

1. Objetivo

O objetivo desta aula é estudar o impacto de memórias cache e o acesso a elas no desempenho de processadores. Nesse contexto, dois estudos de caso são propostos. Ambos abordam o mesmo problema — a soma dos elementos de uma matriz — usando estratégias diferentes para a leitura dos elementos de tal matriz (a primeira envolve o acesso à matriz ao longo de suas linhas enquanto a segunda envolve o acesso à matriz ao longo de suas colunas). **É importante ressaltar que, para a linguagem adotada (C), os elementos de uma mesma linha estão armazenados de forma contígua na memória.**

2. Simulador de cache de dados

O MARS 4.5 possui uma ferramenta para realizar a simulação de 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.

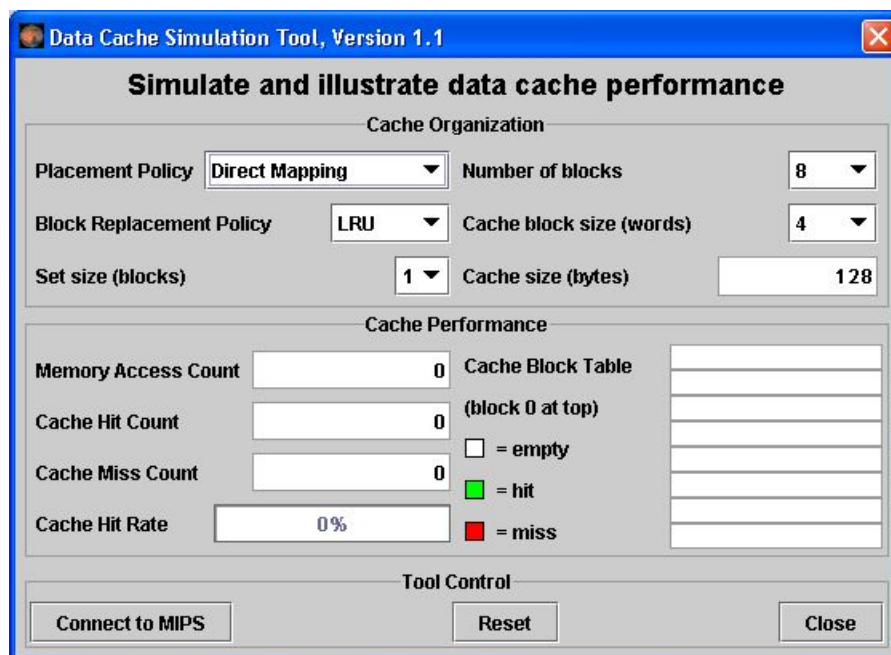


Figura 1: Janela principal do simulador de memória cache de dados.

3. Passos para a análise de desempenho

Para os exercícios propostos neste roteiro, dado um arquivo contendo o programa a ser montado (referenciado em cada exercício), você deverá seguir os seguintes passos para realização de análises de desempenho das caches:

1. Abra o arquivo e preencha as informações faltantes conforme os valores disponíveis 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 memória cache, ambos os simuladores devem ser reinicializados.

4. Exercícios

Nos Exercícios 1 e 2, configure a organização da cache para: **mapeamento direto, LRU e número de blocos igual a 16. O tamanho do bloco é dado no enunciado de cada questão.** Para os Exercícios 3 e 4, mantenha a cache com política de substituição de blocos LRU.

Exercício 1: Análise de desempenho da computação por linha

a) Baixe o arquivo *row-major.asm* do Moodle. Seu código é correspondente 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];
    }
}
```

b) Siga os Passos de 1 a 7 descritos na Seção 3.

c) Responda às Questões 1.1 e 1.2 do relatório correspondentes ao Exercício 1.

Exercício 2: Análise de desempenho da computação por coluna

a) Baixe o arquivo *column-major.asm* do Moodle. Seu código é correspondente 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];
    }
}
```

b) Siga os Passos de 1 a 7 descritos na Seção 3.

c) Responda às Questões 2.1 a 2.4 do relatório correspondentes ao Exercício 2.

Exercício 3: Otimização da configuração da cache para o computação por linha

Para este exercício, use o código *row-major.asm* (o mesmo usado para o Exercício 1).

a) Faça experimentos similares aos já realizados para encontrar a configuração ótima para o programa, ou seja, a **configuração que minimize a taxa de faltas (focando no percentual dado pelo simulador e ignorando o número exato de faltas)** do programa, dada as restrições de que a **cache deve ter pelo menos 2 blocos e no máximo 4096 bytes de dados**. No caso de duas ou mais configurações resultarem na menor taxa de faltas, deve ser dado preferência para a menor configuração (menos bytes de dados), seguida da configuração com menor associatividade. **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).**

b) Responda às Questões 3.1 a 3.3 do relatório correspondentes ao Exercício 3.

Exercício 4: Otimização da configuração da cache para o computação por coluna

Para este exercício, use o código *column-major.asm* (o mesmo usado para o Exercício 2).

a) Faça experimentos similares aos já realizados para encontrar a configuração ótima para o programa, ou seja, a **configuração que minimize a taxa de faltas (focando no percentual dado pelo simulador e ignorando o número exato de faltas)** do programa, dada as restrições de que a **cache deve ter no máximo 64 kbytes** e outras **listadas na Questão 4.1**. No caso de duas ou mais configurações resultarem na menor taxa de faltas, deve ser dado preferência para a menor configuração (menos bytes de dados). **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).**

b) Responda à Questão 4.1 do relatório correspondente ao Exercício 4.