



## Centro Federal de Educação Tecnológica de Minas Gerais

Disciplina: Laboratório de Arquitetura e Organização de

Computadores II

### Prática I - Parte 3

Alunos: Ana Julia Rodrigues e Ítalo Augusto

Prof.: Daniela Cristina Cascini Kupsch

#### Introdução:

Por meio dos softwares Quartus e ModelSim, foi implementada uma memória RAM utilizando a biblioteca LPM. Fizemos uso de uma memória de 32 palavras e 8 bits, e criamos uma cache L1 de duas vias (no código: cacheVia0 e cacheVia1). A memória principal que usamos é diretamente mapeada, a atualização da memória ocorre utilizando a política de write-back, onde a escrita de um valor é feita só no bloco da cache, e escrito na RAM depois que o valor é substituído por outro. A memória principal e a cache foram inicializadas previamente com os valores disponibilizados nos casos teste, e a divisão de bits da cache foi feita da seguinte maneira:

[8]: Valid

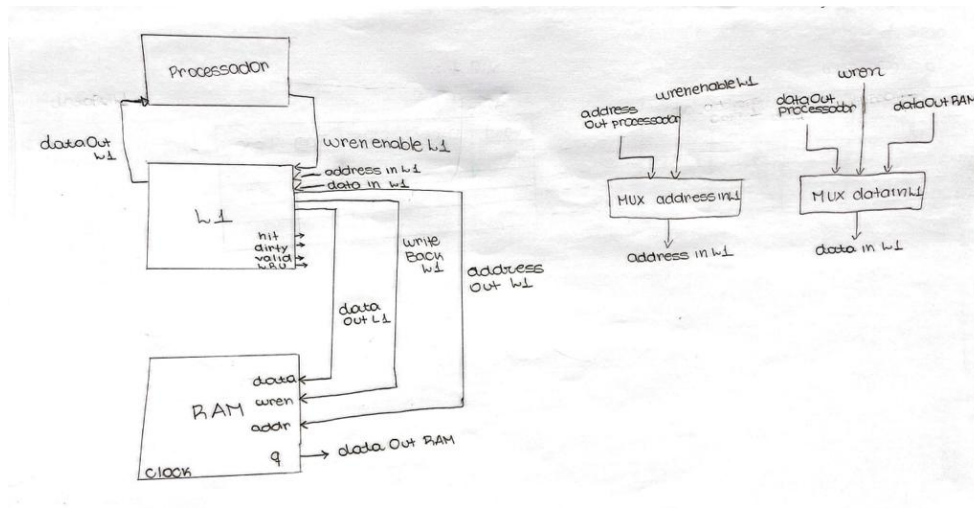
[7]: Dirty

[6]: LRU

[5:3]: TAG

[2:0]: Dados

#### Arquitetura Inicial:



#### Mudanças na arquitetura:

Foi decidido não utilizar os dois mux propostos na implementação (MUX addressInL1 e e MUX dataInL1), com essa mudança o dado de saída é passado diretamente da cache para a memória RAM e vice e versa.

### Código para adicionar na simulação:

```
force -freeze {sim:/p3_main/KEY[0]} 1 0, 0 {50 ps} -r
100
force -freeze sim:/p3_main/clockRAM 1 0, 0 {7 ps} -r 15
add wave -position end    sim:/p3_main/SW
add wave -position end    sim:/p3_main/dataMemOut
add wave -position end    sim:/p3_main/dataCacheOut
add wave -position end    sim:/p3_main/dataOut
add wave -position end    sim:/p3_main/writeBack
add wave -position end    sim:/p3_main/hit
add wave -position end
sim:/p3_main/exibirBlocoVia0Antes
add wave -position end
sim:/p3_main/exibirBlocoVia1Antes
```

### Entradas e saídas esperadas pelos casos teste:

1: 001000000000000000 - Ler - Tag 100 - Indice 00 - MISS  
2: 000000100000000000 - Ler - Tag 000 - Indice 01 - HIT  
3: 1100001000000000101 - Write - Tag 000 - Indice 01 - HIT  
4: 1101001000000000100 - Write - Tag 010 - Indice 01 - MISS  
5: 0000101000000000000 - Ler - Tag 001 - Indice 01 - MISS com WRITE BACK  
6: 1101101000000000001 - Write - Tag 011 - Indice 01 - MISS com WRITE BACK  
7: 0000001000000000000 - Ler - Tag 000 - Indice 01 - MISS  
8: 0001001000000000000 - Ler - Tag 010 - Indice 01 - MISS com WRITE BACK

//8: Valid - 7: Dirty - 6: LRU - [5:3]: TAG - [2:0]: Dados

	VIA	INDICE	INICIO	FIM
1:	0	00	000100011	101100111
2:	0	01	100000011	101000011
3:	0	01	101000011	111000101
4:	1	01	100001111	111010100
5:	0	01	101000101	101001011
6:	1	01	101010100	111011001
7:	0	01	100001011	101000101
8:	1	01	110011001	101010100

## Prints da simulação:

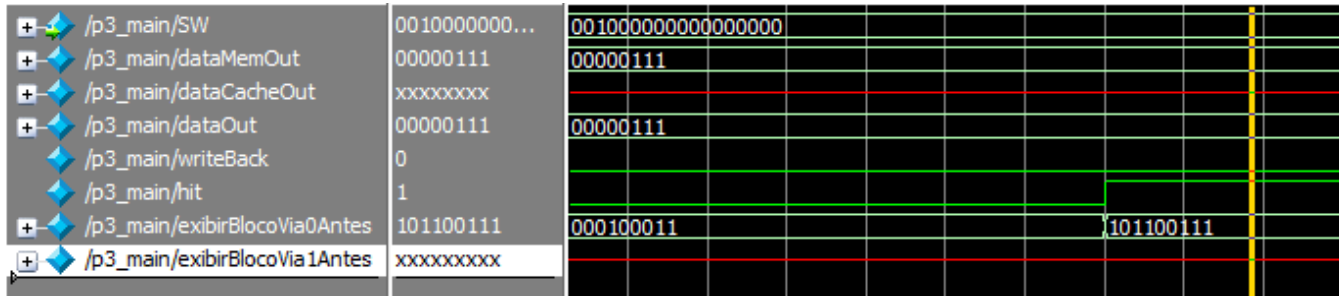


Figura 1 - Primeira instrução (Read miss)

Essa instrução tem como objetivo ler a tag 100 no índice 00 na cache. Verifica-se a tag e a validade e, como a validade está como 0, apresenta miss. Então, é buscado na memória com endereço dos 4 bits menos significativos (0000) e é inserido na via 0, devido ao LRU estar 0. Atualiza o LRU e retorna o dado 111 para o processador.

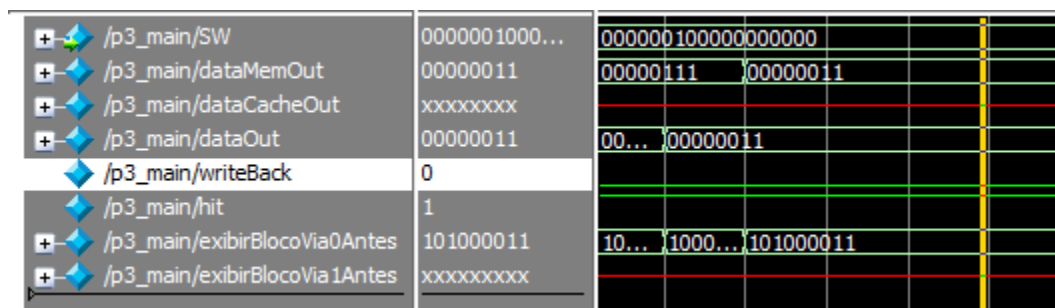


Figura 2 - Segunda instrução (Read hit)

Essa instrução tem como objetivo ler a tag 000 no índice 01 na cache. Verifica-se a tag e o bit de validade, apresentando hit na via 0, atualizando o LRU das vias e retornando o dado 011 para o processador.

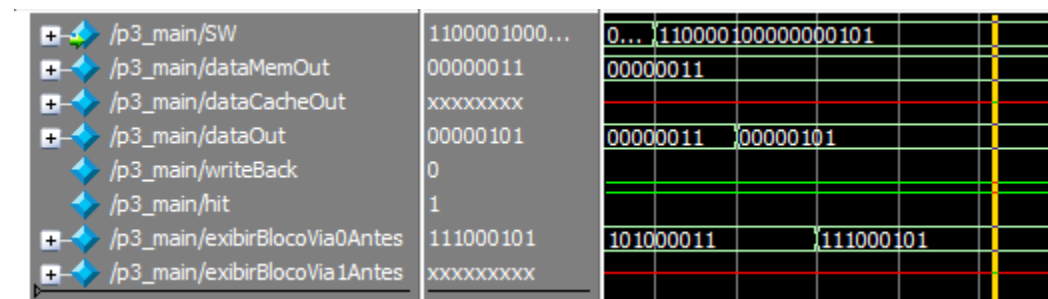


Figura 3 - Terceira instrução (Write hit)



Essa instrução tem como objetivo escrever o dado 001, na tag 011, no índice 01 na cache. Verifica-se a tag e o bit de validade, apresentando miss. Sendo assim, busca o bloco da memória com endereço dos 4 bits menos significativos (1101) e faz a escrita na via 1. O LRU das vias é atualizado, o dado é escrito na cache, o dirty muda para 1 e é feito o write back do bloco 1001 com valor 100.

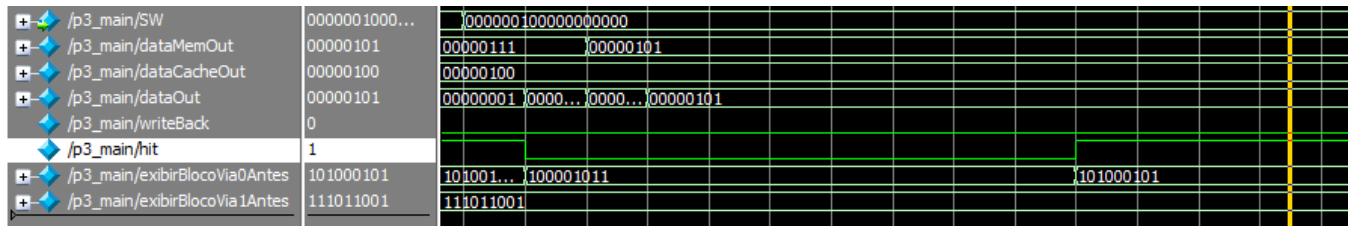


Figura 7 - Sétima instrução (Read miss)

Essa instrução tem como objetivo ler a tag 000 no índice 01 na cache. Verifica-se a tag e a validade e, como a validade está como 0, apresenta miss. Então é buscado na memória com endereço dos 4 bits menos significativos (0001) e é inserido na via 0, devido ao LRU estar 0. Atualiza o LRU e retorna o dado 101 para o processador.

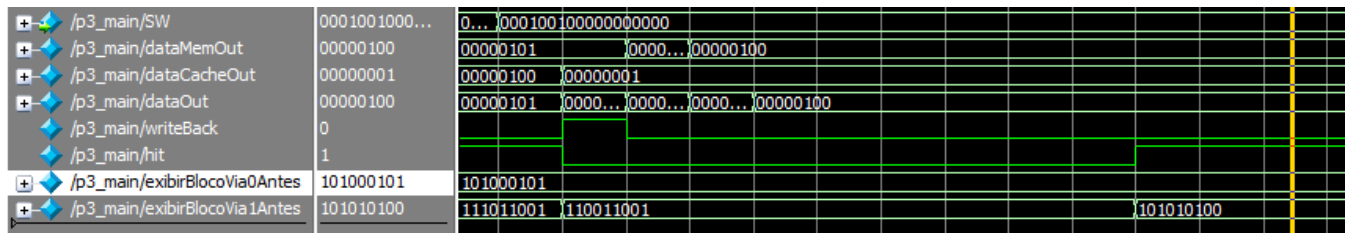


Figura 8 - Oitava instrução (Read miss com write back)

Essa instrução tem como objetivo ler a tag 010 no índice 01 na cache. Verifica-se a tag e o bit de validade, apresentando miss. Sendo assim, busca o bloco da memória com endereço dos 4 bits menos significativos (1001), que tem o valor 100, e faz a escrita na via 1. O LRU das vias é atualizado, o dado é escrito na cache, o dirty muda para 0 e é feito o write back do bloco 1101 com valor 001.