



OREN HANDS ON TRAINING & DEVELOPMENT LTD.  
20 YAIR ROZENBLUM ST. KFAR SABA 4464601 ISRAEL

# altera™ solution acceleration partner

---

## Designing with DSP Builder for Altera FPGAs

### Course Description

Learn the timing-driven Simulink® design flow to implement high-speed DSP designs. This course focuses on implementing DSP algorithms using the advanced blockset capability of DSP Builder—an interface between Quartus Prime software & MATLAB and Simulink from MathWorks.

You'll analyze & design your DSP algorithm using the DSP Builder advanced blockset in MATLAB & Simulink. You'll explore architecture & performance tradeoffs with system-level constraints.

Also you'll verify functionality & performance of generated hardware in the Quartus Prime software. Finally, you'll speed design time by incorporating ready-made ModelIP cores in your design.

The course includes extensive hands-on labs.

**Each attendee receives an official certificate from Altera and from Arm** (Exam must be passed).

### Course Duration

2 days



When innovation meets expertise...



**OREN HANDS ON TRAINING & DEVELOPMENT LTD.**  
20 YAIR ROZENBLUM ST. KFAR SABA 4464601 ISRAEL

## Goals

1. Implement DSP algorithms using Intel DSP Builder Advanced Blockset
2. Incorporate ModelIP and ModelPrim cores in a design
3. Explore design architecture and performance tradeoffs using system level constraints
4. Incorporate a DSP Builder Advanced Blockset model into Platform Designer subsystem
5. Verify the hardware performance and implementation in Quartus Prime software

## Intended Users

Hardware engineers who develop and work with DSP algorithms and computationally intense applications for Altera FPGAs that need high performance

## Prerequisites

- Familiarity with DSP fundamentals and design
- Familiarity with Altera Quartus Prime software
- Familiarity with MATLAB and Simulink from MathWorks
- Familiarity with digital modem design is helpful, but not necessary

## Course Material

1. Course book
2. Lab's handbook and lab files
3. Quartus Prime
4. MATLAB and Simulink version 2017b
5. DSP Builder for Altera FPGAs
6. Modelsim



When innovation meets expertise...

## Table of Contents

### Day #1

#### ❖ DSP Builder for Intel FPGAs

- DSP processing on FPGAs
- FPGA DSP blocks
- DSP Builder for Intel FPGAs
- FPGA design flow: traditional
- FPGA design flow: DSP Builder for Intel FPGAs
- Core technologies
- Advanced Blockset: high performance DSP IP
- Library is technology independent
- Datapath optimization for performance
- Custom IP generation
- Build custom FFTs from FFT Element library
- Filter and Waveform synthesis library
- Hardware inference from input datatypes
- Full support for super sample design
- ALU design folding improves area efficiency
- System-in-the-Loop hardware verification
- Generates reusable IP for Platform Designer
- System requirements

#### ❖ MATLAB Overview

- The Mathworks design environment
- MATLAB environment
- MATLAB variables
- MATLAB arrays
- Useful MATLAB plot commands
- Helpful built-in MATLAB functions
- MATLAB .m files
- Editing and running .m files
- MATLAB script .m file



When innovation meets expertise...



**OREN HANDS ON TRAINING & DEVELOPMENT LTD.**  
20 YAIR ROZENBLUM ST. KFAR SABA 4464601 ISRAEL

### ❖ **Simulink Overview**

- Simulink overview
- Simulating dynamic Simulink systems
- DSP Builder-related MATLAB licenses

### ❖ **Getting Started with DSP Builder**

- Starting MATLAB with DSP Builder for Intel FPGAs
- Starting DSP Builder in Simulink
- Create new DSP Builder model with model wizard
- Example new model top-level testbench
- Design hierarchy
- Navigating the model
- Design configuration blocks
- Control block: general settings
- Control block: clock and reset settings
- Control block: testbench settings
- Device block
- Model primitive blocks
- Model IP blocks
- Interface blocks
- Utilities library
- MATLAB script file for model
- Automatically run MATLAB script
- Running Simulink simulation
- DSP Builder menu
- Quartus Platform Designer tools integration

### ❖ **Advanced Blockset Library Overview**

- Library components
- Additional libraries (Avalon streaming, beta utilities, control, vector utils)
- Base blocks
- Base library control block
- Base library signals block
- Base library device block



When innovation meets expertise...

- FFT blockset
- Filters
- FIR filter
- ModelBus library
- ModelPrim library
- SynthesisInfo block
- Flow control
- Floating point support
- New floating point blocks
- Selection of IEEE format precisions
- Size vs precision trade-off
- Elementary mathematical functions
- Fundamental blocks
- Function improvement
- Extended function coverage: Trig, Root
- Waveform synthesis
- NCO specification
- What is frequency hopping in NCO?
- Continuous phase transition
- Updating frequency matrix

#### ❖ **Design Flow Overview**

- Typical design flow
- Creating MATLAB and Simulink project
- Create model
- New model wizard
- Top level testbench
- Navigating the model
- Synthesizable model
- Device selection
- Adding ModelIP blocks
- Creating ModelPrim subsystem
- SynthesisInfo block
- Processor interface
- MATLAB script/parameters file



When innovation meets expertise...

- Automatically run MATLAB script
- Running Simulink simulation
- Documentation generation
- Design verification
- Tools interoperability

#### ❖ **Connectivity with MultiChannel/MultiRate Systems**

- Protocol for connecting blocks
- Data path vectorization
- ModelPrim vectorization
- Vector initialization
- ModelIP vectorization
- Current design methodology
- Advanced blockset methodology
- TDM
- Parameter definitions
- Parameter example with TDM
- Parameter example with vectorization
- Multi-channel operation – ModelIP
- Connecting ModelIP blocks
- Connecting filters to complex mixer
- Connecting Real mixer to filter
- Connecting ModelIP to ModelPrim
- Multi-channel in ModelPrim subsystem
- Primitive folding
- Primitive blocks folding example (FIR)
- Primitive folding design consideration
- Primitive folding implementation consideration

#### ❖ **Latency Management**

- Cycle accuracy and latency
- Displaying latency for ModelIP blocks
- Latency for ModelPrim subsystem
- Distributed delays
- Latency constraint



When innovation meets expertise...

- Constraining ModelIP latency
- Latency parameter
- Latency and fmax constraint conflict
  
- **Lab #1: Algorithm Implementation (DSB Demodulator)**

## Day #2

### ❖ **Hardware and Memory-Mapped Interface**

- ModelIP interfaces
- Typical FIR filter interface
- Typical memory-mapped interface

### ❖ **Hardware and Documentation Generation**

- Model simulation
- Generated directory structure and files
- Generated documentation

### ❖ **Hardware Functional Verification**

- Comparison with RTL simulation
- Interactive ModelSim session
- Automatic testbench flow
- Invoke automatic testbench flow
- Simulink and RTL mismatches

### ❖ **Hardware – Software Integration**

- System Integration
- Qsys integration
- Qsys component interfaces
- Assigning base addresses
- Nios II runtime control
- System.h file for Nios II BSP



- Header files for Nios II peripherals
- Compilation report
- Resource/timing verification reports
  
- **Lab #2: System Integration (DSP Builder and Platform Designer)**

#### ❖ **Design Optimization: Memory and Multiplier Trade-Offs**

- Memory and multiplier thresholds
- Hard multiplier threshold
- Memory threshold
- Scripting
- Scripting changing parameters
- Scripting run hardware compile

#### ❖ **TDM Hardware Reuse**

- TDM revisited
- TDM design consideration
- TDM support: ModelIP multichannel support
- TDM design: ModelIP trade-off
  
- **Lab #3: Design Exploration (Multiplier Trade-Off)**

#### ❖ **Multi-Channel TDM Example Using FIR ModelIP**

- Example FIR design
- Set the system parameters
- Set filter coefficients
- Compile FPGA
- Increase fmax and implement TDM reuse
- Recompile FPGA
- Design iterations summary

#### ❖ **Putting Everything Together: WiMAX Digital Up Converter (DUC)**

- 2 antenna WiMAX reference design
- Step 1: design datapath



When innovation meets expertise...



**OREN HANDS ON TRAINING & DEVELOPMENT LTD.**  
20 YAIR ROZENBLUM ST. KFAR SABA 4464601 ISRAEL

---

- Step 2: determine channel/wire structure
- Step 3: determine scaling factors
- Step 4: creating synchronizer
- Changing the design with DSP Builder
- Changing the design without DSP Builder
  
- **Lab #4: Design Exploration (TDM Hardware Reuse)**



When innovation meets expertise...