APPLICATION NOTE

Mentor Synthesis Group

Precision Synthesis

Xilinx LogiCORE PCI Core Flow

Mike Fingeroff – Technical Marketing Engineer

Mentor Graphics®

Overview

This appnote describes the procedure for implementing the Xilinx PCI core using Precision Synthesis from Mentor Graphics. This design flow has been fully automated to provide a push-button flow for both synthesis and place and route. Precision Synthesis will automatically recognize the unique files associated with the Xilinx PCI core and configure the necessary synthesis and place and route settings. This automation dramatically reduces the complexity of this flow, including place and route, to the 4 simple steps described on page 1 of this appnote.

Procedure

1) Add all PCI core NGO, UCF, and NCD files along with all core and user HDL files.

Note: *Rather than having to manage two separate projects for both synthesis and P&R, Precision allows the user to include all Xilinx PCI core files in their Precision project.*

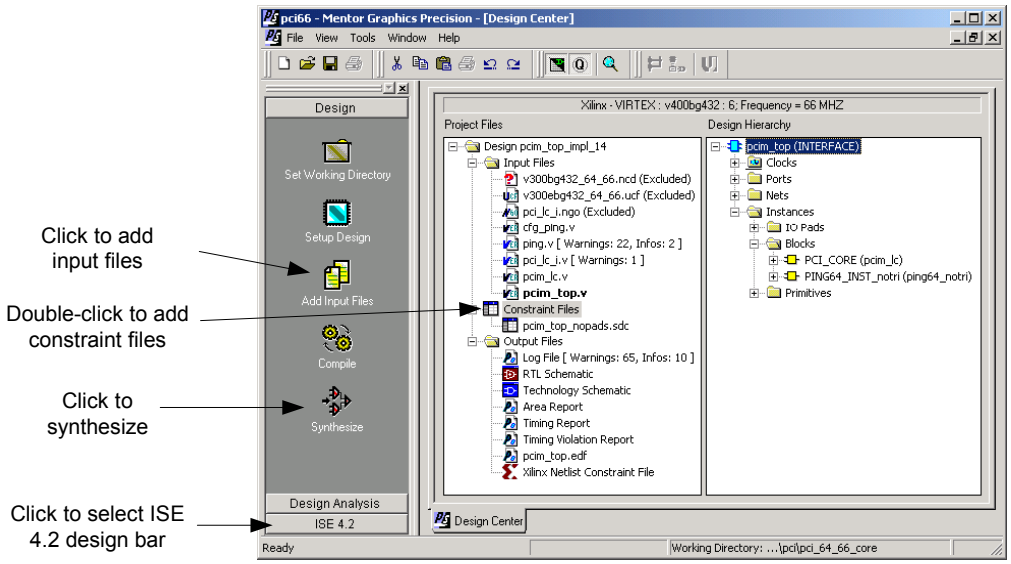

Figure 1 Precision Synthesis PCI Core Project

2) Copy the constraints listed in appendix A and save them to a file with a *.sdc extension. Then add the constraint file to the Precision project. These constraints are required to prevent buffer insertion on the PCI core signals, which have instantiated PCI I/O buffers. The designer may also apply additional constraints to their custom logic at this point in the flow.

3) Click the synthesize button on the Design bar

4) Click the place and route button in the ISE 4.2 design bar.

Note: *Precision automatically sets the required system environment variables, as well as the ngdbuild, map, and par switches. The Precision transcript window can be examined to see how these variables and switches are set.*

Done!

APPENDIX A PCI Core Constraints

```
set_attribute -name nopad -value TRUE -port AD*
set_attribute -name nopad -value TRUE -port CBE*
set_attribute -name nopad -value TRUE -port IDSEL
set_attribute -name nopad -value TRUE -port INTR_A
set_attribute -name nopad -value true -port GNT_N
set_attribute -name nopad -value true -port FRAME_N
set_attribute -name nopad -value true -port DEVSEL_N
set_attribute -name nopad -value true -port IRDY_N
set_attribute -name nopad -value true -port TRDY_N
set_attribute -name nopad -value true -port PERR_N
set_attribute -name nopad -value true -port SERR_N
set_attribute -name nopad -value true -port REQ_N
set_attribute -name nopad -value true -port STOP_N
set_attribute -name nopad -value true -port PAR
set_attribute -name nopad -value true -port RST_N
set_attribute -name nopad -value true -port PCLK

#Uncomment the following lines for the 64-bit/66MHz
Core
#set_attribute -name nopad -value true -port ACK64_N
#set_attribute -name nopad -value true -port PAR64
#set_attribute -name nopad -value true -port REQ64_N

#Uncomment the following lines for the 64-bit/100MHz
XPCI Core
#set_attribute -name nopad -value true -port PMEA_N
```