Arquitetura de Computadores

Detalhes essenciais

- Instruções de vírgula flutuante e SIMD

1. Desempenho
2. Pipelining
3. Memória Cache
4. Periféricas

1. Desempenho

- Desempenho relativo | "Sistema A é n vezes mais rápido que B"

$$n = \frac{\text{desempenho } A}{\text{desempenho } B} = \frac{t_{exec\ B}}{t_{exec\ A}}$$

- $t_{CPU} = \frac{\text{nº de ciclos}}{f_{clock}}$
- $CPI = \frac{\text{nº de ciclos}}{N_{instr.}}$

$$t_{exec.} = N_{instr.} \times CPI \times \frac{1}{f_{clock}}$$

- Lei de Amdahl : Speedup

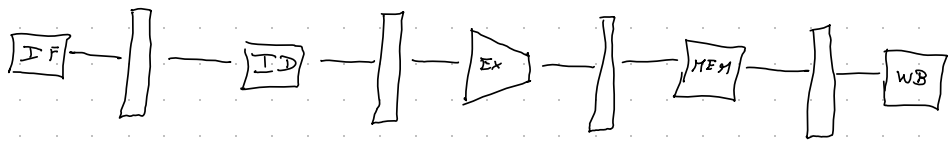
$$S_{global} = \frac{1}{\frac{p}{S_{parte}} + (1-p)}$$

- Benchmark : média geométrica

$$\sqrt[n]{\prod_{i=1}^{n} \left(\frac{t_{referência}}{t_{execução}} \right)}$$

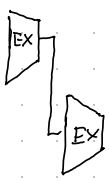
2. Pipelining

→ Andares



→ Atalhos

- Geral



- Loads

- Compilador
  Reordenar instruções de forma a evitar protelamentos
- Branches
  - Condição avaliada em MEM
  - Assume-se que o salto não é tomado ??

→ Registros

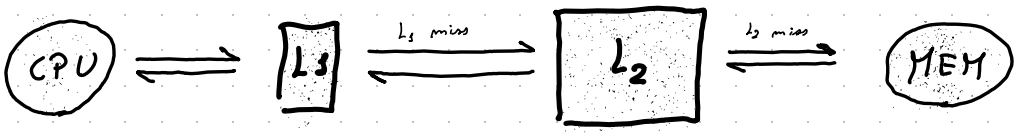
IF/ID | PC, instruction code

ID/Ex | PC, Read data 1, Read data 2, Write register, Imm
Control { ALUSrc, ALUOp, Branch, MemRead, MemWrite, MemtoReg, RegWri

Ex/MEM | Add sum, Zero, Read data 2, ALU result, Write register
Control { Branch, MemRead, MemWrite, MemtoReg, RegWrite }

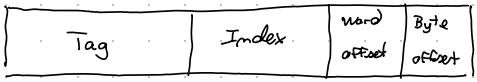
MEM/WB | Read data, Write register, Control { MemtoReg, RegWrite }

3. Memória Cache

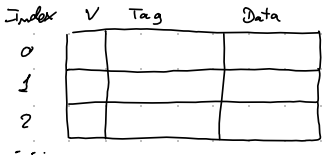


→ **Endereços**

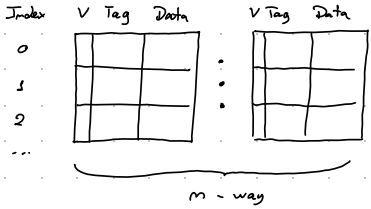
Tag | Restantes bits
Index | 2^m endereços → m bits
WO | 2^m words/bloco → m bits
BO | 2^2 bytes/word → 2 bits



→ **Mapeamento direto**



→ **Associatividade (n-way)**



→ **Write-through e No-allocate** (Simplicidade)

Leitura	Hit	Ler cache
	Miss	• Ler memória • Colocar na cache e atualizar tag e V=1
Escrita	Hit	Escrever em cache e memória
	Miss	Escrever na memória

→ **Write-back e alloc-on-miss** (Menos acessos a memória)

Ler cache
- (Se D=1) Escrever valor da cache na memória
- Ler memória • Colocar na cache e atualizar

Escrever em cache (D=1)
- (Se D=1) Escrever valor da cache na memória
- Escrever em cache e atualizar tag, V=1 e D=1

→ **Desempenho**

$$CPI_{stall} = \frac{Nº\ acessos\ a\ memória}{Nº\ instruções} \times taxa_{falhas} \times penalidade_{falhas}$$

$$t_{acesso_{MEM}} = t_{acerto} + taxa_{falhas} \times penalidade_{falhas}$$

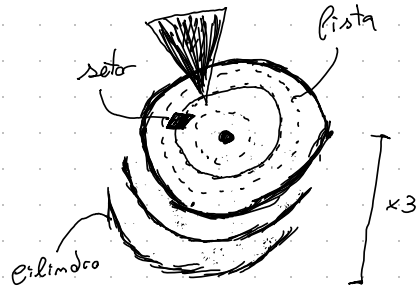
$penalidade_{falhas} \rightarrow \frac{t_{MEM}}{T}$

4. Periféricos

→ Disco Rígido

$$t_{acesso} = t_{procura} + t_{rotação} + t_{transferência} + t_{controlador}$$

$$t_{rotação} = \frac{1}{2} \cdot \frac{60}{RPM}$$



Capacidade / cilindro = Nº setores x Nº bytes / setor

→ Técnicas de E/S

Varrimento / Polling

CPU sonda a verificar se o dispositivo está pronto.

Interrupção

Dispositivo interrompe CPU quando necessário.

DMA : Acesso direto a memória

CPU pede ao DMA para tratar da tarefa e o DMA interrompe o CPU quando for completada.