UNISC
UNIVERSIDADE DE SANTA CRUZ DO SUL

Arquitetura de Computadores II – Trabalho I

Prof. Eduardo Wächter

Simulador de Memória Cache

Objetivos

- Compreender o funcionamento de uma memória cache;
- Desenvolver habilidades de programação;
- Identificar técnicas e métodos discutidos durante a disciplina em casos reais;
- Apresentar de forma sucinta, através de um relatório, o trabalho desenvolvido.

Definição do Problema

O desempenho de um processador está diretamente relacionado com a Infra-estrutura de comunicação e a memória. Para resolver o problema da inexistência de uma memória de tamanho infinito e tempo de acesso baixo, surgiram as memórias cache.

Uma das maneiras de implementar memórias cache é através do **Mapeamento Direto**, conforme a Figura 1. Neste mapeamento, cada posição (ou bloco) de memória pode ser posicionado em apenas um determinado posição (ou bloco) da cache.

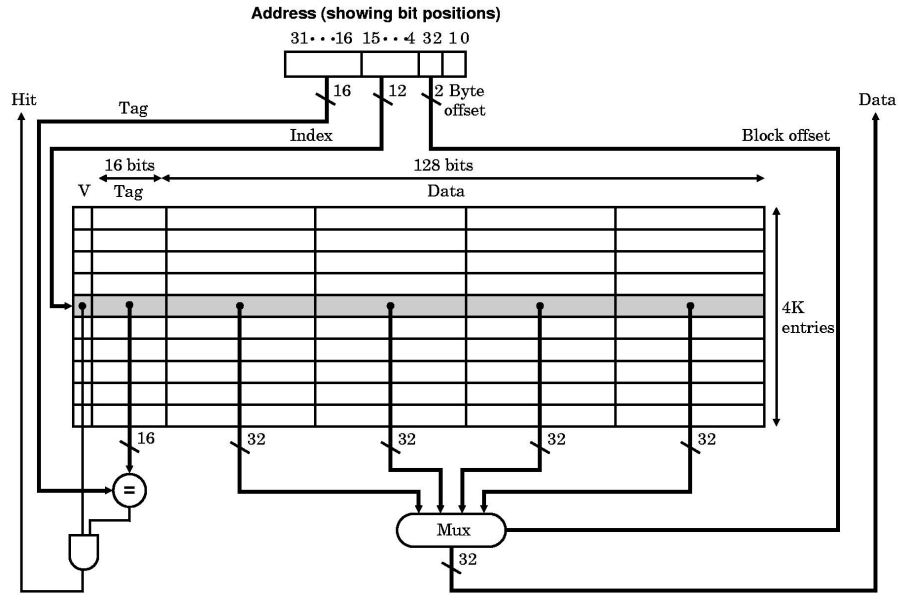

Figura 1 - Exemplo de memória cache com mapeamento direto.

Especificação do Trabalho

Vocês deverão implementar, em uma linguagem de programação qualquer, um **simulador de memória cache com mapeamento direto**, com os seguintes parâmetros:

- Endereço de 32 bits: Os acessos a cache serão feitos através de endereços de 32 bits
- Offset de 2 bits: Bloco de 4 palavras: Cada palavra tem 32 bits, totalizando 16 bytes.
- Índice de 4 bits: Totalizando 16 linhas
- Tag de 24 bits.

Neste caso, a cache de acordo com a especificação, a organização dos endereços ficaria da seguinte forma:

Tag (24 bits)	index (4 bits)	offset (2 bits)	byte (2 bits)
31 .. 8	7 6 5 4	3 2	1 0

O simulador deverá ter dois modos de execução: (i) passo a passo e (ii) Contínuo. No modo de execução passo a passo, o programa deverá exibir o estado da memória cache a cada novo acesso de endereço. No modo contínuo, ele deverá exibir o estado final da cache. Nos dois modos de operação, deverá apresentar o número de **hits** e **misses** que a cache apresentou.

Requisitos

O programa **deverá executar em linha de comando**. Seguindo a seguinte sintaxe:

```
./cachesim [-P] entrada.txt
```

Onde:

cachesim: é o nome dado ao executável do programa

-P: o parâmetro é **opcional** e denota que o programa irá executar no modo passo a passo

entrada.txt: nome do arquivo com os endereços de memória quais serão acessados. Os dados deste arquivo devem estar no formato hexadecimal, como segue:

01234567
09876543
09876545
11223344

Figura 2 - Exemplo de arquivo entrada.txt com os endereços acessados.

Este arquivo entrada.txt simula os acessos a memória feitos pelo processador. Cada linha contém um número hexadecimal com o endereço acessado pelo processador. Por exemplo, na Figura 2, no primeiro ciclo, o processador acessou o endereço 0x01234567, no segundo ciclo o endereço 0x09876543 e assim sucessivamente.

Exemplo de Execução

Numa execução passo a passo, o programa deverá mostrar, de acordo com as figuras a seguir.

Estado inicial:

Idx	V	Tag	data	data	data	data
0	0					
1	0					
2	0					
3	0					
4	0					
5	0					
6	0					
7	0					
8	0					
9	0					
A	0					
B	0					
C	0					
D	0					
E	0					
F	0					

HITS:0 MISSES:0

Leitura do endereço 0x*01234567*

Idx	V	Tag	data	data	data	data
0	0					
1	0					
2	0					
3	0					
4	0					
5	0					
6	1	012345	mem(0123456X)	mem(0123456X)	mem(0123456X)	mem(0123456X)
7	0					
8	0					
9	0					
A	0					
B	0					
C	0					
D	0					
E	0					
F	0					

HITS:0 MISSES:1

Leitura do endereço 0x*09876543*

Idx	V	Tag	data	data	data	data
0	0					
1	0					
2	0					
3	0					
4	1	098765	mem(0987654X)	mem(0987654X)	mem(0987654X)	mem(0987654X)
5	0					
6	1	012345	mem(0123456X)	mem(0123456X)	mem(0123456X)	mem(0123456X)
7	0					
8	0					
9	0					
A	0					
B	0					
C	0					
D	0					
E	0					
F	0					

HITS:0 MISSES:2

Leitura do endereço 0x*09876545*

Idx	V	Tag	data	data	data	data
0	0					
1	0					
2	0					
3	0					
4	1	098765	mem(0987654X)	mem(0987654X)	mem(0987654X)	mem(0987654X)
5	0					
6	1	012345	mem(0123456X)	mem(0123456X)	mem(0123456X)	mem(0123456X)
7	0					
8	0					
9	0					
A	0					
B	0					
C	0					
D	0					
E	0					
F	0					

HITS:1 MISSES:2

Leitura do endereço 0x*11223344*

Idx	V	Tag	data	data	data	data
0	0					
1	0					
2	0					
3	0					
4	1	112233	mem(1122334X)	mem(1122334X)	mem(1122334X)	mem(1122334X)
5	0					
6	1	012345	mem(0123456X)	mem(0123456X)	mem(0123456X)	mem(0123456X)
7	0					
8	0					
9	0					
A	0					
B	0					
C	0					
D	0					
E	0					
F	0					

HITS:1 MISSES:3

Apresentação e Entrega

Este trabalho ***deve*** ser realizado necessariamente em ***dupla (DUAS pessoas)***, sendo que a apresentação será feita em aula, de forma individual.

Este trabalho pode ser realizado em ***dupla (DUAS pessoas)***, sendo que a entrega deve ser na forma de um relatório pelo EAD Virtual contendo:

- Relatório contendo a descrição das atividades realizadas e como compilar e executar o código fonte
- Código Fonte

A data para entrega e apresentação do trabalho é dia ***13/5/2016***.

Avaliação

A avaliação será feita ***individualmente***, onde cada um dos alunos do grupo terá de explicar separadamente o funcionamento do trabalho (Código Fonte e Execução). Os critérios de avaliação serão os seguintes:

- **Relatório (2 pontos):** (Detalhamento e explicação do algoritmo implementado): Na conclusão deve-se avaliar as vantagens e desvantagens da implementação. Lembre-se que não existe solução para todos os problemas, toda implementação necessariamente terá desvantagens. Seu trabalho como projetista é ser crítico e avaliar.

- **Corretude de Funcionamento (3 pontos):** Na apresentação serão feitos testes com diversos tipos de entradas.
- **Estrutura do Programa (2 pontos):** Neste quesito será avaliado a qualidade da codificação apresentada.
- **Parametrização (2 pontos):** Será avaliado a dificuldade para a modificação dos parâmetros como offset, índice e tag. Somente receberão a totalidade dos pontos os programas que não necessitem a reescrita de código para execução com parâmetros diferentes.
- **Qualidade da documentação (1 ponto)**: A capacidade de acrescentar comentários úteis no código fonte.

DÚVIDAS?

Podem me chamar durante a aula ou através do e-mail: eduardow@unisc.br.