

Nome: Richard da Cruz Lopes

RA: 11122015

GitHub: [iRitiLopes \(Richard Lopes\) \(github.com\)](https://github.com/iRitiLopes)

Repositório:

<https://github.com/ufabc-bcc/2021-Arquitetura-Minips-Fase3-iRitiLopes>

Link vídeo: <https://youtu.be/XVwSI-Y3SII>

Projeto Minips - EP3 - Arquitetura de Computadores - UFABC 2021

Bem, o que as fases 1 e 2 foram esta fase 3 não foi. A facilidade de implementação das instruções ainda foi tranquila e a parte mais difícil com certeza deste projeto começou, a implementação das Caches. Talvez essa dificuldade tenha sido na maneira modelada inicialmente do comportamento da memória no emulador. Até que me deparei com questões de performance, principalmente pela maneira em que resolvia a decodificação e a maneira em que as palavras eram armazenadas, como strings, os métodos de transformação dessas strings estavam sendo bastante custosas ao desempenho, foi verificado que ao executar o código de exemplo 18.naive_dgemm estava demorando muito além do que deveria, cerca de 10 minutos (péssimo) então resolvi paralisar o desenvolvimento da Cache para resolver os problemas de performance.

Para resolver o problema de performance foi necessário refatorar grande parte do projeto, remover todo e qualquer tipo de tratamento de strings para número, após algum tempo refatorando com auxílio de uma ferramenta chamada cProfile, foi-se identificando em quais pontos acabavam sendo críticos à performance. Após a refatoração a performance melhorou significativamente caindo para próximo dos 4 minutos para executar o 18.naive_dgemm, ainda não perfeito porém uma grande melhoria.

Após resolver a refatoração de strings matches para o uso de bitmasks e números, foi possível prosseguir, feita a implementação do modo trace e do modo debug. Estes modos por fim acabaram sendo resolvidos utilizando a própria ferramenta de logging da linguagem, alternando o log level dependendo do modo de execução, e por fim chegamos a Cache.

Ao deparar com o problema das Caches, havia uma dificuldade inicial devido a maneira com que a memória havia sido implementada, após um bom tempo analisando como poderia ser implementada, por fim iniciei. O código implementado não é dos melhores que este projeto vivenciou, mas passou a funcionar para os exemplos básicos até que deparado com o SURPRESINHA, motivo até do qual perguntei no canal do *Discord* qual o bug que o Surpresinha pretendia evidenciar, após isso foi corrigido o problemas e pude continuar. Com isso me vi com pouco

tempo para finalizar a implementação dos outros modos de configuração das Caches, portanto o modo que funciona é o 2 (unificado, randômico), a maneira em que foi implementado facilita a modificação dessa cache unificada em seu tamanho, tamanho de linha e mapeamento de n vias(para casos de exemplos mais simples funciona). Gostaria muito de ter conseguido implementar totalmente as Caches, mas me vi com pouco tempo para conseguir a partir do momento em que pude iniciar o desenvolvimento das mesmas.

Por fim, foi muito divertido esse projeto, pretendo futuramente finalizá-lo mesmo após a entrega. Tive que fazer uso de diversos conhecimentos adquiridos ao longo da graduação até aqui. Vejo que essa dificuldade que existiu na implementação das Caches poderia ter sido minimizado se desde o início tivesse continuado implementando os casos de teste (teria ajudado a manter o código conciso e limpo ao longo das novas implementações), fica de aprendizado aos próximos projetos de escopo semelhante (a nível de detalhamento e tamanho). Muito obrigado.