



INSTITUTO TECNOLÓGICO DE AERONÁUTICA
DIVISÃO DE CIÊNCIA DA COMPUTAÇÃO - IEC
DEPARTAMENTO DE SISTEMAS DE COMPUTAÇÃO - IEC-SC

EEA-25 - SISTEMAS DIGITAIS PROGRAMÁVEIS

LISTA DE EXERCÍCIOS Nº 1

PROF. ANDRÉ DA FONTOURA PONCHET

SÃO JOSÉ DOS CAMPOS - SP

15 DE AGOSTO DE 2023

SUMÁRIO

1	EXERCÍCIOS TEÓRICOS	2
1.1	Exercício 1 (1,0)	2
1.2	Exercício 2 (1,5)	2
1.3	Exercício 3 (1,5)	2
1.4	Exercício 4 (1,0)	2
1.5	Exercício 5 (1,5)	2
2	EXERCÍCIO DE SIMULAÇÃO (3,5)	2

1 EXERCÍCIOS TEÓRICOS

1.1 EXERCÍCIO 1 (1,0)

Qual a definição de linguagem HDL. Forneça alguns exemplos.

1.2 EXERCÍCIO 2 (1,5)

Quais as vantagens de utilização de linguagens HDL em projeto de circuitos digitais? Quais os níveis de abstração de uma linguagem HDL? Explique.

1.3 EXERCÍCIO 3 (1,5)

Qual a entidade básica de um projeto descrito em linguagem Verilog? Apresente o diagrama hierárquico completo de um projeto e testbench descrito em linguagem Verilog. Explique a funcionalidade de cada bloco da hierarquia.

1.4 EXERCÍCIO 4 (1,0)

Qual a diferença básica entre uma net e uma variável descritas em linguagem Verilog? Explique.

1.5 EXERCÍCIO 5 (1,5)

Escreva um código Verilog para instanciar (associar pinos com as nets) o bloco apresentado na Figura 1:



Figura 1: Diagrama do Exercício 5.

2 EXERCÍCIO DE SIMULAÇÃO (3,5)

Considere o multiplexador apresentado na Figura 2:

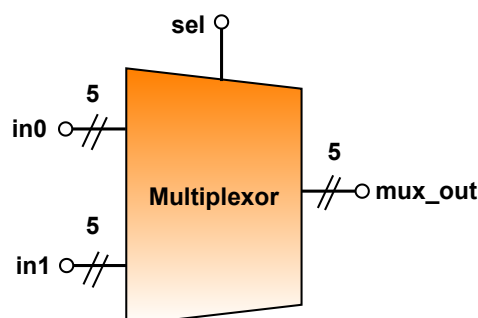


Figura 2: Diagrama do Exercício 1.

Especificações do multiplexador:

- O barramento do multiplexador é de 5 bits por padrão.
- Se $sel = 0$, a entrada $in0$ é transferida para a saída mux_out .
- Se $sel = 1$, a entrada $in1$ é transferida para a saída mux_out .

Crie um arquivo com nome *multiplexor.v*, e utilizando o seu editor favorito, nomeie o módulo com o nome *multiplexor*. Parametrize as entradas e saídas do multiplexador e atribua um valor padrão para a largura dos barramentos de entrada e saída como sendo igual a 5, conforme apresentado na Figura 2. Verifique o funcionamento do multiplexador instanciando o arquivo *multiplexor.v* no arquivo do *testbench* *multiplexor_test.v*, conforme o código apresentado abaixo:

```
'include "multiplexor.v"

module multiplexor_test;

    localparam WIDTH=5;

    reg          sel    ;
    reg [WIDTH-1:0] in0  ;
    reg [WIDTH-1:0] in1  ;
    wire [WIDTH-1:0] mux_out;

    multiplexor
    #(
        .WIDTH ( WIDTH )
    )
    multiplexor_inst
    (
        .sel    ( sel    ),
        .in0    ( in0    ),
        .in1    ( in1    ),
        .mux_out ( mux_out )
    );

    task expect;
    input [WIDTH-1:0] exp_out;
    if (mux_out !== exp_out) begin
        $display("TEST FAILED");
        $display("At time %0d sel=%b in0=%b in1=%b mux_out=%b",
            $time, sel, in0, in1, mux_out);
        $display("mux_out should be %b", exp_out);
    end
    $finish;
```

```

end
else begin
$display("At time %0d sel=%b in0=%b in1=%b, mux_out=%b",
$time, sel, in0, in1, mux_out);
end
endtask

initial begin
sel=0; in0=5'h15; in1=5'h00; #1 expect (5'h15);
sel=0; in0=5'h0A; in1=5'h00; #1 expect (5'h0A);
sel=1; in0=5'h00; in1=5'h15; #1 expect (5'h15);
sel=1; in0=5'h00; in1=5'h0A; #1 expect (5'h0A);
$display("TEST PASSED");
$finish;
end

initial begin
$dumpfile ("multiplexor_test.vcd");
$dumppvars (0, multiplexor_test);
end

endmodule

```

Verifique o funcionamento do multiplexador através dos seguintes comandos no terminal:

$$\left\{ \begin{array}{l} iverilog multiplexor_test.v \\ vvp a.out \\ gtkwave multiplexor_test.vcd \end{array} \right.$$

Apresente as capturas de tela do terminal e do *gtkwave*.