Coincent 3-Year Program in Very-Large-Scale Integration
Partnered by Apsis Solution

Year 1: Live Industrial Training – Build Your Foundation

Gain hands-on industry exposure from day one with 2.5 months of live training in a professional environment. Learn the latest tools and technologies through skill-focused sessions, guided by expert mentors from the industry

Very-Large-Scale Integration Curriculum

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

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.

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.

Chapter 4: Digital Design with Verilog HDL

4.1 Typical Design Flow

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

4.2 Hierarchical Modelling

- **Top-Down Design**: Break system into modules.
- **Module Instantiation**: Allows reuse and clarity.

4.3 Basic Concepts

- **Verilog Data Types**: wire, reg, integer
- **Operators**: Arithmetic, relational, logical, bitwise

4.4 Modules and Ports

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

4.5 Gate-Level Modelling

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

4.6 Dataflow Modelling

- **Continuous Assignments** using assign
- **Boolean Expressions** to describe logic

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)

Year 2: Real-Time Projects – Apply What You've Learned

Transform your knowledge into real-world experience by working on 8 industry-level projects that build your technical and professional skills. Each project enhances your portfolio, strengthening your resume and showcasing your practical abilities. You'll also collaborate in teams, gaining valuable experience in communication, teamwork, and project management—just like in a real work environment.

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.

I
mage 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:

In the 3rd year of Coincent's program, students are guaranteed an internship with partner companies, complete with a formal Internship Offer Letter and a Completion Certificate upon successful completion. This internship is a complimentary part of the 3-Year "Industrial Training + Internship" model, which also includes live classes, expert mentorship, and hands-on project work. This phase bridges academic learning with real-world application, providing students with valuable professional exposure before graduation.

Coincent also offers structured placement preparation to ensure students are job-ready. This includes portfolio building through 8 real-time projects, certifications aligned with Microsoft standards, and dedicated training for interviews. From mock interviews to resume reviews and HR/technical round prep, every element is designed to transition students from classroom learning to career success. By the 4th year, students are equipped not just with knowledge, but with experience, credentials, and confidence to enter the workforce.