Intel® FPGA Technical Training

FPGA for Software Engineers

Course Description

This course closes the gap between hardware and software engineers by providing the software engineer all the necessary FPGA concepts and terms.

The course provides deep overview about the FPGA market, development tools, design languages, and digital design considerations.

The course integrates also practical labs where software engineers can get experience with the FPGA design process.

The course starts with an overview of the programmable devices and their capabilities, continues with a short study of the VHDL language, writing simple test-bench and using simulation tool. The course also describes the synthesis and Place & Route processes, FPGA programming on board, and the verification process.

The course covers the new trends today like embedded soft and hard processors in FPGA (including ARM), DSP process in Simulink and Matlab, IP integration, multiple clock domain and synchronization circuits design issues, embedded logic analyzer and timing constraints.

Course Duration

5 days

Goals

1. Become familiar with programmable logic devices and their features
2. Understand the design process from specification up to FPGA programming and final verification on board
3. Implement combinational and sequential logic
4. Write test-benches
5. Become familiar with the synthesis and Place & Route processes
6. Understand digital design considerations
7. Become familiar with embedded processors design flow in FPGA
8. Become familiar with DSP design flow in FPGA with Matlab
9. Become familiar with debug/verification process and tools for FPGA
10. Integrate IP in FPGA design

Intended Users

Software and system engineers who develop projects that use FPGAs/CPLDs, without previous FPGA background

Previous Knowledge

None

Course Material

1. Simulator: Modelsim
2. Synthesizer and Place & Route: Quartus Prime
3. Demonstration on Intel Evaluation board
4. Course book (including labs)

Table of Contents

Day #1

- **Introduction to Programmable Logic Devices**
  - CPLD architecture and design consideration
  - FPGA architecture
    - LUT
    - FF
    - PLL
    - DSP Block
    - Embedded RAM
    - Embedded Processor
    - FPGA Programming
  - Examples of the newest FPGAs in the market

- **Hard & Soft Processors Embedded FPGA**
  - Introduction to soft processor
  - SignalTap II logic analyzer
  - SOPC builder and Qsys flow
  - Introduction to SoC (FPGA + ARM)
  - Introduction to DSP design flow

- **Synchronous Vs Asynchronous Design**
  - Definition of synchronous and asynchronous systems
  - Skew, races and hazards in digital system
  - Power consumption considerations
  - Metastability

- **Logic Elements Review**
  - Combinational logic
    - LUT
    - Mux
    - Encoder

        - Decoder
        - Logical and arithmetic operators
        - ALU
    - Sequential logic
        - Flip-Flop
        - Setup & hold time
        - Register
        - Counter
        - Shift register
        - LFSR
        - PWM
        - FIFO
        - SDR and DPR
    - 3-state buffer

- **Finite State Machine**
    - FSM concept & block diagram
    - Mealy & Moore Models
    - State encoding
        - Sequential
        - Johnson
        - One Hot
        - Two Hot
        - Defined by user
        - Defined by synthesis
    - Handling the unused states
    - Reset & Fail Safe Behavior
    - Interactive State Machines
        - Unidirectional
        - Bi-Directional

Day #2

- **Introduction to VHDL Language**
  - VHDL history
  - Digital design
  - FPGA design review
    - Simulation
    - Synthesis
    - Place & Route
    - Programming
    - Verification
  - Advantages of VHDL
  - Simulation & Synthesis
  - Demonstration the whole process on board

- **VHDL Basic Structures Overview**
  - Entity
  - Component
  - Architecture
  - Process
  - Functions & Procedures
  - Package & Package Body
  - Library
  - Configuration
  - Top down design

- **VHDL Design Units – Building a Hierarchy**
  - Building MUX from its primitives
  - Port map
  - Test bench and simulation

- **Sequential Processing**
  - Process structure
  - Concurrent signal assignment versus process
  - Examples of combinational and sequential processes

- **Introduction to Synthesis**
  - The synthesis process
  - Inference from simple CSA
  - RTL and technology views
  - Inference from conditional statements
  - Inference from relational and arithmetic operators
  - Inference from loop statements
  - Creating Flip-Flops, counters, RAM, state machines

- **Practical labs**

Day #3

- **Digital Design Considerations**
  - Clock skew
  - Multiplexed clocks
  - Maximum clock rate
  - Introduction to pipeline
  - Jitter
  - PLL/DLL purpose and functionality
  - Misuse of gated clocks
  - Misuse of generated clocks
  - Misuse of asynchronous signals
  - Multi-clock domain systems
  - Metastability and MTBF analysis
  - Synchronization circuits
  - FIFO design
  - Synchronous and asynchronous reset
  - Reset synchronization circuits

- **IP**
  - What is IP?
  - Configuring your IP
  - Buy versus design IP
  - Soft and hard IP

- Integrating IP to the design
- Synthesizing IP and constraints
- Verifying IP functionality through simulation

- **Static & Dynamic Timing Analysis**
  - Introduction to static timing analysis methodology
  - Writing constraints with SDC file
  - Generating reports
  - Analyzing results
  - Performing timing simulation

Day #4

- **Debug Techniques**
  - SignalTap embedded logic analyzer
  - In system content memory editor
  - Signal probe
  - Logic analyzer interface (LAI)

- **FPGA Design Optimizations**
  - Incremental compilation
  - LogicLock
  - Synthesis optimizations
  - Place & Route optimizations
  - Coding style optimizations
  - Multiple place and route
  - Design space explorer
  - Power consumption optimizations

Day #5

- **Embedded Processors**
  - Embedded processor design flow with NIOS II and Qsys
  - Embedded processor design flow with SoC and Qsys
  - Developing software for FPGA embedded processors
  - Embedded system design considerations
  - Migrating tasks between FPGA and processor
  - Debugging the system (HW&SW)