



Departamento de Ciência da Computação

Prof. Bruno de Abreu Silva

GCC260 – Laboratório de Circuitos Digitais

Simulação de circuitos usando o ModelSim

1. Objetivos

- Entender e realizar a simulação de circuitos digitais;
- Implementar um *testbench* básico em Verilog;
- Saber usar o Quartus Prime e o ModelSim Intel FPGA Starter Edition para simulação de circuitos digitais.

2. Como realizar a simulação de circuitos digitais?

Para verificar a correteza da operação de um circuito após o seu desenvolvimento, pode ser realizada uma simulação. Verilog também pode ser usada para realizar a simulação. Para isso, deve ser criado um código especial, chamado de *testbench* (teste de bancada), com o intuito de imitar os testes de circuitos realizados em uma bancada de laboratório.

3. Criando um projeto exemplo para simulação

O circuito projetado deve ser descrito em Verilog e estar de acordo com a especificação a seguir:

- O circuito deve receber como entrada, através das switches do kit DE10-LITE, um número de quatro bits representado como $SW_3SW_2SW_1SW_0$, em que SW_3, SW_2, SW_1, SW_0 são os bits individuais e SW_0 é o bit menos significativo. Projete um circuito lógico que gere um nível ALTO na saída (usando um LED, por exemplo LEDR[0]) sempre que o número binário for maior que 0010 e menor que 1000.

As seguintes etapas devem ser realizadas para desenvolver o projeto:

- Crie uma nova pasta chamada Pratica4, na Área de Trabalho;

- Baixe o código DE10_LITE_Golden_Top_tb.v em: https://github.com/brabreus/GCC260-UFLA/blob/main/Pratica4/DE10_LITE_Golden_Top_tb.v e salve na pasta Pratica4;
- Crie o projeto Pratica4 no Quartus Prime, lembrando de **alterar o diretório** para criar o projeto na pasta Pratica4, **adicionar no projeto o arquivo já baixado** e definir a placa (board) como da família MAX10, kit DE10-Lite **deixando marcada a opção “Create top-level design file”**;
- Entender o código baixado (ver Seção 4 e explicação do Professor);
- Abra o arquivo DE10_LITE_Golden_Top.v acessando *File->Open*;
- Projete o circuito especificado no início desta seção usando o mapa de karnaugh e, em seguida, complete o código em Verilog com a expressão resultante do mapa (utilize o comando **assign**, usado na Prática 2);
- Compile parcialmente o projeto em *Processing->Start...->Start Analysis & Synthesis* (essa opção de compilação é mais rápida e verifica somente se há erros de sintaxe no código Verilog – não gera o arquivo para programação do kit).

4. O arquivo de teste DE10_LITE_Golden_Top_tb.v (*testbench*)

O arquivo de teste **DE10_LITE_Golden_Top_tb.v** define um módulo de teste, as declarações de sinais internos, a instanciação do módulo a ser testado e todos os vetores de teste, que são correspondentes a todas as possibilidades para os valores das entradas. Neste caso, são 16 vetores de testes.

```
// The `timescale directive specifies that
// the simulation time unit is 1ns and
// the simulation timestep is 10ps
`timescale 1ns/10ps
```

```
module DE10_LITE_Golden_Top_tb;
    // signal declaration
    reg [3:0] test_in;
    wire test_out;

    // instantiate the circuit under test
    DE10_LITE_Golden_Top uut (.SW(test_in), .LEDR(test_out));
```

```
// test vector generator
initial
begin
    // test vector 1
    test_in = 4'b0000;
    #200;
    // test vector 2
    test_in = 4'b0001;
    #200;
    // test vector 3
    test_in = 4'b0010;
    #200;
    // test vector 4
    test_in = 4'b0011;
    #200;
    // test vector 5
    test_in = 4'b0100;
    #200;
    // test vector 6
    test_in = 4'b0101;
    #200;
    // test vector 7
    test_in = 4'b0110;
    #200;
    // test vector 8
    test_in = 4'b0111;
    #200;
    // test vector 9
    test_in = 4'b1000;
```

```

#200;
// test vector 10
test_in = 4'b1001;
#200;
// test vector 11
test_in = 4'b1010;
#200;
// test vector 12
test_in = 4'b1011;
#200;
// test vector 13
test_in = 4'b1100;
#200;
// test vector 14
test_in = 4'b1101;
#200;
// test vector 15
test_in = 4'b1110;
#200;
// test vector 16
test_in = 4'b1111;
#200;
// stop simulation
$stop;
end
endmodule

```

O bloco uut é a unidade sob teste (*unit under test*) e os blocos de vetores geradores de testes geram os padrões de entrada dos testes. O código consiste em um módulo responsável por instanciar o módulo DE10_LITE_Golden_Top que será testado e também por gerar o padrão de teste. O bloco *initial* em Verilog é utilizado na simulação e é executado somente uma vez quando a simulação

Construir *testbenchs* de qualidade é uma tarefa que exige bons conhecimentos sobre Verilog. Entretanto, a estrutura básica deste exemplo, nos permite ter um *template* útil para criar *testbenchs* para circuitos combinacionais simples.

A simulação será realizada em outro software específico para simulação, chamado ModelSim. A versão utilizada é a ModelSim - Intel FPGA Starter Edition, que é uma versão gratuita já instalada juntamente com o Quartus Prime. Usaremos a versão 10.5b.

The screenshot displays the ModelSim IDE interface. The top menu bar includes File, Edit, View, Compile, Simulate, Add Library, Tools, Layout, Bookmarks, Window, and Help. Below the menu is a toolbar with various icons for file operations, simulation, and layout. The main window is divided into two panes. The upper pane, titled 'Library', shows a list of libraries and their paths. The lower pane, titled 'Transcript', shows the output of the simulation, including the loading of project work and the reading of modelim.ini.

Name	Type	Path
#220model	Library	#M0001_TECH\altera\hnd\220model
#220model_ver	Library	#M0001_TECH\altera\verlog\220m...
#altera	Library	#M0001_TECH\altera\hnd\altera...
#altera_hsm	Library	#M0001_TECH\altera\hnd\altera_...
#altera_hsm_ver	Library	#M0001_TECH\altera\verlog\altera...
#altera_nrf	Library	#M0001_TECH\altera\hnd\altera_nrf
#altera_nrf_ver	Library	#M0001_TECH\altera\verlog\altera...
#altera_ver	Library	#M0001_TECH\altera\verlog\altera...
#arnia	Library	#M0001_TECH\altera\hnd\arnia
#arnia_hssi	Library	#M0001_TECH\altera\hnd\arnia_hssi
#arnia_hssi_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_pcie_hip	Library	#M0001_TECH\altera\hnd\arnia_p...
#arnia_pcie_hip_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz	Library	#M0001_TECH\altera\hnd\arnia_gz
#arnia_gz_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_hssi_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_pcie_hip	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_pcie_hip_v...	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_v	Library	#M0001_TECH\altera\hnd\arnia_v
#arnia_v_hssi	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	#M0001_TECH\altera\hnd\arnia_gz...
#arnia_gz_v_pcie_hip	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_ver	Library	#M0001_TECH\altera\verlog\arnia...
#arnia_gz_v_hssi	Library	

- Crie a pasta *simulation* dentro da pasta *Pratica4* na *Área de Trabalho*;
- No ModelSim, acesse *File->New->Project...* para criar um novo projeto de simulação;
- Defina *Project Name* como *Pratica4*;

- Defina *Project Location* como a pasta *simulation*;
- Defina o *Default Library Name* como *work*;
- E clique em *Reference Library Mappings*;
- A Figura 1 ilustra o preenchimento feito acima na janela que será exibida. **Todos os passos acima são extremamente importantes e devem ser feitos!**
- Após conferir se está tudo certo, clique em OK;

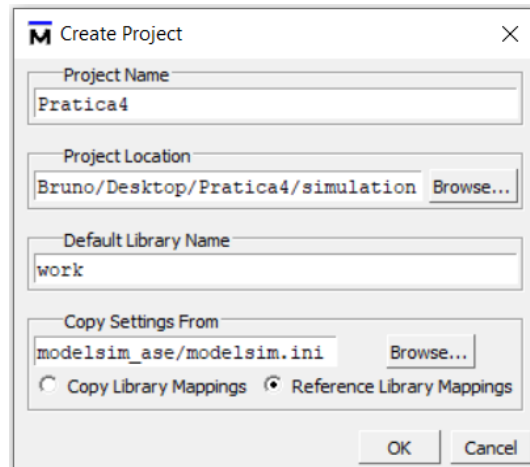


Figura 1: ModelSim - Create Project.

- Na janela seguinte, iremos adicionar os arquivos Verilog no projeto de simulação clicando em *Add Existing File* (Figura 2);

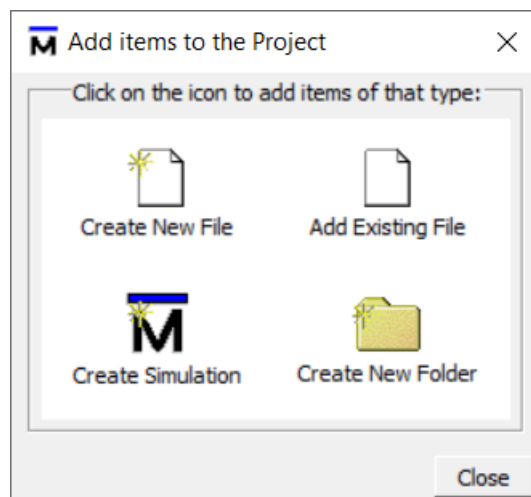


Figura 2: ModelSim, Adding Existing File.

- Na janela que aparecer (Figura 3), clique em Browse e selecione os dois arquivos em Verilog DE10_LITE_Golden_Top.v e DE10_LITE_Golden_Top_tb.v, localizados na pasta Pratica4 (ver Figura 4). Clique em Abrir, certifique-se de que a opção *Reference from current location* esteja marcada e depois em OK. Por fim, clique em Close;

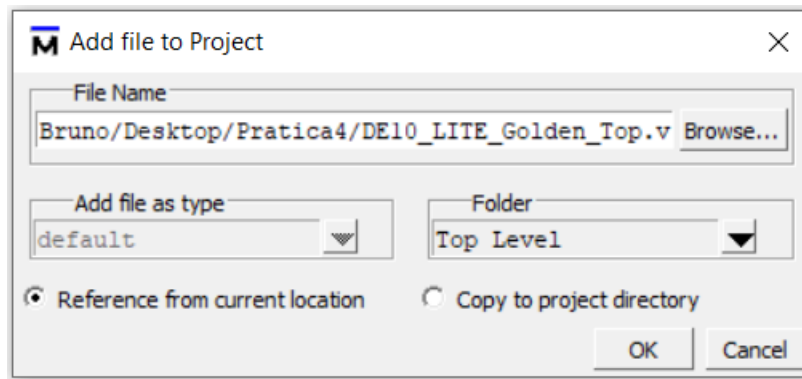


Figura 3: ModelSim - Add file do Project.

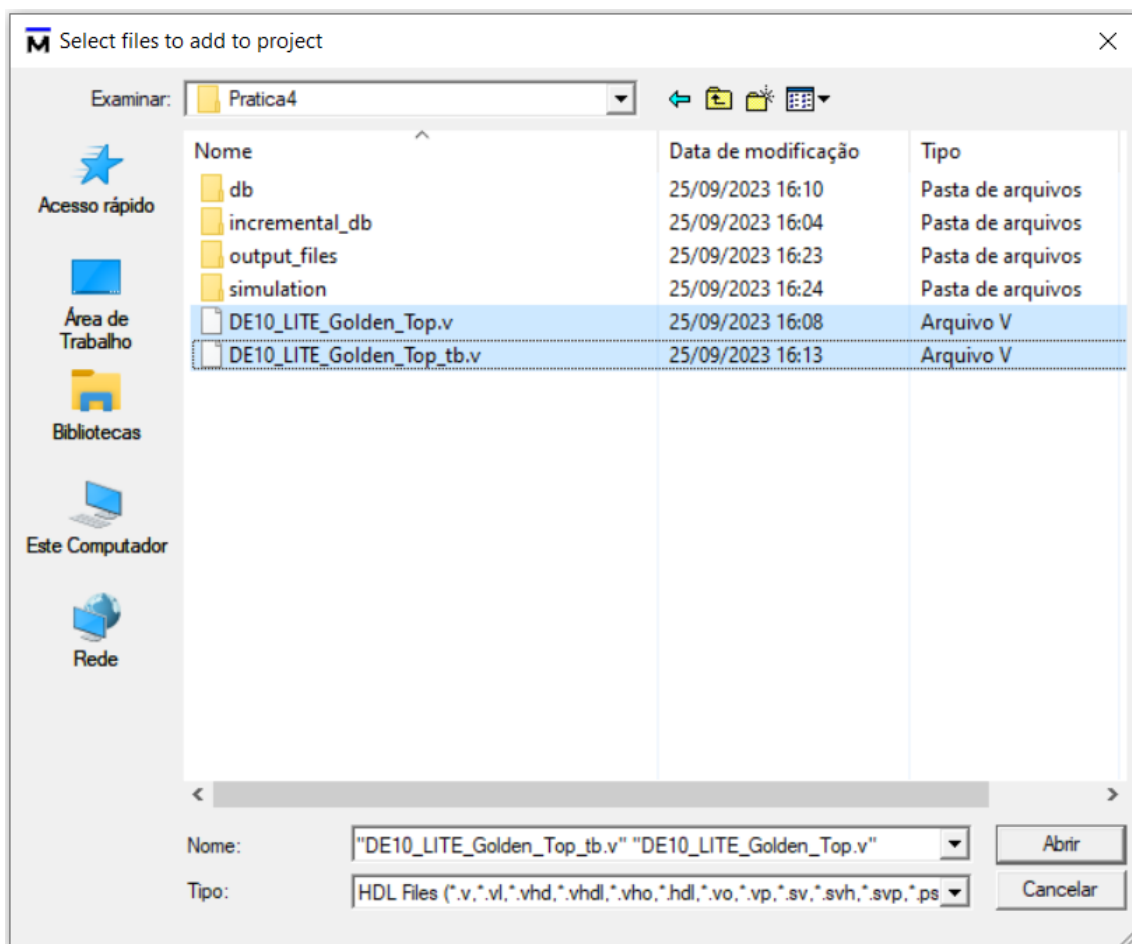


Figura 4: Select files to add to Project.

- Após o projeto ser criado, o software fica na forma exibida na Figura 5;

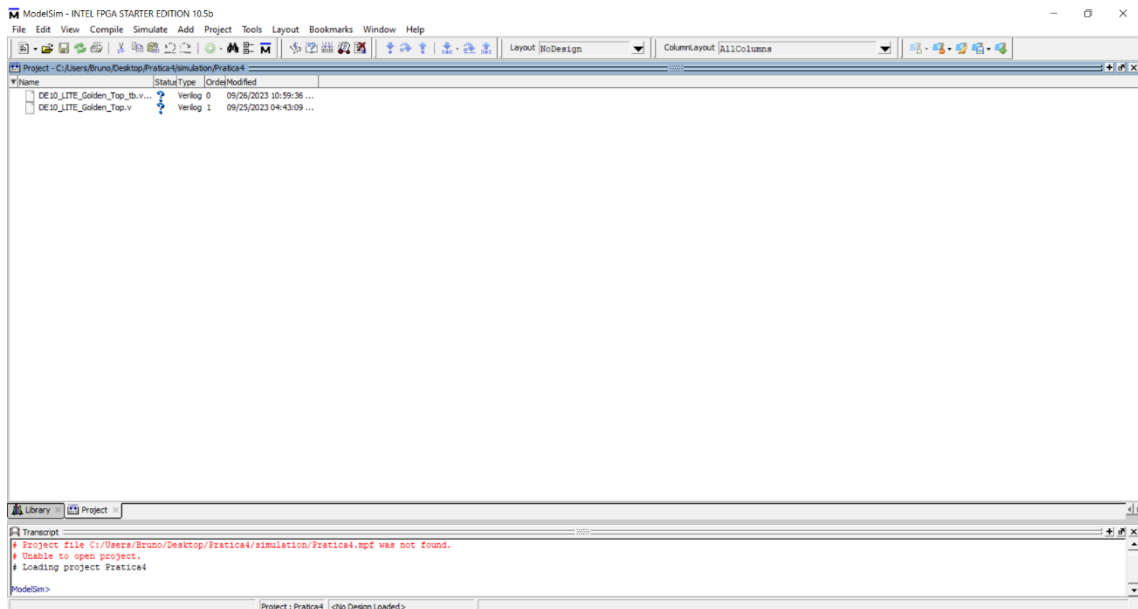


Figura 5: ModelSim - Projeto criado.

- Compile o projeto, acessando, no menu principal, *Compile->Compile All*;
- Após a compilação aparecerão duas letras V em verde, em vez dos pontos de interrogação azuis;
- Acesse, no menu principal, *Simulate->Start Simulation*;
- Na janela que abrir, expanda a Library work, clicando no + ao seu lado esquerdo (Figura 6);

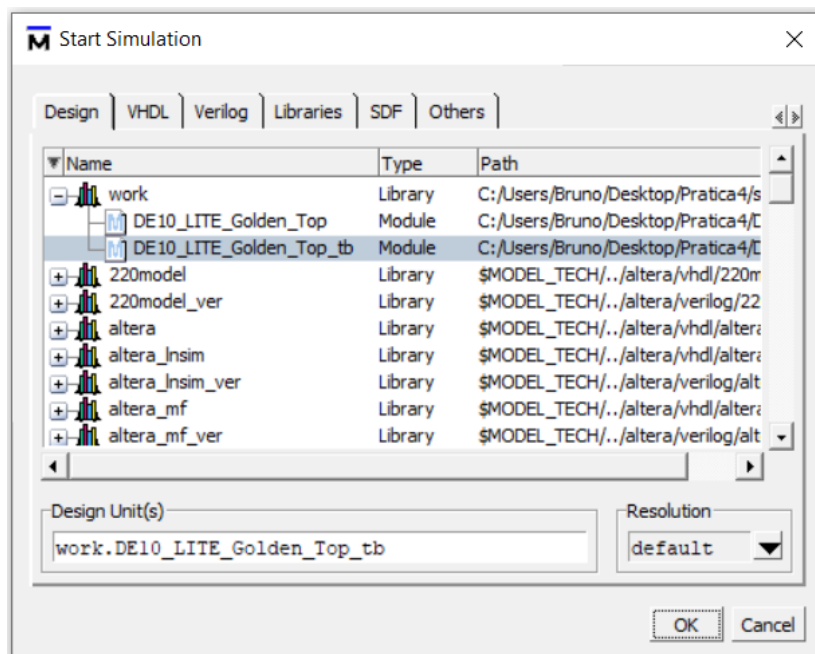


Figura 6: Start Simulation.

- Selecione **somente** o Module *DE10_LITE_Golden_Top_tb* e clique em OK. O modo simulação será exibido conforme a Figura 7 (Caso a tela roxa do meio, chamada Objects não apareça, acesse o menu View e selecione

Objects e, caso a janela Wave preta não apareça, acesse o menu View e selecione Wave);

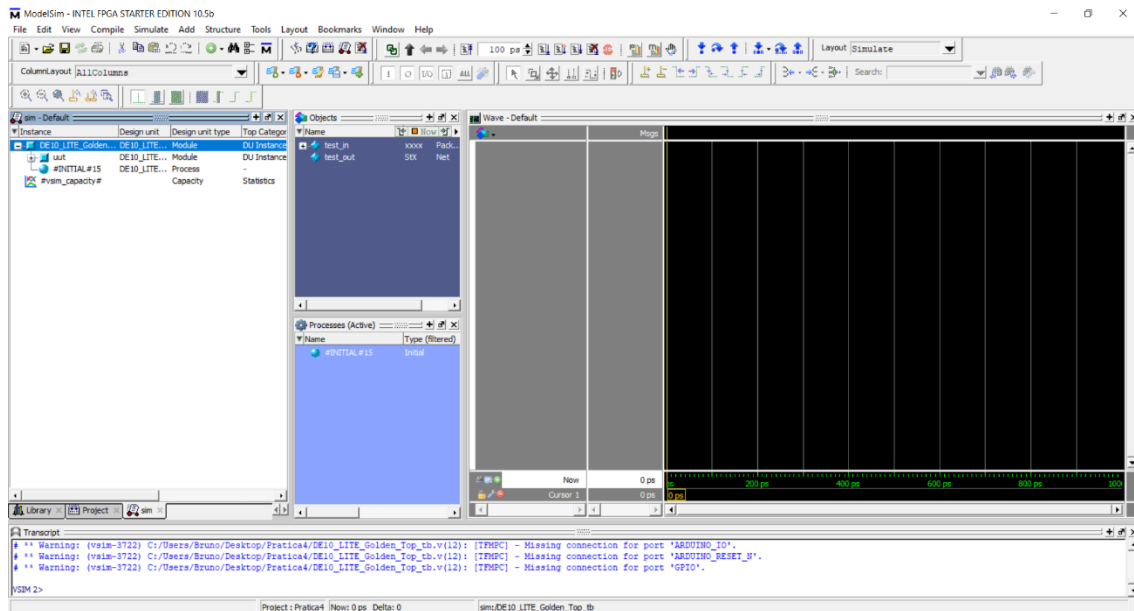


Figura 7: Simulation.

- Na janela branca à esquerda (*sim – Default*), clique em DE10_LITE_Golden_Top_tb e arraste para a região cinza no meio da tela (*Wave – Default*), para adicionar os sinais do *testbench* na *waveform* que será gerada (Figura 8);

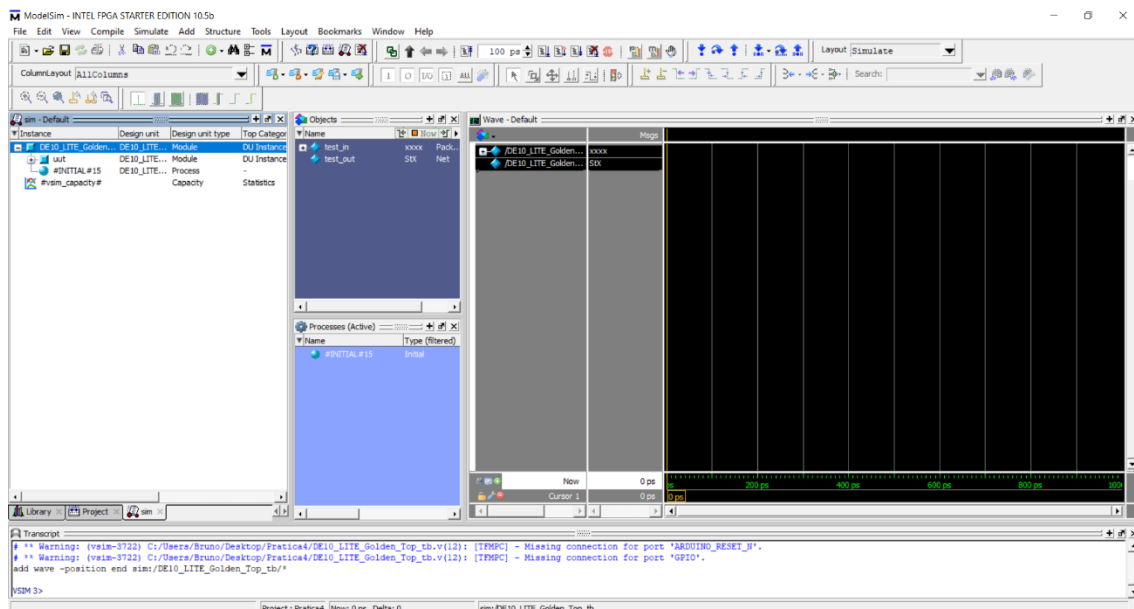


Figura 8: Adicionando sinais à waveform.

- Finalmente, agora basta acessar, no menu principal, *Simulate->Run->Run -All*. A simulação é executada, o arquivo do testbench é aberto para consulta e o waveform é gerado. Duas abas ficam abertas com os resultados na parte inferior direita da tela (Figura 9);

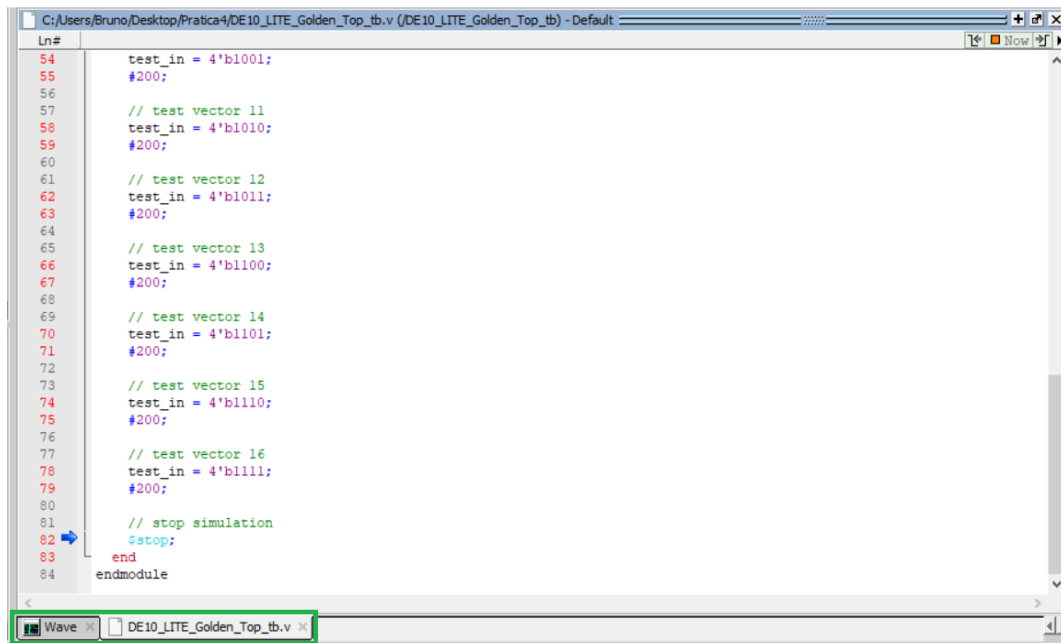


Figura 9: Abas da simulação.

- Clique na aba Wave para ver a forma de onda gerada pela simulação;
- Clique com o botão direito na região preta da tela e clique em Zoom Full para visualizar toda a forma de onda na tela;
- Após corrigir o zoom, verifique se o resultado da simulação na forma de onda está de acordo com o esperado (Figura 10) para este projeto (verifique se a saída está de acordo com a especificação do circuito, sua tabela verdade e expressão booleana gerada pelo mapa de Karnaugh) Obs.: O zoom da waveform pode ser alterado ao se clicar com o botão direito do mouse na região preta e selecionando *Zoom In* ou *Zoom Out* ou mesmo *Zoom Full*.

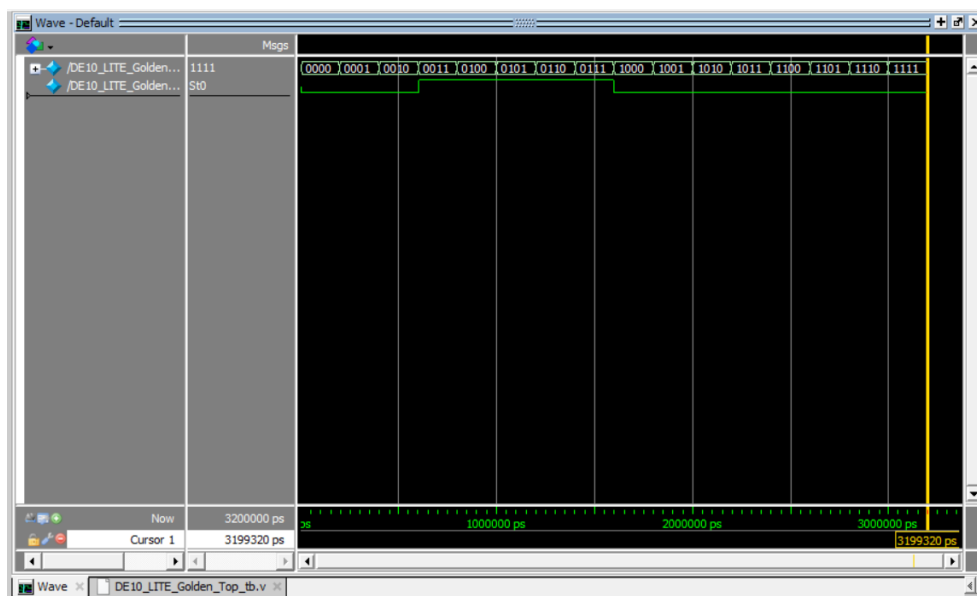


Figura 10: Resultado esperado para a simulação.

6. Testando o circuito na placa DE10-Lite (opcional)

Após a simulação, já sabemos que nosso código está logicamente correto. Portanto, podemos configurar a FPGA para ver o circuito funcionando realmente. Para isso basta compilar totalmente o projeto no Quartus Prime, acessando *Processing->Start Compilation*. E logo em seguida, *Tools->Programmer*, definir USB-Blaster em Hardware Setup e clicar em Start. Com isso, o circuito poderá ser verificado usando as chaves da DE10-Lite e a saída será exibida em um dos LEDs vermelhos.

7. Finalizando a atividade

Apresente o resultado para o Professor e envie a pasta com os arquivos da prática compactados (em .rar ou .zip) para a sala da disciplina no Campus Virtual.