JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -2 EXAMINATION- 2024

M Tech. II Semester (CSE/IT/**ECE**/CE)

COURSE CODE(CREDITS): 21M11EC211 MAX. MARKS: 25

COURSE NAME: Digital System Design Using Verilog HDL

COURSE INSTRUCTORS: DR. HARSH SOHAL MAX. TIME: 1 Hour 30 Minutes

Note: (a)*All questions are compulsory.*

(b)Marks are indicated against each question in square brackets.

(c) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

Q1. [CO1]For given X = 4'b1010, Y = 4'b1101 ; Z = 4'b10x1; evaluate the following expressions: [3]

(i) X & Z (ii) &X

(iii) ^X (iv) X << 2

(v) X && Y (vi) X != Y

Q. 2 [CO3]

(a) What are procedural assignments different from continuous assignments in verilog HDL? Are Blocking and Non Blocking assignments procedural assignments? Differentiate between blocking and non blocking assignments based in its working. [2+1+2]

(b) Give the execution time of each statement in the verilog codes given below (assuming they are working codes): [2+2]

```
                              //CODE 1
reg x, y, z;
reg [15:0] reg_a, reg_b;
integer count;


//All behavioral statements must be inside an initial or always block
initial
begin
        x = 0; y = 1; z = 1; //Scalar assignments
        count = 0; //Assignment to integer variables
        reg_a = 16'b0; reg_b = reg_a; //initialize vectors

        #15 reg_a[2] = 1'b1; //Bit select assignment with delay
        #10 reg_b[15:13] = {x, y, z} //Assign result of concatenation to
                                     // part select of a vector
        count = count + 1; //Assignment to an integer (increment)
end
```

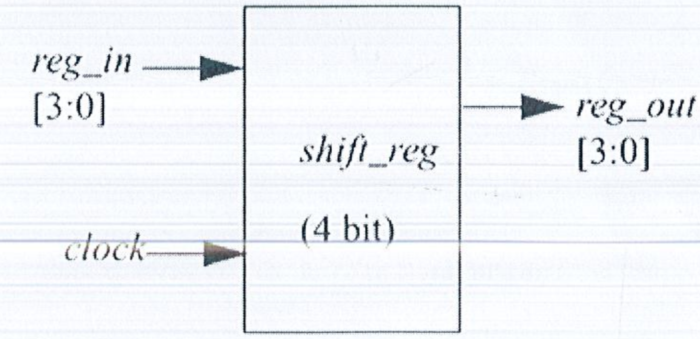
```
                              //CODE 2
reg x, y, z;
reg [15:0] reg_a, reg_b;
integer count;

//All behavioral statements must be inside an initial or always block
initial
begin
        x = 0; y = 1; z = 1; //Scalar assignments
        count = 0; //Assignment to integer variables
        reg_a = 16'b0; reg_b = reg_a; //Initialize vectors

        reg_a[2] <= #15 1'b1; //Bit select assignment with delay
       reg_b[15:13] <= #10 {x, y, z}; //Assign result of concatenation
                                     //to part select of a vector
        count <= count + 1; //Assignment to an integer (increment)
end
```

Q3. [CO2]

(a). What are the basic components of a module? Explain each one of them. Which components are mandatory? [2]

(b) A 4-bit ripple carry adder (Ripple_Add) contains four 1-bit full adders (FA). Define the module FA. Do not define the internals or the terminal list.Define the module Ripple_Add. Do not define the internals or the terminal list. Instantiate four full adders of the type FA in the module Ripple_Add and call them fa0, fa1, fa2, and fa3. (use verilog) [2]

(c) A 4-bit parallel shift register has I/O pins as shown in the figure below. Write the module definition for this module shift_reg. Include the list of ports and port declarations. You do not need to show the internals. [2]



Q4. [CO2]

(a) A full subtractor has three 1-bit inputs x, y, and z (previous borrow) and two 1-bit outputs D (difference) and B (borrow). The logic equations for D and B are as follows:$D = x'.y'.z + x'.y.z' + x.y'.z' + x.y.z$ $B = x'.y + x'.z + y.z$

Write the full Verilog description using dataflow style of modeling, for the full subtractor module, including I/O ports(Remember that + in logic equations corresponds to a logical or operator (||) in dataflow). [2]

(b) Draw the logic circuit implemented by the following verilog code: [2]

```
module Question3_b (out, i0, i1, i2, i3, s1, s0);

// Port declarations from the I/O diagram
output out;
input i0, i1, i2, i3;
input s1, s0;

// Use nested conditional operator
assign out = s1 ? ( s0 ? i3 : i2) : (s0 ? i1 : i0) ;

endmodule
```

(c)
In verilog statement c3 = g2 | p2 & g1 | p2 & p1 & g0 | p2 & p1 & p0 & c_in which operator has high precedence over the other? [1]

Q5. [CO1]

What is an SoC? Can we design and test an SOC solution on FPGA platform? Discuss.[2]