Memória Cache

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Introdução

- SRAM
  - Alta velocidade
  - Baixa densidade
  - Alto custo
  - Alto consumo
- DRAM
  - Baixa velocidade
  - Alta densidade
  - Baixo custo
  - Baixo consumo
  - Necessita *refresh*

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Gap CPU × Memória

- Primeiros microcomputadores: SRAMs
- Microcomputadores de segunda geração: DRAMs
  - Os primeiros XTs: 1 *wait-state*
  - Os últimos XTs: 0 *wait-state*
- Primeiros 80386: FPM-RAMs
- A partir daí o *clock* do barramento das CPUs aumentou e o tempo de acesso das DRAMs não acompanhou
  - Muitos *wait-states* tiram a vantagem de um alto *clock* de CPU
  - Utilizar apenas SRAMs seria muito caro

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Memória Cache

- Localidade espacial e temporal
  - DLLs reduzem a localidade espacial e aumenta a temporal
  - Aspectos RISC $\times$ CISC (discussão obsoleta)
- Pequena quantidade de SRAM
- Grande quantidade de DRAM
- A idéia é semelhante à memória virtual, porém totalmente implementada por *hardware*
- Pode-se pensar em uma hierarquia de memória
  - Cache (SRAM)
  - Memória Principal (DRAM)
  - Memória Secundária (disco)

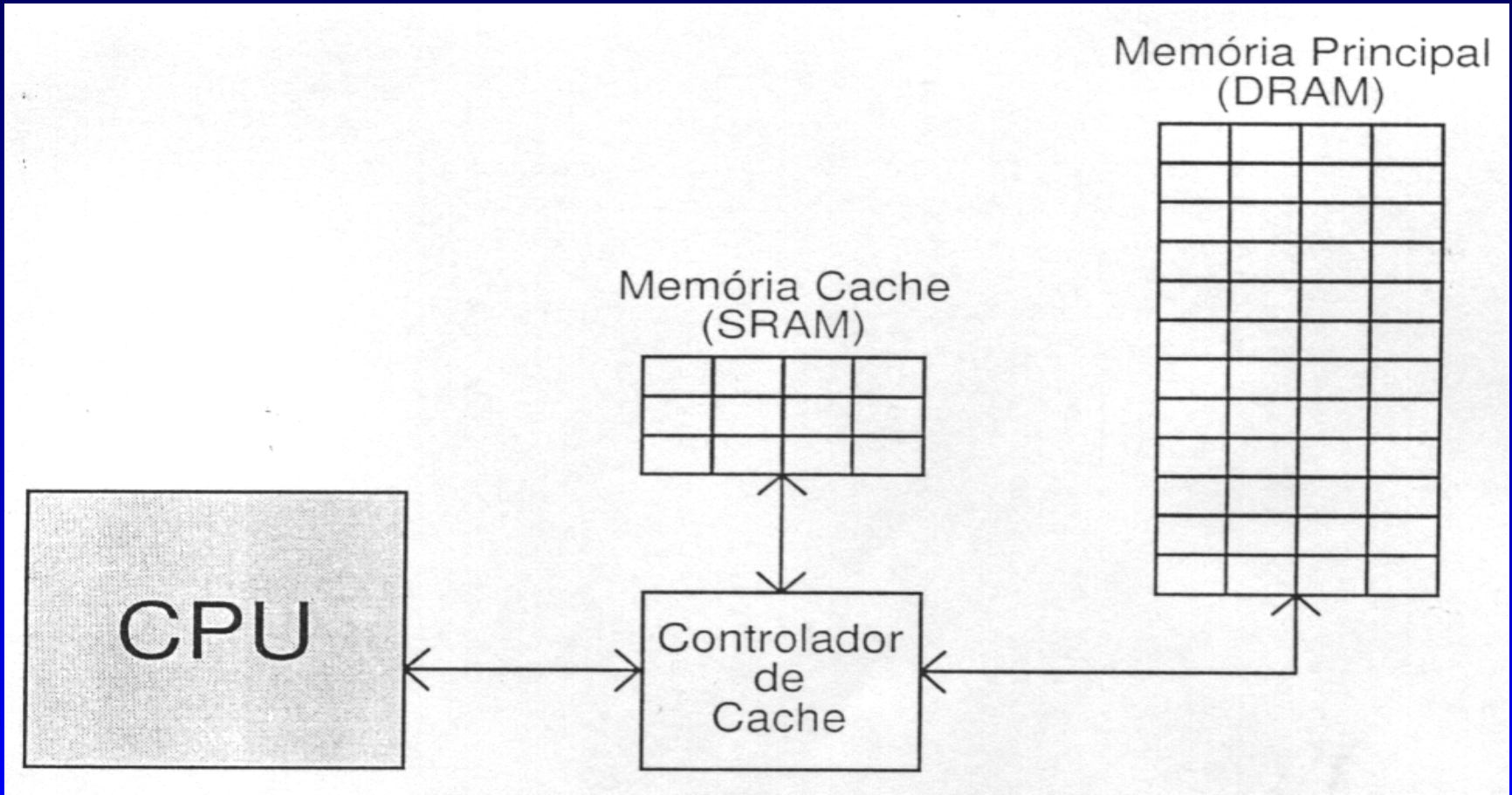
UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Cache × Memória Virtual

- Memória virtual
  - Implementada pelo S. O.
  - Requer suporte do *hardware*
  - Envolve uma tradução de endereços
  - O objetivo é aumentar a memória visível pelos programas
- Memória Cache
  - Totalmente implementada em *hardware*
  - Transparente para o *software*
  - Os dados do cache são apenas uma cópia da memória principal
  - O objetivo é reduzir o tempo de acesso à memória

Controlador de Cache

- *Cache hit*
- *Cache fault*



UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Controlador de Cache

- Determina se um endereço está no cache
- Acessa o cache em caso de acerto
- Move a linha de cache da memória principal para o cache em caso de falha
- Gerencia os acessos entre cache e memória principal

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Questões sobre Cache

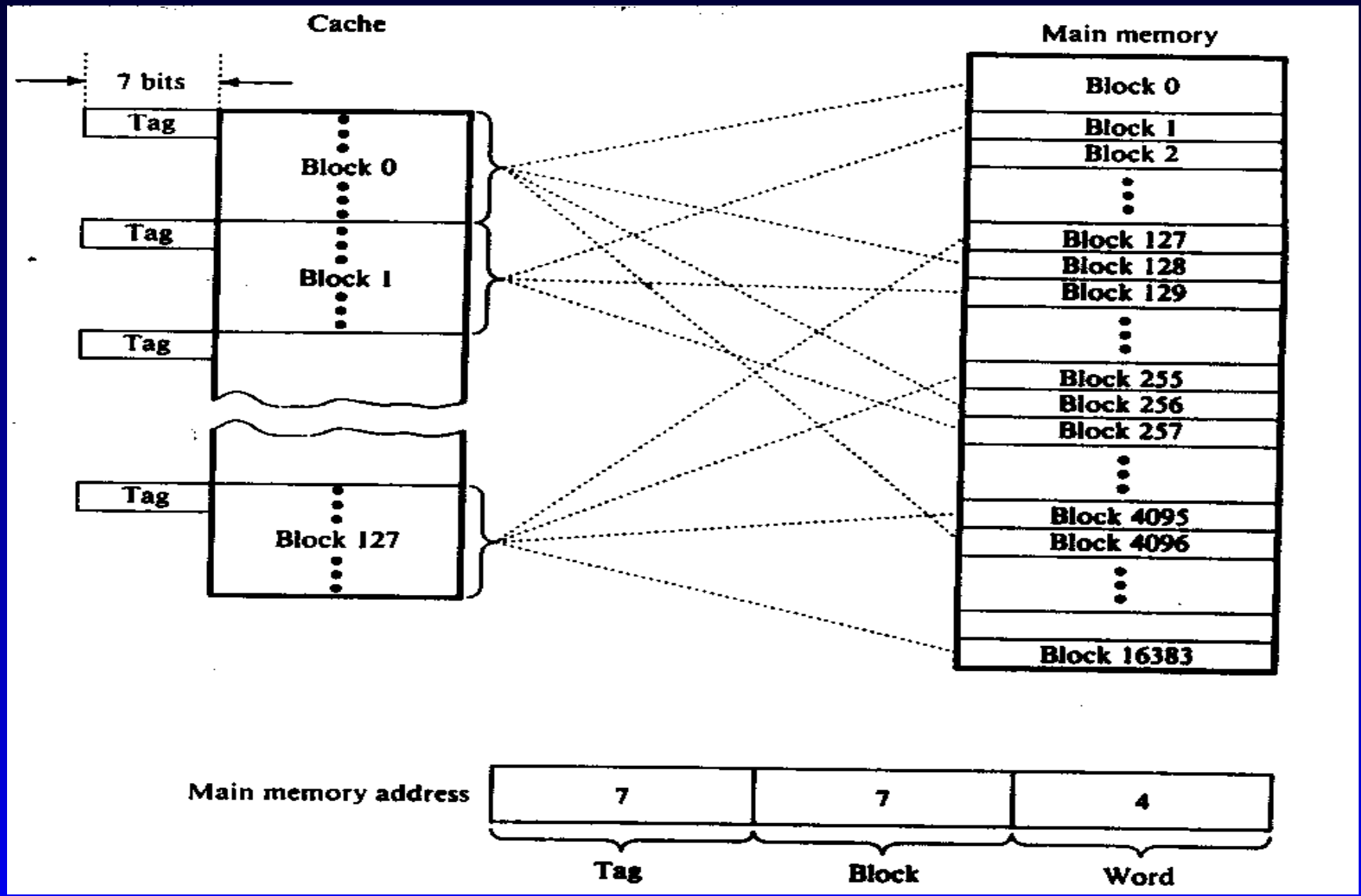
- Organização
- Política de substituição
- Políticas de leitura e escrita
- Endereços virtuais ou físicos
- Consistência
- Unificado ou particionado
- Tamanho
  - do cache
  - da linha
  - do conjunto

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Organização de Cache

- Cache mapeado diretamente
- Cache totalmente associativo
- Cache mapeado associativo por conjunto
- Cache mapeado por setor

Cache Mapeado Diretamente

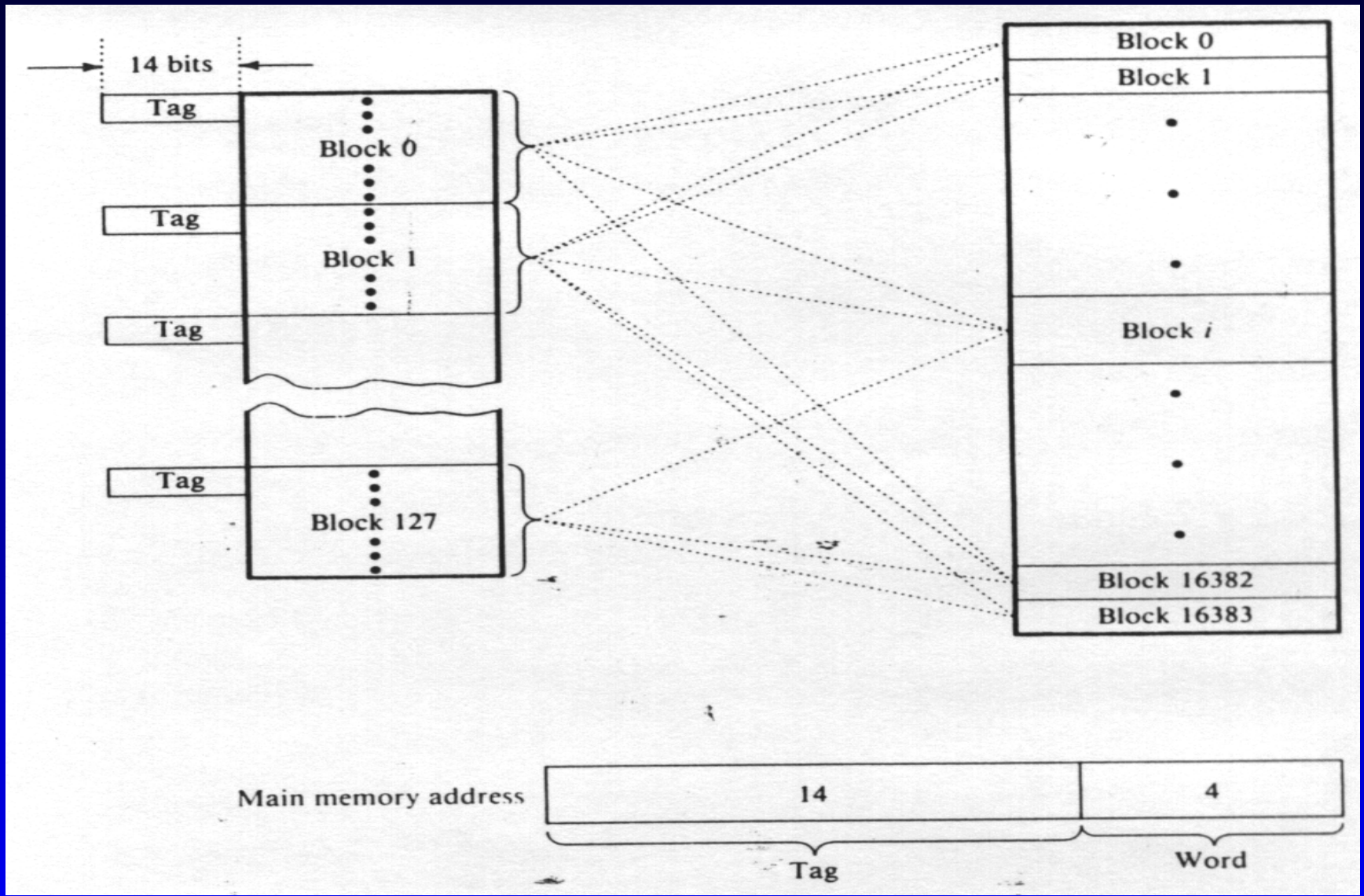


UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Cache Mapeado Diretamente

- Acesso simultâneo ao *tag* e aos dados
- Não é necessária memória associativa
- Algoritmo de substituição é trivial
- Taxa de acerto é baixa se dois (ou mais) blocos mapeados no mesmo slot são utilizados alternadamente
  - Preocupante em sistemas multiprocessadores com cache unificado

Cache Totalmente Associativo

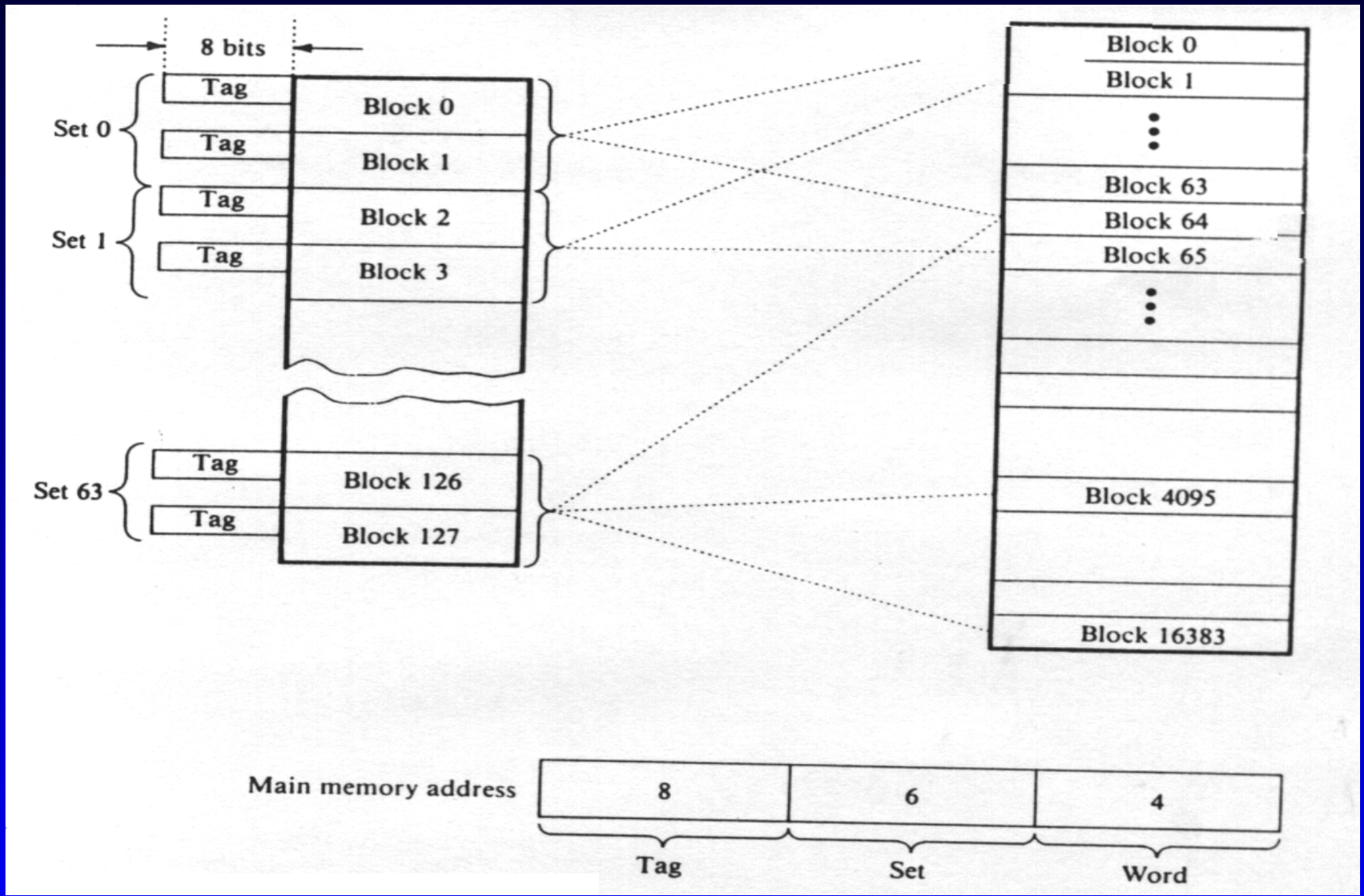


UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Cache Totalmente Associativo

- Alto desempenho e custo
- A flexibilidade de mapeamento permite implementar diversos algoritmos de substituição
- A contenção de blocos é pequena
- Utilizado na TLB do sistema de memória virtual

Cache Associativo por Conjunto

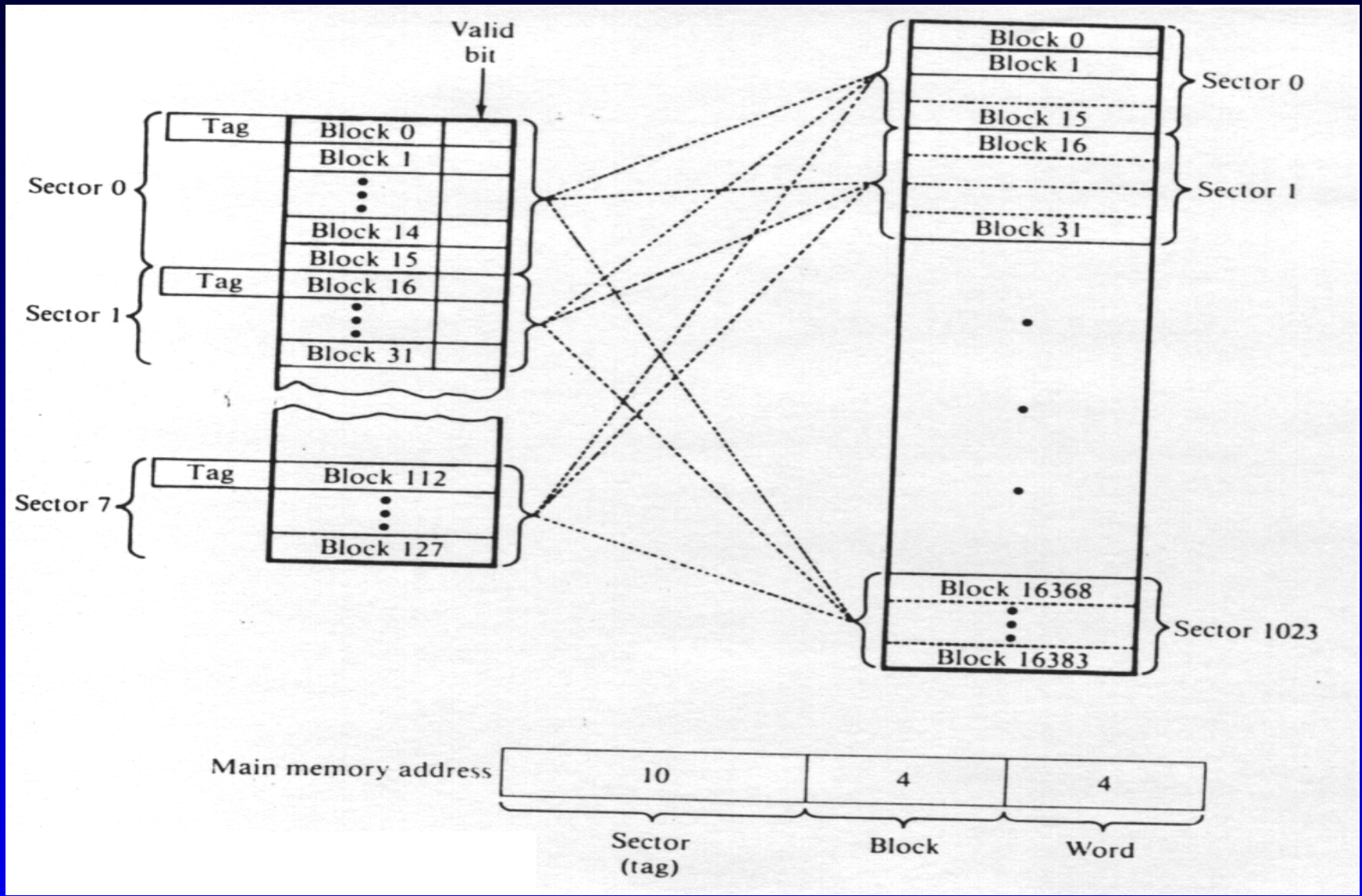


UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Cache Associativo por Conjunto

- Compromisso entre mapeamento direto e totalmente associativo
- Reduz o tamanho e o custo da compração associativa
- Usado pela maioria das CPUs atuais

Cache Mapeado por Setor



UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Cache Mapeado por Setor

- Setores podem ser carregados em qualquer slot
- O mapeamento de setores é congruente
  - Em um slot de setor só podem ser carregados blocos do mesmo setor
- Apenas o bloco que causou a falha é carregado, os demais são marcados inválidos

Política de Substituição

- Utiliza-se políticas simples
- Implementação em *hardware*
- *First-in, first-out* (FIFO)
- *Least used*
- *Least recently used* (LRU)
  - Na média, é o que funciona melhor
- Aleatório

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Política de Acesso à Memória

- Leitura
  - Se acerto os dados são enviados à CPU
  - *Load-through*
  - Preenche a linha e envia
- Escrita
  - *Write-through*
    - *Write-allocate*
    - *No-write-allocate* (*write-protected*)
  - *Write-back*
    - *Simple write-back*
    - *Flagged write-back*
    - *Flagged register write-back*

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Endereços Virtuais ou Físicos

- Se o cache funciona com endereços virtuais, o mapeamento entre endereços virtuais e endereços físicos é evitado em caso de acerto no cache
- Mapeamento não unívoco
  - Endereços virtuais diferentes podem corresponder ao mesmo endereço físico
  - O mesmo endereço virtual em programas diferentes pode corresponder a endereços físicos diferentes
- É necessário manter a consistência utilizando uma TLB inversa

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Consistência do Cache

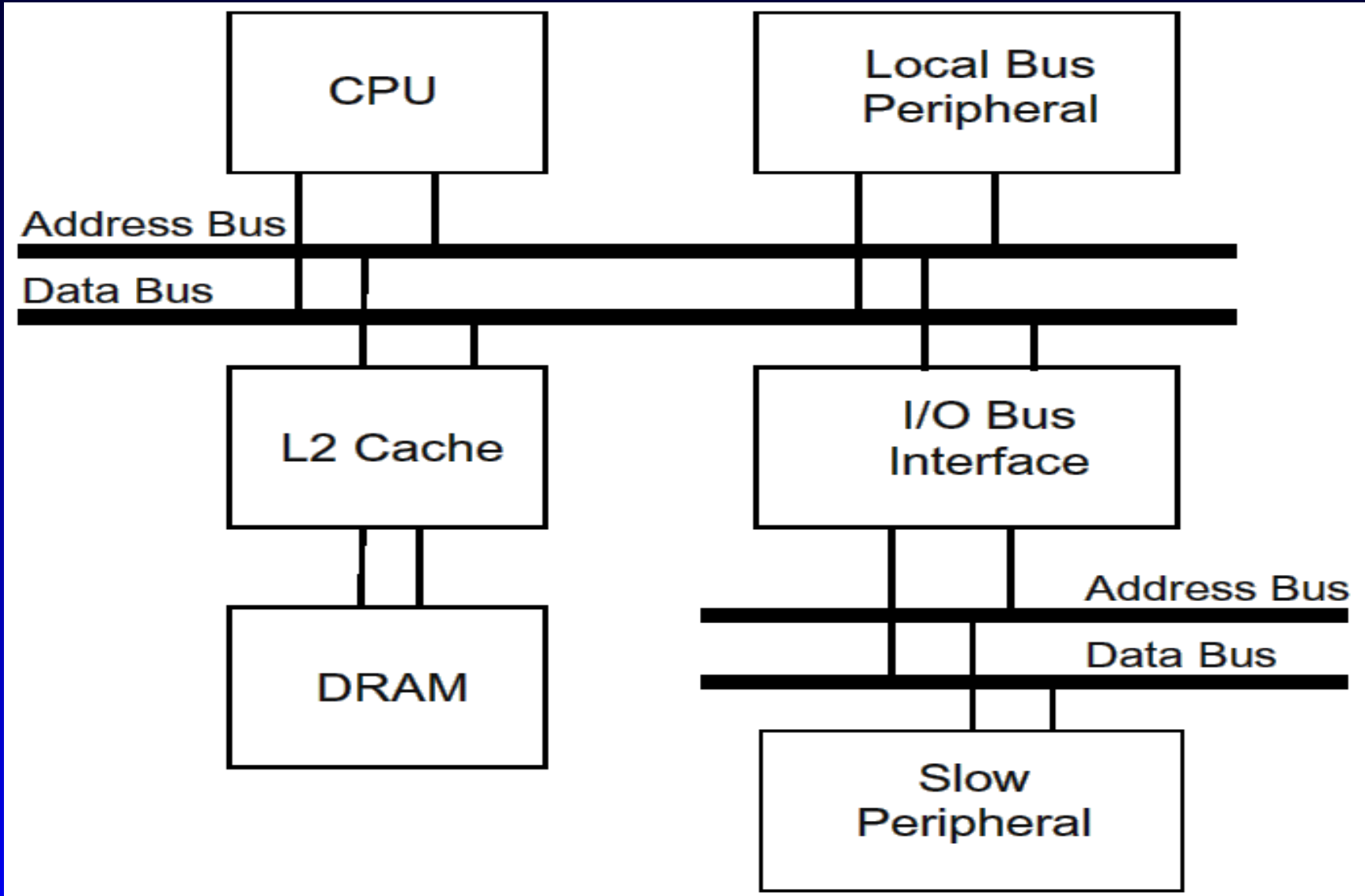
- Em sistemas multimaster é necessário cautela para manter a consistência do cache
  - *Flushing* do cache
  - Invalidação do cache
  - *Snooping* do cache
- Acesso não "cacheado"
  - Tipicamente para I/O mapeado em memória
- SOs podem necessitar fazer flush e invalidar o cache

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

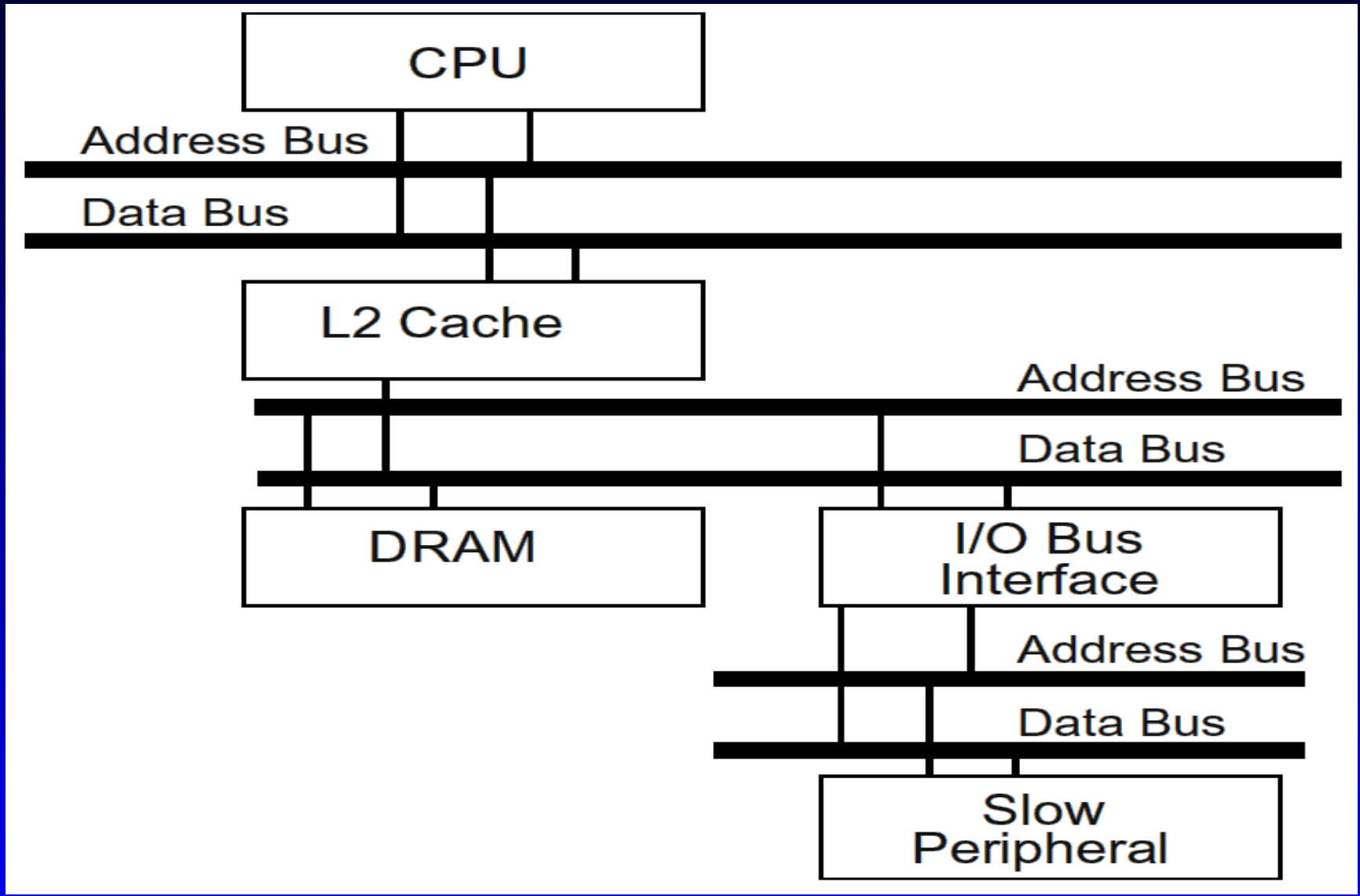
Caches Multiníveis

- Alguns processadores possuem cache no próprio *chip*
  - Tipicamente estes caches são pequenos
  - Trabalham na freqüência de *clock* do *core*
- Pode-se ter um cache maior externo
  - Normalmente a freqüência do *clock* do FSB é menor do que a do *core*

Cache Fracamente Acoplado



Cache Fortemente Acoplado



UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Exemplos

- Mapeado diretamente
  - IBM System/390 modelo 158
- Associativo por conjunto
  - 80486 (4 vias)
  - Pentium (dados: 2 vias, instruções: 4 vias)
  - i860 (4 vias)
  - 68040 (4 vias)
  - SuperSPARC (dados: 4 vias)
- Totalmente associativo
  - MIPS
  - MicroSPARC
  - SuperSPARC (instruções)

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Referências

[1] K. Hwang and F. A. Briggs. *Computer Architecture and Parallel Processing*. McGraw-Hill, New York, 1989.

[2] M. J. Murdocca and V. P. Heuring. *Introdução à Arquitetura de Computadores*. Campus, Rio de Janeiro, RJ, 2000.

[3] C. Schimmel. *UNIX Systems for Modern Architectures*. Addison-Wesley Professinal Computing Series. Addison-Wesley, Reading, MA, 1994.

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Exercício

- Processadores da faília P6 (Pentium II em diante) possuem dois níveis de cache
- O cache L1 é particionado
- O cache de dados é 2-way set associative
- O cache de instruções é 4-way set associtative
- Ambos tem 16 Kbytes (dependendo do modelo) com linhas de 32 bytes
- O cache L2 é unificado, 512 Mbytes (dependendo do modelo) com linhas de 32 bytes e 4-way set associative
- Todos eles utilizam o algoritmo LRU

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Exercício

- Faça um esboço deste sistema de cache explicitando:
  - Número de blocos
  - Número de blocos por conjunto
  - Particionamento do endereço em *tag*, conjunto e palavra
  - Número de bits no *tag*
  - Número de bits validade
  - Númro de bits para impelementar o LRU