U.S.N.										

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

Autonomous Institute Affiliated to VTU

September / October 2023 Supplementary Examinations

Programme: B.E. **Semester: VI**
Branch: Electronics and Communication Engineering **Duration: 3 hrs.**
Course Code: 19EC6PE3SV **Max Marks: 100**
Course: System Verilog and Verification **Date: 20.09.2023**

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

UNIT - I

1	a)	Discuss Block coverage, Conditional coverage and Toggle coverage with suitable example	**08**
	b)	Design a T-FF with asynchronous active high reset and synchronous active low set. Write a test bench with interface block and modports to test the design.	**12**

UNIT - II

2	a)	Distinguish between Program Block and Module. Where do you use program block?	**08**
	b)	Design a divide by 8 synchronous up down counter and test the design using clocking block and modports.	**12**

UNIT - III

3	a)	What are Virtual Interfaces? Explain with suitable example.	**06**
	b)	Design a 4-bit ALU which performs arithmetic and logical operations based on a 3 bit opcode. Write a test bench using transactor class to randomize the inputs and driver class to drive the inputs to the design using virtual interface.	**14**

OR

4	a)	Discuss Inheritance and polymorphism with suitable examples	**10**
	b)	Design a JK Flip Flop and write a test bench using transactor class to randomize the inputs and driver class to drive the inputs to the design using virtual interface.	**10**

UNIT - IV

5	a)	Distinguish between immediate and concurrent assertions with suitable examples	**10**
	b)	Discuss Sequence replication operators with examples.	**10**

UNIT - V

6	a)	Discuss Implicit and explicit bins with examples.	**10**
	b)	With a neat diagram, discuss the System Verilog layered test bench architecture.	**10**

OR

7	a)	Develop a Mealy FSM that detects "0100" sequence with overlap.	**08**
	b)	Test the RTL in 7.a) by developing the following i) Driver Class for defining functional covergroups and coverpoints, Randomizing, driving and sampling the coverage . ii) Transactor class to define the constraints iii) Program block, Interface and Top module	**12**

******