

EEA-25 – Sistemas Digitais Programáveis

Prof. Dr. Harlei Miguel Arruda Leite | IEE, Sala 187

harlei@ita.br

Laboratório 02

1. Objetivos

- Familiarizar com o ambiente de desenvolvimento; Compreender o processo de design de circuitos usando Verilog e implementação em FPGA.

2. Roteiro

2.1. Criando um projeto:

- Crie uma pasta no formato EEA25-Lab1-<MembrosDoGrupo>;
- Dentro da pasta, crie um arquivo denominado full_adder.v;

2.2. No arquivo full_adder.v, implemente o full adder da Figura 1 utilizando descrição estrutural:

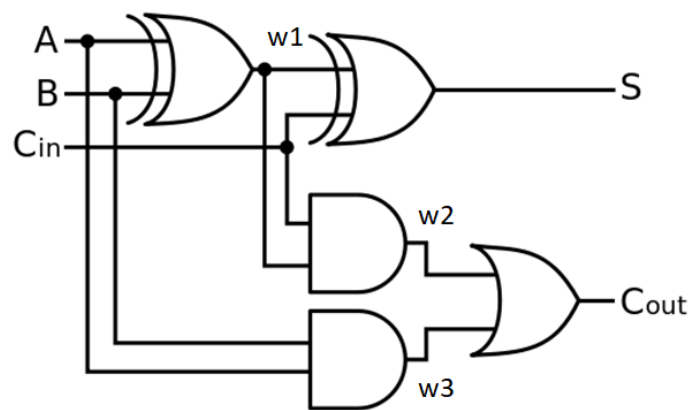


Figura 1 – Full adder.

- A descrição estrutural consiste nos passos abaixo:
 - Descreva o circuito em termos de portas lógicas, blocos e suas interconexões;
 - Use blocos elementares (primitivas) para construir o comportamento do sistema (AND, OR, NAND, NOR, XOR, etc);
 - Crie nodes “wires” (w1, w2 e w3) para realizar a conexão entre entrada/saída.

2.3. Simulação e testbench:

- Antes de sintetizar o circuito na FPGA, é necessário simular. Para isso, é preciso criar um testbench para o full adder;
- Crie um arquivo denominado full_adder_tb.v, e implemente o código da Figura 2;

```
1 // tb for full adder
2
3 `include "full_adder.v"
4
5 module full_adder_tb;
6
7     reg A, B, CIN; // UUT inputs
8     wire S, COUT; // UUT outputs
9
10    // full adder instance
11    full_adder uut( .a(A), .b(B), .cin(CIN),
12                   |   |   |   | .s(S), .cout(COUT));
13
14    initial begin
15        $dumpfile("full_adder_tb.vcd");
16        $dumpvars(0, full_adder_tb);
17    end
18
19    initial begin
20        A = 1'b0; B = 1'b0; CIN = 1'b0; #1
21        A = 1'b0; B = 1'b0; CIN = 1'b1; #1
22        A = 1'b0; B = 1'b1; CIN = 1'b0; #1
23        A = 1'b0; B = 1'b1; CIN = 1'b1; #1
24        A = 1'b1; B = 1'b0; CIN = 1'b0; #1
25        A = 1'b1; B = 1'b0; CIN = 1'b1; #1
26        A = 1'b1; B = 1'b1; CIN = 1'b0; #1
27        A = 1'b1; B = 1'b1; CIN = 1'b1; #1;
28    end
29
30 endmodule
```

Figura 2 – Testbench.

- Abra o terminal e digite o código a seguir:
 - iverilog full_adder_tb.v
 - vvp a.out
 - gtkwave full_adder_tb.vcd
- No gtkwave, selecione as entradas e saídas, como mostra a Figura 3, clique com o botão direito > Recurse Import > Append;

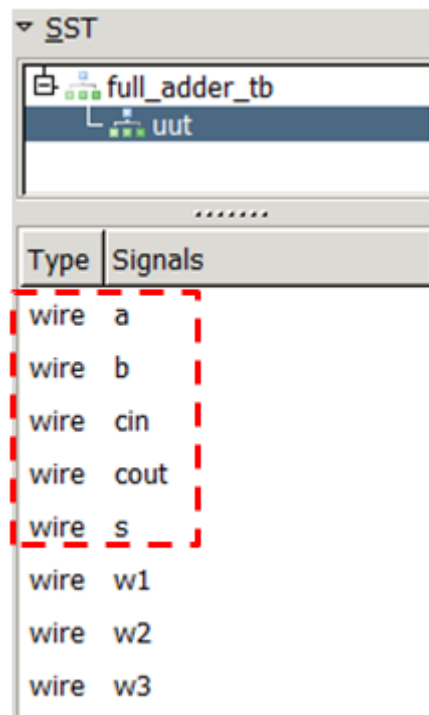


Figura 3 – Selecionando as entradas e saídas do gtkwave.

- Ajuste o zoom da imagem até visualizar toda a onda. Você deverá obter uma imagem como mostra a Figura 4.

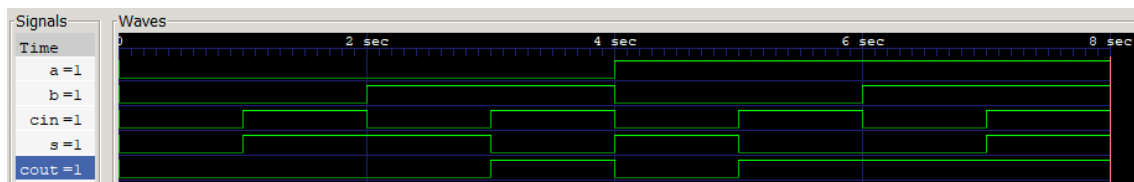


Figura 4 – Entradas e saídas visualizadas no gtkwave.

2.4. Síntese:

- Sabendo que a descrição do circuito está correta, podemos passar para a etapa de síntese;
- Crie o arquivo full_adder.pcf como mostra a Figura 5;
- Crie o arquivo Makefile exatamente como no Laboratório 01, alterando somente o filename para full_adder e pcf_file para full_adder.pcf;
- Conecte a FPGA no computador;
- Abra o terminal e acesse a página do projeto, e então digite o comando:
 - make build
- Não ocorrendo nenhum erro, arraste o arquivo full_adder.bin para o drive iCELink utilizando a interface gráfica, ou então digite make prog_flash no terminal para realizar via linha de comando.

```
1  #   BOARD PINS
2
3  #   FPGA pins:
4
5  #   PMOD I/O in use - OUTPUT-FULL-ADDER
6  set_io cout      C5
7  set_io s         B3
8
9  #   PMOD I/O in use - INPUT-FULL-ADDER
10
11 set_io -pullup yes a      C6
12 set_io -pullup yes b      B4
13 set_io -pullup yes cin    B5
14
15 set_io --warn-no-port PMOD4 E3
16 set_io --warn-no-port PMOD5 E1
17 set_io --warn-no-port PMOD6 C2
18 set_io --warn-no-port PMOD7 B1
19 set_io --warn-no-port PMOD8 A1
20
21 set_io --warn-no-port PMODL2 A3
22 set_io --warn-no-port PMODL3 B6
23
24 set_io --warn-no-port PMODR1 A1
25 set_io --warn-no-port PMODR2 B1
26 set_io --warn-no-port PMODR3 D1
27 set_io --warn-no-port PMODR4 E2
```

Figura 5 – Arquivo full_adder.pcf.

3. Critério de avaliação

- Este laboratório não requer relatório;
- Quando finalizar, chame o professor para verificar o projeto.