

Simulador de Memória Cache

Eliane Isadora Faveron Maciel

¹Centro de Ciências Exatas e Tecnologias (CCET) – Universidade de Caxias do Sul (UCS)
Rua Francisco Getúlio Vargas, 1130 – 95.070-560 – Caxias do Sul – RS – Brasil

{eifmaciel@ucs.br}

Abstract. *This article aims to describe a system developed to simulate Cache Memory operations, developed in Python, this system has input parameters that configure the structure of the Cache memory, and generates the results being based file containing the addresses and operations They will be held in the memory cache.*

Resumo. *Este artigo tem como objetivo descrever um sistema desenvolvido para simular operações de Memória Cache, desenvolvido em Python. Este sistema tem parâmetros de entrada que configuram a estrutura da Memória Cache e gera os resultados, tendo como base um arquivo que contém os endereços e operações que serão realizados.*

1. Introdução

Este artigo descreve análises de simulações de uma Memória Cache. O objetivo é avaliar o desempenho da Memória Cache de Mapeamento Associativo por Conjuntos, levando em consideração a taxa de acerto. As análises realizadas avaliam os seguintes Impactos: Tamanho da Cache, Tamanho do Bloco, Associatividade, Política de Substituição, Largura de Banda. Para cada um destes itens, foram feitos testes em um Sistema de Simulação, com os resultados do sistema foi gerado um gráfico para avaliar o resultado. Descreve-se também dados do funcionamento deste sistema.

2. O Sistema de Simulação

O Sistema de Simulação de Memória Cache, foi desenvolvido com a Linguagem de programação Python. Este sistema simula a organização de uma Memória Cache com Mapeamento Associativo por Conjunto. No sistema pode ser configurado parâmetros que serão usados para a definir diversos tamanhos e organização da Memória Cache. O Sistema também precisa de um arquivo (oficial.cache), onde contém os endereços de Memória para que faça as operações.

Ao final do processamento, o sistema mostrará os dados fornecidos na entrada e os cálculos realizados. Por sua vez, será gerado um arquivo "results.txt" com as informações anteriormente descritas.

2.1. Estrutura de Dados

Especificações da Estrutura dos dados do sistema:

- Classes(Orientada a Objetos) são elas:
 - MemoriaCache - Atributos: *Array* - Quantidade de conjuntos;
 - Linha - Atributos: Rótulo, Endereço, LRU - Guarda a data do Acesso, e LFU - Guarda a quantidade de acessos;
 - Conjunto - Atributos: *Array* - Contém as linhas dos conjuntos.

2.2. O Funcionamento

Para o funcionamento da simulação da Memória Cache, o sistema precisa ser configurado com o tamanho e organização, por isso o usuário deve informar os seguintes parâmetros:

- Política de Escrita;
- Tamanho de Linha;
- Número de Linha;
- Associatividade por conjunto;
- Tempo de Acesso;
- Política de Substituição;
- Tempo de leitura/escrita na Memória principal

Uma vez fornecidos os dados, o sistema calcula o tamanho da Cache e se prepara para o próximo passo, que é a leitura de um arquivo "oficial.cache". Neste arquivo contêm os endereços e operações que serão realizados na Memória cache. Para cada linha do arquivo, o sistema separa o endereço de memória, que possui 32bits, em rótulo, conjunto, e a operação que será realizada na Memória Cache. Para cada operação terá um processo no programa.

Operação de leitura na Memória Cache, o sistema faz uma busca pelo conjunto, se encontrar faz a busca pelo rótulo, se encontra-lo acresce um Hit. Caso não encontre, o conjunto e/ou o rótulo, soma um em Miss, faz a busca na Memória Principal e grava o rótulo na Memória Cache.

Na operação de escrita da Memória Cache, é realizada uma busca na Memória Cache pelo conjunto, caso não encontre é gravado o conjunto e o rótulo, se encontra o conjunto e não encontra o rótulo, grava o rótulo no conjunto.

Para colocar os rótulos no conjunto da Memória Cache, quando o conjunto está cheio o programa leva em consideração a Política de Substituição, pois se o conjunto está cheio deve ser retirado um rótulo e acrescentado o rótulo necessário. Quando o conjunto não está cheio o rótulo pode ser colocado em qualquer posição. As Políticas de Substituição implementadas foram:

- LFU - *Last Frequently Used* - Linha menos usada, é guardado em um contador a quantidade de acessos daquela linha. Se for usado este método, será substituído a linha que foi menos usada.
- LRU - *Last Recently Used* - Linha menos recentemente usada, é guardado em uma variável a data e hora de acesso da linha. Este método substitui a que foi usada a mais tempo.
- Aleatório - O sistema faz um cálculo *Random*, para decidir qual linha irá substituir.

3. Impacto do Tamanho da Cache

Foram feitos testes com Memórias Cache de vários tamanhos, de 2KBytes até 32KBytes, neste caso avalia-se a taxa de acerto. Observando o Gráfico podemos perceber que quanto maior o tamanho da cache maior a taxa de acertos na cache. Isso ocorre devido a localidade espacial, assim o conteúdo da cache não precisa ser substituído muitas vezes aumentando os acertos.

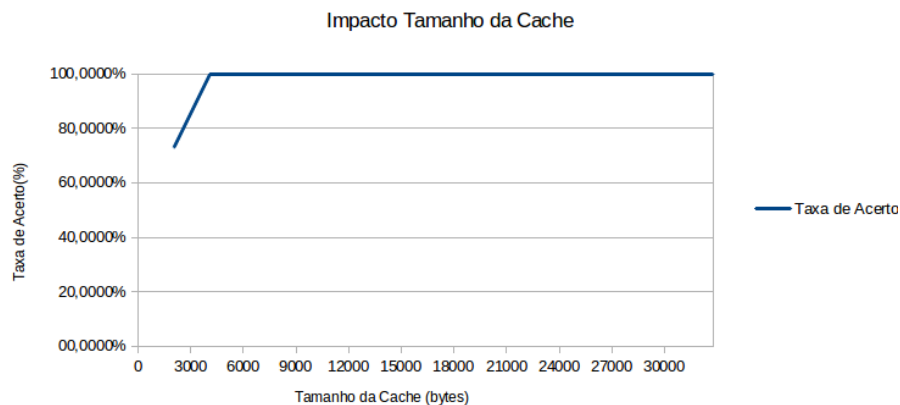


Figura 1. Impacto do tamanho da Cache

4. Impacto do Tamanho do Bloco

Nesta análise os testes foram feitos mantendo o tamanho da Cache de 8Kbytes e alterando o Tamanho do Bloco. Para que a Cache tenha o mesmo tamanho, a medida em que aumentamos o Tamanho do Bloco diminuimos o número de linhas. Assim os testes começaram com Tamanho de Bloco de 4Bytes e Número de Linhas de 2KBytes até Tamanho do Bloco de 2kBytes até Número de Linhas de 4Bytes.

No Gráfico gerado a partir dos resultados podemos analisar a taxa de acerto na Cache, quanto maior o Tamanho do Bloco, menor é a taxa de acertos. Na localidade espacial leva-se em consideração que pode haver referências futuras aos vizinhos, assim quanto menor o número de linhas mais trocas tem que ser feitas na cache, diminuindo a taxa de acertos. Para maior taxa de acertos podemos usar o Tamanho do Bloco de 4 bytes.

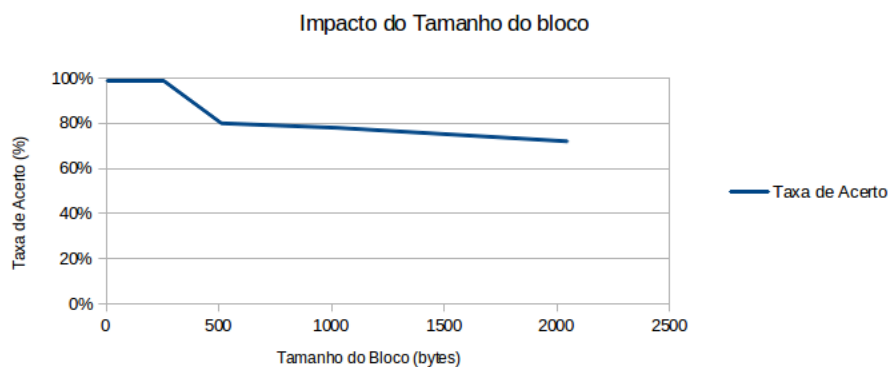


Figura 2. Impacto do tamanho do bloco

4.1. Impacto da Associatividade

Nesta análise de Impacto da Associatividade, os testes foram feitos mantendo o tamanho da Cache de 8Kbytes, Tamanho do Bloco de 128 Bytes e Número de linhas de 64 Bytes, foi alterado apenas Associatividade por conjunto começando em 1 e aumentando até 32. No Gráfico pode ser observado os resultados dos testes, tendo em vista que foi analisado a taxa de acerto na Cache, podemos concluir que aumentar a Associatividade por conjunto

não terá impacto na taxa de acerto. Isso se deve ao fato de que o Tamanho da Cache, Tamanho do Bloco e Número de Linhas são sempre os mesmos.

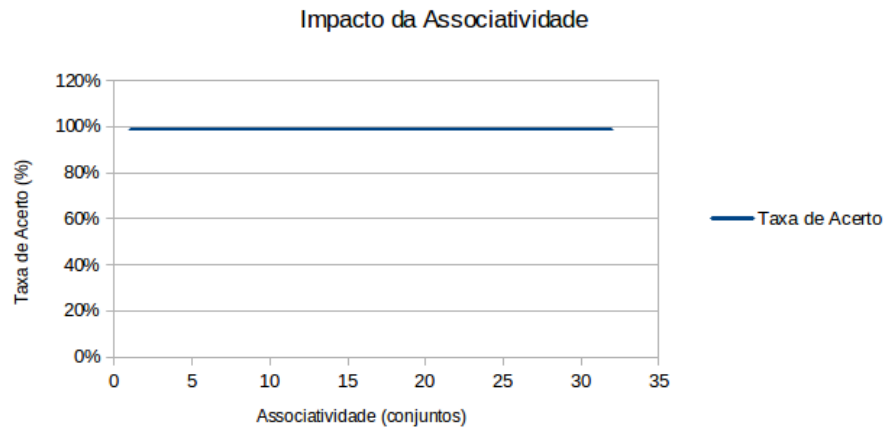


Figura 3. Gráfico - Impacto da associatividade

5. Impacto da Política de Substituição

Nesta análise de Impacto da Política de Escrita, os testes foram feitos aumentando o Tamanho da Cache para os três tipos de Política de Escrita. Conforme podemos analisar no Gráfico que mostra a taxa de acerto na Cache em função do Tamanho da Cache, para os diferentes tipos de Política de escrita. Conclui-se que a taxa de acerto aumenta à medida que o tamanho da cache também aumenta.

As Políticas de substituição tem diferentes taxas de acerto até que o tamanho da Cache tenha 8KBytes, neste tamanho as taxas se igualam. Podemos perceber também que o método de substituição mais eficiente é o Aleatório, neste método é substituído uma linha aleatória do conjunto.

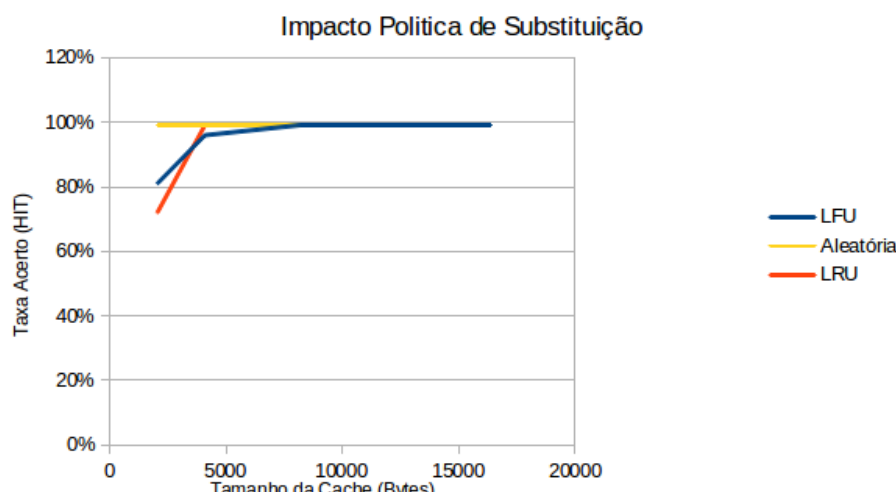


Figura 4. Gráfico - Impacto da Política de Escrita

6. Largura de Banda da Memória

Nesta análise de Largura de Banda da Memória, os testes foram feitos alterando os tamanhos da Cache e as Políticas de Escrita. Nesta análise não é necessário gerar gráfico, podemos concluir que o método de Política de Escrita Write Through tem maior tráfego na memória, pois cada escrita na cache gera uma escrita na Memória principal. Este método tem a vantagem das Memórias ter os dados sempre atualizado.

A Política de Escrita Write-Back, faz as escritas apenas na Cache, e só quando o bloco é substituído atualiza a Memória Principal, por este motivo apresenta menor tráfego de Memória.

7. Conclusão

Conclui-se que, com base nos experimentos realizados no sistema de simulação e análises dos resultados, a melhor opção para o projeto de cache, dentre os analisados, teria as seguintes especificações : Tamanho de bloco de 128 Bytes e Tamanho de Cache de 4 KBytes à 32KBytes, Política de Escrita Write-back, Política de Substituição Aleatório.

O simulador foi desenvolvido para o Mapeamento por Conjuntos de Memória Cache, não é possível fazer simulações de outros métodos de mapeamento.

A maior dificuldade encontrada no desenvolvimento do Simulador, foi a etapa de estruturas de dados, pois a compreensão do método de mapeamento não estava bem claro. Outra dificuldade enfrentada foi na parte de programação, como por exemplo, transformar um número Hexadecimal em binário de 32 bits.

Com este trabalho foi possível aprender mais sobre a Memória Cache, e sua organização, não só com as análises, mas com o sistema, pois vemos na prática o controle da Memória Cache.

8. Anexo A

```
# -*- coding: utf-8 -*-

from memoria_cache import *
from leitura_arquivo import leituraArquivo
import sys
import os

def gravaDados():
    os.system('cls' if os.name == 'nt' else 'clear')
    politicaEscrita = int(raw_input(
        '\tPolítica de Escrita : 0 - write-through;\t1 - write-back; ')
    )
    tamanhoDaLinha = int(raw_input('\tTamanho da Linha : '))
    numeroDeLinhas = int(raw_input('\tNúmero de linhas: '))
    linhasPorConjunto = int(raw_input('\tAssociatividade por conjunto: '))
    tempoAcesso = int(raw_input(
        '\tTempo de acesso quando encontra (hit-time): ')
    )
```

```

politicaSubstituicao = int(raw_input(
    '\tPolitica de Substituição: 0 - LFU;\t1 - LRU;\t2 - Aleatório;')
)

tempoMP = int(raw_input('\tTempo de leitura/escrita:'))

memoriaPrincipal = MemoriaPrincipal()
qtdeConjuntos = numeroDeLinhas/linhasPorConjunto
tamanhoTotalDaCache = numeroDeLinhas * tamanhoDaLinha
# Calcula o tamanho do endereço do conjunto;
resto = 0
aux = qtdeConjuntos
tamanhoEnderecoConjunto = 0
while resto != 1:
    aux = aux/2
    resto = aux
    tamanhoEnderecoConjunto += 1

# Calcula o tamanho do endereço da palavra
resto = 0
aux = tamanhoDaLinha
enderecoPalavra = 0
while resto != 1:
    aux = aux/2
    resto = aux
    enderecoPalavra += 1

# Calcula o rótulo
rotulo = 32 - (enderecoPalavra + tamanhoEnderecoConjunto)
memoriaCache = MemoriaCache(qtdeConjuntos, linhasPorConjunto)
enderecos = leituraArquivo()
leituras = 0
escritas = 0
leiturasNaCache = 0
leiturasNaMP = 0
escritasNaCache = 0
escritasNaMP = 0
encontrouNaCacheLeitura = 0
encontrouNaCacheEscrita = 0
encontrouNaMPLeitura = 0
encontrouNaMPEscrita = 0
for end in enderecos:
    atributos = end.split(' ')
    endereco = atributos[0]
    operacao = atributos[1]
    # Transforma em binário
    my_hexdata = endereco
    scale = 16
    # equals to hexadecimal

```

```

num_of_bits = 32
endBinario = bin(int(my_hexdata, scale))[2:].zfill(num_of_bits)
rotuloEndereco = endBinario[0:rotulo]
enderecoConjunto = endBinario[
    rotulo+1:rotulo+1+tamanhoEnderecoConjunto
]
operacao = operacao.replace('\n', '')
if operacao == "R":
    conjunto = memoriaCache.procuraConjunto(enderecoConjunto)
    leituras += 1
    # Contador de leituras
    leiturasNaCache += 1
    if conjunto:
        # Procura rotulo pelo conjunto encontrado
        retornoRotulo = conjunto.procuraRotulo(rotuloEndereco)
        if retornoRotulo:
            # Se encontrou o rótulo ocorre hit
            encontrouNaCacheLeitura += 1
        else:
            # Caso não encontrou o rótulo da erro
            conjunto.gravaRotulo(
                rotuloEndereco, politicaSubstituicao, politicaEscrita,
                memoriaPrincipal, endBinario
            )
            leiturasNaMP += 1
            encontrouNaMPLeitura += 1
    else:
        conjunto = memoriaCache.gravaConjunto(enderecoConjunto)
        if conjunto:
            conjunto.gravaRotulo(
                rotuloEndereco, politicaSubstituicao, politicaEscrita,
                memoriaPrincipal, endBinario
            )
            leiturasNaMP += 1
            encontrouNaMPLeitura += 1
else:
    escritas += 1
    conjunto = memoriaCache.procuraConjunto(enderecoConjunto)
    escritasNaCache += 1
    if not conjunto:
        conjunto = memoriaCache.gravaConjunto(enderecoConjunto)
        conjunto.gravaRotulo(
            rotuloEndereco, politicaSubstituicao, politicaEscrita,
            memoriaPrincipal, endBinario
        )
    else:
        retornoRotulo = conjunto.procuraRotulo(rotuloEndereco)
        if not retornoRotulo:
            conjunto.gravaRotulo(

```

```

        rotuloEndereco, politicaSubstituicao, politicaEscrita,
        memoriaPrincipal, endBinario
    )
    else:
        encontrouNaCacheEscrita += 1
    if politicaEscrita == 0:
        escritasNaMP += 1
        encontrouNaMPEscrita += 1

totalDeRegistros = leituras + escritas
# Seta total de escritas;
totalDeEscritas = escritas
totalDeLeituras = leituras
taxaDeAcertoCacheLeitura = 0
# Acerto Leitura Cache
if leiturasNaCache:
    taxaDeAcertoCacheLeitura = (
        (encontrouNaCacheLeitura*100.0)/leiturasNaCache
    )
# Acerto Leitura MP
if leiturasNaMP:
    taxaDeAcertoMPLeitura = (encontrouNaMPLeitura*100.0)/leiturasNaMP
if escritasNaCache:
    taxaDeAcertoCacheEscrita = (
        (encontrouNaCacheEscrita*100.0)/escritasNaCache
    )
totalLeituraEscrita = (
    float(encontrouNaCacheLeitura)+float(encontrouNaCacheEscrita)
)
taxaAcerto = totalLeituraEscrita / (leiturasNaCache + escritasNaCache)
tempoMedio = 0.0
if taxaDeAcertoCacheLeitura:
    t1 = tempoAcesso
    # tempoMedio = t1+ (1-h)*t2
    h_taxa = taxaAcerto
    tempoMedio = t1 + ((1 - h_taxa) * tempoMP)
taxaAcerto = taxaAcerto * 100.0
tempoMedio = format(tempoMedio, '.2f')
taxaDeAcertoCacheLeitura = format(taxaDeAcertoCacheLeitura, '.4f')
taxaDeAcertoCacheEscrita = format(taxaDeAcertoCacheEscrita, '.4f')
taxaAcerto = format(taxaAcerto, '.4f')

os.system('cls' if os.name == 'nt' else 'clear')
texto = ""
texto +=
    "\nDADOS DE ENTRADA:\n"
if politicaEscrita == 0:
    texto +=
        "Politica de Escrita: " + str(politicaEscrita)

```



```

        texto +=
            " - write-through\n"
else:
    texto +=
        "Politica de Escrita: " + str(politicaEscrita)
    texto +=
        " - write-back\n"
texto +=
    "Tamanho da linha: " + str(tamanhoDaLinha) + ", \n"
texto +=
    "Numero de linhas: " + str(numeroDeLinhas) + ", \n"
texto +=
    "Associatividade por conjunto: " + str(linhasPorConjunto) + "\n"
texto +=
    "Tempo de Acesso na Cache: " + str(tempoAcesso) + ", \n"

if politicaSubstituicao == 0:
    texto +=
        "Politica de Substituição: " + str(politicaSubstituicao)
    texto +=
        " - LFU\n"
elif politicaSubstituicao == 1:
    texto +=
        "Politica de Substituição: " + str(politicaSubstituicao)
    texto += " - LRU\n"
else:
    texto +=
        "Politica de Substituição: " + str(politicaSubstituicao)
    texto += " - Aleatorio\n"
texto +=
    "Tempo de Acesso na memória principal: " + str(tempoMP) + "ns\n"
texto +=
    "\nRESULTADOS:\n"
texto +=
    "Tamanho da Cache: " + str(tamanhoTotalDaCache) + "\n"
texto +=
    "Total de endereços no arquivo de entrada:\n"
texto +=
    "Total de registros: " + str(totalDeRegistros) + "\n"
texto +=
    "Total de leituras: " + str(totalDeLeituras) + "\n"
texto +=
    "Total de escritas: " + str(totalDeEscritas) + "\n"
texto +=
    "Dados da Cache:\n"
texto +=
    "Total de leituras: " + str(leiturasNaCache) + "\n"
texto +=
    "Total de acertos: " + str(encontrouNaCacheLeitura) + "\n"

```

```

texto +=
    "Taxa de acerto Leitura: " + str(taxaDeAcertoCacheLeitura) + "%\n"
texto +=
    "Total de escritas: " + str(escritasNaCache) + "\n"
texto +=
    "Total de acertos: " + str(encontrouNaCacheEscrita) + "\n"
texto +=
    "Taxa de acerto Escrita: "+str(taxaDeAcertoCacheEscrita) + "%\n"
texto +=
    "Taxa de acertos: " + str(taxaAcerto) + "%\n"
texto +=
    "Tempo médio de acesso da cache: " + str(tempoMedio) + "ns\n"
texto +=
    "Dados da Memória Principal:\n"
texto +=
    "Total de escritas: " + str(encontrouNaMPEscrita) + "\n"
texto +=
    "Total de leituras: " + str(encontrouNaMPLeitura) + "\n"
texto +=
    "Acessos: " + str(leiturasNaMP+escritasNaMP) + "\n"

print texto

    EscreveResultados(texto)

    return

def EscreveResultados(texto):
    arquivo = open('./results.txt', 'w')
    arquivo.writelines(texto)
    arquivo.close()
    return

def main():
    gravaDados()
    return

if __name__ == '__main__':
    main()

```

9. Anexo B

Política de Escrita	0 - write-through
Tamanho da linha	128
Associatividade por conjunto	4
Tempo de Acesso na Cache	10
Política de Substituição	1 - LRU
Tempo de Acesso na memória principal	60 ns

Figura 5. Tamanho da Cache - Dados Fixos

Numero de linhas	16
Tamanho da Cache	2KBytes
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	33278
Taxa de acerto Leitura	73.8592%
Total de escritas	6144
Total de acertos	4096
Taxa de acerto Escrita	66.6667%
Taxa de acertos	72.9961%
Tempo médio de acesso da cache	26.20ns
Total de escritas – MP	6144
Total de leituras – MP	11778
Acessos – MP	17922

Figura 6. Tamanho da Cache - 2Kbytes

Numero de linhas	32
Tamanho da Cache	4KBytes
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45029
Taxa de acerto Leitura	99.9401%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9395%
Tempo médio de acesso da cache	10.04ns
Total de escritas – MP	6144
Total de leituras – MP	27
Acessos – MP	6171

Figura 7. Tamanho da Cache - 4KBytes

Numero de linhas	64
Tamanho da Cache	8KBytes
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas – MP	6144
Total de leituras – MP	25
Acessos – MP	6169

Figura 8. Tamanho da Cache - 8KBytes

Numero de linhas	128
Tamanho da Cache	16KBytes
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45022
Taxa de acerto Leitura	99.9245%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9258%
Tempo médio de acesso da cache	10.04ns
Total de escritas – MP	6144
Total de leituras – MP	34
Acessos – MP	6178

Figura 9. Tamanho da Cache - 16KBytes

Numero de linhas	256
Tamanho da Cache	32KBytes
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45013
Taxa de acerto Leitura	99.9046%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9082%
Tempo médio de acesso da cache	10.06ns
Total de escritas – MP	6144
Total de leituras – MP	43
Acessos – MP	6187

Figura 10. Tamanho da Cache - 32KBytes

Politica de Escrita	0 - <u>write-through</u>
Associatividade por conjunto	2
Tempo de Acesso na Cache	10
Politica de Substituição	1 – LRU
Tempo de Acesso na memória principal	60ns

Figura 11. Impacto da Tamanho do Bloco

Tamanho da linha	4
Numero de linhas	2048
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	44992
Taxa de acerto Leitura	99.8580%
Total de escritas	6144
Total de acertos	6138
Taxa de acerto Escrita	99.9023%
Taxa de acertos	99.8633%
Tempo médio de acesso da cache	10.08ns
Total de escritas – MP	6144
Total de leituras – MP	64
Acessos – MP	6208

Figura 12. Tamanho do Bloco - 4 Bytes

Tamanho da linha	8
Numero de linhas	1024
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	44993
Taxa de acerto Leitura	99.8602%
Total de escritas	6144
Total de acertos	6138
Taxa de acerto Escrita	99.9023%
Taxa de acertos	99.8652%
Tempo médio de acesso da cache	10.08ns
Total de escritas – MP	6144
Total de leituras – MP	63
Acessos – MP	6207

Figura 13. Tamanho do Bloco - 8 Bytes

Tamanho da linha	16
Numero de linhas	512
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	44993
Taxa de acerto Leitura	99.8602%
Total de escritas	6144
Total de acertos	6138
Taxa de acerto Escrita	99.9023%
Taxa de acertos	99.8652%
Tempo médio de acesso da cache	10.08ns
Total de escritas – MP	6144
Total de leituras – MP	63
Acessos – MP	6207

Figura 14. Tamanho do Bloco - 16 Bytes

Tamanho da linha	32
Numero de linhas	256
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	44999
Taxa de acerto Leitura	99.8735%
Total de escritas	6144
Total de acertos	6139
Taxa de acerto Escrita	99.9186%
Taxa de acertos	99.8789%
Tempo médio de acesso da cache	10.07ns
Total de escritas – MP	6144
Total de leituras – MP	57
Acessos – MP	6201

Figura 15. Tamanho do Bloco - 32 Bytes

Tamanho da linha	64
Numero de linhas	128
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45001
Taxa de acerto Leitura	99.8779%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.8848%
Tempo médio de acesso da cache	10.07ns
Total de escritas – MP	6144
Total de leituras – MP	55
Acessos – MP	6199

Figura 16. Tamanho do Bloco - 64 Bytes

Tamanho da linha	128
Numero de linhas	64
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45021
Taxa de acerto Leitura	99.9223%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9238%
Tempo médio de acesso da cache	10.05ns
Total de escritas – MP	6144
Total de leituras – MP	35
Acessos – MP	6179

Figura 17. Impacto da Tamanho do Bloco - 128 Bytes

Tamanho da linha	256
Numero de linhas	32
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45028
Taxa de acerto Leitura	99.9379%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9375%
Tempo médio de acesso da cache	10.04ns
Total de escritas – MP	6144
Total de leituras – MP	28
Acessos – MP	6172

Figura 18. Tamanho do Bloco - 256 Bytes

Tamanho da linha	512
Numero de linhas	16
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	37372
Taxa de acerto Leitura	82.9457%
Total de escritas	6144
Total de acertos	4096
Taxa de acerto Escrita	66.6667%
Taxa de acertos	80.9922%
Tempo médio de acesso da cache	21.40ns
Total de escritas – MP	6144
Total de leituras – MP	7684
Acessos – MP	13828

Figura 19. Tamanho do Bloco - 512 Bytes

Tamanho da linha	1024
Numero de linhas	8
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	36351
Taxa de acerto Leitura	80.6796%
Total de escritas	6144
Total de acertos	4096
Taxa de acerto Escrita	66.6667%
Taxa de acertos	78.9980%
Tempo médio de acesso da cache	22.60ns
Total de escritas – MP	6144
Total de leituras – MP	8705
Acessos – MP	14849

Figura 20. Tamanho do Bloco - 1 KBytes

Tamanho da linha	2048
Numero de linhas	4
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	32768
Taxa de acerto Leitura	72.7273%
Total de escritas	6144
Total de acertos	4096
Taxa de acerto Escrita	66.6667%
Taxa de acertos	72.0000%
Tempo médio de acesso da cache	26.80ns
Total de escritas – MP	6144
Total de leituras – MP	12288
Acessos – MP	18432

Figura 21. Tamanho do Bloco - 2KBytes

Política de Escrita	1 – <u>wite-back</u>
Tamanho da linha	128
Numero de linhas:	64
Tempo de Acesso na Cache	10
Política de Substituição	1 – LRU
Tempo de Acesso na memória principal	60ns

Figura 22. Impacto Associatividade

Associatividade por conjunto	2
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45021
Taxa de acerto Leitura	99.9223%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9238%
Tempo médio de acesso da cache	10.05ns
Total de escritas – MP	0
Total de leituras -MP	35
Acessos – MP	35

Figura 23. Associatividade 2 por Conjunto

Associatividade por conjunto	4
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas – MP	0
Total de leituras -MP	25
Acessos – MP	25

Figura 24. Associatividade 4 por Conjunto

Associatividade por conjunto	8
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas – MP	0
Total de leituras -MP	25
Acessos – MP	25

Figura 25. Associatividade 8 por Conjunto

Associatividade por conjunto	1
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45009
Taxa de acerto Leitura	99.8957%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9004%
Tempo médio de acesso da cache	10.06ns
Total de escritas – MP	0
Total de leituras -MP	47
Acessos – MP	47

Figura 26. Associatividade 16 por Conjunto

Associatividade por conjunto	32
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45026
Taxa de acerto Leitura	99.9334%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9336%
Tempo médio de acesso da cache	10.04ns
Total de escritas – MP	0
Total de leituras -MP	30
Acessos – MP	30

Figura 27. Associatividade 32 por Conjunto

Política de Escrita	0 - <u>write-through</u>
Tamanho da linha	128
Tempo de Acesso na Cache	10
Tempo de Acesso na memória principal	60ns
Associatividade por conjunto	4

Figura 28. Política de Substituição

Politica de Substituição	1 – LRU
Numero de linhas	16
Tamanho da Cache	2048
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	33278
Taxa de acerto Leitura	73.8592%
Total de escritas	6144
Total de acertos	4096
Taxa de acerto Escrita	66.6667%
Taxa de acertos	72.9961%
Tempo médio de acesso da cache	26.20ns
Total de escritas – MP	6144
Total de leituras – MP	11778
Acessos – MP	17922

Figura 29. Politica de Substituição - LRU 2KBytes

Politica de Substituição	1 – LRU
Numero de linhas	32
Tamanho da Cache	4096
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45029
Taxa de acerto Leitura	99.9401%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9395%
Tempo médio de acesso da cache	10.04ns
Total de escritas – MP	6144
Total de leituras – MP	27
Acessos – MP	6171

Figura 30. Politica de Substituição - LRU 4KBytes

Política de Substituição	1 – LRU
Numero de linhas	64
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas – MP	6144
Total de leituras – MP	25
Acessos – MP	6169

Figura 31. Política de Substituição - LRU 8KBytes

Política de Substituição	1 – LRU
Numero de linhas	128
Tamanho da Cache	16384
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45022
Taxa de acerto Leitura	99.9245%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9258%
Tempo médio de acesso da cache	10.04ns
Total de escritas – MP	6144
Total de leituras – MP	34
Acessos – MP	6178

Figura 32. Política de Substituição - LRU 16KBytes

Política de Substituição	0 – LFU
Numero de linhas	16
Tamanho da Cache	2048
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	37360
Taxa de acerto Leitura	82.9190%
Total de escritas	6144
Total de acertos	4607
Taxa de acerto Escrita	74.9837%
Taxa de acertos	81.9668%
Tempo médio de acesso da cache	20.82ns
Total de escritas – MP	6144
Total de leituras – MP	7696
Acessos – MP	13840

A Imagem da Política de Substituição LFU 2KBytes

Figura 33. Política de Substituição - LFU 2KBytes

Política de Substituição	0 – LFU
Numero de linhas	32
Tamanho da Cache	4096
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	44520
Taxa de acerto Leitura	98.8104%
Total de escritas	6144
Total de acertos	5118
Taxa de acerto Escrita	83.3008%
Taxa de acertos	96.9492%
Tempo médio de acesso da cache	11.83ns
Total de escritas – MP	6144
Total de leituras – MP	536
Acessos – MP	6680

Figura 34. Política de Substituição - LFU 4KBytes

Política de Substituição	0 – LFU
Numero de linhas	64
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas – MP	6144
Total de leituras – MP	25
Acessos – MP	6169

Figura 35. Política de Substituição - LFU 8KBytes

Politica de Substituição	0 – LFU
Numero de linhas	128
Tamanho da Cache	16384
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45022
Taxa de acerto Leitura	99.9245%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9258%
Tempo médio de acesso da cache	10.04ns
Total de escritas – MP	6144
Total de leituras – MP	34
Acessos – MP	6178

Figura 36. Política de Substituição - LFU 16KBytes

Política de Substituição	2- Aleatório
Numero de linhas	16
Tamanho da Cache	2048
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas	6144
Total de leituras	25
Acessos	6169

Figura 37. Política de Substituição - Aleatório 2KBytes

Política de Substituição	2- Aleatório
Numero de linhas	32
Tamanho da Cache	4096
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas	6144
Total de leituras	25
Acessos	6169

Figura 38. Política de Substituição - Aleatório 4KBytes

Política de Substituição	2- Aleatório
Numero de linhas	64
Tamanho da Cache	8192
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45031
Taxa de acerto Leitura	99.9445%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9434%
Tempo médio de acesso da cache	10.03ns
Total de escritas	6144
Total de leituras	25
Acessos	6169

Figura 39. Política de Substituição - Aleatório 8KBytes

Política de Substituição	2- Aleatório
Numero de linhas	128
Tamanho da Cache	16384
Total de registros	51200
Total de leituras	45056
Total de escritas	6144
Total de leituras	45056
Total de acertos	45022
Taxa de acerto Leitura	99.9245%
Total de escritas	6144
Total de acertos	6140
Taxa de acerto Escrita	99.9349%
Taxa de acertos	99.9258%
Tempo médio de acesso da cache	10.04ns
Total de escritas	6144
Total de leituras	34
Acessos	6178

Figura 40. Política de Substituição - Aleatório 16KBytes