Vitis Model Composer: A MATLAB and Simulink-based Product

DSP-MCSIM (v1.0)

Course Specification

Course Description

This course provides experience with using the Vitis™ Model Composer tool for model-based designs.

The course provides experience with:

- Creating a model-based design using HDL, HLS, and AI Engine library blocks along with custom blocks in Vitis Model Composer
- Implementing DSP functions using Vitis Model Composer
- Utilizing design implementation tools
- Transforming algorithmic specifications to production-quality IP implementations using automatic optimizations and leveraging the high-level synthesis technology of the Vitis Unified IDE
- Creating Versal™ AI Engine graphs and kernels using Vitis Model Composer
- Connecting AI Engine blocks and non-AI Engine blocks
- Verifying and debugging AI Engine code using the Analysis view of the Vitis Unified IDE
- Simulating and debugging a complex system created using AI Engine library blocks
- Performing hardware validation using Vitis Model Composer
- Integrating an AI Engine design from Vitis Model Composer into the Vitis Unified IDE

What's New for 2024.1

- Updated all the labs and modules with latest ease of use Vitis Model Composer Hub block improvements
- All labs have been updated to the latest software versions

Level – DSP 3

Course Details

- 2 days live instructor-led training (in person or online)
  - 13 lectures
  - 13 labs

Price – $1,600 or 16 AMD Training Credits

Course Part Number – DSP-MCSIM

Who Should Attend? – System engineers, system designers, logic designers, and experienced hardware engineers who are implementing Versal AI Engine, HDL, and HLS algorithms using the MathWorks MATLAB® and Simulink® software and want to use Vitis Model Composer

Prerequisites

- Basic experience with the MATLAB and Simulink software
- Basic understanding of DSP designs and sampling theory
- Comfort with the C/C++ programming language for HLS or AI Engine model designs

Software Tools

- Vivado Design Suite 2024.1
- Vitis Unified IDE 2024.1
- MATLAB with Simulink software R2023b

Hardware

- Architecture: Zynq™ UltraScale+™ MPSoC and Versal™ AI Core series
- Demo board: Zynq™ UltraScale+ MPSoC ZCU104 board*

* This course focuses on the Zynq UltraScale+ and Versal architectures. Check with Morgan Advanced Programmable Systems, Inc. for the specifics of the in-class lab board or other customizations.

After completing this comprehensive training, you will have the necessary skills to:

- Use optimized HDL, HLS, and AI Engine blocks directly from the Simulink tool library browser
- Create, simulate, and debug a Vitis Model Composer design in the Simulink environment using HDL, HLS, and AI Engine block libraries
- Perform co-simulation and hardware verification
- Use DSP blocks in Vitis Model Composer to implement DSP functions
- Implement multi-rate systems in Vitis Model Composer
- Design a processor-controllable interface using Vitis Model Composer
- Generate IPs from C-based design sources using Vitis HLS for use in the Vitis Model Composer environment
- Import custom HDL, HLS, and AI Engines code as blocks into Vitis Model Composer
- Generate output products using automatic code generation
- Connect AI Engine blocks and non-AI Engine blocks
- Perform AI Engine code verification using the Analysis view of the Vitis Unified IDE
- Create, simulate, and debug a complex system created using AI Engine library blocks
- Validate an AI Engine design using hardware emulation
- Integrate an AI Engine design from Vitis Model Composer into the Vitis Unified IDE

Course Outline

Day 1

Introduction to Vitis Model Composer

- **Introduction to Vitis Model Composer**
  Introduces the Vitis Model Composer tool and describe the optimized HDL, HLS, and AI Engine library blocks available in Vitis Model Composer. {Lecture}
- **Basics of the Simulink Environment**
  Describes the Simulink software environment, some of the commonly used signal source and sink blocks available in the Simulink software, and how hierarchical designs are created and protected using masked subsystems. {Lecture, Lab}

Vitis Model Composer for HDL

- **HDL Library in Vitis Model Composer**
  Illustrates how the HDL library can be used in Vitis Model Composer and how to analyze performance and resource usage in Vitis Model Composer. {Lecture, Labs}
- **HDL Library Compilation and Hardware Co-Simulation**
  Covers how to import HDL modules as well as perform HDL co-simulation and hardware verification. Reviews the compilation types for Vitis Model Composer designs. Also introduces Super Sample Rate (SSR) blocks in Vitis Model Composer. {Lecture}
- **DSP Blocks in Vitis Model Composer**
  Describes the DSP blocks in the HDL and AI Engine library. Also reviews the basics of AXI4 interfaces. {Lecture, Lab}
- **Working with Filter Designs**
  Describes the concept of designing filters supported by Vitis Model Composer. {Lecture, Lab}

- **Working with Multi-Rate Systems**
  Explains how a multi-rate DSP system uses multiple sampling rates within a system. {Lecture}

Day 2

Vitis Model Composer for HLS

- **HLS Library in Vitis Model Composer**
  Describes how to create Vitis Model Composer designs using HLS block libraries, import C/C++ code into Vitis Model Composer, and generate output products using automatic code generation. {Lecture, Labs}

Vitis Model Composer for AI Engines

- **AI Engine Library in Vitis Model Composer**
  Demonstrates the usage the AI Engine library in Vitis Model composer for creating an AI Engine design, which involves preparing the kernel and importing the AI Engine code as a block. {Lecture}
- **AI Engine Simulation and Code Generation**
  Illustrates the process of generating AI Engine code with a data flow graph, which involves Simulink simulation with the AI Engine library for functional verification. Also describe the hardware validation flow through generating a hardware image targeting a specific platform for the Simulink environment. {Lecture, Labs}
- **Connecting AI Engine and Non-AI Engine Blocks**
  Explains how to interconnect AI Engine blocks and non-AI Engine (HDL and HLS) blocks. {Lecture}
- **Analyzing and Debugging an AI Engine Design in Vitis Model Composer**
  Shows how to use the Analysis view of the Vitis Unified IDE for viewing and analyzing various parameters that are useful for debugging Versal AI Engines. {Lecture, Lab}
- **Exporting an AI Engine Design to the Vitis Unified IDE**
  Demonstrates how to export an AI Engine design into the Vitis Unified IDE after it has been simulated and validated in Vitis Model Composer. {Lab}

GitHub Examples

- **Exploring Vitis Model Composer Examples in GitHub**
  Introduces different categories of Vitis Model Composer examples in GitHub and describes the methods to access these examples from GitHub. {Lecture}

Register Today

Morgan Advanced Programmable Systems, Inc. (Morgan A.P.S.) delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, Nebraska, North Dakota, South Dakota, and Wisconsin.

Visit **morgan-aps.com/training**, for full course schedule and training information.

Morgan Advanced Programmable Systems

- You must have your tuition payment information available when you enroll. We accept credit cards (Visa, MasterCard, or American Express) as well as purchase orders and Xilinx training credits.

Student Cancellation Policy

- Student cancellations received more than 7 days before the first day of class are entitled to a 100% refund. Refunds will be processed within 14 days.
- Student cancellations received less than 7 days before the first day of class are entitled to a 100% credit toward a future class.
- Student cancellations must be sent here.

Morgan A.P.S. Course Cancellation Policy

- We regret from time-to-time classes will need to be rescheduled or cancelled.
- In the event of cancellation, live on-line training may be offered as a substitute.
- Morgan A.P.S. may cancel a class up to 7 days before the scheduled start date of the class; all students will be entitled to a 100% refund.
- Under no circumstances is Morgan A.P.S. responsible or liable for travel, lodging or other incidental costs. Please be aware of this cancellation policy when making your arrangements.
- For additional information or to schedule a private class contact us here.

Online or in person training with real hardware

- Morgan Advanced Programmable Systems, Inc. has set up a training VPN where engineer participants can take classes online using the same computers and devCards used during in-person training.
- Even better, and upon request, you can use these computers after hours on training days to experiment with labs. This is not possible for in-person training.
- Additionally, just like in-person training, the laptops and devCards, tools, OS, and licensing are set up in advance.
- In some ways, live online-training is better than in-person…for example, you can grant the instructor permission to look at your Vivado, PetaLinux terminal, or Vitis for extended periods of time if your lab is not going exactly has planned to a missed step.
- This is often more comfortable than two engineers crowding around a laptop screen.
- Taking remote training also allows you to learn some tips and tricks for working remote. Whether your devCard is in the lab down the hall, or across the world via VPN, you can control your Xilinx based device quickly and efficiently.