Skill Enhancement Courses (SEC)

Course Code	Course Name	Level	L	T	P	C	CIE	SEE	Total	Pre-requisite
241EC021	PCB Design	FC			2	2	50	50	100	-
241EC094	Soft Computing Tools	IC			2	2	50	50	100	-
241EC015	Android Applications	AC			2	2	50	50	100	-
241EC016	ECAD Tools	AC			2	2	50	50	100	VLSI
241EC017	Verification using Verilog & UVM	AC			2	2	50	50	100	VLSI
Total					**10**	**10**				

Skill Enhancement Courses (SEC)

PCB Design

	L	T	P	C
Course Code: 241EC021	**0**	**0**	**2**	**2**

Course Outcomes:

At the end of the course, student will be able to:

- **CO1:** Illustrate the equipment and components used in PCB design.
- **CO2:** Apply the procedural steps for PCB preparation.
- **CO3:** Make use of chemicals for Etching process.
- **CO4:** Make use of Soldering gun for components mounting.
- **CO5:** Construct PCB for different circuits.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2						2	2		1
CO2	2	2						2	2		1
CO3	2	2						2	2		1
CO4	2	2						2	2		1
CO5	2	2	2					2	2		1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1		1
CO2		1
CO3		1
CO4		1
CO5		1

Practice:

1. Study of lab equipments and components:
   a) CRO
   b) Multimeter
   c) Function Generator
   d) Power supply
   e) Active Components
   f) Passive Components
   g) Bread Board
   h) Soldering gun.
2. Preparation of layout and artwork layout planning.
   a) Layout Planning
   b) General Artwork rules

c) Components Polarity Identification

3. Preparation of Negative from the film.
   a) Artwork
   b) Exposure time
   c) Development
   d) Stop bath
   e) Fixing bath
   f) Film washing
   g) Drying
4. Principle of a simple PCB.
   a) Laminate cleaning
   b) Laminate coating
   c) Thinner
   d) Prefacing
   e) Exposer
   f) Development
   g) Washing
   h) Drying the Image
   i) Post Curing
5. Etching and Drilling of PCB.
   a) Etching
   b) drilling
6. Preparation and mounting components.
   a) Component lead preparation
   b) Component mounting
   c) Soldering
7. PCB preparation of Simple diode circuits
   a) Diode as a switch
   b) Bridge rectifier
8. PCB preparation of Simple transistor circuits
   a) Transistor as a switch
   b) Single stage CE amplifier
9. PCB preparation of Simple IC based circuits
   a) Voltage Regulator using IC 78XX and IC 79XX
10. PCB preparation of IC based circuits
   a) Summing circuit using Op-Amp
   b) Astable multivibrator circuit using IC 555 timer
   c) PLL circuit using IC 565

Reference Books:

1. Printed Circuits Handbook,Clyde F. Coombs Jr., Happy T. Holden, Seventh Edition, McGraw Hill publications, ISBN: 978-0071833950.
2. PCB Design and Layout Fundamentals for EMC, RogerHu, 2019, ISBN: 978-1082079252.
3. The Hitchhiker's Guide to PCB Design, David Ruff, Mike Brown, Cadence Documents, ISBN: 978-0368246968.

Web Links:

1. http://www.nptelvideos.com/lecture.php?id=14721 by Prof. Santhanu Bhattacharya, IIT, Kanpur

Soft Computing Tools

Course Code: 241EC094	L	T	P	C
	0	0	2	2

Course Outcomes:

At the end of the Course, Student will be able to:

CO1: Perform basic matrix operations and random data manipulation using MATLAB.
CO2: Make use of MATLAB to generate Gaussian noise and compute its basic statistical measures.
CO3: Design filters with the help of Simulink toolbox.
CO4: Apply the concepts of convolution and filters to remove noise.
CO5: Utilize the Fuzzy tool box to perform different operations

Mapping of Course Outcomes with Program Outcomes

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2			2			2	2		1
CO2	2	2			2			2	2		1
CO3	2	2			2			2	2		1
CO4	2	2			2			2	2		1
CO5	2	2			2			2	2		1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	
CO2	1	
CO3	1	
CO4	2	
CO5	1	

Practice:

1. Create a 10x10 random matrix with the command A= rand (10). Now do the following operations.
    a. Multiply all elements by 100 and then round off all elements of the matrix to integers with the command A=fix(A).
    b. Replace all elements of A<10 with Zeros.
    c. Replace all elements of A>90 with infinity (int).
    d. Extract all $30 \leq a_{ij} \leq 50$ in a vector b, that is find all elements between 30 and 50 and put them in a vector b.
2. Generation of Gaussian noise (Real and Complex), Computation of its mean, Variance, Mean Square Value and Probability Distribution Function.

3. Removal of noise by Autocorrelation / Cross correlation.
4. Removal of noise by Filters (Wiener/LMS).
5. Introduction to Simulink toolbox
6. Design of Filters using Simulink (LPF, HPF, BPF, BRF)
7. Introduction to Fuzzy Logic toolbox
8. Write a program in MATLAB to perform Union, Intersection and Complement operations.
9. Write a program using Fuzzy Logic to perform arithmetic and Logic Operations
10. Write a program in MATLAB to implement De-Morgan's Law.
11. Write a program in MATLAB to plot various membership functions.
12. Implement Fuzzy Inference System (FIS)

Additional Practice:

1. Identify a relevant problem in your domain and develop an appropriate solution using the Simulink toolbox, applying the knowledge you have gained so far.
2. Identify a relevant problem in your domain and develop an appropriate solution using the Fuzzy Logic toolbox, applying the knowledge you have gained so far.

Web Links:

1. Soft Computing Tools in Engineering (iitkgp.ac.in)

Android Applications

	L	T	P	C
Course Code: 241EC015	**0**	**0**	**2**	**2**

Course Outcomes:

At the end of the course, student will be able to:

- **CO1:** Utilize the android studio for a mobile application
- **CO2:** Demonstrate the android application
- **CO3:** Develop the android User Interface
- **CO4:** Make use android app for gaming apps
- **CO5:** Demonstrate home automation using android app

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2	1		1			1	1		1
CO2	2	3	1		1			1	1		1
CO3	2	2	1		1			1	1		1
CO4	2	2	3		2			1	1		1
CO5	2	2	3		2			1	1		1

Practice:

1. Familiarization with Kotlin programming.
2. Familiarization with Android Studio.
3. Creation of "Hello World" Android App, Running on Android Emulator, connecting the Android device.
4. Basic UI design.
5. Layout design.
6. Advanced UI design
7. JetPack Compose App Design.
8. Accessing Mobile sensors data in android app.
9. API Design.
10. Simple Game App design.

Additional Practice:

1. Android App Compose UI Testing.
2. Write a mobile application that makes use of RSS feed.

Reference Books:

1. Android application development all-in-one for dummies by Burd B. John Wiley & Sons, 2021, ISBN: 978-9354245787.

Web Links:

1. https://developer.android.com/courses
2. Udemy Android App Development - Complete Course online https://www.udemy.com/topic/android-development/
3. Coursera Android app development specialization course https://www.coursera.org/specializations/android-app-development

ECAD Tools

Course Code: 241EC016	L	T	P	C
	0	0	2	2

Course Outcomes:

At the end of the course, student will be able to:

CO1: Comprehend the insight of CAD Tools in modern design.
CO2: Develop combinational logic circuits by using CAD tools.
CO3: Build sequential logic circuits using Verilog HDL operators.
CO4: Analyze the performance of logic schematics using CAD simulation tools.
CO5: Infer the performance of logic circuits using DRC, LVS and PEX.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	1			2			1	1		1
CO2	3	2			2			1	1		1
CO3	2	2			2			1	1		1
CO4	3	2			2			1	1		1
CO5	3	2			2			1	1		1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1		2
CO2		2
CO3		2
CO4		2
CO5		2

Practice:

1. Understanding the working platform with Xilinx Vivado and its device, family and
   a. package Selection.
2. Design and Implementation of Combinational Circuits Priority Encoder and Comparator using data flow & structural style.
3. Design and Implementation of Sequential Circuits to detect a given sequence using with and without overlapping (mealy & Moore machines).
4. Design and Implementation of a traffic light controller in a three road & four road junctions.
5. Exercise on Concatenation, Replication operators, Reduction and Conditional operators in Verilog HDL.
6. Plotting the (i) output characteristics (ii) Transfer characteristics of an n-channel and p-channel MOSFET with Cadence.

7. Working with Schematic for Ring Oscillator with variable amounts of Pull up to pull down ratios.
8. Design a full adder by instantiating the logic gates. Make a comment on design style on its performance.
9. Design a NAND gate by using NMOS, PMOS and CMOS technologies and make a comment on its performance.
10. Design a layout for given logic and test for its DRC, LVS and PEX. Make a note on change in propagation delay with respect to PEX data.

Additional Practice:

3. Design and plot the characteristics of a 4x1 digital multiplexer using pass transistor logic and TG logic.
4. Design and plot the characteristics of a positive and negative latch based on multiplexers

Text Books:

1. CMOS Digital Integrated Circuits : Analysis and Design, S. M. Kang and Y. Leblebici, McGraw-Hill Education, 3rd Edition, ISBN: 978-0071243421.
2. Silicon VLSI Technology: Fundamentals, Practice & Modeling, James D. Plummer, Micheal Deal , Peter B. Griffin, Pearson India, 1st Edition, ISBN: 978-0130850379.

Reference Books:

1. Microchip Fabrication: A Practical Guide to Semiconductor Processing, Peter Van Zant, McGraw-Hill Professional, 6th Edition, ISBN: 978-0071821018.
2. CMOS Circuit Design, Layout and Simulation, R. Jacob Baker, Wiley-IEEE Press, 4th Edition, ISBN: 978-0780334168.
3. J. M. Rabaey, A. P. Chandrakasan and B. Nikolic, Digital Integrated Circuits: A Design Perspective, Pearson Education India, 2nd Edition,ISBN: 978-9332573925.

Web Links:

1. https://www.cadence.com/en_US/home/company/cadence-academic-network/educators/vlsi-fundamentals.html, Cadence Design Systems.
2. https://www.xilinx.com/video/hardware/getting-started-vivado-high-level-synthesis.html , Xilinx Vivado High Level Synthesis.
3. https://onlinecourses.nptel.ac.in/noc23_ee137/preview. Prof. Sneh Saurabh, IIIT Delhi.

Verification Using Verilog & UVM

	L	T	P	C
Course Code: 241EC017	**0**	**0**	**2**	**2**

Course Outcomes:

At the end of the course, student will be able to:

- **CO1:** Develop Basic Testbench using UVM.
- **CO2:** Apply assertion based and Coverage – Driven verification techniques to the testbench.
- **CO3:** Implement Functional Coverage models and verify complex protocols like AXI, PLle etc.,
- **CO4:** Implement TLM interfaces for communication and advanced constrained Networks.
- **CO5:** Verify designs using multiple interfaces and mixed signals.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	1			2			1	1		1
CO2	3	2			2			1	1		1
CO3	2	2			2			1	1		1
CO4	3	2			2			1	1		1
CO5	3	2			2			1	1		1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1		2
CO2		2
CO3		2
CO4		2
CO5		2

Practice:

1. Basic Testbench:
   a. Develop a simple Verilog testbench to simulate a digital design.
   b. Create stimuli and check responses manually.
2. Basic UVM Testbench
   a. Develop a simple UVM testbench framework.
   b. Implement basic UVM components: env, agent, driver, sequencer, and monitor.
3. Stimulus Generation:
   a. Use UVM sequences to generate stimuli.

b. Create constrained random test cases.

4. Score boarding:
    a. Implement a scoreboard in UVM to compare expected vs. actual results.
    b. Use functional coverage to track verification progress.
5. Assertion-Based Verification:
    a. Write System Verilog assertions to check protocol compliance.
    b. Integrate assertions into a UVM testbench.
6. Coverage-Driven Verification:
    a. Develop coverage models using cover groups and cover points.
    b. Analyse coverage results to identify untested scenarios.
7. VIP Integration:
    a. Integrate UVM Verification IP (VIP) into the testbench.
    b. Verify complex protocols like AXI, PCIe, etc.
8. Functional Coverage:
    a. Define and implement functional coverage models.
    b. Analyse coverage data to ensure all design functionalities are tested.
9. Reusable UVM Components:
    a. Create reusable UVM components for different testbenches.
    b. Parameterize components for flexibility.
10. Transaction-Level Modelling (TLM):
    a. Implement TLM interfaces for communication between UVM components.
    b. Use TLM for high-level transaction Modelling.
11. Advanced Stimulus Generation:
    a. Implement advanced constrained random techniques.
    b. Use UVM sequences to control stimulus generation dynamically.
12. Debugging and Logging:
    a. Utilize UVM reporting mechanisms for debugging.
    b. Implement custom error and status messages.

Additional Practice:

1. Multiple Interface Verification:
    a) Verify designs with multiple interfaces.
    b) Develop agents and drivers for each interface.
2. Mixed-Signal Verification:
    a) Integrate Analog models with digital testbenches.
    b) Use Real-number modelling (RNM) techniques.
3. Performance and Timing Verification:
    a) Measure and verify timing constraints.
    b) Use UVM to simulate and check timing behaviours
4. Interoperability Testing:
    a) Test interoperability between different modules or IP blocks.
    b) Use UVM to create complex test scenarios.

Text Books:

1. A Practical Guide to Adopting the Universal Verification Methodology (UVM) Second Edition, ISBN: 978-1300535935.
2. The UVM Primer:A Step-by-Step Introduction to the Universal Verification Methodology by Ray Salemi, ISBN: 978-0974164939.

Reference Books:

1. A Practical Guide for System Verilog Assertions by Srikanth Vijayaraghavan & Meyyappan Ramanathan, ISBN: 978-1489992796.
2. Getting Started with UVM: A Beginner's Guide by Vanessa R. Coppe, ISBN: 978-0615819976.

Web Links:

1. https://www.cadence.com/en_US/home/training/all-courses/86070.html
2. https://www.chipverify.com/tutorials/uvm
3. https://verificationacademy.com/forums/t/systemverilog-or-uvm/39566