Aluno: Laion Luis Ferreira GRR20158911

Professor: Marco Antonio Zanata Alves

Relatório do trabalho 2 da disciplina de Organização e Arquitetura de Computadores.

Para esse trabalho deveríamos implementar uma memória cache de 64 bytes, 4 vias e tamanho da linha de 16 bytes. Fizemos as três partes do trabalho juntos, porem cada um mais focado em uma parte. Eu fiquei com a cache.

Tivemos muita dificuldade em entender a especificação, achamos um pouco confusa. Levamos muito tempo assimilar o que deveria realmente ser feito. Entramos em contato com o Ricardo para tirar essas dúvidas, mas acabou que as vezes ficávamos mais confusos. Pois ele dizia uma coisa, e na especificação dizia outra, por exemplo na parte em que as requisições são mandadas para a cache, que no pdf do trabalho estava dizendo que as requisições tinham que ser com 32 bits, mas o Ricardo mandou um email dizendo que tinha que ser 8 bits, e então fizemos com 8 bits.

Como a nossa cache tem tamanho total de 64 bytes, e o tamanho da linha é de 16 bytes, então, como Ricardo tirou nossa dúvida, que o tamanho total e o tamanho da linha era pra descobrir quantas linhas iria ter a cache, logo na nossa cache seria 64 divididos por 16 igual a 4 linhas. E como a associatividade é 4-via, que são 4 linhas por conjunto, então como só tem 4 linhas, a cache tem somente um conjunto. Usei 4 comparadores, pegando a tag de cada linha e comparando com a tag do endereço dado pelo banco de endereços. Como são 8 bits do endereço, e que como são 16 bytes a linha, então o offset tem 4 bits. E como nossa cache tem só um conjunto de 4 linhas, fica igual uma cache de associatividade total, logo vai ter 0 bits de index, logo 4 bits de tag. Esses 4 bits de tag é o que é comparado com o tag que está na linha da cache, se for igual é hit. Como no início da execução todas as linhas estão invalidas, então vai buscar o dado na memória principal e voltar para a cache, para escolher qual linha vai ser escolhida fizemos uma tabela verdade de 4 entradas e 4 saídas e usamos para escolher pré-definido qual linha os dados, tag, valit, vão ir, assim sempre escolhe algum que tiver invalido. Caso todas estiverem validos, usamos o algoritmo de substituição random, que era o mais fácil, assim escolher uma das linhas e faz a substituição.

No circuito, terá tag_in que é usado quando acontece miss, que vem a linha da memória principal, juntamente com offset, em 4 clocks, para preencher toda a linha que ta

na memoria principal na cache. Fiquei confuso com a especificação do trabalho e com as respostas do Ricardo, então como não deu tempo, deixei que os dados voltem para o banco de endereços com 32 bits.