XAPP411 (v2.1) March 8, 2004

Post Layout Static Timing Analysis of Xilinx Platform for Programmable Systems FPGA Using PrimeTime

Author: Chris Zeh

Summary

This application note describes the interface, setup, and special conditions necessary to use PrimeTime for post-layout-static-timing analysis (STA) with Xilinx devices (Virtex-II, Virtex-II Pro, and Spartan-3). The design, delay, and timing constraint information used by the Xilinx implementation environment is translated to relevant files for use with PrimeTime. This application note is for designers who already use PrimeTime to isolate and close timing issues, and to obtain advance timing analysis features for Xilinx Virtex-II, Virtex-II Pro, and Spartan-3 families. This application note is not intended to be a comprehensive explanation of the Synopsys PrimeTime STA tool. For more information on PrimeTime, see the documentation supplied with the tool.

Introduction

The Xilinx static-timing analysis (STA) tool is called Timing Analyzer (GUI) and TRCE (command line). The TRCE and Timing Analyzer tools are the only STA sign-off tool that Xilinx supports. The Timing Analyzer and TRCE tools analyze the Xilinx timing constraints on a placed and routed design. Xilinx also supports the similar analysis with the Synopsys PrimeTime Static Timing Analysis tool. PrimeTime can read in Verilog, Synopsys Design Constraints (SDC), and Standard Delay Format (SDF), as well as other file formats, for timing analysis.

The Xilinx/PrimeTime interface is intended for designers who are already familiar with PrimeTime for debugging timing issues and using advanced timing analysis capabilities such as case and mode analysis and flexible reporting in their designs. Xilinx has not yet certified PrimeTime as a sign-off timing analysis tool, so you must validate the final timing results with the Xilinx Timing Analyzer. If you are not familiar with the PrimeTime Static Timing Analyzer, then Xilinx recommends that you use the Xilinx Timing Analyzer to debug timing problems.

Software and Required Files

The Xilinx/PrimeTime interface requires the use of Xilinx Foundation ISE Unix or Linux version 6.2i or newer, along with Synopsys PrimeTime version 2003.12 or newer, and PERL version 5 or greater. All of the application programs and the setup files, with TCL functions, that are needed for the Xilinx/Primetime interface are supplied on the Solaris and Linux installation CDs.

The Xilinx/PrimeTime interface requires the PrimeTime compatible Verilog netlist file, the (SDF) file, and the Synopsys Design Constraints (SDC) file. The PrimeTime compatible Verilog netlist file and SDF files are created with the Xilinx "netgen -sta" command. The SDC file can be either manually created or generated with the "create_sdc" Xilinx command. The Verilog file is a netlist that constrains LUTs, flip-flops, Block Rams, and other components from the SimPrims library. The SDF contains all of the delays for the design and the SDC, created by the Create_SDC program, contains all of the translated PCF constraints.

The main focus of the supported Xilinx/Synopsys interface is to have PrimeTime do a similar analysis as the Xilinx Timing Analyzer for Virtex-II, Virtex-II Pro, and Spartan-3. To perform a similar analysis, Xilinx has provided PrimeTime with the design's connectivity, delay information, and timing constraints. This interface greatly helps to minimize false and missing timing violations reported by PrimeTime, and improves designer productivity by quickly identifying and resolving real timing violations, and achieving timing closure using PrimeTime. This is done through the combination of the netgen and create_sdc commands. During the translation of the timing constraint from PCF to SDC, some constraints might not map one-to-one. Therefore, some PCF constraints may require manual translation to SDC. The create_sdc.log file will report the PCF constraints that were not translated to SDC for some reason.

In addition to the netgen and create_sdc commands, the SimPrims library and setup file are needed. The SimPrims library is located at $XILINX/synopsys/libraries/primetime/ and is called “simprims.db.” The setup file, with the TCL commands, for PrimeTime is located at $XILINX/synopsys/examples and is called “template.synopsys_pt.setup.” The “template.synopsys_pt.setup” file can be read in automatically (by renaming it to “.synopsys_pt.setup”) and placed in the current working directory. This file contains the majority of the TCL commands that are needed for a successful analysis with PrimeTime. Another TCL script is also provided, called PTC.pt, and it is also located at the same location as the setup file. The PTC.pt script is used to tell PrimeTime which Path Tracing Controls (PTC) were used by the Xilinx Timing Analyzer.

Recommended Timing Constraints

Many of the Xilinx timing constraints can be translated from the Physical Constraints File (PCF) format to the SDC format. The PCF is generated from the constraints that are in the User Constraints File (UCF). The analysis results, from the Xilinx Timing Analyzer, are based upon the PCF/UCF timing and exception constraints. To create a UCF that will be easily translated into SDC by the create_sdc command, Xilinx recommends the following list of valid timing constraints:

Period Constraint

- Single period constraint format - to constrain all the paths between sequential elements, flip-flops, Block Rams, etc., for a given clock domain.

```
NET "clock_in" TNM_NET = "clock_in";
TIMESPEC "TS_clock_in" = PERIOD "clock_in" 10 ns HIGH 50%;
```

Note: Avoid using PERIOD constraints on internal nets.

Offset Constraint

- OFFSET IN - to constrain all the paths from the FPGA’s primary inputs to the IOB flip-flops.
- OFFSET OUT - to constrain all the paths from IOB flip-flops to FPGA’s primary outputs.
- Global Offset constraints

```
OFFSET = IN 5 ns BEFORE "clock_in";
OFFSET = OUT 5 ns AFTER "clock_in";
```

- Net Offset Constraints

```
NET net_name OFFSET = IN 5 ns BEFORE "clock_in";
NET net_name OFFSET = OUT 5 ns AFTER "clock_in";
```

Set_max_delay (Multi-cycle) Constraints

- To constrain combinatorial paths or paths that are faster or slower than the period constraint paths. Do not use to constrain input-to-register and register-to-output paths.

```
TIMESPEC TS_P2P = FROM PADS TO PADS 10 ns;
TIMESPEC TS_F2R = FROM FFS TO RAMS 8 ns;
```

Exception Constraints

- An exception constraint used to remove paths from the analysis

```
TIMESPEC TS_RAM2FF_TIG = FROM RAMS TO FFS TIG;
TIMESPEC TS_FF12FF12_TIG = FROM instance_ff1 TO instance_ff12 TIG;
```

Note: Avoid using PIN TIG constraints.

The create_sdc command converts these PCF equivalent timing and exception constraints to equivalent SDC commands. Since the SDC file contains non-SDC constructs such as "foreach", using the "read sdc" command will result in a command failure. You must *source* the resulting SDC file in PrimeTime to apply these constraints.

If you want to manually create the SDC from the UCF, the following table shows the correlation between the UCF constraints and the SDC constraints:

UCF Timing and Exception Constraints	SDC Timing and Exception Constraints
PERIOD	create_clock
OFFSET IN (for single input port)	set_input_delay
OFFSET IN (global – associated with all the input ports that are clocked by a specific clock in the design)	No equivalent command available. Translation is done by identifying the input ports that are clocked by a specific clock and use set_input_delay to constrain them.
OFFSET OUT (single output port)	set_output_delay
OFFSET OUT (global - associated with all the output ports that are clocked by a specific clock in the design)	No equivalent command available. Translation is done by identifying the input ports that are clocked by a specific clock and use set_input_delay to constrain them.
Multi-cycle	set_max_delay
TIG	set_false_path

Design Flow

The Xilinx/PrimeTime interface is based on a design that has been implemented through NGDBuild, MAP and PAR to produce the NCD and PCF files. Review the PAR report to ensure that the design has been successfully implemented and then create the PrimeTime compatible Verilog netlist file, SDF, and SDC files. The design flow is shown in the following figure:

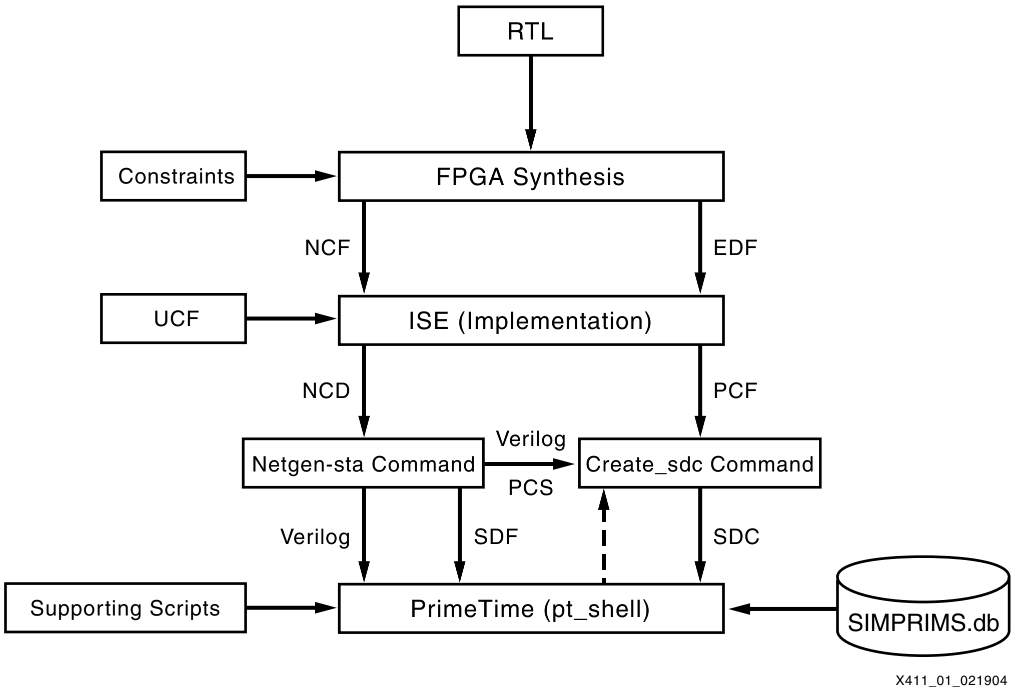

Figure 1: **Design Flow**

- The synthesis tool generates an EDIF netlist and a timing constraint file in the Xilinx Native Constraint Format, NCF.
- You can then invoke the Xilinx Constraint Editor in ISE and create additional timing constraints in UCF.
- In ISE, MAP (assigns LUTs, flip-flops, registers, etc. to available resources on the silicon) merges the NCF and UCF files to form the PCF file that drives timing driven PAR for implementing the design. In addition to timing constraints, the PCF file contains physical information to direct the Placer in ISE.
- Following a successful PAR, run the NetGen program with the –sta option (developed and distributed by Xilinx) that creates a PrimeTime compatible Verilog netlist file, design_sta.v, and the associated SDF, design_sta.sdf.
- The Create_SDC program (developed and distributed by Xilinx) creates PrimeTime's compatible SDC constraints files.

Note: The program invokes PrimeTime in the background, so PrimeTime should be set up and a license should be available. The program loads the design and a few basic constraints, and extracts additional information from PrimeTime to complete the PCF to SDC translation.

- Supporting Scripts are a set of custom TCL scripts that are co-developed with Synopsys to assist PrimeTime in understanding other important timing and delay information from the Xilinx design files. All of the TCL scripts are included in the template.synopsys_pt.setup file.

NetGen has the following options when you enter: netgen –help sta. NetGen extracts the design data from the NCD input file and generates a PrimeTime compatible Verilog netlist file. Do not use the –ngm option in this flow. Your design should be flattened prior to static timing analysis in PrimeTime. The following is an example of the netgen command, overall usage, and a description of several of the required and optional switches.

Example syntax: netgen –sta –pcf main.pcf main.ncd

Usage: netgen <-sta> [-aka] {-bd <BRAM_data_file>[tag <tagName>]}

[-dir dir_name>] [-fn] [-intstyle silent] [-mhf] [-module] [-ne] [-ngm <ngm_file>]

[-pcf <pcf_file>] [-s <speed>] [-tm <top_module_name>] [-w] <infile> [<outfile>]

Where:

-sta	(Required) Generate netlist compatible with static timing analysis tool.
-dir <dir_name>	(Optional) Specify the directory to write the output files.
-fn	(Optional) Flatten the output netlist. (Used with –ngm option)
-mhf	(Optional) Write out multiple hierarchical files. (Use with –ngm option)
-pcf <pcf_file>	(Optional) Use the pcf_file as the input constraint file.
-s <speed>	(Optional) Specify the speed grade, or for minimum delays use '-s min'.
<infile>	(Required) Input file: 'design.ncd'.
<outfile>	(Optional) Output file name (default is '<infile>_sta.v').

Create_SDC has the following options when you enter: create_sdc –help. Create_SDC translates Xilinx design constraints (PCF) to (SDC). The following is an example of the create_sdc command, overall usage, and a description of several of the required and optional switches.

Example syntax: create_sdc –pcf main.pcf –v main_sta.v –pcs main_sta.pcs –sdc main_sta.sdc

Usage: create_sdc -pcf <pcf_file> [-sdc <sdc_file>] -v <netlist_file> -pcs <pcs_file>

Where:

-pcf <pcf_file>	{Required item} Specify the name of the Xilinx PCF constraints file (.pcf).
[-sdc <sdc_file>]	{Optional item} Specify the name of the Synopsys SDC constraints file (.sdc). If this option is not specified, the default SDC file name will be <pcf file>.sdc
-v <netlist_file>	{Required item} Specify the name of the Verilog netlist file (.v). This file is generated from the Xilinx NETGN tool for static timing analysis in PrimeTime.
-pcs <pcs_file>	{Required item} Specify the name of the Xilinx PCS intermediate database file (.pcs).

An example of the main script, to be run in PrimeTime, is below and is called “primetime.pt”. This example script shows the PrimeTime commands that should be used to do a successful timing analysis on a design. The script can be used with any design by changing the file names in the script and placing it into the current working directory. The majority of the commands have the messages written out to a log file. Once the individual PrimeTime commands are executed, you can review the log files for warning and messages. To run the script, you must *source* the file from within PrimeTime or type “pt_shell –f primetime.pt” on the command line. Please ensure that the template.synopsys_pt.setup resides either in your work directory or your home directory before your run the TCL commands.

```
set search_path "."
set link_library /xilinx6install/synopsys/libraries/primetime/simprims.db
read_db /xilinx6install/synopsys/libraries/primetime/simprims.db
read_verilog   main_sta.v > read_verilog.log
read_sdf  -analysis_type on_chip_variation  main_sta.sdf > read_sdf.log
source   /xilinx6install/synopsys/examples/template.synopsys_pt.setup
source -echo -verbose   main_sta.sdc > source_sdc.log
set timing_disable_recovery_removal_checks  true
################# Set PTC variables enabled/disabled ############
set XPTC_reg_sr_q disabled
set XPTC_reg_sr_clk enabled
set XPTC_lat_d_q disabled
set XPTC_ram_d_o disabled
set XPTC_ram_we_o enabled
set XPTC_tbuf_t_o enabled
set XPTC_tbuf_i_o enabled
source PTC.pt
#################################################
# Get the list of paths with positive clock skew
#################################################
ipcs > ipcs.log
########################################################
#Required only for Virtex2p(v2p)/Spartan3(s3) and higher, since the path
#defines MIN/MAX, whereas Virtex2 defines only MAX
#By default PT takes MIN/MAX whereas TRACE takes MAX/MAX
#So this script configures PT for MAX/MAX
xp_clock_latency -arch v2p > xp_clock_latency.log
########################################################################
#Required for analysis based upon a select line of a mux or similar design
#Example is the select line of the BUFGMUX on VirtexII, VirtexII Pro, Spartan3 and higher.
set_case_analysis 1 VCC
set_case_analysis 1 SEL
#########################################################################
xp_check > xp_check.log
check_timing -verbose > check_timing.log
report_analysis_coverage > report_analysis_coverage.log
report_timing -nworst 100 -to FD1_inst/I > with1_script_rt.report
report_timing -path_type full -significant_digit 3 -nworst 1000 -input_pins -nets >
primetime.rpt
quit
```

Special Consideration

In ISE 6.2i, the translation of PCF to SDC and the differences in timing analysis styles are subject to the following limitations that impact the correlation between TRCE and PrimeTime. To ensure accurate timing analysis in PrimeTime, Xilinx used a number of internal and customer designs with various degrees of complexity to identify potential timing discrepancies between the two tools. Xilinx compared the same paths (starting points, midpoints, and ending points) for both PrimeTime and Xilinx Timing Analyzer. If the paths are within five percent of each other, then PrimeTime and Xilinx Timing Analyzer correlate.

Path Tracing Controls (PTC) on the Design

In association with the constraints in the PCF file, there is another group of commands that controls the global timing behavior of the Xilinx library cells in Xilinx Timing Analyzer. These commands control certain timing paths within library cells, for example, the register reset/set to output timing delay can either be enabled or disabled. The following table describes these PTC commands:

Commands	Path Type	Default in TRCE
reg_sr_q	Set/reset to output propagation delay	Disabled
reg_sr_clk	Set/reset to clock setup and hold checks	Enabled
lat_d_q	Data to output transparent latch delay	Disabled
ram_d_o	RAM data to output propagation delay	Disabled
ram_we_o	RAM write enable to output propagation delay	Enabled
tbuf_t_o	TBUF 3-state to output propagation delay	Enabled
tbuf_i_o	TBUF input to output propagation delay	Enabled
io_pad_i	IO pad to input propagation delay	Enabled
io_t_pad	IO 3-state to pad propagation delay	Enabled
io_o_i	IO output to input propagation delay*	Enabled
io_o_pad	IO output to pad propagation delay	Enabled

*Disabled for 3-stated IOBs.

Xilinx created a TCL script called "PTC.pt" to translate these commands to the corresponding SDC command. This script is called from the PrimeTime's main command script. In the main script, you have the option of either disabling or enabling these commands. Typically, the default value of PTCs are not changed

Positive Clock Skew between Synchronous Elements

A positive clock skew occurs on register-to-register paths when the clock delay to the destination register is greater than the clock delay to the source register, (i.e., D2 > D1), as shown in the following figure:

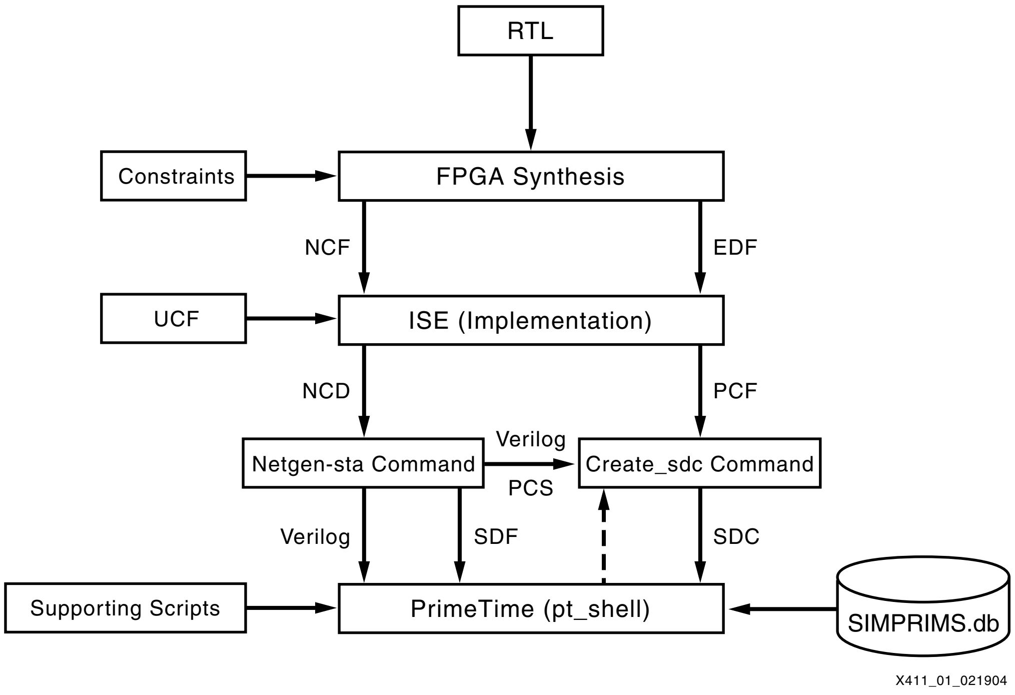

Figure 2: **Clock Skew between Synchronous Elements**

A positive clock skew effectively lengthens the available clock period in which the signal has to propagate. For proper worst-case calculation, the minimum value of the positive clock skew must be used, rather than maximum. In order to be conservative, Xilinx currently assumes a minimum value of zero for positive clock skew. This leads to an inconsistency with PrimeTime, since the maximum delay is passed to PrimeTime through SDF, and PrimeTime does not automatically make the same calculation. Xilinx, in cooperation with Synopsys, created a script that is included in the template.synopsys_pt.setup file and is called "ipcs". This script analyzes data paths in a clock domain and reports if the path has a positive clock skew. It then produces a report of starting and end points that have positive clock skew. Since Xilinx Timing Analyzer does not include positive clock skew in the setup analysis, this report can be used to compare the difference between Xilinx Timing Analyzer and PrimeTime.

Min-Max Analysis

Xilinx strongly recommends not using the absolute minimum delay SDF to perform Min-Max analysis in conjunction with the SDF file containing the maximum delay information. The SDF for Virtex-II Pro, and Spartan-3, include maximum and relative minimum values for min-max analysis. The SDF for Virtex-II only includes maximum values for max analysis.

Xilinx created a TCL script called "xp_clock_latency" that emulates the skew calculation that is done by Xilinx Timing Analyzer. This script is included in the template.synopsys_pt.setup file. In addition to the TCL command, an option must be set to specify to the script which architecture is being used. The supported architectures are Virtex-II Pro and Spartan-3. The options are "-arch v2p" and "-arch s3". An example is shown in the primetime.pt example script file. This TCL script evaluates the clock domains to determine if they are on local or global routing in the device.

LUT Elements

The functionality of LUT elements is specified with the “INIT” strings and with Verilog "defparam” constructs. PrimeTime only accepts synthesizable Verilog syntax; the create_sdc command comments out all of the simulation construction, including the “defparam” in the Verilog netlist. Consequently, the LUT elements appear as black boxes to PrimeTime, and PrimeTime must understand their functionality.   The LUT function is represented as a configured MUX function when the “defparam" constructs are translated.

In addition to performing a case analysis for each LUT element, Xilinx created the"xp_check" TCL script to set the data propagation path through the LUT elements. This command is included in the template.synopsys_pt.setup file. The xp_check script scans the netlist in PrimeTime, identifies all X_LUT4MUX16 elements that are configured to represent 2-1 MUX and sets the proper data propagation path through the MUX’s input to the output during the case analysis. This script should be called from PrimeTime’s main script after set_case_analysis commands and before issuing report_timing in PrimeTime. The work-around is a short-term solution. The long-term solution is for PrimeTime to support the “defparam” construct in the netlist.

Multiplexing Two Clocks with a BUFGMUX

Xilinx recommends that when you instantiate a BUFGMUX element, for multiplexing two clocks, in the RTL code, you should perform case analysis to set the select line of the BUFGMUX to the correct setting for analysis. The xp_check script is still required to condition X_LUT4MUX16 elements that are configured for multiplexing two data signals.

Finding the Minimum Period Path

The minimum period that is reported in Xilinx Timing Analyzer can be confusing. If you are trying to correlate between the Xilinx Timing Analyzer and Primetime, the order of paths might be different. The Xilinx Timing Analyzer reports the constrained paths in order of slack. The first path in the list usually sets the Minimum Period value. If these do not match, then a more detailed path report must be created to examine more paths. The path that is setting the minimum period value may be a two-phase path, or using both edges. In this case, the slack is based upon half the period requirement, but the minimum period value is based upon twice the path delay. Another way to determine which path sets the minimum period is in the data sheet section of the timing report that states the various cross-clock paths.

Pessimistic Delays for SRL16 Cells

Since the SDF contains very conservative delay values for SRL16 cells, PrimeTime might show timing violations on these paths, when Xilinx Timing Analyzer might not.

Other Un-Supported Features

Some of the other issues that the Xilinx/PrimeTime interface has are dealing with DCMs, Timing Ignores (TIGs), and time group creation. PrimeTime does not understand the translations of the following:

- A PERIOD constraint on the CLKFX pin of the DCM. Xilinx is working with Synopsys to fix this in the future.
- The Xilinx FEEDBACK constraint that informs the Xilinx Timing Analyzer that you are using a non-internal feedback path for one or more DCMs. The translation of internal feedback clocks to the DCM is translated correctly.
- The multi-cycle constraint from pads to pad that includes a DCM in the path.
- A period constraint placed on the output net of a DCM or internal nets.
- A TIG on a net or pin.
- The creations of a timing group based upon the rising or falling edges, which is common for constraining Dual Date Rate Inputs and Outputs.

Common Errors and Warnings

After running many regressions on both internal and customer designs, Xilinx has generated a list of errors and warnings that you might encounter from reading the SDC file and running the check_timing command. Some of the errors and warnings have a number associated with the message. You can get more information about each of these messages by typing “man” and then the number, for example: man UITE-416. The following are the common messages:

SDC Errors/Warnings:

Error: Cannot use 'set_output_delay' command on port 'I_cpu_addr[0]'.

Reason: User has specified an offset-out on a TIMEGRP containing input ports.

Warning: Invalid delay direction for port 'I_cpu_addr[0]'.

Reason: User has specified set_output_delay on the input ports.

Warning: There is 1 invalid end point for constrained paths. (UITE-416)

Reason: User has specified a path and the end point may be invalid.

Warning: There is 1 invalid end point for unconstrained paths. (UITE-416)

Reason: User has specified a path and the path is not constrained.

Warning: Negative clock latency specified: -1.113 (UITE-150)

Reason: The create_sdc command called the get segment delay TCL script and the latency calculation for the DCM is negative.

Check Timing Errors/Warnings:

Warning: There are 21 ports with no clock-relative input delay specified.

Reason: No set_input_delay was specified, so PrimeTime creates a default imaginary clock for all input ports that are not used. To have PrimeTime not create a default imaginary clock for all unused input ports, set the value of timing_input_port_default_clock to FALSE.

Warning: There are 83 endpoints, which are not constrained for maximum delay.

Reason: No constraints defined on these output ports.

Warning: There are 3 ports with no clock-relative input delay specified.

Reason: No set_input_delay was specified, so PrimeTime creates a default imaginary clock for all input ports that are not used. To have PrimeTime not create a default imaginary clock for all unused input ports, set the value of timing_input_port_default_clock to FALSE.

Warning: There are 16471 register clock pins with no clock.

Reason: Clock is not defined and may be driven by an internal source. PrimeTime does not expect a set_clock command on an internally driven clock.

Warning: There are 252 ports with no clock-relative input delay specified. Since the variable 'timing_input_port_default_clock' is 'true', a default input port clock will be assumed for these ports.

Reason: No set_input_delay was specified, so PrimeTime creates a default imaginary clock for all input ports that are not used. To have PrimeTime not create a default imaginary clock for all unused input ports, set the value of timing_input_port_default_clock to FALSE.

Conclusion

As Xilinx continues to improve the PrimeTime flow, it is recommended that you use the Xilinx Timing Analyzer as the final sign-off static timing analyzer. Xilinx is continuously improving the usability and reliability of the flow with PrimeTime and Xilinx implementation tools. Xilinx provides many TCL scripts to work-around many of the interface issues.

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.
2/29/04	2.0	Revised Application Note XAPP411.
3/08/04	2.1	Edits to pg. 4, Figure 2 caption, pg. 7, and author name.