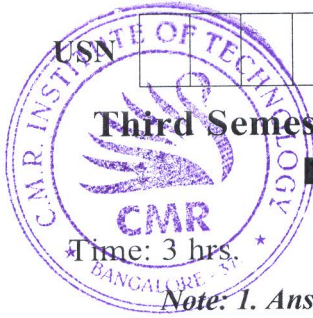


# CBCS SCHEME

BEC302



**Third Semester B.E./B.Tech. Degree Examination, Dec.2023/Jan.2024**  
**Digital System Design Using Verilog**

Max. Marks: 100

- Note:** 1. Answer any FIVE full questions, choosing ONE full question from each module.  
 2. M : Marks , L: Bloom's level , C: Course outcomes.

Module – 1			M	L	C
<b>Q.1</b>	a.	Develop a truth table of logic which takes two , 2 – bit binary numbers as its input and generate on output equal to 1, when the sum of the two numbers is odd.	7	L1	CO1
	b.	Convert the following Boolean function into : i) $f(abc) = (\bar{a} + b) (b + \bar{c})$ – Min term canonical form. ii) $f(xyz) = x + \bar{x} \bar{z} (y + \bar{z})$ – Max term canonical form.	7	L1	CO1
	c.	List the difference between Prime implicant and Essential prime implicant.	6	L1	CO1
<b>OR</b>					
<b>Q.2</b>	a.	Simplify the given Boolean function using Quine Mc Cluskey Minimization Technique for the function $R = f(abcd) = \Sigma(0, 1, 2, 6, 7, 9, 10, 12) + dc(3, 5).$	10	L1	CO1
	b.	Find the minimal sum and minimal product for the given function using K – map method for the function $R = f(abcd) = \Sigma m(0, 1, 3, 7, 8, 12) + dc(5, 10, 13, 14).$	10	L1	CO1
<b>Module – 2</b>					
<b>Q.3</b>	a.	List the difference between decoder and encoder and implement full adder using IC – 74138.	10	L3	CO2
	b.	What is Comparator? Design a 2 – bit digital comparator.	10	L3	CO2
<b>OR</b>					
<b>Q.4</b>	a.	Realize the Boolean function $P = f(wxyz) = \Sigma(0, 1, 5, 6, 7, 10, 15)$ using i) 8 : 1 MUX      ii) 4 : 1 MUX.	10	L2	CO2
	b.	With neat logic diagram, explain carry ahead adder.	10	L2	CO2
<b>Module – 3</b>					
<b>Q.5</b>	a.	Explain the working of master slave JK flip flop with help of Logic diagram , Function table , Logic symbol and Timing diagram.	10	L1	CO3
	b.	Obtain the characteristic equation for : i) SR flip - flop    ii) J – K – flip - flop    iii) D – flip - flop iv) T – flip - flop.	10	L2	CO3
<b>OR</b>					

Q.6	a.	Design a Synchronous 3 – bit up counter using J K – flip - flop.	10	L4	CO3															
	b.	Design a 4 – bit universal shift register using positive edge triggered D – flip - flop and 4 : 1 MUX , to operate as shown in table below : <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>S<sub>1</sub></th> <th>S<sub>0</sub></th> <th>Register Operation</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>- Hold</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 operation</td> </tr> </tbody> </table>	S <sub>1</sub>	S <sub>0</sub>	Register Operation	0	0	- Hold	0	1	- Shift right	1	0	- Shift left	1	1	- Parallel load operation	10	L4	CO3
S <sub>1</sub>	S <sub>0</sub>	Register Operation																		
0	0	- Hold																		
0	1	- Shift right																		
1	0	- Shift left																		
1	1	- Parallel load operation																		
<b>Module – 4</b>																				
Q.7	a.	Illustrate the structure and verilog module and write a verilog code for Half – adder using structural model.	10	L3	CO4															
	b.	What are different types of operators used in HDL with example?	10	L2	CO4															
<b>OR</b>																				
Q.8	a.	Illustrate the structure of Data flow description with example.	10	L3	CO4															
	b.	Write the syntax of conditional signal assignment statement. Write a code for 4 : 1 MUX using conditional signal statement.	10	L2	CO4															
<b>Module – 5</b>																				
Q.9	a.	Write the structure of Verilog behavioral description.	6	L2	CO4															
	b.	Write the syntax of IF statement with example.	7	L2	CO4															
	c.	Write a code for D – Latch using Behavioral description.	7	L2	CO4															
<b>OR</b>																				
Q.10	a.	Write the syntax of While loop statement with example.	10	L2	CO4															
	b.	Write a verilog code of a 3 – bit ripple carry adder using Structural description method.	10	L2	CO4															

CMRIT LIBRARY  
BANGALORE - 560 037

\* \* \* \* \*