U.S.N.										

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

Autonomous Institute Affiliated to VTU

January / February 2025 Semester End Main Examinations

Programme: B.E. **Semester: III**
Branch: Electronics and Communication Engineering **Duration: 3 hrs.**
Course Code: 22EC3PCDSD **Max Marks: 100**
Course: Digital System Design

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

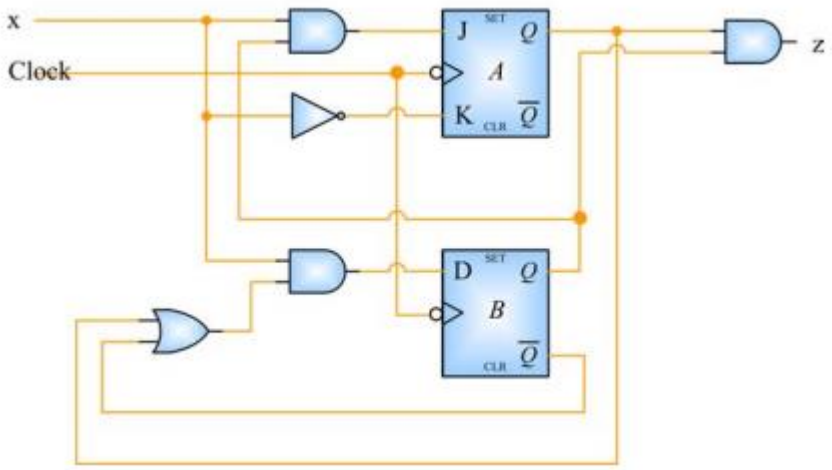
Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT – I	CO	PO	Marks
1	a)	Simplify the expression below using K-map where the output is a 1 for the BCD inputs 7,8,9 and draw the logic diagram for the same. $F(A,B,C,D) = \sum m(0, 1, 2, 8, 9, 12, 13) + d(10, 11, 14, 15)$ Also provide the expression when the don't care terms above are NOT considered.	*CO* 1	*PO1*	**12**
	b)	Compare the various types of modeling provided by VERILOG HDL.	*CO* 1	*PO1*	**8**
		OR			
2	a)	Discuss the components of Verilog module with block diagram and example.	*CO* 1	*PO1*	**8**
	b)	Design a circuit that accepts a 3-bit unsigned number X (X2X1X0) and generates an output binary number Y = 3X + 1 a. Write out a truth table b. Find the minimal POS equations for each output bit c. Show the implementation in OR-AND	*CO* 3	*PO3*	**12**
		UNIT – II			
3	a)	Implement a 32 x 1 Mux using 8 x 1 Mux and 4 x1 Mux.	*CO* 1	*PO1*	**6**
	b)	Implement The following functions using PLA: (i) F1 (A, B, C) = (3, 5, 6, 7) and (ii) F2 (A, B, C) = (0, 2, 4, 7).	*CO* 1	*PO1*	**10**
	c)	Implement 4 to 16 line decoder using 2:4 Line decoder.	*CO* 1	*PO1*	**4**
		OR			
4	a)	Design a 2-bit magnitude comparator. Write the logic diagram and all the logic expressions.	*CO* 3	*PO3*	**10**
	b)	Implement The following functions using PLA: (i) F1 (A, B, C) = (0,1, 3, 4) and (ii) F2 (A, B, C) = (1,2,3,4,5).	*CO* 1	*PO1*	**10**

<table>
<tr><td></td><td></td><td></td><td colspan="3">UNIT - III</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>a)</td><td>Use “initial” and “always” blocks to generate a clock with time period = 40 units and a duty cycle of 15%, with initial value 0. Use the generated clock to draw the expected waveform of a positive-edge triggered D flip-flop with the following “D” input:
<table>
<tr><td>Time</td><td>0</td><td>20</td><td>40</td><td>60</td><td>80</td><td>100</td></tr>
<tr><td>D</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
</table></td><td>CO 3</td><td>PO3</td><td>10</td></tr>
<tr><td></td><td>b)</td><td>Implement NAND structure of JK latch using Verilog.</td><td>CO 1</td><td>PO1</td><td>10</td></tr>
<tr><td></td><td></td><td>OR</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>a)</td><td>Design a priority encoder according to the table below. Describe its behavior using behavioral descripting in Verilog HDL.
<table>
<tr><td colspan="5">Inputs</td><td colspan="2">Outputs</td></tr>
<tr><td>Reset(R)</td><td>Enable(E)</td><td>A</td><td>B</td><td>C</td><td>Y1</td><td>Y0</td></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>X</td><td>Y1</td><td>Y0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>X</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Y1</td><td>Y0</td></tr>
</table></td><td>CO 3</td><td>PO3</td><td>10</td></tr>
<tr><td></td><td>b)</td><td>Analyze CODE1 and CODE2 modules, find the common error in both these codes and if that’s corrected what values do we get on y1 and y2 in both the codes.

<pre>
module CODE1(input a,b,output y1,y2);
initial
begin
y1=a;
y2=b;
end
always@(*)
begin
y1=y2;
y2=y1;
end
endmodule

module CODE2(input a,b,output y1,y2);
initial
begin
y1=a;
y2=b;
end
always@(*)
begin
y1<=y2;
y2<=y1;
end
endmodule
</pre></td><td>CO 2</td><td>PO2</td><td>10</td></tr>
<tr><td></td><td></td><td>UNIT – IV</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>a)</td><td>Analyze the problem with a simple JK Flip flop and provide a solution to overcome this problem. Use a Timing diagram to explain your solution.</td><td>CO 2</td><td>PO2</td><td>10</td></tr>
</table>

		b)	Convert i) SR Flip Flop to T Flip Flop ii) JK Flip Flop to T Flip flop	*CO* 1	*PO1*	**10**
			OR			
	8	a)	With a neat schematic diagram, Analyze the operation and working of a 4-bit UNIVERSAL shift register. Sketch the data shifting operation using a Timing diagram	*CO* 2	*PO2*	**10**
		b)	Design a SYNCHRONOUS counter using T flip flops for the following COUNT Sequence: 000 → 001 → 011 → 100 → 101 → 111 → 000.	*CO* 3	*PO3*	**10**
			UNIT – V			
	9	a)	Design a 3-bit Moore sequence detector to detect the sequence 101 with Overlapping using T flip-flops.	*CO* 3	*PO3*	**10**
		b)	Design a Non-Overlapping Mealy Sequence Detector with Verilog Code for the sequence "1010".	*CO* 3	*PO3*	**10**
			OR			
	10	a)	Analyse the below given sequential logic circuit and write the sate diagram. 	*CO* 2	*PO2*	**10**
		b)	Design a Overlapping Mealy Sequence Detector with Verilog Code for the sequence "11011".	*CO* 3	*PO3*	**10**

****