IC Chip Design and Verification:

Course Objectives	• Gain Proficiency on basic digital electronics including binary systems, logic gates, and Boolean algebra. • Master digital circuit design using HDLs like Verilog for behavioural, data flow, and structural modelling. • Explore advanced concepts like pipelining, state machines, clock domains, and digital verification. • Gain hands-on experience through practical exercises, lab sessions, and project work. • Develop analytical skills to optimize circuit performance and solve digital design challenges.
Course Outcomes	• Apply essential design techniques to electronic circuit engines. • Perform accurate simulations of electronic circuits, ensuring functionality and performance. • Develop skills in the verification of electronic circuits, ensuring designs meet specified requirements. • Apply Static Timing Analysis (STA) techniques for both pre-layout and post-layout timing closure of digital designs. • Integrate knowledge from this course with concepts from Logic Design Theory to enhance overall skills and capabilities in VLSI design.

Course Duration: 45 Hours

Course Content:

Unit 1: Chip - A primer

Continuous and discrete signal - Digital and Analog systems - Semiconductor and Transistor - Evolution of chip - Categories of chip - Activities in chip making process - Timeline - FPGA - Comparison - Aspects of a chip

Unit 2: Digital Electronics - Basics

Binary numbers system - Floating point - Negative number - 1s and 2s complement format - Weighted and Non-Weighted code - Logic gates - Universal gate - Achieving logic using gates - Boolean equation - Boolean algebra - Truth table - K map - Combinational circuit - Sequential circuit - Clock - Flop, latch - Setup and hold time - Setup and hold violation

Unit 3: Digital Design - Part 1

Evolution of HDL - C vs HDL - Verilog basics - Module - Data types - Operators - Assignment types - Time and Event - Loops - Always block - Functions - Tasks - Behavioral modelling - Data flow modelling - Structural modelling

Unit 4: Digital Design - Part 2

Data flow in digital design - RTL - State diagram - State machine - Control path - Sequence detector - Pipeline - Multiple clock domain - Clock domain crossing - CSR - Cache -Synthesis - HDL Design Flow

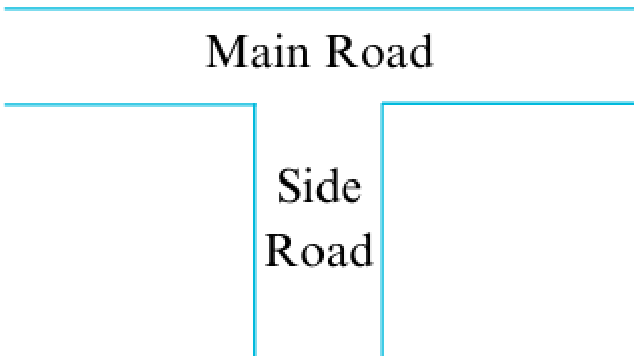
Unit 5: Digital Verification

Importance of verification - HDL design and verification flow - Types of verification - Faults - Scan insertion - Stitching - Fault simulation - Fault coverage - Test vectors - JTAG basics - List of checkers - Verification planning - Testbench development - Test writing - Simulation - Debug - Regression - Coverage and closure.

Test Projects:

Use Cases:

1. Develop a password management system, where security is paramount. One requirement is that if a user changes their password, the new password should differ from the old one by at least 70% to enhance security. How would you implement this requirement using a sequence detector to ensure that the password significantly deviates from the old one for enhance security measure.
2. Consider a manufacturing unit which manufactures plastic bottles, so each bottle has to be paired with a cap, the bottles and the cap are manufactured separately, then placed on a conveyor belt in a manner such that the bottle and a cap are always placed one after the other, so that a robot automatically closes each bottle with the cap right next to it. A system has to detect and give an alert once every four bottle-cap pair has arrived. Hence, four bottles can be packed together after detection. Arrive a state machine for such an alert system. (Hint: You may consider bottle as 0 and cap as 1)
3. Draw a state diagram and generate output signals for traffic signal controller. The signals need to control traffic in a T junction. T junction is intersecting a side road and a main road. By default, the main road traffic light needs to be green. When a vehicle is detected on the side road, the main road light needs to turn yellow and after a delay turn red and then the side road light needs to turn green. The side road gets green light until there are cars on it. Once there are no cars in the side road, the side road light needs to turn yellow and then after a delay turn red and the main road signal need to turn green. The delays can be a fixed input to the controller.



4. A provisional store owner wants to sell items such as beverages, soft drinks, chips and biscuits which are highly in demand in an efficient manner, so he wants to devise a system that automates the buyer and seller interaction where the required product is dispensed to the customer if he provides the

system the required amount for that product.

5. We have witnessed a significant increase in usage of EV vehicles. The main source of energy that fuels these vehicles is a Lithium-Ion (Li-ion) battery. These batteries are vulnerable to certain scenarios which can cause them to explode. Implement a Verilog digital design that provides a solution to this problem by monitoring the temperature and voltage of the battery.
   Note: This design incorporates features such as masking and deglitching to detect and manage faults effectively.
6. Design a digital clock capable of displaying hours, minutes, and seconds. The clock should operate in either 24-hour format or 12-hour format. Implement binary counters for hours and minutes that count from 00 to 23 or 00 to 12 for hours, and from 00 to 59 for minutes. The second's display is configurable, and the clock format (12-hour or 24-hour) should be selectable. The clock should have inputs for a clock signal and a reset signal.
7. Design a router which takes in a 16-bit input data along with 2-bit port select and 1 bit data valid and routes the input data to one of four output ports. Port 0 is 8 bits wide; Port 1 is 16 bits wide; Port 2 is 32 bit wide and Port 3 is 64 bit wide. Each of the ports have individual data valid to indicate a valid cycle of data. The design has to make sure that a continuous stream of data coming in is routed without any extra latency.
8. Consider a smart air conditioner (AC) which has a temperature sensor and a voice controller. The AC automatically turns OFF if the room temperature reaches or goes below 18 degrees Celsius and automatically turns ON if the room temperature reaches or goes above 26 degree Celsius. It also turns ON or OFF based on a voice controller. Model a system to sense and control the temperature, and list the set of possible inputs and outputs of this system?
9. We have witnessed a significant increase in usage of EV vehicles. The main source of energy that fuels these vehicles is a Lithium-Ion (Li-ion) battery. These batteries are vulnerable to certain scenarios which can cause them to explode. A digital design that provides a solution to this problem by monitoring the temperature and voltage of the battery. Arrive at the verification plan and implement the verification code for the design.
10. For a digital clock design capable of displaying hours, minutes, and seconds. The clock should operate in either 24-hour format or 12-hour format. Implement binary counters for hours and minutes that count from 00 to 23 or 00 to 12 for hours, and from 00 to 59 for minutes. The seconds display is configurable, and the clock format (12-hour or 24-hour) should be selectable. The clock should have inputs for a clock signal and a reset signal. Arrive a verification plan which should comprise a test plan and checker plan. Implement the verification code for such a digital design.
11. Router design which takes in a 16-bit input data along with 2 bit port select and 1 bit data valid and routes the input data to one of four output ports.

Port 0 is 8 bit wide, Port 1 is 16 bit wide, Port 2 is 32 bit wide and Port 3 is 64 bit wide. Each of the ports have individual data valid to indicate a valid cycle of data. The design has a continuous stream of data coming in and is routed without any extra latency. Plan and arrive at the verification plan consisting of a test plan and checker plan. Implement the verification code for such a design.

12. A smart air conditioner (AC) which has a temperature sensor and a voice controller. The AC automatically turns OFF if the room temperature reaches or goes below 18 degrees Celsius and automatically turns ON if the room temperature reaches or goes above 26 degrees Celsius. It also turns ON or OFF based on a voice controller. The system senses and controls the temperature. Devise a verification plan with test plan and checker plan. Also implement the verification code for the system design.
13. For a UART receiver design
    a. Arrive at a verification plan detailing tests and checkers
    b. Write code to model and drive the inputs.
    c. Write code to monitor the inputs and outputs.
    d. Write code to predict the expected output and check with actual output.
14. Design the code for the following scenario: The design receives a set of inputs and passes them to an external device. However, within the design, certain logic is applied to conceal the inputs, meaning that the original inputs are transformed into another form. Similarly, if concealed data is inputted into the design, it revises the concealed data back into its original form.
15. Design the code for the following scenario: The design receives a set of inputs and passes them to an external device. However, within the design, certain logic is applied to conceal the inputs, meaning that the original inputs are transformed into another form. Similarly, if concealed data is inputted into the design, it revises the concealed data back into its original form.
16. Design a PWM-based speed controller for a DC motor. The speed controller should be able to adjust the speed of the DC motor based on a given PWM duty cycle. Implement the following components: PWM Generator: Design a module that generates a PWM signal with a configurable duty cycle. The duty cycle should be adjustable in real-time to control the speed of the DC motor. Speed Controller: Implement a speed controller module that takes the PWM signal as input and adjusts the speed of the DC motor accordingly. The speed controller should include a feedback mechanism to ensure the motor runs at the desired speed. Real-Time Operation: Ensure that the speed controller operates in real-time, meaning it responds quickly to changes in the PWM signal and adjusts the motor speed accordingly.
17. Design a PWM-based speed controller for a DC motor. The speed controller should be able to adjust the speed of the DC motor based on a given PWM duty cycle. Implement the following components: PWM Generator: Design a module that generates a PWM signal with a configurable duty cycle. The duty

cycle should be adjustable in real-time to control the speed of the DC motor. Speed Controller: Implement a speed controller module that takes the PWM signal as input and adjusts the speed of the DC motor accordingly. The speed controller should include a feedback mechanism to ensure the motor runs at the desired speed. Real-Time Operation: Ensure that the speed controller operates in real-time, meaning it responds quickly to changes in the PWM signal and adjusts the motor speed accordingly.

18. Imagine you're working on a project to automate a small factory. There are different sensors placed all around the factory that collect information like temperature, pressure, and more. This data needs to be handled carefully and sent to the main control system without losing any of it, even if the control system is busy. Your task is to create a system called a FIFO (First-In, First- Out) to manage this data flow smoothly. Think of it like a queue at a store - the first items in line get served first. Your FIFO should make sure all the data from the sensors gets safely stored and sent to the control system in the order it was collected. Implement a digital design to handle all the sensor data effectively and verify the design.
19. A processor must get instructions and compute these instructions. Then, provide the processed instructions as output. It must be able to process only valid inputs and provide only valid processed outputs. The instructions are primarily in binary system which must be manipulated and processed based on different operations. Implement such a design and verify the design using a simple testbench
20. In a school administration system, there's a computing engine designed to handle various computations related to student exam marks. This computing engine operates in four different modes:

- Individual: Student Average Calculation: Calculate the average marks for one student across all subjects.
- Engineering: Course Cutoff Calculation: Determine the engineering course cutoff based on Physics, Chemistry, and Maths marks.
- Medical: Course Cutoff Calculation: Determine the medical course cutoff based on Physics, Chemistry, and Biology marks.
- Total: Subject Average Calculation: Calculate the average marks for each subject scored by all 30 students.
- Design the RTL module to handle these functionalities efficiently, ensuring accurate calculations and real-time processing and ensure that the design is bug-free.