



# MAX 10 Webinar Labs Summary

## **Lab1 - First MAX1000 Design:**

This tutorial provides comprehensive information to help you understand how to create a simple Intel FPGA Design and run it on your MAX1000 board. This lab will not make you an expert, but at the end, you will understand basic concepts about Quartus Prime projects, such as entering a design using a schematic editor, compiling your design, and downloading it into the FPGA on your development board.

## **Lab 2 – RTL Simulation**

This tutorial provides comprehensive information to help you understand how to simulate your FPGA design in the ModelSim – Intel FPGA Edition simulator. Design simulation verifies your design before programming.

## **Lab 3 – Simple NIOS**

This tutorial provides comprehensive information to help you understand how to create a simple Nios soft processor system in an Intel FPGA and run it on your MAX1000 board. The Nios II processor is a soft intellectual property (IP) processor that you download (along with other hardware components that comprise the Nios II system) onto an Intel FPGA. At the end, you will understand basic concepts about Quartus Platform Designer (formerly Qsys), Software Build Tool and the basic development flow for the Nios II design flow.

## **Lab 4 – Embedded System**

This tutorial provides comprehensive information to help you understand how to create a software project for a Nios II processor system in an Intel FPGA and run the software project on your MAX1000 board. The Nios II processor core is a soft intellectual property (IP) processor that you download (along with other hardware components that comprise the Nios II system) onto an Intel FPGA. This tutorial introduces you to the basic software development flow for the Nios II processor.

## **Lab 5 – Internal Flash**

This tutorial provides comprehensive information to help you understand how to use the internal flash feature of the MAX10 and how to run it on your MAX1000 board. The Nios II processor is a soft intellectual property (IP) processor that you download (along with other hardware components that comprise the Nios II system) onto an Intel FPGA. At the end, you will know how to setup your Quartus project in order to boot your Nios processor from the internal flash memory.

## **Lab 6 – Signal Probe Debugging**

The Signal Probe feature allows you to route a user specified internal node to a top-level I/O without affecting the existing fitting in a design. Using the Signal Probe allows you to investigate internal device signals without performing a full compilation.

## **Lab 7 – SignalTab**

This tutorial provides comprehensive information to help you understand how to analyze and verify your Intel FPGA design within Quartus Prime software. This lab explains how to use Embedded Logic Analyzer Megafunction, and gives detailed, step-by-step procedures on how to set up and run Embedded Logic Analyzer.

## **Lab 8 – In-System Sources & Probes**

The In-System Sources & Probes Editor allows you to easily control any internal signal and provides you with a completely dynamic debugging environment. Traditional debugging techniques often involve using an external pattern generator to exercise the logic and a logic analyzer to study the output waveform during runtime. You can make the debugging cycle more efficient when you can drive any internal signal manually within your design. This feature can be especially helpful for prototyping your design, such as creating a virtual interface, emulating external data and monitoring, changing run time constant on the fly.

## **Lab 9 – Timing Closure**

This tutorial provides comprehensive information to help you understand how to set and use your FPGA design environment to ensure that your design meets the timing requirements. The TimeQuest timing analyzer is a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in a design using industry standard constraint, analysis, and reporting methodology.

## **Lab 10 – Authentication Flash**

This tutorial introduces the operation of authentication flash memory through a demo. This memory provides handshake between the controller and the memory to ensure enhanced security of the data transmitted between the two devices.

## **Bonus Labs**

### **NIOS Tutorial Part 1 – HW Setup of the NIOS II System**

In this lab you will generate a small uController system containing a Nios II CPU, RAM, some peripherals and the internal User-FLASH block as program storage. The program code will be copied from the User-FLASH into the internal memory at startup time.

### **NIOS Tutorial Part 2 - Expanding a NIOS II System**

In this lab you will expand the NIOS II system from part 1 with further peripherals.

## **VHDPlus Lab 1 – Overview of the VHDPlus IDE**

### **Attention! For this lab you need further 3<sup>rd</sup> party tools**

This tutorial provides comprehensive information to help you understand how to use the VHDPlus IDE to accelerate and simplify your FPGA development. You will learn how to create a working program for your hardware with a few clicks, visualize your received data, program with VHDL and our language VHDP, simulate the code with our assistant and you will learn how to add a customized NIOS processor to your design and program it inside the VHDPlus IDE.

### **AnalogMAX – Jupyter Demo**

#### **Attention! For this lab you will need a AnalogMAX board!**

This tutorial demonstrates the various features of AnalogMAX board. It visually displays the values of the interfaces on a graph such as accelerometer, analog-digital converter, smoke and temperature sensor.

### **Downloads & Resources:**

Intel Quartus Lite 18.1

<https://fpgasoftware.intel.com/18.1/?edition=lite&platform=windows>

VHDPlus IDE

<https://vhdplus.com/docs/getstarted#install-vhdplus-ide>

Lab Files, Drivers & Documentation

<https://1drv.ms/u/s!BAiJKI-8fIdh31adEx307wK7Vv-Z?e=NcDACA>