

Project Management for Timing Analysis

Introduction

Engineering today's electronic products presents challenges in achieving specified design speeds, low power consumption, and size reduction. Design techniques are changing to address these challenges and incorporating signal integrity, packaging effects and device layout characteristics has become a necessity. All of these factors have significant impact on timing margins across a design project. In order to achieve a working design, it's critical to have an accurate understanding of timing margins and with timing margins being affected with increased frequency as projects progress, miscommunication among the engineering team members is growing.

To accommodate growing design size and complexity, it is essential to have a methodology for managing timing through the design process and across engineering teams. TimingDesigner offers a way to simplify the exchange of critical timing information among team members and allows users to better manage the specification and timing analysis of high performance interfaces for digital IC and board designs. Timing diagram descriptions are logically organized into projects of single or multi-function components. They are arranged, displayed, and manipulated in an easy-to-read tree structure. Timing diagrams representing various protocol operations are associated with each respective component to accommodate easy navigation for editing and analysis. Combined timing margin violation reports are also included to simplify and streamline the monitoring and management of critical timing requirements. .

This paper takes you through a sample design with TimingDesigner, highlighting the steps you would take to define and organize the critical timing components of your design. It also illustrates how TimingDesigner can simplify the complex process of analyzing, monitoring, and managing timing through the design process. These same techniques can be applied to a wide range of similarly complex designs and applications.

Timing Components Mimic Real Devices and Modules

To determine successful communication between interfaces that connect embedded functions on an ASIC or a programmable IC or between devices on a circuit board, requires accurate timing models that exercise the correct interface protocols. With TimingDesigner, you manage complex timing specifications by aligning your timing plan in the same way you organize your design. Timing models are organized into functional components complete with a port list for component pin descriptions and a netlist to establish component connectivity and to determine what signals control interface protocol operations.

Figure 1 shows how each component of a design project is arranged and displayed in an expandable single tree structure. A component may be an individual module in an ASIC design or a FPGA device, or an IC device on a printed circuit board. Component blocks define a device's modularity for multi-function components and timing diagrams are automatically associated with each component block.

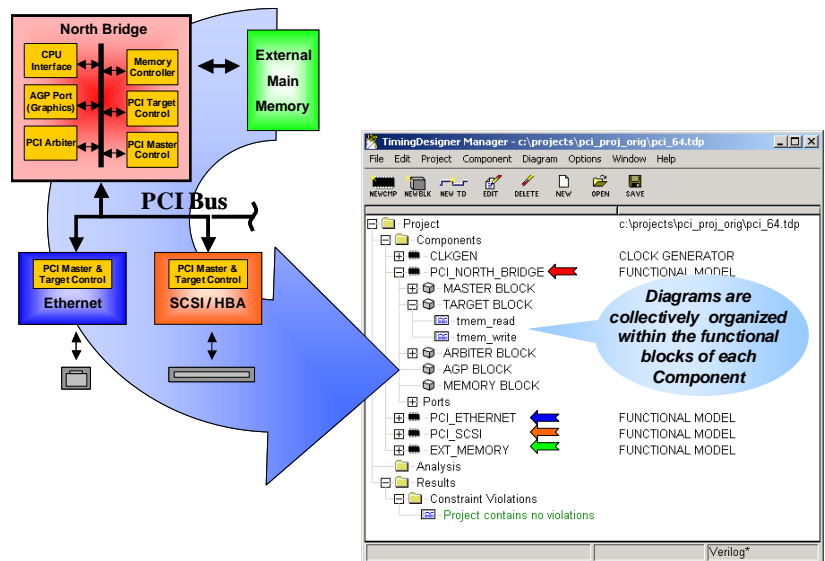


Figure 1 - Project organization within the TimingDesigner Manager Window

Defining component ports and a project netlist

Each component has a defined port list. This is analogous to a pin list for IC devices and signal inputs and outputs for functional design modules. The port list for each component is displayed in the Manager Window or in its own dialog box for editing. It contains port names and their associated direction (input, output, in/out, internal) and adheres to HDL syntax rules.

Checking project timing for errors

Now, checking any or all project timing diagrams for timing errors at any point within the design development is a simple process. Diagrams present in the project are parsed and a report is generated with violated constraint events. If all diagrams are error-free, an easy-to-see green “Project contains no violations” indicator is displayed.

For timing margin violations, a list including the violated constraint name, the component, block, and diagram name is created. A simple double-click on the listed constraint name will bring up the associated diagram with the constraint violation highlighted for easy identification and editing.

Setting up a PCI Bus Design Project

The PCI bus design shown in Figure 2 is an example used to illustrate how TimingDesigner can help manage the complexity of monitoring and analyzing timing issues throughout your design project. This block diagram of a 64-bit PCI bus design has North Bridge, Ethernet, SCSI interface, and South Bridge components.

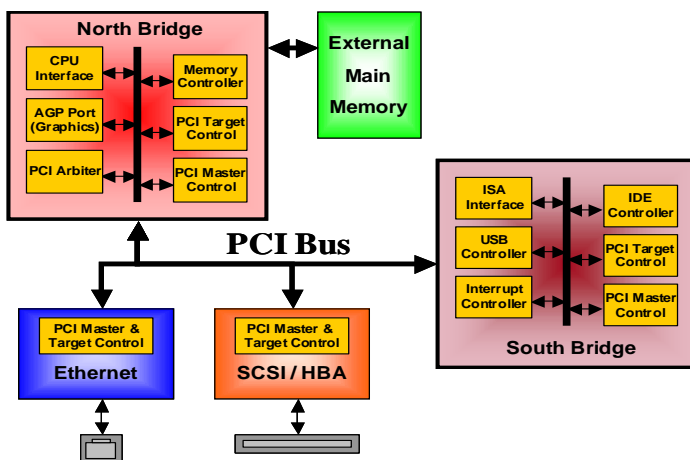


Figure 2 - Block diagram for PCI Bus design

The two bridges are multi-function components that may or may not have direct compatibility with the PCI bus interface. An external memory device also interfaces directly with the North Bridge.

The four design components in Figure 2, two single function and two multi-function, are shown in the TimingDesigner Manager Window exactly how they appear in the block diagram (see Figure 3).

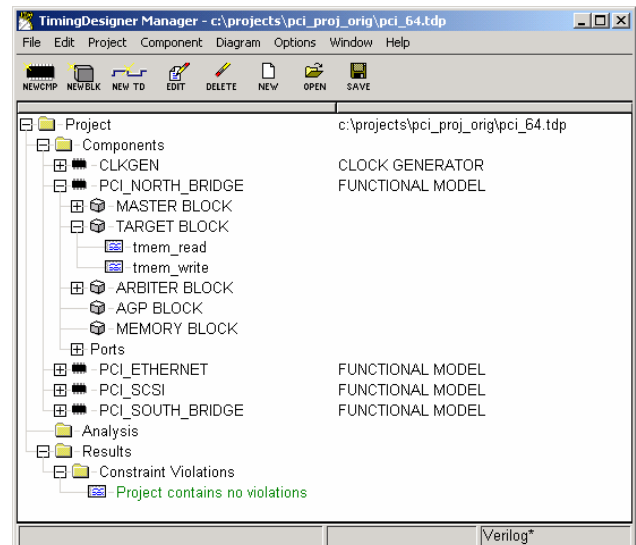


Figure 3 – PCI Bus design in the Manager Window

Note that each component in Figure 2 is a project component within TimingDesigner and the functional modules within the North Bridge module are component blocks.

Components are Functional Models or Clock Generators. Clock Generators are special component functions that serve to clock a design, while Functional Models are components that serve a non-clocking function.

Creating and Organizing Timing Diagrams

Timing diagrams define how to apply specific interface operations that are subject to timing analysis. Creating timing diagrams that include the key elements of your design is a straightforward process. When you have multiple components with similar features, TimingDesigner offers many time saving features. Within the Manager Window, you can easily create a new timing diagram for a component by selecting similar diagram associated with the component and copying it to use as a template. TimingDesigner automatically adds all signals in the port list for each new diagram, shortening the creation process. You can also add diagrams to a component block by using the save feature on an open diagram within the Diagram Window. This allows you to easily copy an existing diagram from another component or anywhere else on your network, into the selected component block.

For each high-level component of the design, you'll find diagrams that are logically organized within the component blocks. As seen in Figure 4, if the Ethernet component is to communicate across the PCI bus to access the external memory component, you would have a PCI memory read command diagram in the master component block of the Ethernet component, and a target memory read command diagram in the corresponding block of the PCI North Bridge since that device is responsible for communication with the external memory device.

current project's directory structure and displayed in the Manager Window tree.

Components can be created from an existing diagram by simply opening the diagram within the Diagram Window, then switching back to the Manager Window, and selecting "Create Component from Diagram" in the New Component Wizard. The resulting diagram assumes the listed signal names and the associated ports with port list filled accordingly. The open diagram is now a diagram under the newly created component block.

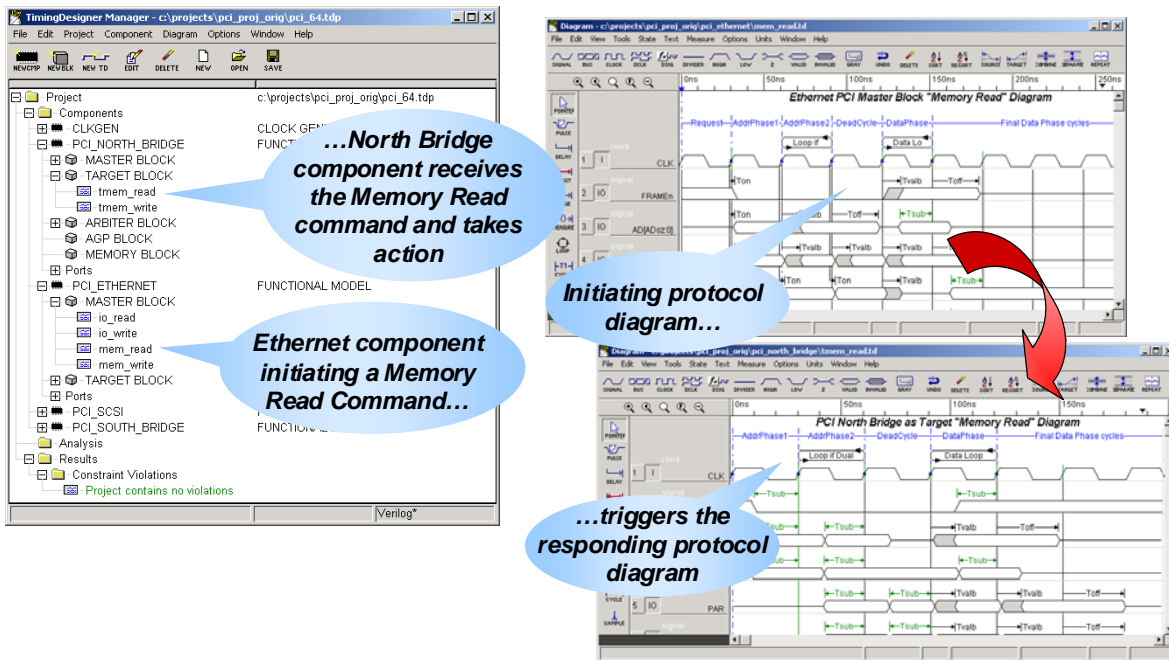


Figure 4 - Manager Window provides coordinating base for timing analysis

Options for Creating Components

Another time saving feature is to create components by copying existing ones from the current project or other unrelated projects. This is a quick-and-easy process using TimingDesigner's Component Wizard. The Component Wizard allows you to browse your machine or the network for the component you need. You create a new component by copying the ports list and all of the associated diagrams of an existing component, maintaining all structural information. You have the option to reverse the port directions of the new component allowing you to create an "initiator/target" component relationship where the like named ports are driving from the initiator and driven to the targets. You'll find all associated files and diagrams for the imported component placed in the

Editing the Netlist

When creating port lists for components, TimingDesigner will assign a net to each port in the list that initially has the same name as the respective port. This allows automatic generation of a netlist, intuitively assembled from the port list information. Editing capabilities are also provided to support any user-directed changes in the project connectivity.

Looking at the PCI project connectivity for bus arbitration, each component on the PCI bus will need to request control of the bus to execute commands. This arbitration process requires unique connectivity to identify and distinguish each bus request. For this design, the North Bridge component has an Arbiter module and diagram to handle the bus requests. Each component that contains a "PCI Master" function module will request the bus at one time or another.

From the Netlist editor, shown in Figure 5, you see the project connectivity where each component has its own individual bus request/grant pairs. The left column lists the ports and the right column lists the associated net name. By selecting a port, you can easily change its net assignment. Netlist names are case sensitive, and are not character restricted.

Netlist connectivity is not necessary when employing timing analysis for simple project designs. TimingDesigner uses the netlist connectivity, along with the port list information when automatically merging two diagrams together for analysis. This information is then used to determine which signals each diagram controls during the merge so that the relative signal placement and constraints are maintained.

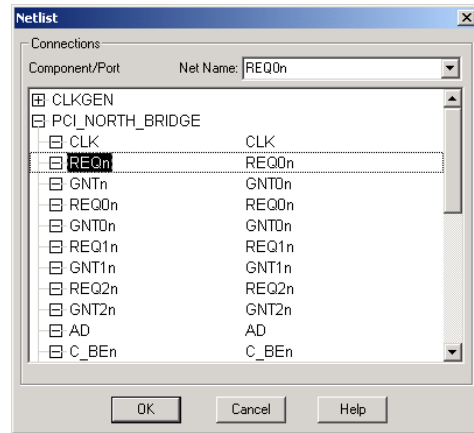
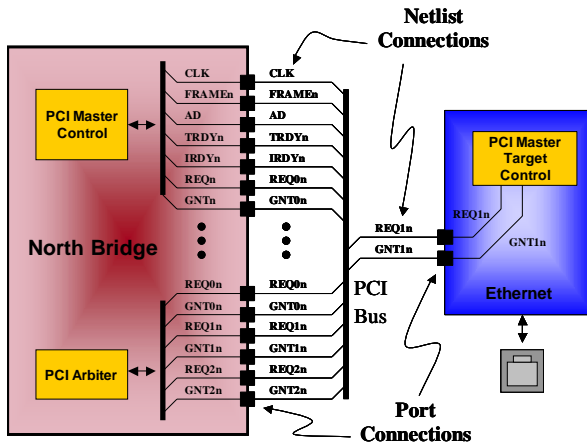


Figure 5 - PCI Ports and their associated netlist connections as shown in the Netlist Editor

In the Netlist editor, you assign the North Bridge component to use the REQ/GNT 0 pair and the PCI_Ethernet component to use the REQ/GNT 1 pair by simply tying the associated ports for each component to the respective net name in the right column. The REQn port for the master client now has the same net names as the REQ0 port for the arbiter, meaning these two ports are now connected. The same procedure is used for the Ethernet component, but assigning the signal pairs to port 1 of the arbiter.

Conclusion

With larger and more complex designs, monitoring and managing the analysis of critical timing objectives across project teams is not a trivial task. With the built-in project management features that TimingDesigner offers, you have a logical way to organize multiple timing diagrams associated with specific devices or functional blocks of a design and a simple way to exchange timing margin information with team members at any stage in the design process.

For more information about TimingDesigner, visit www.TimingDesigner.com.