Designing with Dynamic Function eXchange (DFX) Using the Vivado Design Suite

FPGA-DFX (v1.0)

Course Specification

Course Description

Learn how to construct, implement, and download a Dynamic Function eXchange (DFX) design using the Vivado™ Design Suite. This course covers both the tool flow and mechanics of successfully creating a DFX design.

The emphasis of this course is on:

- Identifying best design practices and understanding the subtleties of the DFX design flow
- Using DFX in AMD FPGAs and adaptive SoCs
- Implementing DFX designs using features such as block design containers (BDC), Abstract Shells, and nested DFX
- Recognizing DFX design considerations for FPGAs and adaptive SoCs
- Using the DFX IPs in the DFX process
- Analyzing and floorplanning DFX designs
- Configuring devices using DFX
- Applying appropriate debugging techniques on DFX designs
- Implementing DFX in an embedded system environment

What's New for 2025.2

- Updated DFX Design Considerations for All AMD Devices module with information on:
  - Embedded I/O usage guidelines
  - Initializing reconfigurable modules after DFX
- All labs have been updated to the latest software versions

Level – FPGA 4

Course Details

- 2 days ILT
- 22 lectures
- 11 labs
- 2 demos (ILT only)

Course Part Number – FPGA-DFX

Who Should Attend? – Digital designers who have a working knowledge of HDL (VHDL or Verilog) and digital design and who want to implement Dynamic Function eXchange techniques

Prerequisites

- Knowledge of VHDL or Verilog
- Experience with the Vivado Design Suite
- Moderate familiarity with digital design techniques
- Experience with Tcl
- Moderate familiarity with the project mode and non-project batch mode flow in the Vivado Design Suite
- *Designing with the Versal Adaptive SoC: Quick Start*

Software Tools

- Vivado Design Suite 2025.2
- Vitis™ Unified IDE 2025.2

Hardware

- Architecture: UltraScale™ FPGAs and Versal adaptive SoCs*
- Demo board:
  - Zynq™ UltraScale+™ MPSoC ZCU104 board*
  - Versal adaptive SoC VCK190 board*

* Check with your local Authorized Training Provider for the specifics of the in-class lab board or other customizations.

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

- Describe what Dynamic Function eXchange is
- Define DFX regions and reconfigurable modules with the Vivado Design Suite
- Generate the appropriate full and partial bitstreams for a DFX design
- Use DFX feature for the Versal devices
- Use the block design container feature of the Vivado IP integrator to create a DFX design
- Enable the Abstract Shell feature in project mode
- Use the RTL modular NoC flow for Versal DFX designs
- Implement a nested DFX design
- Use the ICAP and PCAP components to deliver partially reconfigurable systems
- Implement a DFX system using the DFX Controller IP
- Identify how Dynamic Function eXchange affects various silicon resources, including block RAM, IOBs, fabric, and MGTs
- Implement a Dynamic Function eXchange system using the following techniques:
  - Direct JTAG connection, floorplanning, and timing constraints and analysis
- Debug a DFX design using the Vivado Design Suite
- Implement a DFX system in an embedded environment using the Vitis Unified IDE

Course Outline

Day 1

Basics of DFX

- **Introduction to Dynamic Function eXchange (DFX)**
  Explains what a Dynamic Function eXchange is and defines the terminologies used in DFX. Also provides an overview of the configuration and reconfiguration processes. {Lecture, Demo}

DFX Tool Flow

- **DFX Flow Using the Vivado Design Suite GUI**
  Illustrates the steps for creating a DFX project in the Vivado Design Suite and describes various supported and unsupported features. {Lecture, Lab}
- **DFX Flow Using Vivado Design Suite Tcl Commands**
  Reviews the flow using non-project-based commands, including using implementation constraints and specific characteristics. {Lecture, Lab}
- **DFX for the Versal Architecture**
  Describes the DFX feature for the Versal devices, explains DFX for the network on chip (NoC), and illustrates the NoC topologies supported in the DFX design flow. {Lecture}
- **Block Design Containers in Vivado IP Integrator**
  Describes the block design container feature and how BDCs enable DFX. {Lecture, Lab}
- **Abstract Shell for Dynamic Function eXchange**
  Describes how compilation time can be reduced by using an Abstract Shell. {Lecture, Lab}

© Copyright 2026 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, UltraScale, UltraScale+, Versal, Vitis, Vivado, Zynq, and combinations thereof are trademarks of Advanced Micro Devices, Inc. PCIe is a registered trademark of PCI-SIG Corporation. Other product names used in this publication are for identification purposes only and may be trademarks of their respective owners.

- **RTL Modular NoC Flow for Versal DFX Designs**
  Describes the RTL modular NoC flow for Versal DFX designs. Also introduces the virtual NoC interface concept for DFX designs. {Lecture}
- **Nested DFX**
  Describes using nested DFX, the process by which a Reconfigurable Partition (RP) can be segmented into smaller regions, each of which is partially reconfigurable. {Lecture, Lab}

DFX Design Considerations for AMD Devices

- **DFX Design Considerations for All AMD Devices**
  Covers the requirements, characteristics, and limitations associated with the DFX designs that can simplify the debug process and reduce the risk of design malfunctions. {Lecture}
- **DFX Design Considerations for 7 Series, Zynq SoC, UltraScale, and UltraScale+ Devices**
  Discusses the DFX design consideration methodologies for various AMD device families. {Lecture}
- **DFX Design Considerations for Versal Devices**
  Describes the DFX design requirements that are specific to the Versal devices. {Lecture}

Day 2

DFX Design-Specific IP Blocks

- **DFX Intellectual Property (IP)**
  Reviews the various IPs that are specifically for use with the DFX designs. {Lecture, Lab, Demo}

DFX Design Analysis

- **Floorplanning a DFX Design**
  Demonstrates how to create Pblocks for various devices and how to create a floorplan for a reconfigurable region. {Lecture, Lab}
- **Floorplanning for Versal Devices**
  Illustrates floorplanning methodologies for Versal devices and explains challenges in the Versal floorplanning. {Lecture, Lab}
- **DFX Timing Analysis and Constraints**
  Illustrates how and when to apply different constraint files, the process of performing a DFX timing-level simulation, and the process of performing static timing analysis on a DFX design. {Lecture, Lab}

DFX Configuration and Debugging

- **Configuring Devices Using DFX**
  Reviews the basics of configuration and various configuration modes. {Lecture}
- **Configuration Parameters**
  Covers various configuration parameters, including factors that affect configuration time and configuration debugging. {Lecture}
- **DFX Bitstreams and PDIs**
  Describes the different types of bitstreams for the DFX compilation, including full, partial, blanking, and clearing. Also explains partial programmable device image (PDI) for the Versal™ devices. {Lecture}
- **DFX Bitstream Integrity**
  Describes partial bit file integrity and implementing a DFX through the ICAP for the FPGA devices. {Lecture}
- **DFX Debugging**
  Illustrates the DFX debugging techniques using the Vivado Design Suite debug cores. {Lecture, Lab}

DFX Designs in Embedded Systems

- **DFX in Embedded Systems**
  Describes the embedded design flow in the Vivado Design Suite, the advantages of using a processor with a DFX, and how to connect a processor to the PCAP to control a DFX using the Vitis Unified IDE. {Lecture, Lab}
- **DFX Designs Using the PCIe Core**
  Reviews the advantages of using a PCIe core in a DFX design. {Lecture}

Register Today

Visit the AMD Adaptive Computing Customer Training Portal to view schedules and register online.

© Copyright 2026 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, UltraScale, UltraScale+, Versal, Vitis, Vivado, Zynq, and combinations thereof are trademarks of Advanced Micro Devices, Inc. PCIe is a registered trademark of PCI-SIG Corporation. Other product names used in this publication are for identification purposes only and may be trademarks of their respective owners.