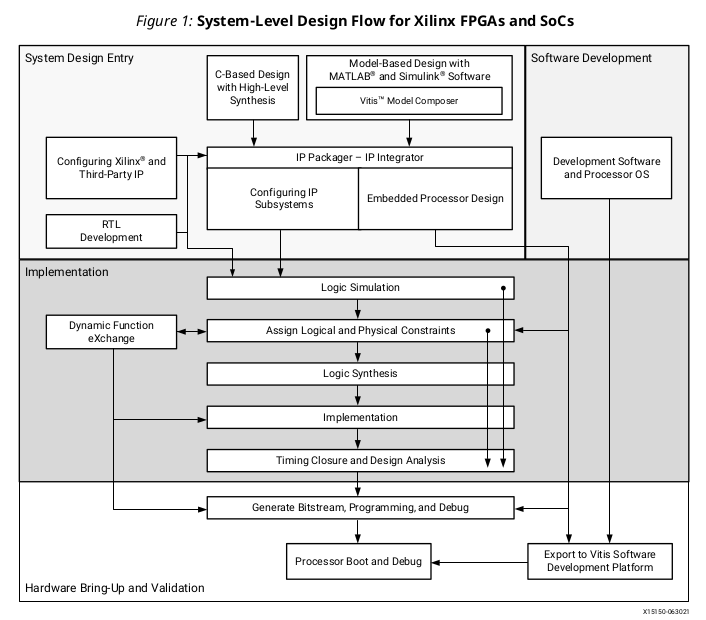
Vivado Design Flows Overview

**1. Vivado System-Level Design Flows**



（1）System Design Entry

（2）Implementation

（3）Hardware Bring-up and Validation

（4）Software Development

a. RTL-to-Bitstream Design Flow

* RTL Design

Vivado synthesis and implementation support multiple source file types, including Verilog, VHDL, SystemVerilog, and XDC.

* IP Design and System-Level Design Integration

The Vivado Design Suite provides an environment to configure, implement, verify, and integrate

IP as a standalone module or within the context of the system-level design. IP can include logic,

embedded processors, digital signal processing (DSP) modules, or C-based DSP algorithm

designs. Custom IP is packaged following IP-XACT protocol and then made available through the

Vivado IP catalog. Xilinx IP utilizes the AXI4 interconnect standard to enable faster system-level

integration.

* IP Subsystem Design

The Vivado IP integrator environment enables you to stitch together various IP into IP

subsystems using the AMBA ® AXI4 interconnect protocol. You can interactively configure and

connect IP using a block design style interface and easily connect entire interfaces by drawing

DRC-correct connections similar to a schematic. Connecting the IP using standard interfaces

saves time over traditional RTL-based connectivity. For more information, see the Vivado

Design Suite User Guide: Designing IP Subsystems Using IP Integrator (UG994).

* I/O and Clock Planning
* Xilinx Platform Board Support
* Synthesis
* Design Analysis and Simulation
* Placement and Routing
* Hardware Debug and Validation

Debug signals can be identified in the RTL design, or inserted after synthesis and are processed throughout the flow. You can add debug cores to the RTL source files, to the synthesized netlist, or in an implemented design using the using the Engineering Change Order (ECO) flow.

b. Alternate RTL-to-Bitstream Design Flows

* Accelerated Kernel Flows

The Xilinx ® Vitis™ unified software platform introduces acceleration use cases into Vivado ®

flows. In this design methodology, Vivado is used to create a platform which is consumed by the

Vitis software platform to add accelerated kernels.For more information on platform creation, see Vitis Unified Software Platform Documentation: Application Acceleration Development

(UG1393).

* Embedded Processor Design

Data hand-off between the hardware and software flows, and validation across these two domains is critical for success. The embedded processor design flow is described in the following resources:

Vivado Design Suite User Guide: Embedded Processor Hardware Design (UG898)

Vivado Design Suite Tutorial: Embedded Processor Hardware Design (UG940)

UltraFast Embedded Design Methodology Guide (UG1046)

* Model-Based Design Using Model Composer

Model Composer is a model-based graphical design tool that enables rapid design exploration

within the MathWorks MATLAB ® and Simulink ® products and accelerates the path to production

for Xilinx devices through automatic code generation. For information, see the Model Composer

User Guide (UG1262).

* Model-Based DSP Design Using Xilinx System Generator

You create the DSP functions using System Generator as a standalone tool, and then package your System Generator design into an IP module that can be included in the Vivado IP catalog. From there, the generated IP can be instantiated into your Vivado design as a submodule. For more information, see the Vivado Design Suite User Guide: Model-Based DSP Design Using System Generator (UG897).

* High-Level Synthesis C-Based Design

The C-based High-Level Synthesis (HLS) tools within the Vivado Design Suite enable you to

describe various DSP functions in the design using C, C++, and SystemC. HLS lets you simulate the generated RTL directly from its design environment using C-based test benches and simulation. C-to-RTL synthesis transforms the C-based design into an RTL module that can be packaged and implemented as part of a larger RTL design, or instantiated into an IP integrator block design.

The HLS tool flow and features are described in the following resources:

Vivado Design Suite User Guide: High-Level Synthesis (UG902)

Vivado Design Suite Tutorial: High-Level Synthesis (UG871)

* Dynamic Function Exchange Design

Dynamic function exchange (DFx) allows portions of a running Xilinx device to be reconfigured in

real-time with a partial bitstream, changing the features and functions of the running design.

The DFx tool flow and features are described in the following resources:

Vivado Design Suite User Guide: Dynamic Function eXchange (UG909)

Vivado Design Suite Tutorial: Dynamic Function eXchange (UG947)

* Hierarchical Design

Hierarchical Design (HD) flows enable you to partition a design into smaller, more manageable

modules to be processed independently. For more information, see the Vivado Design Suite User Guide: Hierarchical Design (UG905).

**2. Vivado Design Suite Use Models**

（1）Working with the Vivado Integrated Design Environment (IDE)

（2）Working with Tcl

（3）Understanding Project Mode and Non-Project Mode

（4）Using Third-Party Design Software Tools

（5）Interfacing with PCB Designers

c. Working in Project Mode and Non-Project Mode

Some users prefer the design tool for automatically managing their design flow process and design data, while others prefer to manage sources and process themselves. The Vivado Design Suite uses a project file (.xpr) and directory structure to manage the design source files, store the results of different synthesis and implementation runs, and track the project status through the design flow. This automated management of the design data, process, and status requires a project infrastructure. For this reason, Xilinx refers to this flow as the Project Mode.

Other users prefer to run the FPGA design process more like a source file compilation, to simply

compile the sources, implement the design, and report the results. This compilation style flow is

referred to as the Non-Project mode.

The main distinctions are that Non-Project mode processes the entire design in memory. No files are

written to disk. While Project mode creates and maintains a project directory structure on disk to manage design sources, results, and project settings and status.