Sri Adichunchanagiri Shikshana Trust®

SJB INSTITUTE OF TECHNOLOGY

Accredited by NBA & NAAC with 'A' Grade

No. 67, BGS Health & Education City, Dr. Vishnuvardhan Road
Kengeri, Bangalore – 560 060

Department of Electronics & Communication Engineering

VHDL [18EC56]

Question Bank Solution

V SEMESTER – B. E

Academic Year: 2021 – 2022 (ODD)

Course Coordinator :	**Mrs Latha S**
Designation :	**Assistant Professor**

D = x'.y'.z + x'.y.z' + x.y'.z' + x.y.z
B = x'.y + x'.z +y.z
Write the full Verilog description for the full subtractor module, including I/O ports (Remember that + in logic equations corresponds to a logical or operator (||) in dataflow). Instantiate the subtractor inside a stimulus block and test all eight possible combinations of x, y, and z given in the following truth table.

9: Declare a register called oscillate. Initialize it to 0 and make it toggle every 30 time units. Do not use always statement (Hint: Use the forever loop).
10: Design a clock with time period = 40 and a duty cycle of 25% by using the always and initial statements. The value of clock at time = 0 should be initialized to 0.
11: Given below is an initial block with blocking procedural assignments. At what simulation time is each statement executed? What are the intermediate and final values of a, b, c, d?

```
initial
begin
a = 1'b0;
b = #10 1'b1;
c = #5 1'b0;
d = #20 {a, b, c};
end
```

12: Repeat question 10 if nonblocking procedural assignments were used.

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

VERILOG HDL

SOLUTIONS

1)

a) module MEM;

//no interface or internals

endmodule

module SC;

//no interface or internals

endmodule

module Xbar;

//no interface or internals

endmodule

b) module IS;

//instantiate the modules

MEM mem1();

SC sc1();

Xbar xbar1();

endmodule

c) module Top;

//Top level block instantiates module IS

IS is1();

Endmodule

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

2)

a) Module FA;

End module

b) Module Ripple_add;

```
FA fa0();
FA fa1();
FA fa2();
FA fa3();

End module
```

3)

```
module shift_reg(reg_out, reg_in, clock);

        output [3:0] reg_out;
        input [3:0] reg_in;
        input clock;

        //internals not shown

Endmodule
```

4)

The basic components of a module definition are:

a. Module, module name
b. port list
c. Parameters
d. Declarations of wires, regs and other variables
e. Instantiation of lower level modules
f. Data flow statements (assign)
g. always and initial blocks
h. tasks and functions
i. endmodule

All components except keyword module, module name and keyword endmodule are optional.

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

5)

```
module test_bench;

    wire [3:0] reg_out;
    reg [3:0] reg_in;
    reg clock;
    shift_reg reg1(reg_out, reg_in, clock);
    // initial block goes here

Endmodule
```

6)

```
module my_or(out, in1, in2);

output out;
input in1, in2;
wire a, b;

//implementation out = (in1' . in2')'
nand (a, in1, in1);
nand (b, in2, in2);

nand (out, a, b);

endmodule

module my_and(out, in1, in2);
output out;
input in1, in2;

wire a;

//implementation out = ((in1.in2)')'
nand (a, in1, in2);

nand (out, a, a);

endmodule

module my_not(out, in);
output out;
input in;

nand (out, in, in);

endmodule


module stimulus;

wire OUT_OR, OUT_AND, OUT_NOT;
```

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

```
reg IN1, IN2;

initial
        $monitor($time, " OUT_OR = %b, OUT_AND = %b, OUT_NOT = %b, IN1 = %b, IN2 = %b",
                                OUT_OR, OUT_AND, OUT_NOT, IN1, IN2);

initial
begin
        IN1 = 1'B1; IN2 = 1'B1;
        #5 IN1 = 1'B1; IN2 = 1'B0;
        #5 IN1 = 1'B0; IN2 = 1'B1;
        #5 IN1 = 1'B0; IN2 = 1'B0;
end


my_or o1(OUT_OR, IN1, IN2);
my_and a1(OUT_AND, IN1, IN2);
my_not n1(OUT_NOT, IN1);
endmodule
```

7)

```
// Define a 1-bit full adder
module fulladd(sum, c_out, a, b, c_in);

// I/O port declarations
output sum, c_out;
input a, b, c_in;

// Internal nets
wire s1, c1, c2;

// Instantiate logic gate primitives

xor (s1, a, b);
and (c1, a, b);

xor (sum, s1, c_in);
and (c2, s1, c_in);

or (c_out, c2, c1);

endmodule


// Define a 4-bit full adder
module fulladd4(sum, c_out, a, b, c_in);

// I/O port declarations
output [3:0] sum;
output c_out;
```

I/c Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

```
input[3:0] a, b;
input c_in;

// Internal nets
wire c1, c2, c3;

// Instantiate four 1-bit full adders.
fulladd fa0(sum[0], c1, a[0], b[0], c_in);
fulladd fa1(sum[1], c2, a[1], b[1], c1);
fulladd fa2(sum[2], c3, a[2], b[2], c2);
fulladd fa3(sum[3], c_out, a[3], b[3], c3);

endmodule


// Define the stimulus (top level module)
module stimulus;

// Set up variables
reg [3:0] A, B;
reg C_IN;
wire [3:0] SUM;
wire C_OUT;

// Instantiate the 4-bit full adder. call it FA1_4
fulladd4 FA1_4(SUM, C_OUT, A, B, C_IN);


// Setup the monitoring for the signal values
initial
begin
$monitor($time," A= %b, B=%b, C_IN= %b,, C_OUT= %b, SUM= %b\n", A, B, C_IN, C_OUT, SUM);
end

// Stimulate inputs
initial
begin
        A = 4'd0; B = 4'd0; C_IN = 1'b0;

        #50 A = 4'd3; B = 4'd4;

        #50 A = 4'd2; B = 4'd5;

        #50 A = 4'd9; B = 4'd9;

        #50 A = 4'd10; B = 4'd15;

        #50 A = 4'd10; B = 4'd5; C_IN = 1'b1;
end

endmodule
```

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

8)

```
//Full Subtractor Module
module fullsub(B, D, x, y, z);

//Output and Input ports
output B, D;
input x, y, z;

//Compute D and B


        # 10 x = 1; y = 0; z = 0;

        # 10 x = 1; y = 0; z = 1;

        # 10 x = 1; y = 1; z = 0;

        # 10 x = 1; y = 1; z = 1;
end

endmodule
```

IIC

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

9)

```
module dummy;

reg oscillate;

initial
begin
    oscillate = 1'b0;
    forever #30 oscillate = ~oscillate;

end

initial
    #3000 $finish;

Endmodule
```

10)

```
    module dummy;

    reg clock;

    initial
            clock = 1'b0;

    always
    begin
            #30 clock = 1'b1;
            #10 clock = 1'b0;
    end

    initial
            #3000 $finish;

    Endmodule
```

11)

```
module block;

reg a, b, c;
reg [2:0] d;

initial
        $monitor($time," a = %b, b = %b c = %b d = %b",a, b, c, d);

initial
begin
```

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

```
        a = 1'b0;
        b = #10 1'b1;
        c = #5 1'b0;
        d = #20 {a, b, c};
end
endmodule
```

12)

```
module block;

reg a, b, c;
reg [2:0] d;

initial
begin
        $monitor($time," a = %b, b = %b c = %b d = %b",a, b, c, d);
        #100 $finish;
end

initial
begin
        a <= 1'b0;
        b <= #10 1'b1;
        c <= #5 1'b0;
        d <= #20 {a, b, c};
end
endmodule
```

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060