

## FPGA Solutions

### Using Synplify<sup>®</sup> Software Synthesis with Xilinx Platform Studio

by Milan Saini, Technical Marketing Manager, Xilinx, Inc.

**Embedded** processor-based platform FPGAs have placed new demands on tool providers. With hardware and software flows converging on to the same device, users are looking to retain their productivity through simplified, intuitive and interoperable tool environments. Xilinx Platform Studio (XPS) is a system design IDE that supports open interfaces making tool integration easy and painless. Xilinx embedded processor designs using Xilinx IP require XPS for platform assembly, configuration and implementation. Users may, however, choose to implement their custom IP blocks with the Synplify product. This article outlines a set of scenarios that allow users to easily integrate a block implemented using Synplify's Synplify product with the rest of the processor design.

#### Background

Xilinx supports two distinctly differentiated embedded processors in its devices. The first is the PowerPCTM 405 from IBM that is embedded in the Virtex-II Pro family of devices. This offers a high performance compute platform with processor clock frequencies of up to 400MHz and a DMIPS rating of up to 600. The second processor is MicroBlaze™, a 32-bit soft CPU from Xilinx based on a standard RISC architecture. This can be implemented in either a Virtex or a Spartan device, offering users with a more portable solution. The clock frequencies for Microblaze can approach 150 MHz with DMIPS ratings of around 125.

XPS provides a common front-end design entry for both processor systems. The most common scenario is for users to assemble their system by selecting from a portfolio of processor peripherals provided by Xilinx. The design is then synthesized and implemented using Xilinx tools.

In situations where users have created custom logic in an external synthesis environment like Synplify<sup>®</sup>, special steps are necessary to integrate their block with the rest of the processor system being assembled in XPS. The scenarios described below outline the way in which a user's custom peripheral can be synthesized through Synplify software and integrated into the XPS design. Please note that it is assumed that the user IP is designed as a block that will attach to the processor via one of the supported interfaces. Currently, the supported interfaces include the CoreConnect™ PLB, OPB, and DCR, and Xilinx proprietary FSL and OCM.

#### Scenario 1: Use XPS Import IP Wizard

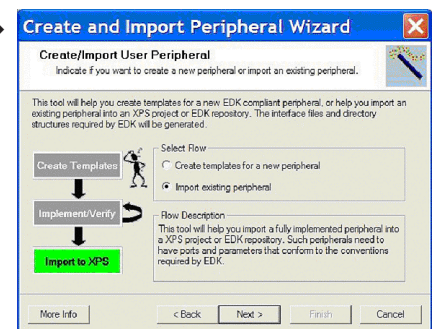
The simplest way to attach a user block to the XPS generated processor system is through the "Import IP Wizard" in the XPS GUI (Figure 1). The Wizard serves two functions:

- Allows users to easily convert their custom logic block into a CoreConnect compliant peripheral ("Tools → Import Peripheral Wizard → Create template for new peripheral").
- Allows users to import their existing CoreConnect IP into the XPS environment ("Tools → Import Peripheral Wizard → Import existing Peripheral").

In both situations, the Wizard helps set up the necessary files and directory structure for the user peripheral.

The following steps outline the flow:

- Create a custom peripheral and synthesize it using the Synplify software.
- From XPS, invoke the Import Peripheral Wizard in one of the following two modes:
  - Create Template:** In this mode, the Wizard will create the bus interfaces necessary to attach the user logic with the selected processor bus. The user logic then has to be manually stitched in with the interface block created by the Wizard. Once completed, the user block is now a CoreConnect peripheral. The user should invoke the wizard again, this time in the "Import Peripheral" mode.
  - Import Peripheral:** In this case, it is assumed that the user has created the necessary bus interfaces to attach to CoreConnect (user creates an HDL wrapper file that instantiates the custom logic represented by the edif netlist). The Wizard will import the block into the XPS environment.
- Follow the steps in the Wizard providing the necessary options and complete the flow.
- "Tools → Generate Netlist".



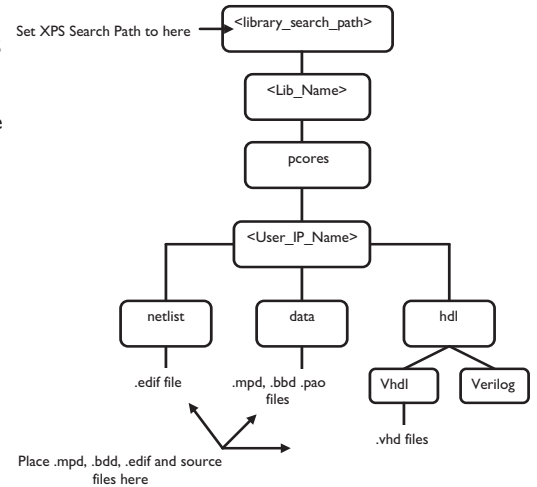
**Figure 1:** The Import IP Wizard in XPS can help attach a custom logic block to a CoreConnect System Bus based design

This creates a design netlist with the user IP component represented by the edif/wrapper file provided by the user and the Wizard, setting up all the necessary files and folders, as well as adding the necessary bus interfaces. An additional mode of the Wizard that can import a non-CoreConnect module is discussed in the “Attaching Glue Logic” section below.

**Scenario 2: Synthesize the User Block with Synplify Software in ISE ProjNav**

In this scenario, the user can synthesize the custom block in ISE by exporting the netlist from XPS to ISE ProjNav. The following steps outline the flow:

1. Create and place the peripheral .MPD file in the structure defined in Figure 2. The XPS file formats are defined in the Chap 14 of the Embedded System Tools Guide (EDK 6.2i). An example of the IP directory structure and file formats can be seen in the EDK SW installation directory under \$XILINX\_EDK/hw.
2. In the .MPD file, set option “Imp\_netlist=False”. This ensures that XPS will not try to synthesize this block using Xilinx tools.
3. In XPS:
  - a. Set peripheral search path to point to the location where the peripheral core is located (Figure 3). Once the .MPD file is in place and the search path is set, the peripheral will appear in the XPS “Add/Edit Cores” menu.
  - b. Attach the block to the rest of the design using “Add/Edit Cores...”
  - c. In the Project Options: Set “Synthesis=None”.
  - d. Select “Tools → Export to ProjNav” button.
4. In ISE ProjNav, set the synthesis tool to Synplify and implement the design.



**Figure 2:** The directory structure and files needed for processing custom user logic.

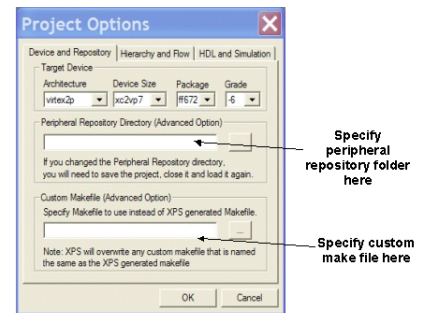
This creates a design netlist using Xilinx tools except for the user IP block where Synplify software is used as the synthesis engine. This flow is recommended if user IP has configurable parameters that can change over the course of a design.

**Scenario 3: Use of a Custom Makefile in XPS**

In this scenario, user can run a custom make file in XPS to invoke Synplify software on the custom IP block.

The following steps outline the flow:

1. Create and place the .MPD file in the structure defined in Figure 2.
2. In the .MPD file, set option “Imp\_netlist=False”.
3. In XPS:
  - a. Set peripheral search path to point to the location where the peripheral core is located (Figure 3).
  - b. Attach the block to the rest of the design using “Add/Edit Cores...”
  - c. Specify a custom makefile ‘mymake.make’ in the Project Options (see Figure 3).
  - d. Copy the content of XPS generated default ‘system.make’ into ‘mymake.make’.
  - e. Create a batch file (say ‘mysynthesis.sh’) that calls out your synthesis tool from the command line.
  - f. Replace references to the default ‘synthesis.sh’ script with ‘mysynthesis.sh’ in your ‘mymake.make’
  - g. “Tools Generate netlist”.



**Figure 3:** Setting project options in XPS

makefile.

Similar to Scenario 2, this creates a netlist of the entire design but specifically calls the Synplify tool to synthesize the user IP blocks. This flow is also recommended when user IP has configurable parameters that may change over the course of a design.

**Attaching Glue Logic to Processor Systems**

If the connection between user logic and the processor system is not through one of the supported interfaces but rather through an arbitrary set of signals, the Import IP Wizard can be used to create an XPS compatible IP block without any CoreConnect interface. The user block and all its ports are then visible in XPS and the user can manually attach these to any ports of the processor system. The following steps outline the flow:

---

## “Xilinx Platform Studio” continued from page 2

1. User creates a custom peripheral and synthesizes it using Synplify software. Write a wrapper HDL that directly instantiates the EDIF as a blackbox.
2. In XPS:
  - a. Invoke “Tools → Import Peripheral Wizard → Import User Peripheral”. Add the EDIF and the HDL wrapper as source files in this Wizard. This will create the necessary infrastructure for importing the block into XPS.
  - b. The block in this case will not have any implicit connections with the processor system. Manually connect its ports as applicable with the rest of the processor system.

A second option is to export the processor system as a 'sub module' (“Options → Project Options → Hierarchy and Flow”) from XPS to ISE ProjNav. This creates a top level wrapper module with the processor system instantiated in it as a sub module. The glue logic and the rest of the FPGA system can be created in the Synplicity® environment and connected with the processor block within this top level module.

### Conclusion

Customer peripherals for Xilinx processors can be synthesized with the users' choice of synthesis tool. The steps above outline ways in which users can synthesize with the Synplify solution and still integrate their IP with the rest of the processor system generated through XPS.



Simply Better Results

Synplicity, Inc.  
600 W. California Avenue  
Sunnyvale, CA 94086 USA  
Phone: (US) +1 408 215-6000  
Fax: (US) +1 408 222-0263

[www.synplicity.com](http://www.synplicity.com)

Copyright © 2004 Synplicity, Inc. All rights reserved. Synplicity, the Synplicity logo, “Simply Better Results,” Synplify, and Synplify Pro are registered trademarks. All other names mentioned herein are trademarks or registered trademarks of their respective companies.