

## Relatório do Laboratório 07

Alunos: Vinicius Henrique Ribeiro (23200351) e Lucas Furlanetto Pascoali (23204339)

Professor: Marcelo Daniel Berejuck

Disciplina: Organização de Computadores I

### Questão 3

A questão pedia para testarmos os dois algoritmos implementados nas questões 1 e 2. O primeiro somava uma matriz A com uma B transposta da maneira tradicional. O segundo usava uma implementação de divisão da matriz principal em matrizes menores de tamanho *block\_size*. Em todos os testes foram usados matrizes quadradas de tamanho 300 (300x300).

No segundo algoritmo, nós definimos o *block\_size* (subdivisão da matriz principal) com um tamanho de 30 (30x30).

Data Cache Simulation Tool, Version 1.2

---

**Simulate and illustrate data cache performance**

---

Cache Organization

Placement Policy	Direct Mapping	Number of blocks	128
Block Replacement Policy	LRU	Cache block size (words)	4
Set size (blocks)	1	Cache size (bytes)	2048

---

Cache Performance

Memory Access Count	270000	Cache Block Table (block 0 at top)  <input type="checkbox"/> = empty <input checked="" type="checkbox"/> = hit <input type="checkbox"/> = miss	
Cache Hit Count	156812		
Cache Miss Count	113188		
Cache Hit Rate	58%		

---

Runtime Log

☐ Enabled

---

Tool Control

Disconnect from MIPSResetClose

A imagem acima testa o primeiro algoritmo, utilizando uma cache com blocos do tamanho de 4 *words*. Sua taxa de hit ficou em 58%.

Data Cache Simulation Tool, Version 1.2

### Simulate and illustrate data cache performance

#### Cache Organization

Placement Policy: **Direct Mapping** Number of blocks: **128**

Block Replacement Policy: **LRU** Cache block size (words): **8**

Set size (blocks): **1** Cache size (bytes): **4096**

#### Cache Performance

Memory Access Count: **270000** Cache Block Table (block 0 at top)

Cache Hit Count: **168062**

Cache Miss Count: **101938**

Cache Hit Rate: **62%**

☐ = empty  
☒ = hit  
☐ = miss

#### Runtime Log

☐ Enabled

#### Tool Control

**Disconnect from MIPS** **Reset** **Close**

A imagem acima testa o primeiro algoritmo, utilizando uma cache com blocos do tamanho de 8 words. Sua taxa de hit ficou em 62%.

Data Cache Simulation Tool, Version 1.2

### Simulate and illustrate data cache performance

#### Cache Organization

Placement Policy
Direct Mapping

Number of blocks
128

Block Replacement Policy
LRU

Cache block size (words)
16

Set size (blocks)
1

Cache size (bytes)
8192

#### Cache Performance

Memory Access Count
270000

Cache Hit Count
175246

Cache Miss Count
94754

Cache Hit Rate

65%

Cache Block Table

(block 0 at top)

= empty

= hit

= miss

#### Runtime Log

☐ Enabled

#### Tool Control

Disconnect from MIPS

Reset

Close

A imagem acima testa o primeiro algoritmo, utilizando uma cache com blocos do tamanho de 16 *words*. Sua taxa de hit ficou em 65%.

Data Cache Simulation Tool, Version 1.2

### Simulate and illustrate data cache performance

#### Cache Organization

Placement Policy: **Direct Mapping** Number of blocks: **128**

Block Replacement Policy: **LRU** Cache block size (words): **32**

Set size (blocks): **1** Cache size (bytes): **16384**

#### Cache Performance

Memory Access Count: **270000** Cache Block Table (block 0 at top)

Cache Hit Count: **179147**

Cache Miss Count: **90853**

Cache Hit Rate: **66%**

☐ = empty  
☒ = hit  
☐ = miss

#### Runtime Log

☐ Enabled

#### Tool Control

**Disconnect from MIPS** **Reset** **Close**

A imagem acima testa o primeiro algoritmo, utilizando uma cache com blocos do tamanho de 32 *words*. Sua taxa de hit ficou em 66%.

Data Cache Simulation Tool, Version 1.2

### Simulate and illustrate data cache performance

#### Cache Organization

Placement Policy: **Direct Mapping** Number of blocks: **128**

Block Replacement Policy: **LRU** Cache block size (words): **64**

Set size (blocks): **1** Cache size (bytes): **32768**

#### Cache Performance

Memory Access Count: **270000** Cache Block Table (block 0 at top)

Cache Hit Count: **179862**

Cache Miss Count: **90138**

Cache Hit Rate: **67%**

☐ = empty  
☒ = hit  
☐ = miss

#### Runtime Log

☐ Enabled

#### Tool Control

**Disconnect from MIPS** **Reset** **Close**

A imagem acima testa o primeiro algoritmo, utilizando uma cache com blocos do tamanho de 64 *words*. Sua taxa de hit ficou em 67%.

Data Cache Simulation Tool, Version 1.2

---

**Simulate and illustrate data cache performance**

---

**Cache Organization**

Placement Policy	Direct Mapping	Number of blocks	128
Block Replacement Policy	LRU	Cache block size (words)	4
Set size (blocks)	1	Cache size (bytes)	2048

---

**Cache Performance**

Memory Access Count	270000	Cache Block Table (block 0 at top) <input type="checkbox"/> = empty <input checked="" type="checkbox"/> = hit <input type="checkbox"/> = miss	
Cache Hit Count	216682		
Cache Miss Count	53318		
Cache Hit Rate	80%		

---

**Runtime Log**

☐ Enabled

---

**Tool Control**

Disconnect from MIPS

Reset

Close

A imagem acima testa o segundo algoritmo, utilizando uma cache com blocos do tamanho de 4 words. Sua taxa de hit ficou em 80%.

Data Cache Simulation Tool, Version 1.2

---

**Simulate and illustrate data cache performance**

---

**Cache Organization**

Placement Policy	Direct Mapping	Number of blocks	128
Block Replacement Policy	LRU	Cache block size (words)	8
Set size (blocks)	1	Cache size (bytes)	4096

---

**Cache Performance**

Memory Access Count	270000	<b>Cache Block Table</b> (block 0 at top) <input type="checkbox"/> = empty <input checked="" type="checkbox"/> = hit <input type="checkbox"/> = miss
Cache Hit Count	239297	
Cache Miss Count	30703	
Cache Hit Rate	89%	

---

**Runtime Log**

☐ Enabled

---

**Tool Control**

Disconnect from MIPS

Reset

Close

A imagem acima testa o segundo algoritmo, utilizando uma cache com blocos do tamanho de 8 words. Sua taxa de hit ficou em 89%.



Data Cache Simulation Tool, Version 1.2

---

**Simulate and illustrate data cache performance**

---

**Cache Organization**

Placement Policy	Direct Mapping	Number of blocks	128
Block Replacement Policy	LRU	Cache block size (words)	16
Set size (blocks)	1	Cache size (bytes)	8192

---

**Cache Performance**

Memory Access Count	270000	<b>Cache Block Table</b> (block 0 at top) <input type="checkbox"/> = empty <input checked="" type="checkbox"/> = hit <input type="checkbox"/> = miss
Cache Hit Count	251293	
Cache Miss Count	18707	
Cache Hit Rate	93%	

---

**Runtime Log**

☐ Enabled

---

**Tool Control**

Disconnect from MIPS

Reset

Close

A imagem acima testa o segundo algoritmo, utilizando uma cache com blocos do tamanho de 16 *words*. Sua taxa de hit ficou em 93%.

Data Cache Simulation Tool, Version 1.2

---

**Simulate and illustrate data cache performance**

---

**Cache Organization**

Placement Policy	Direct Mapping	Number of blocks	128
Block Replacement Policy	LRU	Cache block size (words)	32
Set size (blocks)	1	Cache size (bytes)	16384

---

**Cache Performance**

Memory Access Count	270000	Cache Block Table (block 0 at top) <input type="checkbox"/> = empty <input checked="" type="checkbox"/> = hit <input type="checkbox"/> = miss
Cache Hit Count	257527	
Cache Miss Count	12473	
Cache Hit Rate	95%	

---

**Runtime Log**

☐ Enabled

---

**Tool Control**

Disconnect from MIPS

Reset

Close

A imagem acima testa o segundo algoritmo, utilizando uma cache com blocos do tamanho de 32 *words*. Sua taxa de hit ficou em 95%.

Data Cache Simulation Tool, Version 1.2

### Simulate and illustrate data cache performance

#### Cache Organization

Placement Policy	Direct Mapping	Number of blocks	128
Block Replacement Policy	LRU	Cache block size (words)	64
Set size (blocks)	1	Cache size (bytes)	32768

#### Cache Performance

Memory Access Count	270000	Cache Block Table (block 0 at top) <input type="checkbox"/> = empty <input checked="" type="checkbox"/> = hit <input type="checkbox"/> = miss
Cache Hit Count	260250	
Cache Miss Count	9750	
Cache Hit Rate	96%	

#### Runtime Log

☐ Enabled

#### Tool Control

Disconnect from MIPS
Reset
Close

A imagem acima testa o segundo algoritmo, utilizando uma cache com blocos do tamanho de 64 *words*. Sua taxa de hit ficou em 96%.

De acordo com os testes realizados, podemos afirmar que o segundo algoritmo teve um maior aproveitamento da cache, tendo em todos os testes uma taxa de *hit* maior que o primeiro algoritmo. Vemos também, que há uma melhoria em ambos os algoritmos quando o tamanho do bloco da cache é aumentado. Podemos perceber que, mesmo com um tamanho de bloco da cache de 64 *words*, a taxa de *hit* do primeiro algoritmo continuou consideravelmente menor que a do segundo com um bloco de cache de 4 *words*.