U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

April 2025 Semester End Make-Up Examinations

Programme: B.E. **Semester: III**
Branch: Electronics and Communication Engineering **Duration: 3 hrs.**
Course Code: 23EC3ESHDL **Max Marks: 100**
Course: HDL Programming

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT - I	*CO*	*PO*	**Marks**
1	a)	Explain the design flow of VLSI IC circuits.	-	-	**10**
	b)	Differentiate between the system tasks (i) $monitor and $display (ii) $finish and $stop	-	-	**5**

	c)	Analyze the code snippet to identify the syntax errors and rectify the same. Also identify and explain the method of port-mapping used in module instantiation.
		`module stimulus;`
		`wire a1, b1;`
		`reg c1;`
		`example1 inst(a1, b1, c1);`
		`initial`
		`begin`
		`a = 1’b1; b=1’b0;`
		`#5 $finish;`
		`end`
		`endmodule`
		
		`module example1(input a, b, output reg c);`
		`and g1(c, a, b);`
		`endmodule`
			CO 2	*PO 2*	**5**

		OR			
2	a)	Explain the components of a Verilog module. Also elaborate the port connection rules used in Verilog.	-	-	**10**
	b)	Explain different levels of abstraction in Verilog.	-	-	**5**

B ODD SEM 2024-25

c) In the Verilog code snippet given below, identify all the keywords and the data types of all the variables with their bit-width. — *CO 2* — *PO 2* — **5**

```
module ex2 (input [1:0] a, b, output [7:0] c);
reg r1;
reg [7:0] mem [0:1023];
..................
endmodule
```

UNIT - II

3 a) Develop a Verilog code for a 4-bit carry lookahead adder using continuous assignment statements. — *CO 3* — *PO 3* — **8**

b) Deduce the design block for the given Verilog code and draw the waveforms for the stimulus given: — *CO 2* — *PO 2* — **7**

```
module ex3(output out, input a, b, c);
wire e;
and #5 a1(e, a, b);
or #4 a2(out, e, c);
endmodule
```

```
module stimulus;
reg A, B, C;
wire OUT;
ex3 example(OUT, A, B, C);
initial
begin
A = 1’b0, B = 1’b0, C = 1’b0;
#10 A = 1’b1, B = 1’b1, C = 1’b1;
#10 A = 1’b1, B = 1’b0, C = 1’b0;
#10 $finish;
end
endmodule
```

c) Find the value of y, if y = ((A+B) && (|C)) + (D<<2); and y is declared as a 4-bit reg, given A = 1101, B = 1010, C = 0111 and D = 0101. — *CO 1* — *PO 1* — **5**

OR

4 a) Develop a 3:8 decoder Verilog module instantiating 2:4decoders. Implement the 2:4 decoder using gate-level modelling. — *CO 3* — *PO 3* — **8**

b) Analyze the below given Verilog code, find the values s1 and s2 as per the timing units given in table. — *CO 2* — *PO 2* — **7**

```
module ex4(input a, b, output s1, s2);
assign #10 s1 = a ˆ b;
assign #10 s2 = a | s1;
endmodule
```

	T = 100	T = 150	T = 165	T = 200	T = 250	T = 300
a	1	0	0	1	0	1
b	1	1	1	0	0	1
s1	0					
s2	0					

B.M.S.C.E. M 2024-25

<table>
<tr><td></td><td></td><td>c)</td><td>Implement a 4x1 multiplexer using conditional operators. Draw the mux tree for reference.</td><td><i>CO 1</i></td><td><i>PO 1</i></td><td><b>5</b></td></tr>
<tr><td></td><td></td><td></td><td><b>UNIT - III</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td>a)</td><td>Describe the behavior of the <b>SHIFT REGISTER</b> given below using Verilog. Draw the schematic for reference.
<table>
<tr><th colspan="2">Mode Control</th><th rowspan="2">Register Operation</th></tr>
<tr><th>S1</th><th>S0</th></tr>
<tr><td>0</td><td>0</td><td>No change</td></tr>
<tr><td>0</td><td>1</td><td>Shift Right</td></tr>
<tr><td>1</td><td>0</td><td>Shift Left</td></tr>
<tr><td>1</td><td>1</td><td>Parallel load</td></tr>
</table></td><td><i>CO 1</i></td><td><i>PO 1</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Design a 4-bit priority encoder with LSB having the highest priority. Describe its behavior using behavioral descripting in Verilog HDL. Write the testbench to test the designed encoder.</td><td><i>CO 3</i></td><td><i>PO 3</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td></td><td><b>OR</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>6</td><td>a)</td><td>Using blocking and non-blocking statements, develop two Verilog modules respectively, to swap the contents of two registers:
• with a temporary register
• without a temporary register</td><td><i>CO 1</i></td><td><i>PO 1</i></td><td><b>5</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Use “forever” construct to generate a clock with time-period=40ns and a duty cycle of 15%, with initial value ‘0’.</td><td><i>CO 1</i></td><td><i>PO 1</i></td><td><b>5</b></td></tr>
<tr><td></td><td></td><td>c)</td><td>Describe the behavior of T flip-flop in Verilog. Use “generate loops” and instantiate the T flip-flops to implement a 3-bit asynchronous up-counter.</td><td><i>CO 3</i></td><td><i>PO 3</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td></td><td><b>UNIT - IV</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>7</td><td>a)</td><td>With relevant examples, explain how design partitioning can affect the output of the logic synthesis tool.</td><td>-</td><td>-</td><td><b>8</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Analyze the below given code, and draw the expected logic circuit after logic synthesis (Assume the logic unit to be a mux). Give an alternate Verilog code which can produce the same output.
<pre>
module ex5 (input [3:0] a, b, c, d, input
[1:0] s, output [3:0] out);
     assign out = s[1]?(s[0]?d:c):(s[0]?b:a);
endmodule
</pre></td><td><i>CO 2</i></td><td><i>PO 2</i></td><td><b>8</b></td></tr>
<tr><td></td><td></td><td>c)</td><td>Analyze the Verilog code snippet given below and indicate the hardware realized by standard Synthesis tools. Suggest a preferred coding alternative.
<pre>
module ex6 (input ct, a, output reg out);
always @ (ct or a)
      if (ct)
       out = a;
endmodule
</pre></td><td><i>CO 2</i></td><td><i>PO 2</i></td><td><b>4</b></td></tr>
</table>

B.M.S.C.E. - ODD SEM 2024-25

<table>
<tr><td></td><td></td><td></td><td><b>OR</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>8</td><td>a)</td><td>With flow diagram, explain the RTL to gate level logic synthesis flow.</td><td>-</td><td>-</td><td><b>8</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Analyze the below given code, complete the code (ex7) and draw the expected logic circuit after logic synthesis.
<b>module ex7 (clk, en, a, b, c, d, y);</b>
....
....
<b>always @ (posedge clk)</b>
<b>y = !(en & (a | b) & (c | d));</b>
<b>endmodule</b>
Compare the Verilog code snippets of “ex7” and “ex8” and draw the inferred logic circuit for “ex8”.
<b>module ex8 (en, a, b, c, d, out);</b>
....
....
<b>y = !(en & (a | b) & (c | d));</b>
<b>endmodule</b></td><td><i><b>CO 2</b></i></td><td><i><b>PO 2</b></i></td><td><b>8</b></td></tr>
<tr><td></td><td></td><td>c)</td><td>Analyze the below given code, and draw the expected logic circuit after logic synthesis.
<b>module ex9 (input sin, clk, output [3:0] q);</b>
<b>always @(posedge clk)</b>
<b>begin</b>
<b>q[0] <= sin;</b>
<b>q[1] <= q[0];</b>
<b>q[2] <= q[1];</b>
<b>q[3] <= q[2];</b>
<b>end</b>
<b>endmodule</b></td><td><i><b>CO 2</b></i></td><td><i><b>PO 2</b></i></td><td><b>4</b></td></tr>
<tr><td></td><td></td><td></td><td><b>UNIT - V</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>9</td><td>a)</td><td>Design a Verilog Mealy-type BCD to Ex-3 code converter.</td><td><i><b>CO 3</b></i></td><td><i><b>PO 3</b></i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>With a neat block diagram, explain Mealy and Moore model in sequential circuit design. Highlight the difference between Mealy and Moore FSM considering the state diagram for a sequence detector to detect a sequence “011” in a stream of binary data.</td><td>-</td><td>-</td><td><b>10</b></td></tr>
<tr><td></td><td></td><td></td><td><b>OR</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>10</td><td>a)</td><td>Develop a Moore FSM using Verilog to detect overlapped sequence of “0101” in a stream of 1’s and 0’s. Also write the test bench to test the design.</td><td><i><b>CO 3</b></i></td><td><i><b>PO3</b></i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Explain the architecture of FPGA with the help of block diagram.</td><td>-</td><td>-</td><td><b>10</b></td></tr>
</table>

B.M.S.C. SEM 24-25

******