GANPAT UNIVERSITY

FACULTY OF DIPLOMA ENGINEERING

Programme	Diploma in Electronics and Communication Engineering		
Semester	III	Version	1.0.0.0
Effective from Academic Year	2026-27	Effective for the batch Admitted in	JULY 2025
Course code	1EC3105	Course Name	Digital Design using HDL

I.TEACHING-LEARNING AND ASSESSMENT SCHEME

Course Type	Course Code	Learning Scheme						Assessment Scheme										
		Actual Contact Hrs./Week			SLH	NLH	Credits	Theory				Practical				Based on SL		Total Marks
								FA-TH	SA-TH	TOTAL		FA-PR	SA-PR	TOTAL		SLA		
		CL	TL	LL				MAX	MAX	MAX	MIN	MAX	MAX	MAX	MIN	MAX	MIN	
SEC	1EC3105	-	-	4	4	8	4	-	-	-	-	60	40	100	40	20	8	120

Abbreviation:	**CL-** Classroom Learning	**TL -** Tutorial Learning	**LL -** Laboratory Learning
	SLH - Self Learning Hours	**NLH -** Notional Learning Hours	**SLA -** Self Learning Assessment
	FA - Formative Assessment (Term work +Mid Sem Exam +Attendance)		**SA -** Summative Assessment

II. PRE-REQUISITES

Basic knowledge of Digital Electronics

III. INDUSTRY / EMPLOYER EXPECTED OUTCOMES

Align with the skills, knowledge, and competencies required to meet real-world job demands in sectors like electronics, embedded systems, automation, and Semiconductor. Here's a structured list of expected outcomes.

IV. COURSE LEARNING OUTCOMES

At the end of the course, students will be able to achieve the following course learning outcomes:

CO1: Understand the syntax and structure of Verilog Hardware Description Language (HDL).

CO2: Construct behavioral and structural models for various digital circuits.

CO3: Perform synthesis and implement RTL models on FPGA/CPLD platforms.

CO4: Verify and debug digital designs using test-benches and simulation tools.

V. LABORATORY LEARNING OUTCOME AND ALIGNED PRACTICAL

Sr. No.	Practical/Laboratory Learning Outcome (LLO)	Practical Titles	Relevant COs
1	**LLO 1.1:** Understand the basics of Verilog.	Introduction to Verilog.	CO1
2	**LLO 2.1**: Understand how to use Altera Quartus-II software.	Introduction to Quartus II software.	CO1
3	**LLO 3.1:** Model basic logic gates using dataflow modeling in Verilog.	Write a Verilog code for basic gates using dataflow modeling.	CO2
4	**LLO 4.1:** Model basic logic gates using gate level modeling in Verilog.	Write a Verilog code for basic gates using gate level modeling.	CO2
5	**LLO 5.1:** Model basic logic gates using behavioral modeling in Verilog.	Write a Verilog code for basic gates using behavioral modeling.	CO2
6	**LLO 6.1:** Model universal gates using dataflow modeling in Verilog.	Write a Verilog code for universal gates using dataflow modeling.	CO2
7	**LLO 7.1**: Model universal gates using gate level modeling in Verilog.	Write a Verilog code for universal gates using gate level modeling.	CO2
8	**LLO 8.1:** Model universal gates using behavioral modeling in Verilog.	Write a Verilog code for universal gates using behavioral modeling.	CO2
9	**LLO 9.1:** Design half adder using dataflow modeling in Verilog.	Write a Verilog code for half adder using dataflow modeling.	CO2
10	**LLO 10.1:** Design half adder using gate level modeling in Verilog.	Write a Verilog code for half adder using gate level modeling.	CO2

11	**LLO 11.1:** Design half adder using behavioral modeling in Verilog.	Write a Verilog code for half adder using behavioral modeling.	CO2
12	**LLO 12.1:** Design half subtractor using dataflow modeling in Verilog.	Write a Verilog code for half subtractor using dataflow modeling.	CO2
13	**LLO 13.1:** Design half subtractor using gate level modeling in Verilog.	Write a Verilog code for half subtractor using gate level modeling.	CO2
14	**LLO 14.1:** Design half subtractor using behavioral modeling in Verilog.	Write a Verilog code for half subtractor using behavioral modeling.	CO2
15	**LLO 15.1:** Design full adder using dataflow modeling in Verilog.	Write a Verilog code for full adder using dataflow modeling.	CO2
16	**LLO 16.1:** Design full adder using gate level modeling in Verilog.	Write a Verilog code for full adder using gate level modeling.	CO2
17	**LLO 17.1:** Design full adder using behavioral modeling in Verilog.	Write a Verilog code for full adder using behavioral modeling.	CO2
18	**LLO 18.1:** Design full subtractor using dataflow modeling in Verilog.	Write a Verilog code for full subtractor using dataflow modeling.	CO2
19	**LLO 19.1:** Design full subtractor using gate level modeling in Verilog.	Write a Verilog code for full subtractor using gate level modeling.	CO2
20	**LLO 20.1:** Design full subtractor using behavioral modeling in Verilog.	Write a Verilog code for full subtractor using behavioral modeling.	CO2
21	**LLO 21.1:** Verify logic behavior using Test-benches and Waveform simulation.	Write a test bench for logic gates.	CO4
22	**LLO 22.1:** Verify logic behavior using Test-benches and Waveform simulation.	Write a test bench for logic gates.	CO4
23	**LLO 23.1:** Verify logic behavior using Test-benches and Waveform simulation.	Write a test bench for logic gates.	CO4
24	**LLO 24.1:** Assign physical hardware pins and perform device programming.	FPGA Implementation on Altera MAX 3000A of logic gates using Pin Planner.	CO3
25	**LLO 25.1:** Assign physical hardware pins and perform device programming.	FPGA Implementation Altera MAX 3000A of adders using Pin Planner.	CO3
26	**LLO 26.1:** Assign physical hardware pins and perform device programming.	FPGA Implementation Altera MAX 3000A of subtractors using Pin Planner.	CO3

VI. SUGGESTED MICRO PROJECT / ASSIGNMENTS / ACTIVITIES FOR SELF LEARNING / SKILL DEVELOPMENT (SELF LEARNING)

A.Self-Learning Assignments / Activities

1. Prepare a Chart/Model of Altera MAX3000A kit.
   Demonstrate all parts of Kit.
2. Data Sheet Reading:
   Download the PDF datasheet for the Altera FPGA/CPLD chip on your lab kit (e.g., Cyclone IV or MAX V). Identify the total number of User I/O pins and the voltage levels supported.

B. Mini Projects (Skill Development Focused)

1. Design a 4-Bit Password Lock System:
   Build a simple digital lock where a specific 4-bit sequence (via switches) triggers an "Unlock" LED. An incorrect entry triggers an "Alarm" LED.
2. Build a Working Model of a 4-bit Binary Counter:
   Using Switches and LEDs of the FPGA/CPLD kit design binary counter.
3. Design and Simulate a 4X1 MUX and 1X4 DEMUX.
   Using Switches and LEDs of the FPGA/CPLD kit design MUX and DEMUX.
4. Design an Arithmetic Logic Unit (ALU) Prototype (2-bit):
   Using Switches and LEDs of the FPGA/CPLD kit design 2 bit ALU.

5. Traffic Light Controller for a T-Junction:
   Implement a State Machine (FSM) that controls Red, Yellow, and Green LEDs with specific time delays.
6. Implement a Binary to Gray and Gray to Binary Code Converter:
   Using Switches and LEDs of the FPGA/CPLD kit design Code converter.

VII. LIST OF INSTRUMENTS / EQUIPMENT / TRAINER BOARD	
1	**CPLD Kit** Used for experimenting Altera MAX 3000A.

VIII. LIST OF REFERENCE BOOKS

Sr.No.	Title	Author	Publication
1	Verilog HDL	Samir Palnitkar	Pearson Publication
2	A Verilog HDL Primer	Bhasker J.	BS Publications (BSP), Hyderabad
3	Verilog HDL Synthesis: A Practical Primer	Bhasker J.	BS Publications (BSP), Hyderabad
4	Verilog Quickstart: A Practical Guide to Simulation and Synthesis	Lee, James M.	Springer (New Delhi/International)

IX. LINK OF LEARNING WEB RESOURCE	
1	https://onlinecourses.nptel.ac.in/noc24_cs61/preview
2	https://nptel.ac.in/courses/106105165
3	https://www.nielit.gov.in/calicut/content/fpga-architecture-and-programming-using-verilog-hdl
4	https://www.coursera.org/learn/fpga-hardware-description-languages

X. COs AND POs AND PSOs MAPPING

Course outcome (Cos)	**Programme Outcomes (POs)**							**Programme Specific Outcomes (PSOs)**		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	3	2	1	1	3	0	2	1	1
CO2	3	3	3	3	2	2	2	3	2	3
CO3	3	3	3	2	2	2	2	2	2	3
CO4	3	3	2	2	2	2	0	2	3	2
CO5	3	3	3	1	2	3	0	3	1	1

Legends: - *3- High 2-Moderate/Medium 1-Slight/Low 0-None*