CSL 202	DIGITAL LAB	CATEGORY	L	T	P	CREDIT
		PCC	0	0	3	2

Preamble: This course helps the learners to get familiarized with (i) Digital Logic Design through the implementation of Logic Circuits using ICs of basic logic gates & flip-flops and (ii) Hardware Description Language based Digital Design. This course helps the learners to design and implement hardware systems in areas such as games, music, digital filters, wireless communications and graphical displays.

Prerequisite: Topics covered under the course Logic System Design **(CST 203)**

Course Outcomes: After the completion of the course the student will be able to

CO 1	Design and implement combinational logic circuits using Logic Gates (Cognitive Knowledge Level: **Apply**)
CO 2	Design and implement sequential logic circuits using Integrated Circuits (Cognitive Knowledge Level: **Apply**)
CO 3	Simulate functioning of digital circuits using programs written in a Hardware Description Language (Cognitive Knowledge Level: **Apply**)
CO 4	Function effectively as an individual and in a team to accomplish a given task of designing and implementing digital circuits (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
CO 1	✓	✓	✓	✓				✓				✓
CO 2	✓	✓	✓	✓				✓				✓
CO 3	✓	✓	✓	✓	✓			✓				✓
CO 4	✓	✓	✓	✓				✓	✓			✓

Assessment Pattern

Bloom's Category	Continuous Assessment Test (Internal Exam) (Percentage)	End Semester Examination (Percentage)
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 15 marks

Continuous Evaluation in Lab : 30 marks

Continuous Assessment Test : 15 marks

Viva-voce : 15 marks

Internal Examination Pattern: The marks will be distributed as Design/Algorithm 30 marks, Implementation/Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Design/Algorithm 30 marks, Implementation/Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Fair Lab Record:

All Students attending the Digital Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record, the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, and Aim of Experiment. The left hand page should contain components used, circuit design or a print out of the code used for the experiment and sample output obtained.

SYLLABUS

Conduct a minimum of **8** experiments from **Part A** and a minimum of **4** experiments from **Part B**. The starred experiments in Part A are mandatory. The lab work should be conducted in groups (maximum group size being 4). The performance of a student in the group should be assessed based on teamwork, integrity and cooperation.

Part A (Any 8 Experiments)

- A 2 hour session should be spent to make the students comfortable with the use of trainer kit/breadboard and ICs.
- The following experiments can be conducted on breadboard or trainer kits.
- Out of the 15 experiments listed below, a minimum of 8 experiments should be completed by a student, including the mandatory experiments (5).

1. Realization of functions using basic and universal gates (SOP and POS forms).
2. Design and realization of half adder, full adder, half subtractor and full subtractor using: a) basic gates (b) universal gates. *
3. Code converters: Design and implement BCD to Excess 3 and Binary to Gray code converters.
4. Design and implement 4 bit adder/subtractor circuit and BCD adder using IC7483.
5. Implementation of Flip Flops: SR, D, T, JK and Master Slave JK Flip Flops using basic gates.*
6. Asynchronous Counter: Design and implement 3 bit up/down counter.
7. Asynchronous Counter: Realization of Mod N counters (At least one up counter and one down counter to be implemented). *
8. Synchronous Counter: Realization of 4-bit up/down counter.
9. Synchronous Counter: Realization of Mod-N counters and sequence generators. (At least one mod N counter and one sequence generator to be implemented) *
10. Realization of Shift Register (Serial input left/right shift register), Ring counter and Johnson Counter using flipflops. *
11. Realization of counters using IC's (7490, 7492, 7493).
12. Design and implement BCD to Seven Segment Decoder.
13. Realization of Multiplexers and De-multiplexers using gates.
14. Realization of combinational circuits using MUX & DEMUX ICs (74150, 74154).
15. To design and set up a 2-bit magnitude comparator using basic gates.

PART B (Any 4 Experiments)

- The following experiments aim at training the students in digital circuit design with ***Verilog***. The experiments will lay a foundation for digital design with Hardware Description Languages.
- A 3 hour introductory session shall be spent to make the students aware of the fundamentals of development using Verilog
- Out of the 8 experiments listed below, a minimum of 4 experiments should be completed by a student

Experiment 1. Realization of Logic Gates and Familiarization of Verilog

(a) Familiarization of the basic syntax of Verilog
(b) Development of Verilog modules for basic gates and to verify truth tables.
(c) Design and simulate the HDL code to realize three and four variable Boolean functions

Experiment 2: Half adder and full adder

(a) Development of Verilog modules for half adder in 3 modeling styles (dataflow/ structural/behavioural).
(b) Development of Verilog modules for full adder in structural modeling using half adder.

Experiment 3: Design of code converters

Design and simulate the HDL code for

(a) 4- bit binary to gray code converter
(b) 4- bit gray to binary code converter

Experiment 4: Mux and Demux in Verilog

(a) Development of Verilog modules for a 4x1 MUX.
(b) Development of Verilog modules for a 1x4 DEMUX.

Experiment 5: Adder/Subtractor

(a) Write the Verilog modules for a 4-bit adder/subtractor
(b) Development of Verilog modules for a BCD adder

Experiment 6: Magnitude Comparator

Development of Verilog modules for a 4 bit magnitude comparator

Experiment 7: Flipflops and shiftregisters

(a) Development of Verilog modules for SR, JK, T and D flip flops.
(b) Development of Verilog modules for a Johnson/Ring counter

Experiment 8: Counters

(a) Development of Verilog modules for an asynchronous decade counter.
(b) Development of Verilog modules for a 3 bit synchronous up-down counter.

Practice Questions

PART A

1. Design a two bit parallel adder using gates and implement it using ICs of basic gates
2. A combinatorial circuit has 4 inputs and one output. The output is equal to 1 when (a) all inputs are 1, (b) none of the inputs are 1, (c) an odd number of inputs are equal to 1. Obtain the truth table and output function for this circuit and implement the same.
3. Design and implement a parallel subtractor.
4. Design and implement a digital circuit that converts Gray code to Binary.
5. Design a combinational logic circuit that will output the 1's compliment of a 4-bit input number.
6. Implement and test the logic function $f(A, B, C) = \sum m(0,1,3,6)$ using an 8:1 MUX IC
7. Design a circuit that will work as a ring counter or a Johnson counter based on a mode bit, M.
8. Design a 4-bit synchronous down counter.
9. Design a Counter to generate the binary sequence 0,1,3,7,6,4
10. Design an asynchronous mod 10 down counter
11. Design and implement a synchronous counter using JK flip flop ICs to generate the sequence: 0 - 1 -3 - 5 - 7 - 0.

PART B

1. Develop Verilog modules for a full subtractor in structural modeling using half subtractors.
2. Design a 4 bit parallel adder using Verilog.
3. Develop Verilog modules for a 4 bit synchronous down counter.
4. Write Verilog code for implementing a 8:1 multiplexer.
5. Develop Verilog modules for a circuit that converts Excess 3 code to binary.
6. Write the Verilog code for a JK Flip flop, and its test-bench. Use all possible combinations of inputs to test its working
7. Write the hardware description in Verilog of a 8-bit register with shift left and shift right modes of operations and test its functioning.
8. Write the hardware description in Verilog of a mod-N (N > 9) counter and test it.