

## 28 Struktury a hierarchie pamětí.

Způsoby adresace. Různá šíře adres generovaných CPU (logických adres) a fyzických adres paměti. Mapování, stránkování, segmentace. Přerušování a výjimky. Zdroje přerušování, přerušovací vektory. DMA přenosy. (A0B35SPS)

(pozn. autora: V předmětu SPS toto není. Čerpal jsem z přednášek OSS a APO.)

### 28.1 Struktury a hierarchie pamětí

- Rychlé a malé paměti jsou umístěny blízko CPU.
- Větší a pomalejší paměti jsou dále od CPU.
- Princip cachování nejčastěji používaných dat v rychlých pamětech - zrychlení přístupu k datům.
- od nejrychlejší a nejmenší po největší a nejpomalejší:
  - on-chip L1 caches
  - off-chip L2 caches (SRAM)
  - hlavní paměť (DRAM)
  - vedlejší paměť (pevný disk)

### 28.2 Způsoby adresace

Máme 2 typy adres. Logickou a fyzickou adresu. Logická adresa je adresa, se kterou pracuje CPU. Tato adresa je překládána na fyzickou adresu, což je konkrétní adresa v dané paměti. Výhody - CPU vidí unifikované adresy - pro přístup do hlavní paměti i k IO zařízením. Data v hlavní paměti nejsou poskládána tak jak jdou za sebou, mohou být

zpřeházena - optimalizace využití prostoru paměti a fragmentace. Přes logickou adresu se nechá přistupovat k datům sekvenčně, tak jak jdou za sebou.

Velikost logického adresního prostoru je dána architekturou procesoru - kolik adres je procesor schopen generovat. Např. 32 bitový procesor generuje  $2^{32}$  adres. Velikost fyzického adresního prostoru je dána nainstalovaným hardware počítače - fyzicky dostupnou pamětí.

**Stránkování** Souvislý LAP (Logický Adresní Prostor) není zobrazován jako jediná souvislá oblast FAP. FAP se dělí na úseky zvané rámce, LAP se dělí na úseky dané stránky.

**Struktura logické adresy** Logická adresa se skládá ze dvou částí. První část je index v tabulce stránek - díky němu se v tabulce stránek vyhledá fyzická adresa stránky. Druhá část je offset - posunutí ve stránce.

#### Převod logické adresy na fyzickou

- K logické adrese je v tabulce stránek nalezena fyzická adresa dané stránky/rámce.
- K adrese rámce je přičten offset a výsledek je požadovaná fyzická adresa.

### 28.3 Různá šíře logických a fyzických adres

CPU může mít menší datovou sběrnici, než je šíře fyzické adresy. Např. fyzická adresa má 20 bitů, ale datová sběrnice CPU je jen 16 bitů. Zbývající 4 bity jsou offset segmentu a jsou uloženy ve 4 segmentových registrech CPU.  $FA = (segment \ll 4) + \text{offset}$ .

Opačný případ - CPU generuje více logických adres (např. 64 bitový procesor generuje  $2^{64}$  adres) než je fyzických adres (LA je větší než FA). Tento problém se řeší virtualizací paměti. Virtualizace spočívá v tom, že FAP se rozšiřuje o úseky na vnější paměti (např. na pevném disku).

### 28.4 Mapování, stránkování, segmentace

**Stránkování** LAP je rozdělen do úseků, které na sebe navazují - stránky. Logická adresa odkazuje na adresu stránky, která odkazuje na adresu rámce FAP. Rámce v FAP na sebe nemusí navazovat.

**Segmentace** LAP je rozdělena na segmenty. Segmenty jsou části programu - mají logický význam (hlavní program, procedura, funkce,...), jsou různě dlouhé - nízká vnitřní fragmentace. Výhody - lze určit přístup do nepovolené části paměti (segmentation fault), se segmenty v paměti lze libovolně hýbat, lze nastavovat práva k přístupu do segmentu. Nevýhoda - externí fragmentace.

**Stránkování a segmentace najednou** Výše uvedené metody lze kombinovat. Segmentace vybírá části LAP, stránkování zobrazuje LAP do FAP - LAP je dělena na segmenty, které jsou stránkovány.

## 28.5 Přerušování a výjimky

### Přerušování

- cílem je zlepšení účinnosti systému
- je potřeba provést jinou posloupnost příkazů jako reakci na nějakou „neobvyklou“ událost
- přerušující událost způsobí, že se pozastaví běh procesu v CPU takovým způsobem, aby ho bylo možné později znovu obnovit, aniž by to přerušovaný proces „poznal“
- využití např. při IO operacích
- testování, zda je voláno přerušování alespoň po dokončení každé instrukce
- přerušování bývá často voláno programem
- Maskovatelná, lze je zakázat v stavovém řídicím slovu CPU, případně řízení priorit (periferie, čítače, časovače)
- Nemaskovatelná - ošetření HW chyb, hlídací obvod (Watch Dog)

### Výjimka

- Výjimka – ošetření zvláštních situací, které brání dalšímu vykonávání instrukcí (exception)
  - Matematické přetečení (výsledek instrukce s kontrolou saturace přetekl)
  - Načtena nedefinovaná instrukce (neznámý operační kód instrukce typu IR, nebo neznámá funkce instrukce typu R)
  - Systémové volání (instrukce syscall)

## Zdroje přerušování, přerušovací vektory

**Určení zdroje výjimky přerušování** Softwarové vyhledání (polled exception handling)

- Veškerá přerušování a výjimky spouštějí rutinu od stejné adresy – např. standardní MIPS, adresa 0x00000004
- Rutina zjistí důvod ze stavového registru (MIPS: cause registr)

Vektorová obsluha přerušování

- Již hardware CPU zjistí příčinu/číslo zdroje
- V paměti se nachází na pevné/řídícím registrem specifikované (VBR) adrese tabulka vektorů přerušení
- Procesor převede číslo zdroje na index do tabulky
- Z daného indexu načte slovo a vloží ho do PC ●

Nevektorová obsluha více pevně určených adres podle priorit/důvodu

- Často jsou přístupy kombinované, např. výjimky mají oddělené cílové adresy skoků, využívá tabulku atd. ale veškerá vnější přerušení končí pouze na jednom z vektorů

## 28.6 DMA přenosy

Direct Memory Access - přímý přístup do paměti. Využívá se při přenosu velkého množství dat - data nemusí jít přes procesor a nevytěšňují tak data z cache.

- Program/OS nastaví parametry přenosu
- Procesor nastaví adresy do DMA řadiče, ten na konci přenosu vyvolá přerušení