

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**

**Departamento de Computação – DECOM**

**LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES**

**RELATÓRIO PRÁTICA 2 – PROCESSADOR MULTICICLO**

Jorge Aliomar Trocoli Abdon Dantas

Thiago Ribeiro Corrêa

**Professora: Poliana Corrêa**

**Belo Horizonte**

**01 de outubro de 2023**

1. INTRODUÇÃO

Esta prática consiste na implementação de um processador multiciclo, que implementa as seguintes instruções: LD, ST, MVNZ, MV, MVI, ADD, SUB, OR, SLT, SLL e SLR. As instruções devem ser armazenadas em uma memória RAM. Posteriormente elas serão lidas, escritas e processadas. A implementação da prática será realizada em Verilog-HDL, utilizando o software Quartus II, e as simulações dos resultados através do ModelSim.

A Figura 1 mostra esquematicamente o processador implementado.

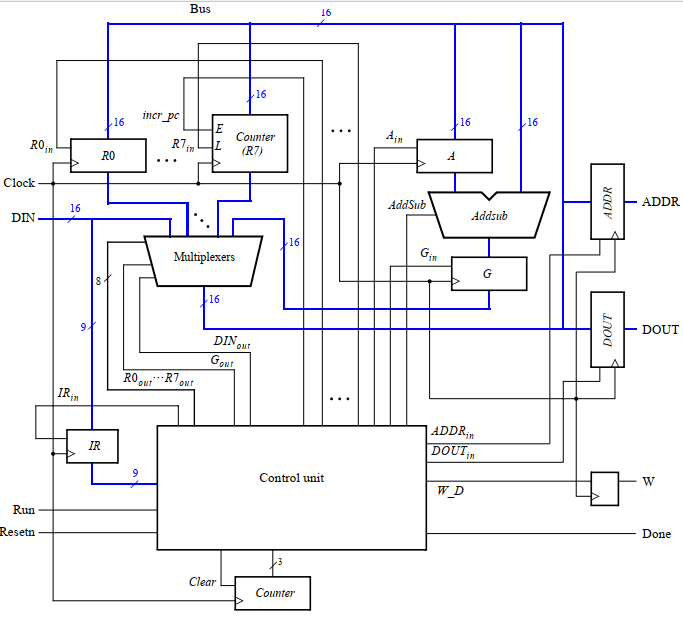


Figura : Esquema do processador multiciclo implementado

A Figura 2 mostra o esquemático do processador atuando juntamente com a memória.

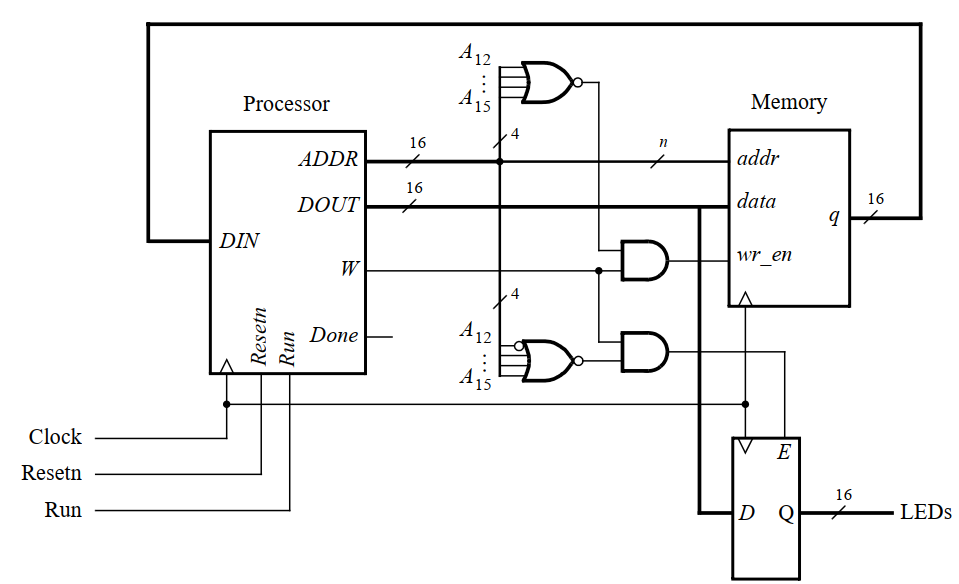


Figura : Esquema do processador multiciclo implementado atuando juntamente com a memória RAM

1. DESENVOLVIMENTO

Tendo como referência a Figura 1, implementou-se os seguintes módulos isolados:

1. registradoresR: registradores de 16 bits, de propósito geral que servem para armazenar os dados durante o processamento. São estes registrados os representados por R0 a R7, A, G, ADDR e DOUT.
2. registradoresI: registrador de 10 bits, responsável por ler a instrução atual, decodifica-la. Os 10 bits podem ser representados por IIIIXXXYYY, onde IIII representa o Opcode das instruções, XXX representa o registrador Rx, e o YYY representa o registrador Ry. As funções, suas características e os Opcodes das instruções implementadas são mostradas na Tabela 1.

Tabela : Características e Opcodes das instruções implementadas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instrução** | **Nome** | **Operação** | **Função desempenhada** | **Significado** | **OPCODE** |
| LD | Load | ld Rx,[Ry] | Rx ← [[Ry]] | Carrega os dados no registrador Rx a partir do endereço de memória especificado no registrador Ry. | 0000 |
| ST | Store | st Rx,[Ry] | [Ry] ← [Rx] | Armazena os dados contidos no registrador Rx no endereço de memória encontrado no registrador Ry. | 0001 |
| MVNZ | Move se não for zero | mvnz Rx, Ry | if G != 0, Rx ← [Ry] | Permite que uma operação MV seja executada somente sob uma determinada condição; a condição é que o conteúdo do registrador G não é igual a 0. | 0010 |
| MV | Move | mv Rx,Ry | Rx ← [Ry] | Conteúdo do registrador Ry é carregado no registrador Rx. | 0011 |
| MVI | Move imediato | mvi Rx,#D | Rx ← D | Imediato D de 16 bits é carregado no registrador Rx. | 0100 |
| ADD | Adição | add Rx, Ry | Rx ← [Rx] + [Ry] | Soma dos conteúdos dos registrados Rx e Ry são carregados no registrador Rx. | 0101 |
| SUB | Subtração | sub Rx, Ry | Rx ← [Rx] - [Ry] | Subtração dos conteúdos dos registrados Rx e Ry são carregados no registrador Rx. | 0110 |
| OR | Ou | or Rx,Ry | Rx ← [Rx] II [Ry] | Soma dos conteúdos dos registrados Rx e Ry são carregados no registrador Rx. | 0111 |
| SLT | Menor que | slt Rx,Ry | If (Rx < Ry) [Rx] = 1 else [Rx] = 0 | Se conteúdo do registrador Rx for menor que o conteúdo do registrador Ry, então Rx = 1. Caso contrário Rx = 0. | 1000 |
| SLL | Deslocamento para esquerda | sll Rx,Ry | Rx = [Rx] << [Ry] | Deslocamento do conteúdo do registrador Rx para esquerda na quantidade de vezes do conteúdo de Ry. Efeito multiplicador | 1001 |
| SLR | Deslocamento para direita | srl Rx,Ry | Rx = [Rx] >>[Ry] | Deslocamento do conteúdo do registrador Rx para direita na quantidade de vezes do conteúdo de Ry. Efeito divisor | 1010 |

1. mux: multiplexador responsável por selecionar, a partir dos sinais de controle da unidade de controle, a saída para o barramento (Bus) de 16 bits.
2. contadoresR: registrador que funciona como registrador de propósito geral e contador de PC. Especificamente, é o registrador R7.
3. ULA : Unidade Lógico Aritmética do processor. A ULA foi implementada com uma saída (ResultadoULA) e três entradas (A, B, Opcode), sendo A e B os operandos. É responsável pelas operações de add, sub, OR, slt, sll e slr. Salienta-se que a ULA não opera em ciclos de clock, e sim quando há mudança em um dos operandos ou operações. Devido a isto, o registrador especializado G fica em sua saída e, quando for o caso, armazena o resultado atual.
4. counter: contador responsável para ler o conteúdo de endereços sucessivos na memória para que esses dados sejam fornecidos ao processador como um fluxo de instruções.
5. dec3to8: decodificador utilizado para transformar a identificação de 3 bits dos registradores Rx e Ry para 8 bits.
6. processador : módulo que consiste na junção dos módulos acima, juntamente com as especificações da unidade de controle. Na unidade controle, tem-se o controle dos principais sinais, como o de sobrescrita de dados em registradores, escolha para a saída do multiplexador, operação realizada na ULA, e reinício do contador de ciclos.
7. ProcessadorMulticiclo: junção do processador com a memória RAM implementada
8. decodificador: decodificador para display de 7 segmentos.
9. RESULTADOS

A seguir serão mostradas as simulações realizadas via ModelSim para testar os módulos que compõem o processador. Após esta etapa será mostrada as simulações das instruções realizadas no processador.

1. **registradoresR**

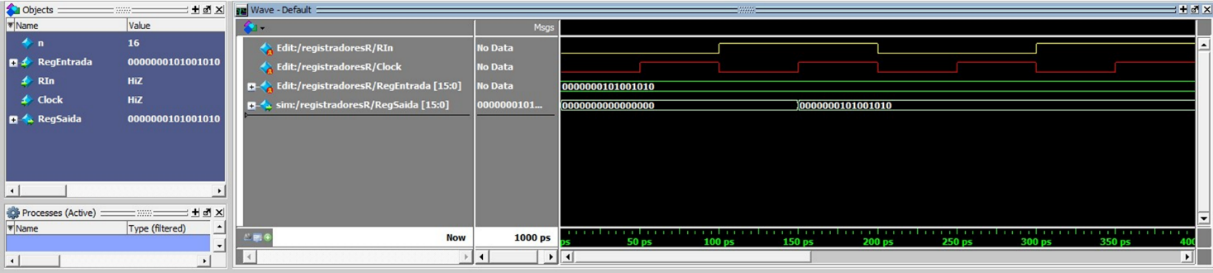


Figura : Simulação do módulo registradoresR

Observando a Figura 3, tem-se que o resultado de simulação está de acordo com o comportamento esperado. Observa-se que quando ocorre a mudança de clock na subida, juntamente com o sinal de controle RIn ativo, o valor da saída do registrador altera de 0 para o valor da entrada do registrador (vide transição em 150ps).

1. **registradoresI**

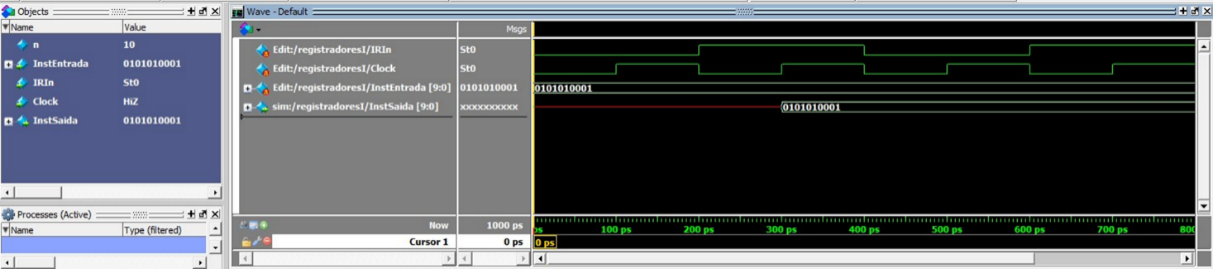


Figura :Simulação do módulo registradores

Observando a Figura 4, tem-se que o resultado de simulação está de acordo com o comportamento esperado. Observa-se que quando ocorre a mudança de clock na subida, juntamente com o sinal de controle IRIn ativo, o valor da saída do registrador altera de indefinido para o valor da entrada do registrador (vide transição em 300ps).

1. **Mux**

Observando a Figura 5, tem-se que o resultado de simulação está de acordo com o comportamento esperado. Observa-se que ao se alterar o sinal de controle ROut, altera-se o valor de saída do mux. Observa-se o mesmo comportamente para a saída do mux associada ao registrador G (vide transição em 900ps) e a entrada DIN (vide transição em 1100ps).

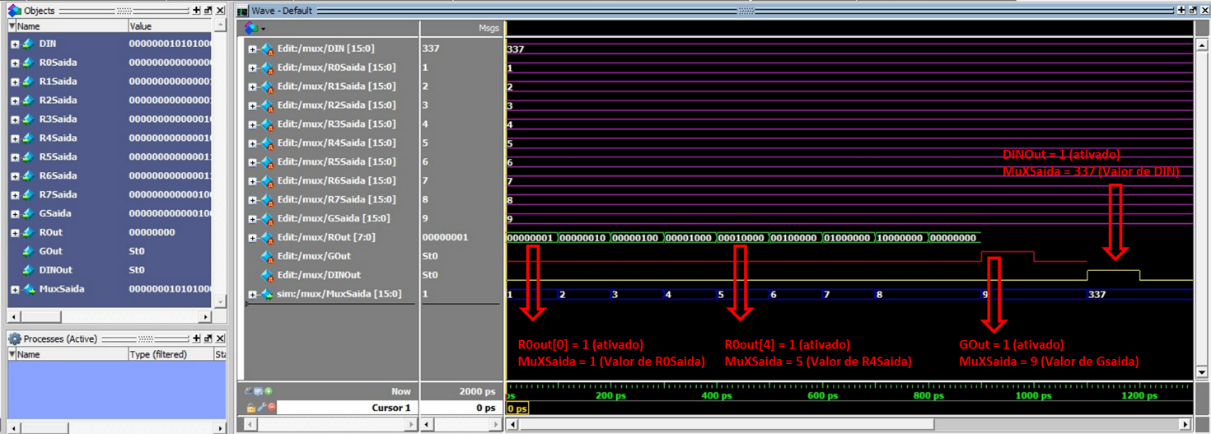


Figura :Simulação do módulo mux

1. **contadoresR**

Como mencionado anteriormente, o módulo contadores pode funcionar como um registrador de uso geral, assim como, um contador de PC. Na Figura 6, tem-se que o resultado de simulação no modulo de registrador. Neste módulo o sinal de controle IncrPc fica desativado. O comportamento esperado nesta situação é semelhante ao do módulo registradoresR.

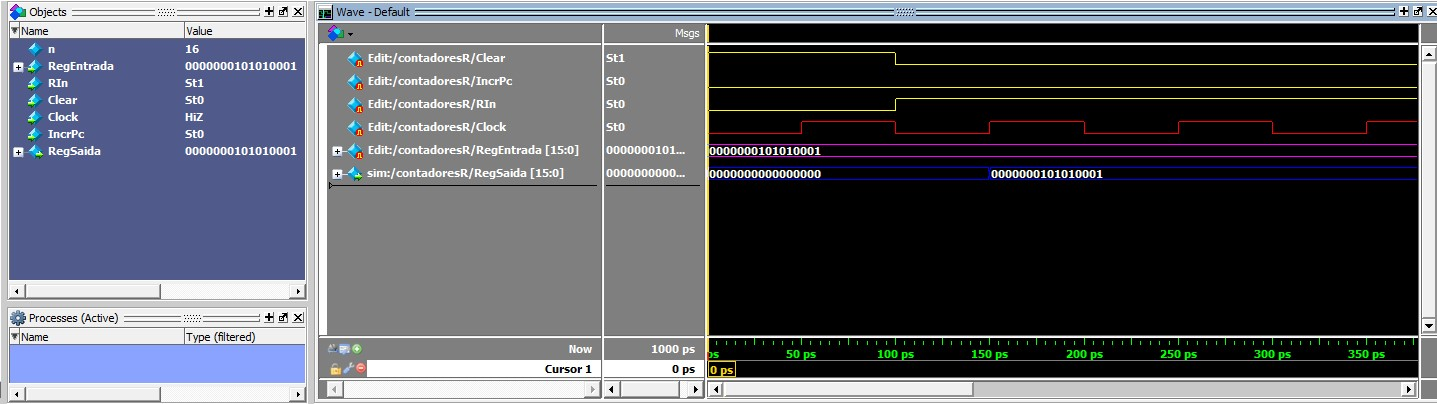


Figura : Simulação do módulo contadoresR - modo registrador

A Figura 7, por sua vez, mostra a simulação do módulo no modo de contador de PC. Neste modo, o sinal de controle RIn fica desativado, e contagem é incrementada em cada mudança de clock na borda de subida. A saída nesta situação é a contagem realizada no módulo.

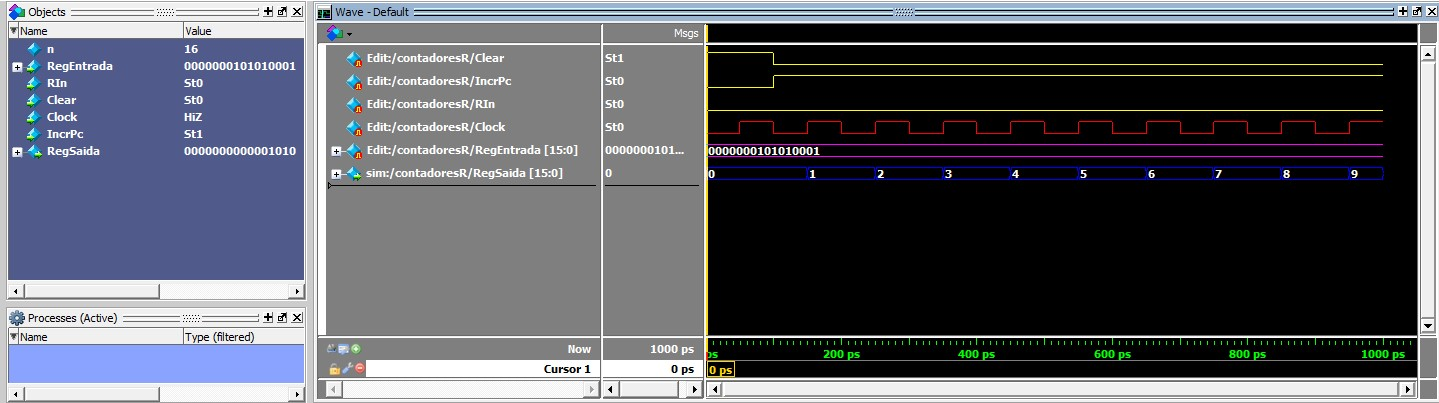


Figura :Simulação do módulo contadoresR - modo contador PC

1. **ULA**

A Figura 8, mostra a simulação do módulo da ULA. Observa-se que todas as operações realizadas na ULA se mostraram de acordo com o esperado.

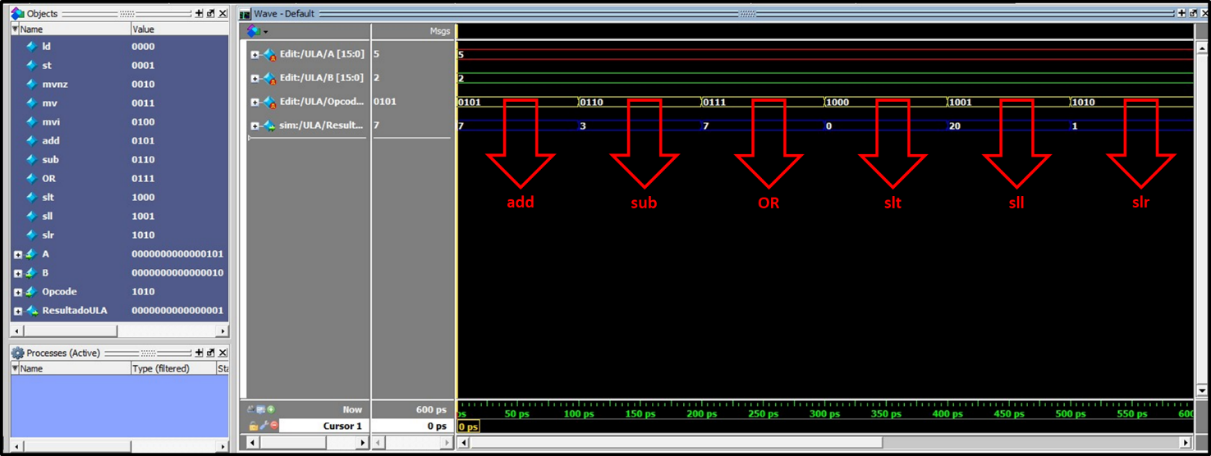


Figura : Simulação do módulo ULA

1. **counter**

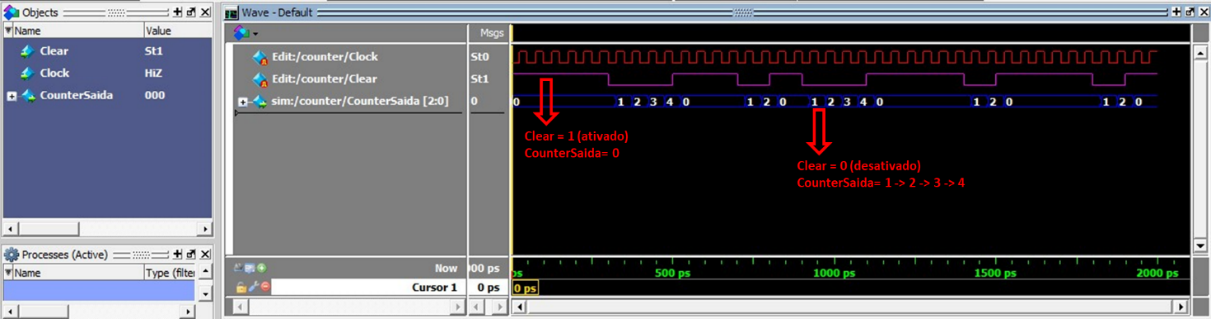


Figura : Simulação do módulo counter

No módulo para contador de ciclos, observa-se que quando o sinal de Clear está ativo a contagem é zerada, independente da mudança do clock. Quando ele é desativado, obseva-se o incremento da contagem em cada subida do clock. Vide Figura 9.

1. **dec3to8 e decodificador**

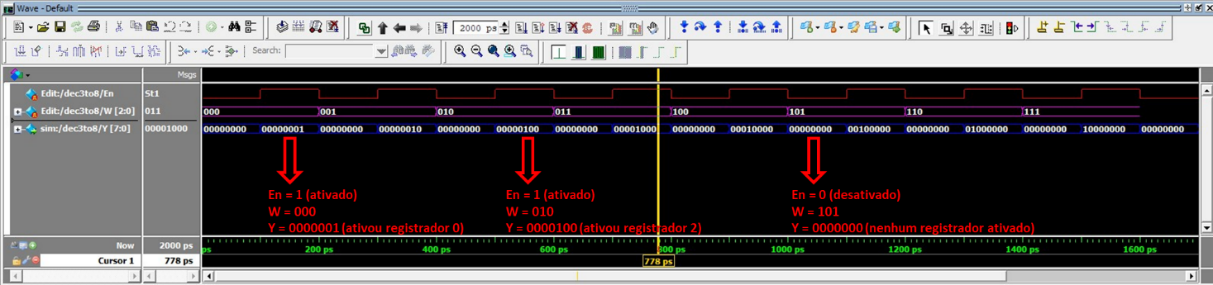


Figura : Simulação do módulo dec3to8

Para os 2 decodificadores, também se observa o comportamento esperado. No módulo dec3to8 observa-se que quando o sinal En está ativo, há a decodificação do valor de entrada, mostrando um valor de 8 bits na saída. Em contrapartida, quando o sinal En está inativo, a saída é sempre 0, mostrando que não há nenhum registrador ativo. Vide Figura 10.

Pela Figura 11, é possível relacionar o resultado do decodificador com o valor mostrado no display de 7 segmentos.

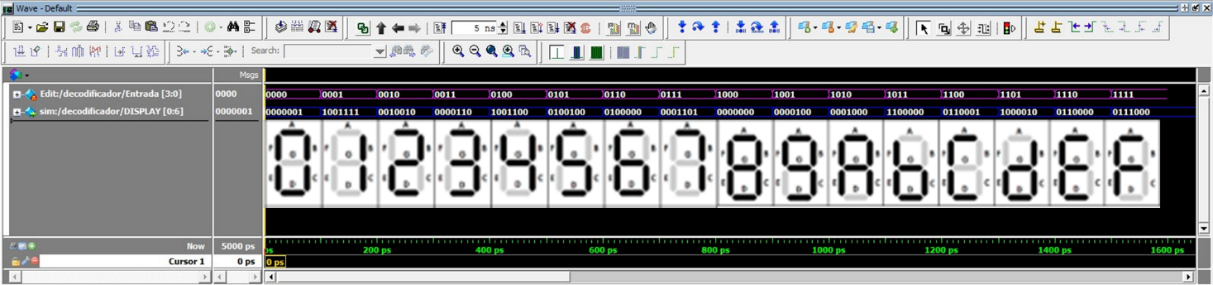


Figura :Simulação do módulo decodificador

1. **processador**

Não se conseguiu simular o comportamento do processador via ModelSim.

1. **ProcessadorMulticiclo**

Não se conseguiu simular o comportamento do ProcessadorMulticiclo via ModelSim. Salienta-se que nos últimos minutos da aula experimental, ao manusear aleatoriamente a placa, observou-se que havia a leitura da memória com os sinais de Run ativo, e os sinais de clock e clear ativos simultaneamente. A primeira vista, parecia que os resultados na placa se mostravam condizentes com os resultados esperados. No entanto, devido ao fim da aula, não pode-se constatar a consistência das respostas. Tentou-se encontrar a inconsistência no código sem sucesso.

Todos os códigos estão disponíveis no arquivo disponibilizado.

1. REFERÊNCIAS

D. A. Patterson, J. L. Hennessy. Organização e Projeto de Computadores - A Interface hardware / software. Rio de Janeiro. Elsevier, 2014. Editora Campus.

Laboratory Exercise 9 - A Simple Processor

Laboratory Exercise 10 - An Enhanced Processor