EC-355	HDL PROGRAMING LAB	L	T	P	C
		-	-	3	2

COURSE OBJECTIVES:

1. To understand the design process of Logic gates / Multiplexers / ALU
2. To understand the design process of Flip flops, Synchronous & Asynchronous Counters
3. To understand the design process of State machines

COURSE OUTCOMES:

After successful completion of the course, the students are able to

1. understand basic logic gates, combinational circuits, sequential circuits, counters and state machines.
2. design combinational and sequential circuits using Verilog.
3. solve mealy and Moore state machines, signed multiplier, shift registers and Traffic light controller using Verilog..
4. analyze simulation results for digital logic circuits..

List of Experiments:

Verilog Modeling and Synthesis of the following Experiments

1. Write a Verilog code for the design of Logic Gates.
2. Write a Verilog code for Combinational Circuits
3. Write a Verilog code for JK, D, T, and SR flip-flops with preset and clear inputs.
4. Write a Verilog code for Parity Generator and Magnitude Comparator.
5. Write a Verilog code for Carry skip Adder.
6. Write a Verilog code for 8-bit Array Multiplication/ Booth Multiplication.
7. Design of 4-bit Binary, BCD counters (synchronous/ asynchronous reset) using Verilog.
8. Design of an N- bit shift register of Serial in Serial out, Serial in parallel out, Parallel in Serial out and Parallel in Parallel out using Verilog.
9. Write a Verilog Code for 3-bit Arbitrary Counter to generate 0,1,2,3,6,5,7 and repeats.
10. Design Mealy and Moore Sequence Detector using Verilog to detect Sequence.Example: 11101 (with and without overlap) any sequence can be specified.
11. Write a Verilog code for an ALU to Perform - ADD, SUB, AND, OR, 1's compliment, 2's Compliment, Multiplication and Division.
12. Write a Verilog code for Traffic Light Controller.
13. Write a Verilog code for Seven Segment Display Interface.
14. Construct an 8-bit dedicated control unit to generate and add the numbers from n down to 1, where 'n' is an 8-bit user input number using Verilog.
15. Construct an 8-bit general data path to generate and add the numbers from n down to 1, where 'n' is an 8-bit user input number using Verilog.

Note: A minimum of 10(Ten) experiments have to be Performed and recorded by the candidate to attain eligibility for Semester End Practical Examination.