

CEFET-MG - Campus II
Departamento de Engenharia de Computação
Laboratório de Arquitetura e Organização de Computadores II

Prática I

Parte III

Alunos: Antônio Augusto Diniz Sousa
Rodrigo Dias Moreira
Professor orientador: Poliana Corrêa

24 de Agosto
2018

CEFET-MG - Campus II
Departamento de Engenharia de Computação
Laboratório de Arquitetura e Organização de Computadores II

Prática I

Parte III

Parte III do Relatório da prática I apresentado à Disciplina de Arquitetura e Organização de Computadores II do Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para conclusão da disciplina.

Alunos: Antônio Augusto Diniz Sousa
Rodrigo Dias Moreira

Professor orientador: Poliana Corrêa

24 de Agosto
2018

Conteúdo

1	Apresentação	1
2	Descrição de atividades	2
2.1	Leitura de dados	3
2.2	Escrita de dados	6
2.3	Junção e Write back	7
2.4	Bloco completo	8
3	Análise dos Resultados	9
4	Conclusão	10

1 Apresentação

A tarefa da terceira parte do trabalho consistia em implementar uma cache associativa por conjunto de 2 vias, com inicialização via arquivo MIF e realização da leitura e escrita utilizando o display de 7-segmentos.

Quanto a implementação, tem-se diversas possibilidades, desde implementar uma lógica sequencial, quanto extrair ao máximo o hardware a fim de montá-lo utilizando apenas operações lógicas.

A dupla optou por implementar uma lógica composta somente por blocos lógicos, a fim de entender melhor todo o funcionamento do hardware, tarefa que na maioria das vezes fica por conta do software utilizado (Quartus) ao programar utilizando funções com nível superior, como por exemplo if, else, etc.

Como o enunciado deixava claro que era necessário inicializar a memória cache com um arquivo do tipo MIF, tornou-se inválido a utilização de registradores, o que facilitaria consideravelmente o trabalho, pois não tínhamos conhecimento de como extrair os dados do arquivo MIF para os registradores. Para suprir esse requisito, foi utilizado um bloco de memória RAM, inicializada pelo Wizard do Quartus, diretamente relacionada com um arquivo do tipo MIF.

2 Descrição de atividades

A fim de minimizar erros, foi montado todo o esquema de blocos, em nível de abstração de portas lógicas, analisando todo o caminho de dados, desde a escolha do dado a ser lido, até a aplicação das políticas de substituição na escrita.

Tendo em vista a abordagem escolhida, é necessário ter em mente que todos os processos estão acontecendo simultaneamente, e não em ordem cronológica ou contexto semelhante. Para facilitar o entendimento do caminho de dados, separamos em duas funções principais, sendo elas a leitura e a escrita de dados.

Antes da implementação das instruções, deve-se estabelecer alguns parâmetros para padronização da arquitetura e garantir o desenvolvimento correto dos blocos e o devido funcionamento de todo o sistema.

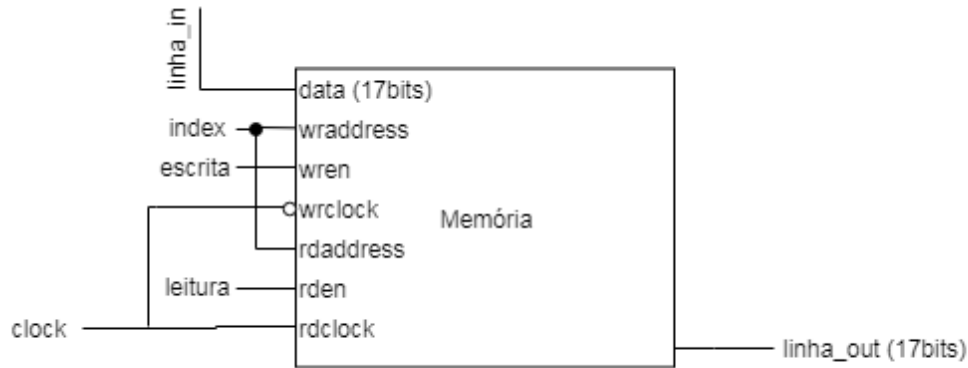
Diante disso, fez-se as seguintes decisões de projeto:

- Memória com 8 palavras, indexadas por 3 bits;
- Index de 3 bits;
- Dado de 4 bits;
- Tag de 2 bits;
- Um bit de LRU por linha;
- Um bit de validade por dado, dois por linha;
- Um bit de dirty por dado, dois por linha;
- Tamanho total da linha de 17 bits.

Diante disso, temos uma cache que abrange até 16 dados e consegue endereçar dados de 4 bits de uma memória principal de até 32 bits, devido a soma do index com a tag. Esse valor foi escolhido diante a implementação feita nas outras partes do trabalho, na qual implementou-se uma memória principal com 32 palavras.

Seguindo todos os parâmetros citados, utilizou-se uma memória RAM de 2 vias para armazenar os dados da cache. Como já mencionado na introdução, isso se deu por causa da necessidade de iniciar a memória com um arquivo MIF. O bloco desenvolvido pode ser visto na Figure 1.

Figura 1: Memória com inicialização MIF



Cada palavra da memória possui 17 bits, para armazenar todos os dados necessários, conforme parâmetros já mencionados.

Figura 2: Linha da cache e utilização das chaves



2.1 Leitura de dados

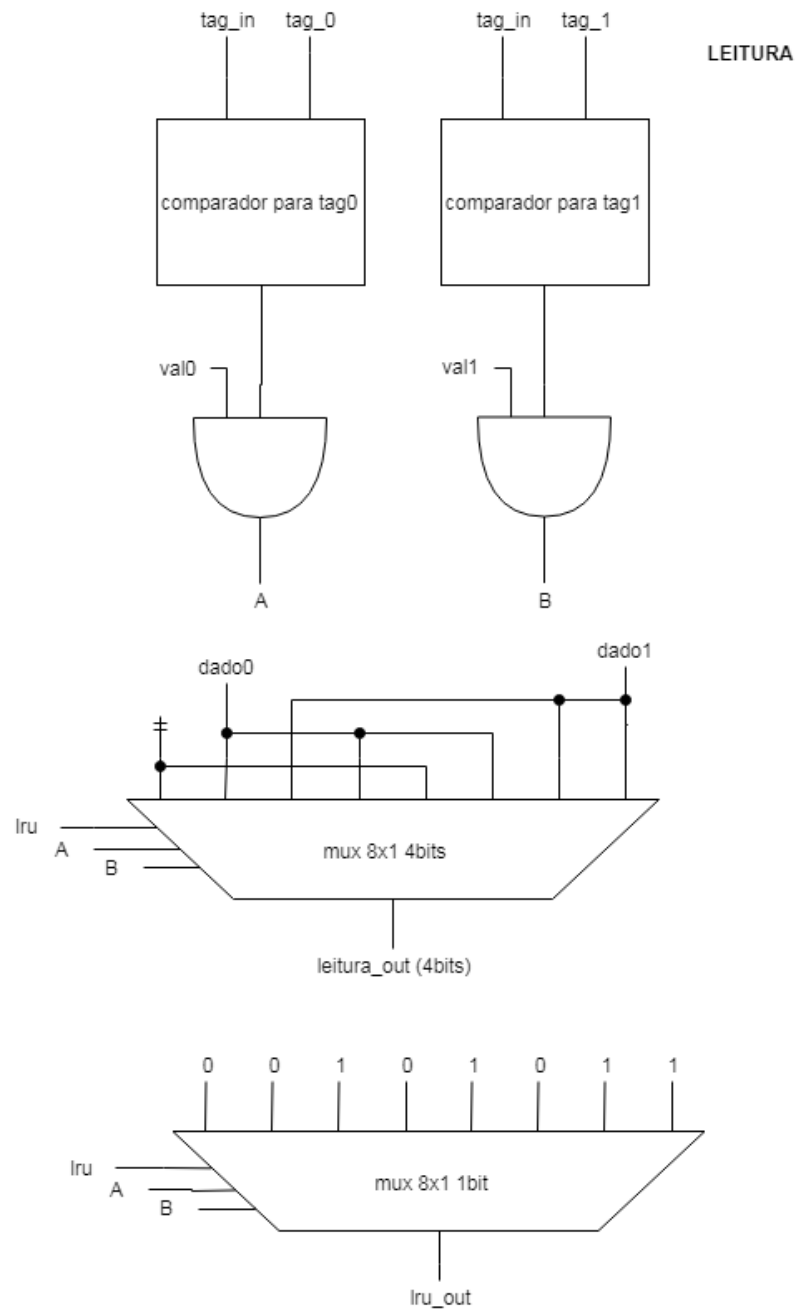
Para a leitura de dados, temos um caminho de dados que pode ser tratado de maneira sequencial, por mais que os dados sejam atualizados praticamente ao mesmo tempo, conforme enumeração abaixo:

1. Buscar linha na memória;
2. Analisar a tag que está sendo buscada;
3. Analisar o bit de validade;
4. Tomar decisão de qual dado será impresso ou miss;
5. Buscar o dado correto a ser mostrado;

6. Atualização do bit correspondente a política LRU;
7. Gravar na memória a linha, com atualização do bit LRU.

Como podemos perceber na listagem feita, temos leitura e escrita na memória para uma única instrução de leitura, sendo ambos no mesmo endereço, correspondente ao index do dado a ser lido. Por esse motivo, implementou-se uma máquina de estados, para separar a parte correspondente a leitura e processamento de dados e a parte de escrita de volta na memória, semelhante ao writeback de um processador MIPS.

Figura 3: Blocos responsáveis pela leitura

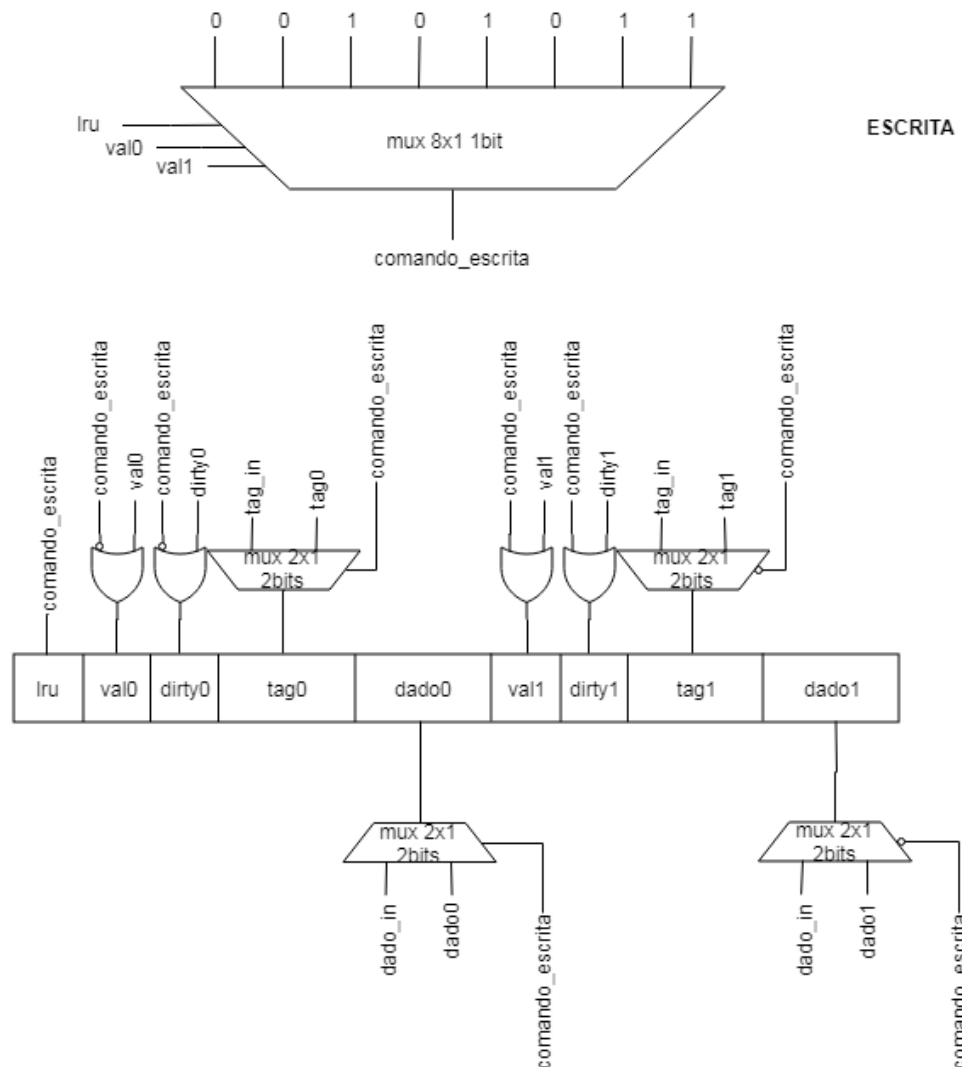


2.2 Escrita de dados

A instrução de escrita de dados não se comporta de maneira tão trivial quanto a de leitura, dado a necessidade de analisar diversos bits e, com base nesses, tomar a decisão quanto a posição em que será feita a escrita do dado.

O primeiro passo a ser feito é o acesso a memória, pois precisamos analisar diversos bits da nossa linha para aplicar a política de substituição correta. Desenvolveu-se, então, um módulo que, através dos bits de validade e de LRU, retorna qual posição da nossa linha deverá ser substituída, que conforme padrão utilizado, 0 corresponde a primeira posição e 1 a segunda. Os blocos responsáveis por esse processo pode ser visto na Figure 4

Figura 4: Blocos responsáveis pela leitura



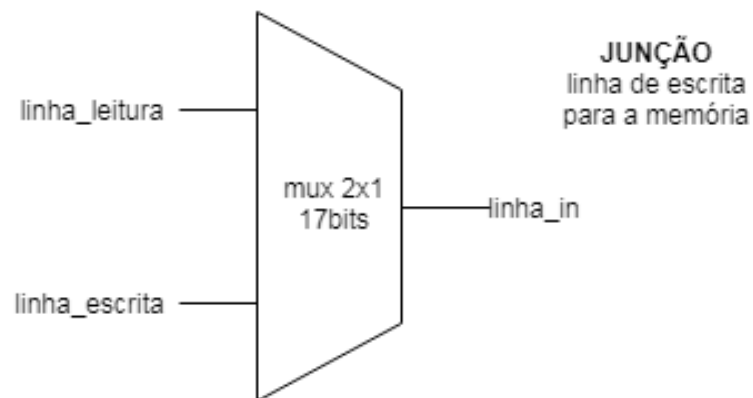
2.3 Junção e Write back

Pode-se observar que tanto os blocos de leitura quanto os de escrita geram no final uma linha, no padrão da linha da memória, que corresponde ao dado que deve ser reescrito na memória. Isso acontece pois a cache funciona de maneira paralela, e irá diferenciar a leitura da escrita apenas na hora de retornar a linha para a memória.

Para direcionar a linha correta para a entrada de escrita da memória,

bastou utilizar um bloco multiplexador 2x1, utilizando como seletor a chave de habilitar escrita, conforme Figure 5.

Figura 5: Junção dos resultados da leitura e escrita



2.4 Bloco completo

O bloco completo pode ser visualizado através desse link.

Como o bloco completo ficou de complexidade elevada diante dos projetos já efetuados, decidimos analisar apenas os blocos básicos, que por sua vez possuem funcionamento trivial, minimizando os erros na hora que juntamos eles em um bloco geral.

3 Análise dos Resultados

A dupla apresentou em sala o circuito parcialmente funcional. Dentre as funcionalidades apresentadas, foi possível efetuar a leitura de um endereço especificado por meio das chaves, observando se havia hit ou miss nas tentativas, bem como o bit LRU atualizado. Constatou-se que a máquina de estados implementada não funcionava corretamente, então optou-se por realizar pulsos de clock para leitura e escrita através de botões separados na placa.

Quanto a escrita

4 Conclusão

A confecção da memória cache e suas políticas e restrições para leitura e escrita trouxeram à dupla um desafio válido. Não apenas na complexidade dos requisitos, frente à projetos em verilog antigos, desenvolvidos pela dupla em disciplinas anteriores, encontrou-se também um novo desafio na abordagem adotada para a descrição do circuito proposto.

A adoção de circuitos combinacionais como método de descrição permitiu as integrantes atingir um novo patamar em um aspecto não tão intensamente experienciado até então na área dos circuitos digitais, o paralelismo. Mesmo diante dos problemas enfrentados na montagem final do circuito, na prática, a dupla pôde constatar erros em seções separadas do circuito, confirmando o funcionamento parcial do circuito. Ao identificarmos as porções as quais residiam os problemas, restava apenas estudar mais profundamente a raiz que os causavam. Entretanto, isso não foi possível dentro da disponibilidade de tempo da dupla.

Assim, considera-se proveitoso e de extrema importância a experiência advinda desta prática diante dos futuros desafios propostos a seguir na disciplina. Foi possível visualizar e identificar caminhos que poderiam facilitar ou dificultar o desenvolvimento dos mesmos, bem como o desenvolvimento de uma maior organização quanto ao processo de desenvolvimento utilizado pelos integrantes, referente à discussão e desenho prévio da divisão em blocos do circuito proposto.