Universidade Federal do Rio Grande do Sul
Instituto de Informática

Organização de Computadores

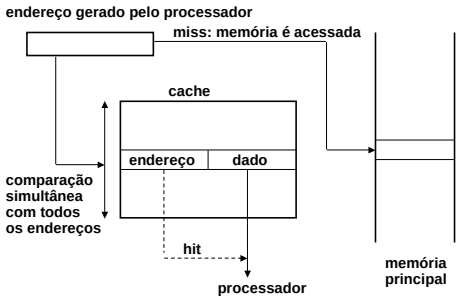
Aula 17

Memória cache
segunda parte

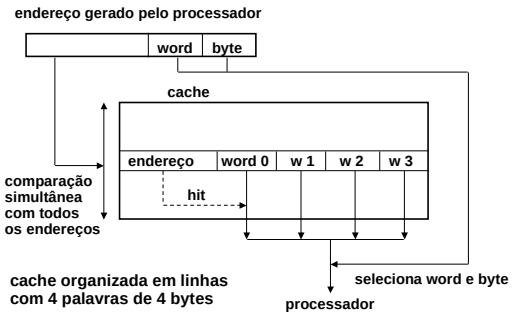
Memória cache
segunda parte

1. Mapeamento completamente associativo
2. Mapeamento direto
3. Mapeamento conjunto - associativo

1. Mapeamento completamente associativo



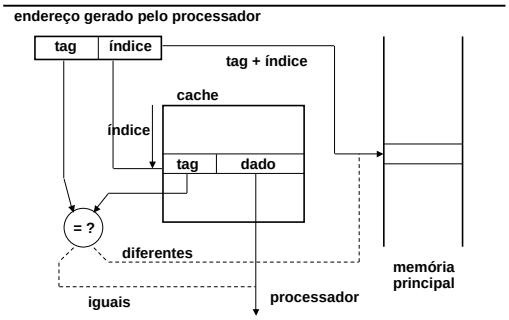
Mapeamento completamente associativo



Mapeamento completamente associativo

- **vantagem: máxima flexibilidade no posicionamento de qualquer palavra (ou linha) da memória principal em qualquer palavra (ou linha) da cache**
- **desvantagens**
  - **custo em hardware da comparação simultânea de todos os endereços armazenados na cache**
  - **algoritmo de substituição (em hardware) para selecionar uma linha da cache como conseqüência de um miss**
- **utilizado apenas em memórias associativas de pequeno tamanho**
  - **tabelas**

2. Mapeamento direto

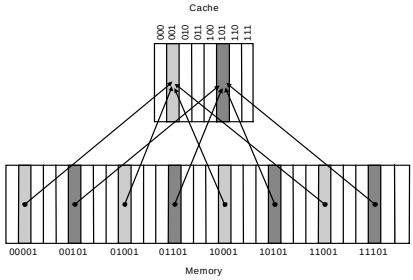


Mapeamento direto

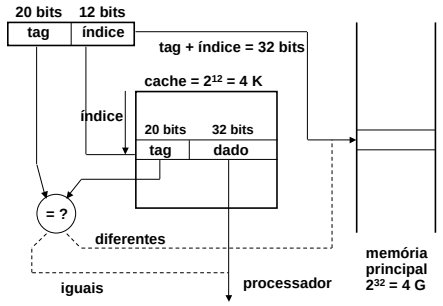
- **endereço é dividido em 2 partes**
  - **parte menos significativa: índice, usado como endereço na cache onde será armazenada a palavra**
  - **parte mais significativa: tag, armazenado na cache junto com o conteúdo da posição de memória**
- **quando acesso é feito, índice é usado para encontrar palavra na cache**
  - **se tag armazenado na palavra da cache é igual ao tag do endereço procurado, então houve hit**
- **endereços com mesmo índice são mapeados sempre para a mesma palavra da cache**

Mapeamento direto

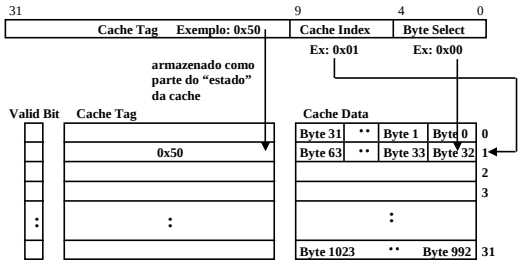
- **mapeamento: endereço é o módulo do número de blocos na cache**



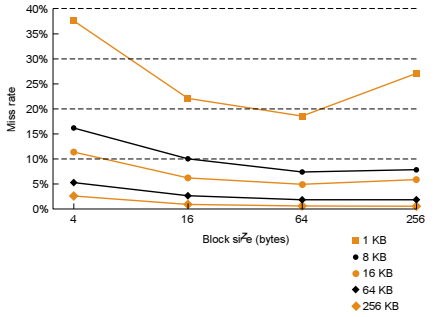
Mapeamento direto – exemplo



Mapeamento direto – uso de linhas



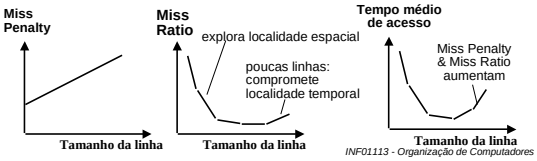
Tamanho da linha x *miss ratio*



Tamanho da linha

- em geral, uma linha maior aproveita melhor a localidade espacial MAS
  - linha maior significa maior *miss penalty*
    - demora mais tempo para preencher a linha
  - se tamanho da linha é grande demais em relação ao tamanho da cache, *miss ratio* vai aumentar
    - muito poucas linhas
- em geral, tempo médio de acesso =

  Hit Time x (1 - Miss Ratio) + Miss Penalty x Miss Ratio



Quantos bits tem a cache no total?

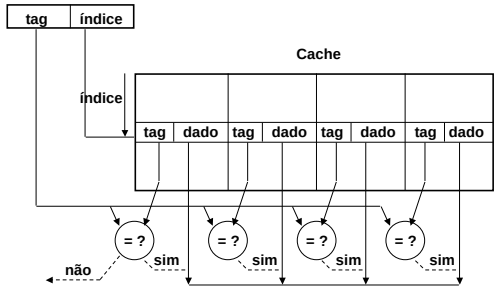
- supondo cache com mapeamento direto, com 64 KB de dados, linha com uma palavra, endereços de 32 bits
- 64 KB -> 16 Kpalavras, 2^{14} palavras, neste caso 2^{14} linhas
- cada linha tem 32 bits de dados mais um tag (32-14-2 bits) mais um bit de validade:

  2^{14} x (32 + 32 -14 -2 +1) = 2^{14} x 49 = 784 x 2^{10} = 784 Kbits
- 98 KB para 64 KB de dados, ou 50% a mais

Mapeamento direto

- vantagens
  - não há necessidade de algoritmo de substituição
  - hardware simples e de baixo custo
  - alta velocidade de operação
- desvantagens
  - desempenho cai se acessos consecutivos são feitos a palavras com mesmo índice
  - *hit ratio* inferior ao de caches com mapeamento associativo
- demonstra-se no entanto que *hit ratio* aumenta com o aumento da cache, aproximando-se de caches com mapeamento associativo
  - tendência atual é de uso de caches grandes

3. Mapeamento conjunto – associativo



Mapeamento conjunto – associativo

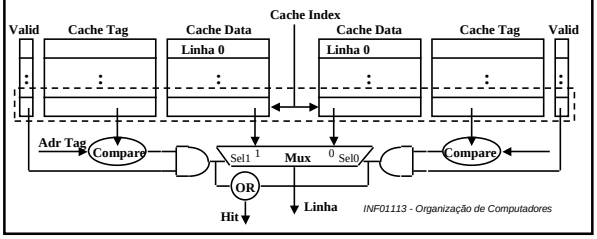
- mapeamento direto: todas as palavras armazenadas na cache devem ter índices diferentes
- mapeamento associativo: linhas podem ser colocadas em qualquer posição da cache
- compromisso: um nº limitado de linhas, de mesmo índice mas diferentes tags, podem estar na cache ao mesmo tempo (num mesmo conjunto)
- nº de linhas no conjunto = associatividade

Mapeamento conjunto – associativo

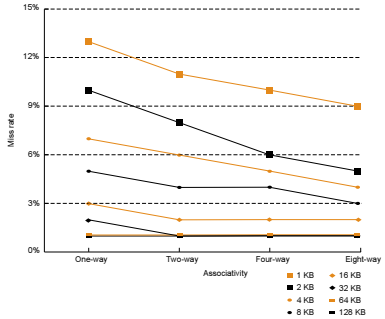
- vantagem em relação ao mapeamento completamente associativo: comparadores são compartilhados por todos os conjuntos
- algoritmo de substituição só precisa considerar linhas dentro de um conjunto
- muito utilizado em microprocessadores
  - Motorola 68040: 4-way set associative
  - Intel 486: 4-way set associative
  - Pentium: 2-way set associative

Desvantagem da cache conjunto-associativo

- conjunto-associativa N-way X mapeamento direto
  - dado tem atraso extra do multiplexador
  - dado vem DEPOIS da decisão *Hit/Miss* e da seleção do conjunto
- numa cache com mapeamento direto, linha da cache está disponível ANTES da decisão *Hit/Miss*
  - possível assumir um *hit* e continuar. Recuperar depois se for *miss*.



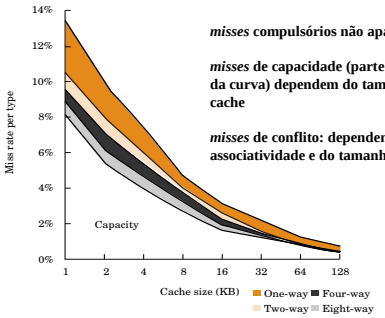
Impacto da associatividade da cache



Fontes de misses

- compulsórios (*cold start* ou chaveamento de processos, primeira referência): primeiro acesso a uma linha
  - é um "fato da vida": não se pode fazer muito a respeito
  - se o programa vai executar "bilhões" de instruções, *misses* compulsórios são insignificantes
- de conflito (ou colisão)
  - múltiplas linhas de memória acessando o mesmo conjunto da cache conjunto-associativa ou mesma linha da cache com mapeamento direto
  - solução 1: aumentar tamanho da cache
  - solução 2: aumentar associatividade
- de capacidade
  - cache não pode conter todas as linhas accessadas pelo programa
  - solução: aumentar tamanho da cache
- invalidação: outro processo (p.ex. I/O) atualiza memória

Fontes de *misses*



***misses* compulsórios não aparecem**

***misses* de capacidade (parte "branca" da curva) dependem do tamanho da cache**

***misses* de conflito: dependem da associatividade e do tamanho da cache**

Quantidade de *misses* segundo a fonte

	Mapeam. direto	**Conj.-associat. N-way**	**Complet. associativa**
Tamanho da cache	Grande	Médio	Pequeno
***Misses* compulsórios**	Mesmo	Mesmo	Mesmo
***Misses* de conflito**	Alto	Médio	Zero
***Misses* de capacidade**	Baixo	Médio	Alto
***Misses* de invalidação**	Mesmo	Mesmo	Mesmo