Laboratório de AOC II
Profa. Daniela Cristina Cascini
Matheus Santos | Gabriel Schmidt

RELATÓRIO PRÁTICA I

PARTE I

OBJETIVO:
Implementação de uma memória RAM utilizando a biblioteca LPM. A leitura e escrita devem ser realizadas utilizando o display de 7-segmentos.

FORMATO DA ONDA e EXPLICAÇÃO DO COMPORTAMENTO DA ONDA se encontram juntas no teste da PARTE II por praticidade de realização de teste, visto que o fato de a memória já estar ocupada graças ao arquivo .mif gera um ambiente mais propício para realização da simulação.

PARTE II

OBJETIVO:
Inicialização da memória utilizando um arquivo (MIF - *memory initialization file*). A leitura e escrita devem ser realizadas utilizando o display de 7- segmentos.

FORMATO DA ONDA:

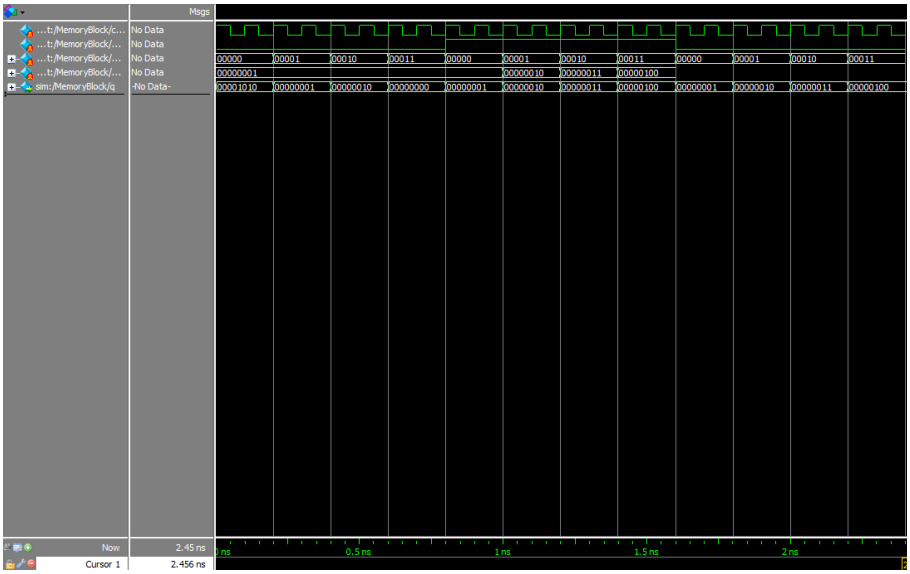

Figura 1: Print consta em anexo caso deseja visualizar melhor

EXPLICAÇÃO DO COMPORTAMENTO DA ONDA:
Execução de um caso de teste geral, no qual primeiramente o MODO DE ESCRITA (wren) está desabilitado e há a leitura das 4 primeiras memórias que constam com os valores ditados pelo .mif 10,1,2,0 respectivamente. Após essa leitura o MODO DE ESCRITA é habilitado, e são escritos os valore 1,2,3,4 nas primeiras posições da memória. Após a escrita é refeita o MODO DE LEITURA torna a ser desabilitado e é então refeita a leitura da 4 primeiras posições, mostrando assim os valores 1,2,3,4.

ARQUIVO .MIF:

```
WIDTH=8;
DEPTH=32;

ADDRESS_RADIX=UNS;
DATA_RADIX=UNS;

CONTENT BEGIN
        0   :   10;
        1   :   1;
        2   :   2;
        [3..31]  :   0;
END;
```

PARTE III

OBJETIVO:
Implementação de uma cache associativa por conjunto de 2 vias, com utilização do MIF e realização da leitura e escrita utilizando o display de 7-segmentos.

PROJETO DE SISTEMA:
- TEORIA:
Em uma cache associativa por conjunto de 2 vias, temos 2 entradas para cada índice da cache, sendo assim os dados selecionados serão baseados no resultado da comparação entre tags das caches paralelas.
- ARQUITETURA:
A partir disso a decisão foi de criar 2 módulos de cache associativa por conjuntos, de 2 conjuntos, sendo cada bloco da cache de 12bits, distribuídos da seguinte maneira:

VALIDADE (1) + LRU (1) + DIRTY (1) + TAG (4) + BLOCO (5)

- IMPLEMENTAÇÃO LEITURA:
FLUXO 1 - Tag correspondente e acesso ao bloco é valido (EXCEÇÃO 1)
- HIT = 1; ACESSADO (LRU) = 0;
FLUXO 2 - Leitura do bloco e saída no circuito
- ACESSADO (LRU) = 1; NÃO ACESSADO (LRU) = 0;
- FIM
EXCEÇÃO 1 – Não existe primeira tag, verifica a próxima (EXCEÇÃO 2)
- HIT = 1; ACESSADO (LRU) = 1;
- FLUXO 2
EXCEÇÃO 2 – Não existe nenhum bloco válido ou existe um bloco válido mas não tem tag correspondente
- ACESSADO = LRU[0];
- Verifica o bit dirty para o caso de ele ser valido, se sim, prossegue, se não FLUXO 2
- Solicitação de escrita da cache na memória
- Selecionado o bloco da cache que deve ser escrito na memória
- CASO ESPECIAL: quando ocorre a leitura de um bloco dirty, é necessário tanto ler quanto escrever na memória
- FLUXO 2

- IMPLEMENTAÇÃO ESCRITA:
FLUXO 1 - Tag correspondente e acesso ao bloco é valido (EXCEÇÃO 1)
- HIT = 1; ACESSADO (LRU) = 0;
FLUXO 2 – Escreve na cache
- DIRTY = 1; ACESSADO (LRU) = 1; NÃO ACESSADO (LRU) = 0;
EXCEÇÃO 1 – Não existe primeira tag, verifica a próxima (EXCEÇÃO 2)
- HIT = 1; ACESSADO (LRU) = 1;

- FLUXO 2

EXCEÇÃO 2 – Não existe nenhum bloco válido ou existe um bloco válido mas não tem tag correspondente

- ACESSADO = LRU[0];
- Verifica o bit dirty para o caso de ele ser valido, se sim, prossegue, se não FLUXO 2
- Solicitação de escrita da cache na memória
- Selecionado o bloco da cache que deve ser escrito na memória
- FLUXO 2

FORMATO DA ONDA:

(Não obtido pois tivemos problemas com o nosso programa, vamos apresentar na próxima aula conforme conversa em sala com a professora Daniela)

ARQUIVO .MIF:

```
WIDTH=5;
DEPTH=32;

ADDRESS_RADIX=UNS;
DATA_RADIX=UNS;

CONTENT BEGIN
        0   :   0;
        1   :   1;
        2   :   2;
        3   :   3;
        4   :   4;
        [5..31]  :   0;
END;
```