sciengineer

Simulink Real-Time HDL Workflow with Speedgoat Hardware

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 COMSOL® Products.

Simulink Real-Time HDL Workflow with Speedgoat Hardware

This two-day course focuses on partitioning Simulink models intended for real-time execution on Speedgoat target machines to execute on the CPU and FPGA. A distinction is made here between the algorithm itself and any I/O functionality which may be necessary to communicate with the outside world. Both rapid Control prototyping (RCP) and hardware-in-the-loop Testing (HIL) are discussed. The course consists of various modules which can be combined according to customer needs.

Prerequisites

Simulink Fundamentals (or Simulink Fundamentals for Aerospace Applications or Simulink Fundamentals for Automotive Applications). Knowledge of Simscape™ preferred.

DURATION	LEVEL
2 Days	Advanced
● ● ●	▬ ▬ ▬

TOPICS

Day 1

- Overview on Workflows
- Setting Up Development and Target Computers
- From Desktop to Real-Time Simulation
- Basic HDL Workflow
- Fixed-Point Conversion

Day 2

- Integrating External Code – Black Boxing
- Speedgoat HDL Coder™ I/O Blocksets
- Implementing Algorithms Together with External HDL Code
- Simscape™ Hardware-in-the-Loop Workflow
- Appendix A: Plant Modeled with Simulink® on FPGA
- Appendix B: HIL I/O Functionality on Configurable FPGAs

TRAINING CONTENT - DAY 1

Overview on Workflows

OBJECTIVE: Understand the concepts of RCP versus HIL. Know the deployment options: CPU versus FPGA.

- Real-time testing workflows
- Levels of model accuracy
- Deployment options on CPUs and FPGAs

Setting Up Development and Target Computers

OBJECTIVE: Be able to set up the communication between target PC and development computer. Be able to run ready-made applications on real-time target machine.

- Setting up development computer and target PC
- Starting and stopping the application
- Viewing signals
- Changing parameters at run time

From Desktop to Real-Time Simulation

OBJECTIVE: Become familiar with the example used during the course. Understand different levels of modeling accuracy. Be able to transform a desktop simulation model to a deployable real-time model.

- Course example: servo motor control
- Different levels of model accuracy
- Simulation with average values
- Simulation with PWM
- From desktop to real-time simulation

Simulink Real-Time HDL Workflow with Speedgoat Hardware

TRAINING CONTENT - DAY 1

Basic HDL Workflow

OBJECTIVE: Become familiar with the basics of HDL Workflow Advisor for programming an FPGA within the Speedgoat target machine. Be able to deploy a very simple model which uses just digital I/O and does not need any special optimization for deployment.

- HDL workflow overview
- Preparing models for HDL code generation
- HDL Workflow Advisor
- Oversampling

Fixed-Point Conversion

OBJECTIVE: Be able to convert a Simulink model using floating point data types to a model using fixed-point data types.

- Converting from floating to fixed-point
- Using internal rules
- Fixed-point scaling and inheritance
- Using the Fixed-Point Tool

SciEngineer's Training Services

TRAINING CONTENT - DAY 2

Integrating External Code – Black Boxing

OBJECTIVE: Be able to configure a Simulink model to make use of functionality that is already available as HDL code.

- Existing external HDL code
- Configuration of model for code generation
- Subsystem for including the external code
- Subsystem for analog input
- Generation of interface model
- Deployment and running of the application

Speedgoat HDL Coder™ I/O Blocksets

OBJECTIVE: Be able to configure a Simulink model to make use of I/O functionality provided by Speedgoat HDL I/O blocksets.

- FPGA library blocks - PWM
- CPU library blocks - PWM
- Including the library blocks into course example model
- Finalizing the model

Implementing Algorithms Together with External HDL Code

OBJECTIVE: Be able to combine application algorithm and I/O functionality on a FPGA. Be able to understand and fix timing issues which may occur when generating HDL code from Simulink models.

- Combining I/O functionality and controller algorithm for FPGA deployment
- Understanding timing on a FPGA
- Using the generic ASIC/FPGA workflow within the HDL Workflow Advisor (HDLWA)
- HDLWA – Timing optimization using clock-rate pipelining
- HDLWA – Timing optimization using enable-based constraints

TRAINING CONTENT - DAY 2

Simscape™ Hardware-in-the-Loop Workflow

OBJECTIVE: Be able to convert Simscape-based models into models only using Simulink blocks which may be deployed to an FPGA.

- Simscape HIL workflow overview
- Using the Simscape HDL Workflow advisor for converting a Simscape model into a Simulink implementation model
- Validating the implementation model
- Preparing the implementation model for HDL code generation
- Generating HDL code
- Running the HIL application

Appendix A: Plant Modeled with Simulink® on FPGA

OBJECTIVE: Be able to prepare a Simulink model with continuous blocks for HDL code generation. Analyze and solve timing issues and handle limited amount of area on the FPGA.

- Continuous simulation
- Discretizing the model
- Model partitioning
- Prepare for HDL code generation
- Optimize for HDL code generation
- Generic HDL workflow
- Code generation report
- Area and timing optimizations
- Validation mode
- Simulink Real-Time FPGA I/O workflow

Appendix B: HIL I/O Functionality on Configurable FPGAs

OBJECTIVE: Be able to use Speedgoat ready-made bitstream files for programming I/O functionality on configurable FPGA boards for HIL testing (with plant algorithm on CPU).

- Splitting Simulation Model into Plant and Controller Part
- Adding Drivers
- Model for TI Microcontroller

sciengineer

Expand your knowledge