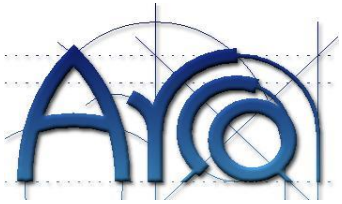
ICTP-IAEA SCHOOL ON HYBRID RECONFIGURABLE DEVICES FOR SCIENTIFIC INSTRUMENTATION

Vivado Design Methodology

Grupo de Arquitectura y Redes de Computadores



Universidad de Castilla-La Mancha

UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

Julio Daniel Dondo
Juliodaniel.dondo@uclm.es

Hw design gap

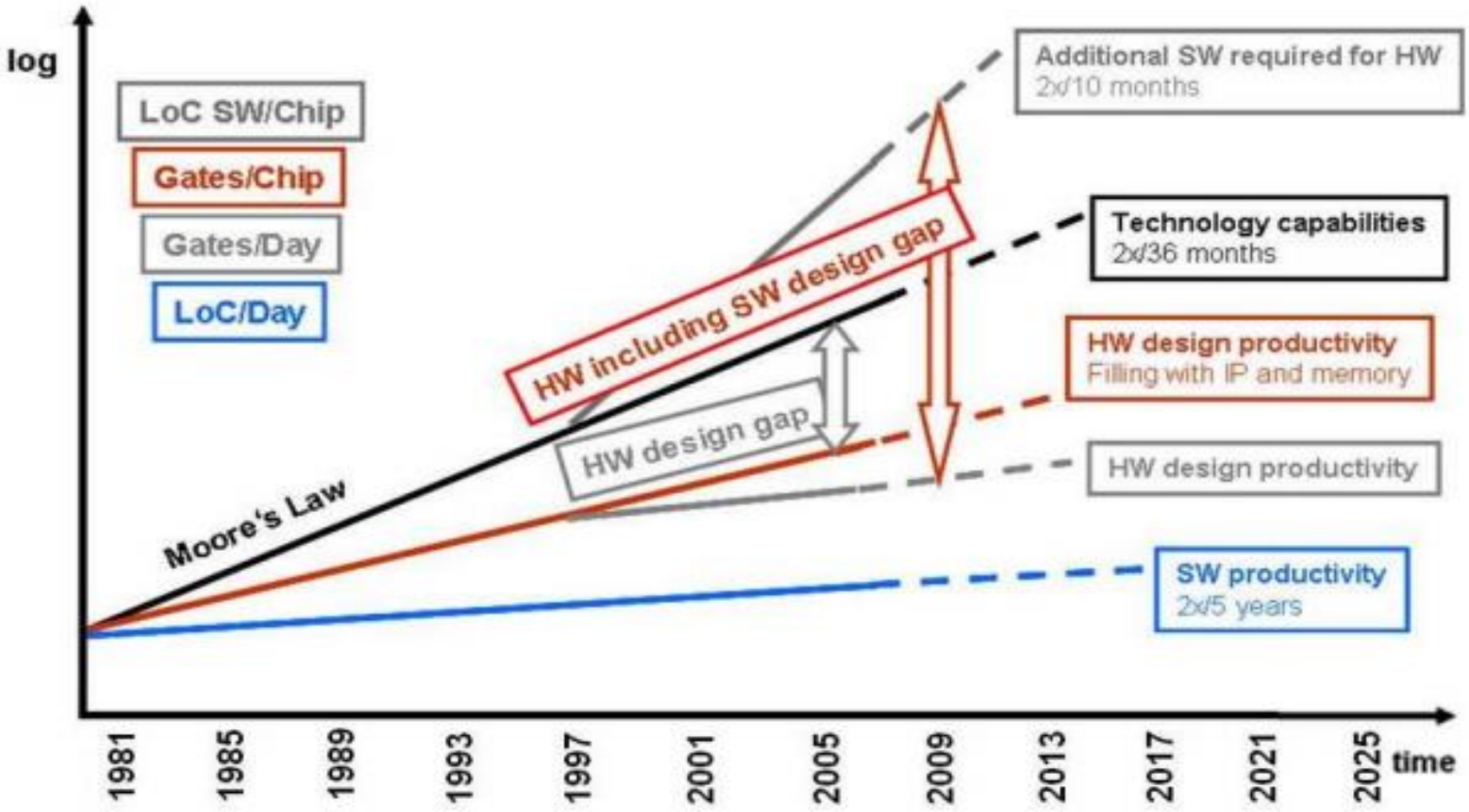
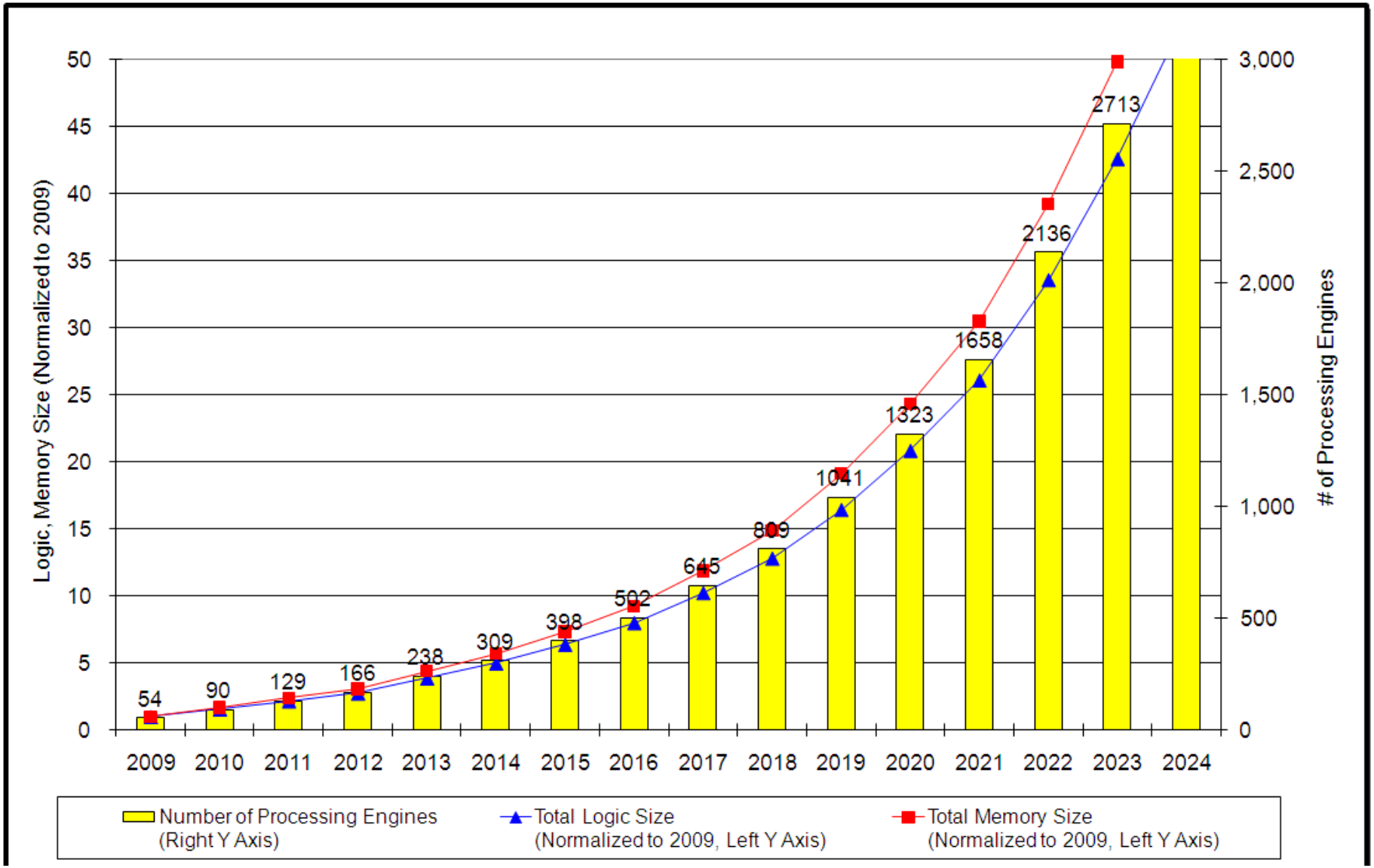


Figure DESN3 Hardware and Software Design Gaps versus Time[5]

THE INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS: *2009*

SoC Design complexity trends



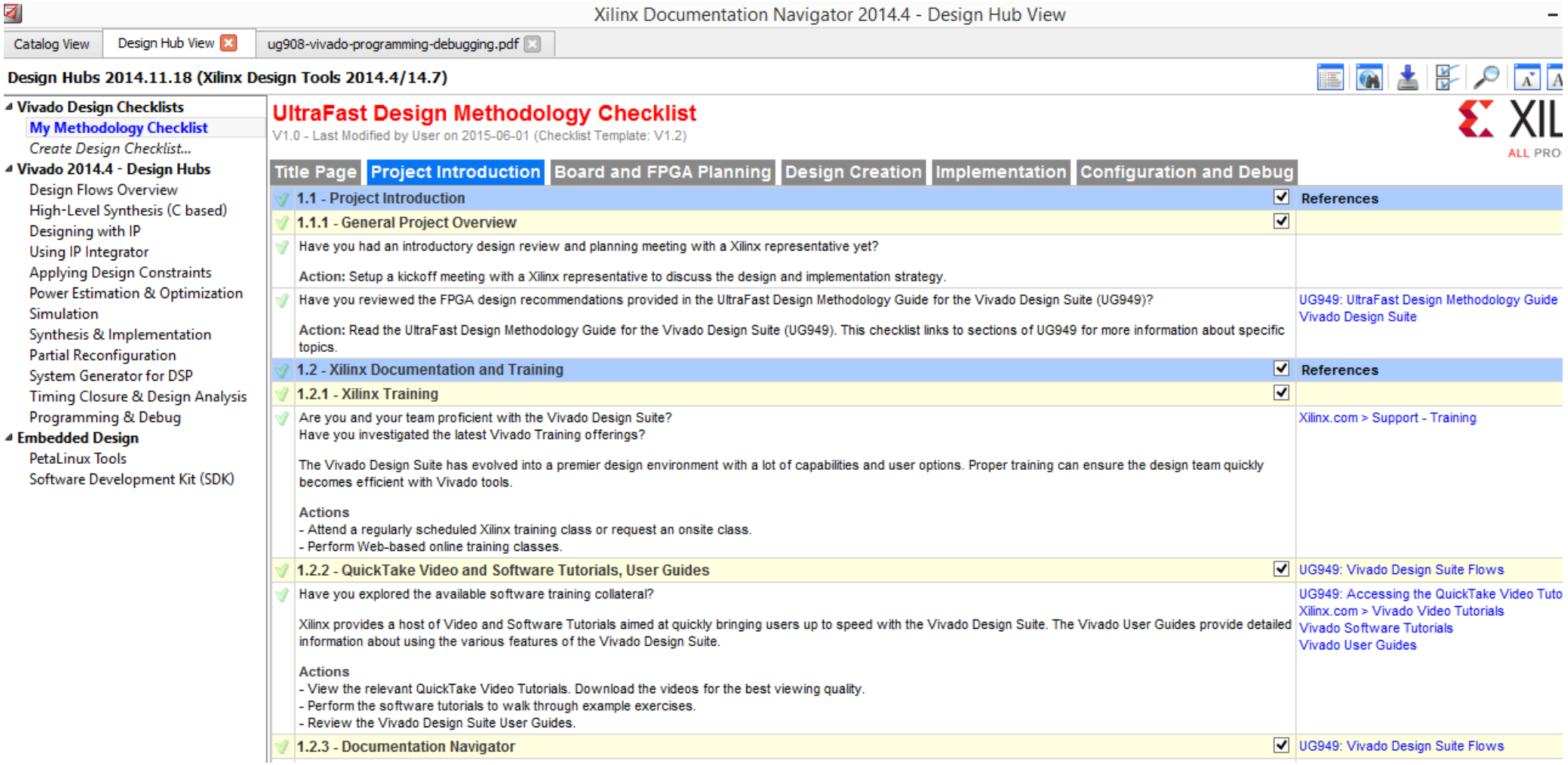
SOC Consumer Portable Design Complexity Trends (ITRS 2010)

New challenges

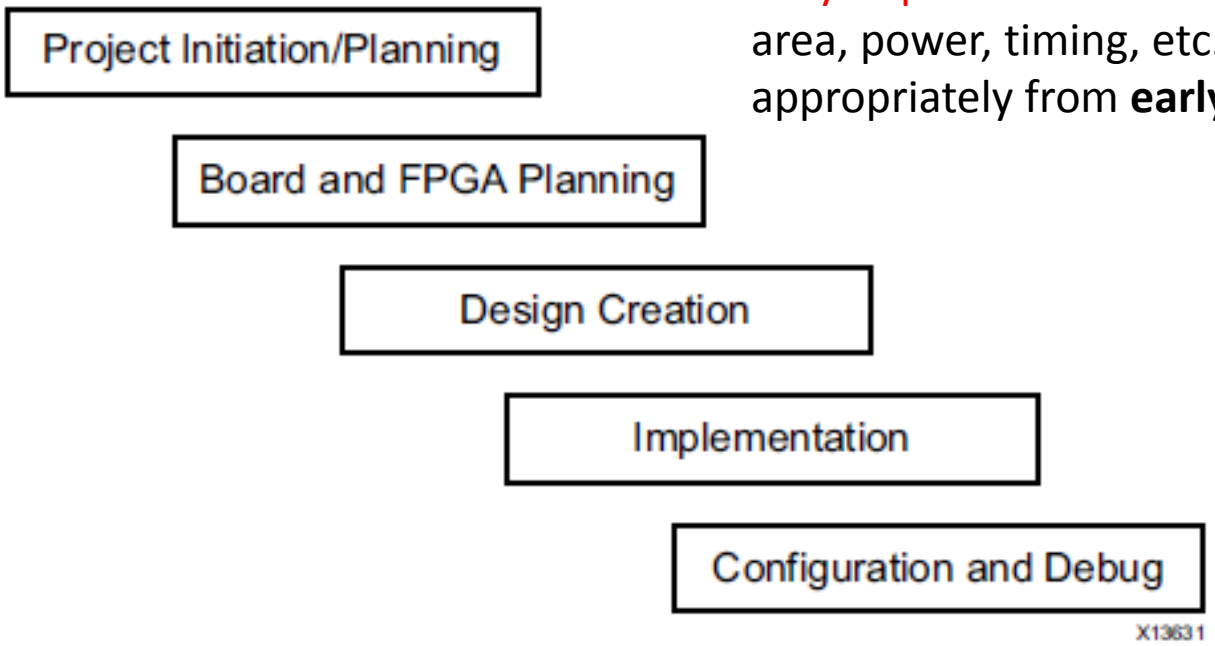
- New capacities => new and better tools

- **Need for Design Methodology**
- **Time to market**
- **Early estimation of project schedule and cost**
- **Best practices**

UltraFast Design Methodology



Design Process



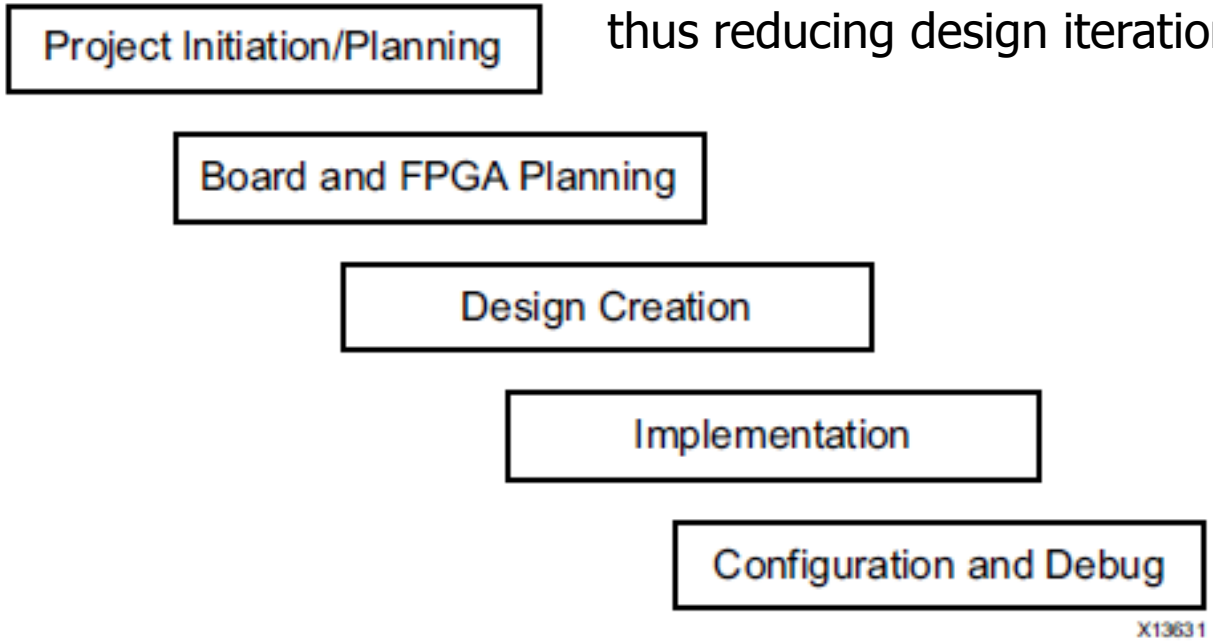
Very Important: monitoring design budgets (such as area, power, timing, etc.) and correcting designs appropriately from **early stages**

Very important: creating correct timing constraints for the system, before entering the implementation phase

Steps in design process

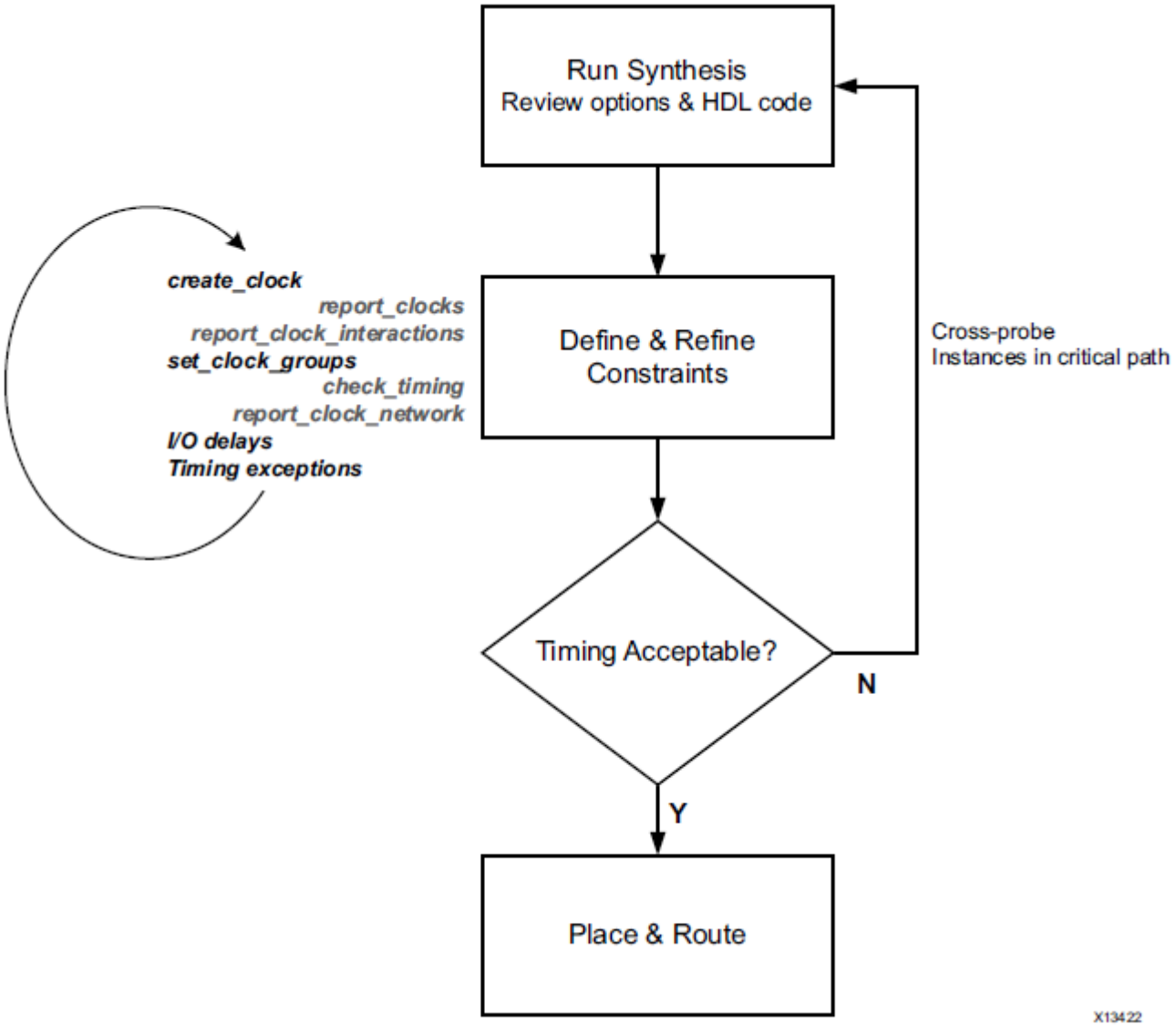
Design Process

The Vivado Design Suite provides you with design analysis capabilities at each design stage.
This allows for design and tool setting modifications earlier in the design processes where they have less overall schedule impact, thus reducing design iterations and accelerating productivity.



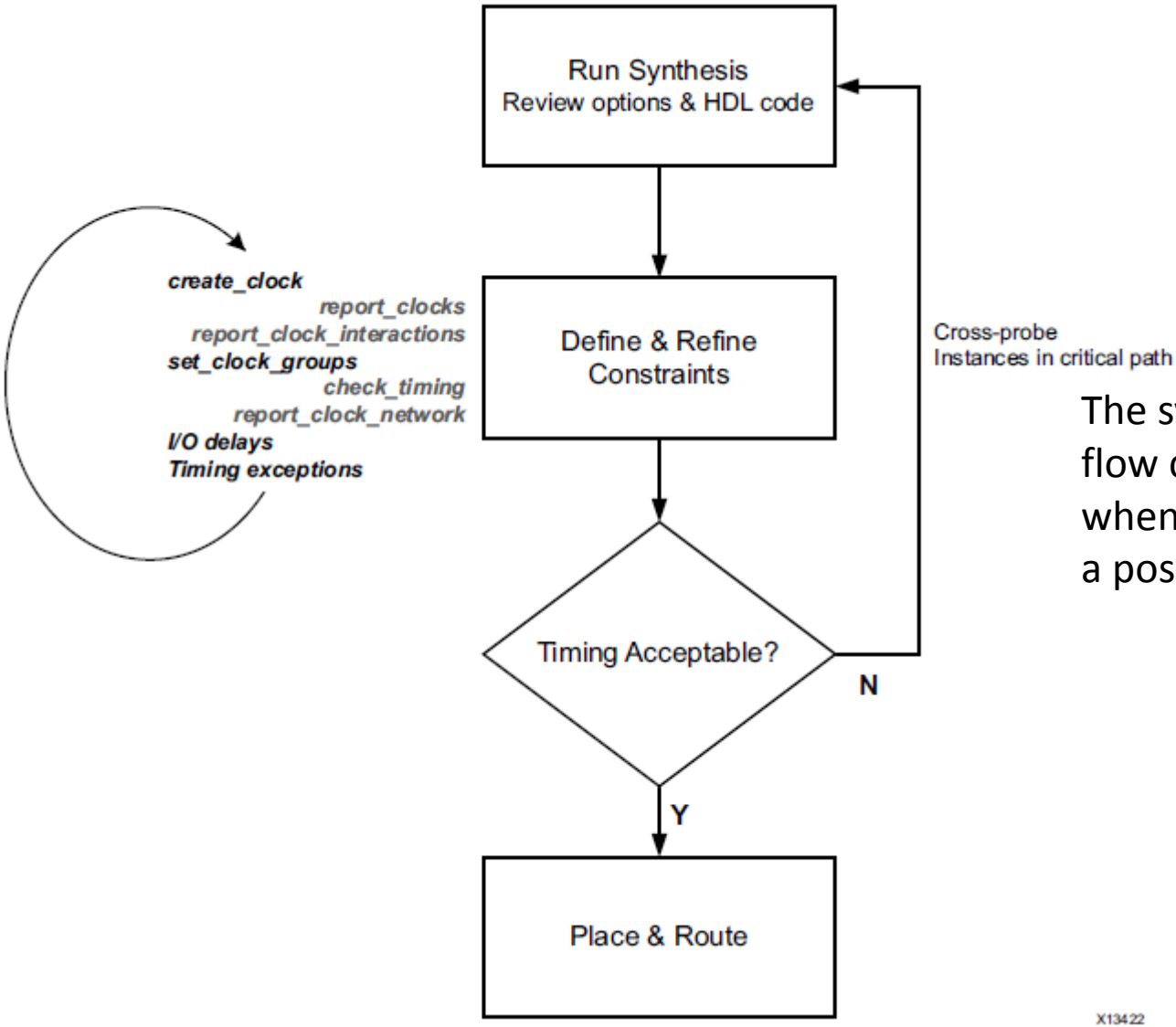
Steps in design process

Design Process



Design methodology for rapid convergence

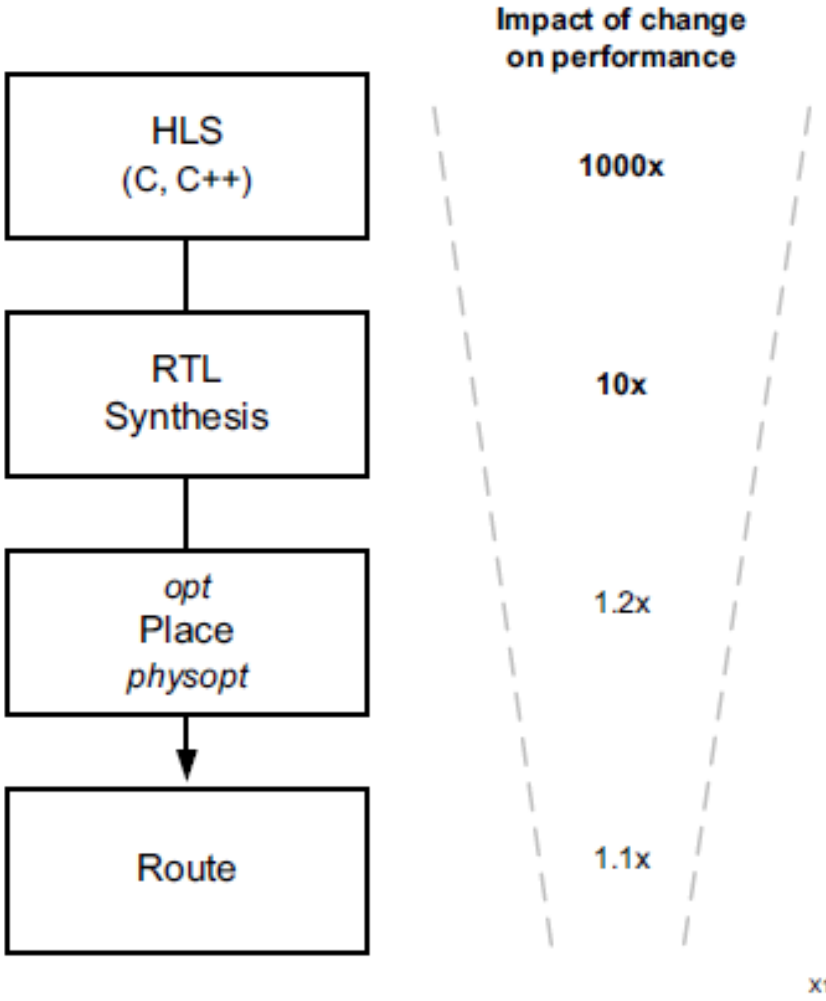
Design Process



The synthesis portion of the design flow can be considered complete when the design goals are met with a positive margin

Design methodology for rapid convergence

Design Process



Early stages in the design flow (C, C++, and HDL synthesis) have a much higher impact on design performance, density, and power than the later stages.

Impact of Design Changes throughout the flow

Design Flows

- ***RTL to Bitstream***
- ***Embedded Processor Design Flow***
- ***High-Level C-based Synthesis Flow***
- ***Partial Reconfiguration Design Flow***

Vivado Design Suite

- The Vivado Design Suite replaces Xilinx ISE® Design Suite of tools.
- It replaces all of the ISE Design Suite point tools, such as Project Navigator, Xilinx Synthesis Technology (XST), implementation, CORE Generator™ tool, Timing Constraints Editor, ISE Simulator (ISim), ChipScope™ Analyzer, Xilinx Power Analyzer, FPGA Editor, PlanAhead™ design tool, and SmartXplorer.
- All of these capabilities are now built directly into the Vivado Design Suite

Vivado Design Suite

The Vivado Design Suite supports the following established industry design standards:

- Tcl
- AXI4, IP-XACT
- Synopsys design constraints (SDC)
- Verilog, VHDL, SystemVerilog
- SystemC, C, C++, OpenCL

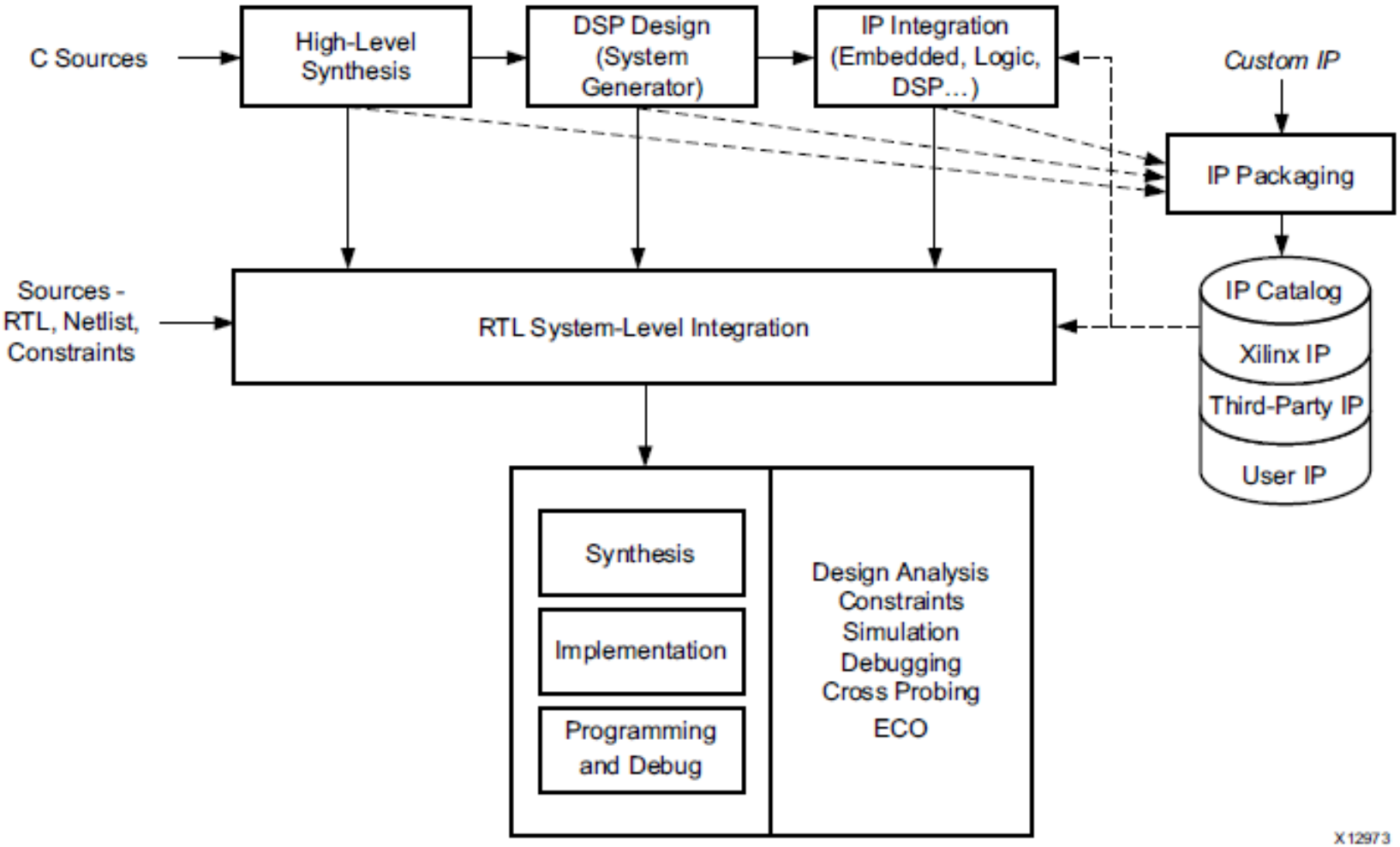
Vivado Design Suite

You can interact with the Vivado Design Suite using:

- GUI-based commands in the Vivado IDE
- Tcl commands entered in the Tcl Console in the bottom of the Vivado IDE,
- Tcl Commands entered in the Vivado Design Suite Tcl shell outside the Vivado IDE,
- or running Tcl scripts from the Vivado IDE or in the Vivado Design Suite Tcl shell
  - A Tcl script can contain Tcl commands covering the entire design synthesis and implementation flow, including all necessary reports generated for design analysis at any point in the design flow
- A mix of GUI-based and Tcl commands

Vivado High Level Design Flow

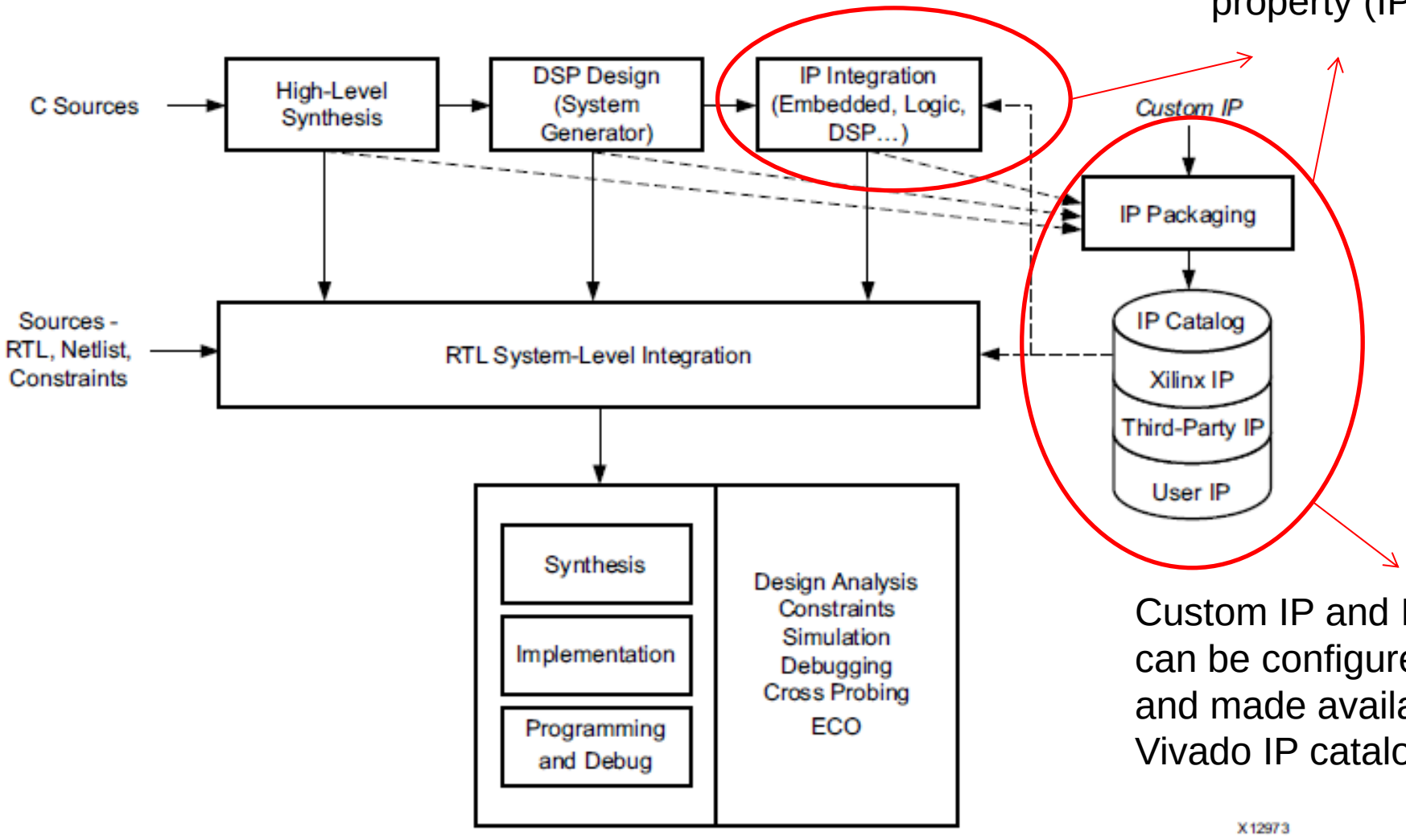
- Vivado Design Suite High-Level Design Flow



Vivado High Level Design Flow

- Vivado Design Suite High-Level Design Flow

Focused on intellectual property (IP)-centric design.



Custom IP and IP block designs can be configured and packaged and made available from the Vivado IP catalog

Use models

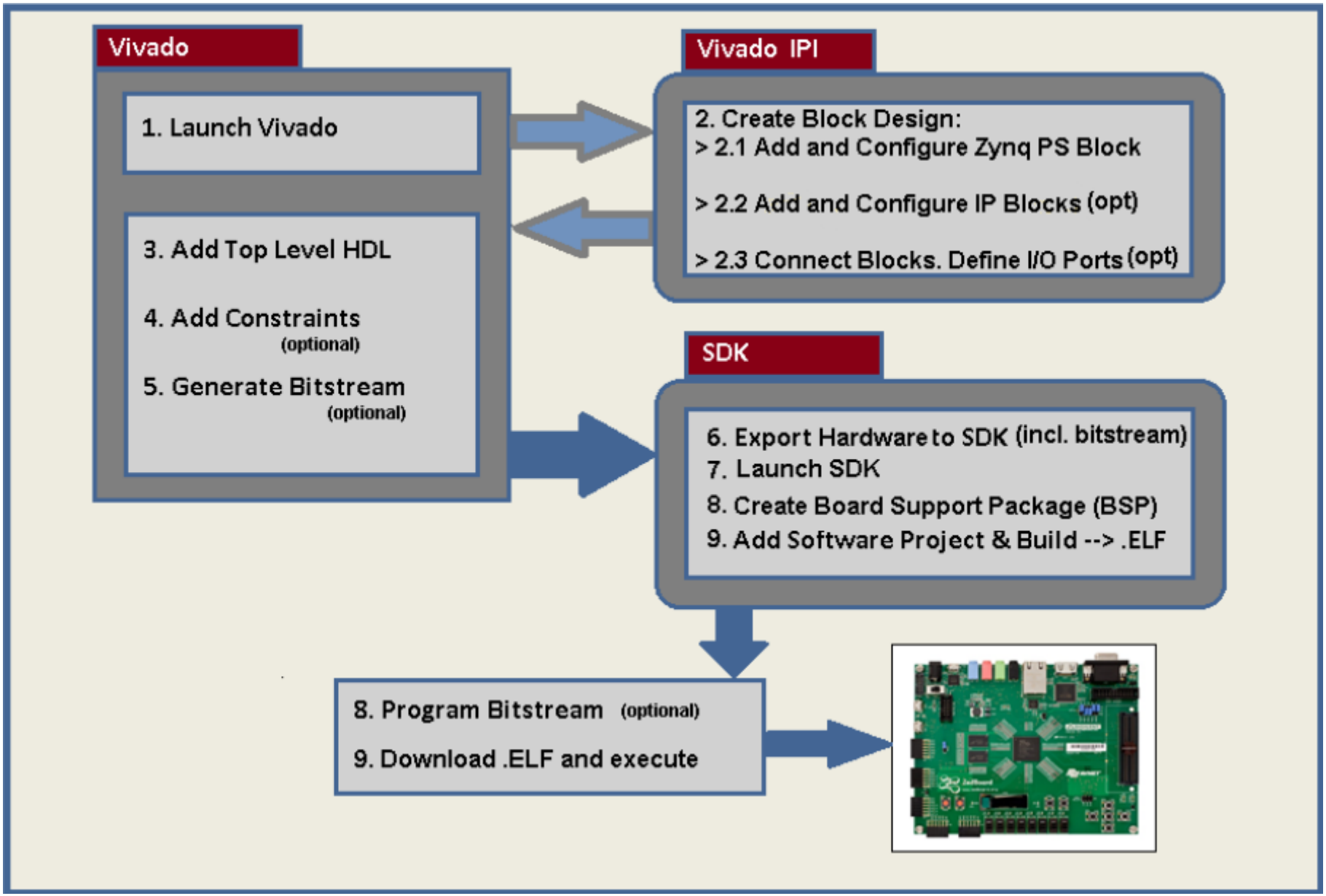
- **Project Mode**
  - A directory structure is created on disk
  - A runs infrastructure is used to manage the automated synthesis and implementation process and to track run status.
  - The entire design flow can be run with a single click.
  - The entire flow can also be scripted using Tcl commands.
- **Non-Project Mode**
  - Sources are accessed from their current locations
  - Each design step is run individually using Tcl commands, and design parameters and implementation options are set using Tcl commands.
  - You can save design checkpoints and create reports at any stage of the design process using Tcl. In addition, you can open the Vivado IDE at each design stage for design analysis and constraints assignment.

Use models

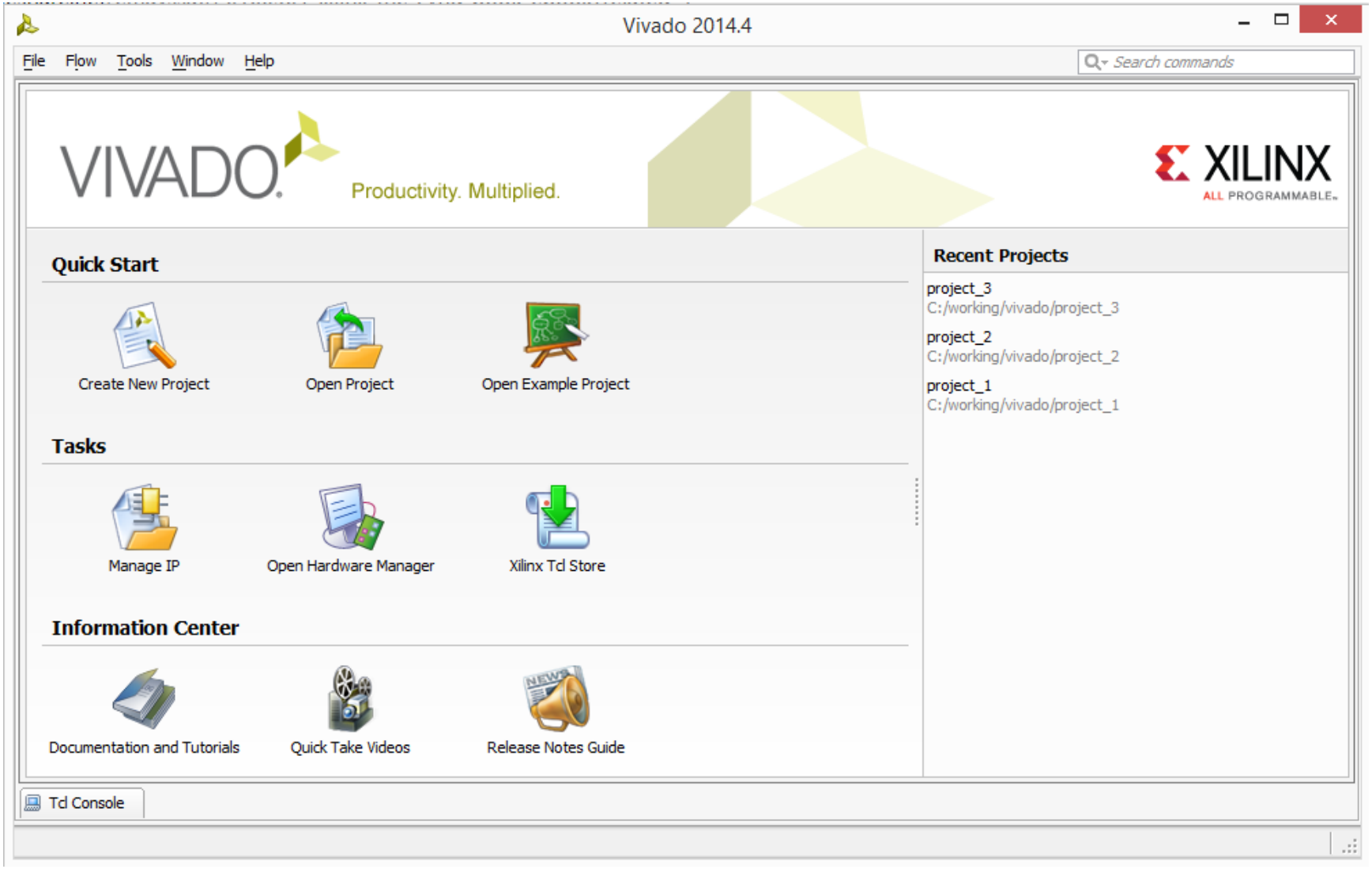
Flow Element	Project Mode	Non-Project Mode
Design Source File Management	Automatic	Manual
Flow Navigation	Guided	Manual
Flow Customization	Limited	Unlimited
Reporting	Automatic	Manual
Analysis Stages	Designs only	Designs and design checkpoints

Project Mode versus Non-Project Mode Features

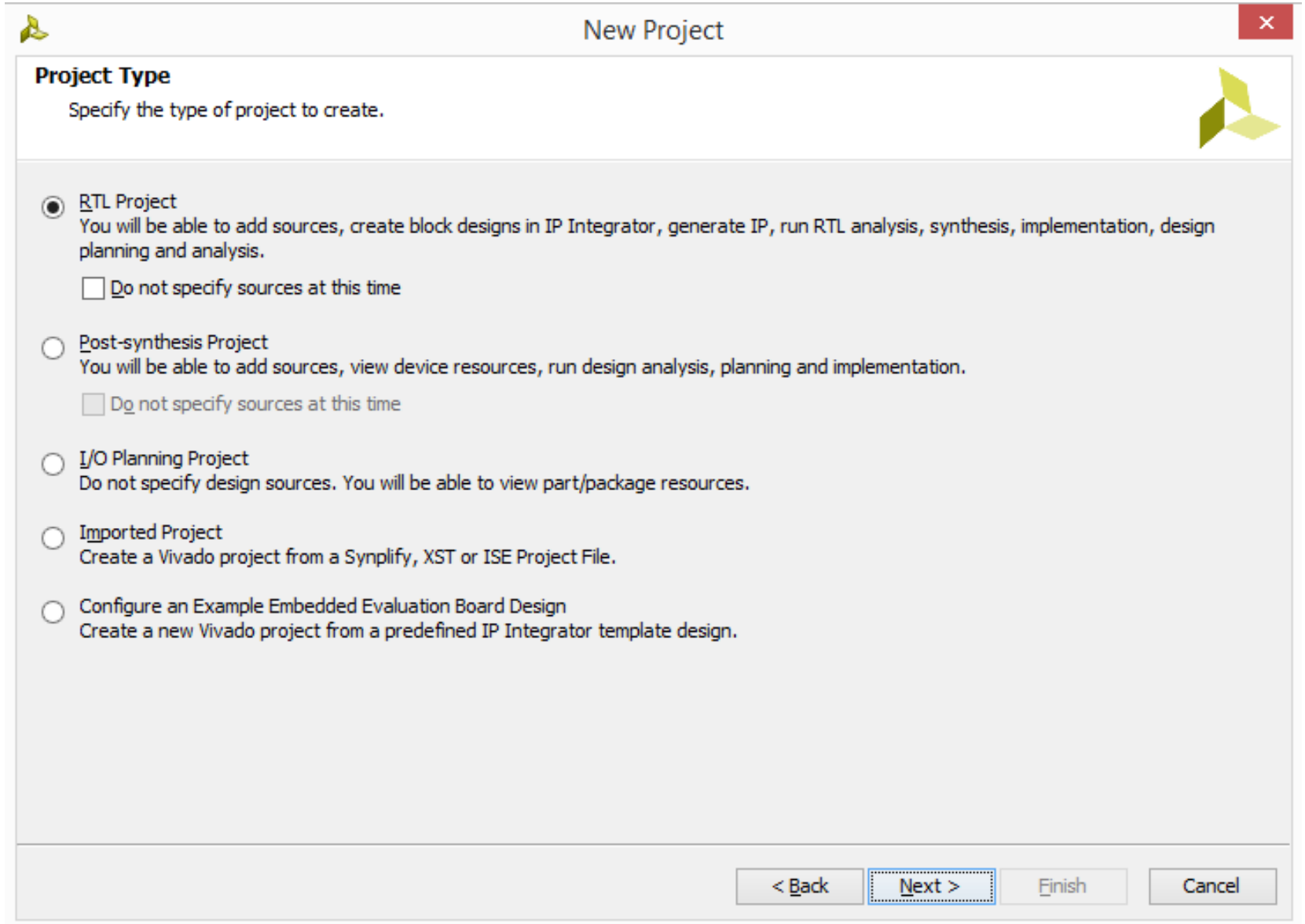
Design Process



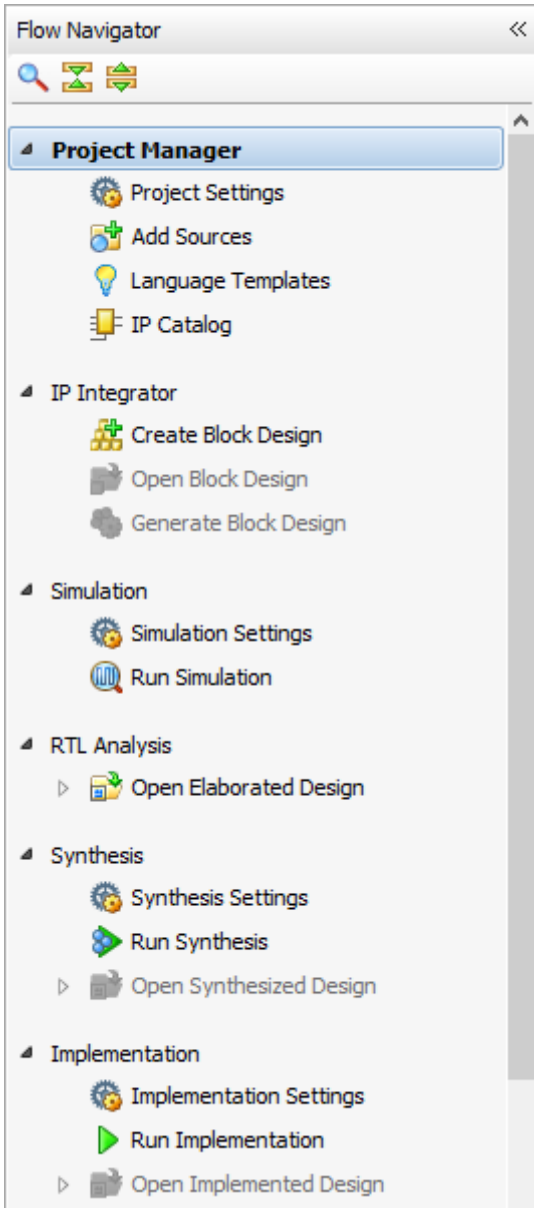
Launch Vivado



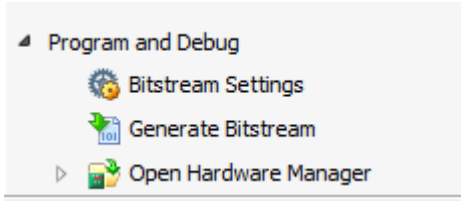
Creating Projects in Vivado



Flow Navigator in Vivado



It provides control over the major design process tasks, such as project configuration, synthesis, implementation, and bitstream generation



Main Design Flow Features

- Vivado Synthesis
- Vivado implementation
- Vivado timing analysis
- Vivado power analysis
- Bitstream Generation

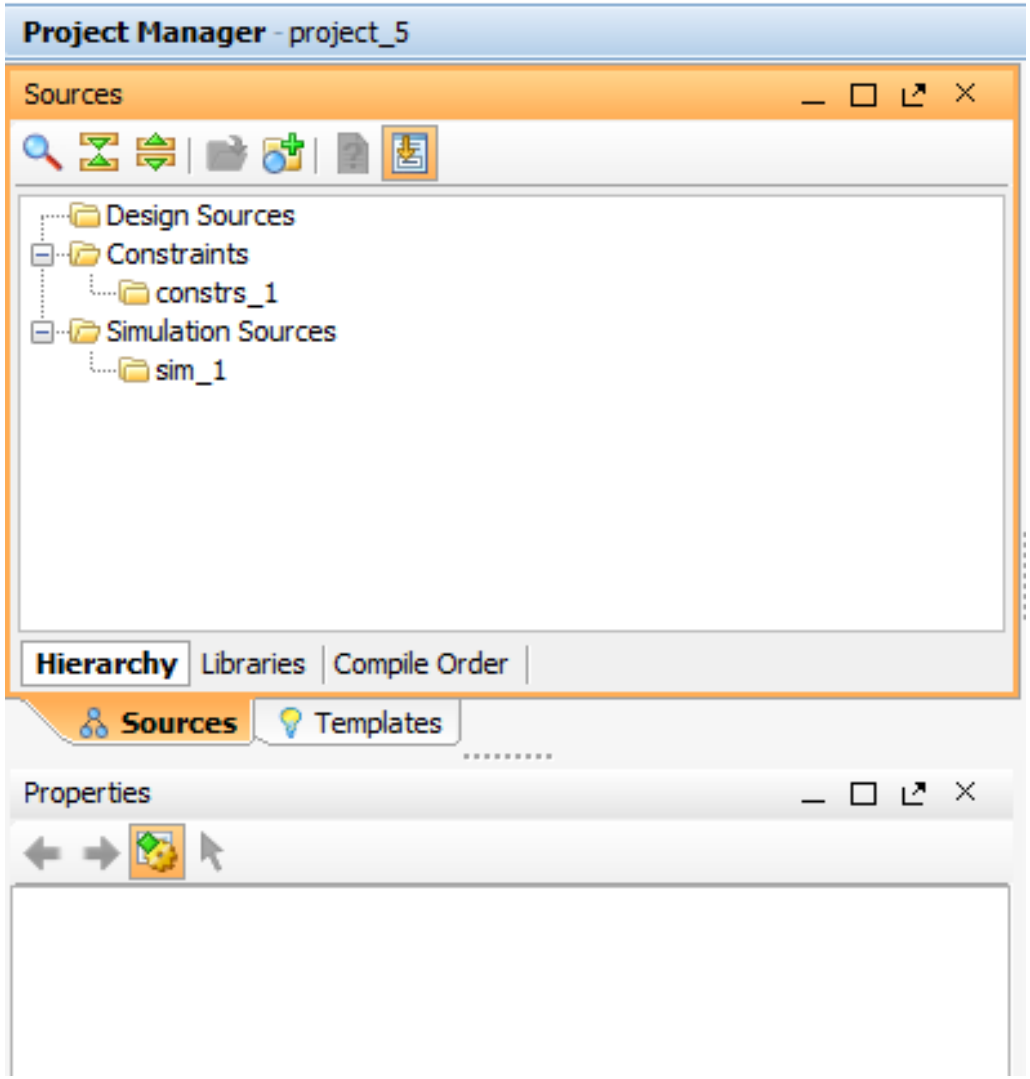
timing driven and use SDC or XDC format constraints

Various reports and analysis features are available at each stage of the design process

You can create multiple runs to experiment with different synthesis or implementation options, timing and physical constraints, or design configuration

Some advanced options are available for implementation, such as Vivado power optimization, Vivado physical optimizer, and run strategies, which assist you with design closure.

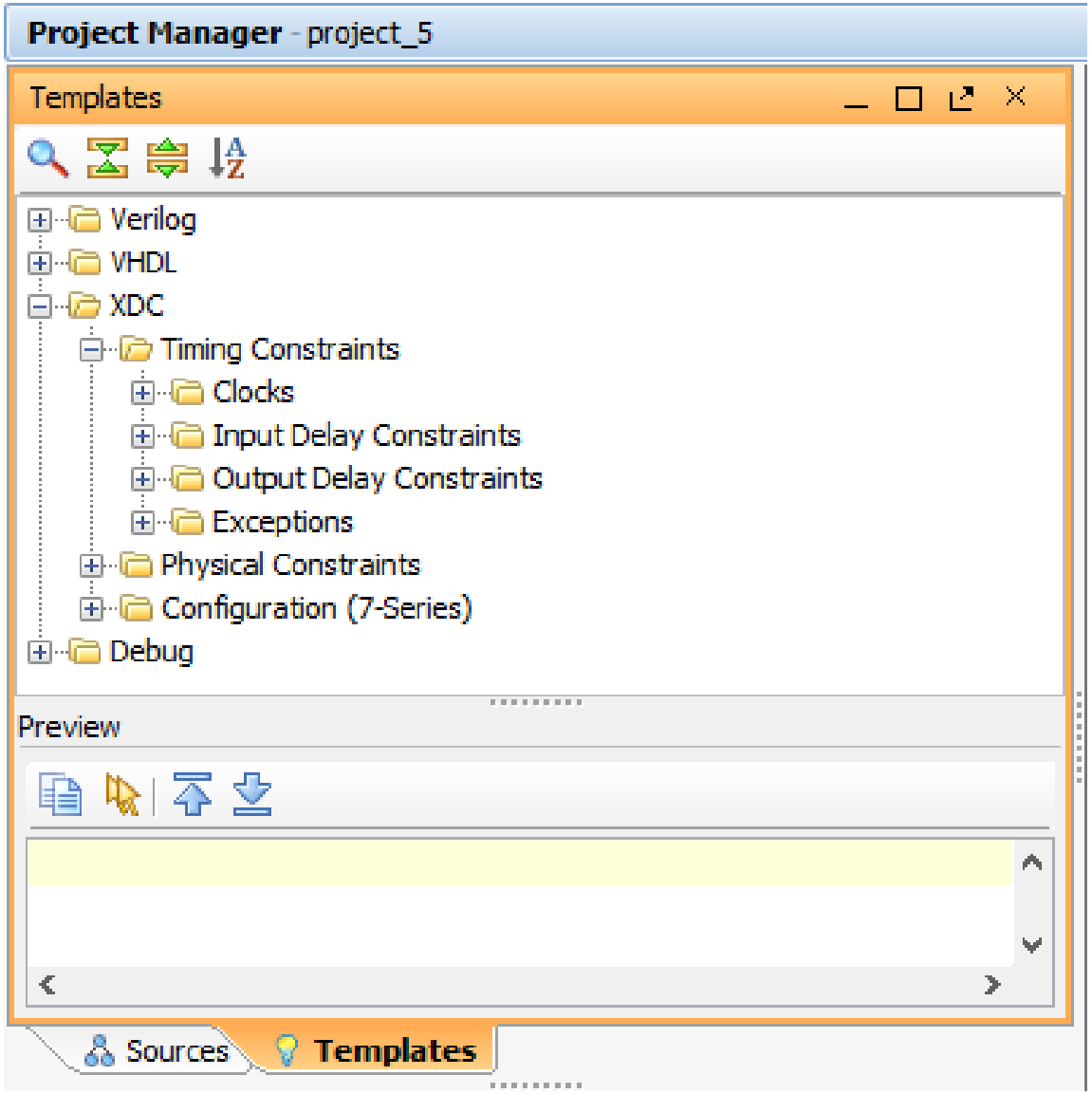
Performing System-Level Design Entry



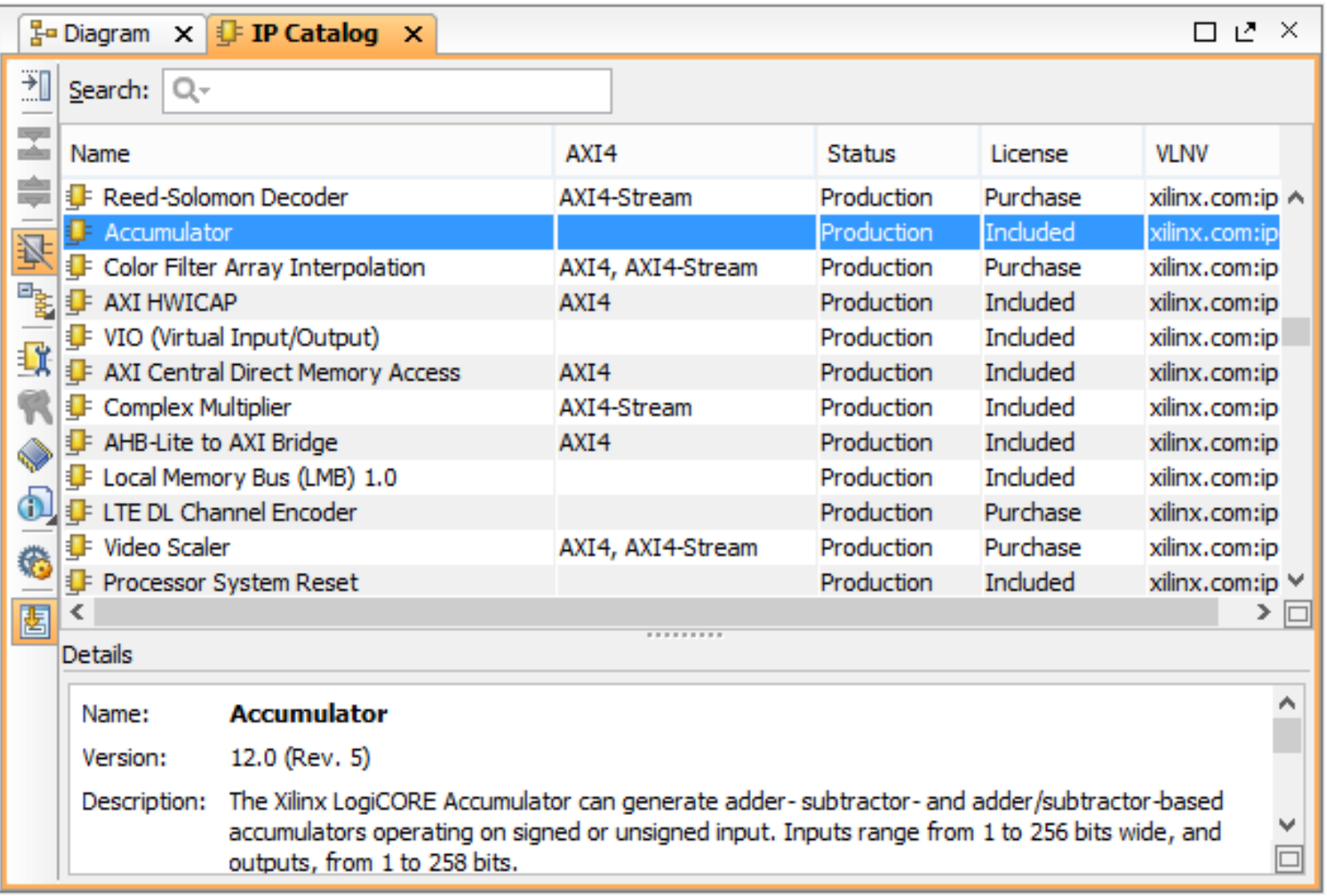
Automated Hierarchical Source File Compilation and Management

The Sources window updates the status of the project sources. A quick compilation of the design source files is performed

Working with templates

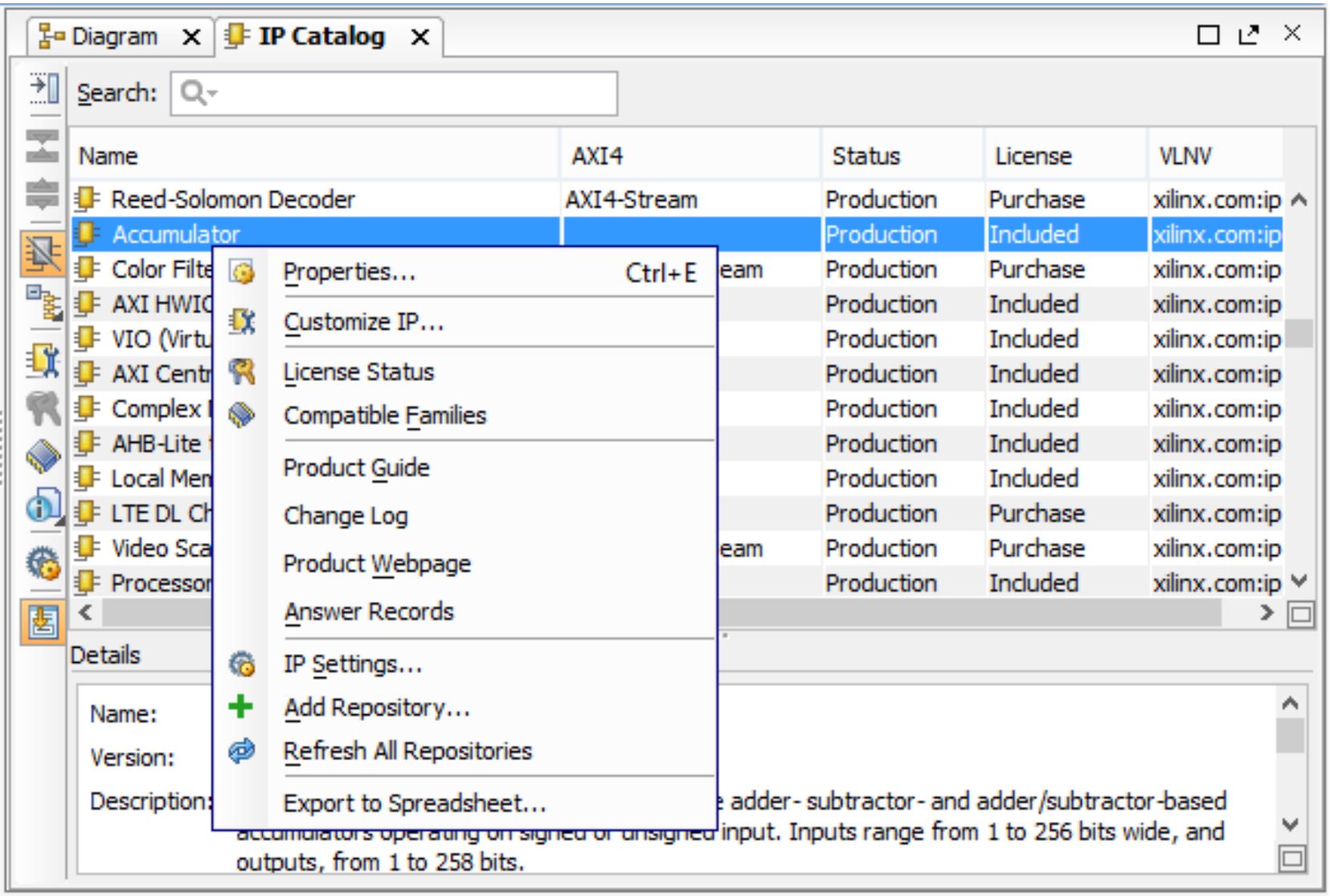


Working with IP



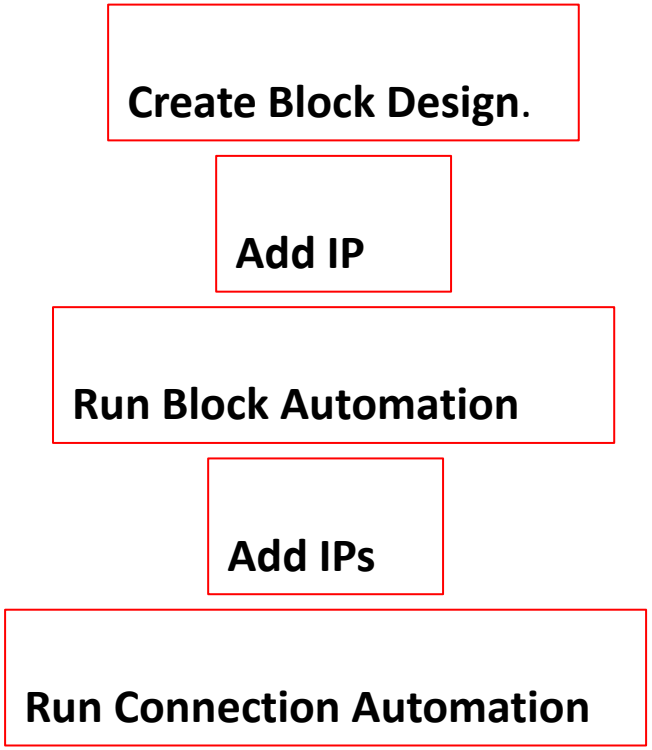
The IP catalog provides quick access to the IP for configuration, instantiation, and validation of IP.

Working with IP

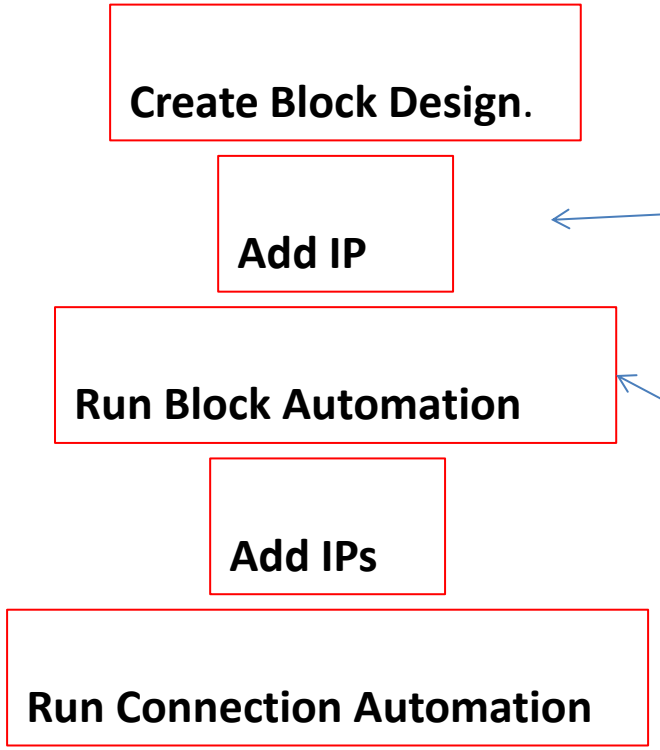


The IP catalog provides quick access to the IP for configuration, instantiation, and validation of IP.

Creating an IP Integrator design



Creating an IP Integrator design

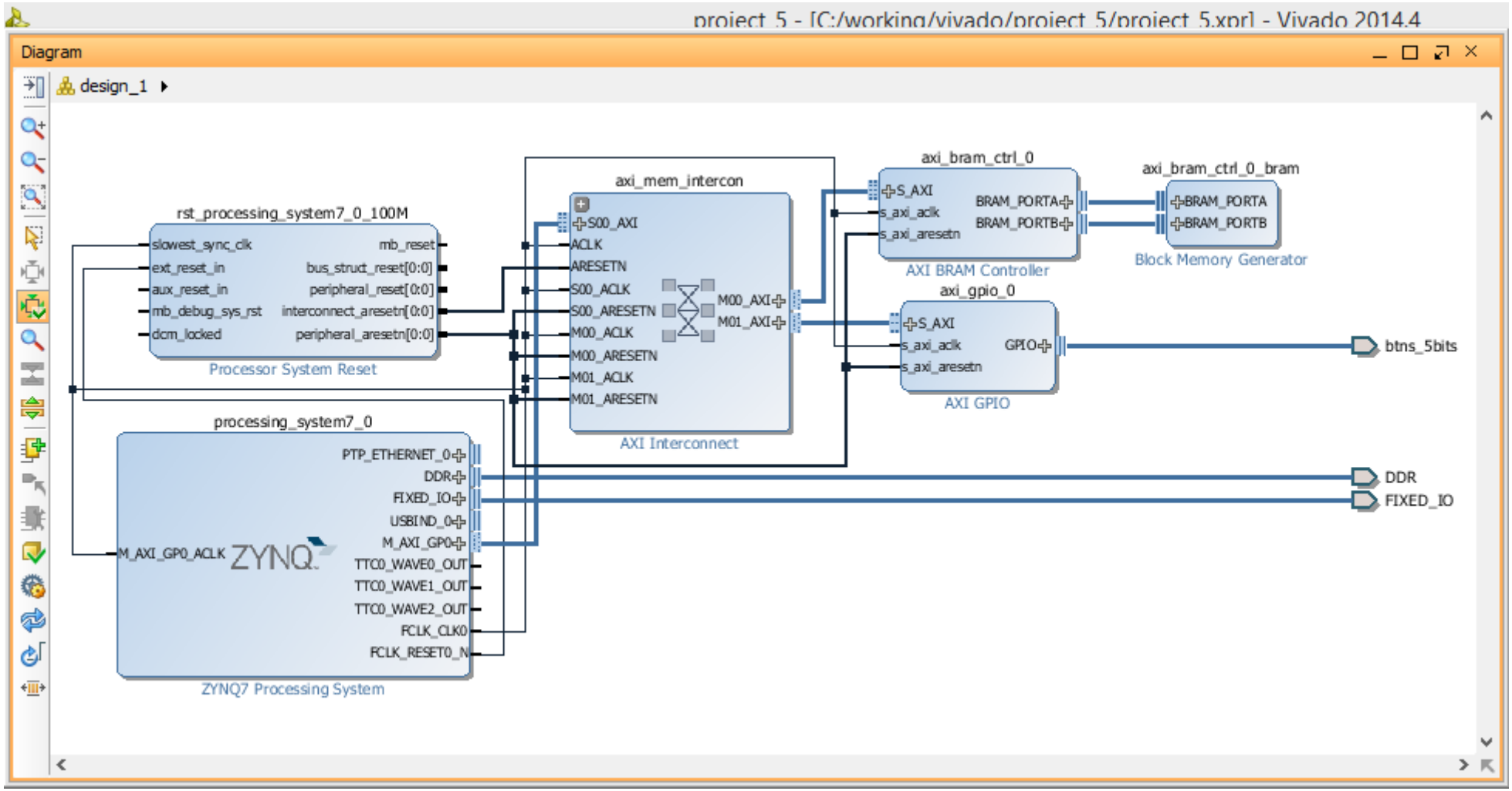


Selecting IP from IP catalog.

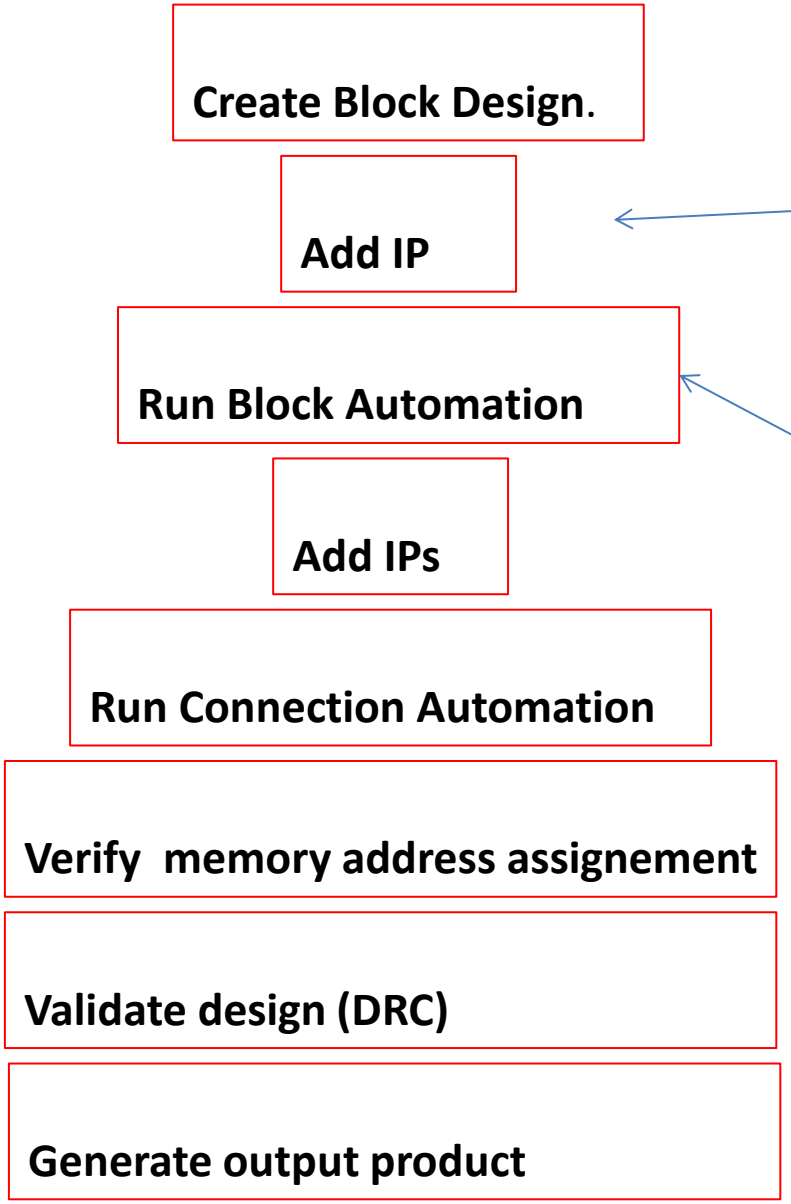
In embedded design the ZYNQ 7 Processing System IP is selected

Applies current board preset and generate external connections for FIXED_IO triggers and DDR interfaces

Creating an IP Integrator design



Creating an IP Integrator design

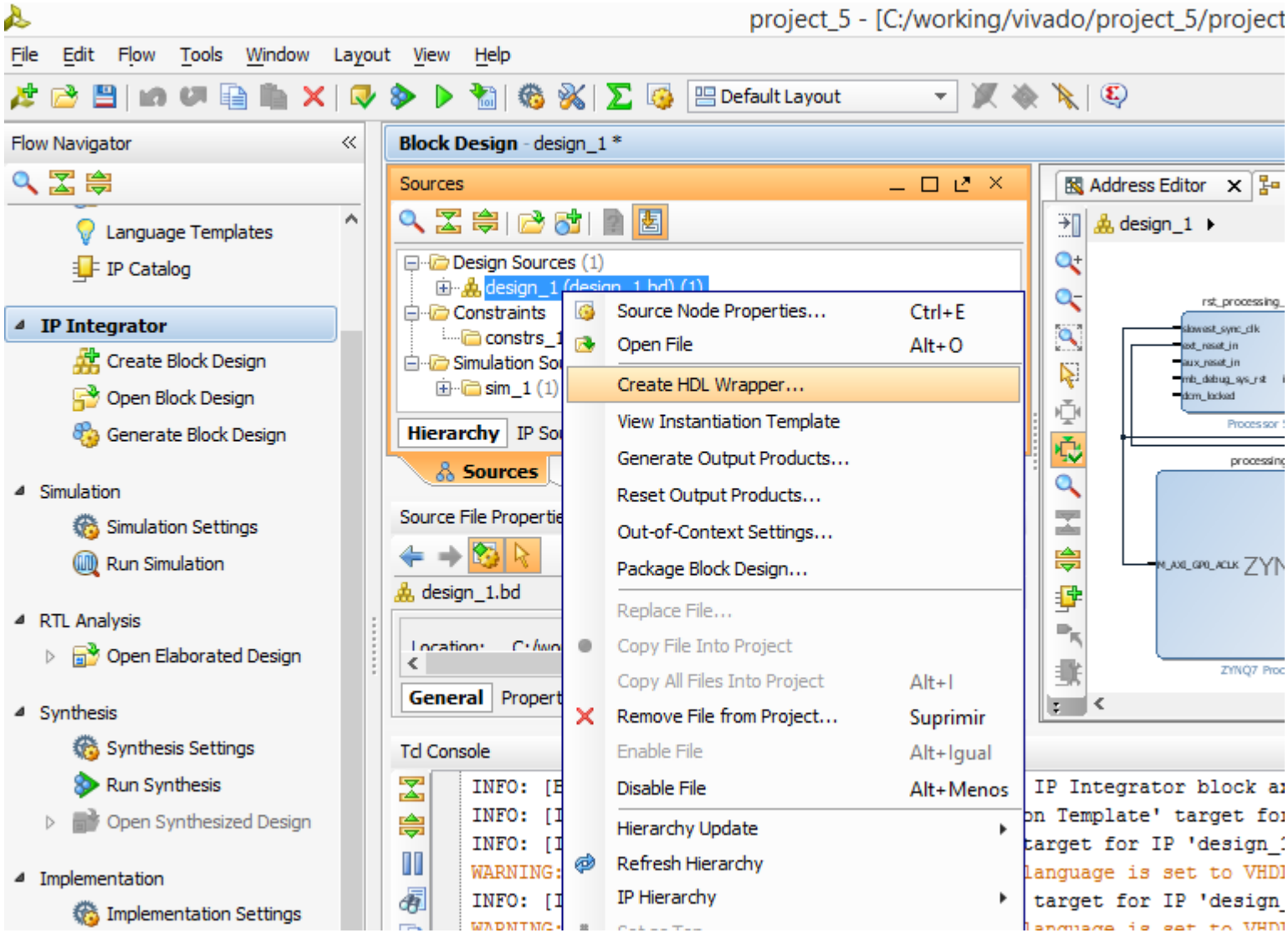


Selecting IP from IP catalog.

In embedded design the ZYNQ 7 Processing System IP is selected

Applies current board preset and generate external connections for FIXED_IO triggers and DDR interfaces

Creating an IP Integrator design



Generating VHDL wrapper

Embedded Processor Hardware Design

PS + PL

- Creating an embedded processor hardware design involves the IP integrator feature of the Vivado Design Suite. In the Vivado IP integrator environment, you instantiate, configure, and assemble the processor core and its interfaces.
- After the design is compiled through implementation, it is exported to the Xilinx Software Development Kit (SDK) for use in the software development and validation flows
- Simulation and debug features

Embedded System Hardware Design

PS + PL

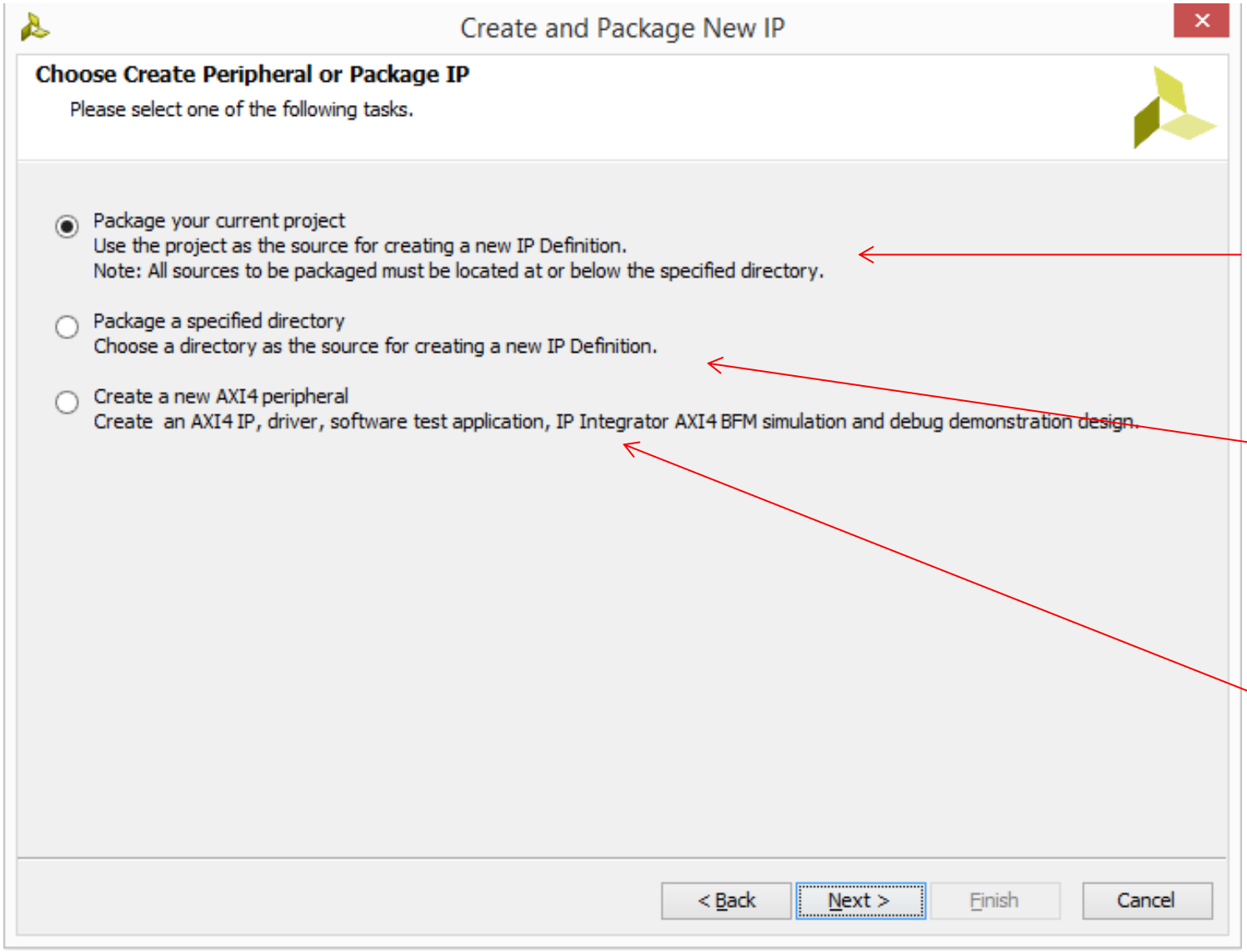
- Hardware Ips can be added in PL part.
- IP can include logic, embedded processors, digital signal processing (DSP) modules, or C-based DSP algorithm designs.
- Custom IP is packaged following IP-XACT protocol and then made available through the Vivado IP catalog
- Xilinx IP utilizes the AMBA AXI4 interconnect standard to enable faster system-level integration

Embedded System Hardware Design

PS + PL

- You can interactively configure and connect IP using a block design style interface
- Connection automation is provided as well as a set of DRCs to ensure proper IP configuration and connectivity.
- Connecting the IP using standard interfaces saves time over traditional RTL-based connectivity.
- These IP block designs are then validated, packaged, and treated as a single design source.

Create and Package new IP

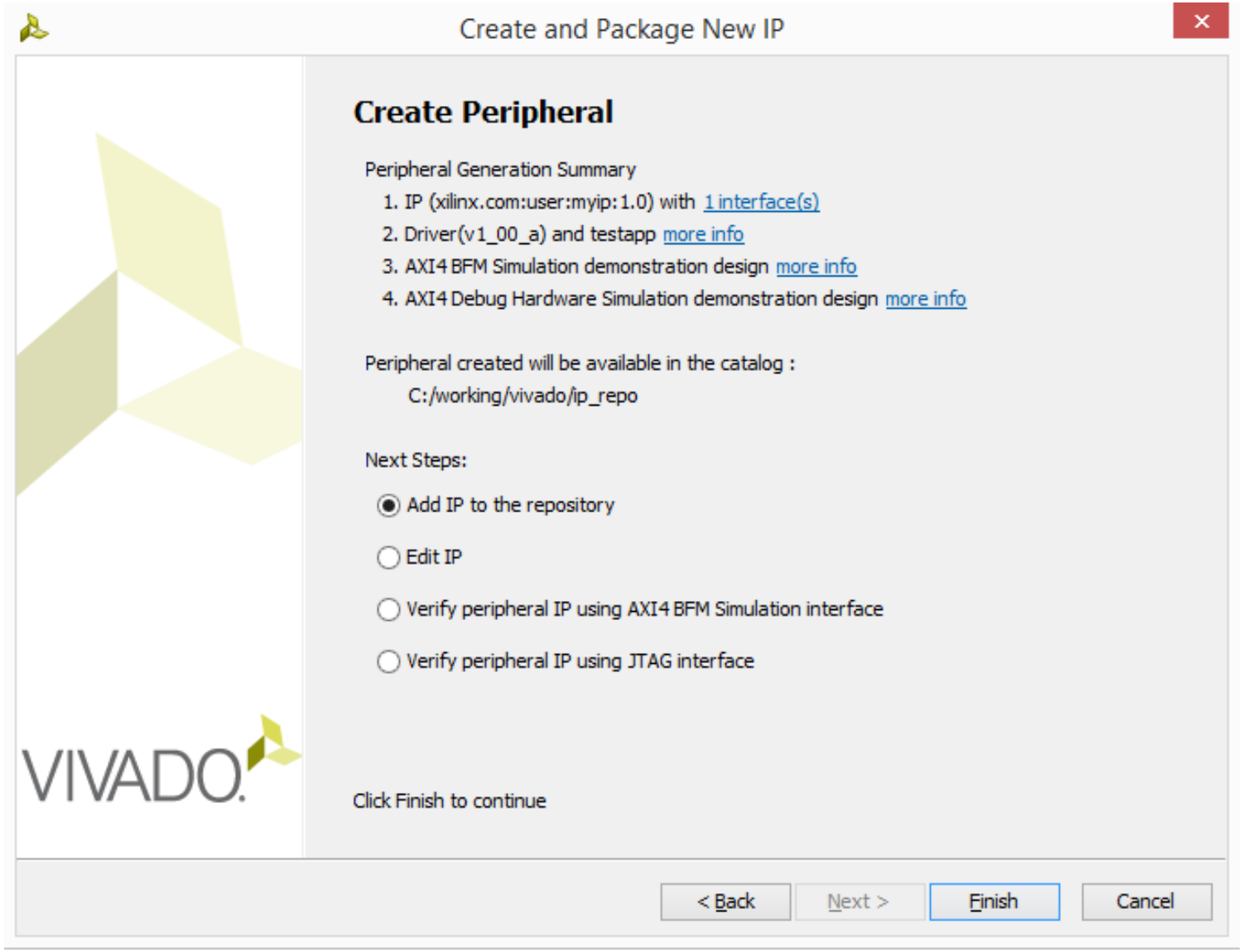


This option lets you package the files associated with your current Vivado project

This option lets you package the files underneath a specific directory within the file system

This option lets you create and package a new AXI4 peripheral with master or slave interface mode.

Create and Package new IP



DSP Design

- The Vivado Design Suite is also integrated directly with the **Xilinx System Generator tool** to provide a solution for implementing DSP functions
- The C-based High-Level Synthesis (HLS) tool within the Vivado Design Suite enables you to describe various DSP functions in the design using C, C++, System C, and OpenCL languages
- C-to-RTL synthesis transforms the C-based design into an RTL module that can be packaged and implemented with the rest of the design. This module can then be instantiated into the RTL design or within Vivado IP integrator

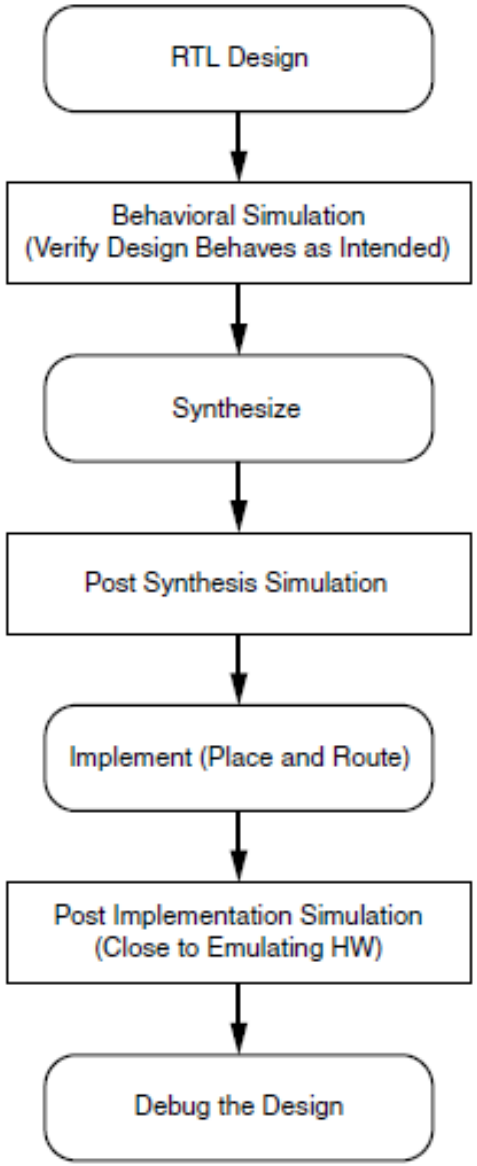
System Generator is a DSP design tool from Xilinx that enables the use of the MathWorks model-based Simulink® design environment for FPGA design.

Design Analysis and Verification

- Design analysis and verification is enabled at each stage of the flow.

- Design analysis features include logic simulation, I/O and clock planning, power analysis, constraint definition and timing analysis, design rule checks (DRC), visualization of design logic, analysis and modification of implementation results, and programming and debugging.

Simulation flow

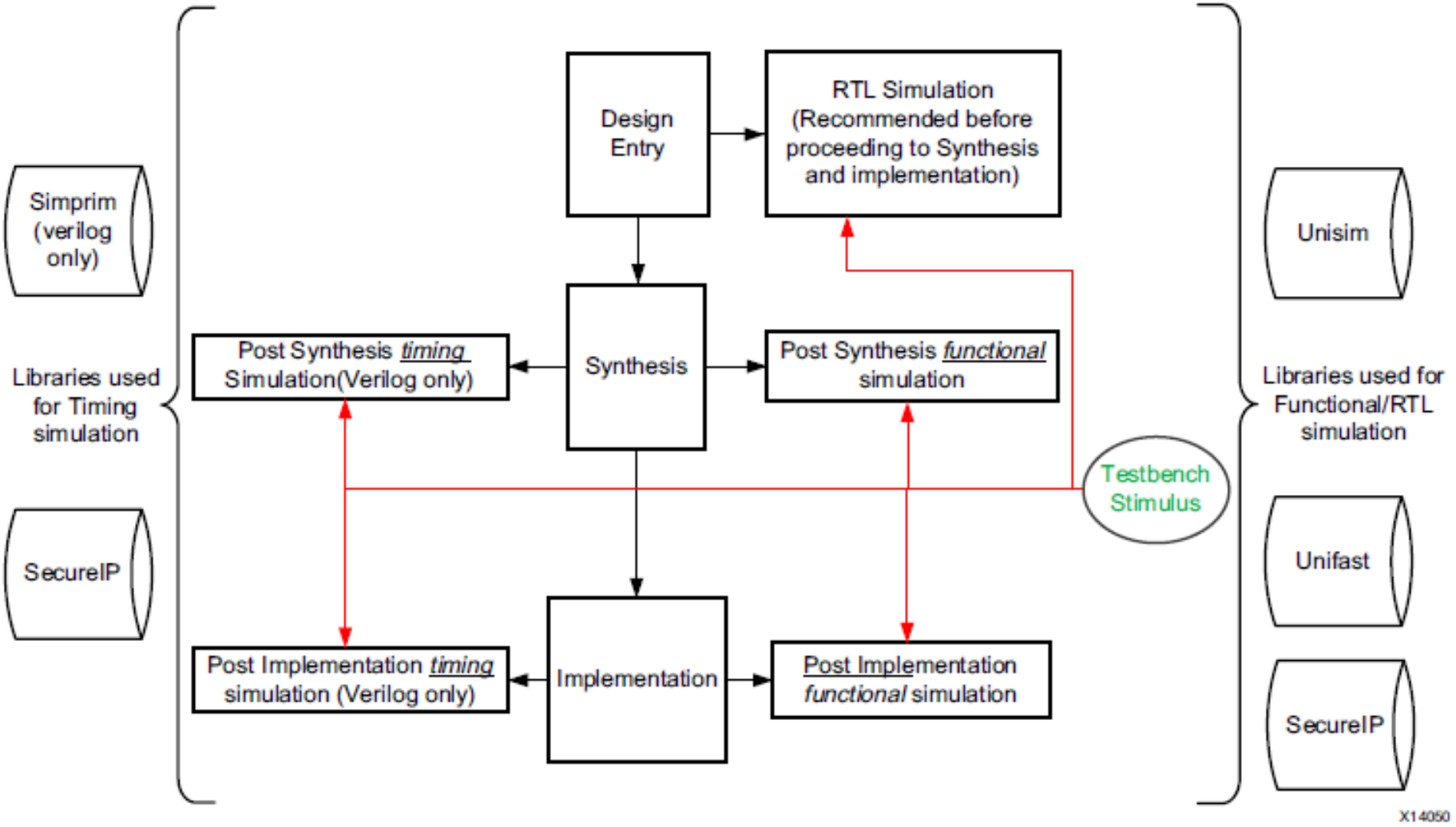


The process of simulation includes:

- Creating a test bench that reflects the simulation actions you want to run
- Selecting and declaring the libraries you need to use
- Compiling your libraries (if using a third-party simulator)
- Netlist generation (if performing post-synthesis or post-implementation simulation)
- Understanding the use of global reset and 3-state in Xilinx devices

Simulation flow

Simulation at various points in the design flow



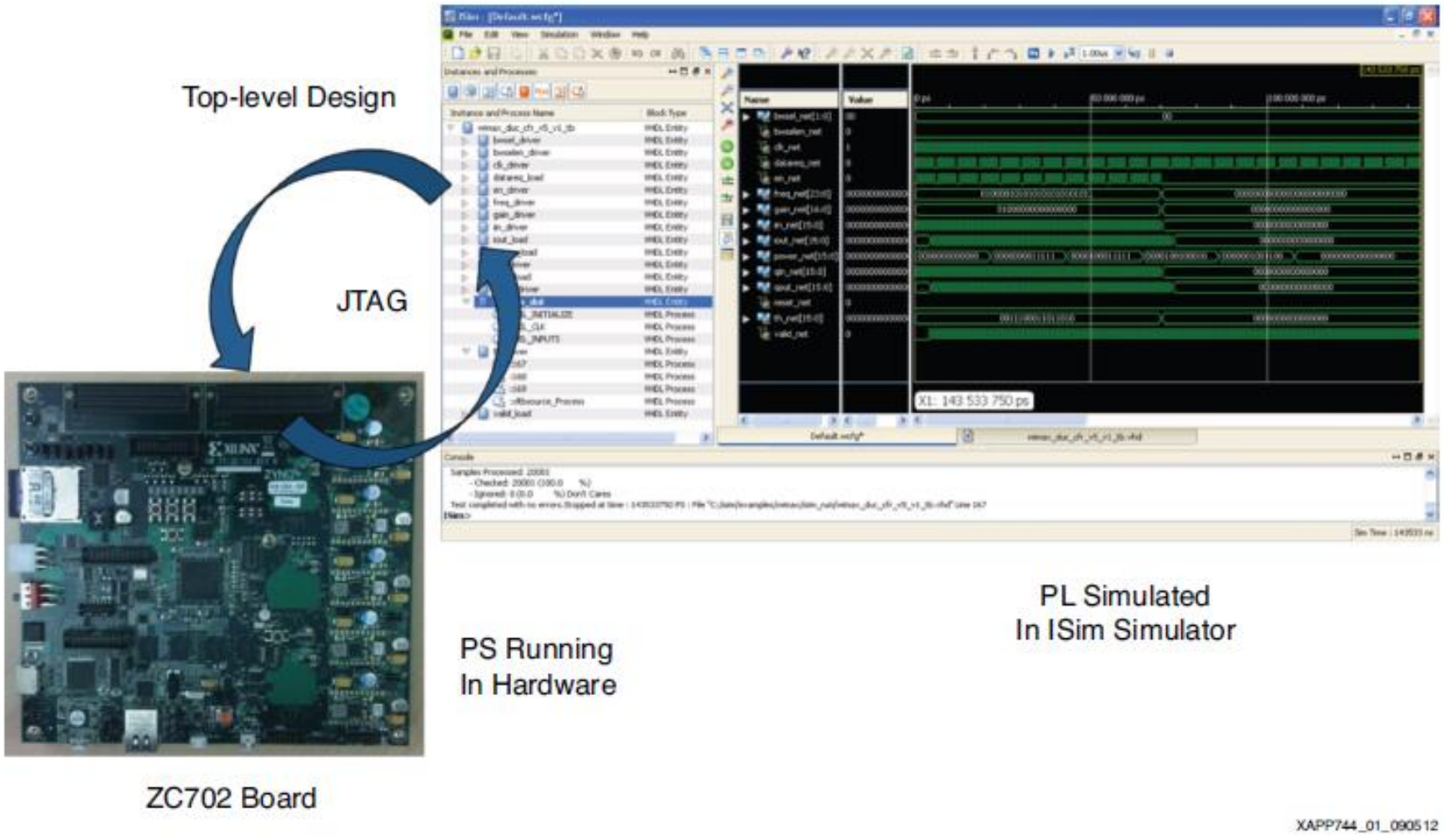
Simulation libraries

Library Name	Description	VHDL Library Name	Verilog Library Name
UNISIM	Functional simulation of Xilinx primitives.	UNISIM	UNISIMS_VER
UNIMACRO	Functional simulation of Xilinx macros.	UNIMACRO	UNIMACRO_VER
UNIFAST	Fast simulation library.	UNIFAST	UNIFAST_VER
SIMPRIM	Timing simulation of Xilinx primitives.	N/A	SIMPRIMS_VER [a]
SECUREIP	Simulation library for both functional and timing simulation of Xilinx device features, such as the: • PCIe® IP • Gigabit Transceiver	SECUREIP	SECUREIP

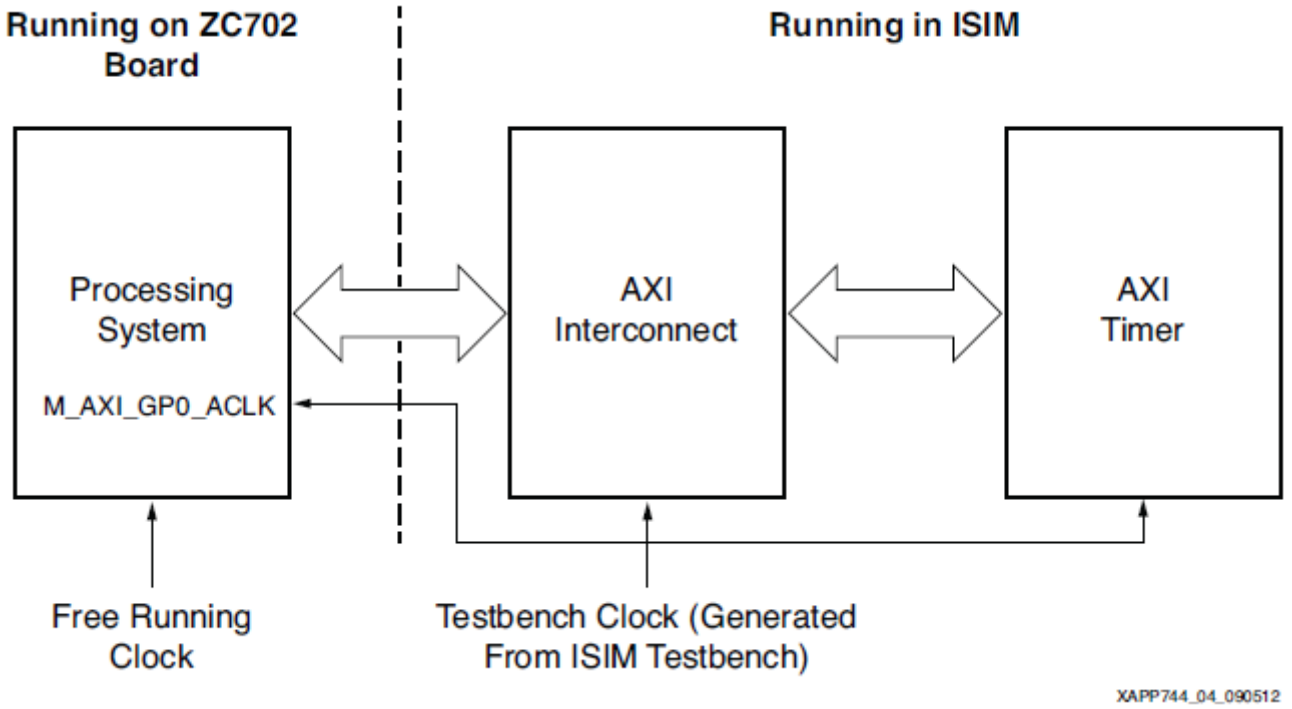
Soported Simulators

- Vivado simulator: Integrated in the Vivado IDE.
- Mentor Graphics QuestaSim/ModelSim: Integrated in the Vivado IDE.
- Cadence Incisive Enterprise Simulator (IES).
- Synopsys VCS and VCS MX.
- Aldec Active-HDL and Rivera-PRO*.

HIL simulation



HIL simulation



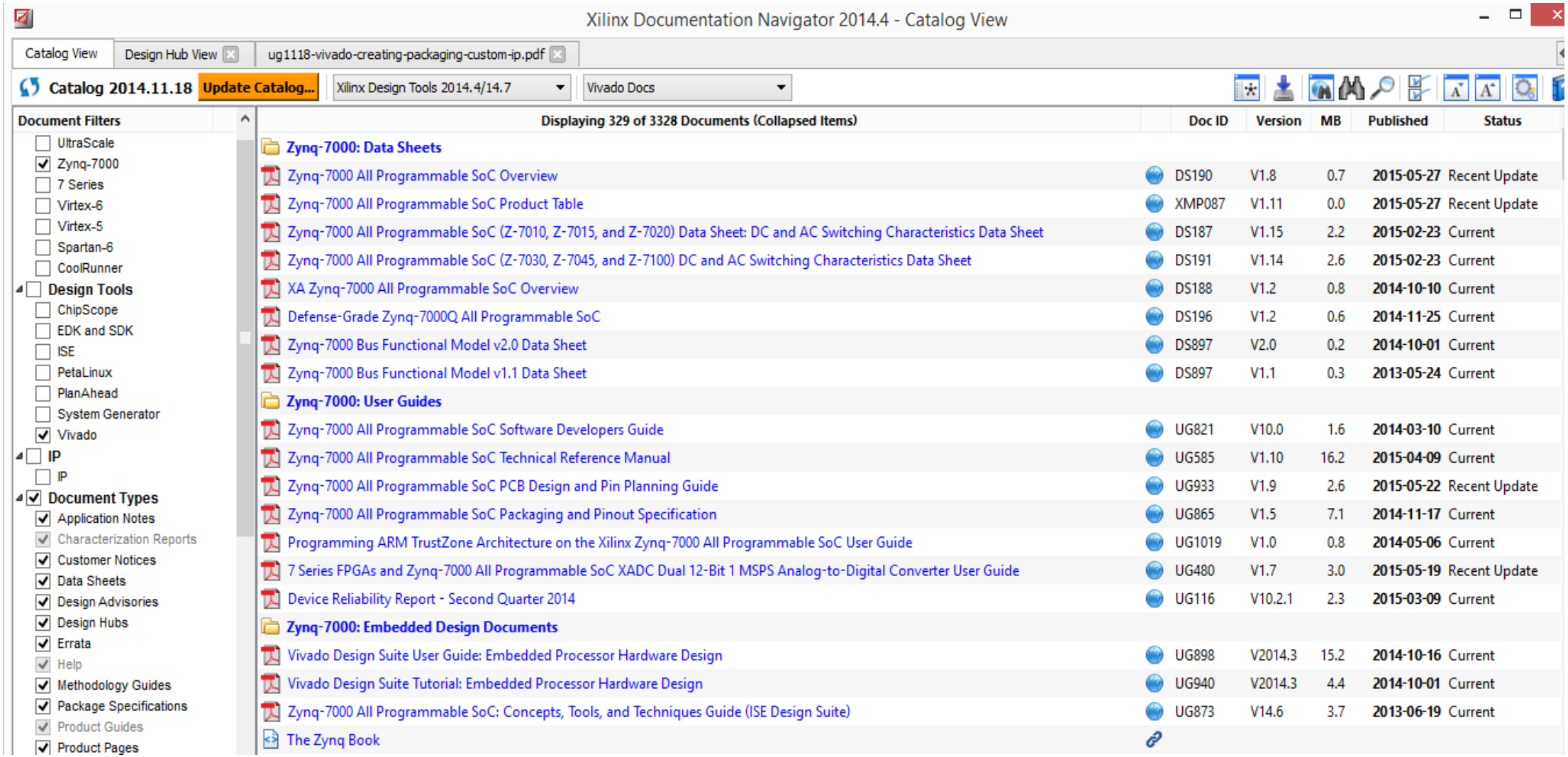
BFM simulation

- The Xilinx® LogiCORE™ IP AXI Bus Functional Model (BFM) cores, developed for Xilinx by Cadence® Design Systems, support the simulation of customer-designed AXI-based IP.

- AXI BFM cores support all versions of AXI (AXI3, AXI4, AXI4-Lite, and AXI4-Stream).

- The BFMs are encrypted Verilog modules.
- BFM operation is controlled by using a sequence of Verilog tasks contained in a Verilog-syntax text file

Documentation and tutorials



Thank you for your attention

Contact information

Juliodaniel.dondo@uclm.es