

M10 Základní cyklus počítače

#technicke_vybaveni_pocitacu

- obvykle se nazývá "fetch-decode-**execute** cycle"
- popisuje zk. kroky opakované při každé instrukci
- architektura RISC má omezený počet instrukcí na základní a jednoduché

1. čtení (*fetch*)

- instrukce se načtou z paměti
- adresa instrukce k provedení je uložena v registru Program Counter (*PC*)
- instrukce se načte z adresy z PC do Instruction Register (*IR*)
- dojde k aktualizaci PC aby ukazovala na další instrukci v paměti

2. dekódování (*decode*)

- načtená informace (*v IR*) obsahuje operační kód (*opcode*) a další informace
- během dekódování jsou jednotlivé části instrukce identifikovány pro další zpracování
- pomocí opcode je určeno, jakou operaci instrukce představuje
- pokud instrukce potřebuje operandy, dekódování je identifikuje a připraví k použití

3. provedení (**execute**)

- vykonává operaci definovanou dekódovanou instrukcí (např.: aritmetické operace, logické operace, přesuny dat, skoky nebo další)
- po provedení instrukce se aktualizují stavové registry obsahující informace o procesoru (např. přetečení)
- výsledky operací jsou zapsány do registrů nebo do paměti

4. zásah do paměti

- umožňuje procesoru přistoupit k paměti
- adresa je buď předem určena nebo je vypočtena na základě aktuálních hodnot v registrech; adresa je následně odeslána na paměťovou sběrnici
- hodnota z interních (*pipeline*) registrů je zapsána na adresu
- po zápisu se aktualizují stavové bity (*bity ovlivňující následující průběh programu*)

Výjimečné stavy při běhu CPU

- stavy které mohou vyžadovat speciální pozornost či manipulaci
- přerušení (*interrupt*)
 - přerušují běžný tok programu
 - vyžadují okamžitou pozornost procesoru
 - vyvolána externím zařízením, chybou programu nebo samotným programem úmyslně
 - pro obsluhu přerušení musí procesor přepnout kontext a reagovat na příslušné události
- výjimky (*exception*)
 - podobné přerušení
 - jedná se o chybový stav
 - vyžadují zvláštní opatření
 - procesor musí přepnout na obsluhu výjimky a přijmout opatření k řešení problému
 - např.: dělení nulou, přetečení při aritmetických operacích nebo přístup k neplatné paměti
- přepínání kontextu (*context switch*)
 - nastává když běžící proces na CPU je pozastaven a CPU přepíná svůj kontext na jiný proces
 - informace (registry, program counter, adt.) pozastaveného procesu jsou uloženy do paměti
 - informace jsou následně aktualizovány aby odpovídaly novému procesu
- stav úspory energie (*halt*)
 - nastává když CPU přechází do režimu nízké spotřeby nebo je dočasně zastaven
 - neprovádí žádné instrukce a čeká na další pokyny
- bezpečnostní režim (*privileged mode*)

- CPU má vyšší úroveň oprávnění než v normálním uživatelském režimu
- má přístup k systémovým zdrojům (speciální registry, instrukce nebo přímý přístup k hardwaru)
- chyby přístupu do paměti (*memory access violations*)
 - program přistoupí k neplatné paměti nebo s ní provede nepovolenou operaci
 - např. pokus o čtení nebo zápis do neexistující adresy paměti

Formát instrukce

- AVR používá specifický formát založený na RISC formátu
- instrukce
 - typu R (*-register*)
 - pracuje s registry
 - provádí aritmetické nebo logické operace
 - `opcode | Rd | Rr`
 - `opcode` → identifikační číslo operace
 - `Rd` → cílový registr (a zároveň zdrojový)
 - `Rr` → zdrojový registr
 - typu I (*-immediate*)
 - pracuje s konstantami
 - konstanty jsou uloženy přímo v instrukci
 - `opcode | Rd | K`
 - `K` → konstanta
 - typu J (*-jump*)
 - pro řízení toku programu
 - `opcode | d`
 - `d` → adresa destinace skoku

Operační

jednotka

- aritmetické a logické operace provádí odděleně od těch ostatních
- instrukce jsou navrženy aby byly jednoduché; vyžadují minimální počet cyklů pro provedení
- každá instrukce provádí pouze jednu konkrétní operaci
- maximalizuje paralelní zpracování; navrženy pro nezávislé vykonávání vedoucí k rychlejšímu výpočtu
- pracuje pouze s daty v registrech; pokud data nejsou v registru, zapíše je
- pevně definovaná šířka slova

znak

- základní strojový kód říkající procesoru, jakou operaci má provést
- "operátor"
 - v kontextu programování se jedná o symbol či zkratku reprezentující operaci (např.: `ADD` - sčítání; `SUB` - odčítání; `AND` - logický AND; `OR` - logický OR; `MOV` - přesun dat; `JMP` - nepodmíněný skok)
 - operátory jsou následně převedeny do binární podoby
 - operátory závisí na konkrétním RISC procesoru

Pipeline registr

- speciální druh registrů používaný pro pipelining
- pipelining - technika umožňující provádění několika fází instrukce současně
- každá fáze má svůj registr; slouží k uložení mezikroků

Řadič

- má za úkol dekodovat instrukce z paměti
- připravuje interní obvody pro provedení instrukcí
- řídí takt procesoru; určuje prioritu provedení instrukce
- spravuje přístup čtení a zápisu do paměti
- řídí tok dat v datových cestách
- detekuje stav výjimek a přerušení; popřípadě dokáže vyřešit neobvyklé stavy.
- provádí skoky a podmíněné instrukce

Dekodér

-