

[Aula 15] Hierarquia de memória 2: *Memória cache*

Prof. João F. Mari
joaof.mari@ufv.br

Aula 15 - Hierarquia de memória 2 - Memória cache

Roteiro

- Mapeando endereços para um bloco de cache multi-palavras
- Blocos maiores, vale a pena?
- Tratando escritas.
- [EX] A cache do processador Instrinsity FastMath

[EX] Mapeando o endereço para um bloco de cache multiword

- Considere uma cache com 64 blocos e um tamanho de bloco de 16 bytes (128 bits ou 4 palavras). Para qual número de bloco o endereço 1200 é mapeado?
 - O bloco é dado pela fórmula:
 - (Endereço do bloco) modulo (Número de blocos na cache)
 - Em que o endereço do bloco é:
 - $\text{Endereço do bloco} = \frac{\text{Endereço (em bytes)}}{\text{Bytes por bloco}} = \frac{1200}{16} = 75$
 - Note que, se (bytes por bloco) é igual a 1 o endereço do bloco é:
 - $\text{Endereço do bloco} = \text{Endereço (em bytes)} = 1200$
 - $75 \bmod 64 = 11$, na verdade, esse bloco mapeia todos os endereços entre 1200 e 1215
- Blocos maiores
 - Exploram a localidade espacial para diminuir as taxas de falhas.

[EX] Mapeando o endereço para um bloco de cache multi-palavras

Memória Principal		...	
Bloco de cache multi-palavra com 16 bytes	palavra ₃	1216	1216 / 16 = 76 % 64 = 12
		1215	
		1214	
		1213	
	palavra ₂	1212	1212 / 16 = 75 % 64 = 11
		1211	
		1210	
		1209	
	palavra ₁	1208	1208 / 16 = 75 % 64 = 11
		1207	
		1206	
		1205	
	palavra ₀	1204	1204 / 16 = 75 % 64 = 11
		1203	
		1202	
		1201	
		1200	1200 / 16 = 75 % 64 = 11
		1199	1199 / 16 = 74 % 64 = 10
		...	

Memória cache						
Índice	V	Tag	Dado 0	Dado 1	Dado 2	Dado 3
63						
62						
...						
13						
12						
11						
10						
9						
...						
1						
0						

Memória cache - Blocos maiores, vale a pena?

- A taxa de falhas pode subir se o tamanho de bloco se tornar uma fração significativa do tamanho da cache.
 - O número de blocos que pode ser armazenado na cache se tornará pequeno
 - Haverá uma grande competição entre blocos
 - É claro, mantendo o tamanho total da cache.
- O custo da falha aumenta na medida que aumenta o tamanho do bloco (maior tempo de transferência)
 - O tempo para buscar o bloco pode ser organizado em duas partes:
 - 1) a latência até a primeira palavra; e
 - 2) o tempo de transferência para o restante do bloco.
 - A segunda parte do tempo aumenta proporcionalmente com o tamanho do bloco.

Memória cache - Blocos maiores, vale a pena?

- A principal desvantagem de aumentar o tamanho do bloco é que a penalidade de falha aumenta
 - **SOLUÇÃO 1) Reinício precoce:**
 - Retornar para o processador quando a palavra solicitada estiver disponível, ao invés de esperar o bloco inteiro
 - O reinício precoce funciona bem para cache de instruções;
 - Execução sequencial é típica.
 - Não funciona bem para cache de dados
 - Requisições são menos previsíveis que a na cache de instruções.
 - **SOLUÇÃO 2) Palavra requisitada primeiro:**
 - Similar ao reinício precoce, mas com acesso direto a palavra requisitada e então transfere o restante do bloco
 - Duas partes: antes e depois da palavra requisitada.
 - Sofre dos mesmos problemas do reinicio precoce.

Memória cache – Tratando escritas

- Uma operação store envia uma palavra para a memória;
 - Garantir que instruções subsequentes que necessitem dessa palavra obtenham o valor atualizado;
 - **Como manter a cache de dados consistente?**
 - **Write-through** (simples, não apresenta bom desempenho):
 - Manter a memória sempre atualizada, sempre escrevendo os dados na memória e na cache;
 - Toda escrita faz com que os dados trafeguem para a memória principal;
 - **[EX]** *Imagine um programa que trabalha com load/stores várias vezes em um vetor, antes de ter uma versão atualizada deste vetor: muitos acessos a memória desnecessários!*
 - Escritas gastam mais de 100 ciclos de máquina
 - De acordo com o SPEC2000, 10% das operações são stores.
 - Para um computador com CPI=1, tem-se $1,0 + 100 \times 10\% = 11$:
 - Redução do desempenho em 11%.

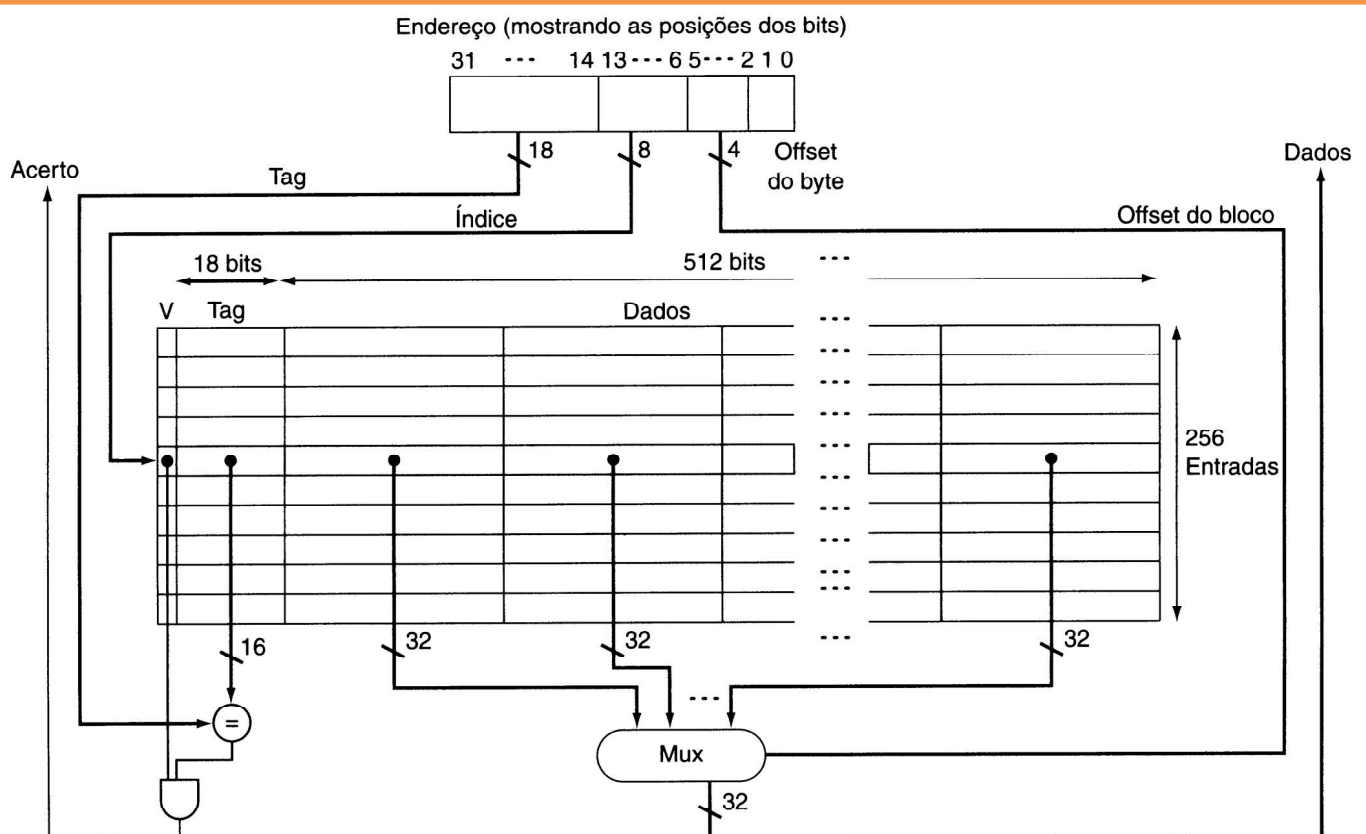
Memória cache – Tratando escritas

- **Write-back:**
 - Alternativa ao *write-through*;
 - Quando ocorre uma escrita, o valor recente é gravado apenas na memória cache;
 - O bloco é escrito em um nível inferior da hierarquia apenas quando é **substituído**;
 - Necessário mais um **bit** na cache para identificar se este bloco deve ser escrito em memória
 - Antes do bloco ser substituído em uma falha.

[EX] Vida real - Intrinsity FastMath

- A cache do processador Intrinsity FastMath
 - Microprocessador embarcado veloz (~2,5 GHz) que usa a arquitetura MIPS e uma implementação de cache simples
 - Pipeline de 12 estágios
 - Caches de instruções e de dados separadas
 - Cada cache tem 16KB, ou 4K palavras, com blocos de 16 palavras.
 - *Write-through* e *write-back*: o sistema operacional pode escolher a política de atualização.
- Cache dividida:
 - Uma cache para os dados e outra para as instruções;
 - Taxa de acertos melhor.

[EX] Vida real - Intrinsity FastMath



Exercício

- Considere um sistema cache multi-palavras:
 - Tamanho da palavra: 8 bits
 - Os endereços também possuem 8 bits:
 - 2^8 ou 256 palavras.
 - A cache possui 16 blocos e cada bloco armazena 2 palavras.
 - Simule as solicitações aos dados nos seguintes endereços.
 - Indique se houve um acerto ou falha.
 - Redesenhe a cache a cada falha.

A – Offset do bloco
B = Índice
C - Tag

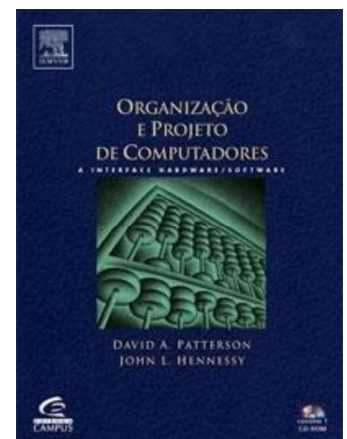
7 – 5	4 – 1	0
C	B	A

#1 – 0001 1101
#2 – 0100 0010
#3 – 0010 0010
#4 – 0001 1100
#5 – 0010 0010
#6 – 0010 1011

Índice	V	Tag	Dado 0	Dado 1
0				
1				
2				
...				
13				
14				
15				

Bibliografia

1. PATTERSON, D.A; HENNESSY, J.L. **Organização e Projeto de Computadores: A Interface Hardware/Software**. 3a. Ed. Elsevier, 2005.
 - Capítulo 7.
2. Notas de aula do prof. Luciano J. Senger:
 - <http://www.ljsenger.net/classroom.html>
3. Notas de aula da profa. Mary Jane Irwin
 - CSE 331 Computer Organization and Design
 - <http://www.cse.psu.edu/research/mdl/mji/mjicourses>



FIM

- FIM:
 - **Aula 15** – Hierarquia de memória 2 – Memória cache.
- Próxima aula:
 - **Aula 16** – Hierarquia de memória 3 – Memória virtual.