Universidade Federal de Pelotas
Instituto de Física e Matemática
Departamento de Informática
Bacharelado em Ciência da Computação

Arquitetura e Organização de Computadores II

Aula 13

Memória Cache: organização, acesso, tratamento de faltas.

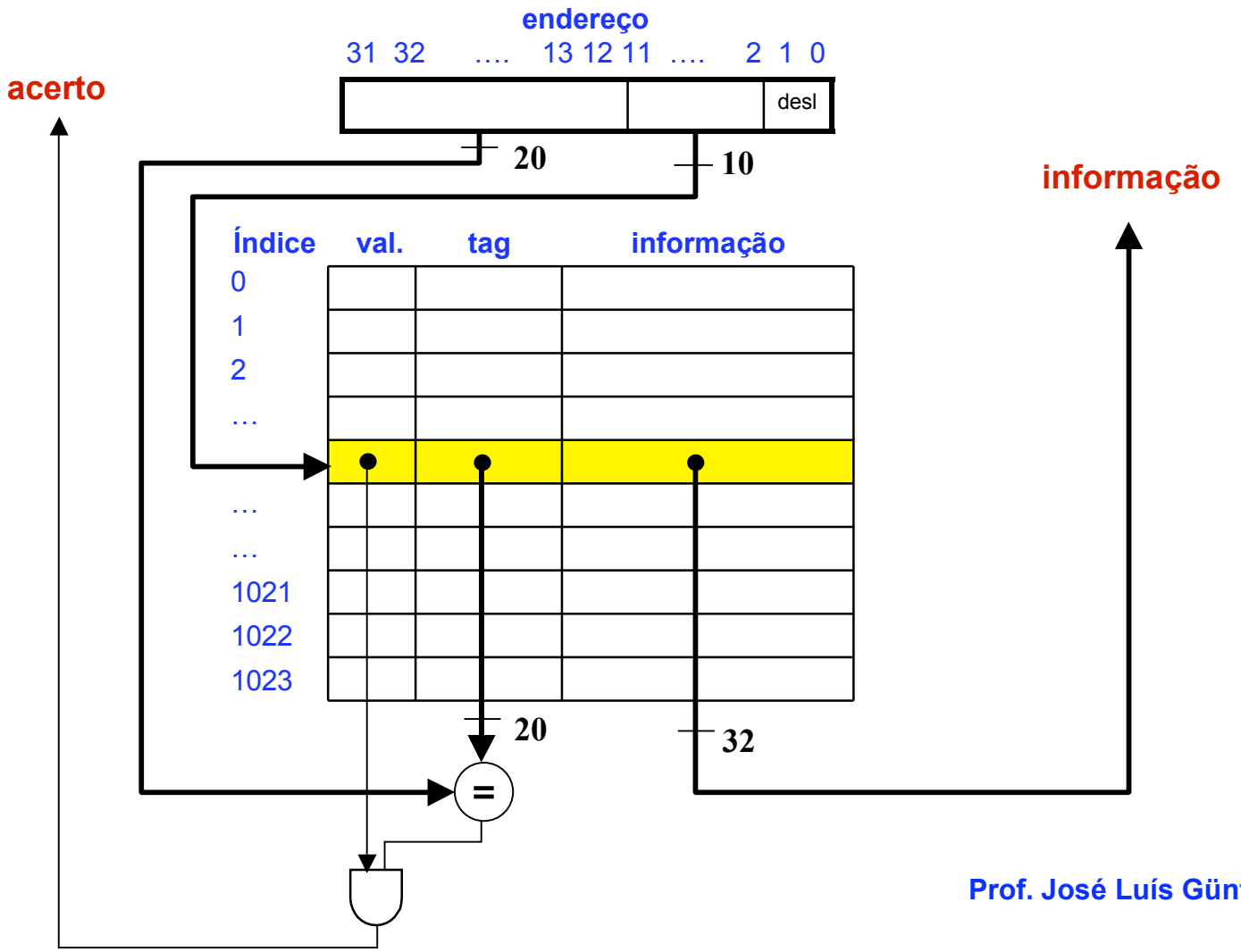
Prof. José Luís Güntzel

guntzel@ufpel.edu.br

www.ufpel.edu.br/~guntzel/AOC2/AOC2.html

3. Hierarquia de Memória: memória cache

Acesso a uma Cache Mapeada Diretamente



3. Hierarquia de Memória: memória cache

Número de Bits de uma Cache

- **O número de entradas em uma cache deve ser potência de dois**
- **O número total de bits necessários à implementação de uma cache é função do tamanho da cache e do tamanho da memória principal (i.e., do tamanho do endereço)**

3. Hierarquia de Memória: memória cache

Número de Bits de uma Cache

Supondo

- uma memória principal com endereços de 32 bits referenciando bytes
- Uma cache mapeada diretamente com 2^n palavras e blocos de uma palavra (4 bytes)

Então:

- Tamanho dos *tags*= 32-(n+2) bits, onde 2 bits são usados para o deslocamento e n para o índice
- Portanto, o número total de bits em uma cache mapeada diretamente é 2^n x (tamanho do bloco + tamanho do *tag* + tamanho do campo de validade)
- Tamanho da cache em questão será 2^n x (32+(32-n-2)+1) = 2^n x (63-n)

3. Hierarquia de Memória: memória cache

Número de Bits de uma Cache

Exemplo

Quantos bits são necessários para implementar uma cache mapeada diretamente com 64 kB entradas, e blocos de uma palavra, ligada a uma memória cujo endereço tem 32 bits?

Solução:

- 64 kB=16 kpalavras = 2^{14} palavras
- Cada bloco tem 32 bits de informação mais o campo do tag = 32 - 14 -2 bits, mais o bit de validade
- Portanto, o tamanho da cache é 2^{14} x (32+(32-14-2)+1) = 2^{14} x 49 = 784 x 2^{10} = 784 kbits

3. Hierarquia de Memória: memória cache

Tratamento de Faltas no Acesso à Cache

- **O Bloco de Controle deve tanto detectar quanto processar a falta** (buscando os dados na memória principal ou em outra cache imediatamente abaixo na hierarquia)
- **Se a cache informar um acerto:**
  - processamento segue (como se a informação tivesse sido obtida na memória principal)
  - O bloco de controle desenvolvido anteriormente (caps 5 e 6) pode ser usado
  - As memórias dos blocos operativos dos caps 5 e 6 podem ser substituídas por caches

3. Hierarquia de Memória: memória cache

Tratamento de Faltas no Acesso à Cache

- **Se a cache informar uma falta:**
  - O processador deve ser parado (congelando o conteúdo de todos os registradores)
  - Um controlador separado ajuda no tratamento das faltas geradas no acesso à cache, comandando a busca da informação (bloco) na memória principal ou na cache de próximo nível
  - Uma vez que o dado tenha sido obtido, a execução é reiniciada no ciclo que gerou a falta no acesso a cache
  - O processamento de uma falta na cache cria uma parada no processamento similar às paradas do pipeline: todos os registradores temporários e visíveis ao programador são congelados, enquanto a informação é transferida da memória

3. Hierarquia de Memória: memória cache

Tratamento de Faltas na Cache de Instruções

Considere os Blocos Operativos Multiciclo e Pipeline do MIPS:

- **Se o acesso a uma instrução resultar em falta, então o conteúdo do IR não é válido (uma vez que o PC é incrementado no primeiro ciclo de relógio, tanto na versão multiciclo quanto na versão pipeline)**
- **É necessário comandar uma leitura no nível inferior da hierarquia da memória, usando o PC-4 (no caso pipeline, recursos extras de HW serão necessários: subtrator ou deslocador)**
- **Instruir a memória para realizar a leitura e esperar a resposta (uma leitura demora muitos ciclos) e então escrever a palavra (instrução) na cache**

Tratamento de Faltas na Cache de Instruções

Resumindo os passos para tratamento de uma falta:

1. **Enviar à memória o valor original do PC (= PC-4)**
2. **Comandar uma leitura da unidade de memória e esperar o resultado**
3. **Escrever o resultado da leitura na entrada da cache, escrevendo também nessa entrada, no campo tag, os bits de mais alta ordem do endereço, e setando o bit de validade**
4. **Reiniciar a execução da instrução a partir do passo número 1, gerando uma nova busca da instrução na cache (mas desta vez, com a certeza de que ela será encontrada)**

3. Hierarquia de Memória: memória cache

Tratamento de Faltas na Cache de Dados

Os passos para o tratamento de faltas no acesso a dados (cache de dados) são essencialmente os mesmos que os usados no tratamento de uma falta de instrução

Também é necessário parar o processador até que o dado necessário esteja disponível na cache

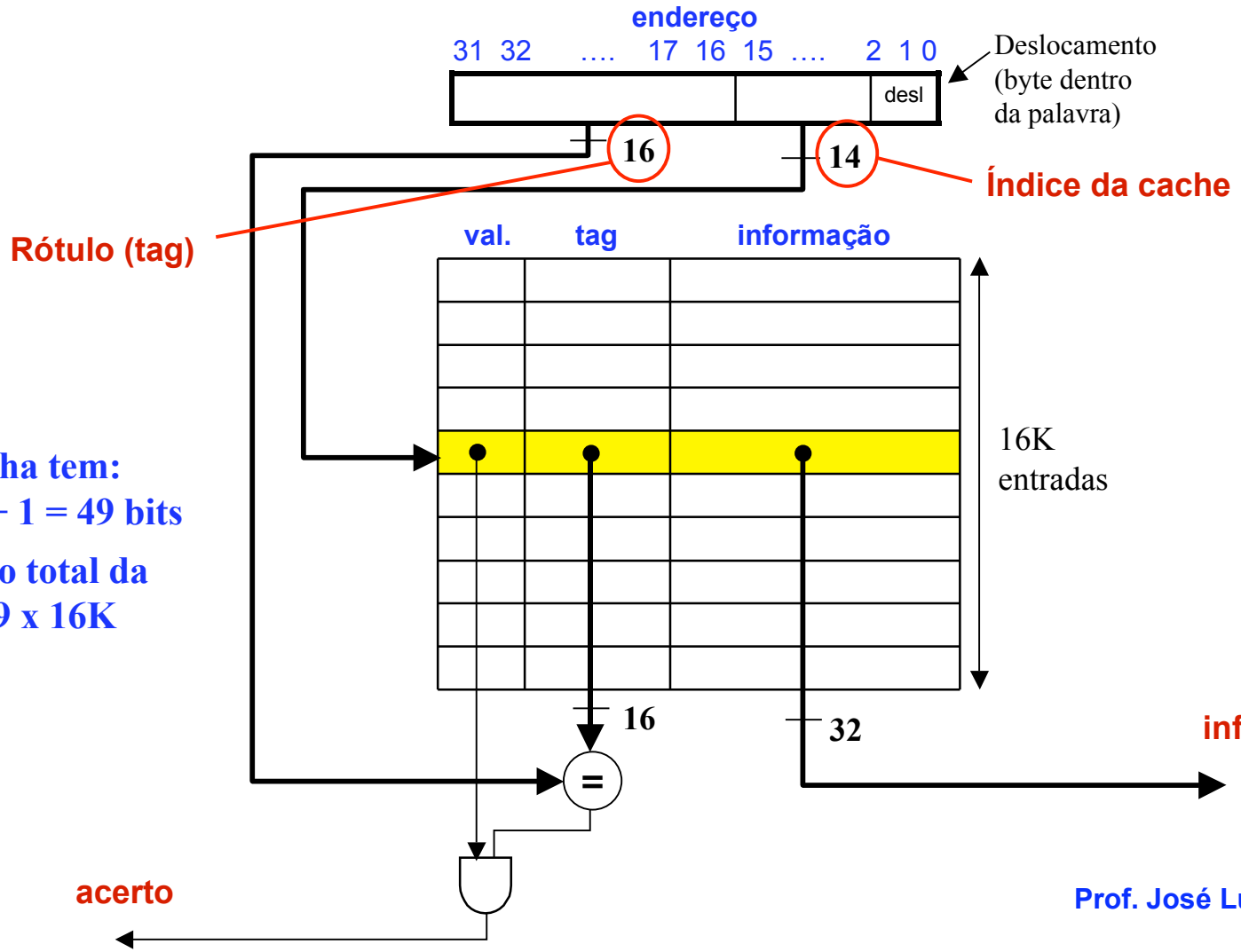
3. Hierarquia de Memória: memória cache

Exemplo: DECStation 3100 (MIPS R2000)

- **Usa o processador MIPS R2000, com pipeline idêntico ao visto em aula**
- **Em velocidade máxima, precisa de uma instrução e um dado a cada novo ciclo de relógio**
- **Cache de dados separada da cache de instruções**
- **Cada cache com 64KB (16K de palavras de 32 bits), com blocos de uma palavra**

3. Hierarquia de Memória: memória cache

Exemplo: DECStation 3100



Cada linha tem:
32 + 16 + 1 = 49 bits

Tamanho total da cache: 49 x 16K

3. Hierarquia de Memória: memória cache

Exemplo: DECStation 3100

Passos para atender a uma solicitação de leitura (em qualquer uma das caches, pois há sinais de controle separados):

1. **Enviar o endereço para a cache apropriada**
   - O endereço vem do PC (para leitura de instrução) ou da ULA (para leitura de dado)
2. **Se a cache sinaliza acerto, a palavra requisitada estará disponível nas linhas de dados. Se a cache sinalizar falta, o endereço causador da falta deve ser enviado à memória principal (ou cache do nível hierárquico inferior). Quando a memória retornar a informação solicitada, esta deve ser escrita na cache.**

3. Hierarquia de Memória: memória cache

Esquema *Write-Through*

Escrita na cache de dados:

- **Suponha que na execução de uma instrução store o dado seja escrito somente na cache de dados. Isto cusará uma inconsistência: após, a escrita na cache, a memória principal terá um valor diferente daquele que foi escrito na cache**
- ***Write-through*: uma solução é escrever o dado tanto na cache quanto na memória principal (esta solução é usada na DECStation 3100)**

3. Hierarquia de Memória: memória cache

Esquema *Write-Through*

No esquema *write-through* não há a necessidade de se considerar se uma escrita gera uma falta ou um acerto na cache. Basta escrever a palavra na cache…

Processo de escrita na DECStation 3100:

1. **Indexar a cache usando os bits 15-2 do endereço**
2. **Escrever os bits 31-16 do endereço no campo do tag, escrever a palavra de informação na parte reservada à informação na entrada da cache e setar o bit de validade**
3. **Escrever a palavra na memória principal usando o endereço inteiro**

3. Hierarquia de Memória: memória cache

Tratamento das Escritas

- **O esquema *write-through* não favorece o desempenho**
- **Qualquer escrita faz com que a memória principal seja escrita também**
- **Os programas podem ter muitas instruções "store". No gcc, 13% das instruções são "store", o que pode elevar o CPI de 1,2 para 2,5 (caso o acesso à memória principal demorar 10 ciclos de relógio…)**

3. Hierarquia de Memória: memória cache

Esquema *Write-Through* com *Buffer* de Escrita

Escrita na cache de dados com *Buffer* de escrita:

- ***Buffer* de escrita armazena o dado enquanto este aguarda para ser escrito na memória**
- **Após escrever o dado na cache e no *buffer* de escrita, o processador pode continuar a execução das instruções**
- **Se o *buffer* de escrita estiver cheio quando o processador tiver que executar uma instrução de escrita, o processador precisa parar, até que haja posição disponível no buffer**

3. Hierarquia de Memória: memória cache

Esquema *Write-Through* com *Buffer* de Escrita

Escrita na cache de dados com *Buffer* de escrita:

- **Se a velocidade da memória para completar as escritas for menor que a taxa à qual o processador está gerando as escritas, nenhum *buffer* (por maior que seja) conseguirá resolver o problema**
- **O tamanho do *buffer* de escrita depende da relação citada no item anterior**
- **Na DECStation 3100 o *buffer* de escrita tem 4 palavras**

Exemplos de Taxas de Faltas

Taxas de falta obtidas com o esquema write-through da DECStation 3100 com programas "típicos"

Programa	Taxa de faltas no acesso a instruções	Taxa de faltas no acesso a dados	Taxa de faltas combinada
gcc	6,1%	2,1 %	5,4 %
spice	1,2 %	1,3 %	1,2 %

3. Hierarquia de Memória: memória cache

Esquema *Write-Back*

Escrita na cache de dados com *Write-back*:

- **No esquema *write-back*, quando ocorre uma escrita, o novo valor é escrito apenas no bloco da cache. Tal bloco somente será escrito na memória principal quando ele tiver que ser substituído na cache**
- **O esquema pode aumentar bastante o desempenho, principalmente quando o processador puder gerar escritas tão rapidamente quanto estas puderem ser tratadas pela memória principal**