

**Universidade Federal de Santa Catarina - Departamento de Informática e Estatística**  
**INE 5411 - Organização de Computadores**  
**Roteiro do Laboratório 7 - Impacto da memória cache no desempenho**

### 1. Objetivo

O objetivo desta aula é estudar o impacto de memórias cache no desempenho. Para isso, serão executados dois programas diferentes que resolvem o mesmo problema: a soma dos elementos de uma matriz. Supõe-se que a matriz foi alocada de forma que **os elementos de uma mesma linha tenham sido armazenados sequencialmente de forma contígua em memória**, como se tivessem sido compilados a partir de um programa-fonte escrito na linguagem C. Cada programa usa uma estratégia diferente para acessar os elementos da matriz: o primeiro programa (`row-major.asm`) percorre os elementos da matriz ao longo de suas linhas, enquanto o segundo programa (`column-major.asm`) percorre os elementos da matriz ao longo de suas colunas.

O programa *row-major.asm* corresponde ao seguinte código-fonte:

```
soma = 0;
for ( linha = 0; linha < X_linhas; linha++ ) {
    for ( coluna = 0; coluna < Y_colunas; coluna++ ) {
        soma = soma + dados[linha][coluna];
    }
}
```

O programa *column-major.asm* corresponde ao seguinte código-fonte:

```
soma = 0;
for ( coluna = 0; coluna < Y_colunas; coluna++ ) {
    for ( linha = 0; linha < X_linhas; linha++ ) {
        soma = soma + dados[linha][coluna];
    }
}
```

### 2. Simulador de cache de dados

O MARS 4.5 possui uma ferramenta para realizar a simulação da cache de dados (*Tools -> Data Cache Simulator*). A janela principal é apresentada na Figura 1. Tal ferramenta está dividida em três partes:

- 1. Organização (Cache Organization):** permite configurar tamanhos e políticas de posicionamento e substituição.
- 2. Desempenho (Cache Performance):** possibilita avaliar o desempenho da cache para uma determinada configuração.
- 3. Controle (Tool Control):** permite reinicializar o simulador de cache (*Reset*) ou conectá-lo (*Connect to MARS*) ao simulador do processador, fazendo com que cada simulador reaja a eventos produzidos pelo outro.

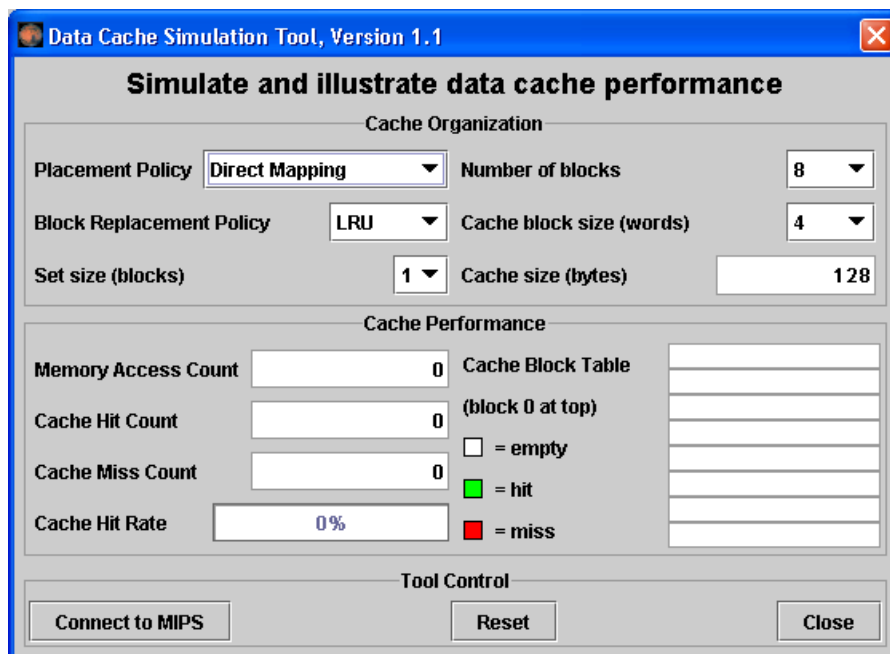


Figura 1: Janela principal do simulador de memória cache de dados.

### 3. Passos para viabilizar a análise de desempenho

Para os experimentos desta aula prática, você vai primeiramente adaptar os arquivos fornecidos (`row-major-codigo-base.asm` e `column-major-codigo-base.asm`, de acordo com o tamanho da matriz a ser definido no Relatório (parâmetros `X_linhas`, `Y_colunas` e `XY_elementos`). O Relatório vai especificar vários experimentos em que você deverá configurar a organização da cache, habilitar a simulação simultânea de cache e processador, executar o programa e monitorar seu desempenho. Você repetirá esse mesmo processo para diferentes organizações de cache. Para isso, siga os passos abaixo:

1. Abra o arquivo do programa e preencha as informações faltantes, conforme especificado no Relatório.
2. Efetue a montagem do programa, selecionando a opção “*Assemble*”.
3. Selecione a ferramenta de simulação de cache de dados (*Tools -> Data Cache Simulator*).
4. Clique no botão “*Connect to MIPS*”.
5. No simulador MARS, a velocidade pode ser ajustada para “n” instruções por segundo junto ao “*Run speed slider*”, possibilitando assim uma melhor visualização do funcionamento da memória cache.
6. Para iniciar uma execução selecione a opção “*Run*” no menu principal do simulador MARS.
7. **Toda vez que houver mudança na configuração da cache, ambos os simuladores devem ser reinicializados (*Reset*).**

**Para testes com caches associativas por conjunto, não se esqueça de atualizar o tamanho dos conjuntos (*set size*) a cada mudança no número de blocos (*number of blocks*).**

**Universidade Federal de Santa Catarina - Centro Tecnológico - Departamento de Informática e Estatística**  
**INE 5411 - Organização de Computadores**

**Relatório do Laboratório 7 - Impacto da memória cache no desempenho**

Aluno ou aluna 1:

Assinatura:

Aluno ou aluna 2:

Assinatura:

**X\_linhas = 600. Y\_colunas = 5. XY\_total = 3000.**

**[1,0] Questão 1.1:** Qual a taxa de acertos  $h$  (*hit rate*) e *cache hit count* obtida para cada um dos tamanhos de bloco abaixo?

a) Blocos de 2 palavras.  $h =$  %. Hit count =      c) Blocos de 8 palavras.  $h =$  %. Hit count =

b) Blocos de 4 palavras.  $h =$  %. Hit count =      d) Blocos de 16 palavras.  $h =$  %. Hit count =

**[0,5] Questão 1.2:** Explique o comportamento da mudança de taxa de acertos com o aumento do tamanho do bloco. A explicação deve envolver uma discussão sobre mudanças na localidade espacial e/ou temporal.

**[1,0] Questão 2.1:** Qual a taxa de acertos  $h$  (*hit rate*) e *cache hit count* obtida para cada um dos tamanhos de bloco abaixo?

a) Blocos de 2 palavras.  $h =$  %. Hit count =      c) Blocos de 8 palavras.  $h =$  %. Hit count =

b) Blocos de 4 palavras.  $h =$  %. Hit count =      d) Blocos de 16 palavras.  $h =$  %. Hit count =

**[0,5] Questão 2.2:** Informe se a cache de dados consegue explorar localidade temporal (sim ou não), justificando.

**Resposta:**      **Justificativa:**

**[0,5] Questão 2.3:** Explique a taxa de acerto observada para os itens a) e b) da Questão 2.1, analisando se os elementos **dados[linha][coluna]** e **dados[linha+1][coluna]** podem pertencer ao mesmo bloco da cache nesses dois cenários.

**[0,5] Questão 2.4:** Explique a razão das taxas de acertos para o item c) e d) da Questão 2.1 não serem nulas, argumentando sobre qual a relação entre o número de colunas e o tamanho do bloco.

**[1,0] Questão 3.1:** Informe as características da configuração de cache encontrada. Para o cálculo de tamanho total, considere que a cache usa 1 bit de validade para cada bloco. O tamanho total da cache inclui bits de dados, tags e validade.

Política de mapeamento	Taxa de acerto (%)	Tamanho da cache (em bytes)	Tamanho dos conjuntos (em blocos)	Nº total de blocos na cache	Tamanho dos blocos (em palavras)	Tamanho do campo tag (em bits)	Tamanho total (em bits)

**[1,0] Questão 3.2:** Considere que o tempo de acerto (*hit time*) da cache encontrada é de 2 ciclos e que a penalidade de falha (*miss penalty*) é de 200 ciclos. Tomando como base o número de acertos, faltas e acessos, informe o tempo médio de acesso a esta cache.

**[1,0] Questão 3.3:** Tomando como hipótese de que o CPI ideal do processador (quando todos acessos à memória resultam em acertos na cache) é 2,5, estime o CPI medido durante a execução do programa.

**[3,0] Questão 4.1:** Informe as características da configuração de cache encontrada. Para o cálculo de tamanho total, considere que a cache usa 1 bit de validade para cada bloco. O tamanho total da cache inclui bits de dados, tags e validade.

Política de mapeamento	Taxa de acerto (%)	Tamanho da cache (em bytes)	Tamanho dos conjuntos (em blocos)	Nº total de blocos na cache	Tamanho dos blocos (em palavras)	Tamanho do campo tag (em bits)	Tamanho total (em bits)
Totalmente associativo			16				
4-associativo				16	64		
Mapeamento direto					128		

Aluno ou aluna 1:

Assinatura:

Aluno ou aluna 2:

Assinatura:

**X\_linhas = 700. Y\_colunas = 6. XY\_total = 4200.**

**[1,0] Questão 1.1:** Qual a taxa de acertos  $h$  (*hit rate*) e *cache hit count* obtida para cada um dos tamanhos de bloco abaixo?

- a) Blocos de 2 palavras.  $h =$  %. Hit count =      c) Blocos de 8 palavras.  $h =$  %. Hit count =  
b) Blocos de 4 palavras.  $h =$  %. Hit count =      d) Blocos de 16 palavras.  $h =$  %. Hit count =

**[0,5] Questão 1.2:** Explique o comportamento da mudança de taxa de acertos com o aumento do tamanho do bloco. A explicação deve envolver uma discussão sobre mudanças na localidade espacial e/ou temporal.

**[1,0] Questão 2.1:** Qual a taxa de acertos  $h$  (*hit rate*) e *cache hit count* obtida para cada um dos tamanhos de bloco abaixo?

- a) Blocos de 2 palavras.  $h =$  %. Hit count =      c) Blocos de 8 palavras.  $h =$  %. Hit count =  
b) Blocos de 4 palavras.  $h =$  %. Hit count =      d) Blocos de 16 palavras.  $h =$  %. Hit count =

**[0,5] Questão 2.2:** Informe se a cache de dados consegue explorar localidade temporal (sim ou não), justificando.

**Resposta:**      **Justificativa:**

**[0,5] Questão 2.3:** Explique a taxa de acerto observada para os itens a) e b) da Questão 2.1, analisando se os elementos **dados[linha][coluna]** e **dados[linha+1][coluna]** podem pertencer ao mesmo bloco da cache nesses dois cenários.

**[0,5] Questão 2.4:** Explique a razão das taxas de acertos para o item c) e d) da Questão 2.1 não serem nulas, argumentando sobre qual a relação entre o número de colunas e o tamanho do bloco.

**[1,0] Questão 3.1:** Informe as características da configuração de cache encontrada. Para o cálculo de tamanho total, considere que a cache usa 1 bit de validade para cada bloco. O tamanho total da cache inclui bits de dados, tags e validade.

Política de mapeamento	Taxa de acerto (%)	Tamanho da cache (em bytes)	Tamanho dos conjuntos (em blocos)	Nº total de blocos na cache	Tamanho dos blocos (em palavras)	Tamanho do campo tag (em bits)	Tamanho total (em bits)

**[1,0] Questão 3.2:** Considere que o tempo de acerto (*hit time*) da cache encontrada é de 2 ciclos e que a penalidade de falha (*miss penalty*) é de 200 ciclos. Tomando como base o número de acertos, faltas e acessos, informe o tempo médio de acesso a esta cache.

**[1,0] Questão 3.3:** Tomando como hipótese de que o CPI ideal do processador (quando todos acessos à memória resultam em acertos na cache) é 2,5, estime o CPI medido durante a execução do programa.

**[3,0] Questão 4.1:** Informe as características da configuração de cache encontrada. Para o cálculo de tamanho total, considere que a cache usa 1 bit de validade para cada bloco. O tamanho total da cache inclui bits de dados, tags e validade.

Política de mapeamento	Taxa de acerto (%)	Tamanho da cache (em bytes)	Tamanho dos conjuntos (em blocos)	Nº total de blocos na cache	Tamanho dos blocos (em palavras)	Tamanho do campo tag (em bits)	Tamanho total (em bits)
Totalmente associativo			8				
4-associativo				16	128		
Mapeamento direto					512		

### **GABARITO**

**X\_linhas = 600. Y\_colunas = 5. XY\_total = 3000.**

**X\_linhas = 700. Y\_colunas = 6. XY\_total = 4200.**

**[1,0] Questão 1.1:** Qual a taxa de acertos  $h$  (*hit rate*) e *cache hit count* obtida para cada um dos tamanhos de bloco abaixo?

- a) Blocos de 2 palavras.  $h = 50$  %. Hit count = **1500 / 2100**  
b) Blocos de 4 palavras.  $h = 75$  %. Hit count = **2250 / 3150**

- c) Blocos de 8 palavras.  $h = 88$  %. Hit count = **2625 / 3675**  
d) Blocos de 16 palavras.  $h = 94$  %. Hit count = **2812 / 3937**

[0,5] **Questão 1.2:** Explique o comportamento da mudança de taxa de acertos com o aumento do tamanho do bloco. A explicação deve envolver uma discussão sobre mudanças na localidade espacial e/ou temporal.

**Como os elementos de uma linha da matriz ocupam endereços contíguos, com o aumento do tamanho do bloco aumenta a captura da localidade espacial e, conseqüentemente, cresce a taxa de acertos.**

[1,0] **Questão 2.1:** Qual a taxa de acertos  $h$  (*hit rate*) e *cache hit count* obtida para cada um dos tamanhos de bloco abaixo?

- a) Blocos de 2 palavras.  $h = 0$  %. Hit count = **0 / 0**  
b) Blocos de 4 palavras.  $h = 0$  %. Hit count = **0 / 0**  
c) Blocos de 8 palavras.  $h = 38 / 25\%$ . Hit count = **1125 / 1050**  
d) Blocos de 16 palavras.  $h = 69 / 62\%$ . Hit count = **2060 / 2622**

[0,5] **Questão 2.2:** Informe se a cache de dados consegue explorar localidade temporal (sim ou não), justificando.

**Resposta: Não Justificativa: Como nenhum elemento da matriz é revisitado, nenhuma localidade temporal pode ser explorada.**

[0,5] **Questão 2.3:** Explique a taxa de acerto observada para os itens a) e b) da Questão 2.1, analisando se os elementos **dados[linha][coluna]** e **dados[linha+1][coluna]** podem pertencer ao mesmo bloco da cache nesses dois cenários.

**Embora os elementos da matriz sejam alocados contiguamente por linha, eles são visitados por coluna e nem sempre é possível explorar a localidade espacial. Para os casos a) e b), os elementos dados[linha][coluna] e dados[linha+1][coluna] nunca pertencem ao mesmo bloco.**

[0,5] **Questão 2.4:** Explique a razão das taxas de acertos para o item c) e d) da Questão 2.1 não serem nulas, argumentando sobre qual a relação entre o número de colunas e o tamanho do bloco.

**Como a matriz é visitada por coluna, a cache só consegue explorar localidade espacial quando o número de palavras por bloco é maior do que o número de colunas, como acontece nos itens c) e d).**

[1,0] **Questão 3.1:** Informe as características da configuração de cache encontrada. Para o cálculo de tamanho total, considere que a cache usa 1 bit de validade para cada bloco. O tamanho total da cache inclui bits de dados, tags e validade.

Política de mapeamento	Taxa de acerto (%)	Tamanho da cache (em bytes)	Tamanho dos conjuntos (em blocos)	Nº total de blocos na cache	Tamanho dos blocos (em palavras)	Tamanho do campo tag (em bits)	Tamanho total (em bits)
<b>M. Direto</b>	<b>100</b>	<b>2048</b>	<b>1</b>	<b>2</b>	<b>256</b>	<b>32-1-8-2=21</b>	<b>(1+21+256*32)*2 = 16428</b>

[1,0] **Questão 3.2:** Considere que o tempo de acerto (*hit time*) da cache encontrada é de 2 ciclos e que a penalidade de falha (*miss penalty*) é de 200 ciclos. Tomando como base o número de acertos, faltas e acessos, informe o tempo médio de acesso a esta cache.

**3k acessos, 2988 hits, 12 misses. Tempo médio =  $(3000*2 + 12*200)/3000 = (6000 + 2400)/3000 = 2,8$  ciclos**

**4,2k acessos, 4183 hits, 17 misses. Tempo médio =  $(4200*2 + 17*200)/4200 = (8400 + 3400)/4200 = \sim 2,81$  ciclos**

[1,0] **Questão 3.3:** Tomando como hipótese de que o CPI ideal do processador (quando todos acessos à memória resultam em acertos na cache) é 2,5, estime o CPI medido durante a execução do programa.

**CPI medido = CPI ideal + ciclos por falhas/total de instruções =  $2,5 + 200*12/37211 = 2,5 + 2400/37211 = 2,5 + \sim 0,065 = 2,565$**

**CPI medido = CPI ideal + ciclos por falhas/total de instruções =  $2,5 + 200*17/51111 = 2,5 + 3400/51111 = 2,5 + \sim 0,067 = \sim 2,567$**

[3,0] **Questão 4.1:** Informe as características da configuração de cache encontrada. Para o cálculo de tamanho total, considere que a cache usa 1 bit de validade para cada bloco. O tamanho total da cache inclui bits de dados, tags e validade.

**X\_linhas = 600. Y\_colunas = 5. XY\_total = 3000.**

Política de mapeamento	Taxa de acerto (%)	Tamanho da cache (em bytes)	Tamanho dos conjuntos (em blocos)	Nº total de blocos na cache	Tamanho dos blocos (em palavras)	Tamanho do campo tag (em bits)	Tamanho total (em bits)
Totalmente associativo	<b>100</b>	<b>16384</b>	<b>16</b>	<b>16</b>	<b>256</b>	<b>32-0-8-2=22</b>	<b>(1+22+256*32)*16 = 131440</b>

4-associativo	92	4096	4	16	64	32-2-6-2=22	$(1+22+64*32)*16 = 33136$
Mapeamento direto	99	16384	1	32	128	32-5-7-2=18	$(1+18+128*32)*32 = 131680$

**X\_linhas = 700. Y\_colunas = 6. XY\_total = 4200.**

Política de mapeamento	Taxa de acerto (%)	Tamanho da cache (em bytes)	Tamanho dos conjuntos (em blocos)	Nº total de blocos na cache	Tamanho dos blocos (em palavras)	Tamanho do campo tag (em bits)	Tamanho total (em bits)
Totalmente associativo	100	32768	8	8	1024	32-0-10-2=20	$(1+20+1024*32)*8 = 262312$
4-associativo	95	8192	4	16	128	32-2-7-2=21	$(1+21+128*32)*16 = 65888$
Mapeamento direto	100	16384	1	8	512	32-3-9-2=18	$(1+18+512*32)*8 = 131224$

C