HandsOn Training

Intel® FPGA Technical Training

Qsys System Integration Tool Design & Optimizations

Course Description

This course will teach you how to quickly build designs for Intel FPGAs using Intel's Qsys system-level integration tool.

You will become proficient with Qsys and expand your knowledge of the Quartus Prime FPGA design software.

You will learn how to quickly integrate IP and custom logic into a system.

Since Qsys makes design reuse easy through standard interfaces, we will examine the Intel Avalon-Memory Mapped and Streaming Interfaces as well as introduce the AMBA AXI interface standard from ARM.

The class provides a significant hands-on component, where you will gain exposure to tool usage as well as system and custom HDL component design.

Course Duration

3 days

Goals

1. Build digital systems in the Qsys tool
2. Integrate the files generated by Qsys into the Quartus Prime design flow
3. Create custom components with Avalon-MM and Avalon-ST interfaces and integrate them into your system
4. Test custom components and entire systems with the Avalon Verification Suite in the Modelsim-Altera
5. Perform in-system control & debugging with System Console
6. Optimize systems to maximize performance and help close timing
7. Customize components using Tcl
8. Exploit Qsys hierarchical capability to add flexibility & scalability to your design

Intended Users

FPGA engineers who would like to use Qsys to build simple & complex systems on chip

Previous Knowledge

Intel FPGAs architecture

Quartus Prime software

ModelSim

VHDL/Verilog

Course Material

1. Simulator: Modelsim
2. Synthesizer and Place & Route: Quartus Prime
3. Course book (including labs)
4. Terasic C5G Cyclone® V GX Starter Kit

P.O.B. 803 Kfar Saba 441080 ISRAEL | P +972-52-5816791 | F +972-77-4702742 | ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il

Table of Contents

Day #1

❖ **Introduction to the Qsys System Integration Tool**

- **What is Qsys?**
  - Traditional system design
  - Automatic interconnect generation
  - Qsys benefits
  - Target Qsys applications
  - Qsys vs SOPC Builder
  - SOPC Builder systems in Qsys

- **Qsys UI**
  - Component library
  - System contents
  - System inspector
  - Address map
  - Clock settings
  - Project settings
  - Generation
  - HDL example
  - Messages
  - Other useful Qsys commands

- **Using Qsys in FPGA Design Flow**
  - FPGA hardware design flow
  - Additional Qsys verification support
  - Qsys system generation

- **Qsys Files**
  - Qsys source files
  - Qsys output files

❖ **Lab #1: Introduction to Qsys Design Flow**

- **Introduction to Qsys Interconnect**
  - Qsys interconnect architecture
  - Qsys supported standard interfaces (Avalon, AXI)
  - Qsys interconnect implementation
  - Arbitration priority
  - Enables simultaneous multi-mastering
  - Qsys memory-mapped packet format
  - Packetized interconnect vs latency
  - NoC architecture

  - Master network interface
  - Slave network interface
  - Pipelining

- **Using Qsys IP**
  - Qsys standard interfaces for IP
    - Clock
    - Reset
    - Avalon-ST
    - Avalon-MM
    - Avalon-C
    - Avalon-TC
    - AXI

  - **Qsys IP**
    - Component library parameter editors
    - Basic components
    - Streaming components
    - Memory components
    - Tristate components
    - Bridge components
    - High-speed interface components
    - Processor components

❖ **Lab #2: Using Qsys IP**

Day #2

❖ **Custom Components, Processor Interfaces & Verification**

- **Creating Custom Components**
  - Custom components interconnect
  - Example custom components
  - Component editor

- **Processor Interfaces**
  - Interfacing FPGA with external processors
  - Memory mapped interface
  - High-speed serial interfaces
  - Good use of SPI/Avalon master bridge
  - Using the NIOS II processor
  - Using the SoC devices

❖ **Lab #3: Build a Data Path Hardware System in Qsys that Incorporates Custom Logic**

❖ **Lab #4: Add Control Plane to the Design and Test the System**

- **System Integration with Qsys**
  - Avalon verification suite
  - Application Programming Interface (API)
  - Avalon verification suite testbench
  - Clock source BFM
  - Reset source BFM
  - Avalon-MM BFMs & API
  - Avalon-ST BFMs & API
  - Avalon-MM/ST monitors, testbenches and API
  - AXI verification IP
  - Component verification system
  - System verification

❖ **Lab #5: Testing Custom Components with BFMs**

❖ **Lab #6: Testing the System with BFMs**

Day #3

❖ **System Console Debug & Hierarchy with Qsys**

- **System Verification with System Console**
  - What is System Console?
  - Usage examples
  - System Console interfaces
  - System Console GUI launch
  - Command line interface
  - Interactive usage tips

- **System Console services**
  - System Console usage flow
  - Qsys components for System Console
  - Linking sevice instances with Quartus II project
  - Linking Quartus project using Tcl script
  - JTAG debug commands
  - *master* service type
  - *monitor* service type
  - *bytestream* service type
  - Dashboards

❖ **Lab #7: Testing the System with System Console**

- **Utilizing Hierarchy in Qsys Designs**
  - Hierarchy in the Quartus II software
  - Hierarchy in Qsys
  - Scalable Qsys systems with hierarchy
  - Adding subsystems to component library
  - Add subsystems to system
  - Connecting systems together
  - Ways to create subsystems
  - Making changes to a subsystem
  - Bridging systems together with Avalon-MM pipeline bridges
  - Subsystem masters
  - Subsystem clocking
  - Subsystem parameterization
  - Tcl commands used in instance script

❖ **Lab #8: Qsys Hierarchy: Expand System to support four parallel video processing channels**