**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA**

**DE MINAS GERAIS**

Arquitetura e Organização de Computadores II

2023.1

Logotipo

Descrição gerada automaticamente

Aluno: Tarcísio Batista Prates – 20193008761

Aluno: Victor [...]

Prática 1

**Objetivo:** Implementação de uma memória RAM, a partir da biblioteca *altsyncram* do Quartus.

Primeiramente, foi realizada a configuração da memória, conforme especificação do manual. Na versão do *Quartus* instalada, a biblioteca *altsyncram* está nomeada como RAM, com diversas opções de quantidade de canais de acesso, assim, seguindo as instruções indicadas no manual, foi selecionada a opção com um único canal. Abaixo segue o print da configuração:

Interface gráfica do usuário

Descrição gerada automaticamente

Imagem 1: Implementação da memória da biblioteca LPM

Conforme mostrado na imagem acima, o módulo conta com quatro entradas e uma saída, sendo as especificações delas:

* Address: 8 bits de tamanho, indica o endereço na memória a ser acessado.
* Data: 8 bits de tamanho, transporta o dado a ser gravado na memória
* Wren: 1 bit de tamanho, sinal de controle, Alto (1) escrita e Baixo (0) leitura.
* Clock: 1 bit, sinal de controle de sincronização para executar o módulo
* Q: 8 bits, transporta a saída da memória resultante da leitura.

Para fazer os acessos à memória, definiu-se através da propriedade “initial” o valor dos registradores ao logo do tempo de execução, de forma a testar a escrita e a leitura na memória. Inicialmente, gravando o valor 2 (dois) na posição 14 (quatorze) e 3 (três) na posição 15 (quinze), essa operação foi executada nos 100 primeiros nano segundos, em seguida, a leitura dessas mesmas posições para verificar a efetivação da operação de gravação. A seguir, o código em *Verilog* para testar o circuito.

Gráfico

Descrição gerada automaticamente com confiança média

Imagem 2: Print do Quartus contendo a descrição do circuito

Para simular a execução no chip indicado no tutorial, no ModelSim, foi adicionado a biblioteca **altera\_mf\_ver** em conjunto com o arquivo principal do projeto. Abaixo, segue a imagem dos sinais resultantes da simulação:

Tela de jogo de vídeo game

Descrição gerada automaticamente

**Escrita Leitura Leitura**

Imagem 3: Waves da simulação do circuito no ModelSim

Cada período do *Clock*, está configurado para durar 100 nanosegundos, logo percebe-se que no primeiro ciclo, são gravados os valores, conforme especificado na descrição do circuito, e nos 100 nanosegundos seguintes, os valores não lidos. É importante observar que a partir da leitura, o resultado na saída é colocado somente na próxima troca de sinal de Clock, isso se dá devido ao delay de leitura.

A seguir, é apresentado o estado final da memória da biblioteca LPM, e conforme o esperado, as posições 14 e 15, possuem gravados 3 e 2, respectivamente.

Interface gráfica do usuário

Descrição gerada automaticamente

Imagem 4: Estado final da memória da biblioteca

**O código-fonte e os arquivos de simulação estão na raiz do projeto.**

Prática 2

**Objetivo:** Implementação de uma memória RAM, a partir da biblioteca *altsyncram* do Quartus com inicialização via arquivo MIF.

Para inicializar a RAM com o arquivo MIF, basta indicar na configuração da RAM, um arquivo para essa inicialização.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

O arquivo MIF foi modificado para carregar a seguinte inicialização na RAM:

Texto

Descrição gerada automaticamente com confiança baixa

Para verificar a correta inicialização da memória, basta inicializar o circuito sem clock, assim não haverá modificações.

Texto

Descrição gerada automaticamente com confiança baixa

Veja que as posições 0 e 1, estão como 15 e 20 respectivamente, tal como indicado no arquivo MIF.

Para verificar o correto funcionamento, projetou-se um contador para preencher todas as posições da ram. Sendo assim, a simulação demonstra seu correto funcionamento.

Interface gráfica do usuário

Descrição gerada automaticamente

Prática 3

**Objetivo:** Implementação de uma memória cache de duas vias, com ligação com uma memória RAM.

A lógica de interação entre as instruções enviadas pelo processador e os componentes da RAM, se dava primariamente pela avaliação dos seguintes parâmetros:

1º - INDEX: Indica qual posição da cache será avaliada

Dessa forma, os parâmetros seguintes serão avaliados para o circuito escolher qual das vias será acessada para responder a instrução do processador.

1º - TAG: Indica qual o valor do bloco buscado, seja para leitura ou escrita

2º - DIRT: Indica se está com um dado que necessita de atualização na RAM

3º - LRU: Indica se o canal foi o último a interagir com o processador

Há também, um outro parâmetro que é o VALID, porém ele serve apenas como controle terciário, não influindo na escolha da posição acessada.

Sendo assim, as condições para entrada de uma instrução para o canal 1, deve responder a tabela:

A tabela abaixo, está na forma reduzida, apresentando apenas o caso em que a TAG não é reconhecida.

|  |  |  |  |
| --- | --- | --- | --- |
| TAG (A) | DIRT (B) | LRU (C) |  |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |

Resolvendo a tabela, verifica—se que: !A( !B!C + B!C), portanto, basta duplicar para o caso em que A (TAG) é verdadeira para completar a tabela.

Assim, para facilitar o desenvolvimento, a primeira parte do código faz a obtenção dos valores desses parâmetros, e os avalia logicamente, reduzindo a sintaxe do código.

Seguindo os mesmos padrões das memorias anteriormente desenvolvidas, as instruções e as memórias possuem as seguintes configurações:

RAM: 8 bits de dado e 256 posições

CACHE: 19 bits de dados e 8 posições

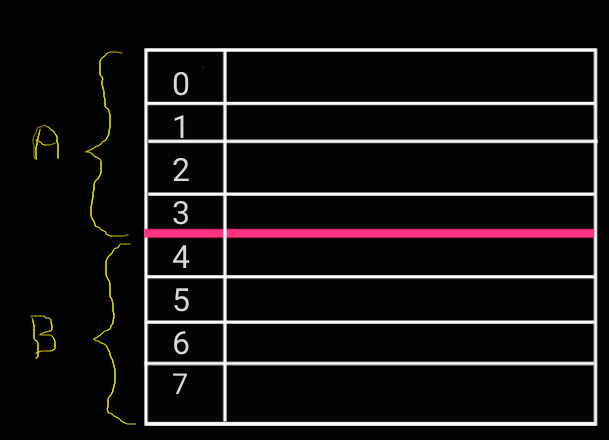
INTRUÇÃO: 19 bits

A organização da cache, foi feita de modo:

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Para simular uma cache de duas vias, projetou-se uma tabela de 8 linhas, em que as instruções vindas do processador enviam um valor de index de 0 a 3, no entanto, na memória cache esse valor é interpretado como sendo INDEX e INDEX\_PLUS, de forma que index plus é o resultado de soma de INDEX + 4, assim é possível realizar ao mesmo tempo o acesso em [0,4], [1,5], [2,6] e [3,7], tal como um cache de duas vias.



Simulação P3

Instrução 1:

Interface gráfica do usuário

Descrição gerada automaticamente

Data\_out = 100 (Conteúdo que veio da RAM)

Hit\_out = 0 (MISS)

wBack\_out = 0 (Write-back)

load\_out = 1 (INVALID)

channel\_out = 1 (Interação ocorreu no canal A)

Instrução 2:

Interface gráfica do usuário

Descrição gerada automaticamente

Data\_out = 101 (Conteúdo que veio da RAM)

Hit\_out = 0 (MISS)

wBack\_out = 0 (Write-back)

load\_out = 1 (INVALID)

channel\_out = 0 (Interação ocorreu no canal B)

Instrução 3:

Interface gráfica do usuário

Descrição gerada automaticamente

Data\_out = 100

Hit\_out = 1

wBack\_out = 0 (Write-back)

load\_out = 0

channel\_out = 1 (Interação ocorreu no canal A)

instrução 4:

Interface gráfica do usuário

Descrição gerada automaticamente

Data\_out = 111 (Conteúdo que veio da RAM)

Hit\_out = 1

wBack\_out = 0 (Write-back)

load\_out = 0

channel\_out = 1 (Interação ocorreu no canal A)

instrução 5:

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Data\_out = 010 (Conteúdo que veio da RAM)

Hit\_out = 1

wBack\_out = 0 (Write-back)

load\_out = 0

channel\_out = 0 (Interação ocorreu no canal B)

instrução 6:

Tabela

Descrição gerada automaticamente com confiança média

Data\_out = 011 (Conteúdo que veio da RAM)

Hit\_out = 0 (MISS)

wBack\_out = 1 (Write-back)

load\_out = 0

channel\_out = 1 (Interação ocorreu no canal A)

instrução 7:

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Data\_out = 001 (Conteúdo que veio da RAM)

Hit\_out = 0 (MISS)

wBack\_out = 0 (Write-back)

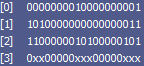
load\_out = 1

channel\_out = 0 (Interação ocorreu no canal B)

CACHE:

INÍCIO

CANAL A CANAL B

 Texto

Descrição gerada automaticamente

FIM

CANAL A CANAL B

Texto

Descrição gerada automaticamente Texto

Descrição gerada automaticamente

RAM:

INÍCIO FIM

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente Tabela

Descrição gerada automaticamente com confiança baixa