Intel® FPGA Technical Training

Intel FPGAs Architecture & Design

Course Description

This course provides all theoretical and practical know-how to design programmable devices of Intel with Quartus Prime software.

The course combines 50% theory with 50% practical work in every meeting. The practical labs cover all the theory and also include practical digital design.

The course starts with an overview of the current programmable logic devices and their capabilities, continues with an in-depth study of the architecture and its various features, coding for synthesis, using the synthesize and Place & Route tools.

The course also touches upon digital design considerations, adding constraints, integrating IP, timing simulation, FPGA programming, and verification.

In addition, the course covers the different tools in Quartus Prime software that assist in design and analysis process.

Course Duration

5 days

Goals

1. Become familiar with Intel FPGA and CPLD families and their capabilities
2. Understand the design process from specification up to FPGA programming and final verification on board
3. Integrate IP into the design
4. Write constraints for clocks, reset, I/O, and exceptions
5. Optimize the design through the synthesis and Place & Route tools
6. Analyze results and solve design problems
7. Program the FPGA and verify functionality
8. Understand the coding style considerations for synthesis in Intel FPGAs

Intended Users

Hardware engineers who program with VHDL/Verilog languages and would like to be specialized with Intel FPGAs and Quartus Prime software

Previous Knowledge

VHDL/Verilog beginners and advanced users who are new to Intel FPGAs.

Course Material

1. Simulator: Modelsim
2. Synthesizer and Place & Route: Quartus Prime
3. Course book (including labs)

Table of Contents

Day #1

- **Introduction to Programmable Logic Devices**
  - CPLD architecture and design consideration
    - Examples: MAX V, MAX II
  - FPGA architecture (LUT, FF, PLL, DSP Block, Embedded RAM, Embedded Processor, Programming)
    - Examples: Stratix V, Arria V, MAX 10, Arria 10, SoC

- **Introduction to Quartus Prime**
  - Subscription Vs Web edition
  - Project creation and the main windows in Quartus Prime
  - Quartus Prime project files and folders
  - Project management (archive, restore, copy, revisions)
  - Design entry methods (schematic, HDL, EDIF)
  - Synthesis and P&R
  - Programmer

- **RTL Coding Guidelines for Quartus Prime Integrated Synthesis**
  - Synthesis directives and attributes
  - Fixed output registers
  - RAM inference
  - Latch inference
  - Combinational loops
  - Finite state machine coding styles

Day #2

- **Example: Cyclone V Architecture Deep Dive**
  - Logic elements and logic array blocks ALM modes, LAB topology and control signals)
  - Memory blocks (control signals, parity bit support, byte enable support, packed mode, address clock enable, mixed width, asynchronous reset, memory modes: FIFO, ROM, Shift register, DPR, SDR, clocking modes, design considerations)
  - Embedded DSP blocks (architecture, operational modes)
  - Clock networks and PLLs (GCLK, RCLK, PCLK networks, PLLs hardware, programmable bandwidth, phase shift implementation, PLL cascading, PLL reconfiguration)
  - I/O interfaces (I/O element features: slew rate control, programmable current strength, open drain output, bus hold, programmable pull-up resistor, programmable delay, PCI clamp diode, on chip termination, I/O standards, I/O banks, clock pins functionality, high speed I/O interface, true differential output buffer support, design guidelines)
  - External memory interfaces (data and data clock/strobe pins, optional parity, DM, and ECC pins, address and control pins, memory clock pins, DDR input and output registers, OCT with calibration, PLL)
  - Configuration and remote system upgrade (configuration requirements and process, AS/AP/PS.FPP.JTAG configurations, enabling remote update, remote system upgrade mode, dedicated remote system upgrade circuitry)
  - JTAG boundary scan testing (IEEE Std. 1149.6, BST operation control, I/O voltage support, BSDL)

- Power requirements (external power supply, hot socketing, POR circuitry)

- **IP Megafunctions**
  - MegaCore, AMPP, OpenCore IPs
  - MegaWizard plug-in manager tool
  - Generated files

Day #3

- **Compilation Process**
  - Processing options
  - Compilation design flows
  - Viewing compilation results
  - Netlist viewers
  - State machine viewer
  - Chip planner
  - Resource property editor

- **Assignment Editor**
  - Synthesis and fitting control
  - Synthesis settings
  - Fitter settings
  - Assignment editor features
  - Design assistance

- **I/O Management**
  - Pin planner
  - I/O assignment analysis
  - Using synthesis attributes in HDL
  - Live I/O checking
  - Swappable pins
  - SSN analyzer
  - Transferring I/O assignments to PCB tools

- **Modeling Finite State Machines and Memories**
  - State machine editor
  - From SMF to HDL
  - Memory editor
  - Create memory initialization file
    Using memory file in design

Day #4

- **Introduction to Timing Analysis**
  - TimeQuest tool overview
  - Basic steps to using TimeQuest (generate timing netlist, enter SDC constraints, update timing netlist, generate timing reports)
  - Using TimeQuest in Quartus II flow
  - Timing analysis basics (Launch Vs Latch edges, setup and hold times, data and clock arrival time, data required time, setup and hold slack analysis, I/O analysis, recovery and removal, timing models)

- **Timing Reports**
  - Reporting in Quartus II Vs reporting in TimeQuest
  - Custom, summary and diagnostic reports
  - Clock transfer, datasheet, Fmax reports
  - Slack histogram report
  - Detailed slack/path report, further path analysis

- **Introduction to Timing Constraints**
  - Importance of constraining
  - Enter constraints
  - SDC netlist terminology
  - Collections

- **SDC Timing Constraints**
  - Internal and virtual clocks
  - Generated clocks (inverted clocks, phase shifted clocks
  - PLL clocks and derive_pll_clocks Altera SDC extension
  - Automatic clock detection and creation

- Non ideal clock constraints (Jitter, latency on PCB)
- Common clock path pessimism removal
- Checking clock constraints
- Report clocks

- **SDC Timing Constraints for I/O**
  - Combinational I/O interface constraints (max & min delay constraints)
  - Synchronous inputs constraints (setup and hold time calculations, set_input_delay max & min, set_output_delay min& max, when to use each constraint, output pin load, signal integrity metrices)
  - Checking I/O constraints (report SDC, report unconstrained path, report ignored constraint)

- **SDC Timing Constraints for Asynchronous paths and Exceptions Overview**
  - Asynchronous paths constraints
  - False path constraints
  - Multi-cycle constraints

Day #5

- **Programming the FPGA**
  - Programming flow
  - Programmer GUI (hardware setup, JTAG settings, JTAG chain debugger tool)
  - Programming and configuration modes
  - Design security keys
  - Secondary programming files
  - Flash loaders

- **Introduction to FPGA Verification**
  - Functional versus gate level simulation
  - Simulating ALTERA IP cores
  - Power estimation and analysis
  - System debugging tools overview (in-system sources and probes, in-system memory content editor, signal probe, SignalTap II, LAI)

ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il | פ: 077-4702742 | ט: 052-5816791 | רח' הכרמל 9, ת.ד. 803 כ"ס 4410801