***System-Level Design Entry***

**Chapter 1: Introduction**

System-level design entry consists of setting up your design, including creating a project (if applicable), creating and adding source files, adding block design and IP cores, elaborating the RTL design, and inserting and configuring debug information. You can enter your design using the graphical user interface (GUI), known as the Vivado Integrated Design Environment (IDE), or using Tcl commands and scripts.

**Chapter 2: Working with Projects**

When working in Project Mode, you can enter your design using various project types. This

chapter describes each project type and explains how to create and manage projects. It also

covers the Project Summary, Project Settings, and how to create a project using a Tcl script.

**1 Project Types**

• RTL Projects

You can add RTL source files, IP from the AMD IP catalog, block designs created in the Vivado IP integrator, digital signal processing (DSP) sources, and EDIF netlists for hierarchical modules. IP can include XCI or XCIX files generated by the Vivado tools, legacy XCO files generated by the CORE Generator tool, and precompiled EDIF or NGC-format netlists.

• Post-Synthesis Projects

You can create projects using synthesized netlists created using Vivado synthesis, XST, or any

supported third-party synthesis tool.

RECOMMENDED: Always reference the Vivado IP using the XCI or XCIX file. AMD does not recommend reading just the IP DCP file. While the DCP does contain constraints, it does not provide other output products that an IP could deliver and that could be needed, such as ELF, COE, and Tcl scripts.

• I/O Planning Projects

You can perform clock resource and I/O planning early in the design cycle by creating an empty I/O planning project. You can define I/O ports within the Vivado IDE or import them with either comma separated value (CSV) or XDC input files. You can also create empty I/O planning projects to explore the logic resources available on the different device architectures.

• Imported Projects

You can import RTL project data from Synopsys Synplify into the Vivado tools. The project source files and compilation order are imported, but implementation results and settings are not.

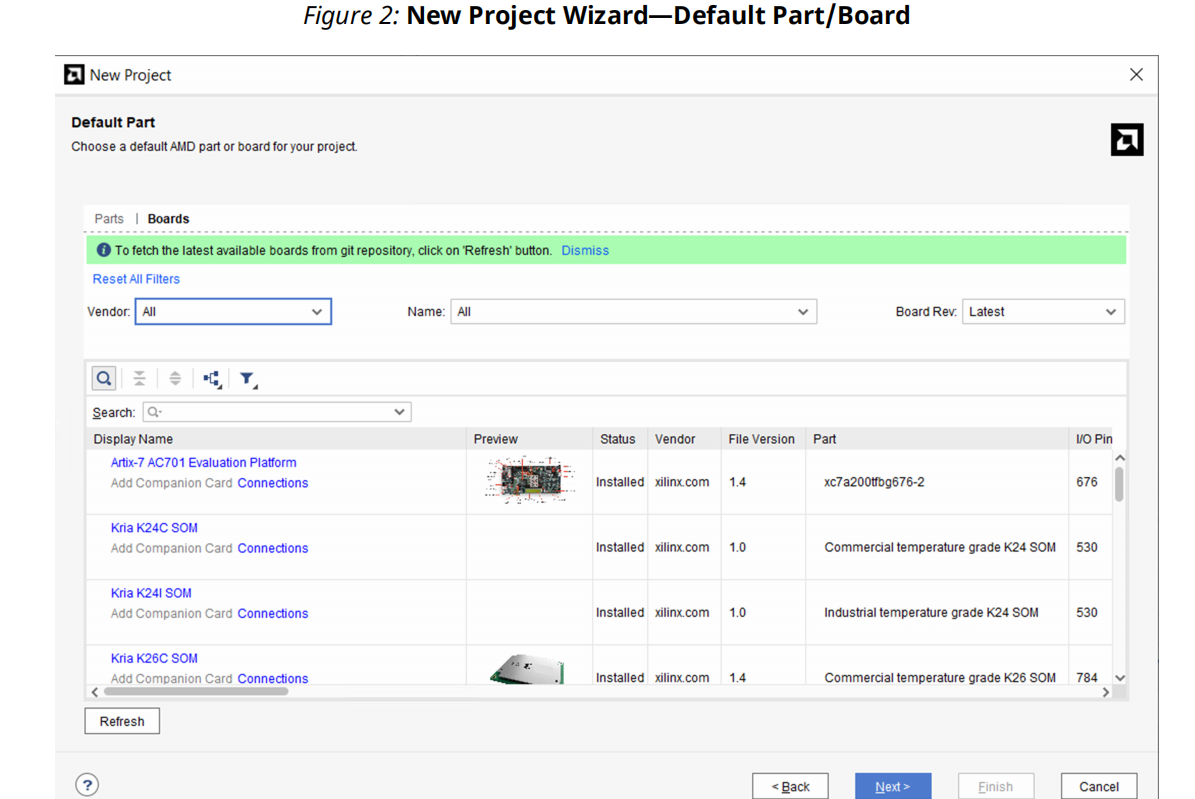
**2 Creating a Project**

VHDL 的一个强大功能是用库来组织 RTL 的不同部分。 通过使用库，不同的设计人员可以做这个工程中自己负责的那部分工作，而不必担心会在命名方面与其他设计师发生冲突。在VHDL中，名为“work”的库在 VHDL 中的用法比较特殊，实际上它指的是“当前库”，如果没有指定当前库的库名，编译器会创建“work”库。不同的项目若都使用“work”库名会产生混淆，因此不同项目需要指定独立的当前库名而避免使用“work”库名。

默认情况下，将 VHDL 文件输入 Vivado 工程时，该工具会将这些文件放入一个名为“xil\_defaultlib”的库中。xil\_defaultlib是当前库的库名，对应于VHDL语句中的“work”。使用“xil\_defaultlib”作为当前库的库名是为了区别其他（引入）项目时，可能会使用“work”作为当前库名，这样以示区别，避免混淆。因此，开发独立库时应使用对应的库名，不要使用“work”库名。

**3 Using the Vivado Design Suite Platform Board Flow**

The Vivado Design Suite lets you create projects using AMD target design platform boards (TDP), or user-specified boards that have been added to a board repository. When you select a specific board, the Vivado design tools show information about the board, and enable additional designer assistance as part of IP customization, and for IP integrator designs.



**3.1 Adding User-Boards to a Repository**

set\_param board.repoPaths [list "<path1>" “<path2>” “...”]

These paths can also be added in the GUI using **Tools → Settings → Vivado Store → Board**

**Repository**

For more information about the Vivado\_init.tcl file refer to the Scripting in Tcl section in

the Vivado Design Suite Tcl Command Reference Guide (UG835).

**3.2 Using the IP Catalog with the Platform Board Flow**