FPGA Programming (Verilog/VHDL) Track

Comprehensive Design & Deployment Skills for Digital Systems

Presented by: Sapiens AI Team

Source: Image Source: FPGA Design - Your Ultimate Guide | Ignys Ltd

Program Overview & Core Topics

Our Program Goal

Our primary objective is to equip participants with the skills to **Design & Deploy Digital Systems on FPGAs.**

FPGA Architecture

FPGA Architecture & Applications

Explore the fundamental building blocks of FPGAs (CLBs, LUTs, BRAMs) and understand their application in prototyping, custom computing, and signal processing.

Source: MathWorks, CodiLime

Verilog & VHDL

Introduction to Verilog & VHDL

Master Hardware Description Languages to describe electronic circuits. Learn syntax, data types, and levels of abstraction for faster, more accurate design.

Source: Wikipedia, Cadence

Logic Design

Combinational & Sequential Logic

Delve into the core principles of digital logic, implementing circuits where outputs depend on current inputs (combinational) and past states (sequential).

Source: Digital Design Principles

State Machines

State Machines & Clock Management

Design robust systems by mastering FSM theory (Mealy/Moore) and effective clock management to ensure synchronized, precisely-timed operations.

Testbenches

Testbenches & Simulation

Learn to create comprehensive testbenches to rigorously verify digital designs, validate circuit behavior, and ensure correctness before deployment.

Mini Project

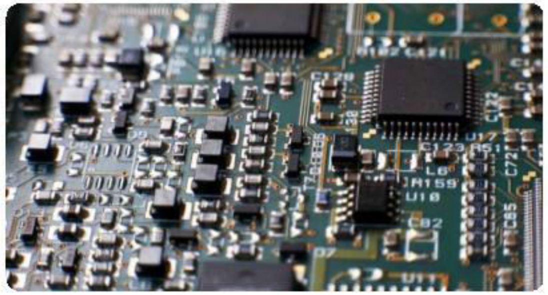
Hands-On Mini Project

Apply all acquired knowledge by designing and implementing a Digital Stopwatch or an 8-bit Calculator on a physical FPGA board.

Phase 1: FPGA & HDL Fundamentals (Month 1)

Week 1: FPGA Architecture & Digital Logic

FPGA Fundamentals: Explore Field-Programmable Gate Arrays as electronic devices with customizable logic circuitry, serving as programmable hardware.

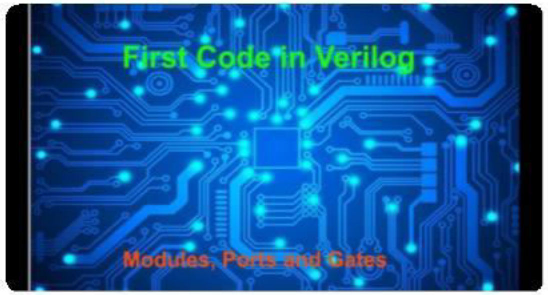


Core Concepts: Review digital electronics, FPGA building blocks, and the design philosophy for these reconfigurable devices.

FPGA Architecture · Digital Logic · Programmable Hardware

Week 2: Intro to Verilog & VHDL

HDLs: Dive into languages like Verilog/VHDL used to describe electronic circuits for designing ASICs and programming FPGAs.



Syntax & Modeling: Understand core concepts and modeling techniques that enable faster, more accurate designs and robust verification.

HDL · Verilog · VHDL · Circuit Modeling

Week 3: Sequential Logic Design

Foundational Elements: Master designing flip-flops and registers, essential for circuits that depend on past states.

HDL Implementation: Learn

Week 4: Testbenches & Simulation

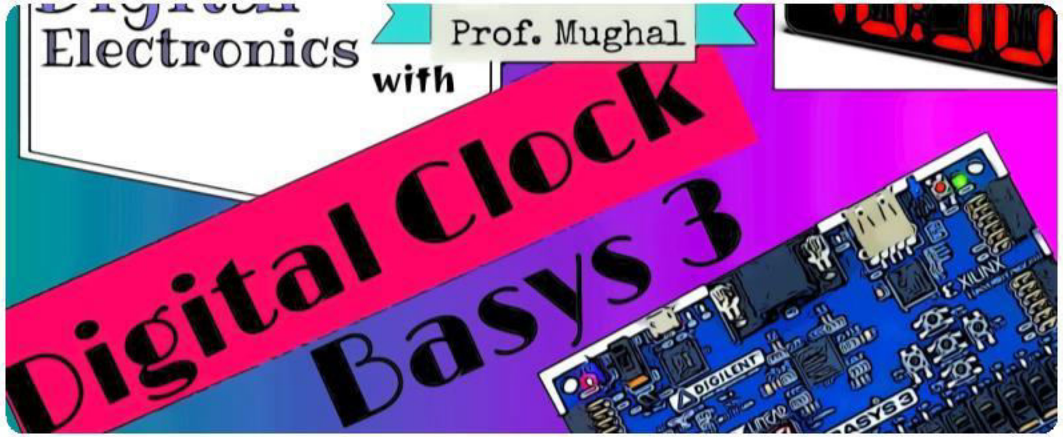
Verification Strategy: Learn to structure testbenches to provide stimulus, simulate behavior, and thoroughly verify HDL designs.

Simulation Techniques: Gain

Phase 1: Advanced HDL & FSMs (Month 2 - Weeks 5-6)

Week 5: Finite State Machines (FSM)

- Delve into the theory of FSMs, essential for designing complex control logic. Understand states, transitions, and outputs for both Mealy and Moore machines.
- Learn to translate FSM designs into robust HDL code, managing signals like `clk`, `rst`, and `enable` for precise control logic implementation.



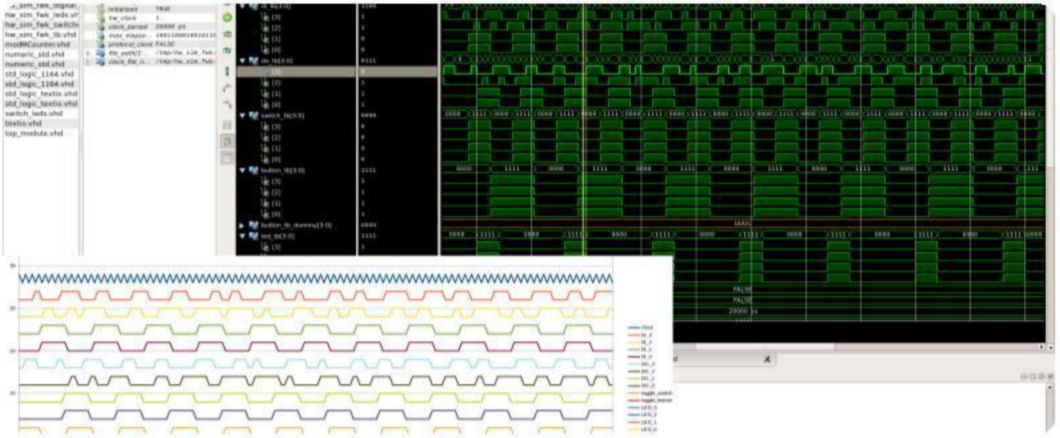
FSM Design

State Machines

Control Logic

Week 6: Advanced Verification

- Master advanced testbenches using procedural blocks (`initial`, `always`) and reusable `tasks` & `functions` for robust stimulus generation.
- Explore conceptual assertions to formally verify design properties and use code coverage metrics to ensure the completeness of your test suite.



Advanced Testbench

Verification

Tasks & Functions

Phase 1: FPGA Specifics & Planning (Month 2 - Weeks 7-8)

Advancing from fundamental HDL, Weeks 7 and 8 dive into critical FPGA specifics and meticulous planning, crucial for **optimizing performance** and ensuring **successful hardware implementation.**

Week 7: FPGA Resource Utilization & Optimization

Resource Management: Deep dive into understanding and optimizing FPGA hardware resources, including LUTs, FFs, BRAMs, and DSPs for efficient design.

Performance Trade-offs: Analyze the critical relationship between **Area vs. Speed**, and how design choices impact physical size and operational frequency. Includes Memory Inferencing.

Timing Analysis: Master concepts of **Setup/Hold Times**

Week 8: Clock Management & I/O Planning

Clock Management Units: Explore on-chip **PLLs** and **MMCMs** to generate, distribute, and modify clock signals for stable, high-frequency operation.

I/O Planning: Understand the principles of pin assignments and I/O standards, crucial for interfacing the FPGA with external components and ensuring signal integrity.

Project Initialization: Begin **Mini-Project Scoping** and

Phase 2: Capstone Project - Design & Development (Month 3 - Weeks 9-10)

Month 3: Capstone Project Kick-off

The Capstone Project is the culmination of all learned skills, requiring participants to apply comprehensive FPGA design principles to a real-world challenge.

Project Options: Choose to design and implement either a robust **Digital Stopwatch** or a fully functional **8-bit Calculator** on a target FPGA board.

Capstone Project | Digital Stopwatch | 8-bit Calculator | FPGA Development

Team & Architecture

Team Dynamics: Form collaborative project teams, fostering effective communication.

Specification & Breakdown: Meticulously define project requirements and create a top-down architectural breakdown into interconnected HDL modules.

HDL Implementation

Code Translation: Translate designs into clean, synthesizable Verilog/VHDL code.

Best Practices: Emphasize modularity, reusability, and adherence to established coding standards.

Unit Testing & Simulation

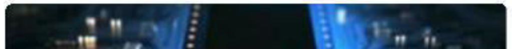
Verification Strategy: Implement comprehensive unit testbenches for each HDL module.

Functional Validation: Conduct thorough simulations to validate correctness and identify corner cases early.

Synthesis & Initial P&R

Hardware Mapping: Execute synthesis to translate HDL into a gate-level netlist for the target FPGA.

Physical Implementation: Perform initial Place & Route, arranging components on the FPGA fabric.



Phase 2: Capstone Project - Implementation & Debug (Month 3 - Week 11)

Seamless System Integration & Bitstream Generation

Top-Level Integration:

Bringing together all pre-verified sub-modules into a cohesive top-level design, ensuring proper hierarchical structure and signal integrity across module boundaries.

Comprehensive Constraint File Generation (XDC/SDC):

Developing detailed constraint files (XDC/SDC) to guide the synthesis and routing tools, specifying timing requirements, pin assignments, and resource usage for optimal performance.

Full System Simulation & Bitstream Generation:

Performing a final, comprehensive simulation to verify end-to-end functionality, followed by generating the FPGA bitstream for hardware implementation.

On-Board Debugging & Iterative Refinement

On-Board Implementation:

Loading the generated bitstream onto the target FPGA board and performing initial verification of basic I/O functionality and communication.

Hardware Debugging with ILA/Signaltap:

Utilizing on-chip debuggers like Xilinx ILA or Intel SignalTap to capture and analyze internal signals in real-time, identifying discrepancies between simulation and hardware behavior.

Troubleshooting & Iterative Design Refinement:

Systematically isolating hardware bugs or timing violations, then implementing corrective changes in HDL and re-verifying in a continuous loop until all project requirements are met.

Project Showcase & Career Launchpad (Month 3 - Week 12)

Final Project Presentation & Live Demonstration

Participants will present their Capstone Projects, demonstrating the culmination of their comprehensive FPGA design skills. This includes a live demonstration on a target **FPGA Board**, which serves as proof that the "electronic device that includes digital logic circuitry you can program to customize its functionality," can bring theoretical designs to tangible hardware. Key learnings in problem-solving and Verilog/VHDL are highlighted.

Capstone Project Live Demo FPGA Board

Comprehensive Project Documentation

Each team submits a professional report detailing their project from initial design specifications to final testing results. This emphasizes industry best practices for technical documentation, including architectural diagrams, HDL code structure, testbench methodologies, and performance analysis, ensuring project reproducibility.

Technical Documentation Project Report

Career Development Workshops

Hands-on sessions to craft impactful resumes and optimize **LinkedIn** profiles for hardware engineering roles. Practical **mock interviews** provide constructive feedback, honing communication and problem-solving skills for real-world job applications in the semiconductor and embedded systems sectors.

Resume LinkedIn Mock Interviews

Networking & Graduation

An exclusive networking event connects participants with leading