ANNEXURE I - Course Curriculum

Stream 1: Software for Product (Embedded)

S. No	Module	Content	Duration in Hrs
1.	Introduction to Embedded Systems	What are embedded systems, characteristics, and differences between Computer and embedded systems.	2
2.	Embedded System Architecture	Classic CPU architecture – generic microprocessor architecture – basic fault-tolerant architecture	3
3.	Introduction to uCs and Processors	Introduction to ATMEL 328 microcontroller	3
4.	Memories and their characteristics	NOR, NAND, EPROMS, E2PROMS, SRAM, DRAM, SPINAND, FRAMS	3
5.	Digital Input/Output ports	Basic logic levels – digital inputs – digital outputs – uses of Digital ports	3
6.	Analog Inputs and Analog to Digital Conversion	Analog to Digital Conversion – Successive Approximation – Sigma Delta ADC – important components of ADC	3
7.	Interrupts	Interrupts in microprocessor and how they work	3
8.	Counters and Timers	Basics of Counters, timers and PWM generation	3
9.	Serial Interfaces	UART, I2C, SPI interfaces	3
10.	Processor Boot up Process	How the uC and Processors boot up	3
11.	Bootloaders and Diagnostics	Introduction to boot loaders and diagnostics	3
12.	Developing uC code and tools	Introduction Code development tools and process	3
13.	Basics of Realtime Programming	Basic Realtime program concepts	3
14.	Introduction to Arduino	An introduction to Arduino and its eco system	3
15.	Developing Processor based System	How embedded product is developed	3
16.	Conclusion	Taking it forward.	1

Stream 2: Design Thinking

S.No	Module Name	Topics Covered	Duration (Hrs.)
1	Introduction to Design Thinking	1. Introduction to the principles of Design Thinking 2. Steps followed in design thinking	2
2	Problem Identification and Empathy Mapping	1. Techniques for identifying problems. 2. Creating empathy maps 3. Synthesize and Define Problem Statements	4
3	Ideation Techniques and Brainstorming	1. Various ideation techniques 2. Effective brainstorming methods 3. SCAMPER Technique, Reverse Brainstorming	4
4	User Research and Prototyping	Conducting user research and prototyping solutions	5
5	Algorithmic Problem-solving Fundamentals	1. Fundamentals of algorithmic problem solving 2. Problem-solving techniques	5
6	Data Structures and Algorithms	1. Overview of commonly used data structures 2. Study of efficient problem-solving	5
7	Advanced problem-solving strategies and techniques	1. Explore advanced strategies for problem- solving. 2. Theory of Inventive problem-solving 3. Agile problem-solving methodology	5
8	Advanced Algorithms and Analytical Thinking	1. Dive into complex algorithms and their analysis 2. Analyze the time and space complexity of algorithms. 3. Explore various Sorting and searching algorithms with complexity	5
9	User Testing and Iterative Design	Learn how to test and iterate designs. Understanding the importance of prototyping in the design process Exploring the process of testing a system or product by end users	5
10	Design thinking in different industries	Explore industry-specific design thinking applications	5

Stream 3: VLSI

S.No	Module	Content	Duration in Hrs
1.	Digital Design Fundamentals	Basic principles of digital electronics and logic design. working of logic gates, flip-flops, Combinational and sequential circuits.	2
2.	VLSI Life Cycle	System Specification, Architectural Design, Functional Design, Logic Design, Circuit Design, Physical Design, Fabrication, Packing and Testing	1
3.	Semiconductor Basics: MOS Device Physics	Device Physics, Structure and Threshold Voltage I-V Characteristics, MOS Non-idealities (Channel length modulation, Body effect, Subthreshold conduction), MOS Intrinsic capacitance	2
4.	Semiconductor Basics: CMOS technology	P – N Junction, Transistor type, MOS Transistor CMOS (inverter, NAND, NOR), CMOS fabrication Physical layout and design rules, Technology scaling	2
5.	Introduction to Hardware Description Languages (HDLs)- Verilog	Introduction to VERILOG: Levels of design Description Module, Simulation and Synthesis Tools, Test Benches, Language constructs and conventions, Levels of design description	4
6.	Digital Design Flow - RTL Design	Elements of RT Level Design, Combinational Elements o RTL Design, RTL methodology, RTL timing, RTL processing element design	10
7.	Logic synthesis and Timing Constraints	Design Flow, Logic Synthesis, Objective Function for Synthesis, Constraints on Synthesis, Synthesis Flow, Static Verification Flow Basics of timing	3
8.	Low Power Design Techniques	Need for low power design, Power Component, , Low power design techniques, Power and reliability, Synthesis power requirement, Physical implementation requirement, Power and reliability requirement	3
9.	Verification Strategies - Universal	UVM Base, Testbench structure and components UVM Phases, Stimulus generation, UVM factory	2

	verification methodology (UVM)	override	
10.	Verification Techniques:	Verification Vs Validation, Methodology component, Verification flow in vlsi design, Type of verification, Formal and Advanced Verification Techniques, Verification plan	2
11.	Design for Testability (DFT)	Need for DFT, Role of DFT, Verification Vs Testing DFT techniques	2
12.	High level design representation	Control Data Flow Graph, Transformation for High Level Synthesis, Code optimization	1
13.	Introduction to Physical design	Overview of the physical design flow, Comparison between frontend and backend	1
14.	Design Flow Methodologies	Steps in the physical design flow (RTL to GDSII)/ASIC design flow, Full custom, semi-custom, Standard cell/ FPGA	4
15.	Circuit and Layout	Semiconductor theory, MOS Transistors: NMOS, PMOS, CMOS, Stick Diagram, Lambda based design rules	5
16.	Conclusion	Taking it forward.	1

Annexure II - Industry Use Cases/Final Projects

Stream 1: Software for Product (Embedded)

Scenario	Use-Case	Description	Duration (Weeks)	Hours	Open-Source Tools to be Enabled
Smart Home Automation Hub	Develop a central hub for smart home automation that controls various IoT devices such as lights, thermostats, and security cameras.	Embedded Systems Focus: Microcontroller programming, sensor integration, and communication protocols.	2	40	Wokwi, PHET, Embedded Kit Online
Wearable Health Monitoring Device	Create a wearable device that monitors health parameters like heart rate, temperature, and activity level.	Embedded Systems Focus: Sensor integration, low-power design, real-time data processing.	2	40	Wokwi, PHET, Embedded Kit Online
Autonomous Drone Navigation System:	Develop a drone with autonomous navigation capabilities, avoiding obstacles and following predefined routes.	Embedded Systems Focus: Sensor fusion, motor control, GPS integration.	2.5	55	Wokwi Simulator, Microcontroller, Wifi Modules and Sensor Modules
Industrial IoT (IIoT) Asset Tracking	Implement an IIoT solution for tracking and monitoring assets in an industrial setting, ensuring efficient resource allocation.	Embedded Systems Focus: RFID/NFC integration, wireless communication, real-time tracking algorithms.	2.5	55	Wokwi, PHET, Embedded Kit Online
Energy-Efficient Home	Description: Design a home automation	Embedded Systems Focus: Sensor	1.5	35	Wokwi Simulator, Microcontroller,

Automation	system that optimizes energy consumption by intelligently controlling lighting, heating, and cooling.	integration, energy monitoring, machine learning algorithms.			Wifi Modules and Sensor Modules
Robotics Arm for Manufacturing	Description: Create a robotic arm for manufacturing processes that can perform precise tasks with automation.	Embedded Systems Focus: Motor control, feedback systems, precision control algorithms.	1.5	35	Wokwi Simulators, GPS modules, Accelerometer

Stream 2: DESIGN THINKING

S.No	Module	Project title
1	Introduction to Design Thinking	1. A user wants the app to recommend personalized recipes based on my dietary preferences and available ingredients. Design Thinking: Conduct user interviews and surveys to understand common dietary preferences and ingredient availability challenges, ensuring the recommendation system caters to diverse user needs. As a project manager, you are assigned with task to generate reports on bug statistics and trends to identify areas of improvement in the software development process. 2. Design Thinking: Identify the reporting needs and preferences of project managers, including the specific data points and visualizations they require.
2	Problem Identification and Empathy Mapping	1. You are part of a team tasked with improving the waiting experience in a busy medical clinic. Discuss in small groups the key concepts and principles highlighted in the presentation (include Empathy map). Share your group's discussion outcomes with the whole class. 2. Sarah wants her students to work together on a group project, easily sharing documents and discussing ideas within the app. Activity: Design a file-sharing feature where students can upload and access project files and integrate a chat function for real-time collaboration. Document the Solution with the

		Empathy Map.
3	Ideation Techniques and Brainstorming	1. You are part of a product design team for a smart home automation system. conduct a brainstorming session to generate ideas for enhancing the user experience of the smart home automation system. Document and present the ideas. 2. John needs help understanding a math concept and wants to ask his classmates for explanations or seek guidance from his teacher. **Activity**: Include the chat or discussion board feature that allows students to post questions, receive answers, and receive input from the teacher. Perform brainstorming session to prepare the list of specifications to design chat sessions. Document and present the ideas.
4	User Research and Prototyping	1. You are part of a team developing a mobile app for a travel planning service. Conduct user research by interviewing potential users to understand their needs and pain points in the travel planning process. Create personas based on the research findings to represent different types of users. Design and create low-fidelity prototypes of the mobile app based on the research insights and user journey maps. 2. Lily wants to review lecture notes and access resources shared by her teacher for an upcoming exam. Develop a prototype for the resource library to be used within the app, where teachers can upload and categorize important documents, making them easily accessible for students.
5	Algorithmic Problem-solving Fundamentals	1. A group of IT students were involved in a coding competition. For the given list of items each group is asked to perform sorting of the items. Analyze the algorithm used and identify the team which has performed the sorting efficiently. 2. Code for the appointment booking is available in Internet. Analyze the code and identify the algorithms used and also provide optimization if needed
6	Data Structures and Algorithms	1. Analyze a website that tracks user orders, suggest appropriate data structure to be used for efficient search of items in the website. List down the time and space complexity of the suggestion provided. Document and present. 2. Code of an application is shared. The speed of the application is of a major concern as linear data structures are used. Modify the application code

		shared to include non-linear data structures and check the application performance. Document and Present
7	Advanced problem-solving strategies and techniques	1. You are part of a team developing a presentation to pitch a new product idea to potential investors. Gather relevant data and insights about the market, customer needs, and the product's unique value proposition. Use visualization techniques like infographics, charts, and diagrams to convey the data and insights effectively. Craft a storytelling narrative that connects the audience emotionally with the product, highlighting its benefits and impact. Practice delivering the pitch presentation, focusing on the flow, clarity, and impact of the visual and narrative components. 2. As a user, I want to have an estimated delivery time provided when tracking my package, so I can plan accordingly. Design Thinking: Incorporate a feature that predicts the estimated delivery time based on the package's current status and historical data. Algorithmic Problem Solving: Implement an algorithm that uses historical delivery data and real-time tracking updates to calculate the estimated delivery time.
8	Advanced Algorithms and Analytical Thinking	1. As a user, I want the app to recommend personalized recipes based on my dietary preferences and available ingredients. Design Thinking: Conduct user interviews and surveys to understand common dietary preferences and ingredient availability challenges, ensuring the recommendation system caters to diverse user needs. Algorithmic Problem Solving: Design and implement an algorithm that analyzes user profiles, dietary restrictions, ingredient databases, and recipe libraries to generate tailored recipe recommendations. 2. As a shipping company manager, I want an analytics dashboard to track and analyze delivery performance metrics. Design Thinking: Understand the specific analytics requirements of shipping company managers and their key performance indicators. Algorithmic Problem Solving: Develop algorithms to collect and process data related to delivery time, customer satisfaction, delivery success rates, and other relevant metrics to generate insights in the analytics dashboard.

9	User Testing and Iterative Design	1. You are part of a team working on improving the checkout process of an e-commerce website. Analyze user feedback and data to identify pain points and areas of improvement in the current checkout process. Conduct usability testing with a small group of users to gather feedback on the proposed design changes. Based on the feedback, make iterative design changes to address the identified issues. 2. For the API, prepared in the previous session, develop a test plan, incorporate the feedback observation and perform the iterative design and feedback loop.
10	More Design thinking scenarios	1. You have been assigned to lead a team responsible for implementing a new employee onboarding program. Develop a detailed project plan outlining the objectives, deliverables, timeline, and resources required for the onboarding program. Delegate tasks to team members based on their expertise and availability. Monitor the progress of each task and provide guidance and support as needed. Conduct regular team meetings to discuss challenges, review progress, and ensure alignment. 2. As a software developer, I want to easily report and track bugs in the system to ensure efficient debugging and issue resolution. Design Thinking: Identify the pain points and challenges faced by software developers in reporting and tracking bugs and design a streamlined and intuitive bug reporting feature. Algorithmic Problem Solving: Implement an algorithm that assigns unique identifiers to each reported bug, tracks the bug's status and progress, and allows developers to attach relevant information and updates.

STREAM 3: VLSI

S. No.	Scenario	Use-Case
1	Adders: Full adder, carry look ahead adder, carry skip adder, carry Select adder	Design module for adder circuits simulate, synthesize all adders circuit, and verify by test benches. Extract report and compare all adder's area, time, and power requirements.
2	16 x1 multiplexer	The structural hierarchical description of a 16 x1 multiplexer (a) Using pure behavioral modeling (b) Structural modeling using 4 to 1 multiplexer specified using behavioral model. (c) Make Structural modeling using 4 to 1 multiplexer specified using behavioral model of 2 to 1 multiplexer. (d) Make Structural modeling using 4 to 1 multiplexer to have a complete structural hierarchical description.
3	4-bit ALU	Design a 4-bit ALU that can perform various arithmetic and logical operations. The ALU should take two 4-bit inputs and a 3-bit control signal to determine the operation (e.g., addition, subtraction, AND, OR XOR, etc.). The output should include a 4-bit result and a carry-out flag.
4	Simple LED blinker	Design a Verilog module to blink an LED with a specific frequency. Verilog Code Structure: Use a counter to generate the blinking frequency. Toggle the LED output
5	Traffic Light Controller	Create a Verilog module to simulate a simple traffic light controller. Verilog Code Structure: Implement a state machine to control the sequence of traffic light signals.
6	Digital Dice	Design a digital dice that simulates the roll of a six-sided die. The module should have a button input to trigger the roll and a 7- segment display output to show the result.
7	8-Bit Shift Register with Parallel Load	Design an 8-bit shift register that supports serial input, serial output, and parallel load operations. The module should have control signals to select the operation mode and clock input for synchronization.
8	Digital Stopwatch	Design a digital stopwatch using Verilog. The stopwatch should have start, stop, and reset buttons. The display should show minutes, seconds, and hundredths of a second.
9	8x8 LED Matrix	Design a controller for an 8x8 LED matrix. The module

	Controller	should allow you to display patterns or scroll text across the matrix. Use multiplexing techniques to control the rows and columns of the matrix.
10	Simple 8-bit RISC Processor	Design a simple RISC (Reduced Instruction Set Computing) processor. The processor should support a small set of instructions (e.g., load, store, add, subtract, branch) and have a basic pipeline architecture. Implement instruction fetch, decode, execute, memory access, and write-back stages
11	Digital Clock	Design a digital clock that displays hours, minutes, and seconds. Include a control interface to set the time and a synchronization signal to reset the clock.
12	Binary Clock	Design a binary clock that displays the current time in binary format using LEDs. Implement the clock logic and ensure accurate timekeeping.
13	4 x 4 multiplier	Design and implement a 4x4 unsigned binary multiplier using Verilog. The multiplier should take two 4-bit binary inputs and produce an 8-bit binary output representing the product.
14	Digital lock	Design a digital lock system with a keypad for inputting a password. The module should compare the entered password with a stored password and trigger an unlock signal if they match.

Annexure III - Assessment Rubrics

Final Test Project/External Assessment Plan:

The Final Test Project will be chosen from the list given above, jointly by the college faculty and the Training Partner. The Final Test Project will be assessed on the following tasks, for 100 marks:

COURSE ASSESSMENT RUBRICS (TOTAL MARKS:100)				
ASSESSMENT CRITERIA	**DESCRIBE THE CRITERIA OF THE BELOW CATEGORY PERFORMANCE**			**TOTAL MARKS**
	FAIR	**GOOD**	**EXCELLENT**	
Problem Definition & Design Thinking	6	8	10	10
Innovation & Problem Solving	10	15	20	20
Implementation of Project	10	15	20	20
Performance of the Project	10	15	20	20
Project Demonstration & Documentation	10	15	20	20
MCQ-based assessment 20 Questions			10	10
Total				100