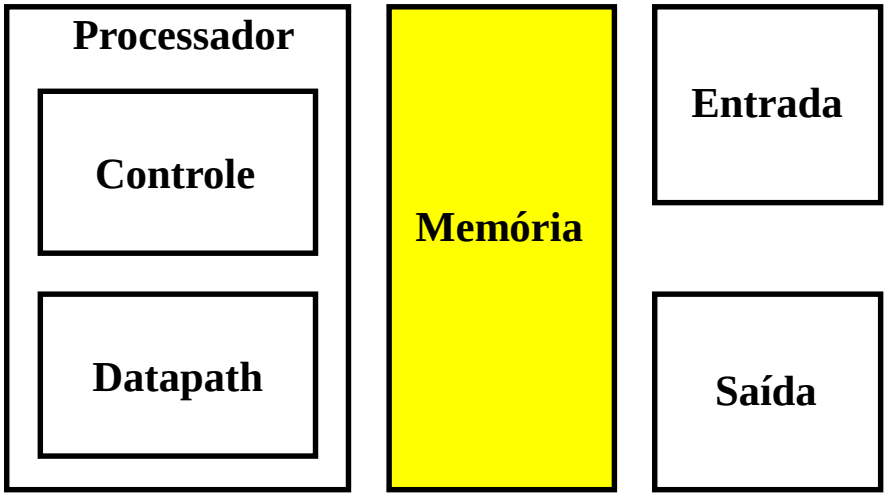
Cache: associatividade e desempenho



Melhoria de desempenho

- **Ciclos gastos com paradas devidas a faltas**

$$\frac{\text{acessos}}{\text{programa}} \times \text{mr} \times \text{penalidade}$$

- **Redução da taxa de faltas (mr)**
  - **Posicionamento mais flexível via associatividade**
    - » **Mapeamento direto**
    - » **Memória associativa por conjunto**
    - » **Memória totalmente associativa**
- **Redução da penalidade**
  - **Memória entrelaçada**
  - **Múltiplos níveis de cache**

Mapeamento direto

- **Bloco da MP → única posição da cache**
- **Consequência: para procurar um bloco**
  - **Uma única posição é pesquisada**
    - » **Bloco encontrado via indexação**
  - **Requer 1 comparador/cache**

Cache totalmente associativa

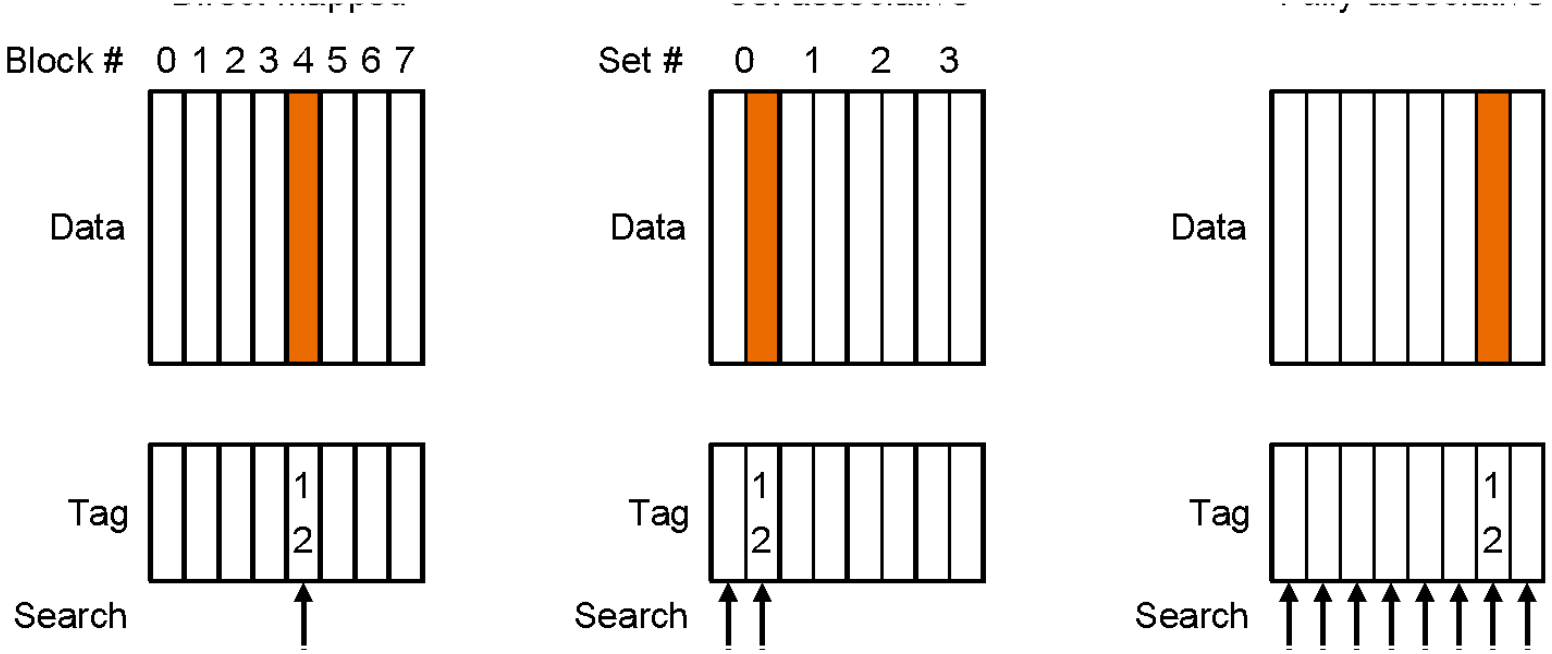
- **Bloco da MP → qualquer posição da cache**
- **Consequência: para procurar um bloco**
  - **Todas as posições precisam ser pesquisadas.**
    - » **Bloco encontrado via associação de um padrão binário**
    - » **Nenhum índice é utilizado**
  - **Requer 1 comparador/posição da cache.**

Cache associativa por conjunto

- **Bloco da MP → número fixo de posições da cache**
  - **Qualquer posição dentro de um único conjunto**
    - » **Cache associativa por conjunto com n alternativas**
    - » **"n-way set-associative cache"**
- **Consequência: Para procurar um bloco na cache**
  - **Um único conjunto é pesquisado**
    - » **Conjunto encontrado via indexação**
  - **Todas as n posições do conjunto são pesquisadas**
    - » **Bloco encontrado via associação de padrão binário**
  - **Requer 1 comparador/posição do conjunto**
- **Mapeamento:**
  - **(Endereço do bloco) mod (número de conjuntos da cache)**

Tipos de posicionamento na cache

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



Exemplo de estrutura

(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				

Grau de associatividade↑

Taxa de faltas↓

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

Exemplo de comportamento

- **Cache**
  - **4 blocos de uma palavra**
- **Alternativas**
  - **totalmente associativa,**
  - **2-way**
  - **mapeamento direto**
- **Sequência de endereços de bloco**
  - **0, 8, 0, 6, 8**
- **Objetivo**
  - **Computar o número de faltas 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 A	Bl. 0	Bl. 1	Bl. 2	Bl. 3
0	F	Mem[0]			

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 A	Bl. 0	Bl. 1	Bl. 2	Bl. 3
0	F	Mem[0]			
8	F	Mem[8]			

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 A	Bl. 0	Bl. 1	Bl. 2	Bl. 3
0	F	Mem[0]			
8	F	Mem[8]			
0	F	Mem[0]			

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 A	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]	

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 A	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 faltas!

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 A	Conj. 0	Conj. 0	Conj. 1	Conj. 1
0	F	Mem[0]			

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 A	Conj. 0	Conj. 0	Conj. 1	Conj. 1
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		

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 A	Conj. 0	Conj. 0	Conj. 1	Conj. 1
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	A	Mem[0]	Mem[8]		

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 A	Conj. 0	Conj. 0	Conj. 1	Conj. 1
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	A	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[6]		

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 A	Conj. 0	Conj. 0	Conj. 1	Conj. 1
0	F	Mem[0]			
8	F	Mem[0]	Mem[8]		
0	A	Mem[0]	Mem[8]		
6	F	Mem[0]	Mem[6]		
8	F	Mem[8]	Mem[6]		

4 faltas!

Exemplo de comportamento

- **Cache totalmente associativa**

Bloco da memória	F ou A	Bl. 0	Bl. 1	Bl. 2	Bl. 3
0	F	Mem[0]			

Exemplo de comportamento

- **Cache totalmente associativa**

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

Exemplo de comportamento

- **Cache totalmente associativa**

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

Exemplo de comportamento

- **Cache totalmente associativa**

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

Exemplo de comportamento

- **Cache totalmente associativa**

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

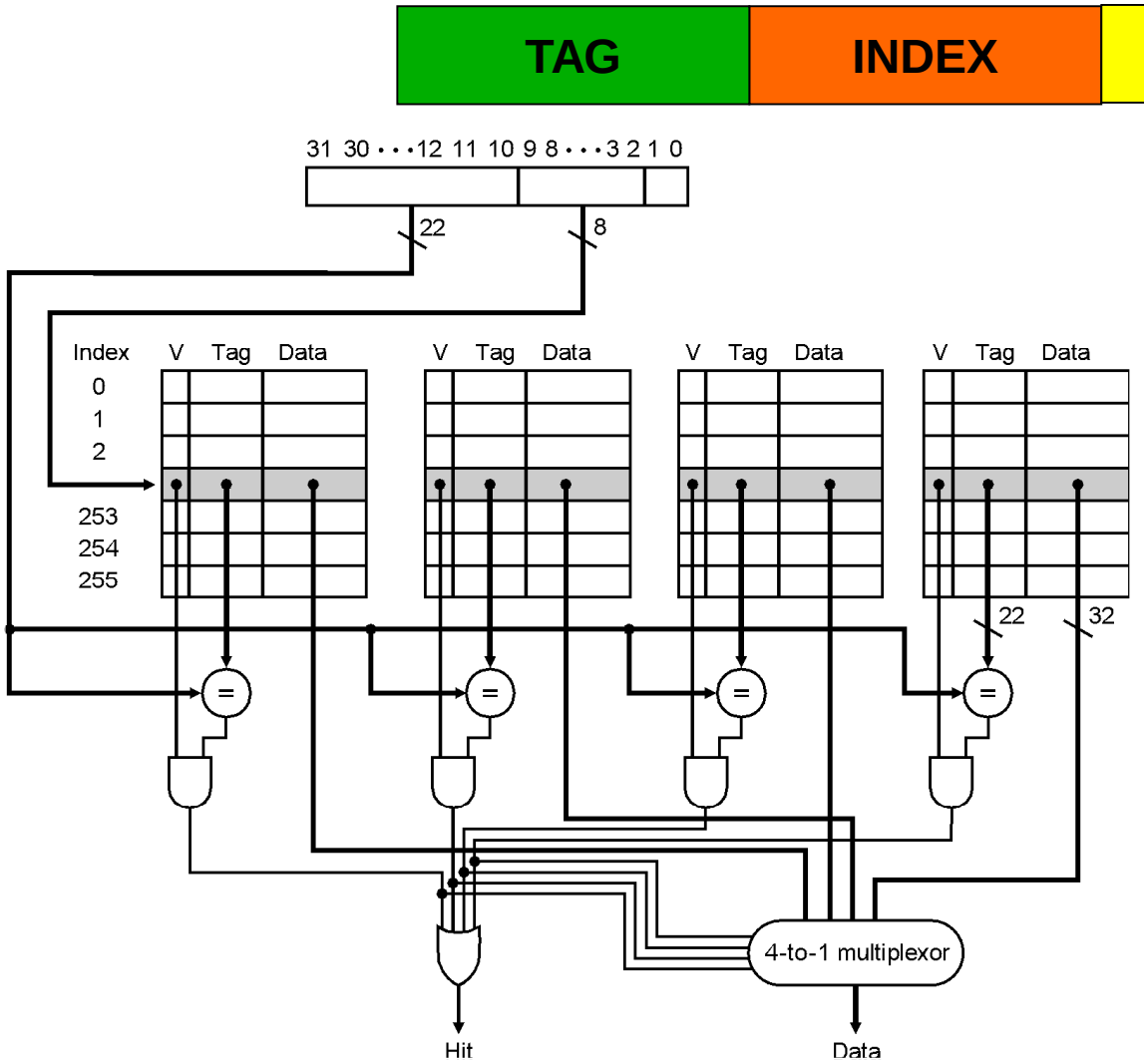
3 faltas!

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 ↑

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 paradas por faltas em cache:**

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

Desempenho da cache

- **Ciclos de paradas por faltas em leitura:**

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

- **Ciclos de paradas por faltas em escrita:**

[Hipóteses: 1) se *write through*: *overhead* insignificante com buffer de escrita; 2) se *write back*: penalidade captura tempo para copiar na memória o bloco a ser substituído; 3) se *write allocate*: penalidade captura tempo de leitura de bloco antes de nele escrever-se uma palavra.

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

Desempenho da cache

- **Combinando escrita e leitura**
  - **Supondo penalidades idênticas**
    - » **Sob a hipótese de *write allocate***
  - **Taxa de faltas (mr) combinada**

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

Cache: exemplo de impacto no CPI

- **Dado um programa, suponha**
  - **mr (I) = 2% e mr (D) = 4%**
  - **CPI = 2 para cache ideal (não gera paradas)**
  - **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

- **Ciclos de parada por falta no acesso a instruções:**

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

- **Ciclos de parada por falta no acesso a dados:**

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

- **CPI total capturando o efeito das paradas:**

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

- **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 total capturando efeito das paradas:**

$$CPI_{total} = 1 + 3{,}44 = 4{,}44$$

- **Razão dos tempos de execução:**

$$\frac{tempo_{execução}\ (real)}{tempo_{execução}\ (ideal)} = \frac{I \times CPI_{real} \times T_r}{I \times CPI_{ideal} \times T_r} = \frac{4{,}44}{1} = 4{,}44$$

Comparação CPI = 2 → 1

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

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

- **Conclusão:**
  - **Quanto menor o CPI, maior o impacto das paradas.**
- **Tendência:**
  - **Emissão múltipla: 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 paradas)**
  - **Loads + stores = 36% (SPECInt2000)**
  - **Frequência 2 vezes maior**
  - **Velocidade da MP não é alterada**
    - » **Penalidade = 2 x 100 = 200 ciclos**
- **CPI capturando apenas o efeito das paradas:**

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

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 frequência**
  - **Mas seu desempenho é apenas 1,2 vezes maior**
    - » **Devido às faltas na cache**
- **Conclusão:**
  - **Quanto maior a f, maior o impacto dos "stalls".**
- **Tendência**
  - **Mesmo quando a frequência da CPU aumenta**
  - **A velocidade da MP não aumenta na mesma proporção**
- **Desempenho: compromisso entre pipeline e cache.**