



**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
INE-DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
INE5411 - ORGANIZAÇÃO DE COMPUTADORES I**

**PEDRO TAGLIALENHA (22203674)
VITOR PRAXEDES CALEGARI (22200379)**

RELATÓRIO LABORATÓRIO 7

**FLORIANÓPOLIS
2023**

1 INTRODUÇÃO.....	3
2 RESULTADOS PRÁTICOS.....	4
2.1 DESEMPENHO DA CACHE NO CÓDIGO LINHA A LINHA.....	4
2.2 DESEMPENHO DA CACHE NO CÓDIGO COLUNA A COLUNA.....	6
3 DISCUSSÃO.....	8
3.1 OBSERVAÇÃO SOBRE ACESSOS A MEMÓRIA.....	8
3.2 CONCLUSÕES SOBRE OS CÓDIGOS MODIFICADOS.....	10
4 CONCLUSÃO.....	13

1 INTRODUÇÃO

A memória cache desempenha um papel crucial na otimização do desempenho dos sistemas computacionais, sendo uma parte essencial da hierarquia de memória. Ela atua como um intermediário entre a mais ampla, porém mais lenta, memória principal e o processador, sua função principal é acelerar o acesso à memória, armazenando temporariamente dados frequentemente utilizados. A eficiência da memória cache é de suma importância para garantir a minimização do tempo de acesso à memória e, consequentemente, o aprimoramento geral do desempenho do sistema.

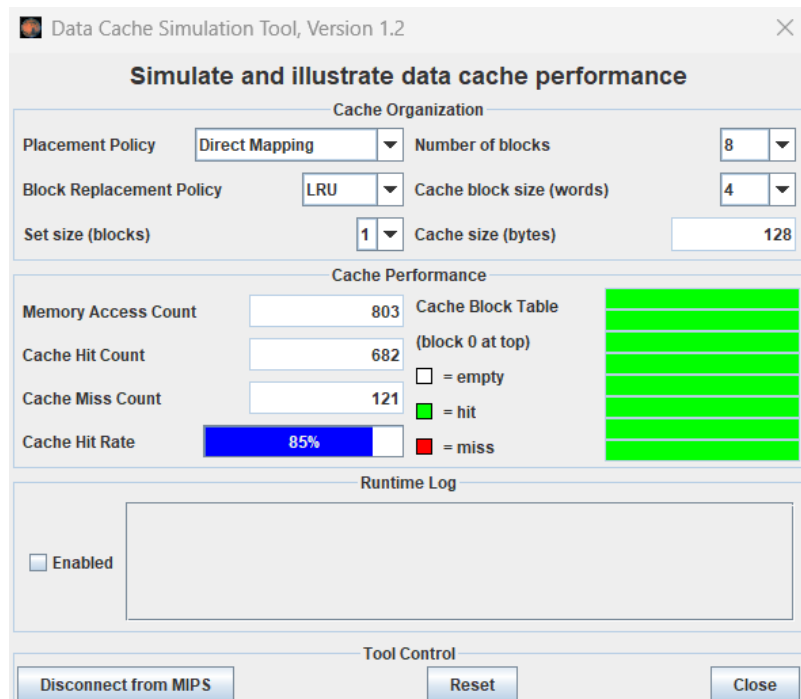
No laboratório 5, foram desenvolvidos dois códigos em assembly, um responsável por preencher uma matriz linha por linha e outro coluna por coluna. A escolha desses padrões visa analisar como o acesso sequencial à memória pode impactar o desempenho da cache. No laboratório 7, utilizaremos a ferramenta Data Cache Simulator do Mars para comparar a eficiência da cache ao executar ambos os códigos.

Este relatório apresentará uma análise dos resultados obtidos, destacando as diferenças de desempenho entre os códigos e também será realizada uma discussão sobre os valores inicialmente esperados versus os encontrados experimentalmente. O entendimento desses aspectos é fundamental para otimizar o desenvolvimento de códigos assembly e maximizar o aproveitamento dos recursos computacionais disponíveis.

2 RESULTADOS PRÁTICOS

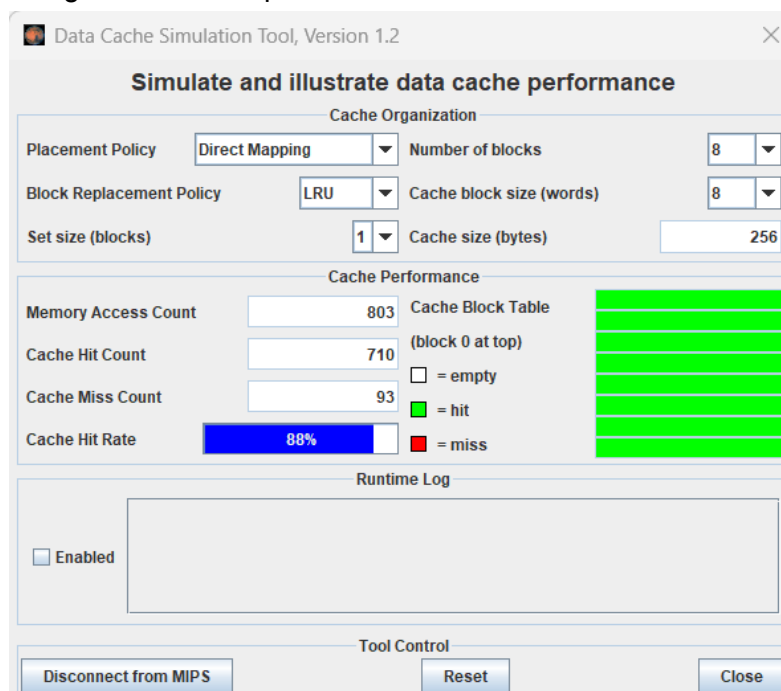
2.1 DESEMPENHO DA CACHE NO CÓDIGO LINHA A LINHA

Imagem 1 - Desempenho da cache com 8 blocos e 4 words



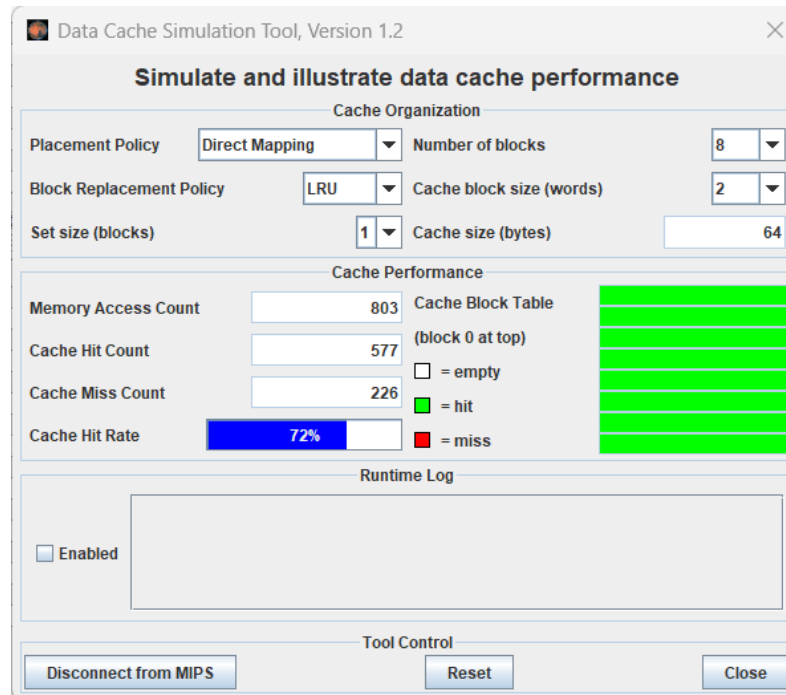
Fonte: Elaborado por autores(2023)

Imagem 2 - Desempenho da cache com 8 blocos e 8 words



Fonte: Elaborado por autores(2023)

Imagem 3 - Desempenho da cache com 8 blocos e 2 words

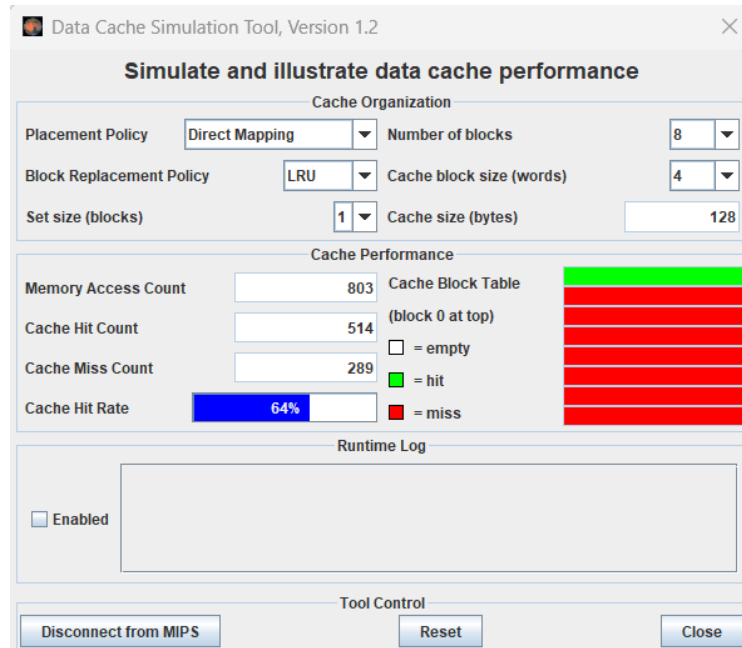


Fonte: Elaborado por autores(2023)

No que diz respeito ao desempenho da cache no código que preenche a matriz linha por linha, os resultados dos testes utilizando diferentes configurações foram notavelmente consistentes. Com 8 blocos contendo 4 palavras cada, registramos um índice de acerto (hit rate) de 85%, enquanto a utilização de 8 blocos com 8 palavras resultou em um valor ligeiramente superior de 88%. Surpreendentemente, a configuração de 8 blocos e 2 palavras apresentou um hit rate de 72%.

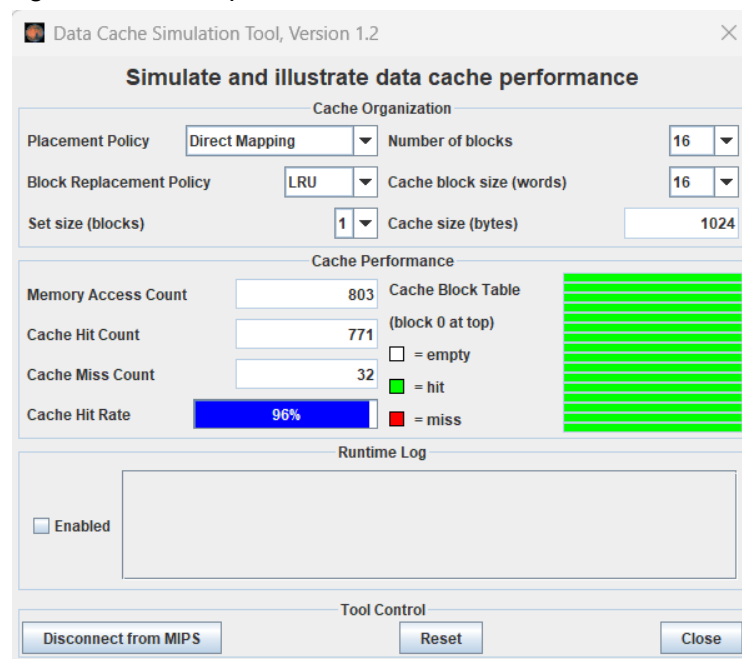
2.2 DESEMPENHO DA CACHE NO CÓDIGO COLUNA A COLUNA

Imagem 4 - Desempenho da cache com 8 blocos e 4 words



Fonte: Elaborado por autores(2023)

Imagem 5 - Desempenho da cache com 16 blocos e 16 words



Fonte: Elaborado por autores(2023)

Ao avaliar o desempenho da cache para o código que preenche a matriz coluna por coluna, os resultados apresentaram variações significativas em relação às configurações da cache. Com 8 blocos e 4 words, o hit rate foi registrado em 64%, indicando uma eficiência inferior comparada ao código linha por linha. No entanto, ao aumentar para 16 blocos e 16 words, o hit rate alcançou o valor de 96%. Essa discrepância sugere uma sensibilidade do código coluna por coluna às configurações específicas da cache, destacando a importância da escolha adequada desses parâmetros para otimizar o desempenho do código.

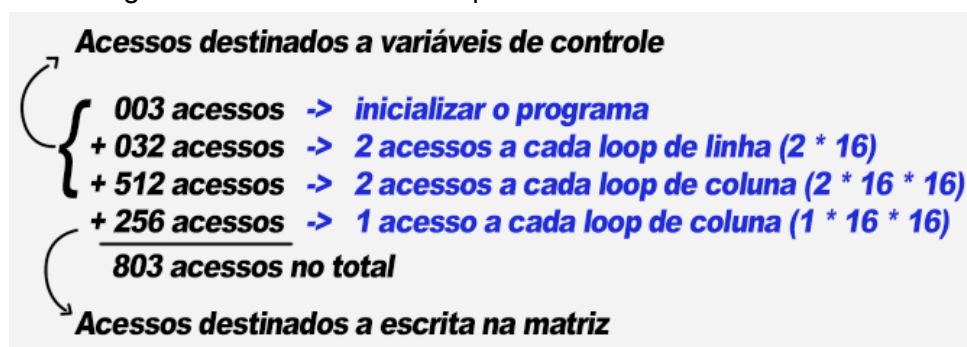
3 DISCUSSÃO

A seção a seguir tem como proposta fazer a análise dos resultados práticos demonstrados na seção anterior, visando entender como o conteúdo apresentado em sala se relaciona aos dados encontrados durante o desenvolvimento deste laboratório.

3.1 OBSERVAÇÃO SOBRE ACESSOS A MEMÓRIA

Os algoritmos concebidos no laboratório 5 foram projetados com o propósito de reproduzir fielmente o código de alto nível apresentado naquela instância. Nesse contexto, o código foi estruturado para acessar as variáveis de controle diretamente da memória. Essa abordagem teve repercussões na taxa de acertos da Cache, uma vez que esta não se limitava apenas a acessos destinados à escrita na matriz.

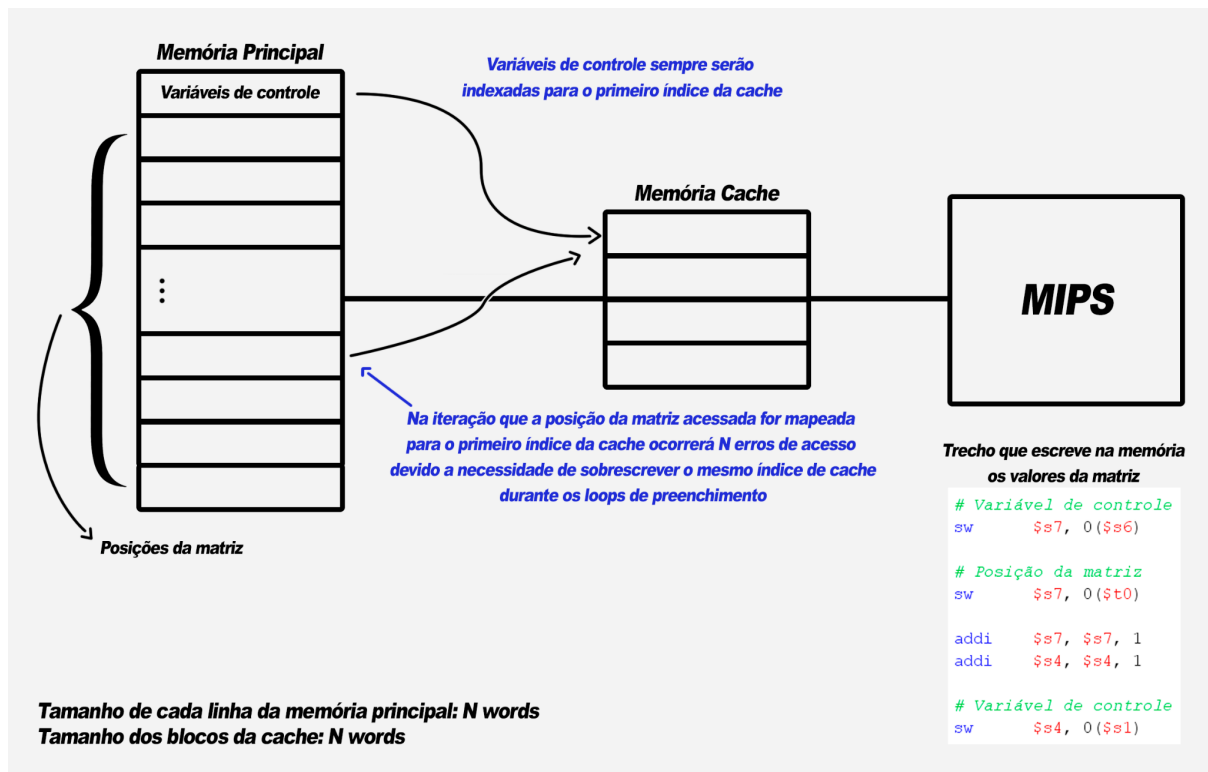
Imagem 6 - Detalhamento da quantidade de acessos a memória



Fonte: Elaborado por autores(2023)

Ao empregar variáveis em memória para o controle dos laços, observou-se um aumento considerável no número de acessos à memória que não eram impactados pelas diferentes lógicas principais entre os programas. Essa circunstância complicou a tarefa de estabelecer previsões e realizar análises comparativas entre os dois códigos, uma vez que o impacto dos distintos métodos de preenchimento da matriz afetava apenas cerca de um terço do total de acessos à memória cache. Além disso, a decisão de utilizar a memória para armazenar as variáveis e atualizá-las no decorrer dos loops resultou em comportamentos inicialmente inesperados, contribuindo para a ampliação da quantidade de erros de acesso à memória cache, conforme exemplificado a seguir:

Imagem 7 - Ilustração sobre funcionamento do código

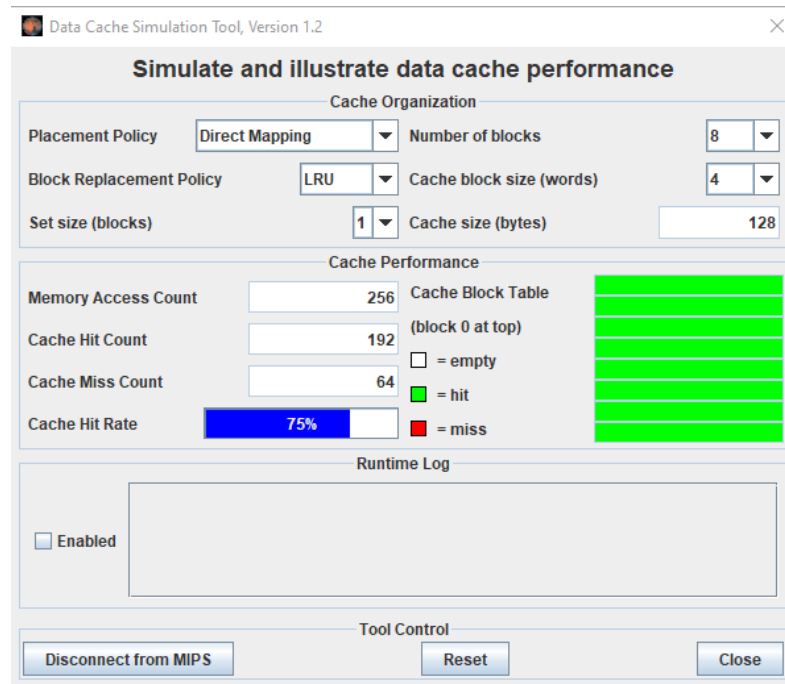


Fonte: Elaborado por autores(2023)

3.2 CONCLUSÕES SOBRE OS CÓDIGOS MODIFICADOS

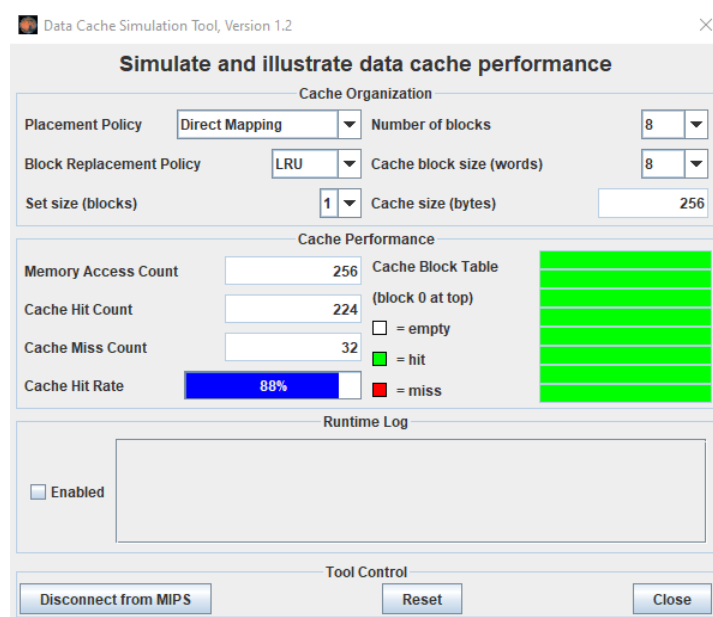
Com o objetivo de atingir os valores teóricos inicialmente estabelecidos, foi desenvolvido um segundo código como uma adaptação dos originais. A principal alteração reside na exclusão das variáveis de controle de memória, assegurando que o acesso à cache seja restrito apenas à escrita de elementos na matriz.

Imagem 8 - Desempenho da cache com 8 blocos e 4 words, código modificado 1



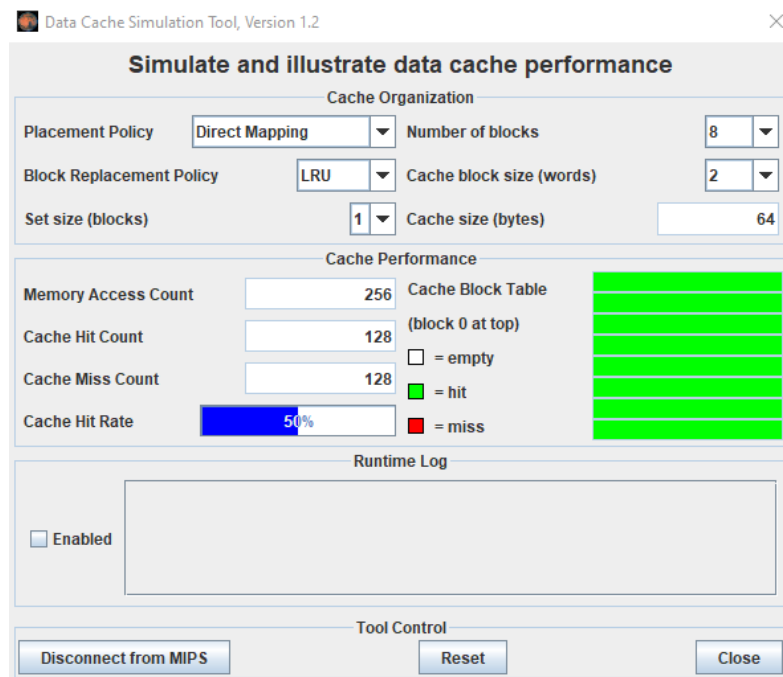
Fonte: Elaborado por autores(2023)

Imagem 9 - Desempenho da cache com 8 blocos e 8 words, código modificado 1



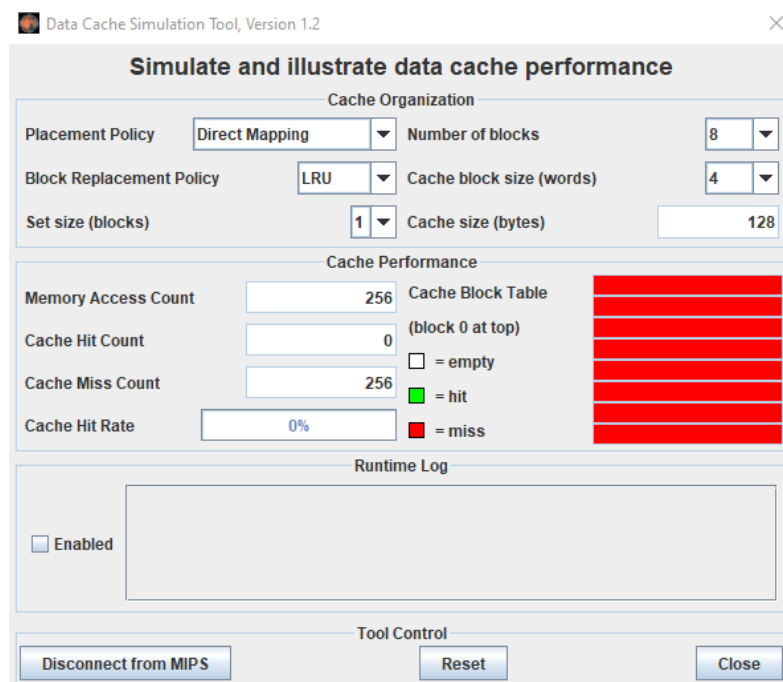
Fonte: Elaborado por autores(2023)

Imagem 10 - Desempenho da cache com 8 blocos e 2 words, código modificado 1



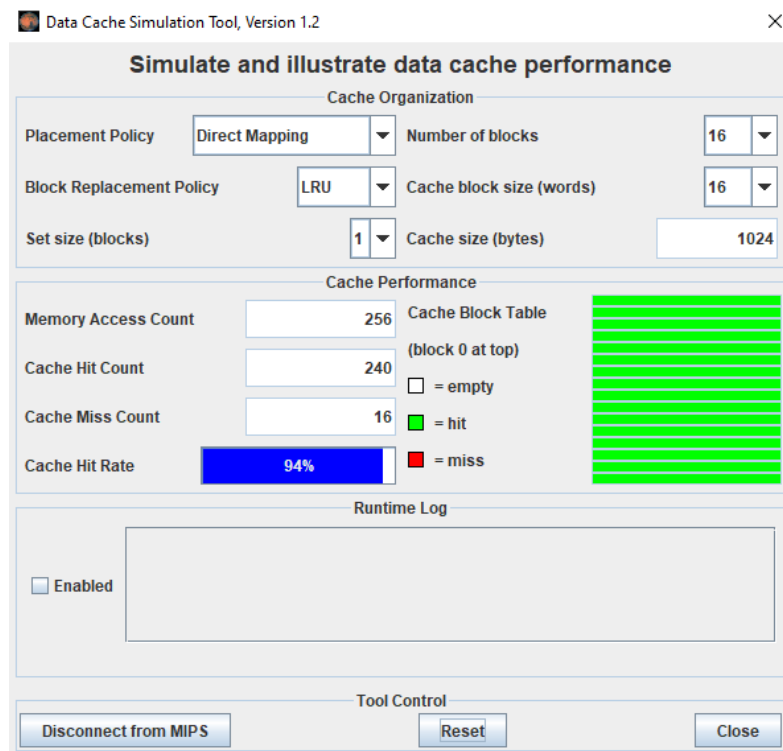
Fonte: Elaborado por autores(2023)

Imagem 11 - Desempenho da cache com 8 blocos e 4 words, código modificado 2



Fonte: Elaborado por autores(2023)

Imagem 12 - Desempenho da cache com 16 blocos e 16 words, código modificado 2



Fonte: Elaborado por autores(2023)

Com os resultados dos códigos modificados, a teoria discutida em sala de aula foi observada com êxito. Conforme previsto, a cache com número de blocos igual a 8 e tamanhos variados entre 4, 8 e 2 apresentou uma taxa de erro de apenas $1/n$, onde "n" é o tamanho do bloco da cache. Isso ocorre devido ao preenchimento "linha a linha" do primeiro código, resultando em falhas ao acessar o primeiro elemento de cada novo bloco, mas sucesso nos outros $n-1$ elementos.

Em adição, evitar a escrita das variáveis de controle da memória tornou ainda mais evidente a falta de eficiência em preencher a matriz pelo método "coluna a coluna". Esse método faz com que cada bloco que o processador acessa na memória cache tenha apenas um único elemento útil, o primeiro, o que faz com que os outros $n-1$ elementos não sejam aproveitados resultando assim em uma taxa de erro de 100% no primeiro caso de teste do código dois.

Por fim, no último caso de teste, o tamanho da memória cache era exatamente o tamanho da matriz, logo só haveriam erros de acesso até a cache ser preenchida por completo, o que era condizente com a explicação teórica estudada em classe anteriormente.

4 CONCLUSÃO

Após a realização dos testes práticos e a análise dos resultados obtidos, observou-se, de acordo com as expectativas, que o desempenho da cache para o código de preenchimento linha a linha é superior ao de coluna a coluna. Tal disparidade pode ser atribuída à natureza sequencial do acesso às posições do vetor pelo código linha a linha, o que permite uma utilização mais eficiente da cache. Por outro lado, o código coluna a coluna, ao acessar as posições em incrementos de 16, não consegue explorar plenamente o potencial da cache, resultando em um desempenho inferior.

Surpreendentemente, os resultados obtidos pela análise experimental não coincidiram com os valores teóricos esperados inicialmente pela equipe. Essa discrepância motivou uma investigação mais aprofundada sobre o funcionamento da memória cache, visando explicar a eficiência observada. A análise mais detalhada revelou aspectos no comportamento da cache que não foram completamente contemplados nas previsões iniciais.

Apesar dos desafios encontrados, o laboratório foi desenvolvido com sucesso, e todos os exercícios propostos para o lab 7 foram completados. Essa experiência não apenas fortaleceu nosso entendimento prático da organização de computadores, mas também ressaltou a importância da análise crítica e da adaptação a resultados inesperados durante o desenvolvimento de projetos relacionados à arquitetura de sistemas computacionais.