

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CEFET-MG**

**PROF.^a POLIANA CORRÊA
LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES II**

PRÁTICA : PARTE I

**GIOVANNI SUARIANI FERREIRA
THALLES AUGUSTO SOARES VERÇOSA**

**BELO HORIZONTE
2025**

INTRODUÇÃO

A primeira atividade prática teve como objetivo implementar uma memória RAM de 64 posições com largura de 16 bits, utilizando a biblioteca LPM (Library of Parameterized Modules). Para facilitar os testes e validação do funcionamento do sistema, a memória foi inicializada com valores aleatórios nas primeiras posições por meio de um arquivo MIF (Memory Initialization File). A interface de interação e verificação das operações foi realizada com o auxílio de displays de 7 segmentos, que exibem informações como o tipo de operação (leitura ou escrita), o valor envolvido, o endereço acessado e o conteúdo correspondente. Como forma de simplificação, foram utilizados apenas os bits menos significativos para exibição nos displays. A verificação prática do sistema foi feita por meio de duas operações de escrita em posições distintas da memória seguidas pelas respectivas leituras.

1 EXERCÍCIOS

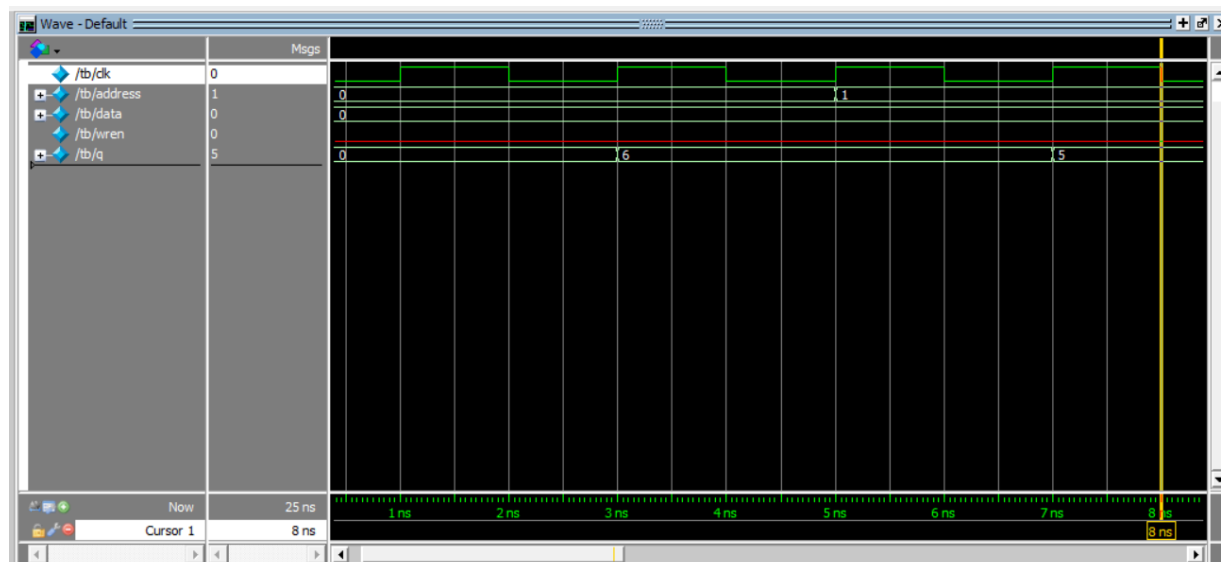
1.1 PRÁTICA 1

Texto.

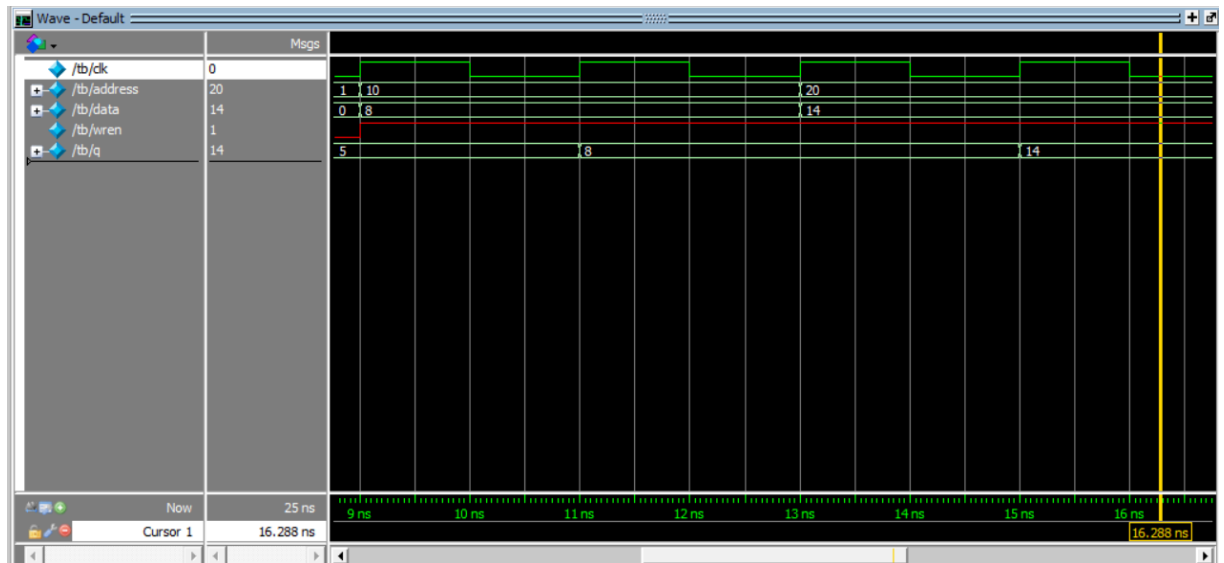
1.1.1 Imagens de tela da execução do programa

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	0	0	0	0	0	0	0	0	-----
8	0	0	0	0	0	0	0	0	-----
16	0	0	0	0	0	0	0	0	-----
24	0	0	0	0	0	0	0	0	-----
32	0	0	0	0	0	0	0	0	-----
40	0	0	0	0	0	0	0	0	-----
48	0	0	0	0	0	0	0	0	-----
56	0	0	0	0	0	0	0	0	-----

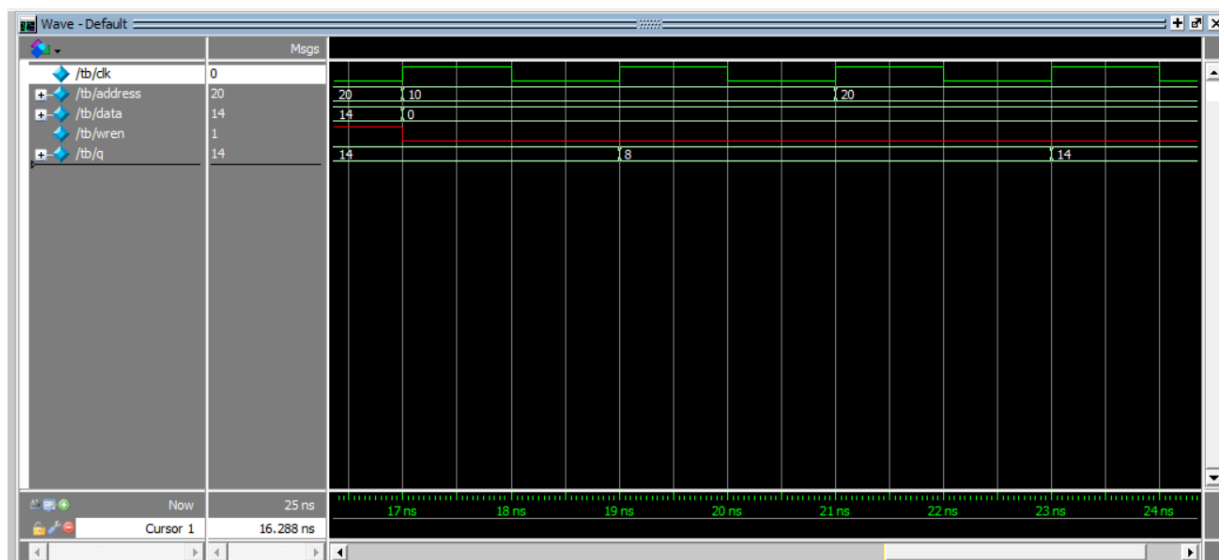
Leitura dos dados iniciais



Escrita dos dados



Leitura dos dados escritos



1.1.2 Principais decisões tomadas

Texto.

1.2 DISPLAY

Texto.

1.2.1 Imagens de tela da execução do programa

Foi feita uma simulação mostrando entradas de 0 a F e, cada uma das entradas é convertida conforme definido no código. Por exemplo, no caso da entrada 0, o valor de saída é 1000000, onde 0 é usado para acender o segmento correspondente à posição do algarismo no display e 1 é usado para apagar o segmento. No caso do F, o valor do HEX será 0001110.

1.2.2 Principais decisões tomadas

```
4'b0000: HEX = 7'b1000000; // Exibe 0
4'b0001: HEX = 7'b1111001; // Exibe 1
4'b0010: HEX = 7'b0100100; // Exibe 2
4'b0011: HEX = 7'b0110000; // Exibe 3
4'b0100: HEX = 7'b0011001; // Exibe 4
4'b0101: HEX = 7'b0010010; // Exibe 5
4'b0110: HEX = 7'b0000010; // Exibe 6
4'b0111: HEX = 7'b1111000; // Exibe 7
4'b1000: HEX = 7'b0000000; // Exibe 8
4'b1001: HEX = 7'b0010000; // Exibe 9
4'b1010: HEX = 7'b0001000; // Exibe A
4'b1011: HEX = 7'b0000011; // Exibe B
4'b1100: HEX = 7'b1000110; // Exibe C
4'b1101: HEX = 7'b1000001; // Exibe D
4'b1110: HEX = 7'b0000110; // Exibe E
4'b1111: HEX = 7'b0001110; // Exibe F
```

O módulo é usado para depuração de resultados que sejam importantes na simulação. Dado um valor de entrada, é feita uma conversão padrão para mostrar o mesmo valor em forma hexadecimal no display de 7 segmentos.

REFERÊNCIAS