* přímá: práce přímo s registrem
* nepřímá: v registru/paměti je adresa ma data, se kterými se pracuje
* relativní: offset od ručité adresy (v registru nebo přímo)
* indexová: báze + offset
  + autoinkrementace + autodekrementace

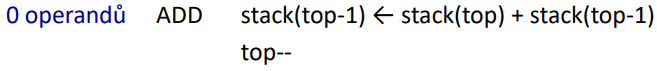
**Střadačově (akumulátorově) orientovaná ISA**

* z kalkulaček (nejstarší ISA)
* Implicitním operandem ALU je vždy střadač (registr)
* jednuduchý HW, rychlé přepínání kontextu
* omezený paralelismus, častá komunikace s pamětí



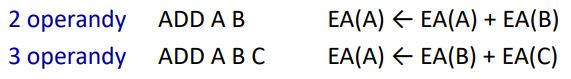
**Zásobníkově orientovaná ISA**

* využití HW zásobníku při vykonávání programu
* jednoduchá a efektivní adresace operandů a dekódování instrukcí
* zásobník je sekvenční (omezený paralelismus) a nelze přistupovat náhodně k datům



**Registrová ISA** (GPR)

* dnes převládá
* registry jsou rychlejší než paměť (včetně cache)
* lze přistupovat náhodně k datům (ne jako u zásobníku)
* omezený počet registrů



### Paměťový subsystém

Zařízení obsahují různé druhy pamětí, liší se rychlostí, přístupem, uchováváním informací atd.

Nejčastěji jsou využívaná data v M1 (primárním cache)

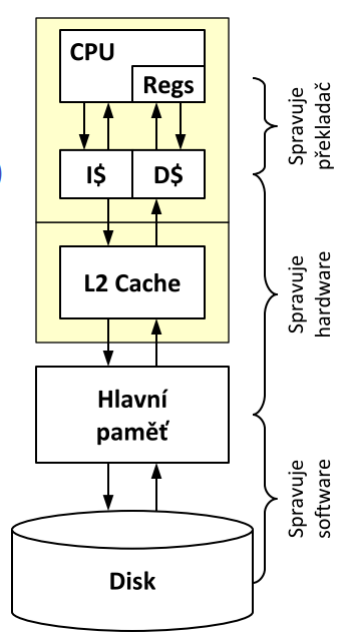
**Paměť**

Vztah k napájení

- Volatilní: uložená paměť zaniká po vypnutí napájení (RAM)

- Non-volatilní: obsah zůstane i po vypnutí napájení (HDD)

Fyzikální přístup: magnetická, optická a další.



**Paměťová hierarchie**

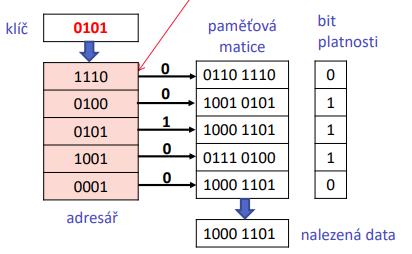
* *Registr*: rychlý, drahý, nejblíž ALU, protože ho přímo využívá (klopné obvody)
* *Cache (skrytá paměť)*: rychlé, dražší, kapacitně menší (~kB), blíž k procesoru
  + *primární*: oddělená, instrukční a datová (I$, D$)
  + *sekundární* (L2): nejlépe na čipu, určitě v pouzdře (SRAM)
* *Hlavní paměť*: pomalejší, levnější, větší (DRAM, ~ GB)
* *Vnější paměť*: ještě pomalejší, ale velká kapacita, soubory swap, magnetické disky, flash paměti

**Cache**

* většinou více vrstev (L1, L2…)
* "malá" rychlá paměť zařazená mezi procesor a hlavní paměť
* využití asociativního přístupu k položkám (CAM)
* obsahuje kopie nejčastěji používaných položek hlavní paměti (princip lokality)
* možné realizovat RAM pamětí

**Plně asociativní paměť**

* adresuje se částí datové položky, která se má vyhledat, tzv. klíčem
* na rozdíl od adresovatelné paměti (např. SRAM) neobsahuje adresový dekodér, ale adresář



**Čtení z cache**

* *cache hit*: data jsou nalezena
  + *hit rate*: poměr cache hit a počet všech dotazů
  + *hit time*: doba nalezení údajů v cache a předání procesoru
* *cache miss*: výpadek cache (je třeba načtení z nižší úrovně)
  + *miss rate*: četnost výpadků cache (Miss rate = 1 – Hit rate)
  + *miss penalty*: doba potřebná k získání dat z pomalejší paměti

**Zápis do cache**

* pokud položka v cache není, zapíše se jen do hlavní paměti
* *průběžný* zápis: nová hodnota se zapíše se zároveň do cache i HP
* *odložený zápis*: zapíše se do cache a při uvolnění z cache se musí zapsat do HP

**Asociativní paměť s omezeným stupněm asociativity**

* nevýhoda Plně asociativní paměti je velikost
* každé položce je určeno místo (nebo několik míst podle stupně asociativity), kde se může nacházet. Toto místo je určené částí adresy položky

**Organizace paměťového subsystému**

