**RELATÓRIO TÉCNICO DO SIMULADOR DE MEMÓRIA CACHE**

Gustavo Henrique Marchiori RA: 115810

Igor Henrique Salviano da Silva RA: 115171

Karoline Harummy Romero Moriya RA: 111849

INTRODUÇÃO

Neste trabalho, simulamos o funcionamento de uma memória cache, que utiliza mapeamento associativo como função de mapeamento, First In First Out (FIFO), Least Recently Used (LRU) e Least Frequently Used (LFU) como algoritmos de substituição e write-back como política de escrita. Abaixo explicamos sobre o funcionamento da memória cache, as funções de mapeamento, os algoritmos de substituição e a política de escrita.

***Objetivos/Justificativa***

O intuito do trabalho é o desenvolvimento de uma aplicação didática, para que o comportamento da comunicação entre memória Cache e memória RAM possa ser simulado. As implementações do mapeamento associativo aleatório, das três políticas de substituição ( FIFO, LFU e LRU) e da política de escrita write-back também possuem foco didático: quanto maior a quantidade de especificações, melhor a possibilidade de aprendizado.

***Funcionamento da memória cache:***

Para cada dado a ser acessado, há uma probabilidade dele estar na memória Cache. Se isso ocorrer, dizemos que houve um Hit e o sistema ganha muito tempo com isso. Caso contrário, ocorre um Miss e o desempenho é bastante prejudicado. A comunicação da memória RAM com a memória Cache exige políticas de mapeamento, políticas de substituição e políticas de escrita.

***Funções de mapeamento:***

Como existe mais blocos na memória principal, do que linhas na cache, é necessário um algoritmo para mapear os blocos da memória principal às linhas da cache e determinar qual bloco ocupará qual linha. Existem três técnicas:

* **Mapeamento direto**: Mapeia cada bloco da memória principal, a somente uma linha de cache possível. Por exemplo, considerando uma cachê de 2 linhas e uma memória principal de N blocos, os blocos de número par são destinados a linha 0, ou seja, eles só podem ocupar a linha 0, e os blocos de número ímpar são destinados a linha 1.
* **Mapeamento associativo**: Nesse caso, é permitido que cada bloco da memória principal seja destinado a qualquer linha da cache. Utiliza os algoritmos de substituição.
* **Mapeamento associativo em conjunto**: A cache é dividida em uma série de conjuntos, cada um contendo várias linhas. Os blocos são, então, mapeados para conjuntos específicos, podendo se posicionar em qualquer lugar linha dentro do conjunto em questão.

***Algoritmos de substituição:***

Quando um bloco da memória principal precisa ir para a cache, mas ocorre da mesma estar cheia, é utilizado os algoritmos de substituição para orientar, quais informações deixarão a cache para novas entrarem. É utilizado pelo mapeamento associativo e pelo mapeamento associativo em conjunto. Há três algoritmos mais comuns:

**LRU** (usado menos recentemente): substitui o bloco, que permaneceu na cache por mais tempo, sem qualquer referência a ele.

**FIFO** (primeiro a entrar, primeiro a sair): substitui o bloco, que esteve na cache durante maior tempo.

**LFU** (usado menos frequentemente): substitui o bloco, que teve menos referência.

***Política de escrita write-back:***

Quando um bloco, que está residente na cache estiver para ser substituído, pode ocorrer dois casos:

1) Se o bloco antigo na cache não tiver sido alterado, então ele pode ser substituído por um novo bloco, sem primeiro atualizar o bloco antigo na memória principal.

2) Se pelo menos uma operação de escrita tiver sido realizada em uma palavra nessa linha da cache, então a memória principal precisa ser atualizada, escrevendo a linha de cache no bloco de memória, antes de trazer o novo bloco.

Há duas principais técnicas utilizadas:

**Write-through**: Todas as modificações, ou seja, operações de escrita, são feitas tanto na cache quanto na memória principal, garantindo que a memória principal seja sempre válida.

**Write-back**: As modificações são feitas apenas na cache. Quando ocorre uma atualização, um bit de modificação associado à linha, é marcado. E quando um bloco for substituído, ele é escrito de volta na memória principal se, e somente se, o bit de modificação estiver marcado.

***Decisões de projeto para a implementação:***

Estruturas de dados utilizadas: foram criadas seis classes, a Cache para representar a memória Cache, a RAM para representar a memória RAM, a Row para representar a linha (ou bloco) da memória RAM ou Cache e a Cell para representar uma célula específica da RAM ou Cache.

Tamanho da memória cache: o tamanho de oito linhas da memória cache foi escolhido pensando nas duas mil e quarenta e oito linhas da memória RAM (o número dois mil e quarenta e oito dividido oito vezes é igual a oito).

Escolha dos algoritmos de substituição: os três algoritmos foram implementados, para que a interação com o software se faça mais completa e didática.

Dentre outras questões relativas à implementação: as decisões tomadas durante o desenvolvimento do projeto foram voltadas a experiência do usuário quanto ao aprendizado.

CONCLUSÃO:

Percebemos que ao desenvolver a simulação do funcionamento da memória cache foi obtido um bom resultado, pois acreditamos que o programa tenha funcionado corretamente.

Para nós, este trabalho foi de grande proveito, pois utilizamos e fixamos melhor os conteúdos do começo do semestre, para desenvolvê-los.

REFERÊNCIAS:

Stallings, William; Arquitetura e organização de computadores; 8ª edição, São Paulo; Pearson Pratice Hall, 2010.