**Lecture Plan**

**L-T-P 3-0-2 Digital Logic Design (CEECC-03)**

**Course Outcomes (CO):** After the complication the course, the students should be able to:

CO 1: Get familiarized with number systems, codes, logic gates and Boolean algebra

CO 2: Develop capability of modeling of the basic digital circuits using VHDL/Verilog

CO 3: Synthesize the design process associated with combinational and sequential circuits

CO 4: Analyze the basic characteristics of various logic families

CO 5: Develop basic understanding of programmable logic devices

|  |  |  |
| --- | --- | --- |
| **Unit No.** | **Course Content** | **Lecture No.** |
| **1** | Introduction, Digital design representations (e.g., truth table, equations, schematic, etc.). | **1** |
| Number representation and conversion | **2** |
| Binary addition and subtraction (unsigned and one’s and two's complement). | **3** |
| Different codes and their properties (e.g., Gray, BCD, 1-Hot etc.), | **4** |
| Boolean equations and terminology (SOP, POS, minterm, maxterm) | **5** |
| Logic implementations of equations (2-level, AND/OR, NAND/NAND, etc.) | **6** |
| **2** | Boolean algebra axioms and single variable theorems, | **7** |
| Karnaugh Maps up to 4-variables with don’t care conditions. | **8** |
| Karnaugh Maps of 5-variables with don’t care conditions. | **9** |
| Parallel adders, BCD adder | **10** |
| **CLASS TEST 1** |  |
| Multiplexers | **11** |
| Encoders, Decoders, Priority encoders | **12** |
| Comparators, Code converters: BCD to Seven Segment Display etc. | **13** |
| Logic implementation using Programmable Logic Devices: PROM and PAL. | **14** |
| Logic implementation using using PLA | **15** |
| Hardware Language(VHDL/Verilog ) Introduction and its types of modeling style | **16** |
| VHDL/ Verilog implementation of adder, subtractor | **17** |
| Verilog implementation of multiplexer, decoders, comparators etc.. | **18** |
| **3** | Latch versus flip-flop, SR and D latch | **19** |
| D, SR, JK, and T flip-flops | **20** |
| Registers, Shift registers | **21** |
| **MID SEMESTER EXAM** |  |
| Counters, | **22** |
| Design & Analysis, Problem description to state table/graph | **23** |
| State assignment (binary, Gray, one-hot), Next state and output equations | **24** |
| Verilog/VHDL Implementation of Flip Flops, latches | **25** |
| Verilog/VHDL Implementation of Registers, Counters | **26** |
| **4** | Transistor-level schematics of CMOS logic gates(NOT,NAND,NOR) | **27** |
| Transistor-level schematics of any logical expression | **28** |
| Properties and characteristics (logic thresholds, delay, Noise margin, fan-inand fan-out, power dissipation) of logic families – CMOS, Figure of Merit | **29** |
| Properties and characteristics (logic thresholds, delay, Noise margin, fan-inand fan-out, power dissipation) of logic families – TTL, Figure of Merit | **30** |
|  | **CLASS TEST 2** |  |
| **5** | FSM Implementation: Mealy. | **31** |
| FSM Implementation: Moore | **32** |
| FSM timing and input/output relationships (i.e., Moore versus Mealy outputs) | **33** |
| FSM + Data path | **34** |
| Introduction to macro-cells, CPLDs and FPGA | **35** |

**LIST OF EXPERIMENTS**

1. Characterization of Logic Family (Use 74HC/LS/HCT04 on a breadboard)
   1. Find out logic threshold values and noise margins.
   2. Propagation Delay time measurement of inverter using ring oscillator with 3, 5, 7, 9 and 11 inverter elements.
2. Design, Implement and verify a: half-adder, (two numbers of 2-bits each) full-adder, half-subtractor, (two numbers of 2-bits each) Full-subtractor using Xilinx Design Suite (Schematic capture) and verify on a CPLD/FPGA Board
3. Design, implement and verify a 2-bit binary comparator using basic gates. Use unsigned binary numbers as inputs. The comparator should have three outputs – Equal, Greater than and Less than.
4. Design, Implement and verify a 3-8 decoder and 3-8 multiplexor using basic gates.
5. Design, implement and verify a circuit to implement any combinational circuit with 4-inputs using a 4-16 multiplexor library symbol in the Xilinx design suite library.
6. Design, implement and verify a using a 4-16 multiplexer: a BCD to Seven Segment Decoder and a Binary to Seven Segment decoder.
7. Design, implement and verify a 4-bit binary counter and a decimal counter using basic gates and DFFs. Demonstrate the output on LEDs.
8. Design, implement and verify using FSM, a D type flip-flop with an enable signal. If enable is ‘1’, the flip flop captures the input signal, if enable is ‘0’, the flip flop retains the previous value.
9. Design, implement and verify, using FSM approach, a 3-bit up-counter with an enable signal. If enable is ‘1’, the counter counts normally. If enable is ‘0’, the counter retains the last value.
10. Implement an electronic dice using FSM approach. The dice should display random numbers between 1 and 6. The input to the dice is an input ‘throw dice’ signal, apart from the clock.

Note: Experiments 2-10 will be implemented on a CPLD/FPGA board.

**Books:**

1. Digital Design and Computer Architecture. David Harris and Sarah Harris (Morgan Kaufman).
2. Digital principles & design : Donald D. Givone, McGraw Hill
3. Digital logic and computer design : M. M. Mano, PHI
4. Charles Roth,” Digital Systems Design using VHDL”, 5th Edition Cengage Learning
5. R.P. Jain, `Modern Digital Electronics`, McGraw Hill education, 4th Edition