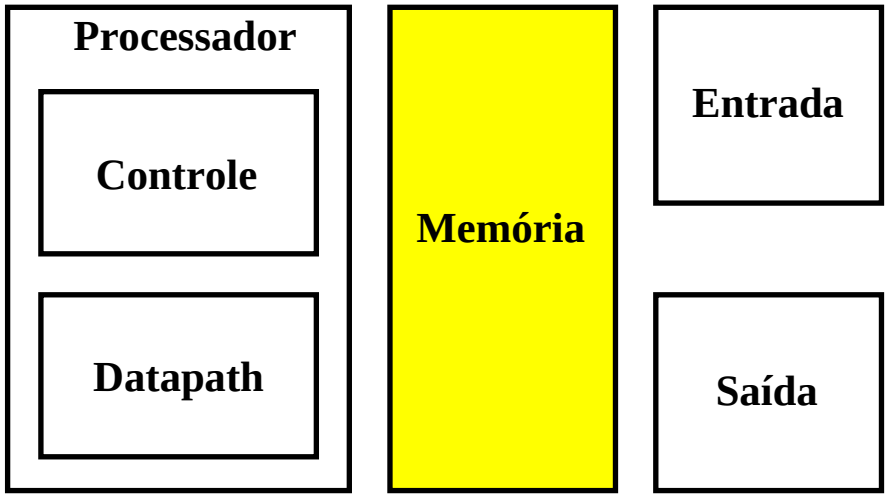
Cache: desempenho e associatividade



Desempenho da cache

- **Tempo de CPU:**

$$\text{tempo}_{\text{execução}} = \left(\text{ciclos}_{\text{CPU}} + \text{ciclos}_{\text{stall}}\right) \times \text{T}$$

- **Ciclos de "stall":**

$$\text{ciclos}_{\text{stall}} = \text{ciclos}_{\text{stall}}\,(\text{leitura}) + \text{ciclos}_{\text{stall}}\,(\text{escrita})$$

Desempenho da cache

- **Ciclos de "stall" devidos à leitura:**

$$\text{ciclos}_{\text{stall}}\,(\text{leitura}) = \frac{\text{leituras}}{\text{programa}} \times \text{mr}\,(\text{leitura}) \times \text{penalidade}\,(\text{leitura})$$

- **Ciclos de "stall" devidos à escrita:**

  (Para "write buffers" suficientemente grandes ou "write-back")

$$\text{ciclos}_{\text{stall}}\,(\text{escrita}) = \frac{\text{escritas}}{\text{programa}} \times \text{mr}\,(\text{escrita}) \times \text{penalidade}\,(\text{escrita})$$

$$+\,\text{ciclos}_{\text{stall}}\,(\text{write buffer})$$

Desempenho da cache

- **Combinando escrita e leitura**
  - **Supondo penalidades idênticas**
  - **Taxa de fracasso (mr) combinada**

$$\text{ciclos}_{\text{stall}}(\text{memória}) = \frac{\text{acessos}}{\text{programa}} \times \text{mr} \times \text{penalidade}$$

OU

$$\text{ciclos}_{\text{stall}}(\text{memória}) = \frac{\text{instruções}}{\text{programa}} \times \frac{\text{fracassos}}{\text{instrução}} \times \text{penalidade}$$

Cache: exemplo de impacto no CPI

- **Dado um programa, suponha**
  - **mr (I) = 2% e mr (D) = 4%**
  - **CPI = 2 para cache ideal (não gera “stalls”)**
  - **Penalidade = 100 ciclos**
  - **Loads + stores = 36% (SPECInt2000)**
- **Objetivo**
  - **Comparar o desempenho de duas configurações:**
    - » **CPU com cache ideal (mr=0)**
    - » **CPU com cache real (mr≠0)**

Comparação ideal x real

- **“Stalls”p/ fracasso no acesso a instruções:**

$$I \times 2\% \times 100 = 2 \times I$$

- **“Stalls” p/ fracasso no acesso a dados:**

$$(I \times 36\%) \times 4\% \times 100 = 1{,}44 \times I$$

- **CPI c/ “stalls”:**

$$CPI_{total} = 2 + 3{,}44 = 5{,}44$$

- **Razão dos tempos de execução:**

$$\frac{\text{tempo}_{\text{execução}}\ (\text{real})}{\text{tempo}_{\text{execução}}\ (\text{ideal})} = \frac{I \times CPI_{\text{real}} \times T}{I \times CPI_{\text{ideal}} \times T} = \frac{5{,}44}{2} = 2{,}72$$

Impacto com redução do CPI

- **O que aconteceria com a aceleração da CPU ?**
  - **Por exemplo: CPI = 2 → 1;**
  - **Sistema de memória permanece o mesmo**
- **CPI com "stalls":**

$$\mathrm{CPI}_{\mathrm{total}} = 1 + 3{,}44 = 4{,}44$$

- **Razão dos tempos de execução:**

$$\frac{\mathrm{tempo}_{\mathrm{execução}}\ (\mathrm{real})}{\mathrm{tempo}_{\mathrm{execução}}\ (\mathrm{ideal})} = \frac{\mathrm{I} \times \mathrm{CPI}_{\ \mathrm{real}} \times \mathrm{T}_{\mathrm{r}}}{\mathrm{I} \times \mathrm{CPI}_{\ \mathrm{ideal}} \times \mathrm{T}_{\mathrm{r}}} = \frac{4{,}44}{1} = 4{,}44$$

Comparação CPI = 2 → 1

- **Em relação à ideal:**
  - **2,72 mais lenta → 4,44 mais lenta**
- **Porcentagem do tempo gasto com "stalls":**

$$\frac{3,44}{5,44} = 63\% \quad \Rightarrow \quad \frac{3,44}{4,44} = 77\%$$

- **Conclusão:**
  - **Quanto menor o CPI, maior o impacto dos "stalls".**
- **Tendência:**
  - **Superescalares: CPI ↓**
- **Desempenho: compromisso entre pipeline e cache.**

Impacto com aumento de f

- **Dado um programa, suponha**
  - **mr (I) = 2% e mr (D) = 4%**
  - **CPI = 2 para cache ideal (não gera "stalls")**
  - **Loads + stores = 36% (SPECInt2000)**
  - **Freqüência 2 vezes maior**
  - **Velocidade da MP não é alterada**
    - » **Penalidade = 2 x 100 = 200 ciclos**
- **Número total de ciclos de "stall" por instrução**

$$2\% \times 200 + 36\% \times 4\% \times 200 = 6{,}88$$

Impacto com aumento de f

- **Razão dos tempos de execução**

$$\frac{\text{tempo}_{\text{execução}}\ (\text{lento})}{\text{tempo}_{\text{execução}}\ (\text{rápido})} = \frac{I \times CPI_{\text{lento}} \times T}{I \times CPI_{\text{rápido}} \times T/2} = \frac{5{,}44}{8{,}88 \times 1/2} = 1{,}23$$

- **O computador tem o dobro da freqüência**
  - **Mas seu desempenho é apenas 1,2 vezes maior**
    - » **Devido aos fracassos na cache**
- **Conclusão:**
  - **Quanto maior a f, maior o impacto dos "stalls".**
- **Tendência**
  - **Freqüência da CPU aumenta**
  - **Mas velocidade da MP não aumenta na mesma proporção**
- **Desempenho: compromisso entre pipeline e cache.**

Melhoria de desempenho

- **Redução de fracassos**
  - **Posicionamento mais flexível**
  - **Associatividade**
    - » **Mapeamento direto**
    - » **Memória associativa por conjunto**
    - » **Memória totalmente associativa**
- **Redução da penalidade**
  - **Múltiplos níveis de cache**

Mapeamento direto

- **Bloco da MP → única posição da cache**
- **Conseqüência: para procurar um bloco**
  - **Uma única posição é pesquisada.**
  - **Requer 1 comparador/cache.**

Cache totalmente associativa

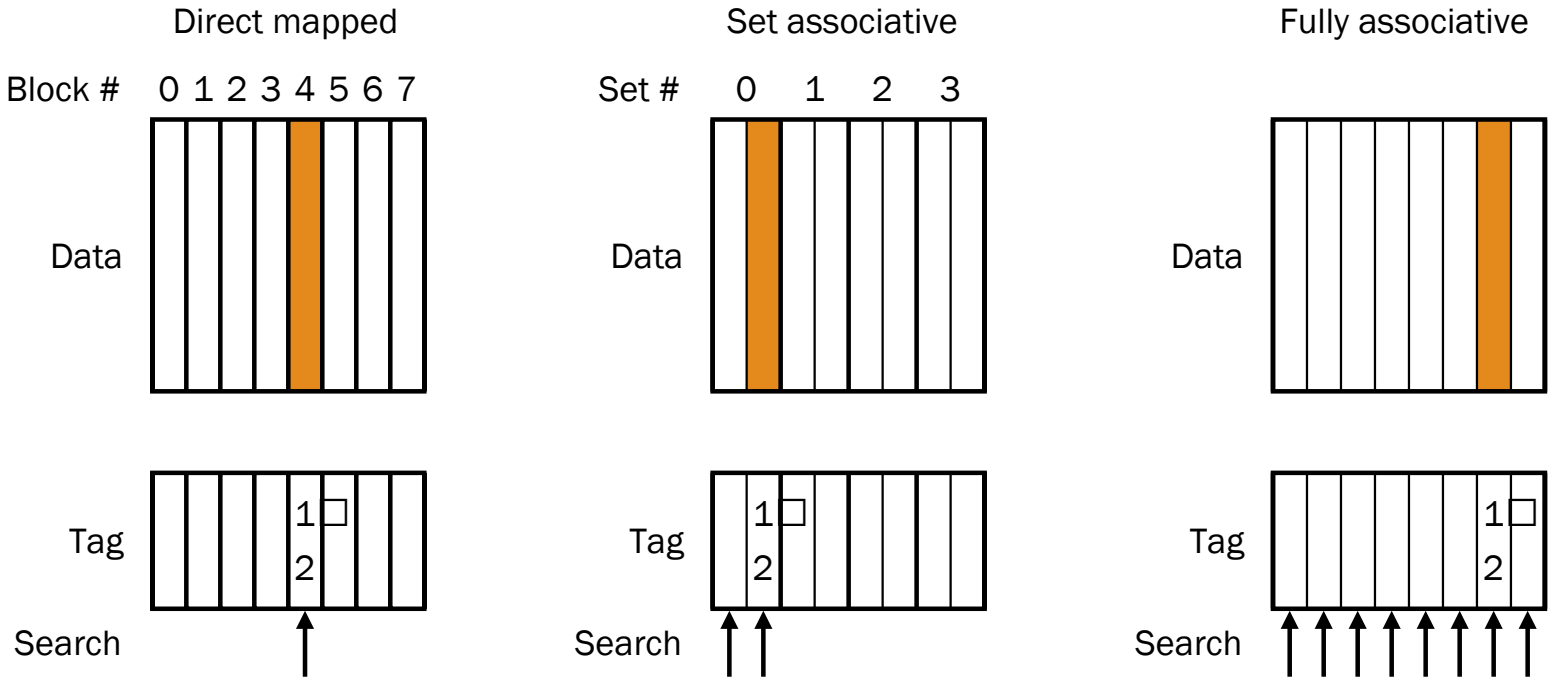
- **Bloco da MP → qualquer posição da cache**
- **Conseqüência: para procurar um bloco**
  - **Todas as posições precisam ser pesquisadas.**
  - **Requer 1 comparador/posição da cache.**

Cache associativa por conjunto

- **Bloco da MP → número fixo de posições da cache**
  - **Qualquer posição dentro de um único conjunto**
    - » **Cache associativa por conjunto de ordem n**
    - » **“n-way set-associative cache”**
- **Conseqüência: Para procurar um bloco na cache**
  - **Um único conjunto é pesquisado.**
  - **Todas as posições do conjunto precisam ser pesquisadas.**
  - **Requer 1 comparador/posição do conjunto**
- **Mapeamento: E modulo S**
  - **E: endereço do bloco S: número de conjuntos na cache**

Tipos de posicionamento na cache

- **Exemplo: bloco de memória cujo endereço é 12**



Exemplo de estrutura

One-way set associative
(direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

Grau de associatividade ↑

Taxa de fracassos ↓

Exemplo de comportamento

- **Cache**
  - **4 blocos de uma palavra**
- **Alternativas**
  - **totalmente associativa,**
  - **2-way**
  - **mapeamento direto**
- **Seqüência de endereços de bloco**
  - **0, 8, 0, 6, 8**
- **Objetivo**
  - **Computar o número de fracassos para cada alternativa**

Exemplo de comportamento

- **Mapeamento direto**

Bloco de memória	Bloco da cache
0	(0 modulo 4) = 0
6	(6 modulo 4) = 2
8	(8 modulo 4) = 0

Bloco da memória	F ou S	Bl. 0	Bl. 1	Bl. 2	Bl. 3
0	F	Mem[0]			
8	F	Mem[8]			
0	F	Mem[0]			
6	F	Mem[0]		Mem[6]	
8	F	Mem[8]		Mem[6]	

5 fracassos!

Exemplo de comportamento

- **2-way**

Bloco de memória	Bloco da cache
0	(0 modulo 2) = 0
6	(6 modulo 2) = 0
8	(8 modulo 2) = 0

Bloco da memória	F ou S	Conj. 0	Conj. 0	Conj. 1	Conj. 1
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[6]		
8	F	Mem[8]	Mem[6]		

4 fracassos!

Exemplo de comportamento

- **Cache totalmente associativa**

Bloco da memória	F ou S	Bl. 0	Bl. 1	Bl. 2	Bl. 3
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	S	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[8]	Mem[6]	
8	S	Mem[0]	Mem[8]	Mem[6]	

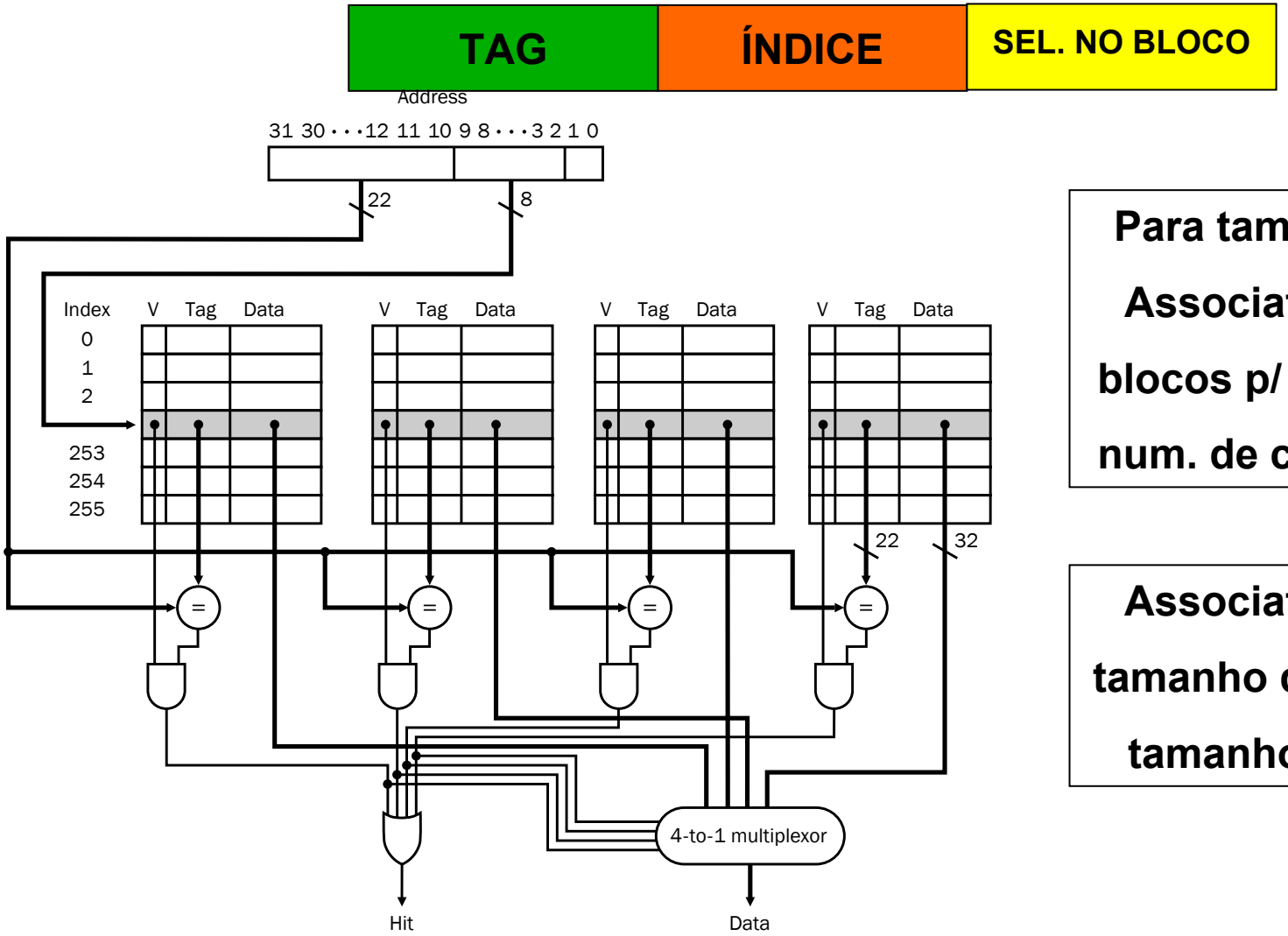
3 fracassos!

O impacto da associatividade

- Cache de dados do Intrinsity FastMATH (16 KB)
- SPEC2000 benchmarks
- Associatividade: de 1 a 8

Associativity	Data miss rate
1	10,3%
2	8,6%
4	8,3%
8	8,1%

Organização de uma Cache n-way



Para tamanho fixo:

Associatividade ↑

blocos p/ conjunto ↑

num. de conjuntos ↓

Associatividade ↑

tamanho do índice ↓

tamanho do tag ↑