

INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 25: Caches associativas e associativas por conjunto

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br



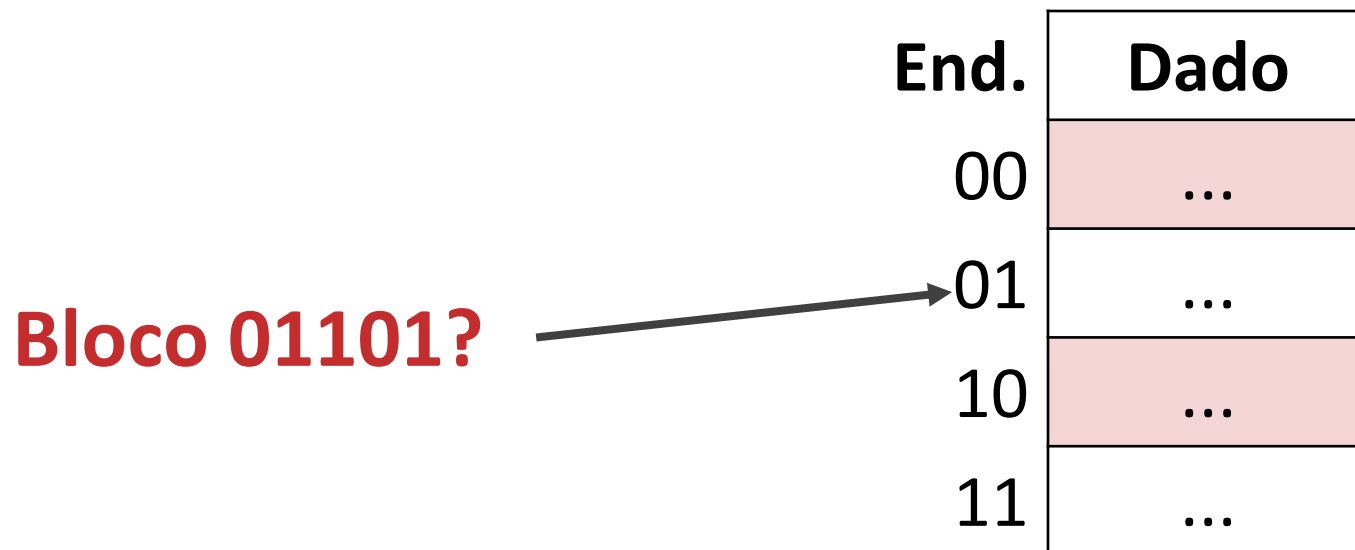
Sumário

- Problemas do mapeamento direto
- Caches associativas
- Caches associativas por conjunto
- Comparação
- Considerações finais

PROBLEMAS DO MAPEAMENTO DIRETO

Problemas do mapeamento direto

- Rápida revisão
 - Mapeamento direto
 - Mapeamento = $E \text{ módulo } N$
 - E: endereço do bloco (não da palavra ou byte)
 - N: número de blocos na cache



Problemas do mapeamento direto

- Exemplo

- Cache com quatro blocos

- Mapeamento direto

- Sequência de acesso a blocos: 0, 8, 0, 6, 8

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
Bloco 0					
Bloco 8					
Bloco 0					
Bloco 6					
Bloco 8					

Problemas do mapeamento direto

- Exemplo
 - Cache com quatro blocos
 - Mapeamento direto
 - Sequência de acesso a blocos: 0, 8, 0, 6, 8

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
Bloco 0	Falha	Mem[0]			
Bloco 8	Falha	Mem[8]			
Bloco 0	Falha	Mem[0]			
Bloco 6	Falha	Mem[0]		Mem[6]	
Bloco 8	Falha	Mem[8]		Mem[6]	

Problemas do mapeamento direto

- Mas isso nunca aconteceria....
 - Cache com dois blocos
 - Blocos de duas words
 - Item do array: int acc, int val, int x, int y;

```
for ( i=1 ; i<=N ; i++ )  
    array.acc[i] = array.val[i] + array.acc[i-1];
```

Problemas do mapeamento direto

- Mas esse caso é um exagero...
 - Cache com quatro blocos de uma palavra cada

```
int matriz[N][N];    //N = 100
for ( j=0 ; j < N ; j++)
    for ( i=1 ; i < N ; i++)
        matriz[i-1][j] = matriz[i][j];
```


Problemas do mapeamento direto

- Acesso à memória
 - $\text{matriz}[i][j] \rightarrow \text{matriz}[i*N+j]$
 - No caso
 - $\text{matriz}[3][4] \rightarrow \text{matriz}[304]$
 - Mapeamento: $304 \bmod 4 = 0$
 - $\text{matriz}[2][4] \rightarrow \text{matriz}[204]$
 - Mapeamento: $204 \bmod 4 = 0$
 - Ou seja, **falhas e mais falhas....**

Problemas do mapeamento direto

- Como poderíamos resolver esse problema de conflito?
 - **Que tal um mapeamento mais flexível?**

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
Bloco 0	Falha	Mem[0]			
Bloco 8	Falha	Mem[8]			
Bloco 0	Falha	Mem[0]			
Bloco 6	Falha	Mem[0]		Mem[6]	
Bloco 8	Falha	Mem[8]		Mem[6]	

CACHES ASSOCIATIVAS

Caches associativas

- Extremo oposto a caches com mapeamento direto
 - Mapeamento direto: apenas uma posição para um endereço de bloco
 - Mapeamento associativo: **todas as posições para um endereço de bloco!**

Caches associativas

- Exemplo
 - Cache com quatro blocos
 - Cache associativa
 - Sequência de acesso a blocos: 0, 8, 0, 6, 8

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
Bloco 0					
Bloco 8					
Bloco 0					
Bloco 6					
Bloco 8					

Caches associativas

- Exemplo
 - Cache com quatro blocos
 - Cache associativa
 - Sequência de acesso a blocos: 0, 8, 0, 6, 8

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
Bloco 0	Falha		Mem[0]		
Bloco 8	Falha		Mem[0]		Mem[8]
Bloco 0	Acerto		Mem[0]		Mem[8]
Bloco 6	Falha	Mem[6]	Mem[0]		Mem[8]
Bloco 8	Acerto	Mem[6]	Mem[0]		Mem[8]

Caches associativas

- Como calcular o mapeamento?
 - **É só escolher um slot disponível**
- E se não houver slot disponível?
 - Políticas de substituição
 - Circular
 - Aleatório
 - *Least Recently Used* (**LRU**)
 - Substitui bloco que foi acessado menos recentemente
- Como identificar quem está mapeado?
 - Tag

Caches associativas

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Palavras de 32 bits
 - Blocos de 2 palavras
 - Cache de 16 blocos
 - Endereço de memória: 0110 0101 1010

Cache	Tag	End. bloco	End. Interno	
			word	byte
Map. Direto	01100	1011	0	10
Associativa	011001011	-	0	10

Caches associativas

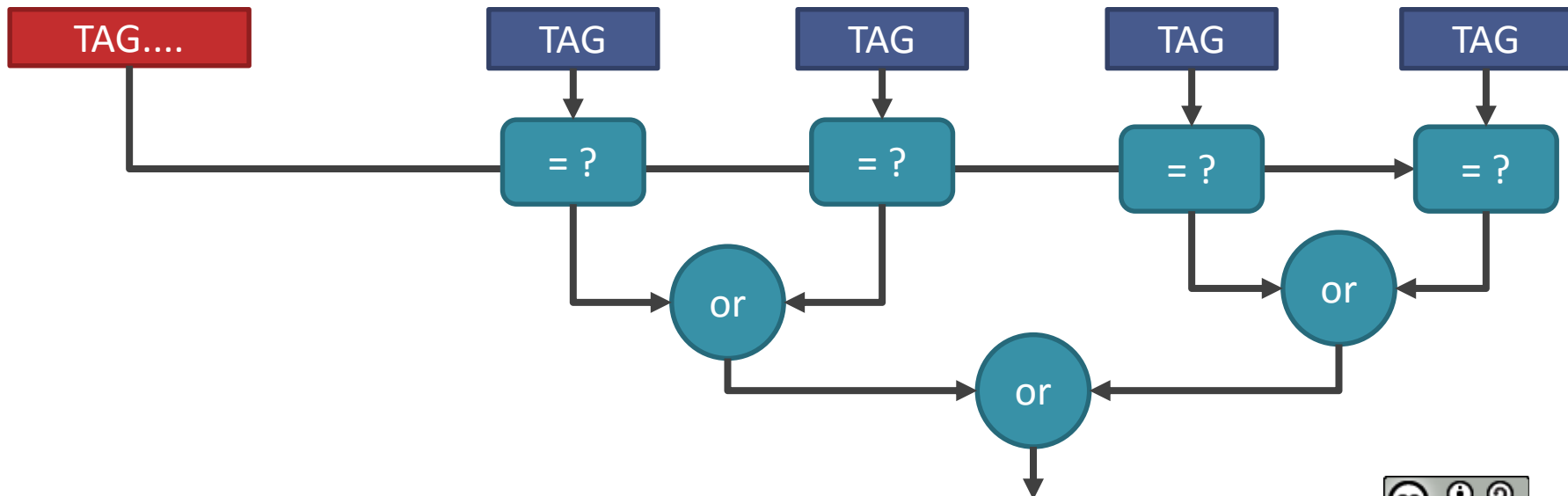
- Cálculo do tamanho da cache e da tag
 - Memória de 2^m bytes (endereçáveis)
 - Cache com 2^c blocos
 - Blocos de 2^p palavras
 - Palavras de 2^b bytes (2^{b+3} bits)
 - 1 bit de validade por bloco
 - Tamanho do tag: **$m - (p+b)$** bits
 - Armazenamento da cache: $2^{(c+p+b)}$ bytes
 - Tamanho efetivo: $2^c * (2^{p+b+3} + 1 + m - (p+b))$ bits

Caches associativas

- Vantagens (vs mapeamento direto)
 - Maior taxa de acertos
 - Maior desempenho :D
- Desvantagens
 - Circuito mais complexo
 - Múltiplas comparações de tag em paralelo
 - Muito caro para caches grandes :<
 - Reduções
 - Pode levar a tempo de acesso maior :~

Caches associativas

- Falácia: Caches associativas comparam tags em sequência
 - Circuitos trabalham em paralelo
 - N comparadores para N blocos
 - Árvore de redução



CACHES ASSOCIATIVAS POR CONJUNTO

Caches associativas por conjuntos

- Caminho do meio
 - **Agrupar blocos de cache em conjuntos**
 - Um endereço só pode ser mapeado para um conjunto
 - Dentro do conjunto, o endereço pode ser mapeado para qualquer posição

Caches associativas por conjuntos

- Vantagens
 - **Maior grau de associatividade leva a maior taxa de acertos**
 - Menos conflitos
 - **Menor grau de associatividade requer menos recursos**
 - Comparações apenas dentro de um conjunto

Caches associativas por conjuntos

- Exemplo

- Cache com quatro blocos

- Cache 2-associativa

- Sequência de acesso a blocos: 0, 8, 0, 6, 8

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		Conjunto 0		Conjunto 1	
		Posição 0	Posição 1	Posição 0	Posição 1
Bloco 0					
Bloco 8					
Bloco 0					
Bloco 6					
Bloco 8					

Caches associativas por conjuntos

- Exemplo

- Cache com quatro blocos

- Cache 2-associativa (conjuntos de dois blocos)
- Sequência de acesso a blocos: 0, 8, 0, 6, 8

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		Conjunto 0		Conjunto 1	
		Posição 0	Posição 1	Posição 0	Posição 1
Bloco 0	Falha	Mem[0]			
Bloco 8	Falha	Mem[0]	Mem[8]		
Bloco 0	Acerto	Mem[0]	Mem[8]		
Bloco 6	Falha	Mem[0]	Mem[6]		
Bloco 8	Falha	Mem[0]	Mem[8]		

Caches associativas por conjuntos

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Palavras de 32 bits, blocos de 2 palavras
 - Cache de 16 blocos
 - Endereço de memória: 0110 0101 1010

Cache	Tag	End. bloco ou conj.	End. Interno	
			word	byte
Map. Direto	01100	1011	0	10
Associativa	011001011	-	0	10
2-Associativa	011001	011	0	10
8-Associativa	01100101	1	0	10

Caches associativas por conjuntos

- Comparação para cache com oito blocos

Cache 1-associativa (mapeamento direto)

TAG	D
	(0)
	(1)
	(2)
	(3)
	(4)
	(5)
	(6)
	(7)

Cache 2-associativa

TAG	D	TAG	D
	(0)		(4)
	(1)		(5)
	(2)		(6)
	(3)		(7)

Cache 4-associativa

TAG	D	TAG	D	TAG	D	TAG	D
	(0)		(2)		(4)		(6)
	(1)		(3)		(5)		(7)

Cache 8-associativa (totalmente associativa)

TAG	D	TAG	D	TAG	D	TAG	D	TAG	D	TAG	D	TAG	D	TAG	D
	(0)		(1)		(2)		(3)		(4)		(5)		(6)		(7)

Caches associativas por conjuntos

- Maior associatividade
 - Mais blocos por conjunto
 - Maior número de comparações
 - Maiores tags
 - Menor índice para escolher conjunto
 - Maior taxa de acertos

Caches associativas por conjuntos

- Cálculo do tamanho da cache e da tag
 - Memória de 2^m bytes (endereçáveis)
 - Cache com 2^n conjuntos
 - Conjuntos com 2^c blocos
 - Blocos de 2^p palavras
 - Palavras de 2^b bytes (2^{b+3} bits)
 - 1 bit de validade por bloco
 - Tamanho do tag: $m - (n+p+b)$ bits
 - Armazenamento da cache: $2^{(n+c+p+b)}$ bytes
 - Tamanho: $2^{n+c} * (2^{p+b+3} + 1 + m - (n+p+b))$ bits
- Total de blocos = 2^{n+c}**

Caches associativas por conjuntos

- Exercício
 - Memória de 4GB (2^{32} bytes)
 - Palavras de 32 bits
 - Blocos de 8 palavras
 - Cache de 128 blocos
 - Endereço de memória: 0xBEBAC0CA
 - Qual o mapeamento do bloco e a tag para
 - Cache com mapeamento direto?
 - Cache totalmente associativa?
 - Cache 4-associativa?

Caches associativas por conjuntos

- Exercício
 - Memória de 4GB (2^{32} bytes)
 - Palavras de 32 bits
 - Blocos de 8 palavras
 - Cache de 128 blocos
 - Qual o tamanho efetivo da cache
 - com mapeamento direto?
 - totalmente associativa?
 - 4-associativa?

CONSIDERAÇÕES FINAIS

Considerações finais

- Cache de mapeamento direto
- Cache associativa
- Cache associativa em conjunto
- Tags e tamanhos efetivos

INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 25: Caches associativas e associativas por conjunto

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

