

INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 24: Cache com mapeamento direto

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br



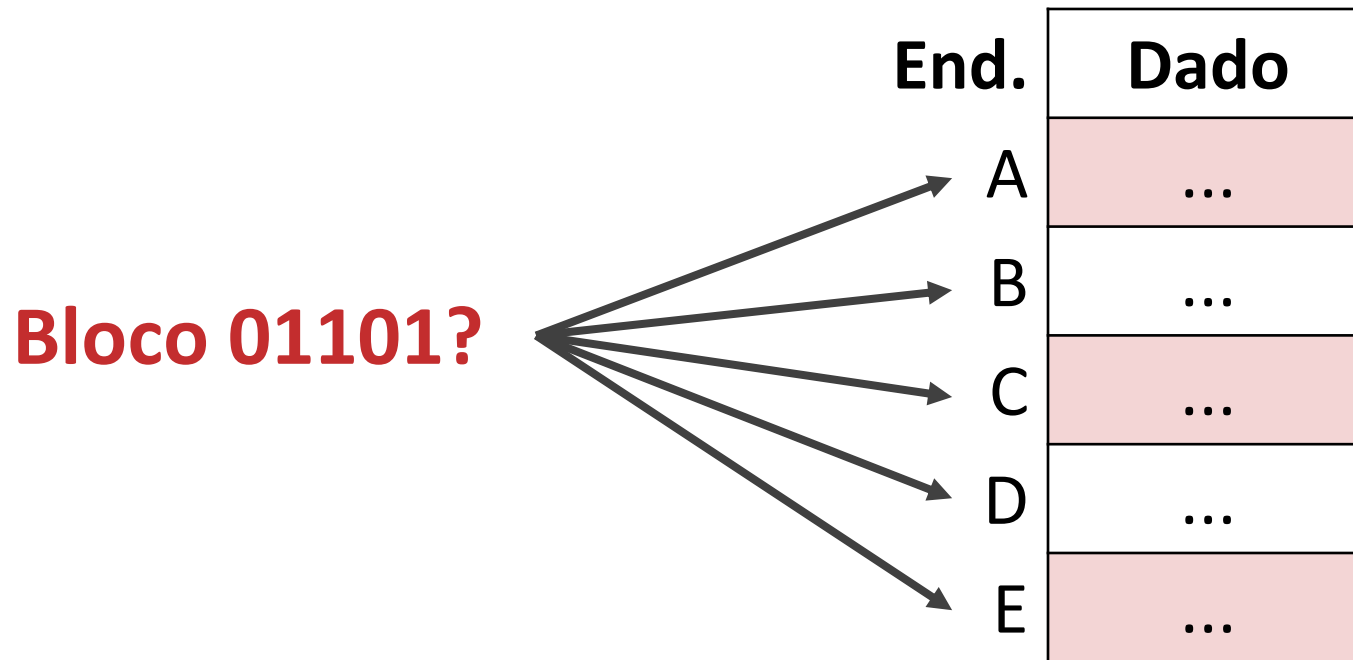
Sumário

- Princípios do mapeamento direto
- Cálculo de endereço
- Tags e validade
- Hits e misses
- Tamanho efetivo de caches
- Considerações finais

PRINCÍPIOS DO MAPEAMENTO DIRETO

Princípios do mapeamento direto

- Dado um endereço de memória,
 - onde pode ser encontrado o dado na cache?
 - onde deve ser armazenado o dado trazido do próximo nível da hierarquia?



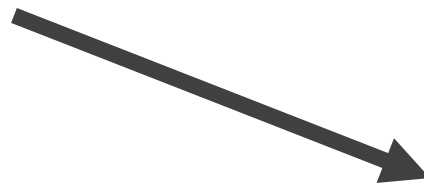
Princípios do mapeamento direto

- Regra do mapeamento direto
 - **Cada posição da memória é mapeada para apenas uma posição da cache**
 - Função de mapeamento $M_d: \mathbb{N} \rightarrow \mathbb{N}$
 - Mapeamento = $E \bmod N$
 - E : endereço do bloco
 - N : número de blocos na cache
 - Módulo significa o resto da divisão inteira

Princípios do mapeamento direto

- Regra do mapeamento direto
 - Mapeamento = $E \bmod N$
 - Mapeamento = $13 \bmod 5 = 3$

Bloco 01101 (13)?



End.	Dado
0	...
1	...
2	...
3	...
4	...

Princípios do mapeamento direto

- Se $N = 2^n$
 - Cálculo em binário simplificado
 - Exemplo
 - Endereço do bloco: 010101
 - Tamanho da cache: 2 blocos (2^1)
 - Mapeamento: $010101 \bmod 10 = 010101 = 1$
 - Posição indexada na cache pelos n bits menos significativos do endereço do bloco
 - Como eu chego no endereço do bloco?

CÁLCULO DE ENDEREÇO

Cálculo de endereço

- Como calcular o endereço de um bloco e seu mapeamento em uma cache de mapeamento direto?
 - **Abordagem cutelo**



Cálculo de endereço

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Palavras de 32 bits
 - Blocos de 2 palavras
 - Cache de 16 blocos
 - Endereço de memória: 011001011010

Cálculo de endereço

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Palavras de 32 bits
 - Blocos de 2 palavras
 - Cache de 16 blocos
 - Endereço de memória: 011001011010

0110010110 | 10

Endereço da word | 4 bytes na word

Cálculo de endereço

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Palavras de 32 bits
 - Blocos de 2 palavras
 - Cache de 16 blocos
 - Endereço de memória: 011001011010

011001011 | 0

Endereço do bloco | 2 words no bloco

Cálculo de endereço

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Palavras de 32 bits
 - Blocos de 2 palavras
 - Cache de 16 blocos
 - Endereço de memória: 011001011010

01100 | 1011

Tag do bloco | mapeamento para
16 blocos

Cálculo de endereço

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Palavras de 32 bits
 - Blocos de 2 palavras
 - Cache de 16 blocos
 - Endereço de memória: 011001011010
 - 01100 – Tag do bloco
 - 1011 – Mapeamento do bloco (16 blocos)
 - 0 – Words no bloco (2 words)
 - 10 – Bytes na word (4 bytes)

Cálculo de endereço

- Exercício
 - Memória de 32KB (2^{16} bytes)
 - Palavras de 64 bits
 - Blocos de 4 palavras
 - Cache de 32 blocos
 - Endereço de memória: 1110 0111 1111 0000
 - Qual o mapeamento do bloco?

Cálculo de endereço

- 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?

TAGS E VALIDADE

Tags e validade

- Exemplo
 - Memória de 1KB (2^{10} bytes)
 - Blocos de 2 palavras de 32 bits
 - Cache de 4 blocos
 - Endereço de memória: 01 1001 0110
 - 01100 – Tag do bloco
 - 10 – Mapeamento do bloco (4 blocos)
 - 1 – Words no bloco (2 words)
 - 10 – Bytes na word (4 bytes)
 - Como saber se o dado endereçado está na cache?

Tags e validade

- Exemplo

01100**10**110

01101**10**111

01110**10**100

10000**10**100

End.

Palavra

Dado

00

0

0x44444444

1

0x55555555

01

0

0x66666666

1

0x77777777

10

0

0x88888888

1

0x99999999

11

0

0x00000000

1

0x11111111

De quem é o dado?
O dado é válido?

Tags e validade

- Identificação de quem está usando a cache
 - Uso da tag do bloco (etiqueta)

0110010110

01100 – Tag do bloco

10 – Mapeamento

1 – Words

10 – Bytes

End.	Palavra	Tag	Dado
00	0	...	0x44444444
	1		0x55555555
01	0	...	0x66666666
	1		0x77777777
10	0	01100	0x88888888
	1		0x99999999
11	0	...	0x00000000
	1		0x11111111

Tags e validade

- Validade do dado
 - Bit de validade (0 = não, 1 = sim)

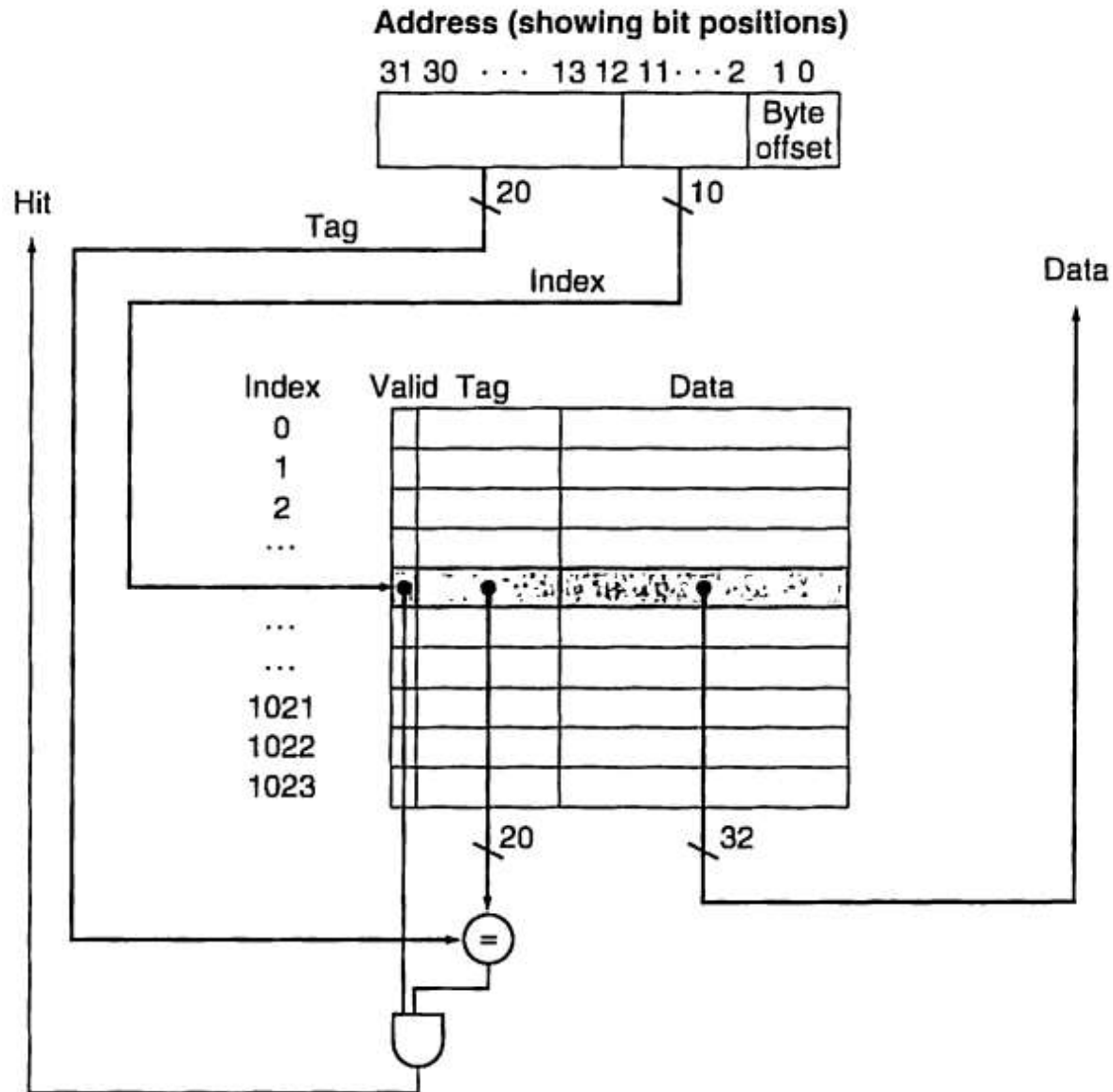
End.	Palavra	Validade	Tag	Dado
00	0	0	...	0x44444444
	1			0x55555555
01	0	0	...	0x66666666
	1			0x77777777
10	0	1	01100	0x88888888
	1			0x99999999
11	0	1	...	0x00000000
	1			0x11111111

Tags e validade

- Blocos = linhas de cache (p/ map. direto)
 - Melhor visualização do que é armazenado na cache
 - 1 bit de validade e 1 tag por bloco
 - **Identificador de palavra não é armazenado**

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	0	...	0x44444444	0x55555555
01	0	...	0x66666666	0x77777777
10	1	01100	0x88888888	0x99999999
11	1	...	0x00000000	0x11111111

Tags e validade



HITS E MISSES

Hits e misses

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Blocos de 2 palavras de 64 bits
 - Cache de 4 blocos

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	N			
01	N			
10	N			
11	N			

Hits e misses

- Exemplo
 - Sequência de acessos
 - 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000
 - Cache começa vazia

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	N			
01	N			
10	N			
11	N			

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x000 = ~~0000~~**0000** 0000

- Cache miss, traz dado do bloco para a memória

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	N			
01	N			
10	N			
11	N			

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x000 = ~~0000~~ **0000** ~~0000~~

- MEM(0x000) = dados na memória na posição

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0000 00	MEM(0x000)	MEM(0x008)
01	N			
10	N			
11	N			

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x220 = ~~0010 0010 0000~~

- Cache miss, traz dado do bloco para a memória

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0000 00	MEM(0x000)	MEM(0x008)
01	N			
10	N -> S	0010 00	MEM(0x220)	MEM(0x228)
11	N			

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x448 = ~~0100 0100~~ 1000

- Tag não bate = cache miss, substitui dados

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0000 00	MEM(0x000)	MEM(0x008)
01	N			
10	S	0010 00	MEM(0x220)	MEM(0x228)
11	N			

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x448 = ~~0100 0100~~ **1000**

- Tag não bate = cache miss, substitui dados

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0100 01	MEM(0x440)	MEM(0x448)
01	N			
10	S	0010 00	MEM(0x220)	MEM(0x228)
11	N			

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x778 = ~~0111~~ **0111** 1000

- Cache miss, traz dado do bloco para a memória

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0100 01	MEM(0x440)	MEM(0x448)
01	N			
10	S	0010 00	MEM(0x220)	MEM(0x228)
11	N -> S			

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x770 = ~~0111~~ **0111** 0000

- Cache hit por localidade espacial

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0100 01	MEM(0x440)	MEM(0x448)
01	N			
10	S	0010 00	MEM(0x220)	MEM(0x228)
11	S	0111 01	MEM(0x770)	MEM(0x778)

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x000 = ~~0000~~**0000** 0000

- Tag não bate = cache miss, substitui dados

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0100 01	MEM(0x440)	MEM(0x448)
01	N			
10	S	0010 00	MEM(0x220)	MEM(0x228)
11	S	0111 01	MEM(0x770)	MEM(0x778)

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000

- 0x000 = ~~0000~~**0000** 0000

- Cache hit, localidade temporal

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0000 00	MEM(0x000)	MEM(0x000)
01	N			
10	S	0010 00	MEM(0x220)	MEM(0x228)
11	S	0111 01	MEM(0x770)	MEM(0x778)

Hits e misses

- Exemplo

- Sequência de acessos

- 0x000, 0x220, 0x448, 0x778, 0x770, 0x000, 0x000
 - 4 misses para trazer dados, 1 p/ dado substituído
 - 2 hits

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	S	0000 00	MEM(0x000)	MEM(0x000)
01	N			
10	S	0010 00	MEM(0x220)	MEM(0x228)
11	S	0111 01	MEM(0x770)	MEM(0x778)

TAMANHO EFETIVO DA CACHE

Tamanho efetivo da cache

- Exemplo
 - Memória de 4KB (2^{12} bytes)
 - Blocos de 2 palavras de 64 bits
 - Cache de 4 blocos

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	N			
01	N			
10	N			
11	N			

Tamanho efetivo da cache

- Exemplo

- Armazena 4 blocos * 2 words * 8 bytes = 64 bytes = 512 bits

- Quanto espaço a cache ocupa realmente?

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	N			
01	N			
10	N			
11	N			

Tamanho efetivo da cache

- Exemplo

- Armazena 4 blocos * 2 words * 8 bytes = 64 bytes = 512 bits

- Quanto espaço a cache ocupa realmente?

- $4 * 1 + 4 * 6 + 4 * 2 * 64 = 4 + 24 + 512 = 540$ bits

End.	Validade	Tag	Dado 1ª word	Dado 2ª word
00	1	6	64	64
01	1	6	64	64
10	1	6	64	64
11	1	6	64	64

Tamanho efetivo da cache

- Cálculo do tamanho da cache
 - 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: $m - (c+p+b)$ bits
 - Armazenamento da cache: $2^{(c+p+b)}$ bytes
 - **Tamanho efetivo:** $2^c * (2^{p+b+3} + 1 + m - (c+p+b))$ bits

CONSIDERAÇÕES FINAIS

Considerações finais

- Cache de mapeamento direto
 - Um bloco é mapeado apenas para uma posição
 - Implementação simples
- Cálculo de endereço
- Cálculo de tamanho da cache

INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 24: Cache com mapeamento direto

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

