

Programming Xilinx Zynq SoCs with MATLAB and Simulink

SciEngineer's training courses are designed to help organizations and individuals close skills gaps, keep up-to-date with the industry-accepted best practices and achieve the greatest value from MathWorks® and COMSQL® Products.



Programming Xilinx Zynq SoCs with MATLAB and Simulink

This two-day course focuses on developing and configuring models in Simulink and deploying on Xilinx Zynq-7000 All Programmable SoCs. This course shows how to generate, validate, and deploy embedded code and HDL code for software/hardware codesign using Embedded Coder and HDL Coder.

Prerequisites

Simulink Fundamentals (or Simulink Fundamentals for Automotive Applications or Simulink Fundamentals for Aerospace Applications). Knowledge of C and HDL programming languages.

TOPICS

Day 1

- Zynq Platform Overview and Environment Setup
- Introduction to Embedded Coder and HDL Coder
- IP Core Generation and Deployment
- Using AXI4 Interface
- Processor-in-the-Loop Verification

Day 2

- Data Interface with Real-Time Application
- Integrating Device Drivers
- Custom Reference Design



Zynq Platform Overview and Environment Setup

OBJECTIVE: Configure Zynq-7000 platform and MATLAB environment.

Introduction to **Embedded Coder** and HDL Coder

OBJECTIVE: Configure Simulink models for embedded code generation and effectively interpret the generated code.

- Zynq-7000 overview
- Setting up Zynq platform and software
- Configuring MATLAB environment
- Testing connectivity to Zyng hardware

- Architecture of an embedded application
- Generating ERT code
- Code modules
- Data structures in generated code
- Configuring a Simulink model for HDL code generation
- Using HDL Workflow Advisor

IP Core Generation and Deployment

OBJECTIVE: Use HDL Workflow Advisor to configure a Simulink model, generate and build both HDL and C code, and deploy to Zyng platform.

- Configuring a subsystem for programmable logic
- Configuring the target interface and peripherals
- Generating the IP core and integrating with SDK
- Building and deploying the FPGA bitstream
- Generating and deploying a software interface model
- Tuning parameters with External Mode

Using AXI4 Interface

OBJECTIVE: Use various AXI interfaces for data communication between processing system and programmable logic.

- AXI interface overview
- AXI4-Lite applications
- Using AXI4-Stream
- AXI4 performance considerations

Processor-in-the-Loop Verification

OBJECTIVE: Use processor-in-the-loop to verify the algorithm running on Zynq platform and profile the execution times in your production algorithm.

- Processor-in-the-loop (PIL) workflow on Zynq
- PIL verification with model reference
- Code execution profiling with PIL
- PIL considerations



Integrating **Device Drivers**

Custom **Reference Design**

OBJECTIVE: Develop device driver interfaces for integrating peripherals on processing system.

OBJECTIVE: Create and package reusable IP for Vivado and register custom boards and reference designs.

- Workflow for developing device drivers
- Using the Legacy Code Tool
- GPIO interface
- Cross-compiling device drivers

 Motivations for a custom reference design

- Creating reusable IP for Vivado
- Reference design overview
- Customizing a reference design
- Registering board and custom reference design

Data Interface with Real-Time Application

OBJECTIVE: Use the UDP interface to stream data between Simulink and the real-time application running on Zynq platform.

• Data interface overview

- Configuring UDP blocks for data streaming
- Synchronizing data between Simulink and Zyng

• Data interface with AXI Stream

- Design partitioning
- Data interface considerations



Expand your knowledge

