

Advanced Verilog for ASIC and FPGA Design

HDL programming, verification and timing trade-offs

FA-0555 · 2 days

Software Development

An advanced HDL programming workshop on synthesizable design, verification, state machines, timing constraints and ASIC/FPGA trade-offs.

Introduction

This two-day workshop is for participants who already write basic Verilog and understand digital circuits. It develops selected advanced HDL programming and verification techniques for ASIC and FPGA designs through focused examples and design discussions.

The course covers efficient synthesizable coding, hierarchical designs, testbenches, selected SystemVerilog verification features, and introductory timing analysis. Area, speed and power trade-offs are explored in illustrative designs; two days does not replace a complete chip-design or sign-off programme.

Learning outcomes

The course teaches participants to:

- Compare HDL coding and optimisation choices for ASIC and FPGA targets.
- Distinguish synthesizable design constructs from simulation-only usage, using the selected tool's supported subset.
- Organise state machines and hierarchical HDL projects and review testbench structures.
- Use selected assertion-based verification features where supported by the simulator.
- Read basic SDC constraints and static timing reports and discuss area, speed and power trade-offs.
- Debug illustrative designs and explain the rationale for design decisions.

Prerequisites

- A foundational knowledge of Verilog and its basic constructs.
- Prior experience or familiarity with digital circuit design.

A laptop with a compatible Verilog/SystemVerilog simulator and synthesis environment. Required tools, supported features and access arrangements should be confirmed before the workshop.

Training outline

1. Day 1 — Advanced HDL design and verification

Efficient digital design

- Optimization strategies for ASIC vs. FPGA.
- Efficient use of synthesizable constructs within the selected tool's supported subset.

Verification and validation

- Advanced testbench structures and methodologies.
- Selected SystemVerilog assertion-based checks and simulator support.

State machines and hierarchical projects

- Understanding and implementing state machines.
- Designing and managing large-scale hierarchical projects.

Tasks, functions and modelling

- Tasks and functions: reusable modelling, synthesizable usage where supported, and simulation-only behaviour.
- Selected SystemVerilog enhancements for high-level modelling and verification; distinguish language features from tool support.

2. Day 2 – Timing, FPGA trade-offs and design review

Timing and constraints

- Introduction to SDC (Synopsys Design Constraints).
- Basics of static timing analysis and its significance.

FPGA architecture and optimisation

- Exploration of FPGA architectures and design considerations.
- Strategies for area, speed, and power optimization.

Illustrative design reviews

- Illustrative ASIC/FPGA examples: design decisions and trade-offs.
- Strategies for debugging and validating large-scale designs.

Design discussion and questions

- Discuss ASIC/FPGA coding and verification challenges using illustrative scenarios.
- Open Q&A to address individual queries and challenges.