Coincent
Learning Drives Excellence

Coincent 3 Year Program Curriculum
Very-Large-Scale Integration

Partnered by

APSIS
Solutions
INNOVATING THROUGHOUT

Empowering Learners,
Accelerating Careers.

coincent.ai

ABOUT COINCENT

Coincent offers a 3-Year Program that is a well-structured, career-focused initiative designed to equip students with practical skills, real-world experience, and strong placement support. The program is tailored to ensure progressive learning and career readiness across three year phases.

Why It's Unique

- Only one batch per year with limited seats (150 students) per Domain to maintain quality.
- Prepares students step-by-step to become job-ready by graduation.

"Very-Large-Scale Integration at Coincent – Learn by Doing"

VLSI is the process of integrating **thousands to billions of transistors** on a single chip to create complex electronic circuits.
It is the backbone of **modern electronics**, powering everything from smartphones to microprocessors.

Key Points:

- Enables compact, high-performance integrated circuits (ICs)
- Used in designing CPUs, GPUs, and memory chips
- Improves speed, reduces power consumption and size
- Essential in semiconductor, embedded, and IoT industries

3-Year Program Structure Breakdown

TABLE OF CONTENTS

Year 1 :- Industrial Training

Chapter 1: Orientation

1.1 Introduction

- Purpose: Understand what VLSI is and why it matters in modern electronics.
- Definition: VLSI refers to the process of integrating millions (or billions) of transistors onto a single chip.
- Real-World Impact: Found in smartphones, microprocessors, memory devices, etc.

1.2 Overview of VLSI Course

- Course Goals: Introduce digital design, semiconductor technology, and hardware description languages (HDLs)
- Outcome: Ability to design and simulate digital circuits, understand chip architecture, and implement designs using Verilog and FPGA.

1.3 Evolution of VLSI

- History: From SSI (Small Scale Integration) to ULSI (Ultra Large Scale Integration)

- Milestones:
  - ○ 1950s: Vacuum tubes
  - ○ 1960s: Discrete transistors
  - ○ 1970s: SSI/MSI
  - ○ 1980s: LSI/VLSI
  - ○ Today: SoC and AI chips

Moore’s Law: Predicts doubling of transistors every 18-24 months.

1.4 VLSI Design Flow

- Steps:
  - ○ Specification
  - ○ Architectural Design
  - ○ RTL Design (Verilog)
  - ○ Functional Verification
  - ○ Logic Synthesis
  - ○ Place and Route (Physical Design)
  - ○ Fabrication
  - ○Testing & Packaging

Benefit:

Gain a strong foundational understanding of how modern electronic systems are built — from historical evolution to real-world application and design methodologies using Verilog and FPGA.

Output:

Learners will be able to design, simulate, and implement digital circuits, understand the end-to-end VLSI design flow, and appreciate the role of VLSI in advanced computing systems like smartphones, processors, and AI chips.

Chapter 2: Introduction to Digital World

2.1 Introduction & Number System

- Binary, Octal, Decimal, Hexadecimal: Conversion and representation.
- Signed Numbers: 2's complement, 1's complement.
- Use in digital logic: All digital systems use binary internally.

2.2 Logic Design

- Logic Gates: AND, OR, NOT, NAND, NOR, XOR, XNOR.
- Truth Tables and Timing Diagrams
- Universal Gates: NAND & NOR can implement all logic.

2.3 Boolean Algebra

- Rules and Theorems: Associative, distributive, DeMorgan's laws.
- Karnaugh Maps (K-Maps) for simplification.
- Use: Minimize logic for efficient circuits.

2.4 Combinational Circuits

- Circuits without memory.
- Examples:
  - Adders (half, full)
  - Multiplexers
  - Demultiplexers
  - Encoders/Decoders

2.5 More Combinational Circuits

- ALU: Arithmetic Logic Units
- Comparators: Compare binary values.
- Parity Generators/Checkers

2.6 Sequential Circuits

- Circuits with memory elements (flip-flops).
- Flip-Flops: SR, JK, D, T.
- Registers and Counters
- Clocking: Edge-triggered designs.

2.7 Switch Debounce

- Mechanical switches can produce multiple transitions.
- Debouncing: Filters out noise using:
    - RC circuit
    - SR flip-flop
    - Software logic

2.8 Timing Analysis

- Setup Time, Hold Time
- Propagation Delay
- Metastability: Occurs when signals violate setup/hold times.

Benefit:

Develop a strong understanding of how digital systems process and store information through number systems, logic gates, combinational and sequential circuits, and timing analysis—critical for any hardware or VLSI design.

Output:

Learners will be able to design, simplify, and analyze digital circuits such as adders, multiplexers, flip-flops, counters, and ALUs, and apply essential timing and debouncing techniques in real-world embedded or FPGA systems.

Chapter 3: Introduction to Semiconductors

3.1 Semiconductors

- Types: Intrinsic (pure), Extrinsic (doped)
- Doping: N-type and P-type
- Diodes and basic transistor operation

3.2 BJT (Bipolar Junction Transistor)

- Three terminals: Emitter, Base, Collector
- Use: Amplification and switching

3.3 FET (Field Effect Transistor)

- MOSFET: Widely used in VLSI
- CMOS: Uses both PMOS and NMOS
- Power efficiency: CMOS is low power, scalable.

3.4 FPGA Introduction and Architecture

- FPGA: Field-Programmable Gate Array
- Architecture:
    - CLBs (Configurable Logic Blocks)
    - LUTs (Look-Up Tables)
    - Flip-flops
    - Interconnects and I/O blocks

- Advantage: Reconfigurable hardware for prototyping or production.

Learning these concepts provides a strong foundation in electronics and digital design, enabling students to understand how modern electronic devices function at the hardware level. These skills are essential for careers in VLSI design, embedded systems, and hardware engineering.

Outputs/Benefits:

- Understand and design efficient semiconductor-based circuits, transistors, and amplifiers.
- Apply knowledge of BJTs and FETs for building logic gates and switching circuits.

Chapter 4: Digital Design with Verilog HDL

4.1 Typical Design Flow

- Steps:
  - RTL Coding (Register Transfer Level)
  - Simulation
  - Synthesis
  - Implementation on FPGA

Understand the complete RTL-to-FPGA implementation process.

Outcome: Build end-to-end digital systems.

4.2 Hierarchical Modelling

●Top-Down Design: Break system into modules.

●Module Instantiation: Allows reuse and clarity.

Learn modular, scalable design practices.

Outcome: Simplifies complex systems with reusable components.

4.3 Basic Concepts

Verilog Data Types: wire, reg, integer

●Operators: Arithmetic, relational, logical, bitwise

Benefits - Master data types and logical operators in Verilog.

Outcome: Write accurate, efficient digital code.

4.4 Modules and Ports

Module Definition: module name (input, output);
Ports: Directional I/O — input, output, inout

Understand Verilog module structure and I/O ports.

Outcome: Design well-structured digital blocks.

4.5 Gate-Level Modelling

- Use built-in logic gate primitives (and, or, not)
- Good for learning, but rarely used in practice.
- Learn basic logic gate usage for simple circuits.

Outcome: Understand digital logic foundations.

4.6 Dataflow Modelling

- Continuous Assignments using assign
- Boolean Expressions to describe logic

Use Boolean expressions for hardware logic.

Outcome: Design efficient combinational circuits.

4.7 Behavioral Modelling

- always and initial blocks
- If-else, case, loops
- Use for describing algorithmic behavior.

4.8 Combinational Circuits Design

- Without memory: Adders, multiplexers, decoders
- Sensitivity List: Use always @(*) to model combinational logic

4.9 Introduction to CMOS

- CMOS Design Rules
- Inverter Design
- Power Dissipation and Scaling
- Logic-level implementation and layout.

4.10 Sequential Circuits Design

- Flip-flop based design using Verilog
- Synchronous Design: All elements triggered by a clock

Finite State Machines (FSMs)

❖ Mastering these concepts helps students understand the entire digital design flow—from RTL coding and simulation to FPGA implementation. It enhances skills in writing efficient Verilog code, using hierarchical and modular design practices, and modeling logic through gate-level, dataflow, and behavioral approaches. Learners also gain knowledge of CMOS fundamentals, power-efficient logic, and both combinational and sequential circuit design. This comprehensive understanding prepares them to develop and implement scalable digital systems for real-world applications.

Benefits and Outputs:

- Understand end-to-end digital design flow, from RTL to FPGA implementation.
- Gain proficiency in Verilog coding, data types, operators, modules, and ports.
- Apply hierarchical modeling for modular and reusable circuit design.
- Learn different modeling styles: gate-level, dataflow, and behavioral.
- Design core digital components like adders, multiplexers, and FSMs.
- Understand CMOS design, layout, and power optimization.

- Build complete, scalable, and efficient digital systems for real-world use.

Year 2 :- Application & Project Phase:

– Year 2 is full of hands-on-experience on 6 live projects –

Basic Circuits Written in Verilog HDL, Simulated and Implemented on the FPGA

This project involves designing fundamental digital circuits such as adders, multiplexers, and counters using Verilog HDL. The circuits are first simulated using tools like ModelSim or Xilinx Vivado to verify functionality. After successful simulation, the designs are synthesized and implemented on an FPGA board (e.g., Xilinx or Altera). The goal is to understand RTL (Register Transfer Level) design, synthesis, and hardware implementation. The project builds a foundation for larger system designs. Key concepts include clocking, combinational and sequential logic, and timing analysis. It helps bridge theoretical digital logic with practical FPGA deployment.

Sequence Detector

This project focuses on developing a Finite State Machine (FSM) using Verilog to detect a specific binary sequence (e.g., 1011) from a serial input stream. It involves designing states, transitions, and output logic. Simulation is done to validate the behavior under all input conditions. The design is then synthesized and implemented on an FPGA to visualize real-time output via LEDs or 7-segment displays. Sequence detectors are widely used in digital communication and pattern recognition. Tools used include Vivado, ModelSim, and FPGA boards. The project enhances FSM design and HDL coding skills.

Traffic Light Controller

This project simulates and implements a traffic light system using Verilog HDL and FSM logic. It controls light sequences based on predefined timing or input signals like pedestrian requests or vehicle sensors. The controller is first tested in simulation, then deployed on an FPGA with LEDs to represent the lights. It introduces timing constraints, state-based design, and output control logic. The project helps students understand real-world automation systems using HDL. Tools: Xilinx Vivado, ModelSim, Nexys A7 FPGA Board (or equivalent). It's a practical example of hardware-based state machines.

Digital Design

The Digital Design project involves the implementation of combinational and sequential circuits using Verilog HDL. It includes modules like ALUs, registers, memory, and control logic for understanding digital system design. Simulation and synthesis are performed using industry-standard tools. Emphasis is placed on RTL coding, modular design, testbenches, and hardware debugging. It provides hands-on experience in building subsystems that serve as the foundation for CPUs or controllers. This project is critical for mastering the logic behind digital computing systems.

Image Processing on FPGA

This project demonstrates the use of FPGA hardware to perform real-time image processing tasks such as edge detection, filtering, or grayscale conversion. Images are captured or read into the FPGA, processed using custom logic written in Verilog or VHDL, and displayed on a monitor or screen. This involves understanding pixel-level operations, memory interfacing, and performance optimization. Image processing on FPGAs offers high speed and parallelism, making it suitable for surveillance, robotics, and medical imaging. Tools include MATLAB (for preprocessing), Vivado, and FPGA development boards with VGA or HDMI.

16-Bit RISC Processor

This project involves designing a simple 16-bit RISC (Reduced Instruction Set Computer) processor from scratch using Verilog HDL. It includes the creation of components such as ALU, register file, control unit, instruction decoder, and memory interface. The processor supports a limited but efficient instruction set for executing basic programs. The design is simulated, tested using custom instructions, and implemented on an FPGA. The goal is to understand microprocessor architecture, datapath design, and instruction execution cycle. It's a capstone project that showcases complete system-level digital design. Tools: Xilinx Vivado, ModelSim, and FPGA board (e.g., Basys 3).

Year 3 – Placement & Internship Phase:

1. **Guaranteed Internship Phase**

- In Year 3, Coincent guarantees an internship with partner companies. The internship includes a formal Internship Offer Letter and a Completion Certificate upon successful completion.
- This is part of 3 year Program "Industrial Training + Internship" model — It covers live classes, mentorship, and project work, but the internship phase itself is completely complimentary.

2. **Structured Placement Preparation**

- Coincent supports students in portfolio-building with multiple completed projects (typically around 8) and Microsoft-aligned certifications .
- Coincent provides mock interviews, resume reviews, and training for HR and technical rounds — all aimed at preparing you for real-world hiring.

3. **Final Take**

- Coincent's 3rd year transforms theory into practical experience through a guaranteed internship, builds a robust credentials portfolio, and equips you with placement-ready skills via mock interviews and resume prep. If you're in your 4th year, this phase sets you on a clear trajectory from "training" to "hired."

Top VLSI Job Roles

The **top VLSI Job Roles** that are trending and have high potential for growth:

Role	Key Focus
1. RTL Design Engineer	Writes Register Transfer Level code (using Verilog/VHDL) for digital logic circuits.
2. Physical Design Engineer	Handles floorplanning, placement, clock tree synthesis (CTS), and routing of ICs.
3. Design Verification Engineer	Verifies RTL functionality using simulation, UVM, and testbenches.
4. DFT Engineer (Design for Test)	Adds test structures like scan chains, MBIST to ensure chip testability.
5. STA Engineer (Static Timing Analysis)	Analyzes and ensures timing closure across the entire chip.
6. Analog Layout Engineer	Designs layout for analog blocks (ADC, DAC, PLLs) with precision and minimal noise.
7. Mixed-Signal Design Engineer	Works on systems that combine digital and analog elements, like SERDES or PHY blocks.
8. FPGA Design Engineer	Implements and tests designs on FPGA boards for prototyping or low-volume products.
9. EDA Tool Developer	Builds or extends Electronic Design Automation tools (Cadence, Synopsys, etc.).
10. SoC Integration Engineer	Integrates various IPs and subsystems into a complete System-on-Chip.