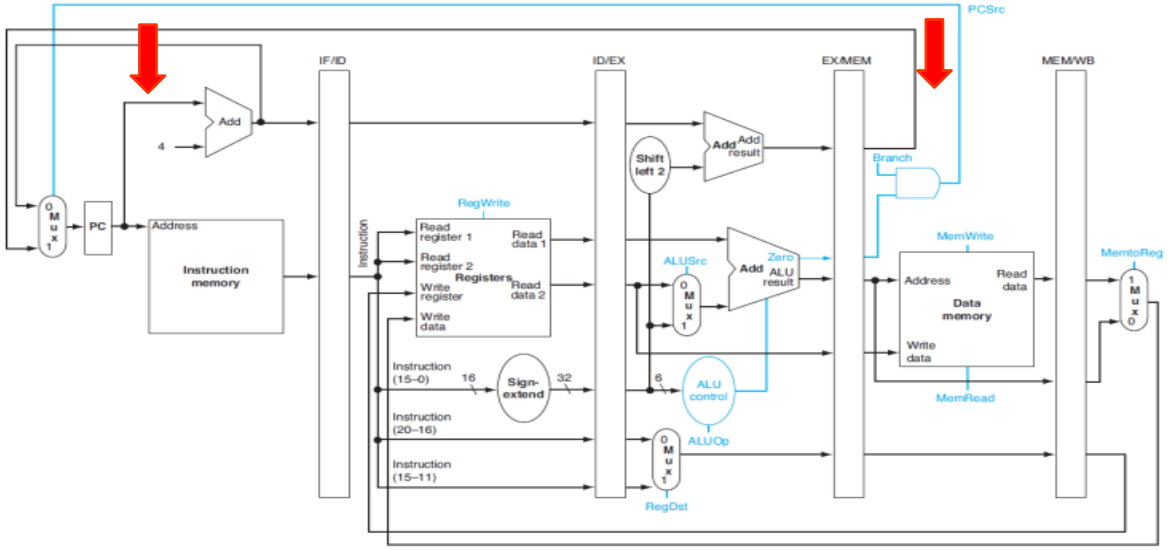
Memória Cache: Controle

Anotações do material suplementar (apresentações PPT) ao Livro do Hennessy e Patterson e do material do Prof. Celso Alberto Saibel Santos (DI/CT).

Implementação do controle de cache

- Nas CPUs projetadas anteriormente (monociclo e *pipeline*), bastaria trocar as memórias de instruções e dados dos esquemas por caches de instruções e dados



Controle de LEITURAS

Unidade de controle (*hardware*) separada para memória-cache atuando junto ao controle da CPU pipeline. Controle é similar para caches de instruções e de dados trivial com *hits* e mais complexo com *misses*:

Hits: como se não houvesse cache;

Misses: "parar o *pipeline*", similar a um conflito. O tratamento é mais simples: congelar os registradores para esperar a leitura da memória e recuperar a falha;

No caso de falha (miss) na leitura de uma instrução:

1. Enviar o valor de PC (PC – 4) a ser lido na memória
2. Solicitar que a memória leia o valor e esperar o resultado
3. Escrever a entrada da cache correspondente
4. Reiniciar a execução da instrução, com sua leitura a partir da cache (agora, será um hit!)

Controle de ESCRITAS

Não é tão simples em se tratando de caches (hierarquia de memória):

Prob1: ***Store* na cache? E o valor na memória?**

- Deve-se assegurar que o dado escrito na cache também será escrito na memória. Do contrário, memória e cache podem se tornar **inconsistentes**

Prob2: ***Write misses***

- (1) Buscar o bloco na memória e colocá-lo na cache, (2) reescrever o bloco que causou a falha de cache e (3) recopiar o bloco alterado para a memória principal

Tratamento de ESCRITAS

Solução mais simples: *write-through*

- Memória e cache são escritas ao mesmo tempo, mas se for necessário uma espera para isso, os valores da cache poderão ser perdidos!
  - Usar uma fila no caminho de escrita para a memória;
  - *Buffers de escrita* permitem “equalizar” o serviço quando a taxa de chegada de requisições é variável;

Escrita apenas no retorno: *write-back*

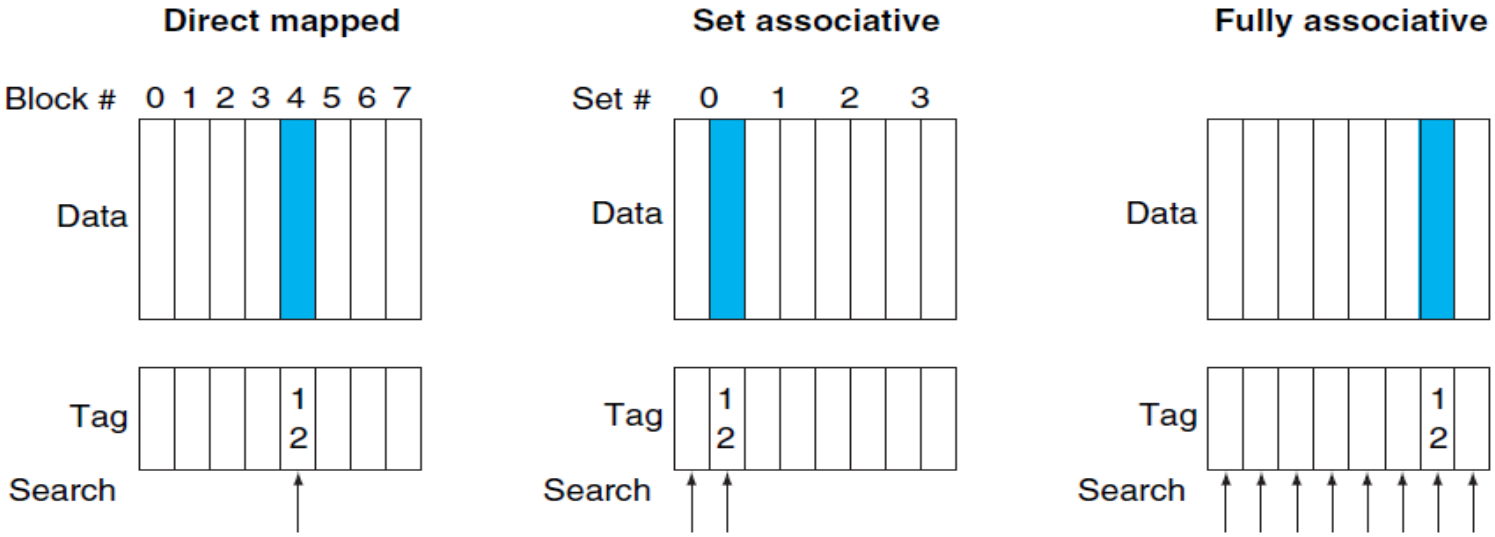
- Atualização quando o bloco for substituído e se houver ocorrido alteração;
- O bloco é reescrito no nível inferior da hierarquia quando é trocado na cache;

Melhorando o desempenho:

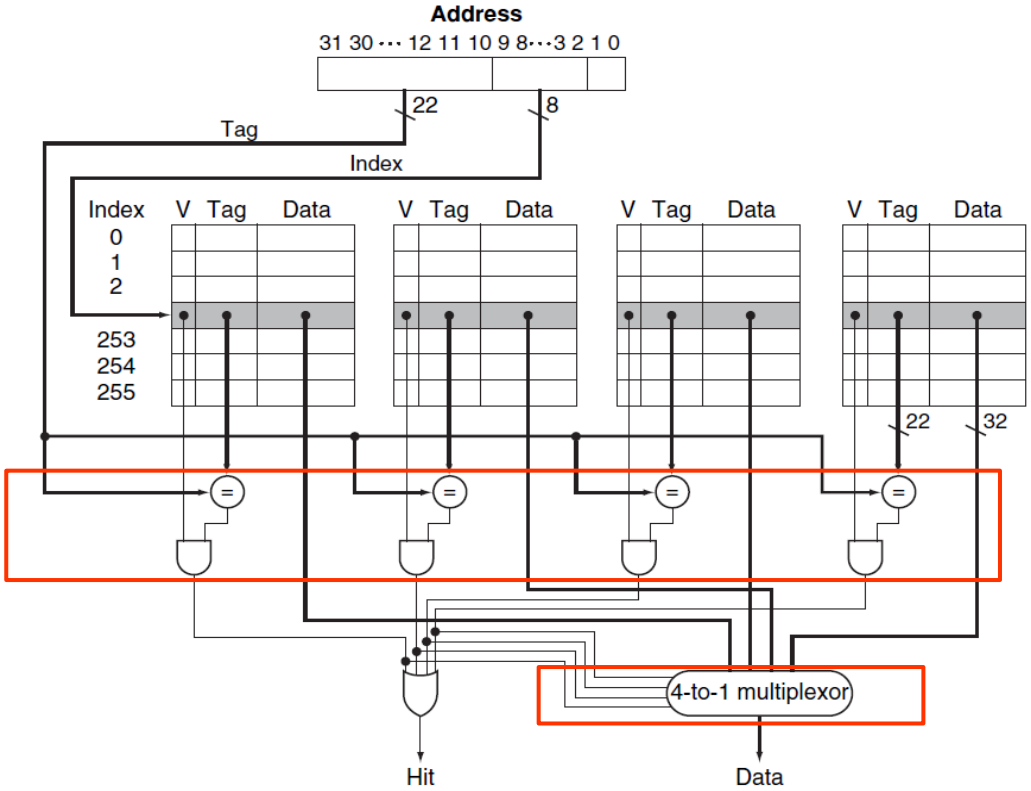
Outras formas de posicionamento dos blocos além do mapeamento direto

Posicionamento de blocos na cache

1. Conjunto (*set*) associativo; totalmente (*fully*) associativo
2. Mapeamento Direto → *one-way* associativo



Cache 4-way associativa



Desempenho x associatividade

Associativity	Data miss rate
1	10.3%
2	8.6%
4	8.3%
8	8.1%

FIGURE 5.15 The data cache miss rates for an organization like the Intrinsity FastMATH processor for SPEC CPU2000 benchmarks with associativity varying from one-way to eight-way. These results for 10 SPEC CPU2000 programs are from Hennessy and Patterson [2003].

Associatividade x Taxa de falhas

Em geral: aumento do grau de associatividade → redução da taxa de faltas

Desvantagem: mais tempo de tratamento nos acertos

Exemplo (Hennessy & Patterson, p.500, 3ª Ed):

Três pequenas caches, cada uma com 4 blocos/1 palavra:

1. Mapeamento direto
2. Mapeamento associativo por conjunto de 2 posições
3. Totalmente associativo (por conjunto de 4 posições)

Achar o número de faltas em cada uma, considerando a sequência de endereços de blocos: 0, 8, 0, 6, 8.

Cache com Mapeamento Direto

Endereço do Bloco	Bloco da Cache
0	(0 módulo 4)= 0
6	(6 módulo 4)= 2
8	(8 módulo 4)= 0

Endereço do bloco de memória acessado	Acerto ou falta	Conteúdo dos Blocos da Cache Após Referência			
		0	1	2	3
0	falta	**Mem[0]**			
8	falta	**Mem[8]**			
0	falta	**Mem[0]**			
6	falta	Mem[0]		**Mem[6]**	
8	falta	**Mem[8]**		Mem[6]	

Conclusão: 5 faltas para 5 referências

Cache associativa por conjunto 2-way

Endereço do Bloco	Conjunto da Cache
0	(0 módulo 2)= 0
6	(6 módulo 2)= 0
8	(8 módulo 2)= 0

Mem[8] foi escolhido para ser substituido (LRU)

Endereço do bloco de memória acessado	Acerto ou falha	Conteúdo dos Blocos da Cache Após Referência			
		Conjunto 0		Conjunto 1	
0	falta	**Mem[0]**			
8	falta	Mem[0]	**Mem[8]**		
0	acerto	Mem[0]	Mem[8]		
6	falta	Mem[0]	**Mem[6]**		
8	falta	**Mem[8]**	Mem[6]		

Conclusão: 4 faltas para 5 referências

Cache totalmente associativa

Endereço do bloco de memória acessado	Acerto ou falha	Conteúdo dos Blocos da Cache Após Referência			
		Bloco 0	Bloco 1	Bloco 2	Bloco 3
0	falta	**Mem[0]**			
8	falta	Mem[0]	**Mem[8]**		
0	acerto	Mem[0]	Mem[8]		
6	falta	Mem[0]	Mem[8]	**Mem[6]**	
8	acerto	Mem[0]	Mem[8]	Mem[6]	

Conclusão: 3 faltas para 5 referências

Para a cache anterior...

2 bits [0:1] → byte dentro da palavra

8 bits [2:9] → índice da cache (número de entradas)

22 bits [10:31] → rótulo (tag) armazenado na cache

Tamanho da cache:

Cada linha de um conjunto tem: 32 bits (1 bloco possui uma palavra) + 22 (tag) + 1 (bit de validade) = 55 bits

A cache tem 2^8 = 256 entradas

4 conjuntos

Tamanho total = 55 x 4 x 256 = 55 Kbits

- Note que apenas 32Kbits = 32 x 4 x 256 guardam informação

Cache 4-way associativa