

Organização e Arquitetura de Computadores

Memória Cache: Mapeamento Direto

Alexandre Amory

Edson Moreno

Na Aula Anterior ...

- Vimos que existem memórias
 - De **rápido acesso** mas pouca capacidade de armazenamento.
Ex.: SRAM
 - De **alta capacidade de armazenamento**, porém muito lenta. Ex.: disco magnético
- O conceito de hierarquia de memória
 - Combina hierarquicamente memórias rápidas (prox do processador) e com alta capacidade de armazenamento (+ longe do processador)
 - Da a impressão de alto desempenho e grande memória
 - Baseado nas características de **localidade** temporal e espacial

Na Aula de Hoje

- Memória cache
 - Política de atualização de blocos de memória baseado em **mapeamento direto**

Introdução

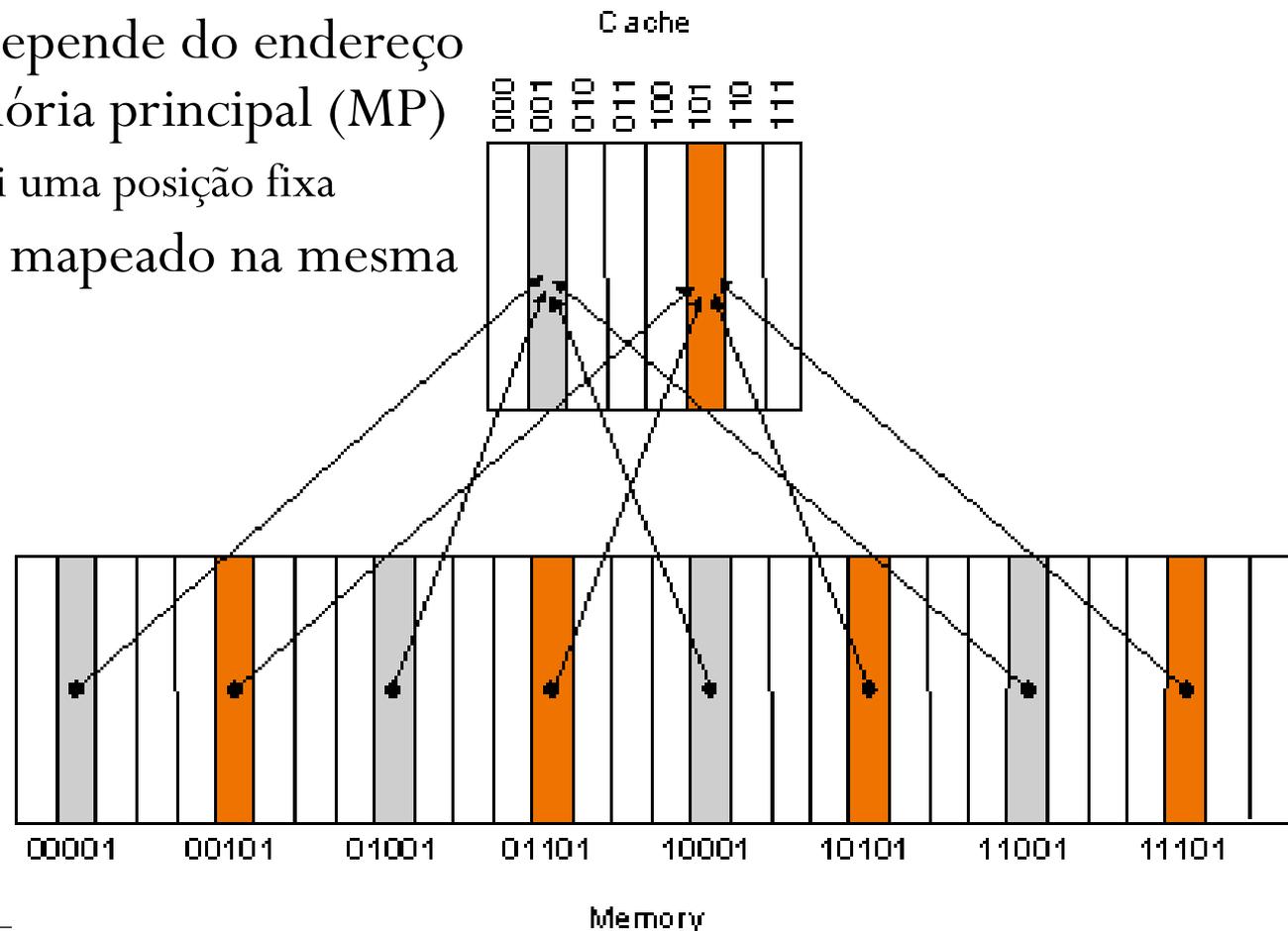
- Funcionalidade
 - Área de memória rápida e com informações dinâmicas
- Cache só pode ter parte dos dados do nível mais abaixo
 - Tamanho menor
- Problemas
 - Como identificar se o dado procurado está na cache?
 - Se estiver, como acessar de forma rápida?
 - Se não estiver, como buscar eficientemente de níveis inferiores?
 - Qual dado tirar da cache para colocar o novo dado?
- Processador não sabe qual memória física tem o dado
 - Gera apenas endereços e a hierarquia se encarrega de acessar a informação endereçada

Mapeamento de Endereços

- Como fazer para pesquisar um dado na cache?
 - Fazer cache com todos endereços não faz sentido
 - Efetuar varredura seqüencial na cache leva muito tempo
- Solução
 - Fazer mapeamento de endereços
- Objetivo
 - Relacionar informações (dados e instruções) da memória principal com posições da cache
- Formas de mapeamento de memórias cache
 - Direto
 - Associativo
 - Conjunto associativo

Mapeamento Direto

- Forma mais simples de mapeamento
 - Posição na cache depende do endereço da palavra na memória principal (MP)
 - Cada palavra possui uma posição fixa
 - Grupo de palavras mapeado na mesma posição da cache



CACHE



Memória Principal

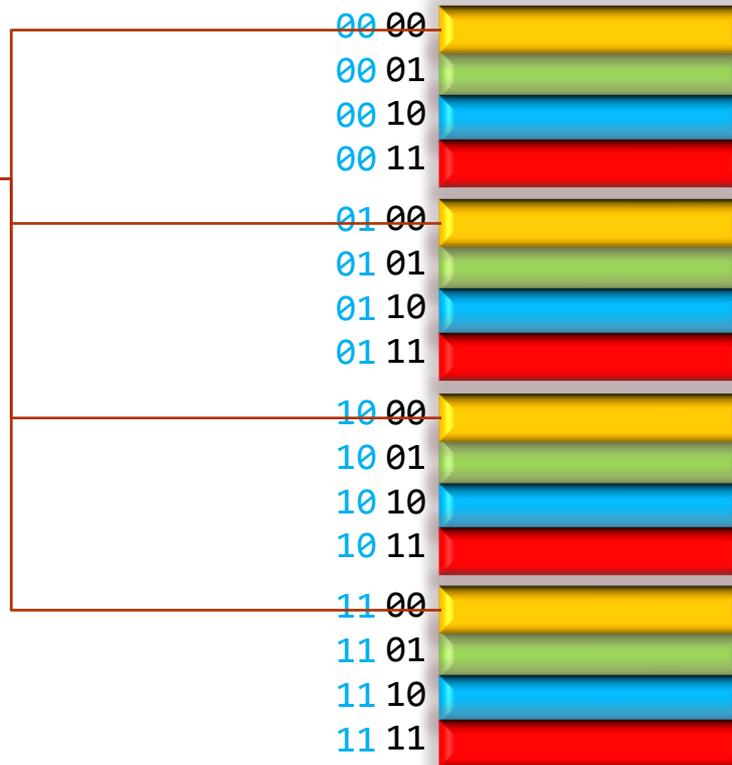


CACHE



Mapeamento da linha 00 da cache

Memória Principal

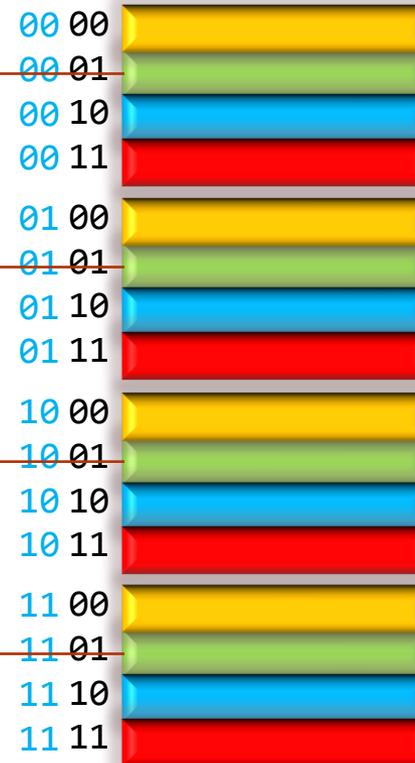


CACHE



Mapeamento da linha 01 da cache

Memória Principal



CACHE



E se a linha da cache estiver não preenchida ?

Resposta: **bit de validade**

Memória Principal



CACHE



endereço 4 bits (16 posições)

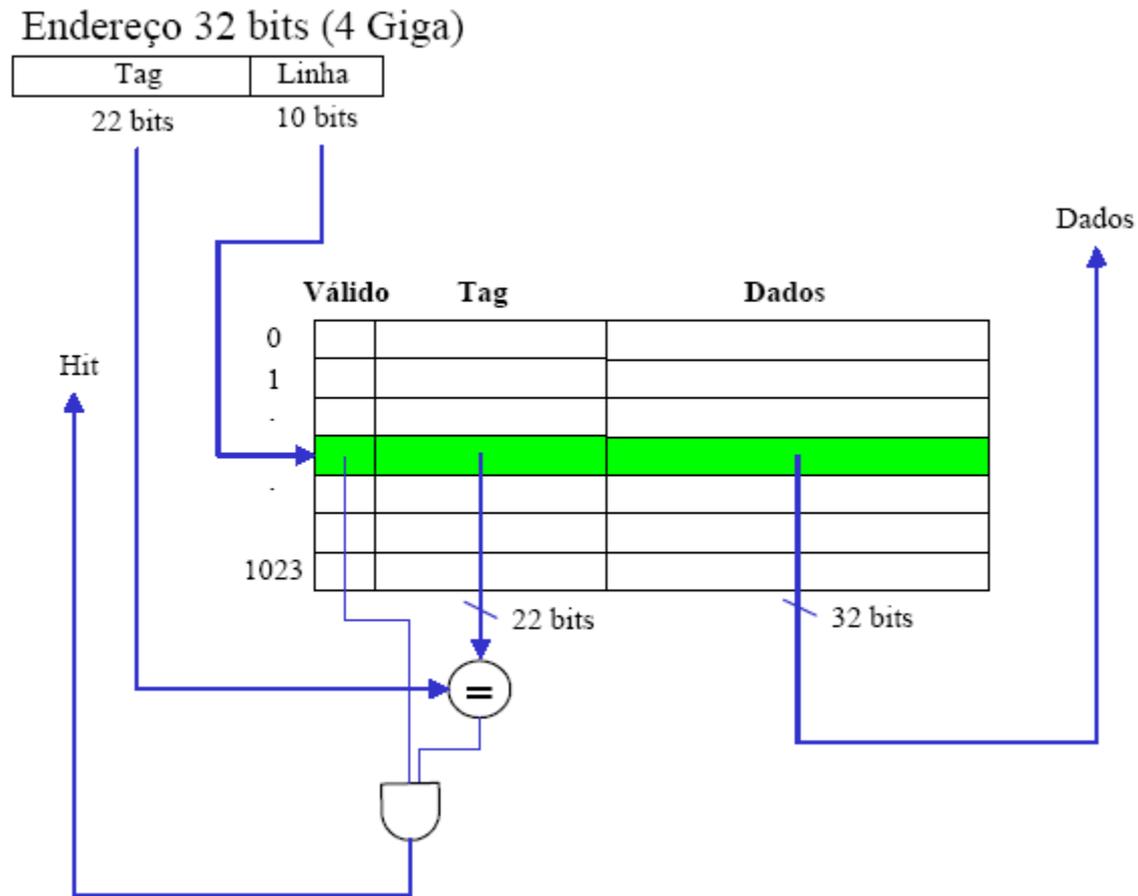
tag (2 bits) | linha (2 bits)

Memória Principal



Bit validade e Tag

- Divisão de bits no registrador de endereçamento
 - Exemplo de uma cache com 1024 linhas (2^{10}) com palavra de 32 bits



Mapeamento Direto: Passos

1. Usar os bits menos significativos do endereço para determinar a linha da cache

Ex. endereço: 0001

Linha da cache: 01

2. Se (bit de validade da posição for válido) então

Se (Tag diferente de endereço) então

Acusar miss; buscar dados no nível inferior

Senão

Ler posição (hit)

Senão

Acusar miss; buscar dados no nível inferior

Exemplo Cache Mapeamento Direto

CACHE



Como fica o conteúdo da cache depois
desta seqüência de acessos à memória
? Assumir cache inicialmente vazia

0000,0010,0110,0000,1000,0000

Memória Principal



Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	0	00	00
01	0	01	01
10	0	10	10
11	0	11	11

Validade inicia em '0'.

0000,0010,0110, 0000,1000,0000

Memória Principal

00	00	00
00	01	01
00	10	10
00	11	11
01	00	00
01	01	01
01	10	10
01	11	11
10	00	00
10	01	01
10	10	10
10	11	11
11	00	00
11	01	01
11	10	10
11	11	11

Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	0		
11	0		

Preenche linha 00 da cache com o endereço 0000.

0000, 0010, 0110, 0000, 1000, 0000

MISS!

Memória Principal

00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	1	00	0010
11	0		

Preenche linha 10 da cache com o endereço 0010.

0000, **0010**, 0110, 0000, 1000, 0000

MISS!

Memória Principal

00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	1	01	0110
11	0		

A cache possuía o endereço 0010 e não o endereço 0110, portanto, Miss!

0000,0010,**0110**, 0000,1000,0000

MISS!

Memória Principal

00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	1	01	0110
11	0		

O endereço 0000 está na cache.

0000,0010,0110, 0000,1000,0000

Hit!

Memória Principal

00	00	0000
00	01	0001
00	10	0010
00	11	0011
01	00	0100
01	01	0101
01	10	0110
01	11	0111
10	00	1000
10	01	1001
10	10	1010
10	11	1011
11	00	1100
11	01	1101
11	10	1110
11	11	1111

Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	10	1000
01	0		
10	1	01	0110
11	0		

A cache possuía o endereço 0000 e não o endereço 1000, portanto, Miss!

0000,0010,0110, 0000,1000,0000

Miss!

Memória Principal

00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	1	01	0110
11	0		

A cache possuía o endereço 1000 e não o endereço 0000, portanto, Miss!

0000,0010,0110, 0000,1000,0000

Miss!

Memória Principal

00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Exemplo Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	1	01	0110
11	0		

Resultado da seqüência: 5 misses e 1 hit!

0000,0010,0110, 0000,1000,0000

Memória Principal

00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Exercício Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	1	01	0110
11	0		

Memória Principal

00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Como fica o conteúdo da cache depois desta seqüência de acessos à memória ? Responda o conteúdo da cache ao final da seqüência e o nro de hits/misses. Assumir cache parcialmente preenchida, como representado acima.

0000,0110,0000, 0110,1000,0000

Exercício Cache Mapeamento Direto

CACHE

	v	tag	dados
00	1	00	0000
01	0		
10	1	01	0110
11	0		

Resultado: 4 hits e 2 misses

o status final (por coincidência) é igual ao status inicial da cache.

0000, 0110, 0000, 0110, 1000, 0000

Memória Principal

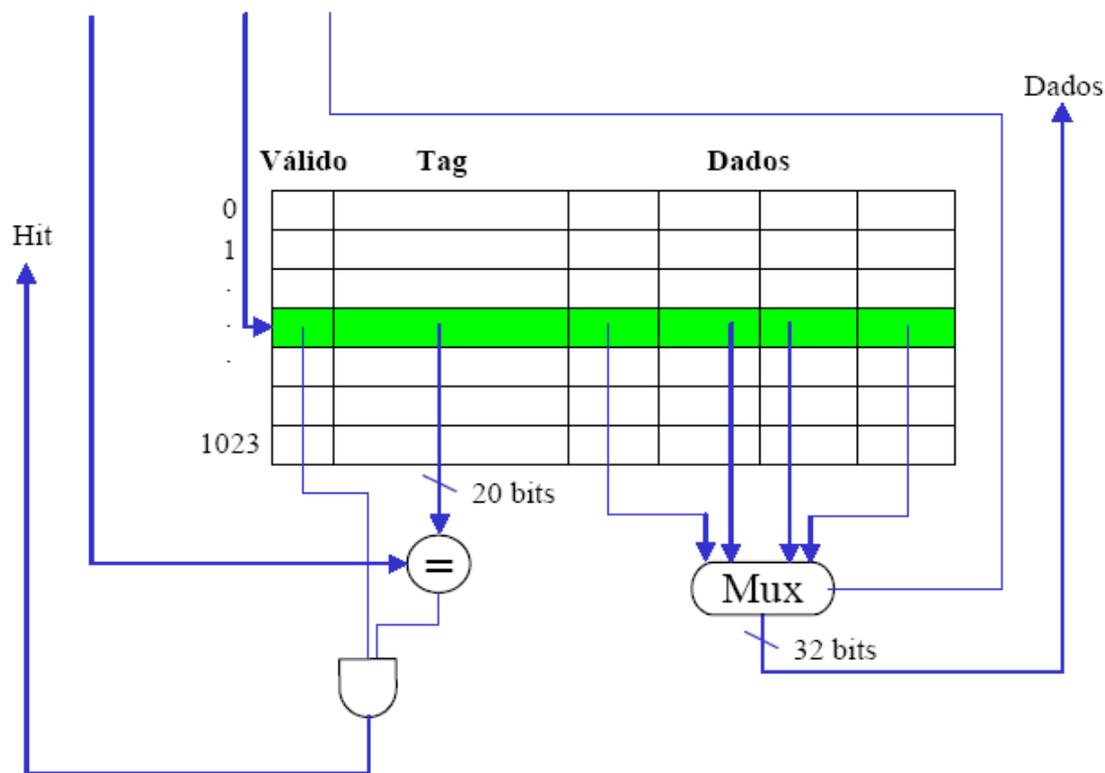
00	00	0000
00	01	0110
00	10	1000
00	11	1110
01	00	0000
01	01	0110
01	10	1000
01	11	1110
10	00	0000
10	01	0110
10	10	1000
10	11	1110
11	00	0000
11	01	0110
11	10	1000
11	11	1110

Transferência de Blocos

- Transferência de blocos entre níveis de memória
 - Exemplo da divisão de blocos em uma cache com 1024 linhas (bloco com 4 palavras de 32 bits)

Endereço 32 bits (4 Giga)

Tag	Linha	P
20 bits	10 bits	2 bits



O que ganho ao utilizar blocos ao invés de palavras?

Mapeamento Direto com Blocos

CACHE



endereço 6 bits (64 posições)

tag (2 bits) | linha (2 bits) | bloco (2 bits)

Memória Principal

00 00 00
00 00 01
00 00 10
00 00 11

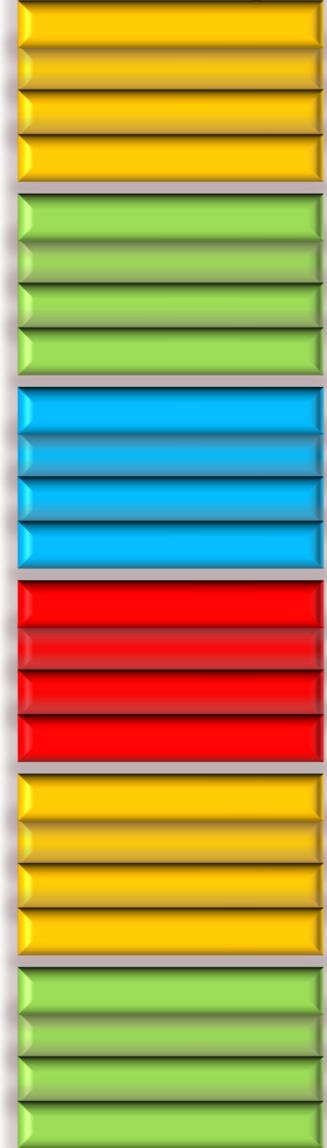
00 01 00
00 01 01
00 01 10
00 01 11

00 10 00
00 10 01
00 10 10
00 10 11

00 11 00
00 11 01
00 11 10
00 11 11

01 00 00
01 00 01
01 00 10
01 00 11

01 01 00
01 01 01
01 01 10
01 01 11



...

Mapeamento Direto com Blocos

CACHE



Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

000000,000010,000110, 000000,001000,000000

Memória Principal

00 00 00
00 00 01
00 00 10
00 00 11

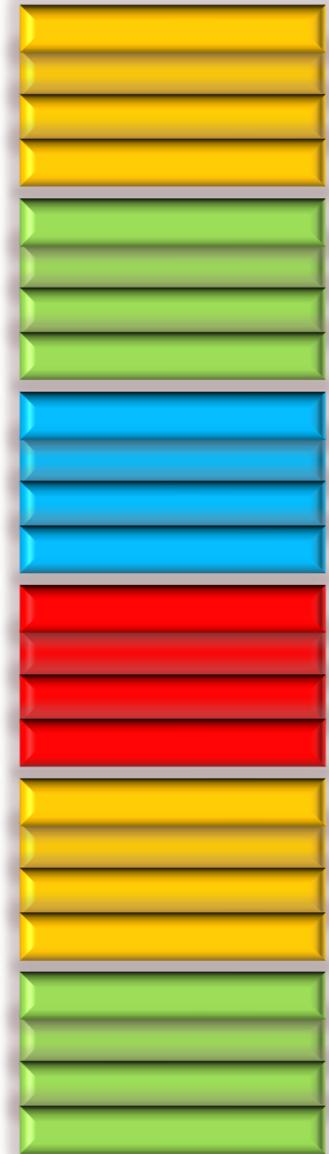
00 01 00
00 01 01
00 01 10
00 01 11

00 10 00
00 10 01
00 10 10
00 10 11

00 11 00
00 11 01
00 11 10
00 11 11

01 00 00
01 00 01
01 00 10
01 00 11

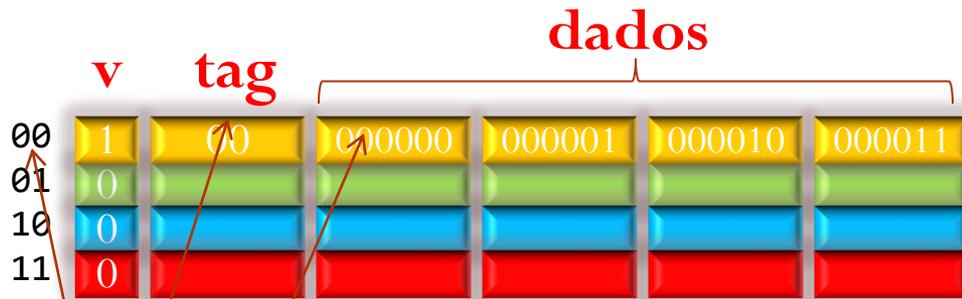
01 01 00
01 01 01
01 01 10
01 01 11



...

Mapeamento Direto com Blocos

CACHE



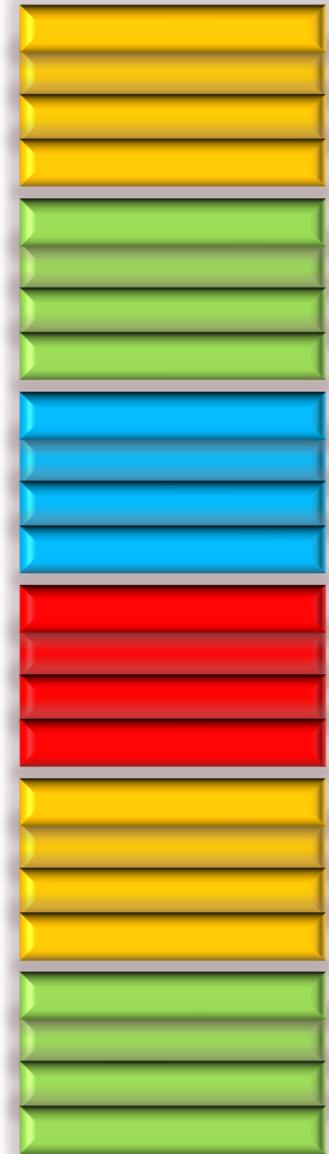
Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

000000, 000010, 000110, 000000, 001000, 000000

Miss!

Memória Principal

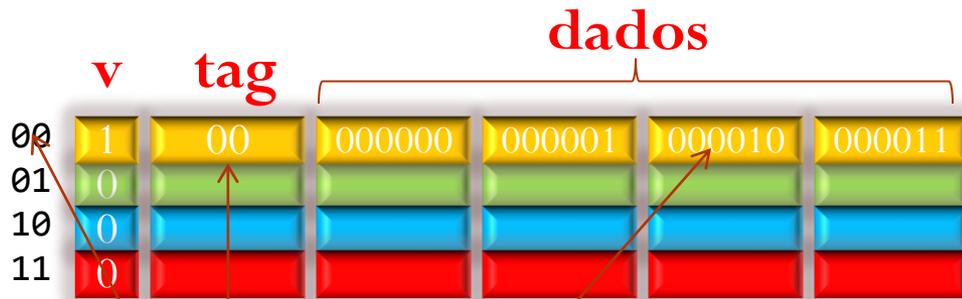
00 00 00
 00 00 01
 00 00 10
 00 00 11
 00 01 00
 00 01 01
 00 01 10
 00 01 11
 00 10 00
 00 10 01
 00 10 10
 00 10 11
 00 11 00
 00 11 01
 00 11 10
 00 11 11
 01 00 00
 01 00 01
 01 00 10
 01 00 11
 01 01 00
 01 01 01
 01 01 10
 01 01 11



...

Mapeamento Direto com Blocos

CACHE



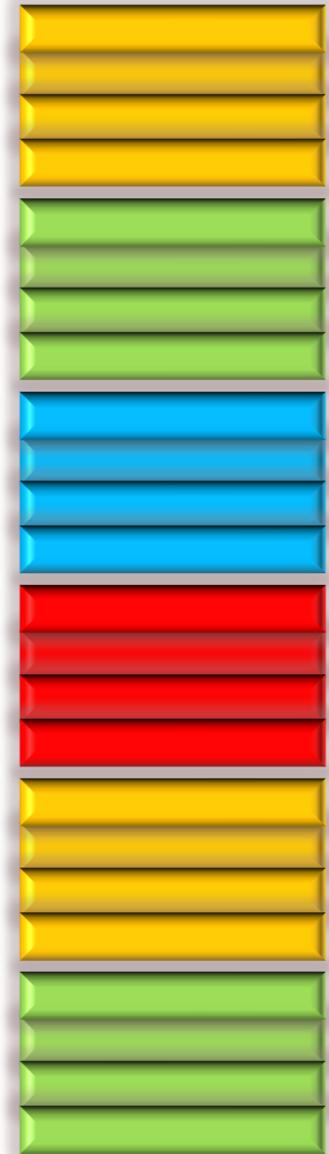
Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

000000, 000010, 000110, 000000, 001000, 000000

Hit!

Memória Principal

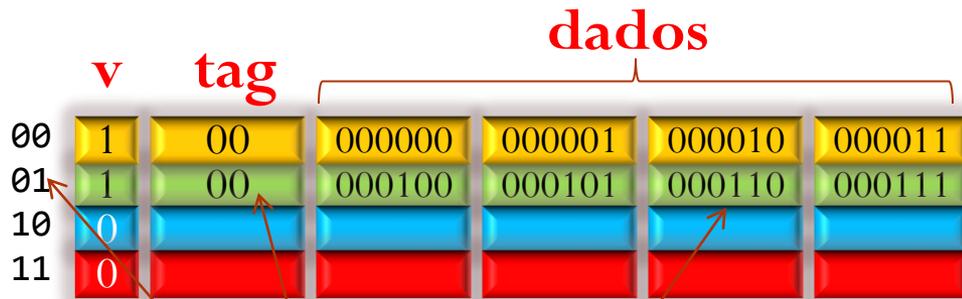
00 00 00
 00 00 01
 00 00 10
 00 00 11
 00 01 00
 00 01 01
 00 01 10
 00 01 11
 00 10 00
 00 10 01
 00 10 10
 00 10 11
 00 11 00
 00 11 01
 00 11 10
 00 11 11
 01 00 00
 01 00 01
 01 00 10
 01 00 11
 01 01 00
 01 01 01
 01 01 10
 01 01 11



...

Mapeamento Direto com Blocos

CACHE



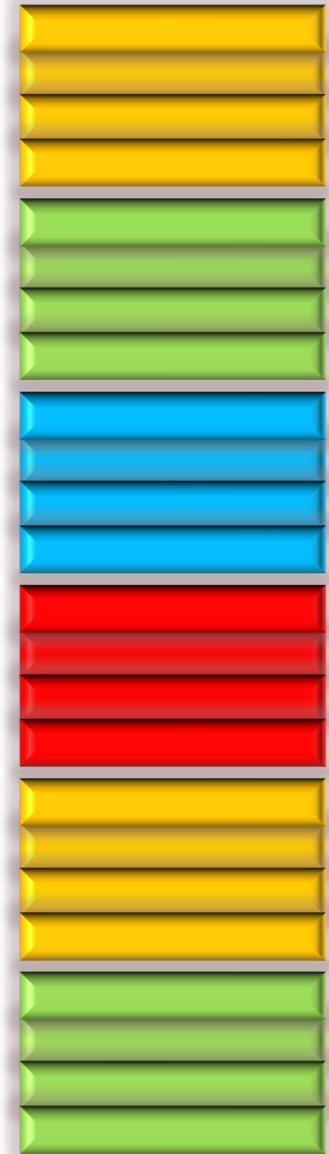
Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

000000, 000010, 000110, 000000, 001000, 000000

Miss!

Memória Principal

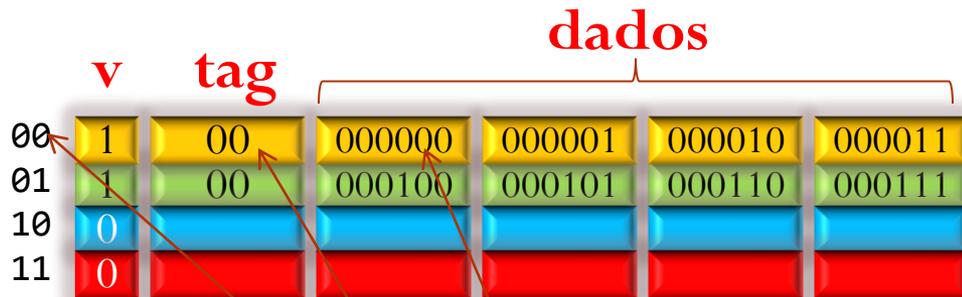
00 00 00
 00 00 01
 00 00 10
 00 00 11
 00 01 00
 00 01 01
 00 01 10
 00 01 11
 00 10 00
 00 10 01
 00 10 10
 00 10 11
 00 11 00
 00 11 01
 00 11 10
 00 11 11
 01 00 00
 01 00 01
 01 00 10
 01 00 11
 01 01 00
 01 01 01
 01 01 10
 01 01 11



...

Mapeamento Direto com Blocos

CACHE



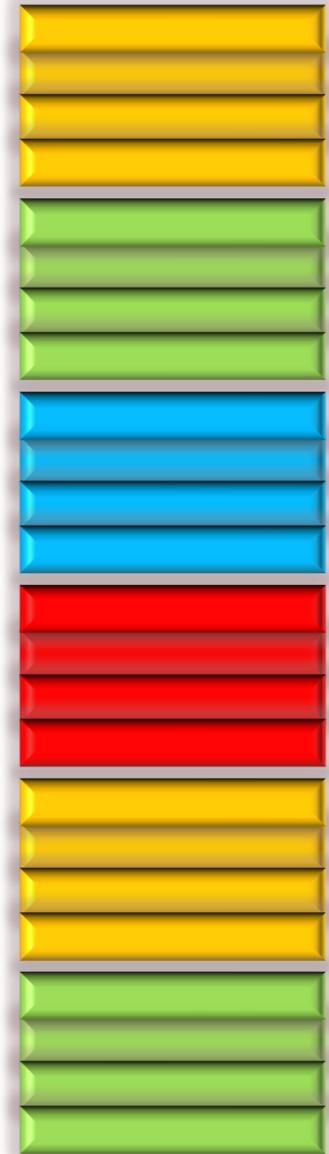
Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

000000, 000010, 000110, 000000, 001000, 000000

Hit!

Memória Principal

00 00 00
 00 00 01
 00 00 10
 00 00 11
 00 01 00
 00 01 01
 00 01 10
 00 01 11
 00 10 00
 00 10 01
 00 10 10
 00 10 11
 00 11 00
 00 11 01
 00 11 10
 00 11 11
 01 00 00
 01 00 01
 01 00 10
 01 00 11
 01 01 00
 01 01 01
 01 01 10
 01 01 11



...

Mapeamento Direto com Blocos

CACHE

	v	tag	dados			
00	1	00	000000	000001	000010	000011
01	1	00	000100	000101	000110	000111
10	1	00	001000	001001	001010	001011
11	0					

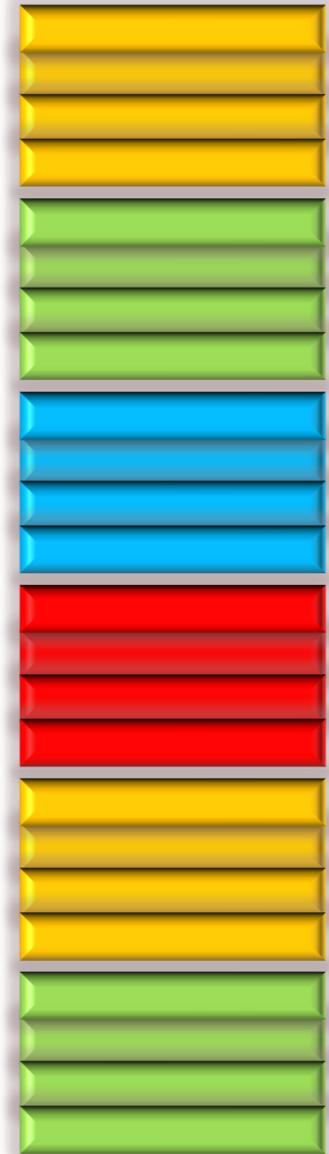
Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

000000,000010,000110,000000,**001000**,000000

Miss!

Memória Principal

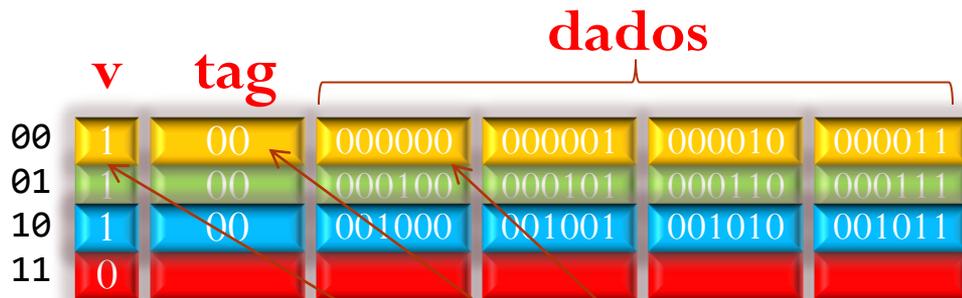
00 00 00
 00 00 01
 00 00 10
 00 00 11
 00 01 00
 00 01 01
 00 01 10
 00 01 11
 00 10 00
 00 10 01
 00 10 10
 00 10 11
 00 11 00
 00 11 01
 00 11 10
 00 11 11
 01 00 00
 01 00 01
 01 00 10
 01 00 11
 01 01 00
 01 01 01
 01 01 10
 01 01 11



...

Mapeamento Direto com Blocos

CACHE



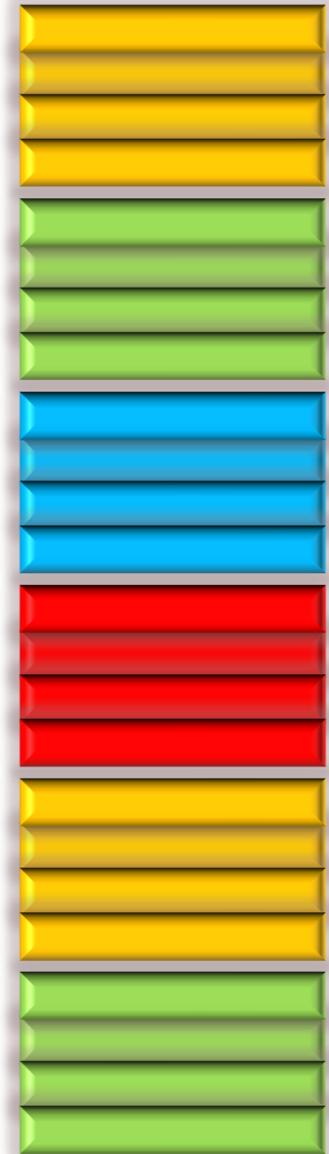
Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

000000, 000010, 000110, 000000, 001000, 000000

Hit!

Memória Principal

00 00 00
 00 00 01
 00 00 10
 00 00 11
 00 01 00
 00 01 01
 00 01 10
 00 01 11
 00 10 00
 00 10 01
 00 10 10
 00 10 11
 00 11 00
 00 11 01
 00 11 10
 00 11 11
 01 00 00
 01 00 01
 01 00 10
 01 00 11
 01 01 00
 01 01 01
 01 01 10
 01 01 11



...

Mapeamento Direto com Blocos

CACHE

	v	tag	dados			
00	1	00	000000	000001	000010	000011
01	1	00	000100	000101	000110	000111
10	1	00	001000	001001	001010	001011
11	0					

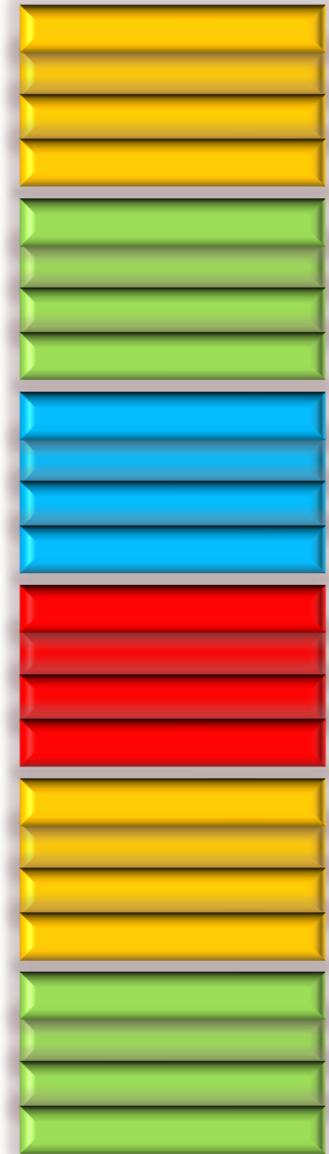
Como fica a cache ai final da seguinte seqüência de acessos ? Qnts misses e hits ocorreram ?

Resposta: 4 hits e 2 misses

000000,000010,000110,000000,001000,000000

Memória Principal

00 00 00
 00 00 01
 00 00 10
 00 00 11
 00 01 00
 00 01 01
 00 01 10
 00 01 11
 00 10 00
 00 10 01
 00 10 10
 00 10 11
 00 11 00
 00 11 01
 00 11 10
 00 11 11
 01 00 00
 01 00 01
 01 00 10
 01 00 11
 01 01 00
 01 01 01
 01 01 10
 01 01 11



...

Exercício: Mapeamento Direto com Blocos

CACHE

	v	tag	dados			
00	1	00	000000	000001	000010	000011
01	1	00	000100	000101	000110	000111
10	1	00	001000	001001	001010	001011
11	0					

Como fica o conteúdo da cache depois desta seqüência de acessos à memória ? Responda o conteúdo da cache ao final da seqüência e o nro de hits/misses. Assumir cache parcialmente preenchida, como representado acima.

000000,000001,000010, 000011,010001,000000,000001

Memória Principal

00	00	00
00	00	01
00	00	10
00	00	11
00	01	00
00	01	01
00	01	10
00	01	11
00	10	00
00	10	01
00	10	10
00	10	11
00	11	00
00	11	01
00	11	10
00	11	11
01	00	00
01	00	01
01	00	10
01	00	11
01	01	00
01	01	01
01	01	10
01	01	11

...

Exercício: Mapeamento Direto com Blocos

CACHE

	v	tag	dados			
00	1	00	000000	000001	000010	000011
01	1	00	000100	000101	000110	000111
10	1	00	001000	001001	001010	001011
11	0					

Memória Principal

00	00	00
00	00	01
00	00	10
00	00	11
00	01	00
00	01	01
00	01	10
00	01	11
00	10	00
00	10	01
00	10	10
00	10	11
00	11	00
00	11	01
00	11	10
00	11	11
01	00	00
01	00	01
01	00	10
01	00	11
01	01	00
01	01	01
01	01	10
01	01	11

Como fica o conteúdo da cache depois desta seqüência de acessos à memória ? Responda o conteúdo da cache ao final da seqüência e o nro de hits/misses. Assumir cache parcialmente preenchida, como representado acima.

Resposta: 5 hits e 2 misses

000000,000001,000010, 000011,010001,000000,000001

...

Exercícios

1. Considerando um espaço de endereçamento de 1 Giga (2^{30}) posições na memória principal. Como ficaria a divisão de bits para uma cache de 2048 linhas que trabalha com blocos de 8 palavras?

Dica: se precisarem, usem a tabela no final deste slide

2. Tomando-se a cache anterior, diga qual o percentual de dados de memória pode efetivamente estar ocupando essa cache, considerando palavras de 8 bits?

Conclusões e Questões

- Vantagens do mapeamento direto
 - Hardware barato
 - Procura simples (posição fixa)
 - Simplicidade / Velocidade
- Desvantagens do mapeamento direto
 - Pode ter mau aproveitamento das posições da cache (dependendo dos endereços gerados)
 - *Miss rate* pode ser alto !
 - Usa parte da cache para controle (armazena tags e bit de validade)

Anexo: Tabela de conversão de endereçamento de memória

- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024 = 1 \text{ KB}$
- $2^{11} = 2048 = 2 \text{ KB}$
- $2^{12} = 4096 = 4 \text{ KB}$
- ...
- $2^{20} = 1,048,576 = 1 \text{ MB}$
- $2^{21} = 2,097,152 = 2 \text{ MB}$
- $2^{22} = 4,194,304 = 4 \text{ MB}$
- ...
- $2^{30} = 1,073,741,824 = 1 \text{ GB}$
- $2^{31} = 2,147,483,648 = 2 \text{ GB}$
- $2^{32} = 4,294,967,296 = 4 \text{ GB}$
- ...