Intel® FPGA Technical Training

Designing with Intel SoC for Hardware Engineers

Course Description

This course provides all theoretical and practical know-how to design Intel SoC devices under Quartus Prime software.

The course combines 50% theory and 50% practical work on Terasic DE1-SoC evaluation board.

The course starts with Intel SoC families overview and their capabilities, continues with deep methodic training of the SoC architecture.

The course teaches the HPS architecture and its building blocks, how to manage SoC system, how to configure system based on SoC, how to transfer data through the Bus system and internal interconnect, how to connect external memories, how to build a system with Qsys, how to handle interrupts and how to use efficiently pin muxing.

The second part of the course focuses on practical use of simulation models (BFMs), creating SoC test-benches, and performing different boot processes with and without operating system.

The course ends with SoC debug interfaces overview, how and when to use them, cross-triggering between CPUs and FPGAs and how to use the system console manager.

Course Duration

3 days

Goals

1. Become familiar with Intel SoC families, their capabilities and when to use them
2. Understand SoC design hardware and software flow from specification to programming and final verification on the board
3. Integrate IPs into the SoC design (also custom peripherals)
4. Configure the SoC system (clocks, PLLs, Resets, Peripherals)
5. Use efficiently the Qsys tool
6. Use Bus Functional Models (BFMs) to simulate SoC behavior
7. Program and use SignalTap II to verify the SoC functionality
8. Understand and choose the right Boot scheme
9. Handle Interrupts using the Generic Interrupt Controller (GIC)
10. Connect external memories to the SoC

Intended Users

Hardware and system engineers who would like to design with Intel SoC technology

Previous Knowledge

Quartus Prime software

Qsys

SignalTap II Embedded Logic Analyzer

Course Material

1. Simulator: Modelsim
2. Synthesizer and Place & Route: Quartus Prime
3. Terasic DE1-SoC Evaluation board
4. Course book (including labs)

Table of Contents

Day #1

- **System on Chip (SoC) Overview**
  - Intel SoC the best of both worlds
  - System-level benefits of SoC
  - SoC device portfolio and key features
  - Development boards available
  - Hardware and software development perspectives
  - System development flow with Qsys and DS5

- **HPS Overview**
  - HPS IP features
  - HPS block diagram
  - Cortex-A9 overview
  - HPS memory views
  - Default detail address map
  - Generic Interrupt Controller (GIC) overview

- **System Management**
  - System management overview
    - HPS input clocks and clock groups
    - FPGA interface clocks
  - HPS Clock Manager overview
    - HPS Clock Manager – PLLs (main, peripheral, SDRAM)
    - Flash controller clocks
    - HPS entry/exit 'Safe Mode'
  - SoC device reset pins
    - Reset Manager overview (cold/warm/debug)
    - Reset Manager integration
  - FPGA Manager overview
    - HPS configuring FPGA fabric
  - System Manager overview
    - I/O features

        - Managed peripherals
    - Scan Manager overview

- **Interconnects**
    - Interconnect overview
    - Level 3 interconnect up/downsizing
    - AXI bridges architecture
    - Global Programmers View (GPV)
    - High performance paths
    - FPGA-to-HPS bridge drawbacks
    - Level 4 peripheral bus interconnect

- **Peripherals**
    - HPS peripherals overview
    - On-chip ROM features
    - On-chip RAM features
    - SDRAM controller features
    - HPS SDRAM controller configuration
    - Maximizing SDRAM performance
    - Considerations when accessing HPS SDRAM from FPGA

- **Direct Memory Access Controller (DMA)**
    - DMA overview
    - DMAC data transfer features
    - DMAC peripheral flow control features
    - HPS DMA capabilities
    - When to use and not to use HPS DMA

Day #2

- **Hardware Design Flow**
  - Typical design flow
  - Qsys tool
  - Automatic interconnect generation
  - Create Quartus Prime project for SoC device
  - Start a new system in Qsys
  - Add IP to Qsys system
  - Add custom components
  - Methods to connect components
  - HPS in Qsys
  - HPS-Nios II system block diagram
  - Generate completed system
  - Hardware/software design flow overview
  - Generated software handoff files

- **Avalon and AXI Standards**
  - Qsys-supported standard interfaces
  - Advantages of using standard interfaces
  - Avalon-MM interfaces
  - AXI overview
  - AXI features
  - Handshake examples
  - AXI write transaction
  - AXI read transaction
  - Component editor – AMBA support
  - AXI specification
  - Qsys memory-mapped packet format
  - Which protocol to choose: Avalon or AXI?

- **HPS Component Configuration**
  - Hard processor system component
  - General options & Boot control
  - Events
  - General Purpose I/O (GPIO)
  - Debug APB

- System Trace Macrocell
- Cross Trigger Interface (CTI)
- Trace port interface
- Boot from FPGA
- AXI bridges
- FPGA-HPS bridge interfaces
- Accessing HPS memory from FPGA
- FPGA-to-HPS SDRAM interface
- Resets
- DMA control
- Interrupts
- GIC overview (SGIs, PPIs, SPIs)
- Peripheral pin multiplexing
- HPS I/O muxing overview
- Ethernet
- Other peripheral options (QSPI, SPI master, UART)
- Pin usage & conflicts
- HPS pin assignments
- HPS clocks
- SDRAM embedded memory interface

❖ **LAB #1: Creating an ARM based SoC system using Qsys**

❖ **LAB #2: Add Custom Component to SoC System**

Day #3

- **HPS Simulation**
  - Bus Functional Models (BFMs)
  - Simulation flow
  - Slave component testing
  - Master component testing
  - HPS system testing
  - HPS simulation support – interfaces
  - Generate Testbench Qsys system
  - Testbench directory structure
  - Qsys Testbench system – HPS system
  - Writing the test – AXI BFM API overview
  - Testbench example
  - Using conduit BFMs
  - Run simulation script

- **SoC FPGA Configuration and Booting**
  - HPS boot stages
  - SoC configurations & Boot sequences
  - Boot schemes – independent
  - Boot schemes – FPGA first
  - Boot schemes – HPS first
  - HPS Power On/Reset
  - HPS Boot ROM
  - HPS preloader
  - HPS user Bootloader
  - HPS Linux OS start up
  - 'Bare Metal' programming
  - SoC Boot phases
  - HWLibs components

- **Hardware Debug**
  - Debug interfaces (JTAG, Ethernet)
    - SignalTap II Logic debug
    - System console
    - FPGA adaptive debugging

- System console overview
  - Usage examples
  - System console interfaces
  - System console GUI launch
  - System console services
  - Service types
- SignalTap II cross triggering
  - Cross triggering
  - Cross Triggering Interface (CTI)
  - Intel SoC debug architecture
  - Export CTI to custom hardware
  - SignalTap II configuration for cross trigger
- ARM DS-5 debugger
  - Debug perspective – registers view
  - Run debugger and SignalTap II Logic Analyzer

- **LAB #3: exercise the FPGA using the system console tool**
- **LAB #4: debugging hardware using SignalTap II Logic Analyzer**