Capítulo 5

Grande e rápida: Explorando a Hierarquia de Memória

Datapath MIPS

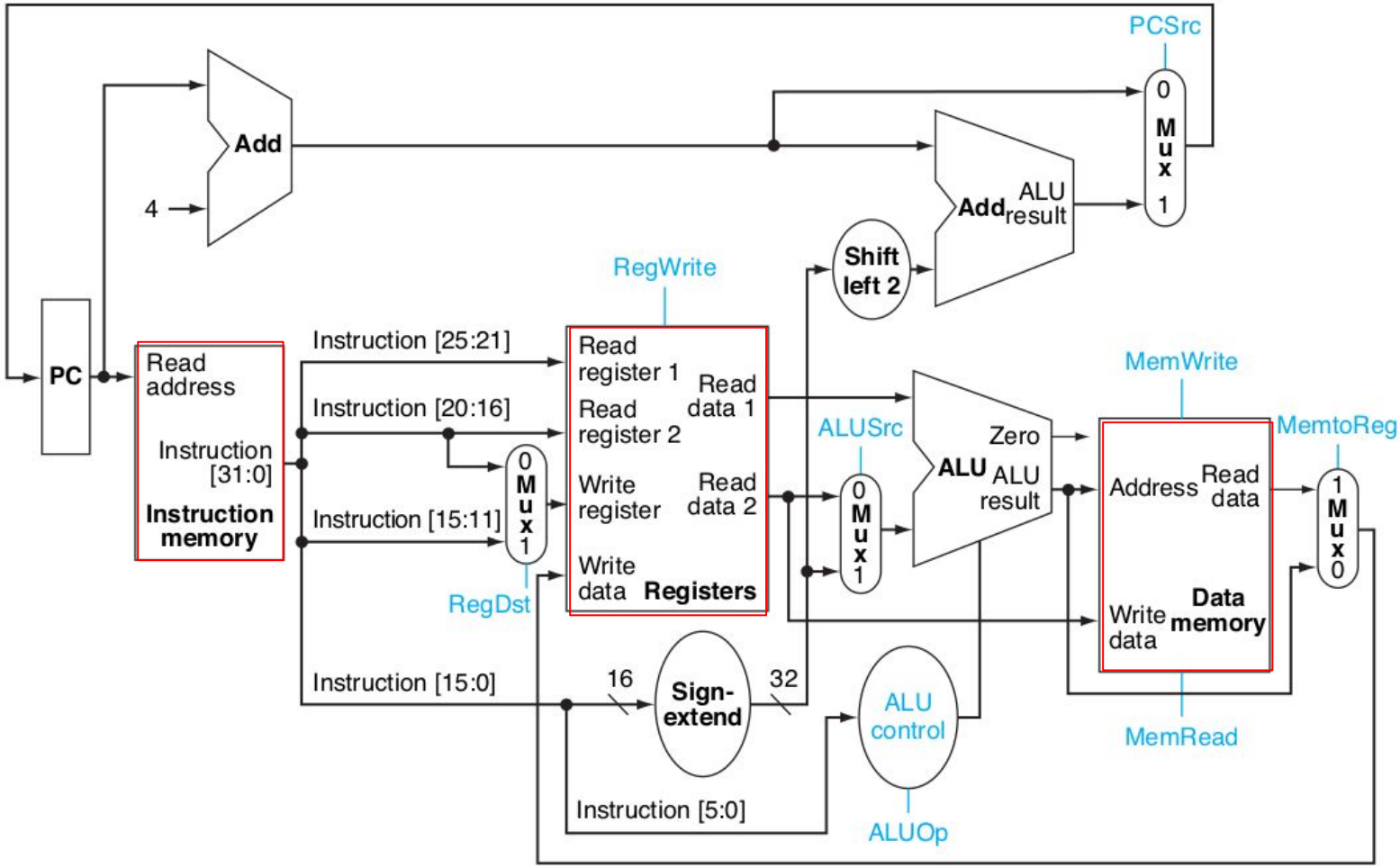


FIGURE 4.15 The datapath of Figure 4.11 with all necessary multiplexors and all control lines identified.

Tecnologia de Memória

Tecnologia de memória	Tempo de acesso típico	US$ por GB em 2012
SRAM	0,5-2,5 ns	$500-$1000
DRAM	50-70 ns	$10-$20
Flash	5.000-50.000 ns	$0,75-$1,00
Disco magnético	5.000.000-20.000.000 ns	$0,05-$0,10

- Memória ideal
  - Tempo de acesso SRAM
  - Capacidade e custo/GB do disco

Princípio da Localidade

- Os programas acessam uma pequena proporção de seu espaço de endereço a qualquer momento
- Localidade temporal
  - Se um item foi referenciado, ele tende a ser novamente referenciado em seguida
    - Ex.: loops, reutilização de operandos
- Localidade espacial
  - Se um item foi referenciado, itens cujo endereço estão próximos tendem a ser referenciados em seguida
    - Ex.: execução do código, acesso a vetores e matrizes

Princípio da Localidade

Exercício 5.1

Neste exercício, veremos as propriedades de localidade de memória do cálculo de matriz. O código a seguir é escrito em C, em que os elementos dentro da mesma linha são armazenados de forma contígua. Suponha que cada palavra seja um inteiro de 32 bits.

a.	for (I=0; I<8; I++)
	for (J=0; J<8000; J++)
	A[I][J]=B[I][0]+A[J][I];

5.1.1 [5] <§5.1> Quantos inteiros de 32 bits podem ser armazenados em uma linha de cache de 16 bytes?

a.	4 inteiros de 32 bits

5.1.2 [5] <§5.1> Referências a quais variáveis exibem localidade temporal?

a.	I, J

5.1.3 [5] <§5.1> Referências a quais variáveis exibem localidade espacial?

a.	A[I][J]

Princípio da Localidade

Matriz A[I,J]								
	J=0	J=1	J=2	J=3	J=4	J=5	...	J=7999
I=0	1	2	3	4	5	6	...	8000
I=1	8001	8002	8003	8004	...			
I=2								
I=3								
I=4								
I=5								
I=7								
I=7								

Memória	
End	32 bits (4 bytes)
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9

Aproveitando a localidade

- Hierarquia de memória
- Armazene tudo no disco
- Copie os itens acessados recentemente (e próximos) do disco para uma memória DRAM menor
  - Memória principal
- Copie os itens acessados mais recentemente (e próximos) da DRAM para uma memória SRAM menor
  - Memória cache anexada à CPU

Níveis de Hierarquia de Memória

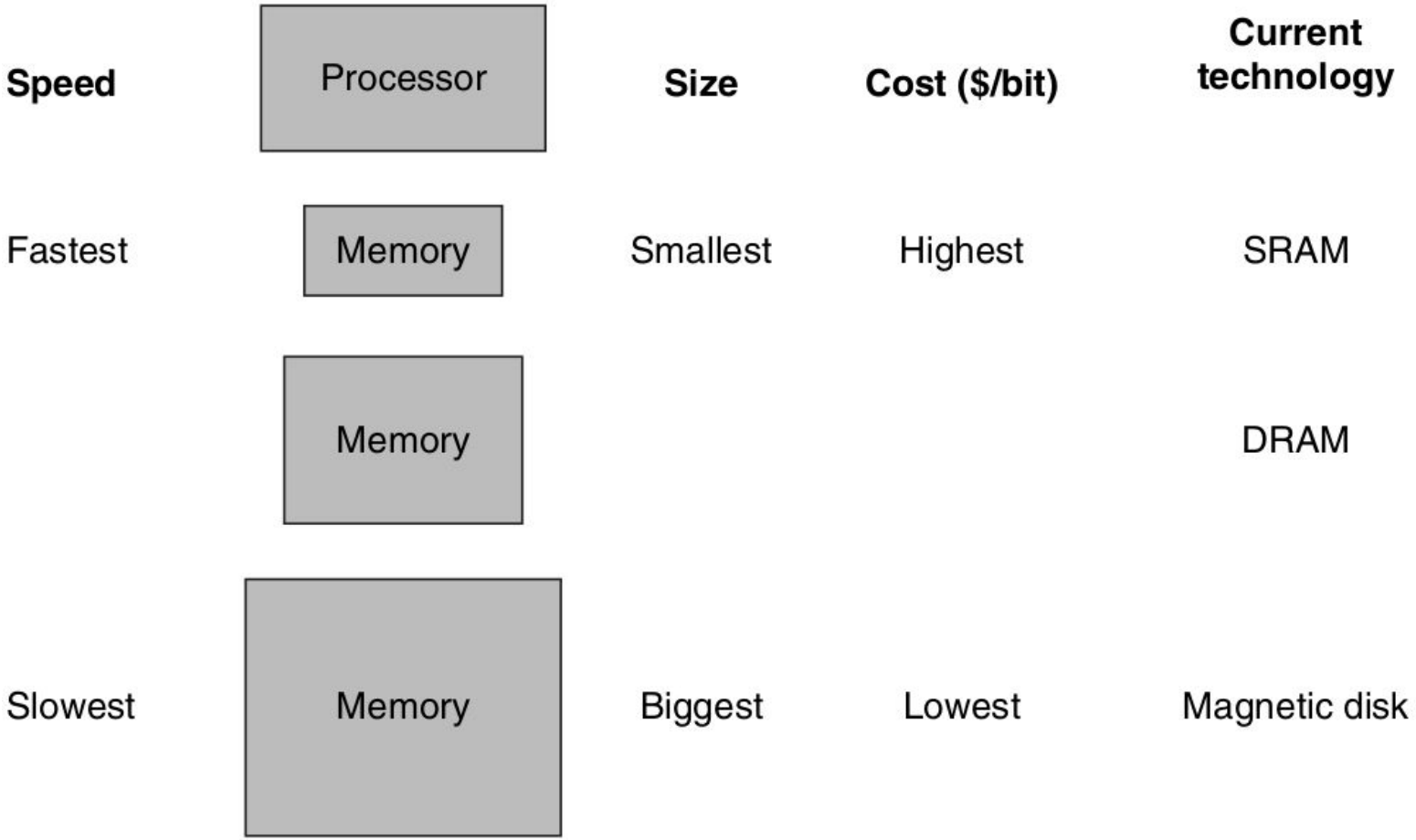
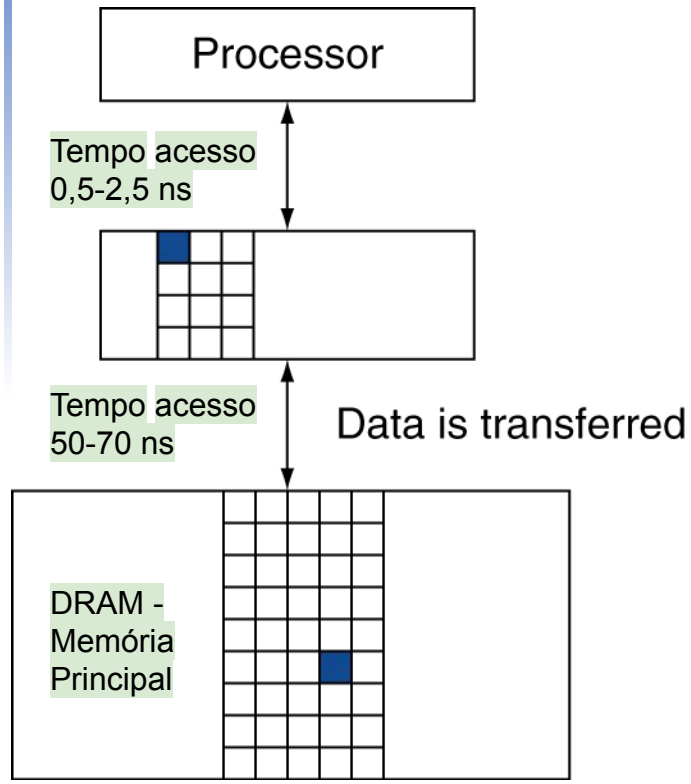


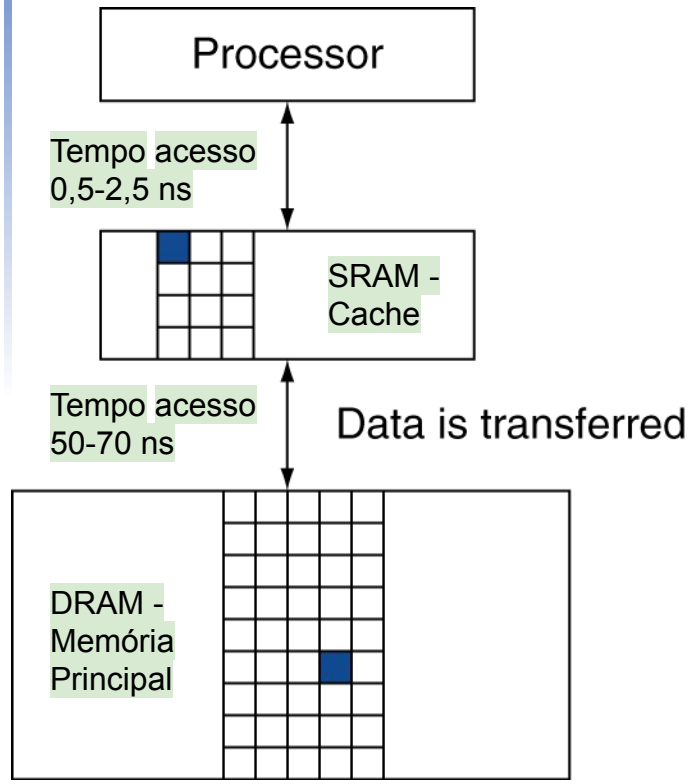
FIGURE 5.1 The basic structure of a memory hierarchy.

Níveis de Hierarquia de Memória



- Bloco (também conhecido como linha): unidade de cópia
  - Podem ser várias palavras
- Acerto (hit): dados são encontrados em algum bloco do nível superior
  - Taxa de acerto: Percentual dos acessos à memória encontrados no nível superior
  - Tempo de acerto: Tempo para acessar o nível superior, que consiste em:
    - Tempo de acesso + Tempo para determinar hit/miss

Níveis de Hierarquia de Memória



- Bloco (também conhecido como linha): unidade de cópia
  - Podem ser várias palavras
- Falha (miss): dados devem ser buscados no nível inferior
  - Taxa de falhas: 1 - Taxa acertos
  - Penalidade por falha: Tempo para trocar o bloco do nível superior, que consiste em:
    - Tempo para trocar o bloco do nível superior + Tempo para entregar o bloco ao processador
  - Tempo acerto < Penalidade por falha

Níveis de Hierarquia de Memória

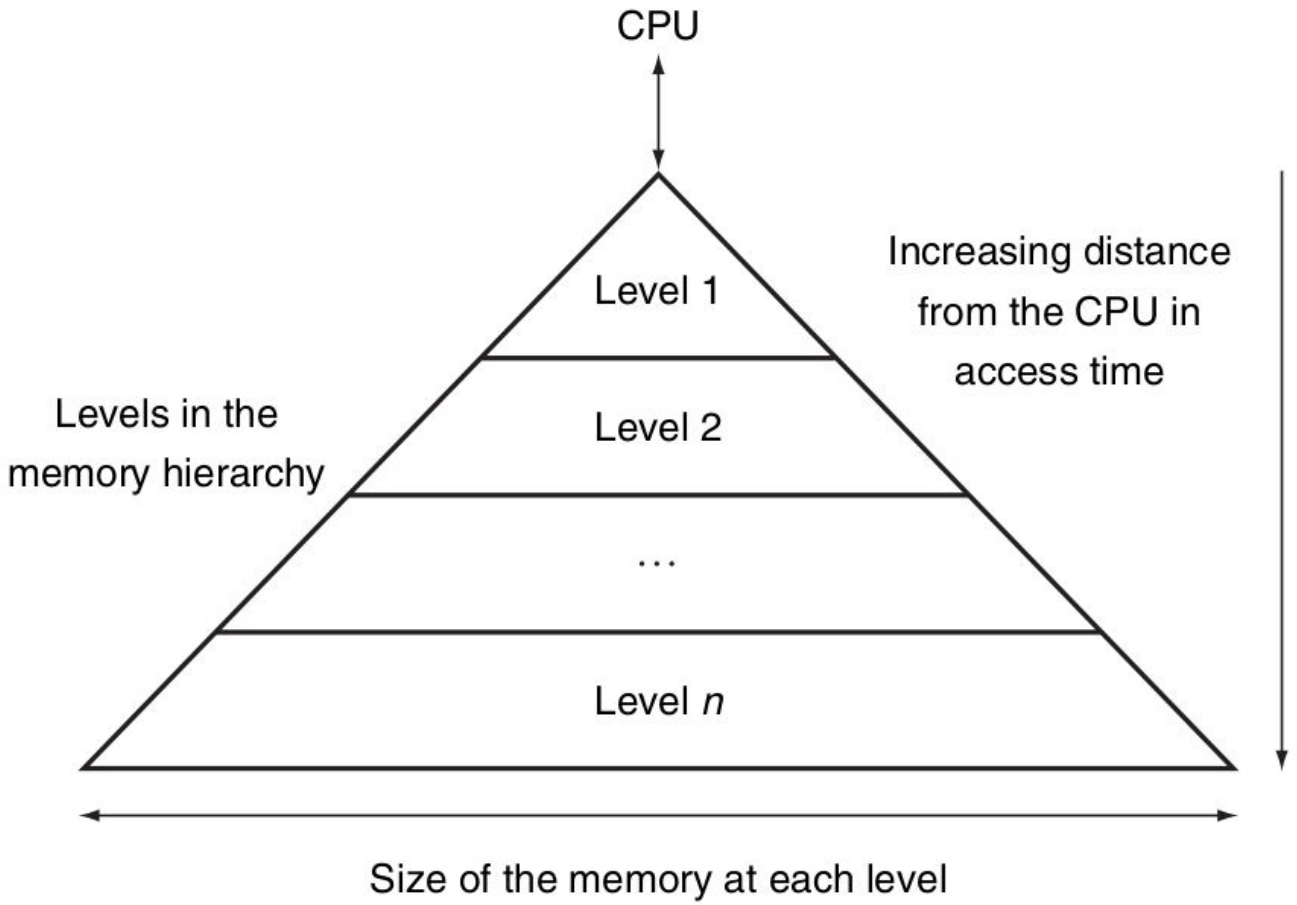
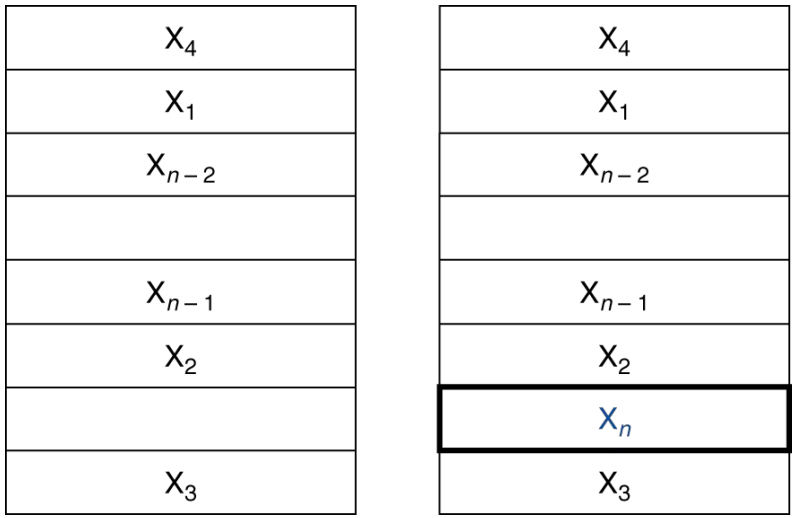


FIGURE 5.3 This diagram shows the structure of a memory hierarchy: as the distance from the processor increases, so does the size.

Memória Cache

- Memória cache
  - O nível da hierarquia de memória mais próximo da CPU
- Acessos dados X1, ..., Xn – 1, Xn

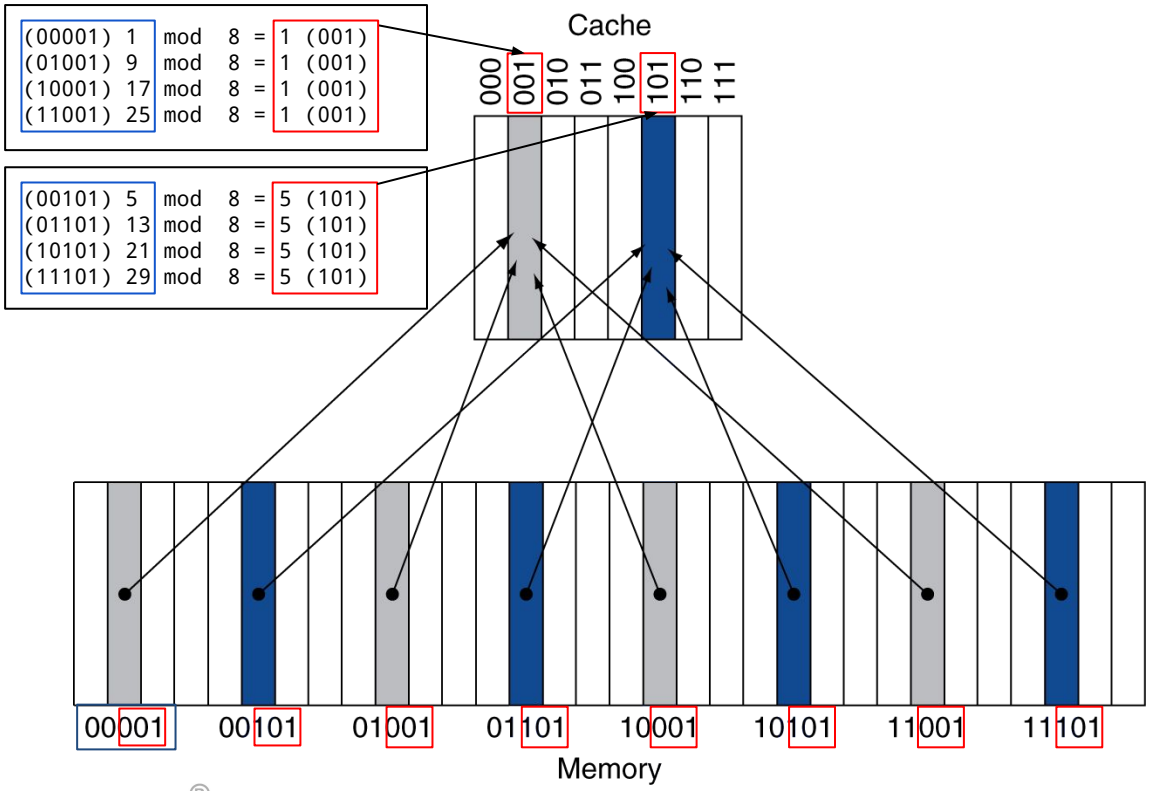


a. Before the reference to X_n b. After the reference to X_n

- Como sabemos se os dados estão presentes?
- Para onde olhamos?

Cache Mapeamento Direto

- Localização determinada pelo endereço
- Mapeado direto: apenas uma escolha
  - (Endereço de bloco) modulo (Número Blocos na cache)



- Número de Blocos é uma potência de 2
- Usa bits de endereço de ordem inferior

Tags e Bit Validade

- Como sabemos qual bloco específico está armazenado em uma posição de cache?
  - Junto com o dado, armazena parte do endereço do bloco
    - Bits de ordem mais alta
    - Chamados de tag
- E se não houver dados na posicção?
  - Bit validade: 1 = presente, 0 = não presente
  - Inicialmente 0

Exemplo Cache

- 8 blocos, 1 palavra/bloco, mapeamento direto
- Estado inicial

Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

A cache inicialmente está vazia, com todos os bits de validade (entrada V da cache) inativos (N).

Exemplo Cache

End. Decimal	End. Binário	Hit/miss	Bloco Cache
22	10 110	Miss	110

O campo tag conterá apenas a parte superior do endereço. O endereço completo de uma palavra contida no bloco de cache i com o campo tag j para essa cache é j × 8 + i ou, de forma equivalente, a concatenação do campo tag j e o campo índice i. Por exemplo, o índice 010 bin possui tag 10 bin e corresponde ao endereço 10110 bin .

Indice	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	**Y**	**10**	**Mem[10110]**
111	N		

O processador requisita os seguintes endereços: 10110 bin (falha)

Exemplo Cache

End. Decimal	End. Binário	Hit/miss	Bloco Cache
26	11 010	Miss	010

Index	V	Tag	Data
000	N		
001	N		
010	**Y**	**11**	**Mem[11010]**
011	N		
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

O processador requisita os seguintes endereços: 11010 bin (falha)

Exemplo Cache

End. Decimal	End. Binário	Hit/miss	Bloco Cache
22	10 110	Hit	110
26	11 010	Hit	010

Index	V	Tag	Data
000	N		
001	N		
010	Y	11	Mem[11010]
011	N		
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

O processador requisita os seguintes endereços: 10110 bin (acerto), 11010 bin (acerto)

Exemplo Cache

End. Decimal	End. Binário	Hit/miss	Bloco Cache
16	10 000	Miss	000
3	00 011	Miss	011
16	10 000	Hit	000

Index	V	Tag	Data
000	Y	10	Mem[10000]
001	N		
010	Y	11	Mem[11010]
011	Y	00	Mem[00011]
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

O processador requisita os seguintes endereços: 10000 bin (falha), 00011 bin (falha), 10000 bin (acerto)

Exemplo Cache

Word addr	Binary addr	Hit/miss	Cache block
18	10 010	Miss	010

Quando o endereço 10010 bin (18) é referenciado, a entrada para o endereço 11010 bin (26) precisa ser substituída, e uma referência a 11010 bin causará uma falha subsequente.

Index	V	Tag	Data
000	Y	10	Mem[10000]
001	N		
010	Y	11	Mem[11010]
011	Y	00	Mem[00011]
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

O processador requisita os seguintes endereços: 10010 bin (falha)

Exemplo Cache

Word addr	Binary addr	Hit/miss	Cache block
18	10 010	Miss	010

Quando o endereço 10010 bin (18) é referenciado, a entrada para o endereço 11010 bin (26) precisa ser substituída, e uma referência a 11010 bin causará uma falha subsequente.

Index	V	Tag	Data
000	Y	10	Mem[10000]
001	N		
010	**Y**	**10**	**Mem[10010]**
011	Y	00	Mem[00011]
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

O processador requisita os seguintes endereços: 10010 bin (falha)

Referências

- Seções 5.1 e 5.2 - Organização e Projeto de Computadores - A Interface Hardware/Software, David A. Patterson & John L. Hennessy, Campus, 4 edição, 2013.
- Seção 5.1- Computer organization and design: the hardware/software interface/David A. Patterson, John L. Hennessy, Elsevier, 5th ed, 2013.