

Summary

This Application Note address the design flow used to insert a PCI Target LogiCore into a VHDL design that is processed using FPGA Compiler. The flow using Design Compiler is similar. The PCI modules consist of a 32-bit target interface and a back-end interface unit (BIU). The designer can add logic to the BIU to customize it to their application.

Table of Contents

Table of Contents	1
Introduction	1
The FPGA Compiler Design Flow	2
New Design Flow	3
Using Design Compiler	3
Verification	3
Summary	3

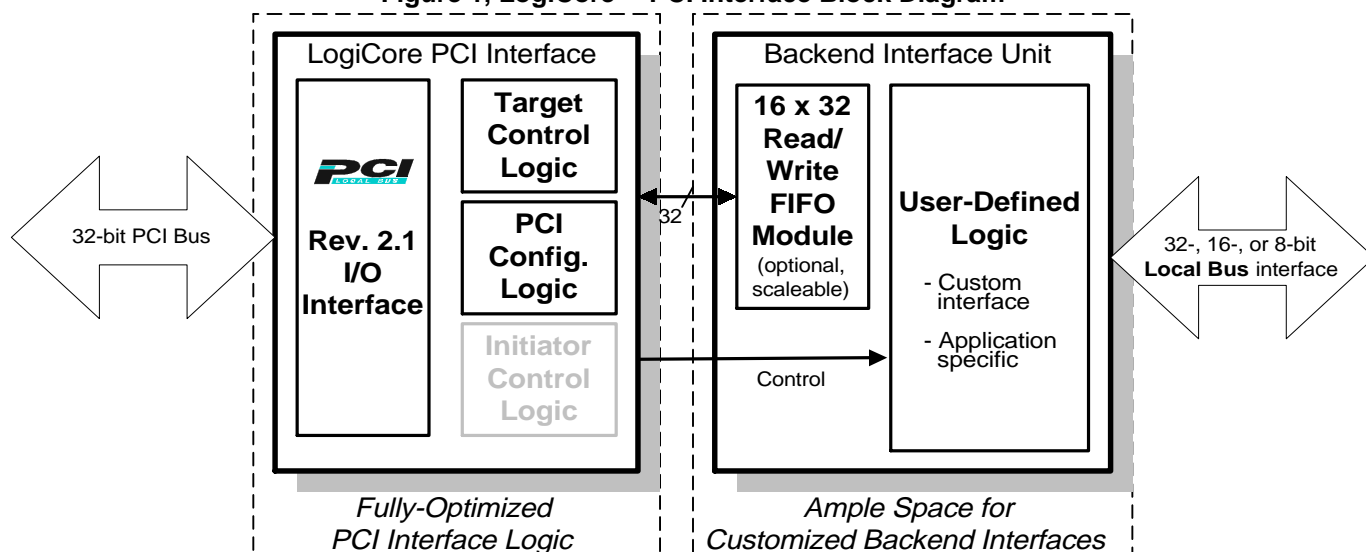
Introduction

One of the classic trade-offs in system design is that generic design using HDL may sacrifice performance compared to designs optimized for a particular target architecture. The problem is no one wants to make their design highly specific for a given architecture since that limits the possibilities for retargeting at a later time. This problem gets acute when considering the use of macro

cores or building-block elements. If the cores are too general purpose, they may be capable of being targeted to many architectures but performance may suffer. Xilinx has addressed this problem by introducing LogiCore™ Solutions, drop-in modules for FPGAs. These drop-in building blocks are unique because they are the only pre-implemented and fully verified drop-in modules in the FPGA industry and are pre-defined and optimized for a particular Xilinx FPGA family and/or device(s). Because they are highly optimized to the target architecture, the designer saves the time involved in learning, optimizing and verifying the design.

The LogiCore™ drop-in modules are ready to use with pre-determined timing, performance, and area. This approach leverages the experience of design experts and makes the results available to a large number of users. As a result, designers can dramatically cut development time, significantly reduce design risk, and have access to the best performance and lowest component cost available.

Figure 1, LogiCore™ PCI Interface Block Diagram



The FPGA Compiler Design Flow

The normal FPGA Compiler design flow using the Xilinx-supplied XSI libraries and interface is:

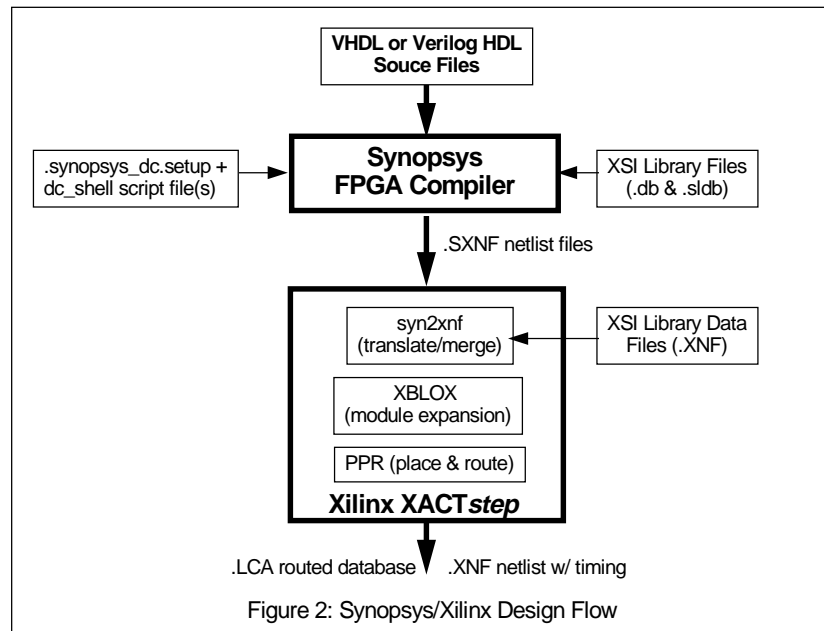
1. Analyze/elaborate source code (from lower levels upwards)
2. Set timing constraints
3. Define I/O ports as pads (optional)
4. Insert I/O pads (optional)
5. Compile
6. Run timing and area reports
7. Convert CLBs/IOBs to gates (output formatting)
8. Write XNF file (.SXNF format)
9. Place & Route with XACTstep tools

When using the PCI LogiCore there are several important differences in the design flow. Because the core is highly optimized, the place-and-route netlist contains many physical constraints (placement and timing) that guarantee the final performance. In order to take advantage of this we want this netlist to be used for physical implementation. We would also like the compiler to be able to time through the core and optimize the logic adjacent to it, however, so a mechanism is needed to achieve both of these goals. The principal changes to the design flow are:

1. LogiCore elements will be read in from post-route netlists to give FPGA Compiler access to both function and timing.
2. No optimization will be performed on the LogiCore elements.
3. LogiCore elements will be removed from the design database before writing the output netlist for physical implementation.
4. Physical implementation will use LogiCore design netlist with imbedded physical and timing constraints.

Using this design methodology, we can use the post-route, gate-level netlist of the LogiCore modules for functional simulation, timing estimation, and optimization of surrounding logic taking into account the function and timing of the logic at the boundaries of the LogiCore modules. After synthesis of the user-defined logic, use of the original LogiCore netlist with all of its placement and timing constraints ensures the guaranteed performance of the module will be met. Probably the most important point is remembering to remove the LogiCore modules from the design database. This will cause the output

Figure 2, Synopsys/Xilinx Design Flow



netlist to have “holes” in it where the modules go and the XACTstep tools will search for XNF design files with those names to merge in. If the modules are not removed from the design the output netlist will have the generic gate-level description read in earlier in the process and will not contain the valuable mapping, timing, and physical constraint information needed to obtain the desired performance.

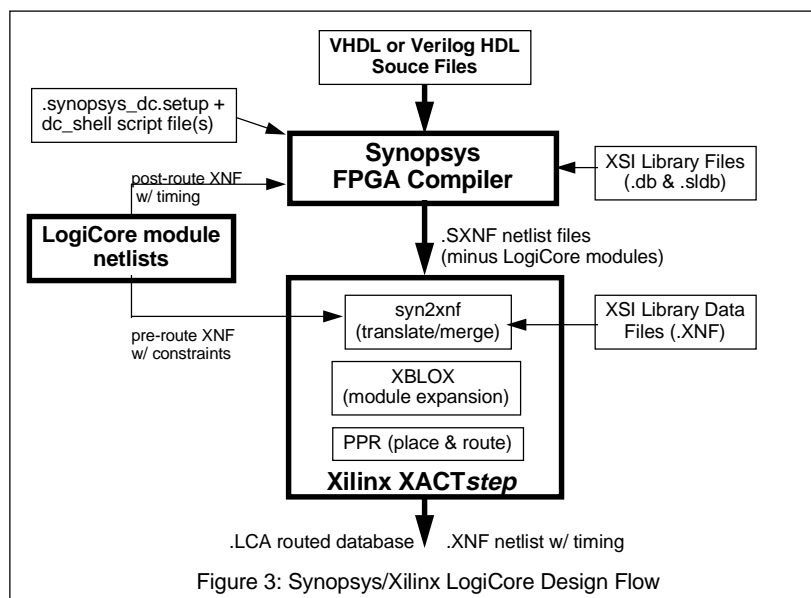
Recommended practice for Xilinx design in Synopsys is to handle global reset assignment and global clock buffering at the top level of the design. Therefore, the top level VHDL file includes an instantiated global clock buffer BUFGP for the system clock. The buffered clock signal (CK) is used to drive the PCI bus interface (CTRL-32T) and the back-end interface unit (BIU). This buffered clock should also be used for any user-defined logic.

There is also an instantiated STARTUP symbol at the top level that defines which signal will be connected to the system global set/reset net. If the user-defined VHDL code includes reset behavior, it should use this master reset signal (RST_N) if possible. The XACT tools will remove redundant connections to individual set or reset pins on registers since this signal is also connected to the GSR pin of the STARTUP symbol.

Figure 3, Synopsys/Xilinx LogiCore Design Flow

New Design Flow

1. Read in LogiCore controller module and back-end interface in netlist form (post-route w/ timing) using XNF reader
2. Analyze/elaborate balance of source code (including user logic connected to back-end interface)
3. set_dont_touch on LogiCore elements
4. set timing constraints
5. set_port_is_pad for any user I/O pads in back-end logic (LogiCore elements contain I/O pads already so no pad insertion is necessary for those modules).
6. insert_pads (if using auto pad insertion for back-end logic)
7. Compile
8. Run timing and area reports
9. Convert CLBs/IOBs to gates (output formatting)
10. remove_design for all LogiCore elements
11. Write XNF file
12. Place & Route with XACTstep tools (LogiCore design files will be merged in automatically from the working directory)



Verification

As mentioned previously, the post-route XNF netlist of the LogiCore modules read into FPGA Compiler can also be used for functional simulation since it consists entirely of post-route simulation primitives found in both the VSS simulation library and the Xilinx Verilog simulation libraries.

After the design is processed with the XACTstep tools it can be simulated with timing using the same back-annotation interfaces as any conventional Synopsys design. Back annotation interfaces for both Synopsys' VHDL System Simulator (VSS) and Verilog XL are available from Xilinx. Although this example used VHDL, the entire process can be reproduced using Verilog and the design flow is exactly the same.

Summary

This paper has focused on a design methodology that allows the use of pre-defined LogiCore drop-in modules to be used in combination with user-defined and synthesized logic when using FPGA Compiler to target a Xilinx FPGA device. This combination provides a valuable combination of guaranteed high performance from the drop-in modules and the flexibility and retargetability of HDL driven synthesis.

North America

Corporate Headquarters

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
U.S.A.

Tel: 1+(408) 559-7778
FAX: 1+(408) 559-7114

Northern California

Xilinx, Inc.
3235 Kifer Road
Suite 320
Santa Clara, CA 95051

Tel: (408) 245-1361
FAX: (408) 245-0517

Southern California

Xilinx, Inc.
15615 Alton Parkway
Suite 280
Irvine, CA 92718

Tel: (714) 727-0780
FAX: (714) 727-3128

New Hampshire

Xilinx, Inc.
61 Spit Brook Road
Nashua, NH 03060

Tel: (603) 891-1096
FAX: (603) 891-0890

Pennsylvania

Xilinx, Inc.
65 Valley Stream Parkway
Suite 140
Malvern, PA 19355

Tel: (215) 296-8302
FAX: (215) 296-8378

Texas

Xilinx, Inc.
4100 McEwen
Suite 237
Dallas, TX 75244

Tel: (214) 960-1043
FAX: (214) 960-0927

Illinois

Xilinx, Inc.
939 N. Plum Grove Road
Suite H
Schaumburg, IL 60173

Tel: (708) 605-1972
FAX: (708) 605-1976

North Carolina

Xilinx, Inc.
5952 Six Forks Road
Raleigh, NC 27609

Tel: (919) 846-3922
FAX: (919) 846-8316

Europe

United Kingdom

Xilinx, Ltd.
Suite 1B, Cobb House
Oyster Lane, Byfleet
Surry KT14 7DU
UNITED KINGDOM

Tel: (44) 932-349401
FAX: (44) 932-349499

France

Xilinx Sarl
Espace Jouy Technology
21, rue Albert Calmette, Bt. C
78353 Jouy en Josas Cedex
FRANCE

Tel: (33) 1-34-63-01-01
FAX: (33) 1-34-63-01-09

Germany

Xilinx, GmbH
Dorfstr. 1
85609 Aschheim
München
GERMANY

Tel: (49) 89-904-5024
FAX: (49) 89-904-4748

Japan

Xilinx, K.K.
Daini-Nagaoka Bldg. 2F
2-8-5, Hatchobori Chuo-ku
Tokyo 104
JAPAN

Tel: (03) 3297-9191
FAX: (03) 3297-9189

Asia Pacific

Hong Kong

Xilinx Asia Pacific
Unit No. 2318-2309
Tower 1, Metroplaza
Hing Fong Road
Kwai Fong, N.T.
HONG KONG

Tel: (852) 410-2717
FAX: (852) 418-1600
E-mail: hongkong@xilinx.com



The Programmable Logic CompanySM