

Universidade Federal do Pará

Faculdade de Engenharia da Computação e Telecomunicações

EC1008 - Arquitetura e Organização de Computadores

Trabalho de disciplina – Implementação de um simulador de memória cache

Trabalho de disciplina – implementar um simulador de memória cache

➤ Equipe alfa

Nome:	Matrícula:	E-mail:
Mércia Costa da Silva	201907040020	cmercias749@gmail.com
Devid Emanuel da Silva Barboza	201906840018	devid.barboza31@hotmail.com
Marcos Antônio Silva de Barros	201907040013	m.antonio22069@gmail.com
Caylon Solon Monte da Costa	201906840011	caylonsolon@hotmail.com
Flavio Oliveira de Sena	201907040023	sennadf14@gmail.com

O documento seguinte contém o relatório técnico com os devidos tópicos:

1. Resumo
2. Introdução
3. Descrição do simulador
4. Descrição do sistema avaliado
5. Resultado da avaliação
6. Conclusão
7. Bibliografia

Relatório técnico

Resumo:

O presente relatório técnico contém a descrição do simulador de memória cache implementado em Python. Essa ferramenta tem o intuito de simular seu funcionamento e desempenho.

Durante a simulação, foram utilizados os seguintes itens:

- Funções de mapeamento: determina como a cache é preenchida
 1. Mapeamento direto;
 2. Mapeamento associativo;
 3. Mapeamento associativo por conjunto.
- Políticas de substituição: determina como os dados serão substituídos
 1. FIFO;
 2. RANDOM.
- Representação de endereço feita por símbolos de 32 bits.
- É importante notar que os endereços de 32 bits são fornecidos em hexadecimal e convertidos para decimal

Introdução:

No decorrer dos sistemas computacionais, a diferença de velocidade entre a unidade central de processamento e a memória principal se tornou um obstáculo ao aumento do desempenho dos computadores. Uma das formas encontradas para acabar com essa problemática foi o emprego de uma pequena porção de memória mais veloz e de custo mais elevado, denominada memória cache.

A memória cache é um tipo de memória que trabalha em conjunto com o processador. Todos os processadores trazem uma quantidade de cache embutida no encapsulamento. Esse tipo de memória possui alta velocidade e tem por função armazenar temporariamente os dados e instruções que possuem maior probabilidade de serem usados pelo processador, reduzindo o número de acessos à memória principal. Sua utilização visa a melhora no desempenho do sistema. Ela possibilita que o processador trabalhe com toda a capacidade e tenha o mínimo de tempo ocioso possível. Cada fabricante utiliza a memória cache de uma forma diferente.

O projeto relatado nesse artigo tem por objetivo geral descrever o funcionamento de um simulador de memória cache, compreende-se por simulador uma forma de dinamizar o entendimento dos conceitos que a memória cache envolve. Dessa maneira, apresenta-se todo o processo de criação desse simulador e, em seguida, a sua execução.

Descrição do simulador:

Para o desenvolvimento desse simulador foi utilizado a linguagem Python. Desse modo, foram criadas algumas funções para facilitar o manuseamento desse simulador, segue abaixo todas as funções aplicadas e uma breve descrição das mesmas:

- `Verifica_posicao`: verifica se existe na cache uma posição de memória que ainda não foi utilizada, se existir, essa posição é retornada;
- `Imprime_contador_fifo`: função de debug que exibe o estado do contador FIFO;
- `Inicia_contador_fifo`: seta os valores do contador FIFO para que a primeira substituição ocorra no primeiro elemento que faz parte do conjunto;
- `Get_ncpm`: retorna o valor inteiro do resto da divisão dos atributos;
- `ImprimeCd`: exibe informações referentes ao mapeamento direto;
- `ImprimeCa`: exibe informações referentes ao mapeamento associativo;
- `Inicio_cache`: cria uma cache zerada utilizando como dicionário usando como valor padrão -1;
- `Verifica_posicao`: verifica se a posição está no modo associativo ou associativo por conjunto;
- `Get_lista_posicoes`: retorna uma lista com todas as posições da memória que fazem parte de um conjunto;
- `Substituição_RANDOM`: as posições que serão substituídas serão definidas de maneira aleatória;
- `Substituição_FIFO`: essa substituição será feita como uma fila, onde o primeiro elemento que entra é o primeiro que sai;
- `Executar_mapeamento_direto`: executa a operação de mapeamento direto;
- `Executar_mapeamento_associativo`: executa a operação de mapeamento associativo;
- `Executar_mapeamento_associativo_conjunto`: executa a operação de mapeamento associativo, ou seja, não existe uma posição específica para o mapeamento de uma posição de memória;
- `verifica_posicao_CA_AC`: verifica se a posição está no modo associativo ou associativo por conjunto;
- `alunos`: exibe o nome dos integrantes da equipe.

Descrição do sistema de cache avaliado:

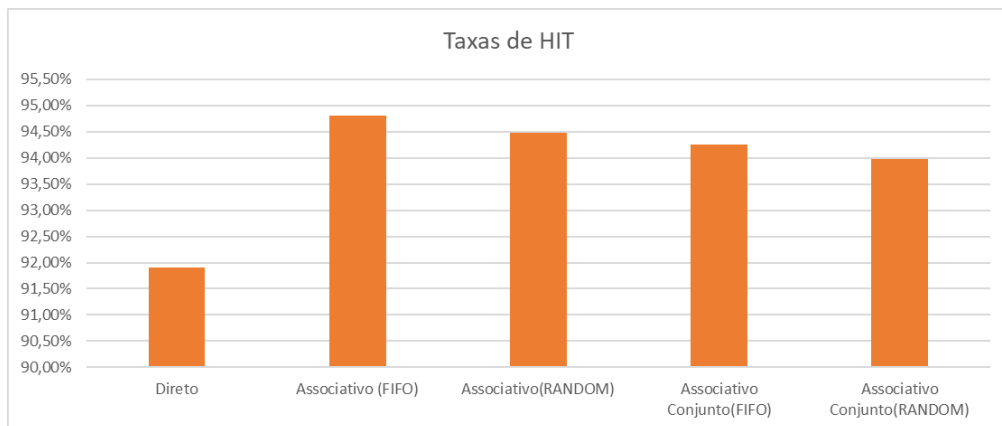
- Funções de mapeamento: determina como a cache é preenchida
 1. Mapeamento direto: cada bloco da memória principal é mapeado para apenas uma linha de cache.
 2. Mapeamento associativo: um bloco pode ser alocado em qualquer linha de cache.
 3. Mapeamento associativo por conjunto: a cache é dividida em conjuntos, com certo número de linha cada e um bloco pode ser associado a qualquer linha de um conjunto já pré-determinado.
- Políticas de substituição: determina como os dados serão substituídos
 1. FIFO: princípio da fila, o bloco que chegou primeiro deve sair para dar lugar a um novo bloco;
 2. RANDOM: o sistema escolhe aleatoriamente o novo método que deve ser removido, ele sai da cache dando lugar ao bloco que foi acessado.
- Erro e acerto de cache:
 1. Hit ou taxa de acerto de cache: quando o processador busca uma informação na memória e essa informação se encontra lá;
 2. Miss ou taxa de erro de cache: quando o processador busca uma informação e não a encontra na memória.
- Configurações diferentes de memória cache:
 1. 1024 espaços de memória e 4 conjuntos
 2. 2048 espaços de memória e 8 conjuntos

Resultados da avaliação:

- Após a execução e a comparação, as seguintes descrições foram extraídas:
- 1. O mapeamento direto possui o melhor tempo de execução, porém, a sua taxa de acerto foi a menor em comparação com o restante;
- 2. Embora o mapeamento associativo tenha a maior taxa de acerto, obteve o tempo mais prolongado durante a execução;
- 3. O mapeamento associativo por conjunto alcançou uma taxa média de acerto e um tempo médio de execução, ambos os resultados foram tidos como médios ao serem comparados com os demais tipos de mapeamento. Sendo assim, o associativo por conjunto foi tido como o que possui melhor custo-benefício.
- 4. Houve significativa diferença nas comparações das diferentes configurações de memória cache;
- 5. Por fim, encontra-se total de acessos a memória. Que foi de 921004.

- Para obter uma comparação ainda mais nítida, a seguir estão os gráficos com algumas das relações supracitadas, há comparações das diferentes configurações:

1. Gráfico com comparativo das taxas de acerto de memória cache:

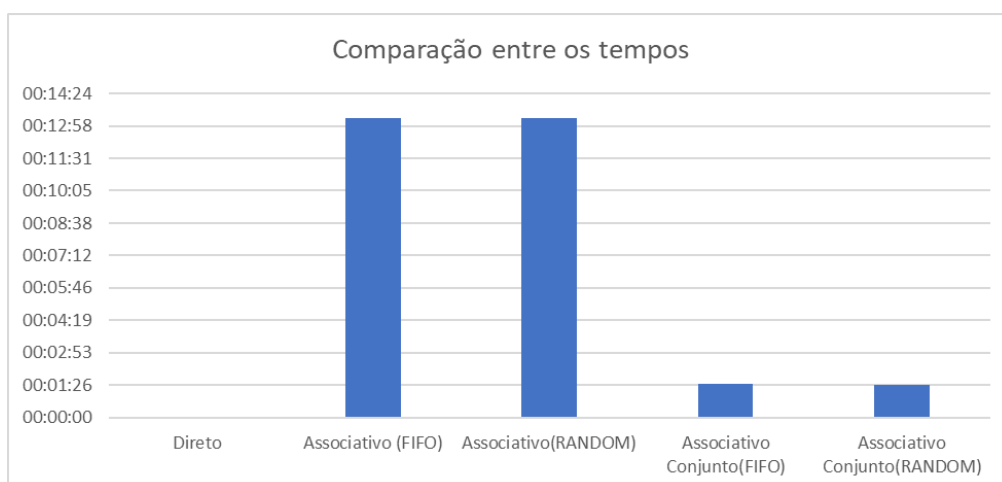


Características:

-2048 espaços de memória

-8 conjuntos

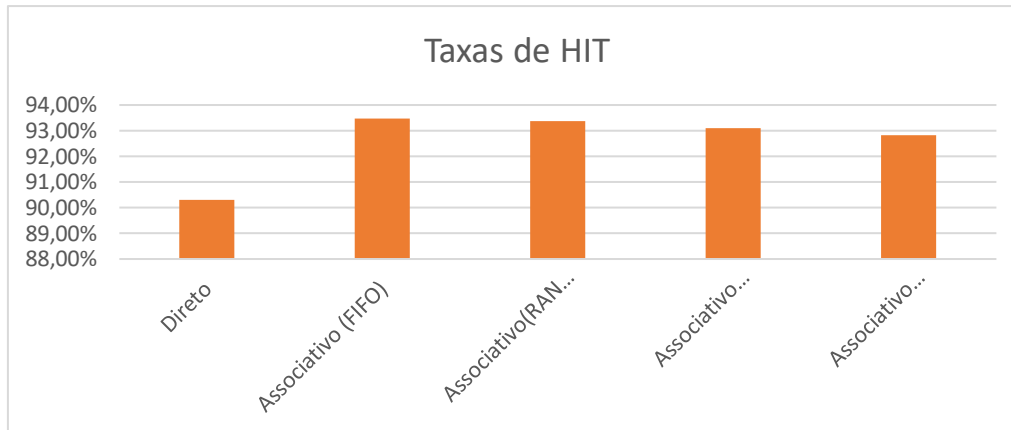
2. Gráfico com comparativo do tempo de execução:



3. Gráficos com descrição dos tipos de mapeamento e suas especificações:

Tipo de mapeamento	Quantidade de espaços na memória cache	Política de Substituição	HIT	MISS	TAXA DE HIT	Tempo de execução
Direto	2048	-	846378	74626	91,90%	00:00:01
Associativo (FIFO)	2048	FIFO	870187	47786	94,81%	00:13:18
Associativo (RANDOM)	2048	FIFO	870187	50817	94,48%	00:13:17
Associativo Conjunto (FIFO)	2048	RANDOM	868156	52848	94,26%	00:01:29
Associativo Conjunto (RANDOM)	2048	RANDOM	865590	55414	93,98%	00:01:28

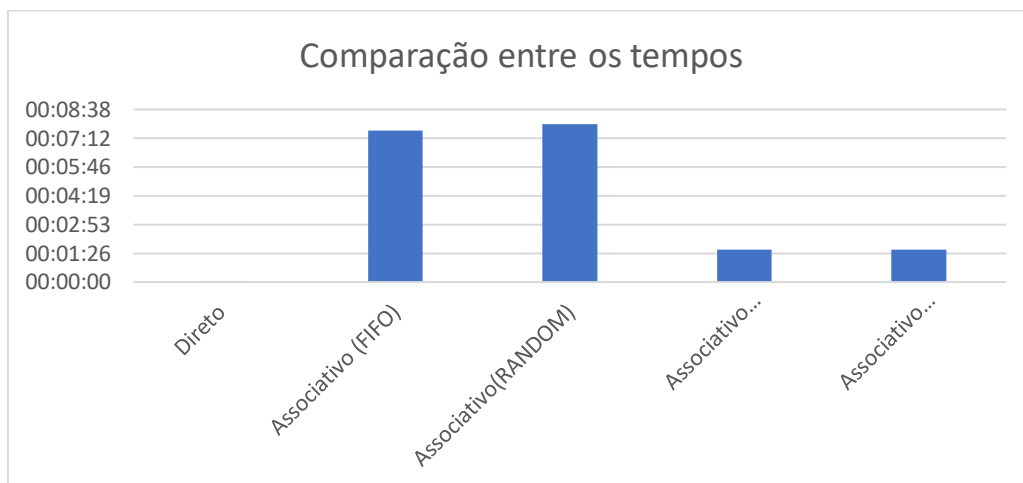
4. Gráficos com comparativo das taxas de acerto de memória cache:



Características:

- 1024 espaços de memória
- 4 conjuntos

5. Gráfico com comparativo do tempo de execução:



6. Gráfico com descrição dos tipos de mapeamento e suas especificações:

Tipo de mapeamento	Quantidade de espaços na memoria cache	Politica de Substituição	HIT	MISS	TAXA DE HIT	Tempo de execução
Direto	1024	-	831647	89357	90,30%	00:00:01
Associativo (FIFO)	1024	FIFO	860955	60049	93,48%	<u>00:07:35</u>
Associativo(RANDOM)	1024	FIFO	859910	61094	93,37%	00:07:54
Associativo Conjunto(FIFO)	1024	RANDOM	857447	63557	93,10%	00:01:38
Associativo Conjunto(RANDOM)	1024	RANDOM	854907	66097	92,82%	<u>00:01:38</u>

Conclusão:

Assim, a memória cache é uma memória mais veloz e, por conta disso, causa um impacto positivo, ou seja, reduz o tempo médio de acesso aos dados armazenados na memória. Tendo em vista isto, houve a simulação, que contou com testes segundo traços fornecidos pelo professor e duas configurações de memória cache diferentes: 1024 e 2048. Além desse, os demais parâmetros pedidos foram implementados devidamente e, dessa maneira, o trabalho foi concluído e houveram resultados como: elevado acerto de cache, cerca de 90% e obteve-se o mapeamento associativo por conjunto como o mapeamento com melhor custo-benefício.

Bibliografia:

<https://docs.python.org/3/>

https://github.com/anselmobattisti/simulador_cache

STALLINGS, Willian. Arquitetura e Organização de computadores. 10ªed. São Paulo: Pearson Education do Brasil, 2017.