XILINX®

XAPP411 (v1.1) October 23, 2001

PrimeTime Interface for Xilinx/Synopsys

Author: Steve Elzinga

Summary

This application note describes the interface, setup, and known issues necessary to use PrimeTime with the Xilinx implementation tools. The interface requires running scripts that translate relevant Xilinx files to a format that us useable by PrimeTime. This application note is not intended as a comprehensive explanation of Synopsys' PrimeTime tool. For more information regarding PrimeTime, please see the documentation supplied with that tool.

Introduction

The Xilinx static timing analysis (STA) tool called Timing Analyzer (TRCE for command line) analyzes timing constraints in a placed and routed design. With the release of the Xilinx 4.1i software, Xilinx supports timing analysis of a placed and routed design using a popular STA tool from Synopsys called PrimeTime. PrimeTime has the ability to read in Verilog and Standard Delay Format (SDF), as well as other file formats. The Xilinx/PrimeTime interface is intended for users who are already familiar with using PrimeTime as a tool for debugging timing issues in their designs. Since PrimeTime has not yet been certified by Xilinx as a sign-off timing analysis tool, you must validate the final timing results that are reported by PrimeTime with the Xilinx Timing Analyzer. Xilinx recommends that users who are not familiar with PrimeTime use Timing Analyzer to debug timing issues. Figure 1 shows where PrimeTime fits in the Xilinx design flow.

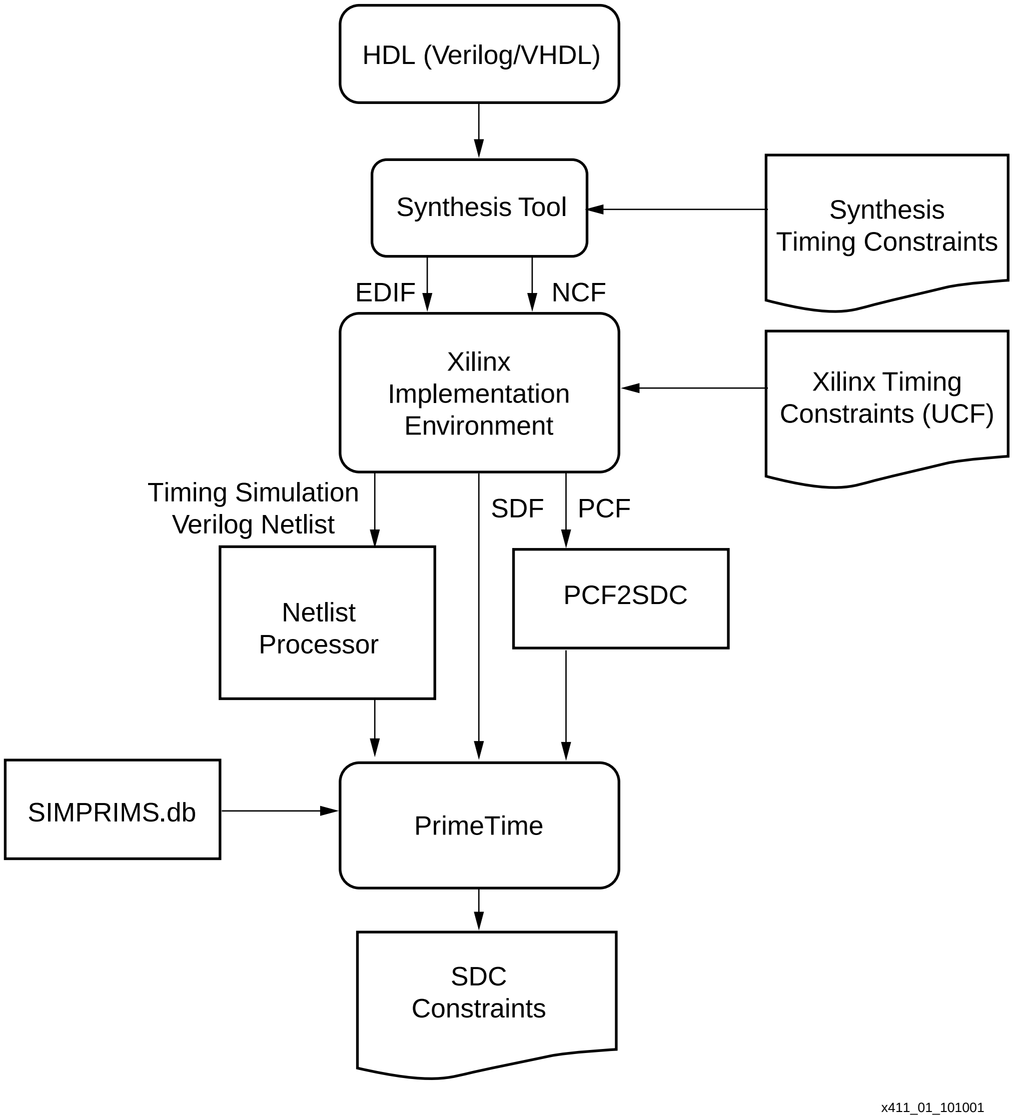

HDL (Verilog/VHDL)
Synthesis Tool
Synthesis Timing Constraints
EDIF
NCF
Xilinx Implementation Environment
Xilinx Timing Constraints (UCF)
Timing Simulation Verilog Netlist
SDF
PCF
Netlist Processor
PCF2SDC
SIMPRIMS.db
PrimeTime
SDC Constraints
x411_01_101001

Figure 1: **PrimeTime / Xilinx Design Flow**

Software

You will need the following software to use the interface between Xilinx and PrimeTime.

- Xilinx Foundation ISE Unix version 4.1i or greater.
- Synopsys PrimeTime version 1999.05 or greater.
- PERL version 5 or greater.

Required Files

You will need the following design files for PrimeTime.

- The timing simulation Verilog file.
- The Standard Delay Format file (.sdf).
- A Synopsys Design Constraints file (.sdc).

The timing simulation and standard delay format (.sdf) files are produced by Xilinx. The Synopsys design constraint file will either be made manually or generated from the pcf2sdc script.

Xilinx Files Supplied with Installation

The following files are provided in the $XILINX/synopsys/libraries/primetime directory with the Xilinx software installation on the Unix platform.

filter.pl

Definition: This is a PERL script that removes simulation only parts of the Verilog code from the back-annotated Verilog file produced by Xilinx.

Usage: perl5 filter.pl <input_file>.v <output_file>.v

pcf2sdc.pl

Definition: This is a PERL script that translates the global constraints PERIOD, OFFSETs, and FROM:TO for pads to pads, from PCF syntax to Synopsys Design Constraints (SDC) syntax.

Usage: perl5 pcf2sdc.pl <input_file>.pcf <output_file>.sdc

simprims.db

Definition: This is the library file that defines all of the primitives that will be in the back annotated Verilog file.

Usage: set link_library <path>/simprims.db

Xilinx Files Available for Download

The following files are available for download from ftp://ftp.xilinx.com/pub/applications/xapp/xapp411.tar.gz:

get_segment_delay.tcl

Definition: This is a TCL script that gets the delay value of a particular segment of the design.

Usage: Use this script in conjunction with one of the PrimeTime scripts described below.

one_dll_template.pt

Definition: This is a PrimeTime script template that calculates the delay of a DLL network in a similar manner as TRCE. This script calls the get_segment_delay.tcl file.

Usage: You can execute this script from the main project script or modify it, and use it as the main project script. You must run the template for every DLL network in the design. Also, you must modify the template so that the names associated with the DLL in the script match the names associated with the DLL in the design.

two_dll_template.pt

Definition: This is a PrimeTime script template that calculates the delay of a DLL network containing two DLLs in a manner similar to TRCE. This script calls the get_segment_delay.tcl file.

Usage: You can execute this script from the main project script or modify it, and use it as the main project script. You must run the template for every DLL network in the design. Also, you must modify the template so that the names associated with the DLL in the script match the names associated with the DLL in the design.

Best Constraining Methods

Initially, the most common global constraints will be translated from Physical Constraints File (PCF) format into Synopsys Design Constraints (SDC) format. The PCF file is generated from the constraints in the User Constraints File (UCF). To help in creating a UCF that will be easily read by the pcf2sdc translator, please use the following guide:

Period Constraint

- Single period constraint format
  - UCF Syntax

    NET "clock_in" TNM_NET = "clock_in";

    TIMESPEC "TS_clock_in" = PERIOD "clock_in" 10 ns HIGH 50%;
  - PCF Syntax

    TS_clock_in = PERIOD TIMEGRP "clock_in" 10 ns HIGH 50.000 % ;
- Duty Cycle
  - UCF Syntax

    NET "clock_in" TNM_NET = "clock_in";

    TIMESPEC "TS_clock_in" = PERIOD "clock_in" 10 ns HIGH 40%;

    Or

    TIMESPEC "TS_clock_in" = PERIOD "clock_in" 10 ns LOW 40%;
  - PCF Syntax

    TS_clock_in = PERIOD TIMEGRP "clock_in" 10 ns HIGH 40.000 % ;

Offset Constraint

- Global Offset constraints
  - UCF Syntax

    OFFSET = IN 5 ns BEFORE "clock_in";

    OFFSET = OUT 5 ns AFTER "clock_in";
  - PCF Syntax

    OFFSET = IN 12 ns BEFORE COMP "clock_in" ;

    OFFSET = OUT 12 ns AFTER COMP "clock_in" ;
- Net Offset Constraints
  - UCF Syntax

    NET net_name OFFSET = IN 5 ns BEFORE "clock_in";

    NET net_name OFFSET = OUT 5 ns AFTER "clock_in";
  - PCF Syntax

    COMP "net_name" OFFSET = IN 5 ns BEFORE COMP "clock_in" ;

  COMP "net_name" OFFSET = OUT 5 ns AFTER COMP "clock_in" ;

Maxdelay

- UCF Syntax

  TIMESPEC TS_P2P = FROM PADS TO PADS 10 ns;
- PCF Syntax

  TS_P2P = MAXDELAY FROM TIMEGRP "PADS" TO TIMEGRP "PADS" 10 ns ;

Special Considerations

There are some components and situations in the Xilinx flow that warrant some special considerations when doing a timing analysis with PrimeTime.

Related Clocks

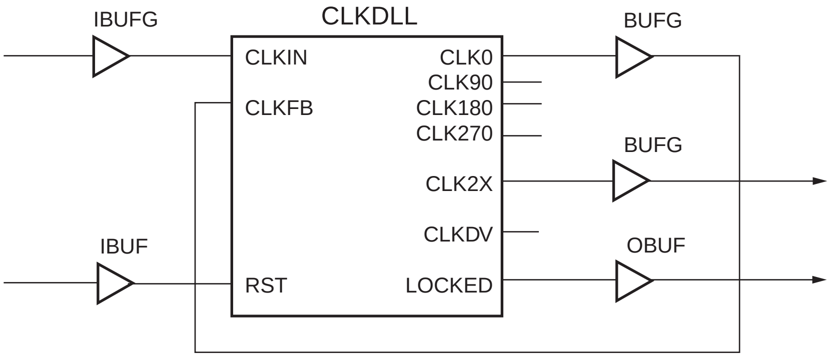
In the Xilinx 4.1i software, paths between clocks are covered if the two clocks are defined to be related. Clocks are related by using the following constraint:

```
TS_clk1 = PERIOD TIMEGRP "clk1" 30 ns HIGH 50.00%;
TS_clk2 = PERIOD TIMEGRP "clk2" TS_clk1 * 2;
```

The period of clk2 is defined to be twice that of clk1, therefore clk2 has a period of 60 ns. Constraining the clocks in this manner will cause them to be related. TRCE will then report on all paths between clk1 and clk2. However, the pcf2sdc translator does not translate period constraints defined in terms of another period constraint. Any clocks constrained in this manner will have to be hand entered into the sdc file.

CLKDLL

The CLKDLL is a component that was first introduced in the Virtex architecture. It can be used to minimize the clock skew in the device. Some special processing must be performed whenever there is a CLKDLL in the design. Run the get_segment_delay.tcl script along with the DLL PrimeTime scripts to mimic how TRCE reports delays when a CLKDLL is used. Figure 2 shows the inputs and outputs of a CLKDLL.



x411_02_101001

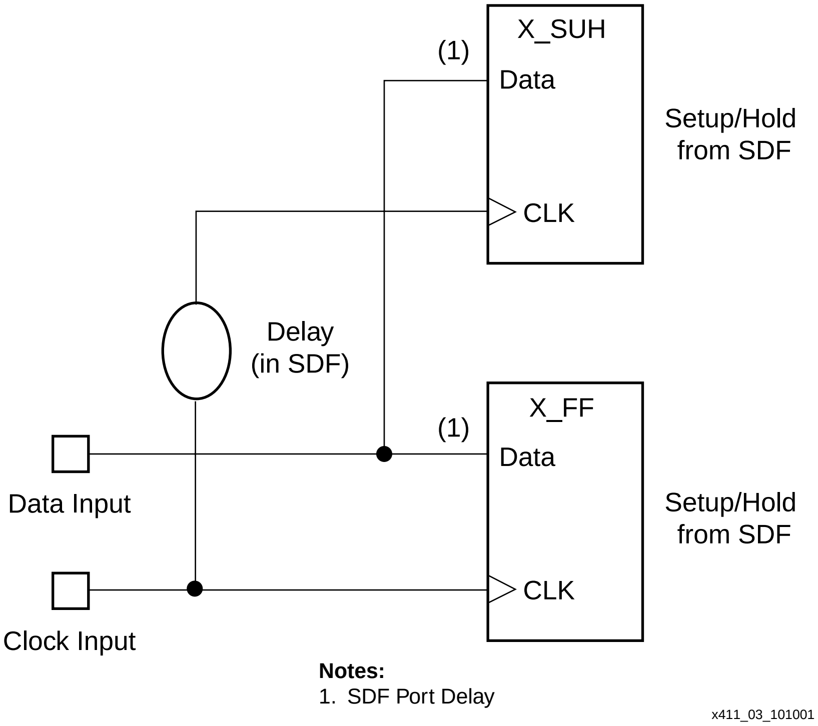
Figure 2: **CLKDLL**

X_SUH

The X_SUH is an element that is inserted by the Xilinx software whenever timing information for IOB flip-flops is needed. The X_SUH component has the clock setup and hold time information for both positive and negative edge flip-flops. Normally, an X_FF element complete with its own timing values is used for a flip-flop. However, since the IOB flip-flops have guaranteed setup and hold times, the X_SUH component was created to hold the timing values for these flip-flops. TRCE automatically turns off the X_FF element whenever there is an X_SUH element so that conflicting timing values are not reported. Furthermore, there are default timing values associated with the X_SUH in the simprims.db library. If the X_SUH has no timing values then the default values found in the library will be used. PrimeTime will report on all elements that have associated values in the library. If IOB flip-flops are used, PrimeTime will report on both the X_FF and the X_SUH components. Therefore, timing reports from PrimeTime should be carefully analyzed when setup and hold time values are reported on the X_FF when the X_SUH is used.

Even though only one clock edge will be used, PrimeTime will also report on both positive and negative clock edges of the X_SUH, causing a discrepancy between the TRCE and PrimeTime reports. This discrepancy could include errors or warnings that should be carefully analyzed.

Figure 3 represents the inputs and outputs of an X_SUH.



Notes:
1. SDF Port Delay

x411_03_101001

Figure 3: **X_SUH**

Positive Clock Skew

A positive clock skew will occur on register to register paths where the clock delay to the destination register (D2) is greater than the clock delay to the source register (D1) (Figure 4). A positive clock skew effectively lengthens the available clock period in which a signal has to propagate. For proper worst case calculations, however, the minimum value of the positive clock skew must be used, rather than the maximum. In order to be most conservative, Xilinx currently assumes a minimum value of zero for the positive clock skew. TRCE automatically sets the minimum value to zero during its calculations. This leads to an inconsistency with PrimeTime, since the maximum delay is passed to PrimeTime through the back-annotated netlist, and PrimeTime does not automatically make the same calculation. This will be fixed in a future release. Negative clock skew, which relies on the subtraction of the maximum delay, is handled correctly by both tools, and should not result in any discrepancies.

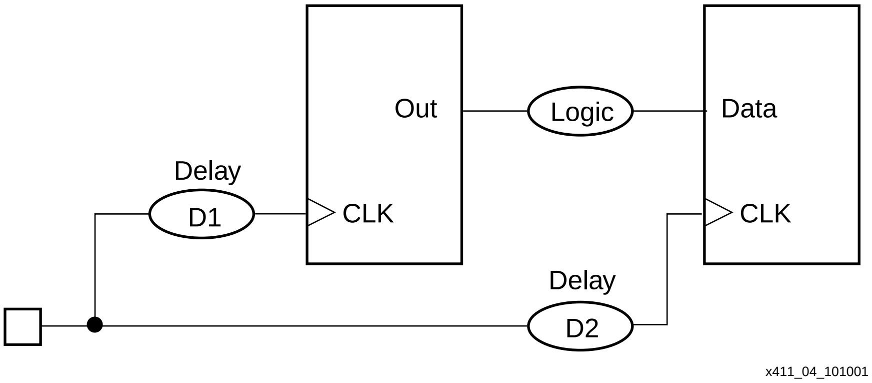


Figure 4: **Positive Clock Skew**

BUFGMUX

The global buffer primitive for the Virtex-II architecture is called the BUFGMUX. If a BUFGMUX is used in your design, then you must make the following modifications.

- For simulation purposes, pull-down components (X_PD) are inserted on the clock network in the Verilog timing netlist. These components will need to be commented out of the Verilog netlist before PrimeTime can process it. If you do not remove the X_PD, PrimeTime will view the clock path as unconstrained. If the BUFGMUX is used, the X_PD component will be attached to the clock network in a manner similar to the following:

```
X_PD <instance_name> ( .O (<clock_name>);
```

- For simulation purposes, there are latches associated with the use of the BUFGMUX in the Verilog timing netlist. These extra latches must be removed before PrimeTime can do a timing analysis. One method is to include the following PrimeTime script command:

```
set_false_path -from <bufgmux_instance_name>/* ;
set_false_path -to <bufgmux_instance_name>/* ;
```

Support

For PrimeTime specific issues, please visit Synopsys' website at http://www.synopsys.com. For other issues with this application note please contact the Xilinx hotline at:

- Web case: http://support.xilinx.com/support/clearexpress/websupport.htm
- Telephone: 1-800-255-7778

Known Issues

There are no known issues for this release.

New known issues will be logged in the Xilinx Answer Database accessible from http://support.xilinx.com. PrimeTime related issues can be searched using the "PrimeTime" keyword.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/19/01	1.0	Initial Xilinx release.
10/23/01	1.1	Fixed typo in Figure 1.