HandsOn Training

Intel® FPGA Technical Training

Designing with Nios II Processor for Hardware Engineers

Course Description

This course provides all theoretical and practical know-how to design Intel FPGAs based on the Nios II soft processor under Quartus Prime software.

The course combines 50% theory and 50% practical work on Terrasic DE series evaluation board.

The course starts with Nios II processor system hardware development, Nios II soft core capabilities, and continues with deep methodic training of the Nios II architecture.

The course teaches the Nios II architecture and its memory, peripherals, how to manage SoC system, how to configure system based on Nios II, 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, how to develop software and ways for debugging.

The second part of the course focuses on appending custom instruction and custom components to enhance performance, use of simulation models (BFMs), and creating SoC test-benches.

The course ends with multi-Nios II systems (multicore) design, the design considerations, and how to debug multiple processors at the same time.

Course Duration

2 days

Goals

1. Become familiar with Intel Nios II processor, its capabilities and when to use it
2. Understand SoC design hardware and software flow from specification to programming and final verification on the board
3. Develop software for Nios II processor
4. Integrate custom components into the SoC design
5. Integrate custom instructions to a Nios II processor
6. Configure the SoC system (clocks, PLLs, Resets, cache, TCM, on-chip memory, off-chip memory, MMU/MPU, Peripherals)
7. Use Bus Functional Models (BFMs) to simulate SoC behavior
8. Use system console to debug hardware
9. Handle Interrupts using the Nios II internal or external interrupt controller
10. Design multi-Nios II processor systems

Intended Users

Hardware and system engineers who would like to design with Intel Nios II soft processor

Previous Knowledge

Quartus Prime software and Qsys

SignalTap II Embedded Logic Analyzer

ModelSim

Course Material

1. Simulator: Modelsim
2. Synthesizer and Place & Route: Quartus Prime
3. Terrasic Cyclone V GX Evaluation board
4. Course book (including labs)

Table of Contents

Day #1

- Nios II software build tools
- Creating new software projects
- Application and BSP projects
- Application and BSP from template
- Create BSP project by itself
- Create application project by itself
- Importing projects into Workspace
- Creating a new source file in GUI
- Importing source files to a project
- Creating linked resources
- Setting project properties
- Application project properties
- Setting compiler flags (App and BSP)
- Add libraries to application project
- BSP project properties
- BSP editor
- Build properties for individual files
- Project directory structure
- .elf and system.h output files
- Program target hardware
- Running code on a target
- Run configurations window
- System ID peripheral check
- Nios II debugger
- Nios II debug perspective
- Debug windows
- Breakpoints
- Watchpoints
- Nios II instruction set simulator

- **System Verification with System Console**
  - What is system console?
  - Usage examples
  - System console operation requirements
  - System console interfaces
  - System console GUI
  - Command line interface
  - GUI interactive usage tips
  - System console services

- System console usage flow
- Master service type
- Dashboards
- Widgets
- Other system console uses

❖ **Lab #2: Build a Software Project for the Nios II Processor**

Day #2

- **RTL Simulation with the Nios II Processor**
  - RTL simulation for a Qsys system
  - Enable testbench generation
  - Qsys testbench output files
  - Building for an RTL simulation
  - Running an RTL simulation
  - "msim_setup.tcl" simulation script
  - Component simulation options

- **Custom Components**
  - What are custom components?
  - Example signal mapping
  - Component editor GUI
  - "hw.tcl" file
  - Custom component integration into Qsys system
  - Edit component parameters

- **Nios II Processor Custom Instructions**
  - What are custom instructions?
  - Block diagram of a system with custom instructions
  - Custom instructions support in Nios II development tools
  - Create custom instructions in Qsys
  - Add custom instruction to system
  - C language software interface
  - Assembly language interface
  - Why custom instructions?

- Floating point custom instructions
- Floating point CI macros
- Nios II custom instruction user guide
- Custom instruction vs. custom component
- Multi-cycle custom instructions
- Accelerating CRC example

- **Working with Development Boards**
  - Ensure unused I/O are tri-state
  - Flash memory configuration
  - Hardware configuration from Flash
  - Boot copier
  - Nios II Flash Programmer
  - Extra features
  - What if you have a custom board?
  - What if Factory Safe Flash Image overwritten?
  - Booting from the MAX 10 device on-chip Flash

❖ **LAB #3: Custom Components and Custom Instructions**

- **Application Examples for the Nios II Processor**
  - Component tradeoff
  - State machine replacement
  - Custom microcontroller
  - General purpose system controller
  - I/O processing controller
  - Off-loading existing CPU
  - Build multi-processor systems
  - Multi-CPU architectures
  - Shared memory, mutex, and mailboxes
  - Software for multi-processor systems
  - Launch group for multi-processor systems
  - Sending interrupts
  - Control plane and data plane

❖ **LAB #4: Build and Explore a Multi-Nios II CPU System**