SYNOPSYS®

CUSTOMER EDUCATION SERVICES

PrimeTime Workshop

Lab Guide

10-I-034-SLG-015 2018.06

Synopsys Customer Education Services
690 E. Middlefield Road
Mountain View, California 94043

Workshop Registration: http://training.synopsys.com

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Document Order Number: 10-I-034-SLG-015
PrimeTime Lab Guide

1 PrimeTime Flow

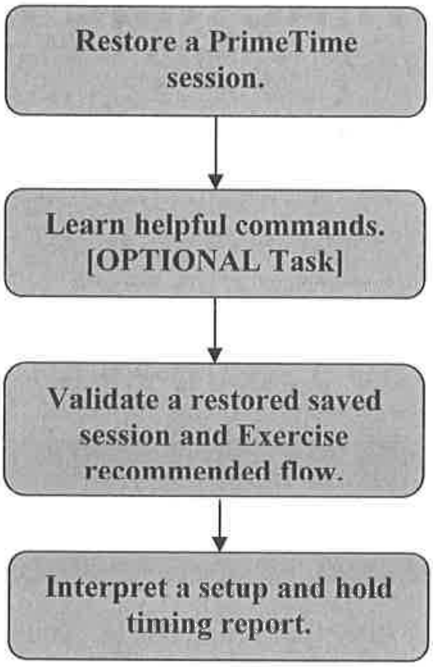
Learning Objectives

After completing this lab, you should be able to:

- Restore a previously saved PrimeTime session
- OPTIONALLY, Take advantage of helpful PrimeTime commands that will make you more efficient when using PrimeTime interactively and show you how to find more information on commands, variables and your design library
- Validate a restored save-session.
- Exercise recommended Primetime flow
- Interpret key components of a timing report for setup and hold timing checks

Lab Duration:
45 minutes

Overview


Restore a PrimeTime session.

Learn helpful commands. [OPTIONAL Task]

Validate a restored saved session and Exercise recommended flow.

Interpret a setup and hold timing report.

Relevant Files and Directories

All files for this lab are located in the **`lab1_flow`** directory under your home directory.

`lab1_flow/`	Current working directory
`common_setup.tcl`	multi-tool shared setup file
`pt_setup.tcl`	tool-specific PrimeTime setup file
`pt_scripts/`	Run file directory
`pt.tcl`	Run file
`.synopsys_pt.setup`	automatically-read PT setup file.
`orca_savesession`	Saved session directory
RUN.tcl	Run script for orca_savesession

Answers / Solutions

This lab guide contains answers and solutions to all questions. If you need some help with answering a question or would like to confirm your results, check the back portion of this lab.

Instructions

Your goal is to get used to PrimeTime by validating a restored save session, exercising the recommended flow, and analyzing the setup/hold reports.

Task 1. Restore a PrimeTime Sesssion

Invoke a previously saved PrimeTime session to perform STA.

1. Invoke PrimeTime from the **`lab1_flow`** workshop lab Unix directory.

```
unix% cd lab1_flow
unix% pt_shell
```

2. Restore a previously saved PrimeTime session. This step will read in the design netlist, libraries, and constraints. The design is now ready for analysis.

Note: The **`orca_savesession`** below is a Unix directory.

Note: The **`orca_savesession`** can be recreated, if needed, using: **`pt_shell -f RUN.tcl | tee -i run.log`**

Note: Any *PARA-124 Errors* during the execution of RUN.tcl can be safely ignored for the purpose of our labs.

Note: PrimeTime supports command, option, variable and file completion. Type a few letters and then hit the tab key.

```
pt_shell> restore_session orca_savesession
```

3. Generate coverage analysis report

```
pt_shell> report_analysis_coverage
```

Question 1. What is the name of the design under analysis?

....

Question 2. How many *setup* and *hold* violations does **ORCA** have?

....

4. Generate global timing report

```
pt_shell> report_global_timing
```

Question 3. How many are reg-reg *setup* and *hold* violations?

....

Task 2. [OPTIONAL TASK] Explore Helpful Commands

1. Execute the following three history short cut commands:

```
pt_shell> history
pt_shell> !!
pt_shell> !2
```

Question 4. Describe the difference between the last two history commands above.

....

2. Use up and down arrows to scroll through the history event list as an alternative to the previous step.

   Type the following to see all the available key bindings (in the default emacs editing mode).

```
pt_shell> list_key_bindings
```

3. Explore the **page mode** alias; execute the following command, which will generate a report that scrolls off the screen:

```
pt_shell> report_timing -group [get_path_group *]
```

4. Turn on page mode.

```
pt_shell> page_on
pt_shell> !rep
```

5. Use the **space bar** to page through a long report. Quit from a long report in page mode by typing "**q**". If you want to turn off page mode, use the command alias **page_off**.

6. Send a timing report to a separate window with the **view** Tcl procedure.

```
pt_shell> view report_timing –group [get_path_group *]
```

7. Find the command to restore a PrimeTime session and then display help information on this command.

```
pt_shell> help restore*
pt_shell> man restore_session
pt_shell> restore_session -help
```

Note: The following is an alternative way to display syntax help.

```
pt_shell> help -v restore_session
```

Question 5. From the last command above, does the command **`restore_session`** accept switches?

....

8. The time unit in PrimeTime is determined by the main technology library.

To find the time unit for **ORCA**, first list all libraries in memory.

Note: The * in the following report indicates the main library.

```
pt_shell> list_lib
```

9. Generate a report for the main library which will state the time unit.

Note: Use copy and paste to avoid mistyping the lib name. The time unit is at the very top of the report.

```
pt_shell> report_lib cb13fs120_tsmc_max
```

Question 6. What is the time unit used for timing reports (as well as all other reports) for the **ORCA** design?

....

Note: Do not forget to use “q” to quit from a long report in page mode and return to the `pt_shell` prompt without reading the entire report!

10. Display units used by the current design.

```
pt_shell> report_units
```

Task 3. Validate an Existing PrimeTime Session

In this task, you will validate the inputs that have been read into PrimeTime: the current design and libraries, the backannotation and constraints.

1. Verify that the current design is your top-level module: ORCA

```
pt_shell> current_design
```

2. Compare the unix paths of the libraries to what has been read into PrimeTime

```
pt_shell> printvar search_path
pt_shell> printvar link_path
pt_shell> list_libraries
```

Question 7. Have the 4 libraries in the link_path been successfully read into PrimeTime?

....

Question 8. Which library defines the defaults for time units, operating conditions, and other delay calculation information?

....

Question 9. What time unit is used?

....

3. Verify that the nets are completely annotated.

```
pt_shell> report_annotated_parasitics
```

Question 10. Are there any nets that are not annotated?

....

Question 11. What option to `report_annotated_parasitics` would be good to use as a 'next step' in debugging the missing nets?

....

4. Verify that the design is completely constrained.

```
pt_shell> check_timing
```

Question 12. What option to `check_timing` would be good to use as a 'next step' in debugging the missing constraints?

....

5. Verify that the checks in your cells are completely exercised; look at possible causes for your findings.

```
pt_shell> report_analysis_coverage
pt_shell> report_case_analysis
```

Question 13. Is it logical that many of your timing checks are untested?

....

6. Quit PrimeTime.

```
pt_shell> quit
```

Task 4. Execute the Run Script and Analyze the run

1. Execute the run script logging the results to the log file **run.log**.

```
UNIX> pt_shell -f ./pt_scripts/pt.tcl | tee -i run.log
```

Question 14. Were there any errors during the execution of the run script?

....

2. If there are any errors, address these first before moving on to the next step.

3. Evaluate your log file. With a text editor, open your log file. Search for the update timing messages (`UITE-214`), `print_message_info` output, and the `quit` output. Then, in your `profile` directory, examine the file `tcl_profile_sorted_by_cpu_time`.

Question 15. What step required the most CPU time?

....

Question 16. What commands were causing `UITE-214` messages?

....

....

Question 17. Can any of the timing updates be avoided?

....

Question 18. Why might the `quit` command output be a good place to start before reviewing your log file?

....

Task 5. Analyze STA Reports

Generate and intepret two STA reports for setup and hold for **`SYS_CLK`**.

1. Invoke PrimeTime and restore the session that you saved in the previous task

```
unix% pt_shell
pt_shell> restore_session my_savesession
```

2. Execute the following to display the clocks in **ORCA**:

```
pt_shell> report_clock
```

Question 19. How many clocks are in **ORCA**?

....

3. Create a single, "short" timing report for setup for the clock **`SYS_CLK`**. Use command-line expansion (the tab key) to expand both the command AND the options `-group` and `-path`.

```
pt_shell> report_timing -group SYS_CLK -path short
```

Note: The lines containing the data path cells and their delays are removed from the data arrival section making this report "short".

Note: The above command generates a report for setup by default.

Question 20. There are at least 4 clues that this report is for setup and not for hold. How many can you identify?

....

....

Question 21. Identify the instance names of the start and end point flip-flops.

....

Question 22. The clock skew for this timing path is 0.511ns; which two lines in the report can you use to calculate this?

....

Question 23. How does this clock skew affect slack (i.e. does the clock skew help or hurt slack)?

....

Question 24. How large is the violation in comparison to the clock period?

....

4. Generate a timing report for hold time.

The following is a short cut that will execute the last command in history starting with the letters "**`rep`**" and add the switch **`-delay min`** (which will generate a report for hold time).

```
pt_shell> !rep -delay min
```

Question 25. There are at least 4 clues that indicate this is a hold report and not a setup report. How many can you find?

....

Question 26. How does the clock skew in this hold report affect slack (i.e. does the clock skew help or hurt slack)?

....

5. Quit PrimeTime.

```
pt_shell> quit
```

This completes Lab 1. Return to lecture.

Answers / Solutions

Question 1. What is the name of the design under analysis?

The design is **ORCA**.

Question 2. How many *setup* and *hold* violations does **ORCA** have?

There are 23 setup and 53 hold violations.

Question 3. How many are **reg-reg** *setup* and *hold* violations?

There are 9 setup and 53 hold violations of type **reg-reg**.

Question 4. Describe the difference between the last two history commands above.

The command `!2` repeats the 2nd command executed in history. The command `!!` repeats the last executed command in history.

Question 5. From the last command above, does the command **`restore_session`** accept switches?

Yes, it accepts a session name, allowing a user to specify which of many saved sessions to restore.

Question 6. What is the time unit used for timing reports (as well as all other reports) for the **ORCA** design?

1ns.

Question 7. Have the 4 libraries in the link_path been successfully read into PrimeTime?

Yes – the four libraries specified with `link_library` are the same as those displayed with `list_libraries`.

Question 8. Which library defines the defaults for time units, operating conditions, and other delay calculation information?

cb13fs120_tsmc_max – the asterisk to the left of the library (from `list_libraries`) indicates ‘main library’

Question 9. What time unit is used?

```
pt_shell> report_lib  cb13fs120_tsmc_max
****************************************
Report : library
Library: cb13fs120_tsmc_max
****************************************

Time Unit                         : 1 ns
```

```
pt_shell> report_units
****************************************
Report : units
Design : ORCA
****************************************

Units
---------------------------------------------
-
Capacitive_load_unit              : 1e-12 Farad
Current_unit                      : 1e-06 Amp
Resistance_unit                   : 1000 Ohm
Time_unit                         : 1e-09 Second
```

Question 10. Are there any nets that are not annotated?

Yes: there are some internal driverless nets and some boundary pin-to-pin nets that are not annotated.

Question 11. What option to `report_annotated_parasitics` would be good to use as a 'next step' in debugging the missing nets?

`report_annotated_parasitics -help` shows options; the `-list_not_annotated` option shows the net names; you might want to focus in on the pin-to-pin nets by using the `-pin_to_pin_nets` option.

Question 12. What option to `check_timing` would be good to use as a 'next step' in debugging the missing constraints?

`check_timing -help` displays a list of options: the `-verbose` option lists the unconstrained endpoints

Question 13. Is it logical that many of your checks are untested?

Yes – test_mode, scan_en, and power_save are constrained off – performing STA on additional modes might enable more checks to be tested.

Question 14. Were there any errors during the execution of the run script?

No. The "run script" is setup such that any errors will terminate the script in the middle of execution. If the script completes, no errors occurred during execution of the run script. Moreover, from the "Diagnostics Summary" messages at the end of the log file from the quit command, there were no errors during the run.

Question 15. What step required the most CPU time?
For this lab, sourcing the constraints took the most CPU time.

Question 16. What commands were responsible for `UITE-214` messages?
`update_timing -full` and a call to a macro PLL_SHIFT which invoked `update_timing`

Question 17. Can any of the timing updates be avoided?
Yes, there is an `update_timing` both before and after `set_propagated_clocks` – the one before `set_propagated_clocks` is unnecessary. One `update_timing` command is in pt.tcl – the other is in a constraint file (orca_pt_other.tcl) sourced by orca_pt_constraints which is sourced by pt.tcl

Question 18. Why might the `quit` command output be a good place to start before reviewing your log file?
It gives a high-level summary of potential trouble spots: messages, both warning and information, timing updates, and performance statistics.

Question 19. How many clocks are in **ORCA**?

There are 6 clocks in **ORCA**.

Question 20. There are at least 4 clues that this report is for setup and not for hold. How many can you identify?

The most glaring clue is the "library setup time" in the generated report. The more subtle clues are:

- The clock edges used for the data arrival and data required are 0ns and 8ns respectively (and not 0ns and 0ns as for hold time).

- The "**Path Type**" in the header is **max** which indicates that this report is for setup (a "**Path Type**" of **min** indicates a hold report).
- Finally, the data arrival time is after the data required time and the slack is violated (whereas if this was a report for hold, the slack would be met!)

Question 21. Identify the instance names of the start and end point flip-flops.

```
Startpoint: I_ORCA_TOP/I_BLENDER/s3_op2_reg[18]
               (rising edge-triggered flip-flop clocked by SYS_CLK)
Endpoint: I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]
               (rising edge-triggered flip-flop clocked by SYS_CLK)
Path Group: SYS_CLK
Path Type: max
Min Clock Paths Derating Factor : 0.900
```

Question 22. The clock skew for this timing path is 0.511 ns. Which lines in the report can you use to calculate this?

```
Point                                                 Incr       Path
-----------------------------------------------------------------
clock SYS_CLK (rise edge)                             0.000      0.000
clock network delay (propagated)                      3.247      3.247
I_ORCA_TOP/I_BLENDER/s3_op2_reg[18]/CP (sdnrb1)       0.000      3.247 r
I_ORCA_TOP/I_BLENDER/s3_op2_reg[18]/Q (sdnrb1)        0.516 &    3.763 r
...
I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D (sdnrb1)        8.242 &   12.004 f
data arrival time                                               12.004

clock SYS_CLK (rise edge)                             8.000      8.000
clock network delay (propagated)                      2.736     10.736
.....
```

Question 23. How does this clock skew affect slack (i.e. does the clock skew help or hurt slack)?

The clock skew hurts the slack for setup in this specific timing report. The clock latency to the start point flip-flop causes the data to arrive 0.511ns later causing a larger setup timing violation..

Question 24. How large is the violation in comparison to the clock period?

The clock period is 8ns. The violation is 0.987. It is approximately 12% of the clock period.

Question 25. There are at least 4 clues that indicate this is a hold report and not a setup report. How many can you find?

The most glaring clue is the highlighted "library hold time" in the report below. The more subtle clues are:

- The clock edges used for the data arrival and data required are 0ns and 0ns respectively.
- The "**Path Type**" in the header is **min** which indicates that this report is for hold time.
- Finally, the data arrival time is before the data required time and the slack is violated.

Question 26. How does the clock skew for this hold report affect slack (i.e. does the clock skew help or hurt slack)?

From the report, you can see (after applying the clock network delay) that the capture clock edge arrives later than the launch clock edge. This hurts the hold slack, causing a violation.

```
Point                                                 Incr       Path
--------------------------------------------------------------------
clock SYS_CLK (rise edge)                             0.000      0.000
clock network delay (propagated)                      2.360      2.360
I_ORCA_TOP/I_PARSER/out_bus_reg[10]/CP (sdcrq1)       0.000      2.360 r
I_ORCA_TOP/I_PARSER/out_bus_reg[10]/Q (sdcrq1)        0.353 &    2.713 r
...
I_ORCA_TOP/I_BLENDER/rem_green_reg/D (sdcrn1)         0.142 &    2.855 f
data arrival time                                                2.855

clock SYS_CLK (rise edge)                             0.000      0.000
clock network delay (propagated)                      3.206      3.206
clock reconvergence pessimism                        -0.216      2.990
I_ORCA_TOP/I_BLENDER/rem_green_reg/CP (sdcrn1)                   2.990 r
library hold time                                     0.006      2.997
data required time                                               2.997
--------------------------------------------------------------------

data required time                                               2.997
data arrival time                                               -2.855
--------------------------------------------------------------------
slack (VIOLATED)                                                -0.142
```

2 Constraining Methodology

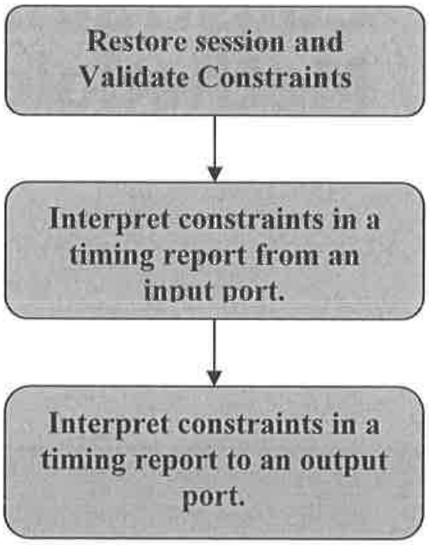
Learning Objectives

After completing this lab, you should be able to:

- Validate constraints by checking for
  - Constraint Completeness and
  - Untested timing checks
- Identify and interpret constraints in a timing report
  - Clock constraints
  - Interface constraints

Lab Duration:
30 minutes

Overview


Restore session and Validate Constraints
Interpret constraints in a timing report from an input port.
Interpret constraints in a timing report to an output port.

Relevant Files and Directories

All files for this lab are located in the `lab2_constraints` directory under your home directory.

`lab2_constraints/`	Current working directory
`orca_savesession/`	Session to restore for labs
`.synopsys_pt.setup`	PT setup file
RUN.tcl	Run script for orca_savesession

Answers & Solutions

This lab guide contains answers and solutions to all questions. If you need some help with answering a question or would like to confirm your results, check the back portion of this lab.

Instructions

Task 1. Validate constraints

1. Invoke PrimeTime from the **`lab2_constraints`** directory and restore the PrimeTime session using the **`orca_savesession`** directory.

   Note: The **`orca_savesession`** can be recreated, if needed, using: **`pt_shell -f RUN.tcl | tee -i run.log`**

   Note: Any *PARA-124 Errors* during the execution of RUN.tcl can be safely ignored for the purpose of our labs.

2. Check for constraint completeness

```
pt_shell> check_timing -verbose
```

   Question 1. Are all registers in the design clocked?

   ....

   Question 2. Are there any missing constraints? Can you explain?

   ....

3. Check for the untested timing checks in the design

```
pt_shell> report_analysis_coverage
```

   Question 3. Nearly two thirds of the setup/hold checks are untested! – What are the 2 causes?

   ....

   Question 4. Why are there unexercised `min_pulse_width` checks?

   ....

   Question 5. How many output delay constraints are there for setup and for hold and are these constraints met or violated?

   ....

Task 2. Analyze a Timing Report For Input Delay Constraint

1. Generate a report for the input delay constraints applied to the port `pad[0]`.

```
pt_shell> report_port -input_delay pad[0]
```

Question 6. What are the min and max arrival times to **`pad[0]`**?

....

Question 7. What is the name of the external start point clock constraining **`pad[0]`**?

....

2. Generate a timing report for setup starting at the port **`pad[0]`**.

Answer the following questions using this report.

Use your job aid labeled "timing reports" for help recalling the appropriate switch for report_timing.

Question 8. Which lines in the timing report did you use to ensure the reported path starts at the port **`pad[0]`** and is for setup?

....

Question 9. List all user specified constraints in this timing report.

....

Question 10. Where must the clock latency be included for the start point clock **`PCI_CLK`**?

....

Question 11. Describe the direction of the port **`pad[0]`** (i.e. is it an input, output or inout port).

....

Question 12. Describe the end point of this timing path (i.e. is it an output port or an internal flip-flop).

....

3. Generate a new report from the same port `pad[0]` for setup, which also shows the details of the calculated clock network delay.

   Use the job aid labeled "timing reports" for help recalling the appropriate switch for `report_timing`. Remember to take advantage of `history` commands.

   Question 13. How large is the clock source latency versus the clock network latency for the end point clock `PCI_CLK`?

   ....

   Question 14. Where has the clock `PCI_CLK` been defined (the clock definition point)?

   ....

4. Generate a report starting at the port `pad[0]` for hold time.

   Question 15. Does the value of the input external delay constraint match your expectations?

   ....

Task 3. Analyze a Timing Report For Output Delay Constraint

1. Generate a report for the output delay constraints applied to the port `pad[0]`.

Question 16. What are the min and max output delay constraints for this port?

....

Question 17. How will the negative min output delay constraint be applied to this port (i.e. will it impose a positive or negative hold requirement)?

....

Question 18. What is the name of the external end point clock constraining this port?

....

2. Generate a "short" timing report ending at the port `pad[0]` for hold time.

Question 19. Describe the start point of this timing path (i.e. is it an input port or an internal flip-flop).

....

Question 20. Does the path group for this timing path match your expectations?

....

Question 21. Does the "data required time" match your expectations?

....

3. Optionally, apply the following constraint which will impose a positive output delay constraint for hold on `pad[0]` and then re-execute the steps in this task to see the affect.

```
pt_shell> set_output_delay -min 1.0 -clock PCI_CLK pad[0]
```

Quit PrimeTime.

This completes Lab2. Return to lecture.

Answers / Solutions

Question 1. Are all registers in the design clocked?

Yes. There are no clock pins reported following the message Information: Checking 'no_clock'.

Question 2. Are there any missing constraints? Can you explain?

Yes. There are 2 output ports reported to be missing their output delays.

Warning: There are 2 endpoints which are not constrained for maximum delay.

sd_CKn

sd_CK

The above warning can be ignored since these 2 are the clock output ports (Use the **`report_clock`** command to confirm) that should not be constrained for output delay.

Question 3. Nearly two thirds of the setup/hold checks are untested! – What are the 2 causes?

`constant_disabled` and **`false_path`**. (Using **`report_analysis_coverage -status untested -check "setup hold"`**)

Question 4. Why are there unexercised `min_pulse_width` checks?

`min_pulse_width` checks are exercised only if the pins have clocks. Since these are non clock asynchronous pins like "set or clear", no clocks have been defined on them. (Use the command **`report_analysis_coverage -status untested -check min_pulse_width`** to confirm)

Question 5. How many output delay constraints are there for setup and for hold and are these constraints met or violated?

From **`report_analysis_coverage`**, there are 75 output delay constraints for both setup and hold; 9 output delay constraints for setup are violated; 39 output delay constraints for hold are violated. Remember to verify that all output ports are constrained for both setup as well as for hold.

```
pt_shell> restore_session orca_savesession
pt_shell> report_analysis_coverage
Type of Check         Total          Met          Violated        Untested
--------------------------------------------------------------------------------
setup                  9629     3575 ( 37%)       13 (  0%)     6041 ( 63%)
hold                   9629     3517 ( 37%)       71 (  1%)     6041 ( 63%)
recovery               1316     1210 ( 92%)        0 (  0%)      106 (  8%)
removal                1316     1204 ( 91%)        6 (  0%)      106 (  8%)
min_period               20       20 (100%)        0 (  0%)        0 (  0%)
min_pulse_width        7273     5957 ( 82%)        0 (  0%)     1316 ( 18%)
out_setup                75       66 ( 88%)        9 ( 12%)        0 (  0%)
out_hold                 75       36 ( 48%)       39 ( 52%)        0 (  0%)
--------------------------------------------------------------------------------
All Checks            29333    15585 ( 53%)      138 (  0%)    13610 ( 46%)
```

Question 6. What are the min and max arrival times to **pad[0]**?

The min and max arrival times are 2ns and 8ns respectively (with the same constraint for both rise and fall data transitions at the port pad[0]).

Question 7. What is the name of the external start point clock constraining **pad[0]**?

The name of the clock is **PCI_CLK**.

Question 8. Which lines in the timing report did you use to ensure the reported path starts at the port **pad[0]** and is for setup?

```
pt_shell> report_timing -from pad[0]
Startpoint: pad[0] (input port clocked by PCI_CLK)
Endpoint: I_ORCA_TOP/I_PCI_CORE/d_out_i_bus_reg[0]
              (rising edge-triggered flip-flop clocked by PCI_CLK)
Path Group: PCI_CLK
Path Type: max
```

Question 9. List all user specified constraints involved in this timing report.

The clock period is a constraint. The clock **PCI_CLK** is propagated (not ideal). The input external delay (which comes from an input delay constraint).

Question 10. Where must the clock latency be included for the start point clock **PCI_CLK**?

The clock network delay is zero. Therefore, the only other place to represent the external clock latency is as a part of the input delay constraint (i.e. the input external delay). The appropriate way to model this is to use the switches **-network_latency_included** and **-source_latency_included** for **set_input_delay.**

Question 11. Describe the direction of the port **pad[0]** (i.e. is it an input, output or inout port).

The port **pad[0]** is an inout port; therefore, it is both a timing path start point as well as a timing path end point!

```
Point                                                Incr       Path
----------------------------------------------------------------------
clock PCI_CLK (rise edge)                            0.000      0.000
clock network delay (propagated)                     0.000      0.000
input external delay                                 8.000      8.000 r
pad[0] (inout)                                       0.000      8.000 r
```

Question 12. Describe the end point of this timing path (i.e. is it an output port or an internal flip-flop).

The end point is a rising-edge triggered flip-flop clocked by **PCI_CLK** (it is actually a timing model that looks like a flip-flop with setup and hold timing checks).

Question 13. How large is the clock source latency versus the clock network latency for the end point clock **`PCI_CLK`**?

Shown below is only the data required time section of the timing report. The source latency is 0ns. The clock network latency is 0.768ns (15.768 – 15.000).

```
pt_shell> !rep -path full_clock
. . .

clock PCI_CLK (rise edge)                             15.000     15.000
clock source latency                                   0.000     15.000
pclk (in)                                              0.000     15.000 r
pclk_iopad/CIN (pc3d01)                                0.728 H   15.728 r
I_CLOCK_GEN/I_PLL_PCI/CLK (PLL)                       -1.220 H   14.508 r
I_CLOCK_GEN/bufbdfG1B1I1_1/Z (bufbdf)                  0.184 &   14.692 r
I_CLOCK_GEN/U21/Z (mx02d2)                             0.165 &   14.857 r
I_CLOCK_GEN/bufbdfG2B1I1_2/Z (bufbdf)                  0.153 &   15.010 r
I_CLOCK_GEN/U17/ZN (invbdk)                            0.076 &   15.086 f
I_CLOCK_GEN/U14/ZN (invbdk)                            0.059 &   15.145 r
I_CLK_SOURCE_PCLK/Z (bufbdk)                           0.168 &   15.313 r
invbd7G5B1I2/ZN (invbd7)                               0.108 &   15.421 f
I_ORCA_TOP/invbdkG5B2I3_1/ZN (invbdk)                  0.120 &   15.541 r
I_ORCA_TOP/I_PCI_CORE/buffd7G5B3I32/Z (buffd7)         0.140 &   15.680 r
I_ORCA_TOP/I_PCI_CORE/d_out_i_bus_reg[0]/CP (sdcrq1)
                                                       0.088 &   15.768 r
library setup time                                    -0.153     15.615
data required time                                               15.615
```

Question 14. Where has the clock **`PCI_CLK`** been defined (the clock definition point)?

The clock **`PCI_CLK`** is defined at the input port **`pclk`**. The clock definition point separates the clock source latency from the clock network latency.

Question 15. Does the value of the input external delay constraint match your expectations?

Yes. From `report_port` above, the input external delay should be 2ns with respect to the rising edge of `PCI_CLK`. This is confirmed in the timing report below.

```
pt_shell> report_timing -delay min -from pad[0]


Startpoint: pad[0] (input port clocked by PCI_CLK)
Endpoint: I_ORCA_TOP/I_PCI_CORE/d_out_i_bus_reg[0]
            (rising edge-triggered flip-flop clocked by PCI_CLK)
Path Group: PCI_CLK
Path Type: min
Min Data Paths Derating Factor  : 0.900
Min Clock Paths Derating Factor : 0.900

Point                                                   Incr       Path
--------------------------------------------------------------------------
clock PCI_CLK (rise edge)                              0.000      0.000
clock network delay (propagated)                       0.000      0.000
input external delay                                   2.000      2.000 r
pad[0] (inout)                                         0.000      2.000 r
pad_iopad_0/PAD (pc3b03)                               0.044      2.044 r
pad_iopad_0/CIN (pc3b03)                               0.662 &    2.706 r
```

Question 16. What are the min/max output delay constraints for this port?

```
pt_shell> report_port -help

Usage:
 report_port           # Report port info
   [-verbose]              (Show all port info)
   [-design_rule]          (Only port design rule info)
   [-drive]                (Only port drive info)
   [-input_delay]          (Only port input delay info)
   [-output_delay]         (Only port output delay info)
   [-wire_load]            (Only port wire load info)
   [-nosplit]              (Don't split lines if column overflows)
   [port_list]             (List of ports)

pt_shell> report_port -output_delay pad[0]

                   Output Delay
                  Min             Max        Related Related
 Output Port    Rise   Fall   Rise     Fall   Clock    Pin
---------------------------------------------------------------
pad[0]        -1.00   -1.00   4.00     4.00  PCI_CLK   --
```

Question 17. How will the negative min output delay constraint be applied to this port (i.e. will it impose a positive or negative hold requirement)?

In lecture, it was stated that a negative hold output delay constraint will impose a positive hold requirement.

Question 18. What is the name of the external end point clock constraining this port?

The port **pad[0]** is constrained with respect to **PCI_CLK**.

Question 19. Describe the start point of this timing path.

The start point of the timing path is an internal flip-flop.

```
pt_shell> report_timing -delay min -to pad[0]


Startpoint: I_ORCA_TOP/I_PCI_CORE/pad_out_buf_reg[0]
             (rising edge-triggered flip-flop clocked
by PCI_CLK)
Endpoint: pad[0] (output port clocked by PCI_CLK)
```

Question 20. Does the path group for this timing path match your expectations?

Yes. The path group is **PCI_CLK** which is the same as the external capture clock name.

Question 21. Does the "data required time" match your expectations?

Yes. The capture clock edge is zero, which is appropriate for hold. The hold requirement of 1ns is positive, and the propagated clock network delay is 0ns. The data required section of the timing report is shown below.

```
clock PCI_CLK (rise edge)                         0.000      0.000
clock network delay (propagated)                  0.000      0.000
output external delay                             1.000      1.000
data required time                                           1.000
```

```
# Answers for optional step
pt_shell> report_port -output_delay pad[0]
                     Output Delay
                   Min              Max          Related Related
 Output Port    Rise    Fall    Rise    Fall    Clock   Pin
------------------------------------------------------------------
  pad[0]          1.00    1.00    4.00    4.00  PCI_CLK   --

pt_shell> report_timing -to pad[0] -delay min -path short
. . .
  Point                                                 Incr       Path
  ---------------------------------------------------------------------------
  clock PCI_CLK (rise edge)                            0.000      0.000
  clock network delay (propagated)                     0.772      0.772
  I_ORCA_TOP/I_PCI_CORE/pad_out_buf_reg[0]/CP (sdcrq1)
                                                       0.000      0.772 r
  I_ORCA_TOP/I_PCI_CORE/pad_out_buf_reg[0]/Q (sdcrq1)  0.350 &    1.122 r
  pad_out_buf_reg_0_ASTttcInst778/Z (bufbd4)           0.199 &    1.321 r
  pad_iopad_0/PAD (pc3b03)                             2.081 H    3.402 r
  pad[0] (inout)                                       0.044      3.445 r
  data arrival time                                               3.445

  clock PCI_CLK (rise edge)                            0.000      0.000
  clock network delay (propagated)                     0.000      0.000
  clock reconvergence pessimism                        0.000      0.000
  output external delay                               -1.000     -1.000
  data required time                                             -1.000
  ---------------------------------------------------------------------------
  data required time                                             -1.000
  data arrival time                                              -3.445
  ---------------------------------------------------------------------------
  slack (MET)                                                     4.445


#From the above report, you can see that using a positive hold constraint
#for an output delay INCREASES the positive slack.  This confirms that
#specifying a negative hold constraint for an output delay actually
#specifies the hold requirement on the output port.
```

This page is left blank intentionally.

3 Generating Reports

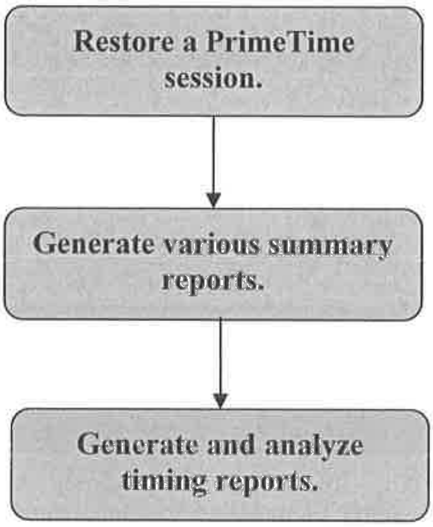
Learning Objectives

After completing this lab, you should be able to:

- Generate summary reports for the violations in **ORCA**
- Analyze timing reports for setup and hold
- Apply the correct timing report switches
- Identify half clock cycle paths

Lab Duration: 45 minutes

Overview


Restore a PrimeTime session.
Generate various summary reports.
Generate and analyze timing reports.

Relevant Files and Directories

All files for this lab are located in the `lab3_reports` directory under your home directory.

`lab3_reports/`	Current working directory
`orca_savesession/`	Session to restore for labs
`.synopsys_pt.setup`	PT setup file

Answers & Solutions

This lab guide contains answers and solutions to all questions. If you need some help with answering a question or would like to confirm your results, check the back portion of this lab.

Instructions

Task 1. Setup PrimeTime For Lab 3

1. Invoke PrimeTime from the `lab3_reports` (which is a symbolic link to the `lab2_constraints`) workshop lab directory.

   Restore the PrimeTime session using the `orca_savesession` directory.

2. Find the variable that controls the significant digits for many reports and set it to 4 significant digits. [Hint: `aa significant`]

   Question 1. What is this variable's default value? [Hint: man page]

   ....

Task 2. Generate Summary Reports

From lab 1, we know that there are setup violations in **ORCA**.

1. Answer the following questions by generating the appropriate summary reports:

   Question 2. Identify the top five setup violations with the worst slack. The required details are the endpoint names and the slack.

   ....

   Question 3. List the 2 clock domains that have violating setup timing paths, and the 5 clock domains that have violating hold timing paths (**ORCA** has 6 clock domains in total).

   ....

   Question 4. Identify how many hold violations are on input paths, how many on output paths, and how many are register-to-register violations.

   ....

2. Generate a report for the worst slack for setup to each bit of a 16-bit bus ending at the output ports `sd_DQ[0]` to `sd_DQ[15]` (the output ports are all constrained by a single clock, `SD_DDR_CLK`).

   Question 5. List the end point with the largest margin (the best slack).

   ....

3. Generate a high-level overview of the quality of the design

   Question 6. Which clock group has the highest number of violating paths?

   ....

Task 3. Analyze Timing Reports for Setup and Hold

1. Turn page mode on.
2. Execute the following command to generate a timing report for **PCI_CLK**:

```
pt_shell> report_timing -group PCI_CLK
```

Question 7. Does this timing path meet or violate timing?

....

Question 8. What type of timing path is this - internal flip-flop to flip-flop, input, or output timing path?

....

3. Generate a timing report for hold time for the same clock group **PCI_CLK**.

```
pt_shell> report_timing -group PCI_CLK -delay min
```

Question 9. What type of timing path is this - internal flip-flop to flip-flop, input, or output timing path?

....

Question 10. How many cells are on the data path of this timing path?

....

Question 11. The cell delay used for the clock pin (**CP**) to **Q** pin of the start point flip-flop is for a rise transition. Offer one possible reason why this results in a worse slack for hold than using the faster fall delay through this flip-flop?

....

In the next step, you will continue to explore and confirm your answer for the above question.

Question 12. What additional information do you need to confirm your answer for the above question?

....

4. Generate another timing report for the same timing path for hold time but with a fall transition at the end point (instead of a rise transition).

   Use copy and paste to avoid mistyping the end point and start point pin names.

   Use the job aid labeled "timing reports" to find the appropriate switches for report_timing.

Question 13. Which lines in this report did you use to confirm that the correct path has been reported?

....

Question 14. Was the guess correct – the faster fall delays results in a faster data arrival time but a smaller hold time requirement and thus a better slack?

....

Task 4. Apply the Correct Timing Report Switches

1. Answer the following questions by experimenting and exploring in PrimeTime.

   Use the job aid labeled "timing reports" for help identifying the appropriate commands and switches.

   Question 15. Write the command to generate a single timing report for each path group for setup.

   ....

   Question 16. Write the command to generate a single timing report for setup for each path group which has a violation.

   ....

   Question 17. What are the names of the two path groups that have violating timing paths in **ORCA** (the answer will come from the result of the previous question)?

   ....

   Question 18. Write the command to generate a timing report with the worst slack for setup to any output port constrained by the clock `PCI_CLK`.

   ....

   Question 19. There are a few latches in **ORCA**; write the command to identify the data pins of these latches.

   ....

   Question 20. Write the command to generate a timing report for hold to the `D` pin of the `latched_clk_en_reg` latches.

   ....

Task 5. dentify Half-Clock Cycle Paths

The clock **SDRAM_CLK** constrains many half clock cycle paths in **ORCA** (i.e. it constrains paths from a falling edge triggered flip-flop to a rising edge triggered flip-flop and vice versa).

These paths must be carefully monitored for various reasons (e.g. the duty cycle of **SDRAM_CLK** is not yet well defined or for analysis of the clock skew).

1. Execute the following command to report the clock period for **SDRAM_CLK** and use this information to answer the following questions:

```
pt_shell> report_clock SDRAM_CLK
```

Question 21. Given that the first number under the waveform column is the first rising edge for the clock **SDRAM_CLK** and the second number is the falling edge – what duty cycle has been defined for this clock?

....

Question 22. Describe the specific clock edges that will be used in a timing report for setup for a timing path constrained by the rising edge of **SDRAM_CLK** to the falling edge of **SDRAM_CLK**.

....

Question 23. For this same timing path, describe the specific clock edges that will be used in a timing report for hold timing checks.

....

2. Confirm the information in the following table by generating the appropriate timing reports for the half clock cycle timing paths constrained by the clock **SDRAM_CLK**.

Launch clock edge	Capture clock edge	Worst Setup Slack	Launch clock edge	Capture clock edge	Worst Hold Slack
Rise 0ns	Fall 3.75ns	0.680ns	Rise 7.5ns	Fall 3.75ns	3.558ns
Fall 3.75ns	Rise 7.50ns	0.635ns	Fall 3.75ns	Rise 0ns	3.514ns

Question 24. Which switch is useful for generating the worst 10 timing reports for each of these half clock cycle timing paths?

....

Question 25. Why does PrimeTime report "no constrained paths?" (hint – the options PrimeTime is using are shown immediately following the `report_timing` command)

....

Question 26. What additional option must you use to report the worst 10 timing paths?

....

3. Quit PrimeTime.

This completes Lab 3. End of Day-1.

Answers / Solutions

Question 1. What is this variable's default value?

```
set_app_var  report_default_significant_digits -default
→ report_default_significant_digits = "2"
pt_shell> set report_default_significant_digits 3
```

Question 2. Identify the top five setup violations with the worst slack. The details that are required are the endpoint names and the slack.

The following command will list all setup violations sorted by slack. Use page mode to quit from the long report because the only information desired are the top 5 violations.

```
# No need to type the entire command name!
pt_shell> report_analysis -status violated -check setup -nosplit

Constrained                                Related             Check
Pin                                        Pin       Clock     Type    Slack
-----------------------------------------------------------------------------
I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D     CP(rise)  SYS_CLK   setup  -0.9872
I_ORCA_TOP/I_BLENDER/s4_op1_reg[31]/D     CP(rise)  SYS_CLK   setup  -0.8410
I_ORCA_TOP/I_BLENDER/s4_op2_reg[30]/D     CP(rise)  SYS_CLK   setup  -0.8305
I_ORCA_TOP/I_BLENDER/s4_op1_reg[15]/D     CP(rise)  SYS_CLK   setup  -0.6918
I_ORCA_TOP/I_BLENDER/s4_op1_reg[30]/D     CP(rise)  SYS_CLK   setup  -0.6843
```

Question 3. List the 2 clock domains that have violating setup timing paths, and the 5 clock domains that have violating hold timing paths (**ORCA** has 6 clock domains in total). (the following answer just shows the path group headers – the names of the endpoints have been removed to conserve space.

```
pt_shell> report_constraint -all_violators \
                     -max_delay -min_delay

max_delay/setup ('PCI_CLK' group)
max_delay/setup ('SYS_CLK' group)

min_delay/hold ('PCI_CLK' group)
min_delay/hold ('SDRAM_CLK' group)
min_delay/hold ('SD_DDR_CLK' group)
min_delay/hold ('SYS_2x_CLK' group)
min_delay/hold ('SYS_CLK' group)
```

Question 4. Identify how many hold violations are on input paths, how many on output paths, and how many are register-to-register violations.

```
pt_shell> report_global_timing

Hold violations
--------------------------------------------
       Total    reg->reg in->reg reg->out in->out
--------------------------------------------
WNS     -0.4375  -0.1420 -0.2363 -0.1281  -0.4375
TNS    -12.6317  -3.5616 -2.5768 -0.5954  -5.8979
NUM      100        59      18       7       16
```

Question 5. List the end point with the largest margin (the best slack).

The output port **`sd_DQ[0]`** has the largest margin at 1.5994ns.

Generally, the following command will only generate a single report for every end point because nworst is, by default, 1 and there is only a single clock constraining every output port. However, because increasing the value of `max_paths` causes an implicit `slack_lesser` 0 to be used, and because all the slacks to this endpoint are positive, PrimeTime will not report any paths unless we change the value of `slack_lesser` to a large positive number – in this case, 100.

```
pt_shell> report_timing -path end -max 16 -slack_lesser 100 -to sd_DQ*
****************************************
Report : timing
        -path_type end
        -delay_type max
        -slack_lesser_than 100.0000
        -max_paths 16
        -sort_by slack
Design : ORCA
Version: J-2014.06
Date   : Wed Jul 23 12:01:51 2014
****************************************
Endpoint              Path Delay   Path Required    CRP      Slack

sd_DQ[14] (inout)      8.4133 f       9.6649       0.2418   1.4933
sd_DQ[6] (inout)       8.4133 f       9.6649       0.2418   1.4934
sd_DQ[5] (inout)       8.4133 f       9.6649       0.2418   1.4934
sd_DQ[4] (inout)       8.4133 f       9.6649       0.2418   1.4934
sd_DQ[3] (inout)       8.4133 f       9.6649       0.2418   1.4934
sd_DQ[10] (inout)      8.4112 f       9.6649       0.2418   1.4955
sd_DQ[15] (inout)      8.4111 f       9.6649       0.2418   1.4955
sd_DQ[13] (inout)      8.4111 f       9.6649       0.2418   1.4955
sd_DQ[12] (inout)      8.4111 f       9.6649       0.2418   1.4955
sd_DQ[9] (inout)       8.4111 f       9.6649       0.2418   1.4955
sd_DQ[11] (inout)      8.4106 f       9.6649       0.2418   1.4960
sd_DQ[2] (inout)       8.4054 f       9.6649       0.2418   1.5013
sd_DQ[1] (inout)       8.4054 f       9.6649       0.2418   1.5013
sd_DQ[8] (inout)       8.4032 f       9.6649       0.2418   1.5035
sd_DQ[7] (inout)       8.4032 f       9.6649       0.2418   1.5035
sd_DQ[0] (inout)       8.0654 f*      9.6649       0.0000   1.5994
```

Question 6. Which clock group has the highest number of violating paths?

```
report_qor -only_violated
```

```
SDRAM_CLK has 30 violating paths
```

Question 7. Does this timing path meet or violate timing?

It violates timing with a slack of -1.157ns.

Question 8. What type of timing path is this - internal flip-flop to flip-flop, input, or output timing path?

This is an output timing path ending at the output port named `pad[1]` .

Question 9. What type of timing path is this - internal flip-flop to flip-flop, input, or output timing path?

This is an internal timing path. The end point looks like a flip-flop, but is one of many timing arcs in a RAM timing model.

Question 10. How many cells are on the data path of this timing path?

There is only one cell on this data path, the start point flip-flop. The timing path consists of a start point flip-flop tied directly to the end point flip-flop.

Question 11. The cell delay used for the clock pin (`CP`) to `Q` pin of the start point flip-flop is a rise delay. Offer one reason why this would result in a worse slack for hold than using a fall delay through this flip-flop?

Typically, fall delays are faster than rise delays and would offer a worse slack for hold! The one exception is if the fall delay at the data pin of the end point flip-flop resulted in a smaller library hold time requirement. This is what occurs in this case.

Question 12. Write the command to generate a single timing report for each path group for setup.

```
pt_shell> report_timing -group [get_path_group *]
```

Question 13. What additional information do you need to confirm your answer for the above question?

Generate another timing report where the data arrival time is calculated with fall transition at the end point and compare the two reports. In this way you can confirm that the library hold time is in fact smaller with a falling transition at the data pin of the end point flip-flop and thus the resulting slack better. You will explore this in the next lab step.

Question 14. Which lines in this report did you use to confirm that the correct path has been reported?

Note: The backslash in the command below is a line continuation character.

```
pt_shell> report_timing -delay min_fall \
                    -to I_ORCA_TOP/I_PCI_READ_FIFO/PCI_RFIFO_RAM/A1[1] \
                    -from I_ORCA_TOP/I_PCI_READ_FIFO/count_int_reg[1]1/CP
```

```
Startpoint: I_ORCA_TOP/I_PCI_READ_FIFO/count_int_reg[1]1
              (rising edge-triggered flip-flop clocked by PCI_CLK)
Endpoint: I_ORCA_TOP/I_PCI_READ_FIFO/PCI_RFIFO_RAM
              (rising edge-triggered flip-flop clocked by PCI_CLK)
Path Group: PCI_CLK
Path Type: min
Min Data Paths Derating Factor  : 0.900
Min Clock Paths Derating Factor : 0.900

Point                                                      Incr       Path
----------------------------------------------------------------------------
clock PCI_CLK (rise edge)                                 0.000      0.000
clock network delay (propagated)                          0.779      0.779
I_ORCA_TOP/I_PCI_READ_FIFO/count_int_reg[1]1/CP (sdcrq1)
                                                          0.000      0.779 r
I_ORCA_TOP/I_PCI_READ_FIFO/count_int_reg[1]1/Q (sdcrq1)
                                                          0.344 &    1.123 f
I_ORCA_TOP/I_PCI_READ_FIFO/PCI_RFIFO_RAM/A1[1] (ram32x32)
                                                          0.016 &    1.139 f
data arrival time                                                    1.139

clock PCI_CLK (rise edge)                                 0.000      0.000
clock network delay (propagated)                          1.088      1.088
clock reconvergence pessimism                            -0.274      0.814
I_ORCA_TOP/I_PCI_READ_FIFO/PCI_RFIFO_RAM/CE1 (ram32x32)              0.814 r
library hold time                                         0.211 *    1.025
data required time                                                   1.025
----------------------------------------------------------------------------
data required time                                                   1.025
data arrival time                                                   -1.139
----------------------------------------------------------------------------
slack (MET)                                                          0.114
```

Question 15. Was the guess correct – the faster fall delays results in a faster data arrival time but a smaller hold time requirement and thus a better slack?

Yes! The data arrival time is faster (1.139ns versus 1.150ns) but the hold time requirement is smaller (0.211ns versus 0.411ns) thus the slack is better than the original timing report. Recall that hold time (and setup time) are a function of the transition at the data pin of the flip-flop.

Question 16. Write the command to generate a single timing report for setup for each path group which has a violation.

```
pt_shell> report_timing -group [get_path_group *]
-slack_lesser_than 0
# When using PrimeTime interactively - abbreviate
# command names or switches by typing enough letters to
# distinguish from other commands or switches - or,
# better yet, use command expansion by pressing tab
pt_shell> report_timing -slack_less 0
```

Question 17. What are the names of the two path groups that have violating timing paths in **ORCA** (the answer will come from the result of the previous question?

The two path groups are **PCI_CLK** and **SYS_CLK**.

Question 18. Write the command to generate a timing report with the worst slack for setup to any output port constrained by the clock **PCI_CLK**.

```
pt_shell> help all_*
pt_shell> all_outputs -help
pt_shell> report_timing -to [all_outputs -clock PCI_CLK]
# Or, another way to do the same thing
pt_shell> report_timing -to [all_outputs] -group PCI_CLK
```

Question 19. There are a few latches in **ORCA**; write the command to identify the data pins of these latches.

```
pt_shell> all_registers -level_sensitive -data_pins
```

Question 20. Write the command to generate a timing report for hold to the D pin of the latched_clk_en_reg latches.

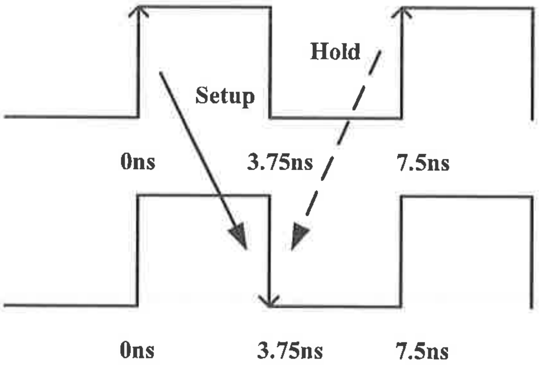
```
# Use copy and paste to avoid mistyping the long end point pin name
pt_shell> report_timing -delay min \
                -to I_ORCA_TOP/I_BLENDER*/latched_clk_en_reg/D
```

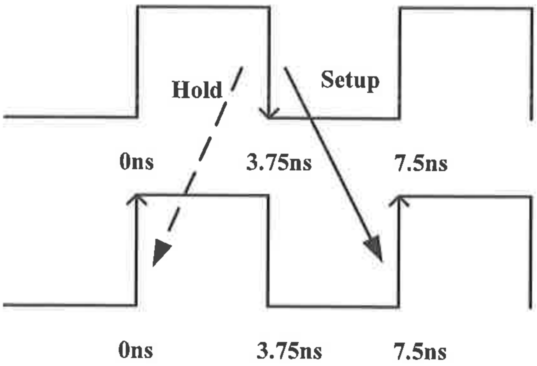
Question 21. Given that the first number under the waveform column is the first rising edge for the clock **`SDRAM_CLK`** and the second number is the falling edge – what duty cycle has been defined for this clock?

The rising edge of **`SDRAM_CLK`** is at 0ns, the falling edge at 3.75ns and the period is 7.50ns. The duty cycle is 50%.

Question 22. Describe the specific clock edges that will be used in a timing report for setup for a timing path constrained by the rising edge of **`SDRAM_CLK`** to the falling edge of **`SDRAM_CLK`**.

Use the following clock waveform for this and the next question. The clock edges will be 0ns to 3.75ns.





Question 23. For this same timing path, describe the specific clock edges that will be used in a timing report for hold timing checks.

The clock edges will be 7.5ns to 3.75ns.

```
# Commands for the final task
# The backslash is a line continuation character
# The switch -delay min_max will generate one report for setup and
# one for hold
report_timing -rise_from [get_clocks SDRAM_CLK] \
             -fall_to [get_clocks SDRAM_CLK] -delay_type min_max
report_timing -fall_from [get_clocks SDRAM_CLK] \
 -rise_to [get_clocks SDRAM_CLK] -delay_type min_max
```

Question 24. Which switch is useful for generating the worst 10 timing reports for each of these half clock cycle timing paths?

The switch **`-max_paths 10`** to the above command.

Question 25. Why does PrimeTime report "no constrained paths?"

The **`-max_paths`** option implicitly sets another option: **`slack_lesser_than 0`**, which, because slack is positive, results in no constrained paths.

Question 26. What additional option must you use to report the worst 10 timing paths?

Since there are no violating paths that could be reported in this case, add the option: **`slack_less_than 100`**

This page is left blank intentionally.

4

Constraining Multiple Clocks

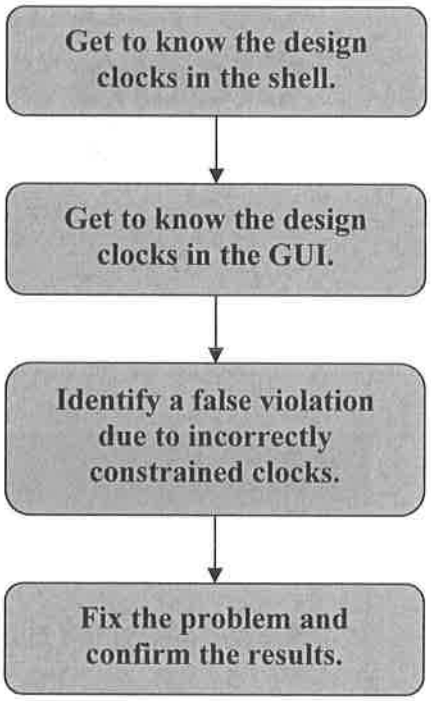
Learning Objectives

After completing this lab, you should be able to:

- Apply the commands taught in lecture to gather information about the design clocks
- Use the GUI for another view of the design clocks and their relationships

Lab Duration:
45 minutes

Overview


Get to know the design clocks in the shell.
Get to know the design clocks in the GUI.
Identify a false violation due to incorrectly constrained clocks.
Fix the problem and confirm the results.

Relevant Files and Directories

All files for this lab are located in the **lab4_clocks** directory under your home directory.

lab4_clocks/	Current working directory
orca_savesession/	Initial Saved ORCA session
orca_savesession_violations/	Saved ORCA session with an issue
RUN.tcl	Run script for ORCA
.synopsys_pt.setup	PT setup filef
scripts/	
orca_pt_variables.tcl	Variable script

Instructions

Task 1. Get to Know the Design Clocks

1. Make sure your current directory is `lab4_clocks`

2. Invoke PrimeTime (pt_shell).

   Restore the session saved in `./orca_savesession`

   Take advantage of command and file name completion by typing a few letters and then using the tab key.

3. Use the commands taught in lecture to answer the following questions.

   Use the job aid labeled "Clocks and More" for help recalling the specific commands.

Question 1. How many clocks are in this design and how many of these are generated?

....

Question 2. Which input ports have defined, master clocks?

....

Question 3. Which output ports have defined, outgoing clocks?

....

Question 4. Are the clocks propagated or ideal?

....

Question 5. Which 3 clock pairs have constrained timing paths?

....

Task 2. Use the GUI to Report Clock Relationships

If your design has many clocks, the GUI may simplify the task of understanding how the clocks are related.

1. Start the GUI by executing the following command.

```
pt_shell> start_gui
```

Note: The original pt_shell session is still running in the terminal window. You can keep the GUI open and use either the shell or the GUI interface as appropriate to the desired tasks.

2. Look at clock domain crossings: Open the "clock domain matrix" from the pull-down menu: **Clock → Clock Analyzer.**

The *ClockAnalyzer* window that opens (expand if needed by clicking on the plus signs to the left of the clocks) should match the information from **`check_timing`** when reporting the clock crossings in the design. Mouse over the blocks in the matrix to see information on what type of false paths exist. It is sometimes easier to digest this information as a graphical matrix table in comparison to the text output from **`check_timing -override clock_crossing -verbose`**.

The left part of the window lists each master clock and any generated clocks that are created from each master clock.

Question 6. What is the master clock for **`SYS_2x_CLK`**?

....

Question 7. **`SYS_2x_CLK`** is defined on which pin/port (its "source")? (note: you may have to drag the clock matrix out of the way, exposing more columns of information about the clocks)

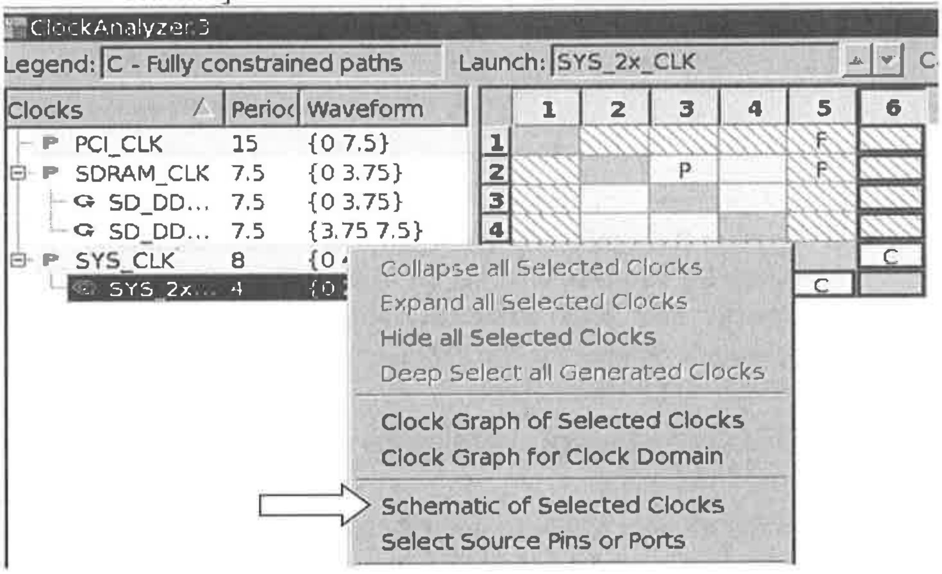
....

Question 8. The master clock for **`SYS_2x_CLK`** is defined on which pin/port?

....

3. Explore in more detail by displaying the clock schematic for SYS_2x_CLK: select the clock, then right mouse button->`Schematic of Selected Clocks`. Expand the fanin for the schematic for the MUX called I_CLOCK_GEN/U20 [Hint: To locate/highlight U20, use Select -> By Name] by double-clicking the input stubs, as shown in the following screen captures. Continue the double clicks until the fanin is exhausted [Example: an input port

has been reached]

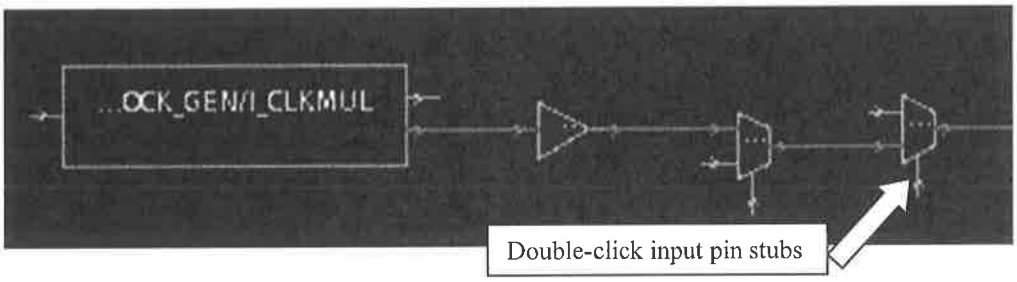


ClockAnalyzer.3

Legend: C - Fully constrained paths   Launch: SYS_2x_CLK

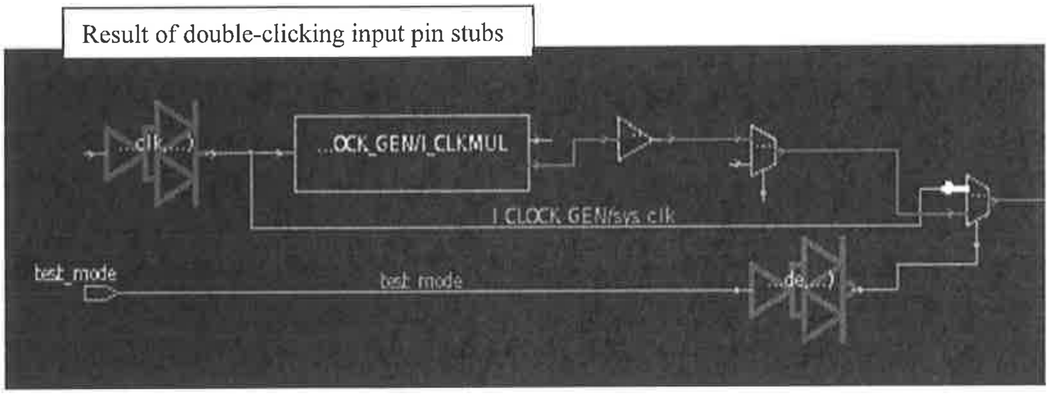
Clocks	Perioc	Waveform
PCI_CLK	15	{0 7.5}
SDRAM_CLK	7.5	{0 3.75}
SD_DD...	7.5	{0 3.75}
SD_DD...	7.5	{3.75 7.5}
SYS_CLK	8	{0
SYS_2x...	4	{0

1
2
3
4
5
6
F
P
F
C
C
Collapse all Selected Clocks
Expand all Selected Clocks
Hide all Selected Clocks
Deep Select all Generated Clocks
Clock Graph of Selected Clocks
Clock Graph for Clock Domain
Schematic of Selected Clocks
Select Source Pins or Ports



Double-click input pin stubs

Result of double-clicking input pin stubs



Question 9. What port is connected to the select pin of the MUX I_CLOCK_GEN/U20?

....

Question 10. Does seeing the schematic give you insight into the clocking scheme for test?

....

4. Explore clock relationships with the abstract clock graph: Close the schematic window, then, on the TopLevel window, select `Clock-> Clock Graph for All Clocks.` If necessary, display a toolbar next to the schematic by pressing the F8 key. Display various elements by checking the toolbar and pressing Apply.

5. Find a pair of muxed clocks: In the Abstract Clock Graph toolbar, select Mux and click Apply.

6. In the Abstract Clock Graph, find instance I_CLOCK_GEN/U10 of mx02d1. [Hint: To locate/highlight U10, use Select -> By Name]

Question 11. What clocks drive I_CLOCK_GEN/U10?

....

7. From the clock graph window, 'zoom into' an interesting object by displaying a schematic for it: Select I_CLOCK_GEN/U10, then **Schematic ➔ Schematic View**.

Question 12. What port drives the select line to I_CLOCK_GEN/U10?

....

8. Go back to the Abstract Clock Graph.

Question 13. From the abstract clock graph window, is it possible to open and display the same clock schematic for SYS_2x_CLK you displayed in the clock analyzer [Right Click on SYS_2x_CLK and find the option]?

....

9. Close the *Clock Analyzer* window by clicking on the small "**X**" in its upper right corner.

10. Close the *Clock Schematic* and *Clock Analyzer* windows by clicking on the small "**X**" in the upper right corner.

Task 3. Use the GUI to explore detail of timing paths

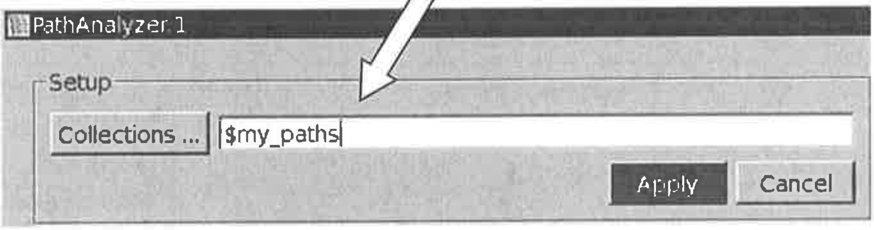
Investigate paths between launch and capture clocks – in this case, you will look at network latency for the launch and capture paths clocked by SYS_CLK.

1. Propagate all the clocks to have the clock network delays calculated by Primetime before examining paths, by executing these commands in the shell, which remains open behind the GUI (this will take a minute or so to complete). Tell PrimeTime to save the arrival times for all pins (this is what you will examine). Then, define a collection of timing paths to examine.

```
set_propagated_clock [all_clocks]
set timing_save_pin_arrival_and_slack true
update_timing
set my_paths [get_timing_paths -max 10 -group SYS_CLK
-path full_clock_expanded]
```

2. Enter your collection of violating paths from the pull-down menu **Timing → Path Analyzer**.

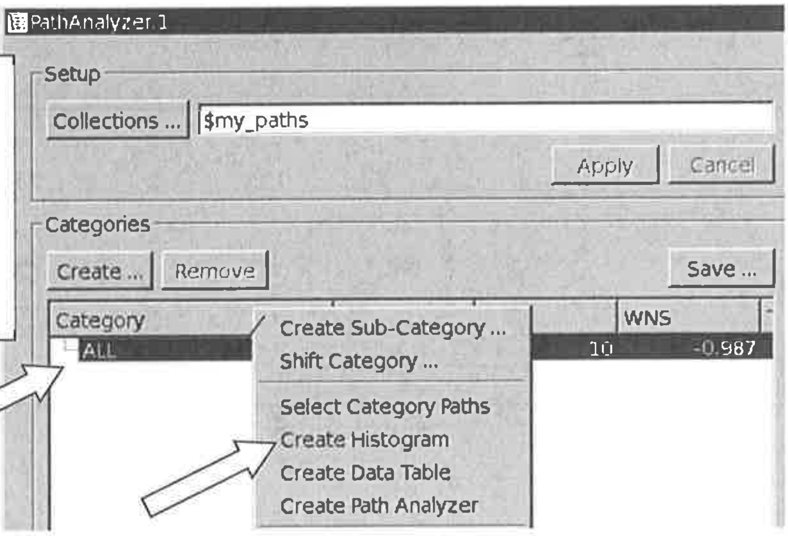
Enter your collection of timing paths and click `Apply`



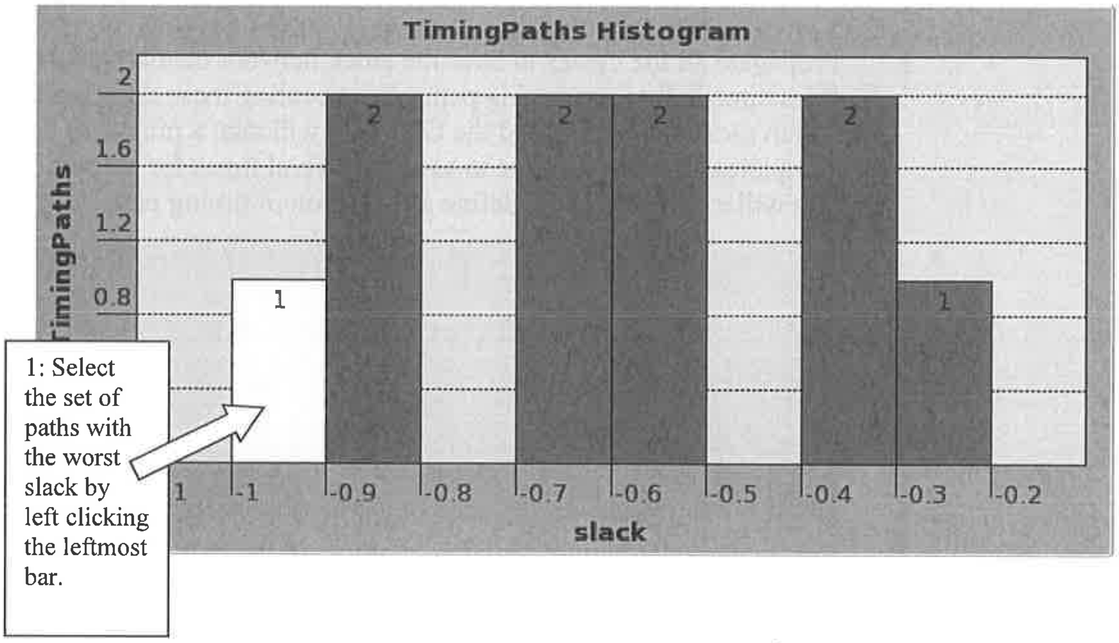
$my_paths

3. Bring up a histogram of your ten timing paths.

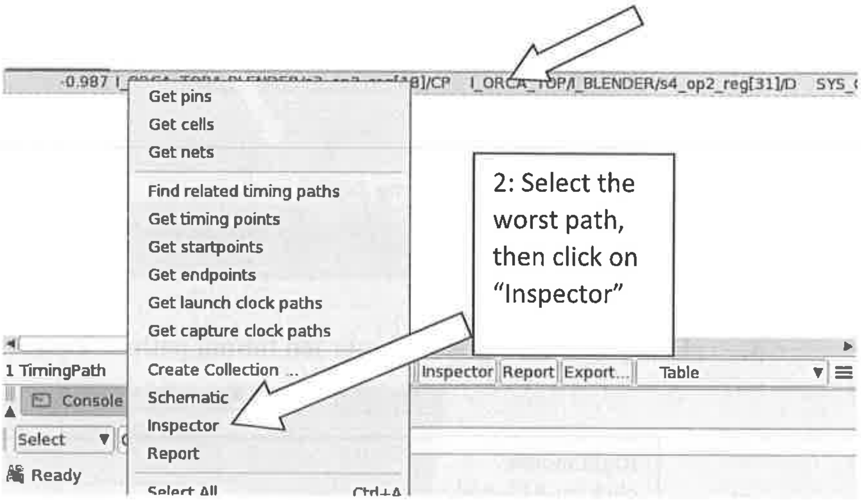
Right mouse click on ALL and select Create Histogram, then select OK on the resulting TableHistogram dialog box.



4. From the histogram, bring up the Path Inspector on a selected path.

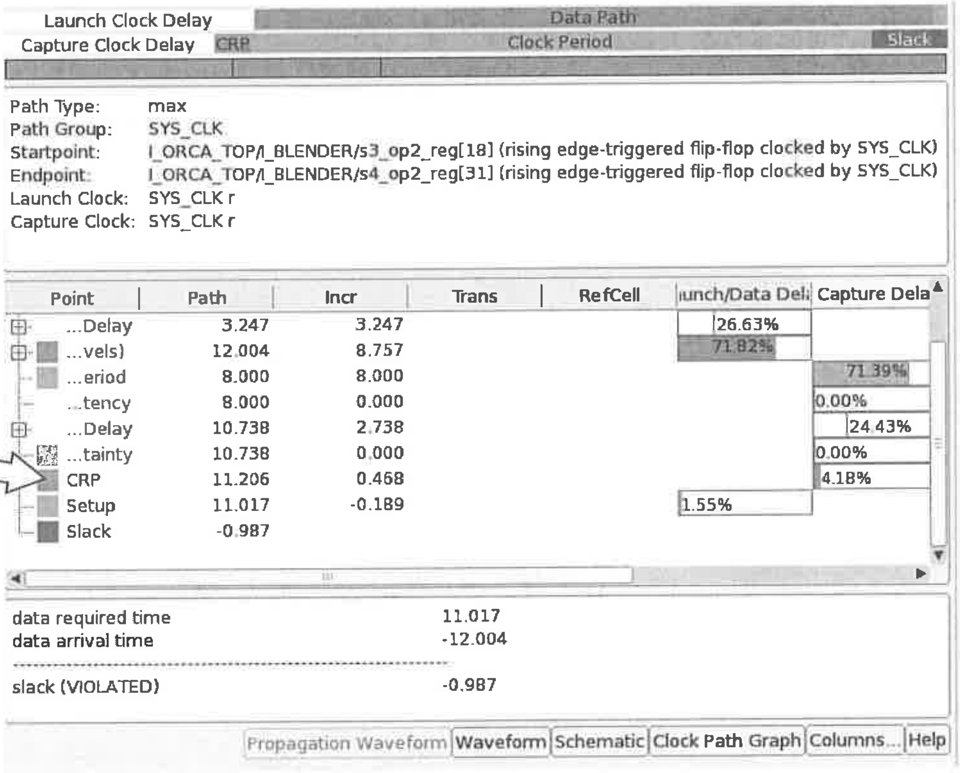


1: Select the set of paths with the worst slack by left clicking the leftmost bar.



2: Select the worst path, then click on "Inspector"

5. In the Path Inspector, examine clock reconvergent pessimism: In the data required and data arrival section, scroll down until you find CRP. Then, scroll across until you find the percent of delay for the CRP.



```
Path Type:      max
Path Group:     SYS_CLK
Startpoint:     I_ORCA_TOP/I_BLENDER/s3_op2_reg[18] (rising edge-triggered flip-flop clocked by SYS_CLK)
Endpoint:       I_ORCA_TOP/I_BLENDER/s4_op2_reg[31] (rising edge-triggered flip-flop clocked by SYS_CLK)
Launch Clock:   SYS_CLK r
Capture Clock:  SYS_CLK r
```

Point	Path	Incr	Trans	RefCell	unch/Data Del	Capture Dela
...Delay	3.247	3.247			26.63%	
...vels)	12.004	8.757			71.82%	
...eriod	8.000	8.000				71.39%
...tency	8.000	0.000				0.00%
...Delay	10.738	2.738				24.43%
...tainty	10.738	0.000				0.00%
CRP	11.206	0.468				4.18%
Setup	11.017	-0.189			1.55%	
Slack	-0.987					

```
data required time                                  11.017
data arrival time                                  -12.004
------------------------------------------------------------
slack (VIOLATED)                                    -0.987
```

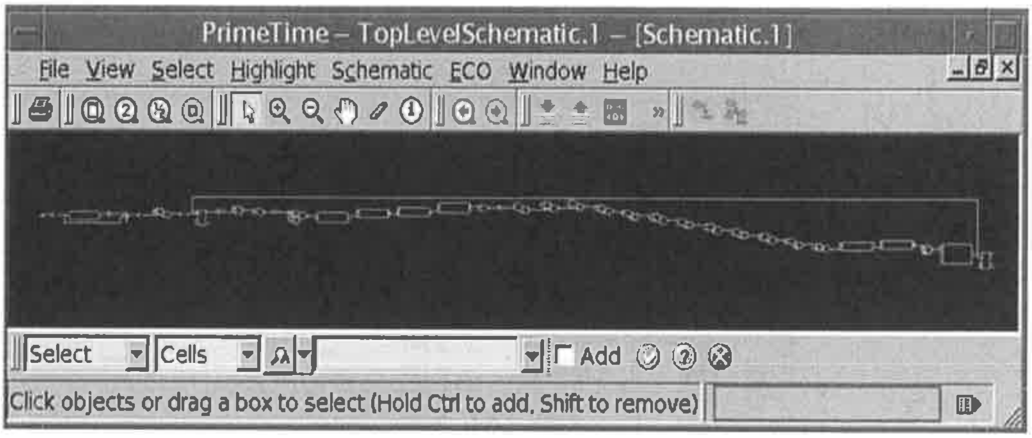
Question 14. What percent of the capture delay comes from CRP?

....

Question 15. Is this percent representative of all designs?

....

6. Look at a schematic of the path by clicking on the `Schematic` tab on the bottom of the path inspector window.

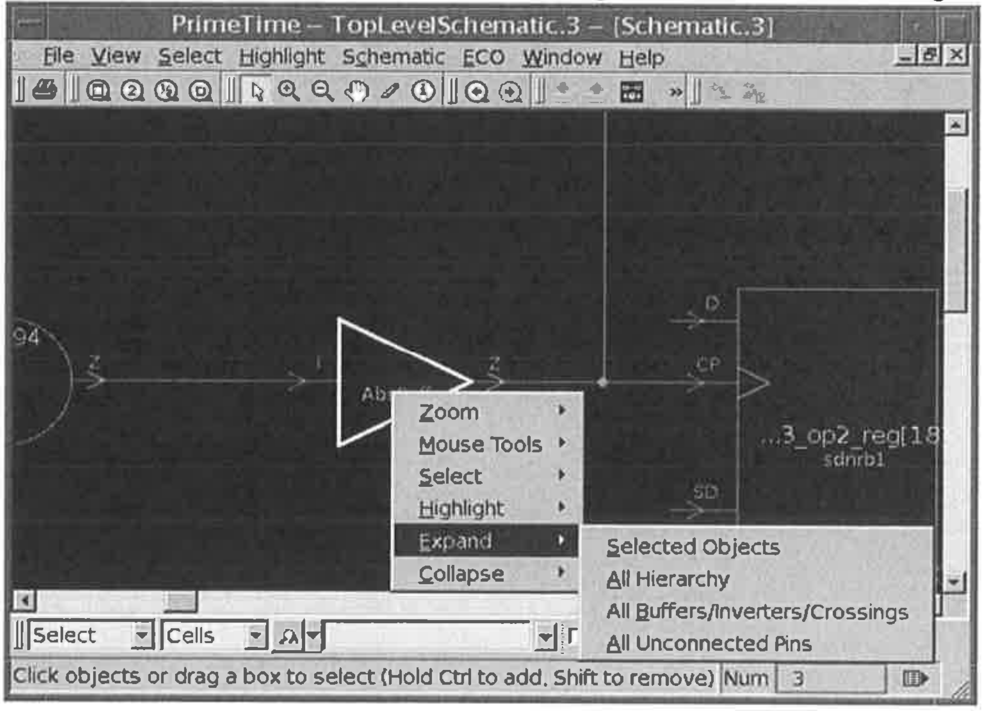

PrimeTime – TopLevelSchematic.1 – [Schematic.1]
File View Select Highlight Schematic ECO Window Help
Select
Cells
Add

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)

7. In the schematic window, find the CRP (clock reconvergent pessimism) point. This is the last pin before the launch and capture paths diverge.

Note: Mouse “gestures” or “strokes” are available for easier zooming: While pressing the middle mouse button drag the cursor vertically for ‘zoom full’; Drag diagonally up across an object to zoom in, and down across an object to zoom out.

8. To see arrival times on this pin, if necessary, you may have to first ‘expand’ the pin’s buffer. (By default, PrimeTime ‘collapses’ buffer trees into a single



Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)

9. View the arrival times (and any other attributes of interest) by selecting the output pin of the buffer just before the register, then by selecting View->Property

Question 16. How wide is the arrival window for thee buffer output pin?

....

Question 17. Does this match what we saw earlier in the data arrival data required section of the path inspector?

....

10. Examine the path waveform: Click on the Waveform tab at the bottom of the Path Inspector window.

Question 18. What can you add to the wavforms by clicking the right mouse button in the waveform window?

....

11. Close the GUI while keeping the original pt_shell session going in the terminal window:

```
File → Close GUI (in the main GUI window)
Or
pt_shell> stop_gui (in the pt_shell window)
```

12. Exit PrimeTime.

Task 4. Report a False Violation

1. Bring up PrimeTime and restore the saved session `orca_savesession_violations`

2. Determine the number and type of timing violations in **ORCA**:

```
report_analysis_coverage
```

Question 19. How many, and what kind of violations does **ORCA** have?

....

3. Generate a "short" timing report for the worst slack for an **out_setup** timing check.

Question 20. How will you identify the endpoint port which has the worst slack for **out_setup** (use the job aid labeled "Timing Reports" for help recalling the two appropriate switches)?

....

Question 21. Which clocks (launch and capture) are involved in this violation?

....

From task 1, you know that SD_DDR_CLK is a generated clock defined at an output port. The purpose of defining outgoing clocks is that PrimeTime calculates source latency for this clock and include this latency as part of the data required time.

4. Look at the data required time section of the timing report from the last step and notice that no clock latency is reported.

Confirm this with the following command:

```
# This report will return nothing as PrimeTime has not
# calculated source latency for SD_DDR_CLK
pt_shell> report_clock -skew SD_DDR_CLK
```

Question 22. Why has PrimeTime not calculated source latency for the outgoing clock **`SD_DDR_CLK`**?

....

After speaking with the designer, it turns out there was a miscommunication. The designer was expecting you to turn on a variable that will propagate all clocks!

5. There is a variable that can be used to make all clocks propagated. Use the Tcl procedure `aa` to help you identify the appropriate variable:

```
aa propagate
```

Question 23. What is the name of this variable?

....

Question 24. Using a man page, explain what this variable will do?

....

6. Use the man page for `check_timing` to find the name of the additional check that will flag all ideal clocks.

The following command opens the man page in a pop-up window with a scroll bar that simplifies viewing long reports.

```
pt_shell> vman check_timing
```

The above command is an alias created in the .synopsys_pt.setup file. It uses a command called view that is available on SolvNet, Doc Id 014947.

The alias vman will not work if the "wish" executable, the main executable in the Tk package, is not installed and made available in your lab environment

Question 25. How will you modify **`check_timing`** to add a check to validate that all clocks are propagated?

....

7. Quit *PrimeTime*.

Task 5. Re-Execute the Run Script to reduce violation

1. You are provided with the file **./scripts/orca_pt_variables.tcl** that will accomplish the following two things.

   - Adds to the default checks performed by **check_timing** the check that will flag ideal clocks.
   - All created clocks will be created as propagated clocks.

2. Execute the run script **./RUN.tcl** from the **lab4_clocks** Unix directory

   Log the results to the log file run.log.

```
unix> pt_shell -f ./RUN.tcl | tee -i run.log
```

3. Invoke PrimeTime and restore the newly saved session in the Unix directory **./orca _savesession**

4. Use the appropriate commands to confirm the information below:

   - The **out_setup** violations have been reduced.
   - All clocks are propagated.
   - Execute **check_timing** to confirm it is performing its default checks in addition to the check for ideal clocks.
   - The source latency is now calculated for **SD_DDR_CLK.**
   - The timing report to **sd_DQ[3]** includes this calculated source latency.

   There will be additional violations (more setup violations as well as out_hold violations) that you can ignore.

5. Quit PrimeTime.

This completes lab 4. Return to lecture.

Answers / Solutions

Question 1. How many clocks are in this design and how many of these are generated?

This information can be gathered from **`report_clock`**, or using the following commands.

```
pt_shell> sizeof_collection [all_clocks]
6
pt_shell> sizeof_collection [get_generated_clocks *]
3
```

Question 2. Which input ports have defined, master clocks?

```
pt_shell> rpt_clock_ports
Port Name      Direction    Clock Name      Is Generated
----------------------------------------------------------
  pclk            in          PCI_CLK          false
  sys_clk         in          SYS_CLK          false
  sdr_clk         in          SDRAM_CLK        false
  sd_CK           out         SD_DDR_CLK       true
  sd_CKn          out         SD_DDR_CLKn      true
```

Question 3. Which output ports have defined, outgoing clocks?

From the same report, **`sd_CK`** and **`sd_CKn`**.

Question 4. Are the clocks propagated or ideal?

Use **`report_clock`** to see that all the design clocks are ideal.

Question 5. Which 3 clock pairs have constrained timing paths?

```
pt_shell> check_timing -over clock_crossing -verbose
Information: Checking 'clock_crossing'.
Information: There are 4 clocks having domains interacting.

*                       all paths are false paths
#                       part of paths are false paths

From Clock              Crossing Clocks
------------------------------------------------------------------
PCI_CLK                 SYS_CLK*
SDRAM_CLK               SD_DDR_CLK#, SYS_CLK*
SYS_2x_CLK              SDRAM_CLK*, SYS_CLK
SYS_CLK                 PCI_CLK*, SDRAM_CLK*, SYS_2x_CLK
```

Question 6. What is the master clock for **SYS_2x_CLK**?

SYS_CLK

Question 7. SYS_2x_CLK is defined on which pin/port (its "source")?

I_CLOCK_GEN/I_CLKMUL/CLK_2X (You may have to drag the window containing the matrix out of the way in order to see the source pins)

Question 8. The master clock for SYS_2x_CLK is defined on which pin/port?

sys_clk

Question 9. What port is connected to the select pin of the MUX I_CLOCK_GEN/U20?

test_mode

Question 10. Does seeing the schematic give you insight into the clocking scheme for test?

Yes – two clocks come into the mux: one from the clock generator, one directly from the port. The test_mode port controls the select line, making it possible to bypass the clock generator during test mode and letting the design be driven directly from the port.

Question 11. What clocks drive I_CLOCK_GEN/U10?

SYS_CLK and SYS_2x_CLK

Question 12. What port drives the select line to I_CLOCK_GEN/U10?

power_save

Question 13. From the abstract clock graph window, is it possible to display the same clock schematic you displayed in the clock analyzer?

Yes. Select the clock SYS_2x_CLK (you may have to zoom in to select just the clock), then press right mouse button and select `Schematic for Selected Clocks`. You may have to expand input or output stubs (by double clicking on them) to get the exact same schematic.

Question 14. What percent of the capture delay comes from CRP?
4.18%

Question 15. Is this percent representative of all designs?
No, this number is dependent on the particular design and on the particular path.

Question 16. How wide is the arrival window for the buffer output pin?
It is 2.86939 minus 2.40107, or about .468.

Question 17. Does this match what we saw earlier in the data arrival data required section of the path inspector?
Yes.

Question 18. What can you add to the waveforms by clicking the right mouse button in the waveform window?
You can add input pins and output pins, allowing you to see the detail for the whole path.

Question 19. How many, and what kind of violations does **ORCA** have?

There are 23 *setup* violations, 53 *hold* violations, and 32 *out_setup* violations.

Question 20. How will you identify the endpoint port which has the worst slack for out_setup?

```
pt_shell> report_timing -path short -to [all_outputs]
  OR
pt_shell> page_on
pt_shell> report_analysis_coverage -status violated -check out_setup
Type of Check       Total          Met          Violated         Untested
```

```
------------------------------------------------------------------------
out_setup             75        43  ( 57%)        32  ( 43%)          0  (   0%)
------------------------------------------------------------------------
All Checks            75        43  ( 57%)        32  ( 43%)          0  (   0%)

Constrained                          Related    Check
Pin                                  Pin        Type
Slack
--------------------------------------------------------------------------
sd_DQ[6]                             SD_DDR_CLK   out_setup          -1.4006
sd_DQ[5]                             SD_DDR_CLK   out_setup          -1.4006
sd_DQ[4]                             SD_DDR_CLK   out_setup          -1.4006
```

Question 21. Which clocks (launch and capture) are involved in this violation?

```
pt_shell> report_timing -to sd_DQ[6] -path short
****************************************

  Startpoint: sdr_clk (clock source 'SDRAM_CLK')
  Endpoint: sd_DQ[6] (output port clocked by SD_DDR_CLK)
  Path Group: SD_DDR_CLK
  Path Type: max
  Min Clock Paths Derating Factor : 0.9000

  Point                                        Incr       Path
  --------------------------------------------------------------
  clock SDRAM_CLK (fall edge)                3.7500     3.7500
  clock source latency                       0.0000     3.7500
  sdr_clk (in)                               0.0000     3.7500 f
  ...
  sd_DQ[6] (inout)                           4.4006     8.1506 f
  data arrival time                                     8.1506

  clock SD_DDR_CLK (rise edge)               7.5000     7.5000
  clock reconvergence pessimism              0.0000     7.5000
  output external delay                     -0.7500     6.7500
  data required time                                    6.7500
  --------------------------------------------------------------
  data required time                                    6.7500
  data arrival time                                    -8.1506
  --------------------------------------------------------------
  slack (VIOLATED)                                     -1.4006
```

Question 22. Why has PrimeTime not calculated source latency for the outgoing clock **`SD_DDR_CLK`**?

The clocks (specifically the master clock) must be propagated for *PrimeTime* to calculate the source latency for generated clocks. All clocks in this design are ideal.

Question 23. What is the name of this variable?

```
pt_shell> aa propagate
********* Commands **********
 remove_propagated_clock # Remove a propagated clock specification
 set_propagated_clock # Specify propagated clock latency
********* Variables **********
case_analysis_propagate_through_icg = "false"
timing_all_clocks_propagated = "false"
timing_clock_gating_propagate_enable = "false"
timing_propagate_interclock_uncertainty = "false"
timing_propagate_through_non_latch_d_pin_arcs = "false"
```

Question 24. Using a man page, explain what this variable will do?

All clocks created after this variable is set to true will be created as propagated clocks.

The clocks will be set to propagated in the next task.

Question 25. How will you modify **`check_timing`** to add a check to validate that all clocks are propagated?

The added check is named **`ideal_clocks`**. You can add this check to the variable `timing_check_defaults` using `lappend` such that it is executed automatically with `check_timing`.

This will be done in the next task.

```
# Answers for TASK 5 STEP 1
# Add the following to ./scripts/orca_pt_variables.tcl
lappend timing_check_defaults ideal_clocks
set timing_all_clocks_propagated true
```

```
# Answers for Task 5 Step 4
# Out_setup violations should be reduced.
pt_shell> report_analysis_coverage

# All clocks should be propagated
pt_shell> report_clock

# The command check_timing does not flag ideal clocks
pt_shell> check_timing
Information: Checking 'no_clock'.
Information: Checking 'no_input_delay'.
Information: Checking 'partial_input_delay'.
Information: Checking 'ideal_clocks'.
. . .
# The source latency is being calculated for SD_DDR_CLK
pt_shell> report_clock -skew SD_DDR_CLK

# The source latency is applied to the timing report to
SD_DQ[6]
pt_shell> report_timing -to sd_DQ[6] -path short
  Startpoint: sdr_clk (clock source 'SDRAM_CLK')
  Endpoint: sd_DQ[6] (output port clocked by SD_DDR_CLK)
  Last common pin: sdr_clk
  Path Group: SD_DDR_CLK
  Path Type: max
  Min Clock Paths Derating Factor : 0.9000

  Point                                    Incr       Path
  ---------------------------------------------------------------
  clock SDRAM_CLK (fall edge)            3.7500     3.7500
  clock source latency                   0.0000     3.7500
  sdr_clk (in)                           0.0000     3.7500 f
  ...
  sd_DQ[6] (inout)                       4.4006     8.1506 f
  data arrival time                                 8.1506

  clock SD_DDR_CLK (rise edge)           7.5000     7.5000
  clock network delay (propagated)       2.9149    10.4149
  clock reconvergence pessimism          0.0000    10.4149
  output external delay                 -0.7500     9.6649
  data required time                                9.6649
  ---------------------------------------------------------------
  data required time                                9.6649
  data arrival time                                -8.1506
  ---------------------------------------------------------------
  slack (MET)                                       1.5143
```

5 Additional Constraints

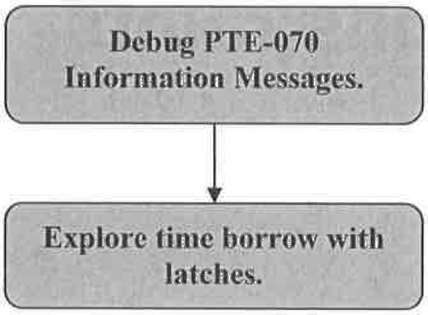
Learning Objectives

After completing this lab, you should be able to:

- Apply user specified annotated delays to explore time borrowing with latches
- Debug `PTE-070` messages regarding non-unate cells on the clock path

Lab Duration: 30 minutes

Overview



Debug PTE-070 Information Messages.

Explore time borrow with latches.

Relevant Files and Directories

All files for this lab are located in the `lab11_specific` directory under your home directory.

`lab5_additional/`	Current working directory
`orca_savesession/`	Saved session for ORCA
RUN.tcl	Script to create orca_savesession
`logs/`	Log files from run script
.synopsys_pt.setup	PT setup file

Instructions

Task 1. Debug PTE-070 Information Messages

1. Invoke PrimeTime from the **`lab5_additional`** Unix directory.

   Restore the session saved under **`./orca_savesession`**.

2. Shown below is the full message regarding a non-unate path on the clock network.

   In the next step, you will be asked to generate a timing report through this pin. In order to copy and paste and avoid typos – either find this message in the log file from another terminal window or use the Unix command **`grep`** from within PrimeTime as shown below.

```
# From ./logs/run.log
Information: A non-unate path in clock network detected.
Propagating both inverting and noninverting senses of clock
'SDRAM_CLK' from pin
'I_ORCA_TOP/I_SDRAM_IF/sd_mux_dq_out_0/Z'. (PTE-070)
```

```
pt_shell> sh grep -A 1 -B 1 PTE-070 logs/run.log
```

Note: The command **`sh`** (or alternatively **`exec`**) allows you to execute Unix commands from within the PrimeTime shell.

3. Generate a timing report for setup through the above pin and answer the following questions.

   The following alias has been created in the PrimeTime setup file and will generate a timing report in a pop-up window with a scroll bar using the view utility found on SolvNet, Doc Id 014947.

```
pt_shell> vrt -through <through pin>
```

Question 1. Which lines in the timing report did you use to validate it is for setup and the timing path start point is the source for the clock **`SDRAM_CLK`**?

....

Question 2. How does this timing report confirm that the pin in the warning above is on a data path (i.e. a clock source being used and constrained as a data path) and not on a clock path?

....

Question 3. Which sense is propagated through the above pin (i.e. positive unate or negative unate)? Look for a small arrow in the timing report which will locate the specific pin of interest.

....

4. Generate at least one additional timing report to show the use of a negative unate timing arc through the pin of interest.

Question 4. Which lines in the timing report did you use to validate it is for setup, the timing path start point is the source for the clock **SDRAM_CLK** and that the timing arc is negative unate for the pin of interest?

....

Question 5. Explain why this warning can be ignored (and suppressed) for these timing paths?

....

5. Do not quit PrimeTime.

Task 2. Explore Time Borrow and Latches

There is only one latch in this design.

1. Use the following commands to find it:

Take advantage of command and option completion with the tab key.

```
pt_shell> all_registers -level_sensitive
pt_shell> !! -clock_pin
pt_shell> all_registers -level_sensitive -data_pins
```

Question 6. What is the name of the clock pin for this latch?

....

Question 7. What are the names of the three data pins?

....

2. Generate a timing report starting at the latch for setup time (be specific by using the clock pin as the start point and not just the cell name!).

   This lab will refer to this timing report as "path segment #2".

   The function of this latch in the **ORCA** design is to generate a clock gating signal to turn on and off the clock `SYS_CLK`.

   Question 8. Describe how you know this latch is not experiencing time borrow from the previous stage?

   ....

3. Generate a timing report for the previous stage (this lab will refer to this timing report as "path segment #1").

   Use the `D` input pin of the latch as the end point of this timing path.

   Question 9. How much more time can path segment #1 take before it would start borrowing time from path segment #2?

   ....

4. Force path segment #1 to borrow time from path segment #2 by annotating a net delay of 4ns as shown below:

```
# Use cut and paste to avoid typos on the pin name
pt_shell> set_annotated_delay -net 4 \
          -to I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D
```

5. Generate the timing report for path segment #1 again (take advantage of the up and down arrows to scroll through the history event list).

   Question 10. How much time is path segment #1 borrowing from path segment #2?

   ....

   Question 11. What is the slack for path segment #1?

   ....

6. Re-generate the timing report for path segment #2. Before you can do that, you have to perform a full timing update!

```
pt_shell> update_timing -full
```

Note: The start point of the timing path will now be the **D** pin of the latch (not the clock pin as used before) because you are interested in reporting the timing path that includes time borrow.

```
pt_shell> report_timing -from \
             I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D
```

Question 12. Does the time given to path segment #1 now match your expectations?

....

7. Change the latch behavior for transparency; that is, make it transparent when data arrives between the opening and closing edges of the clock.

```
set_app_var timing_enable_through_paths true
```

8. Repeat your timing report to the latch D pin. Notice that, even though the latch is transparent, you can still specify the D pin as an endpoint.

```
report_timing -to \
"I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D"
```

Question 13. What is the startpoint?

....

9. Do a timing report FROM the startpoint you just identified.

```
report_timing -from \
I_ORCA_TOP/I_PARSER/blender_clk_en_reg/CP
```

Question 14. What is the path endpoint?.

....

Question 15. What are the transparency open and close edges?.

....

Question 16. Did data arrive between them?.

....

Question 17. Was there time borrowing?.

....

Question 18. Was slack positive?.

....

10. Quit PrimeTime.

This completes Lab 5. Return to lecture.

Answers / Solutions

Question 1. Which lines in the timing report did you use to validate it is for setup and the timing path start point is the source for the clock **SDRAM_CLK**?

```
pt_shell> report_timing -through I_ORCA_TOP/I_SDRAM_IF/sd_mux_dq_out_0/Z

  Startpoint: sdr_clk (clock source 'SDRAM_CLK')
  Endpoint: sd_DQ[0] (output port clocked by SD_DDR_CLK)
  Path Group: SD_DDR_CLK
  Path Type: max
```

Question 2. How does this timing report confirm that the pin in the warning above is on a data path (i.e. a clock source being used and constrained as a data path) and not on a clock path?

If no report was generated ("path is unconstrained"), this pin is on a clock path. Because a timing report was generated, this pin is on a data path.

Question 3. Which sense is propagated through the above pin (i.e. positive unate or negative unate)? Look for a small arrow in the timing report which will locate the specific pin of interest.

A positive unate timing arc (fall to fall) is reported through this pin.

```
# Shown are the relevant lines in the data path


I_ORCA_TOP/I_SDRAM_IF/buffd7G5B2I36/Z (buffd7)         0.1634 &   4.8748 f
I_ORCA_TOP/I_SDRAM_IF/sd_mux_dq_out_0/Z (mx02d4) <-    0.5431 &   5.4179 f
```

Question 4. Which lines in the timing report did you use to validate it is for setup, the timing path start point is the source for the clock **SDRAM_CLK** and that the timing arc is negative unate for the pin of interest?

```
pt_shell> pt_shell> report_timing \
              -rise_through    I_ORCA_TOP/I_SDRAM_IF/sd_mux_dq_out_0/Z
Startpoint: sdr_clk (clock source 'SDRAM_CLK')
Endpoint: sd_DQ[0] (output port clocked by SD_DDR_CLK)
Path Group: SD_DDR_CLK
Path Type: max
```

```
...

I_ORCA_TOP/I_SDRAM_IF/buffd7G5B2I36/Z (buffd7)          0.1634 &    4.8748 f
I_ORCA_TOP/I_SDRAM_IF/sd_mux_dq_out_0/Z (mx02d4) <-     0.4928 &    5.3676 r
```

Question 5. Explain why this message can be ignored for these timing paths?

The message indicates that both senses of the clock will be used when propagating the clock through this mux – this is the default behavior. However, because the clock is being used as data, PrimeTime actually propagates both senses (both positive and negative unate), even in older versions of PrimeTime. This is what is desired and therefore this information message can be ignored.

Question 6. What is the name of the clock pin for this latch?

```
pt_shell> all_registers -level_sensitive -clock_pins
{"I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/EN"}
```

Question 7. What are the names of the three data pins?

```
pt_shell> all_registers -level_sensitive -data_pins
{"I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D",
"I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/SC",
"I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/SD"}

# For step 2, generate a timing report for path segment 2
pt_shell> report_timing -from
I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/EN
```

Question 8. Describe how you know this latch is not experiencing time borrow from the previous stage?

If this latch were experiencing time borrow, there would be a line in the report stating the amount of time given to the start point (i.e. to the previous stage). This line is not present in this timing report.

```
# For step 3, generate a timing report for path segment 1
pt_shell> report_timing -to
I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D
```

Question 9. How much more time can path segment #1 take before it would have to start borrowing time from path segment #2?

It can take 3.465ns more before it would start borrowing time from path segment #2 (equivalent to the positive slack).

Question 10. How much time is path segment #1 borrowing from path segment #2?

Path segment #1 is borrowing 0.5306ns from path segment #2. This is noted in the data required time section of the timing report.

Question 11. What is the slack for path segment #1?

The slack is zero. PrimeTime borrows exactly as much as is needed to make the slack equal zero.

Question 12. Does the time given to path segment #1 match your expectations?

Yes – the time given to start point in the timing report for path segment #2 will match the time borrowed in the timing report for path segment #1.

```
pt_shell> report_timing -from I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D
pt_shell> report_timing -to I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D
```

Question 13. What is the startpoint?

I_ORCA_TOP/I_PARSER/blender_clk_en_reg/CP

Question 14. What is the path endpoint?

I_ORCA_TOP/I_BLENDER/U794/A (a gating check)

Question 15. What are the transparency open and close edges?

6.6454 and 10.4433

Question 16. Did data arrive between them?

yes – at time 7.1759

Question 17. Was there time borrowing?

no

Question 18. Was slack positive?.

yes

The timing report through the transparent latch is on the next page:

```
pt_shell> report_timing -from I_ORCA_TOP/I_PARSER/blender_clk_en_reg/CP
Startpoint: I_ORCA_TOP/I_PARSER/blender_clk_en_reg
              (rising edge-triggered flip-flop clocked by SYS_CLK)
  Endpoint: I_ORCA_TOP/I_BLENDER/U794
              (rising clock gating-check end-point clocked by SYS_CLK)
  Last common pin: I_CLK_SOURCE_SYS_CLK/Z
  Path Group: **clock_gating_default**
  Path Type: max
  Min Clock Paths Derating Factor : 0.9000

  Point                                                  Incr       Path
  ------------------------------------------------------------------------
  clock SYS_CLK (rise edge)                              0.0000     0.0000
  clock network delay (propagated)                       2.7139     2.7139
  I_ORCA_TOP/I_PARSER/blender_clk_en_reg/CP (sdcrb1)     0.0000     2.7139 r
  I_ORCA_TOP/I_PARSER/blender_clk_en_reg/Q (sdcrb1)      0.4621 &   3.1759 r
  I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D (slnlq1)     4.0000 *   7.1759 r
  ------------------------------------------------------------------------
   transparency window #1

  clock SYS_CLK (fall edge)                                         4.0000
  clock latency                                          2.3250     6.3250
  clock reconvergence pessimism                          0.3203     6.6454
  transparency open edge                                            6.6454

  clock SYS_CLK (rise edge)                                         8.0000
  clock latency                                          2.3596    10.3596
  clock reconvergence pessimism                          0.3203    10.6800
  library setup time                                    -0.2367    10.4433
  transparency close edge                                          10.4433

  available borrow at through pin                        3.2674
  ------------------------------------------------------------------------
  I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/D (slnlq1)     0.0000     7.1759 r
  I_ORCA_TOP/I_BLENDER/latched_clk_en_reg/Q (slnlq1)     0.3164 H   7.4924 r
  I_ORCA_TOP/I_BLENDER/U794/B1 (ora21d4)                 0.0263 &   7.5186 r
  data arrival time                                                 7.5186

  clock SYS_CLK (rise edge)                              8.0000     8.0000
  clock network delay (propagated)                       1.6425     9.6425
  clock reconvergence pessimism                          0.2160     9.8585
  I_ORCA_TOP/I_BLENDER/U794/A (ora21d4)                             9.8585 r
  clock gating setup time                               -0.2000     9.6585
  data required time                                                9.6585
  ------------------------------------------------------------------------
  data required time                                                9.6585
  data arrival time                                                -7.5186
  ------------------------------------------------------------------------
  slack (MET)                                                       2.1398
```

7 Path-Based Analysis

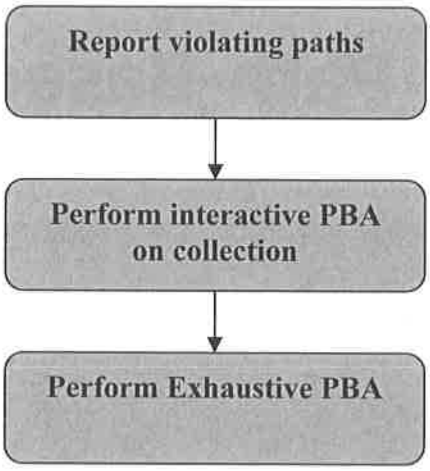
Learning Objectives

After completing this lab, you should be able to:

- Improve PrimeTime accuracy using
  - Path mode PBA
  - Exhaustive PBA

Lab Duration:
30 minutes

Introduction



Relevant Files and Directories

All files for this lab are located in the **`lab7_pba`** directory under your home directory.

`Lab7_pba/`	Current working directory
`.synopsys_pt.setup`	automatically-read PT setup file
`orca_savesession`	Saved session directory
RUN.tcl	Run script for orca_savesession

Instructions

Task 1. Report violating paths in the current design

In this task, we will report the worst violating slacks calculated during worst_slew propagation for the design.

1. Change into the working directory for this lab:

```
unix% cd lab7_pba
```

2. Bring up PrimeTime and restore a saved session called `orca_savesession`.

```
unix% pt_shell
pt_shell> restore_session orca_savesession/
```

Note: The **`orca_savesession`** can be recreated, if needed, using: **`pt_shell -f RUN.tcl | tee -i run.log`**

Note: Any *PARA-124 Errors* during the execution of RUN.tcl can be safely ignored for the purpose of our labs.

3. Identify the number of setup violations and identify the worst violating path to an endpoint.

```
pt_shell> report_analysis_coverage -check setup \
 -status violated
```

Question 1. How many setup checks are violated?

....

Question 2. Identify the size of the worst violating slack and the endpoint to which it is reported?

....

Task 2. Perform *path mode* PBA

In this task, we will interactively recalculate the slack for the worst violating path reported.

1. Determine the worst setup slack violation using PBA mode *path*

```
pt_shell> report_timing -to
I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D -pba_mode path
OR
pt_shell> set path [get_timing_path -to
I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D -pba_mode path]
pt_shell> report_timing $path
```

Question 3. How does `report_timing` indicate that the path is recalculated?

....

Question 4. How much is the worst slack violation after interactive path-based analysis `-pba_mode` path ?

....

Question 5. Did the PBA slack improve or become worse than the GBA slack or was there no change to the slack number? Are the results as expected?

....

Note: PBA does not change the PrimeTime database, so you cannot use `report_analysis_coverage`. In this lab, you can use either `report_timing` or `get_timing_paths` commands with the `-pba_mode` option.

Task 3. Performing Exhaustive PBA

In this task, we will run the design under exhaustive PBA mode.

1. Identify the top ten setup violations and the violating endpoints for the clock group "SYS_CLK" design.

```
pt_shell> redirect -tee GBA.rpt { report_timing  \
-path summary -group SYS_CLK -max_paths 10 -nosplit}
```

2. Next, run the exhaustive PBA on the setup violation reported for the design and generate a summary report :

```
pt_shell> redirect -tee PBA.rpt { report_timing  \
-path summary -group SYS_CLK  \
-max_paths 10 -nosplit  -pba_mode exhaustive }
```

3. Compare GBA.rpt to PBA.rpt and check :

Question 6. How many violating paths are returned by GBA and by PBA?

....

```
pt_shell> sizeof_collection [get_timing_paths \
    -slack_less 0 -group SYS_CLK -max_paths 10]
pt_shell> sizeof_collection [get_timing_paths \
    -slack_less 0 -group SYS_CLK -max_paths 10 \
    -pba_mode exhaustive]
```

Question 7. What is the value of worst violation after exhaustive PBA; what is the associated timing path endpoint name?

....

Question 8. Are there any UITE-480 warnings?

....

Task 4. [OPTIONAL Task] Generate GBA vs. PBA Summary Reports

In this optional task, we will generate the *global*, *qor* and *constraint* reports using GBA, PBA mode *path* and PBA mode *exhaustive* for comparison.

```
redirect -tee GBAsummary.rpt {
        report_global_timing
        report_qor
        report_constraint -all_violators
}

redirect -tee PBApath.rpt {
     report_global_timing - pba_mode path
     report_qor -pba_mode path
     report_constraint -all_violators - pba_mode path
}

 redirect -tee PBAexhaust.rpt {
  report_global_timing -pba_mode exhaustive
  report_qor -pba_mode exhaustive
  report_constraint -all_violators -pba_mode exhaustive
}
```

This completes Lab 7. End of Day-2.

Answers / Solutions

Question 1. How many setup checks are violated?
23

Question 2. Identify the size of the worst violating slack and the endpoint to which it is reported?

Slack to endpoint
I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D: -0.7737

Question 3. How does `report_timing` indicate that the path is recalculated?

The Path Type is followed by 'recalculated'. See the following example:

```
pt_shell> report_timing -to I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D -
pba_mode path
****************************************
Report : timing
        -path_type full
        -delay_type max
        -max_paths 1
        -pba_mode path
****************************************

  Startpoint: I_ORCA_TOP/I_BLENDER/s3_op2_reg[18]
              (rising edge-triggered flip-flop clocked by SYS_CLK)
  Endpoint: I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]
              (rising edge-triggered flip-flop clocked by SYS_CLK)
  Path Group: SYS_CLK
  Path Type: max  (recalculated)
```

Question 4. How much is the worst slack after interactive path-based analysis `-pba_mode` path?
-0.7711

Question 5. Did the PBA slack improve or become worse than the GBA slack or was there no change to the slack number? Are the results as expected?

The slack improved as expected.
(Remember that when a path is recalculated, the slack can only remain the same or improve.)

Question 6. How many violating paths are returned by GBA and by PBA?

By GBA : 9 violating paths

By PBA: 8 violating paths

Question 7. What is the value of worst violation before and after exhaustive PBA; what is the associated timing path endpoint name?

With PBA :
I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D
-0.7737

With PBA:
I_ORCA_TOP/I_BLENDER/s4_op2_reg[31]/D
-0.7711

Question 8. Are there any UITE-480 warnings?

No, there are no UITE-480 warnings – PBA exhaustive recalculation is complete.

8 SI Delay Analysis

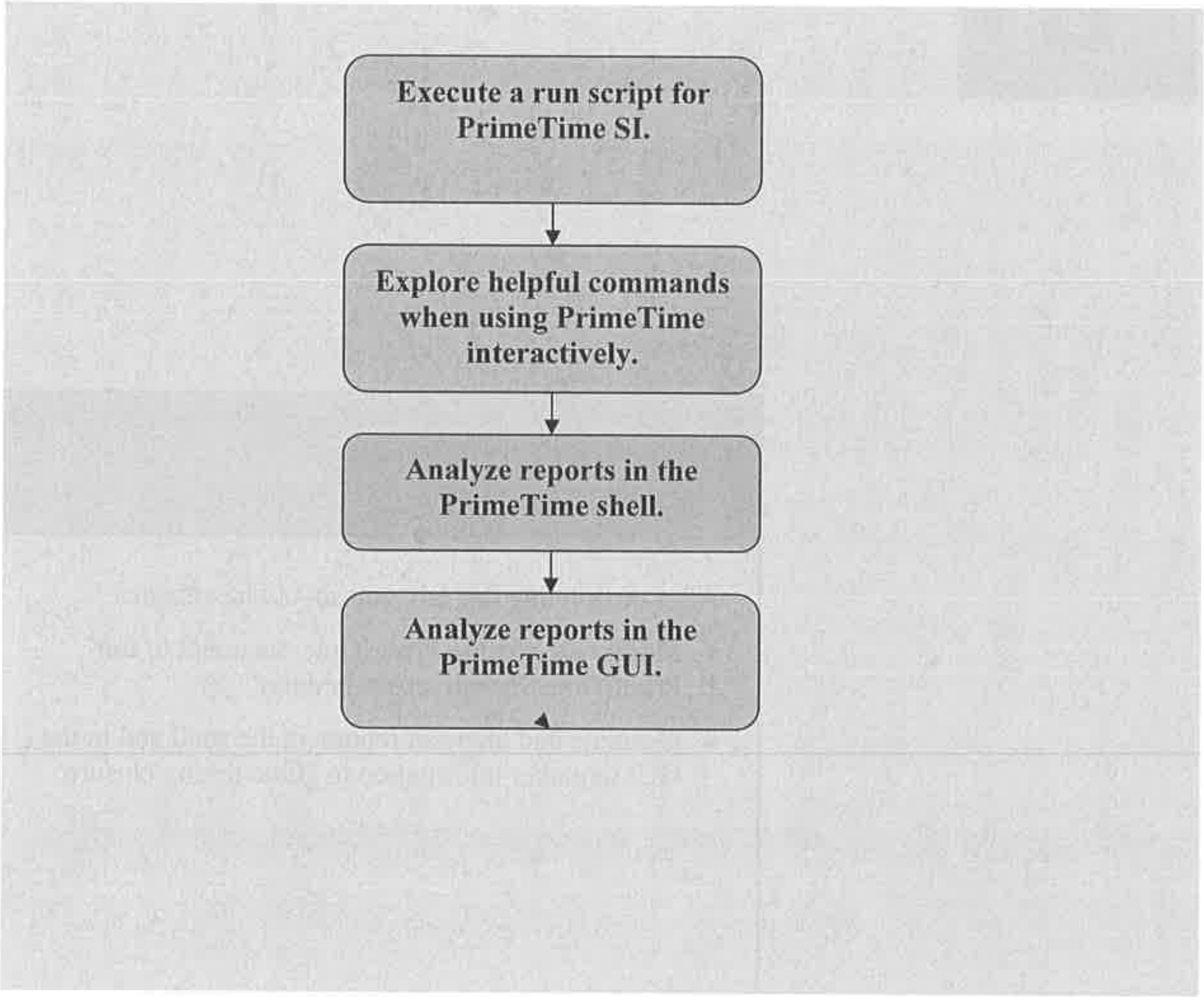
Learning Objectives

After completing this lab, you should be able to:

- Modify an existing PrimeTime run script to run PrimeTime SI with crosstalk delay.
- Generate and interpret reports in the shell and in the GUI to gather information to guide timing closure.

Lab Duration:
45 minutes

Overview


Execute a run script for PrimeTime SI.
Explore helpful commands when using PrimeTime interactively.
Analyze reports in the PrimeTime shell.
Analyze reports in the PrimeTime GUI.

Answers & Solutions

This lab guide contains answers and solutions to all questions. If you need some help with answering a question or would like to confirm your results, check the back portion of every lab.

File Locations

Relevant Files and Directories

`ref/`	
`db/`	Technology libraries
`design_data/`	Parasitics and design netlist
`scripts/`	Supporting scripts
`tcl_procs/`	Useful Tcl procedures
`lab8_SI_delay/`	
`RUN.tcl`	Run script for regular PrimeTime
`SI-RUN.tcl`	Run script for PrimeTime SI
`.synopsys_pt.setup`	Setup file
`max_func_savesession/`	Saved session of regular PrimeTime
`rpt/`	Reports generated during run script

Instructions

Task 1. Execute a Run Script for PrimeTime SI

1. Answer the following question.

   Question 1. List the two steps that must be included to modify a “regular” PrimeTime run script for PrimeTime SI?

   ....

2. Change directory to **lab8_si_delay**. Optionally, take a look at the run script for PrimeTime SI, **SI-RUN.tcl** and the associated files.

3. Execute PrimeTime SI.

```
unix% pt_shell -f SI-RUN.tcl | tee -i sirun.log
```

4. After the run script completes, invoke PrimeTime and restore the saved session created during the run script execution. The saved session directory name includes a time stamp such that multiple runs are saved in distinct Unix directories.

```
unix% pt_shell
```

Note: PrimeTime supports command, option, variable and file completion. Type a few letters and then hit the tab key.

```
pt_shell> restore_session simax_func_savesession_<time_stamp>
```

5. Using the Tcl procedure called **aa** (always ask), find all the commands/variables containing the letters “**si_**”.

```
pt_shell> aa si_
```

The above Tcl procedure is available on SolvNet, Doc Id 012959.

Task 2. Generate and Interpret SI Reports in the Shell

1. Report the total number of violations in this design. Answer the following questions.

   As a side note, regular PrimeTime passed timing. The new violations are all associated with crosstalk. If you are interested, the session was saved for the regular PrimeTime run under the Unix directory max_func_savesession.

Question 2. How many violations are in this design?

....

Question 3. How large is the worst slack for setup in this design?

....

2. Generate a timing report to the endpoint with the largest negative slack.

```
# Use copy and paste to avoid typos
pt_shell> report_timing -crosstalk -nets -transition \
                    -nosplit -to <endpoint>
```

Note: Optionally, use the *view* alias to open the timing report in a separate window with a scroll bar.

```
pt_shell> view report_timing -crosstalk -nets -transition \
                         -nosplit -to <endpoint>
```

The above Tcl procedure that is aliased by view is available on SolvNet, Doc Id 014947. The alias view will not work if the "wish" executable, the main executable in the Tk package, is not installed and made available in your lab environment.

Question 4. Identify the net with the largest delta delay on this path – how large is the worst delta delay?

....

Question 5. How is this delta delay represented (i.e. is it in time units or expressed as a ratio to VDD)?

....

Question 6. Is the delta delay positive or negative and does this match your expectation?

....

Question 7. What is the stage delay associated with this large delta delay?

....

Question 8. What is the meaning of the "&" symbol in this report?

....

3. Generate a timing report for hold to this same endpoint.

```
# Alternatively, use the up arrow for previous commands
pt_shell> !! -delay min
```

Question 9. Are the delta delays positive or negative and does this match your expectations?

....

Note the net delays along this path are negative. The stage delays however, which are the meaningful delay numbers to pay attention to, are positive.

4. Use the job aid to find and execute the appropriate command to identify victim nets with the largest delta delay. Answer the following questions.

Question 10. How large is the largest delta delay in this design?

....

Question 11. Are the nets reported as victims or aggressors?

....

Question 12. Are the reported nets associated with violating timing paths?

....

Question 13. Find the switch for this command that will allow clock nets with large delta delays to be included in this report (by default they are not) (hint – use a man page or help)?

....

```
# Use this alias to open a man page in a pop-up window!
pt_shell> vman report_si_bottleneck
```

Question 14. What is one solution to reduce the delta delay on these nets and thus either eliminate or reduce the timing violations in this design? (Hint – use the job aid or the man page for **report_si_bottleneck**)

....

5. Use the job aid to find and execute the appropriate command to identify victim nets with the largest ratio between delta delay and stage delay.

Question 15. What does the cost in this report represent (i.e. is it in time units or is it a ratio)?

....

Question 16. Are the nets reported as victims or aggressors?

....

Question 17. What is one solution to reduce the delta delay on these nets and thus either eliminate or reduce the timing violations in this design? (Hint – use the job aid or the man page for **report_si_bottleneck**)

....

6. Use the job aid to find and execute the appropriate command to identify the strongest aggressors in this design. Answer the following questions.

Question 18. What does the cost in this report represent (i.e. is it in time units or is it a ratio to VDD)?

....

Question 19. Are the reported nets reported as victims or aggressors?

....

Question 20. Are the aggressor nets themselves necessarily on violating timing paths?

....

Question 21. What is one solution to reduce the delta delays caused by these aggressor nets and either eliminate or reduce the timing violations in this design? (Hint – use the job aid or the man page for **report_si_bottleneck**)

....

7. Use the job aid to find and execute the appropriate command to report the worst case clock skew and the associated crosstalk contributions. Answer the following question.

Question 22. Does this report confirm that the clock nets are well shielded?

....

Task 3. Generate SI Reports in the GUI

With long timing reports, finding the largest delta delays can be a challenge. The GUI can simplify this task.

1. Start the GUI by executing the following command.

```
pt_shell> start_gui
```

2. From the "Crosstalk" pulldown menu, select "Delta Delay Histogram" and then click "ok" in the dialog box that appears.

3. In the histogram that appears, select the right most bin which will contain the worst delta delay(s) for setup.

4. Next, from the "Crosstalk" pulldown menu, select "Accumulated Bump Voltage Histogram" and then click "ok" in the dialog box that appears. This will show the victim nets and their accumulated bump voltages (from the best to the worst)

5. Explore any additional options of interest from the Crosstalk menu.

6. Close the GUI by either selecting **File → Close GUI** or by typing the command **stop_gui** in the shell.

Congratulations – this completes Lab-8!

Answers / Solutions

Question 1. List the two steps that must be included to modify a "regular" PrimeTime run script for PrimeTime SI?

```
# Set this variable at the beginning of the run script
set si_enable_analysis true
# Keep coupling caps when reading parasitics
read_parasitics -keep_coupling -format SBPF route_xtalk.sbpf
```

Question 2. How many violations are in this design?

```
pt_shell> report_analysis_coverage
```

There are 14 violations and they are all setup checks.

Question 3. How large is the worst slack for setup in this design?

```
pt_shell> report_analysis_coverage -status violated
Constrained                          Related   Check
Pin                                  Pin       Type               Slack
------------------------------------------------------------------------
s4_op1_reg_30_/D                     CP(rise)  setup            -0.1422
s4_op1_reg_23_/D                     CP(rise)  setup            -0.1080
```

Question 4. Identify the net with the largest delta delay on this path – how large is the worst delta delay?

The largest delta delay on this path is 0.1024 ns.

Question 5. How is this delta delay represented (i.e. is it in time units or expressed as a ratio to VDD)?

It is in time units. The time units for the main library is ns.

Question 6. Is the delta delay positive or negative and does this match your expectation?

The delta delay is positive. This makes sense because this report is for setup and along the data path the worst case slow down crosstalk effects should be considered.

Question 7. What is the stage delay associated with this large delta delay?

The stage delay is the cell + net delay which is 0.5522 + 0.1025 = 0.6547 ns. The ratio between delta delay and stage delay is approximately 15%.

Question 8. What is the meaning of the "**&**" symbol in this report?

This indicates that the delays are calculated using detailed net parasitic data. The following tables comes from the man page for **report_timing**.

```
Symbol          Annotation
------          ----------
H               Hybrid annotation
*               SDF back-annotation
&               RC network back-annotation
$               RC pi back-annotation
+               Lumped RC
<none>          Wire-load model or none
```

Question 9. Are the delta delays positive or negative and does this match your expectations?

The delta delays are now negative and this makes sense. This is a hold report and along the data path the worst case speed up crosstalk effects should be considered.

Question 10. How large is the largest delta delay in this design?

The largest delta delay on violating timing paths for setup and for hold in the design is 0.1738 ns. Use the following to gather this information.

```
pt_shell> report_si_bottleneck -cost_type delta_delay
```

Question 11. Are the nets reported as victims or aggressors?

These are victim nets.

Question 12. Are the reported nets associated with violating timing paths?

Yes, by default the switch **–slack_lesser_than** is set to zero. Therefore, the reported victim nets are all on violating timing paths for either setup or for hold.

Question 13. Find the switch for this command that will allow clock nets with large delta delays to be included in this report (by default they are not) (hint – use a man page or help)?

```
# Type the following, including the dash, followed by a tab

pt_shell> report_si_bottleneck -

cost_type            max                   min          nosplit
pre_commands         slack_lesser_than  include_clock_nets
max_nets         minimum_active_aggressors      post_commands
significant_digits
```

Question 14. What is one solution to reduce the delta delay on these nets and thus either eliminate or reduce the timing violations in this design? (Hint – use the job aid or the man page for **report_si_bottleneck**)

Reduce the coupling capacitance by separating the victim and aggressor nets using `set_coupling_separation`.

Question 15. What does the cost in this report represent (i.e. is it in time units or is it a ratio)?

The cost is a ratio between the delta and stage delay. Use the following to generate this report.

```
pt_shell> report_si_bottleneck -cost_type delta_delay_ratio
```

Question 16. Are the nets reported as victims or aggressors?

The nets are victims.

Question 17. What is one solution to reduce the delta delay on these nets and thus either eliminate or reduce the timing violations in this design? (Hint – use the job aid or the man page for **report_si_bottleneck**)

Large ratios are typically attributed to slow victim transitions (a weak victim). This can be fixed by upsizing the victim or inserting a buffer on the victim.

Question 18. What does the cost in this report represent (i.e. is it in time units or is it a ratio to VDD)?

The cost is a ratio to VDD – the sum of all switching bumps induced on all violating victim nets associated with that aggressor net. Use the following to generate this report.

```
pt_shell> report_si_bottleneck -cost_type delay_bump_per_aggressor
```

Question 19. Are the reported nets reported as victims or aggressors?

These are aggressor nets.

Question 20. Are the aggressor nets themselves necessarily on violating timing paths?

No. Only the victim nets must be on violating timing paths as defined by the switch **–slack_lesser_than**.

Question 21. What is one solution to reduce the delta delays caused by these aggressor nets and either eliminate or reduce the timing violations in this design? (Hint – use the job aid or the man page for **report_si_bottleneck**)

Downsize the strong aggressors or reduce the coupling capacitance by net isolation.

Question 22. Does this report confirm that the clock nets are well shielded?

Yes. The worst case clock skew has not been made worse by delta delay contributions. Use the following to generate this information.

```
pt_shell> report_clock_timing -type skew -crosstalk -verbose
```

9 SI Noise Analysis

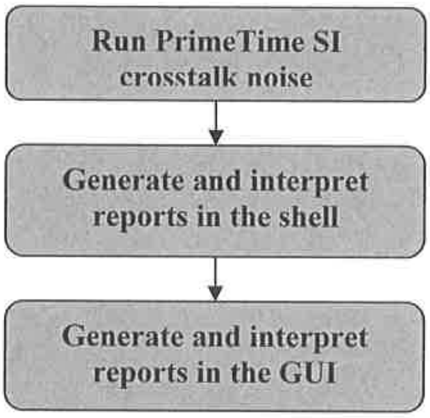
Learning Objectives

After completing this lab, you should be able to:

- Run PrimeTime SI for crosstalk noise.
- Generate and interpret the key noise reports in the shell and in the GUI.

Lab Duration:
45 minutes

Overview


Run PrimeTime SI crosstalk noise
Generate and interpret reports in the shell
Generate and interpret reports in the GUI

Relevant Files and Directories

`ref/`	
`db/`	Technology libraries
`design_data/`	Parasitics and design netlist
`scripts/`	Supporting scripts
`tcl_procs/`	Useful Tcl procedures
`lab9_si_noise/`	
`run_noise.pt`	Run script for PTSI noise
`.synopsys_pt.setup`	Setup file
`SE_PE_savesession/`	Saved session for run with noise
`noise.log`	Log file of SI run
`rpt/`	Reports generated during run script

Instructions

Task 1. Run PrimeTime SI for Noise Analysis

A run script for noise has already been executed and logged for you.

1. Change directory to **lab9_si_noise**.
2. Open the run script **run_noise.pt** and answer the following question.

   Question 1. List at least three steps in this run script that are important or relevant to noise analysis?

   ....

   Question 2. Check the appropriate boxes to indicate what is turned <u>on</u> by default during a noise analysis (controlled by the command **set_noise_parameters**)?

   □ Beyond rail analysis　□ Between-the-rails analysis

   □ Noise propagation　□ Timing windows are used to filter aggressors

3. Open the log file, **noise.log**, to answer the following question.

   Question 3. Which command initiated a <u>timing</u> update (look for message **UITE-214**)?

   ....

   Question 4. Does the timing update include analysis for crosstalk delays?

   ....

   Question 5. Which command initiated a <u>noise</u> update (look for messages **NOISE-010**)?

   ....

   Question 6. From lecture, what information is used from the <u>timing</u> update for noise analysis?

   ....

Task 2. Interpret Reports for Noise in the Shell

1. Invoke PrimeTime from the **lab9_si_noise** Unix directory and restore the session saved in the directory **SE_PE_savesession**.

   Take advantage of command and file completion with the tab key!

2. Execute the following commands to identify the worst slack for noise using the slack type "area".

```
pt_shell> report_noise -all_violators -slack_type area
pt_shell> report_noise -verbose -slack_type area
```

Question 7. Does this design contain any noise violations?

....

Question 8. What is the worst slack for noise (i.e. the smallest positive slack) in area units and describe the relevant region (above_low or below_high)?

....

Question 9. How many "effective" aggressor nets are listed for this "worst" victim net?

....

Question 10. Identify the following information for this victim net: the driving cell pin name, the load pin name, the net name?

....

Question 11. Describe the meaning of the term "Propagated" in this report and offer one reason why this row is empty (filled with dashes)?

....

3. Gather more information on this same net by executing the following. In this step, answer the following questions by referring to the top part of the this report which gives general information about the noise calculation.

```
# Use copy and paste from report_noise to avoid typos
pt_shell> report_noise_calculation -above -low -from <pin> -to <pin>
```

Question 12. Describe what is meant by the line "Steady state resistance source: library set iv curve"?

....

Question 13. Were any noise derating factors applied to this net and does this match your expectations from viewing the run script?

....

Question 14. Describe what is meant by "noise effort threshold"?

....

4. Answer the following questions by referring to the next portion of the same report which details the noise calculations.

Question 15. What command allows a user to exclude specific aggressor nets from analysis or define them as having an infinite timing window (refer to the job aid for help)?

....

5. Finally, answer the following questions by referring to the last section of the same report which details the noise slack calculations.

Question 16. Describe what is meant by the line "Constraint type: library immunity table"?

....

Question 17. What is the noise constraint (i.e. the required noise bump height in library voltage units)?

....

Question 18. The slack "area" is different from the noise bump "area" in the previous section. Can you describe how they are different?

....

6. View the worst 5 nets considering all the register data pins using slack type area_percent.

```
pt_shell> report_noise -data_pins -nworst 5 -slack_type area_percent
```

Question 19. Describe "how far off from the constraint" the worst flip-flop data pins are (i.e. lots of margin or little margin)?

....

Question 20. Describe what is meant by a slack of infinity?

....

7. View the noise on the clock pins of registers in this design.

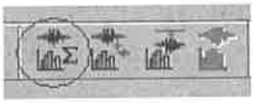
```
pt_shell> report_noise -clock_pins
```

Task 3. Run Noise Analysis in the GUI

Identifying nets with large bump heights may be of interest (independent of slack and constraint height). The GUI fascilitates this type of analysis.

1. Execute the following to open the GUI.

```
pt_shell> start_gui
```



2. Open the "Accumlated Noise Bump Histogram" by selecting the appropriate toolbar button. You can also select it using the *Noise* pulldown menu. Select "ok" in the dialog box that appears to select all the default settings.

   This histogram lists either nets or load pins (selected in the dialog box) ordered by bump height (not by slack) in the below_high region (selected in the dialog box).

   Note: The "Composite Noise Bump Histogram" will generate the same results for this design. The composite noise bump includes propagated noise whereas the accumlated noise bump does not.

3. Select the bin with the worst nets (the right most bar) to display the net names with the largest noise bumps by height.

   Question 21. How many nets have a noise bump height larger than 0.5?

   ....

   Note: A Tcl procedure is available on SolvNet (Doc Id 009418) to report the same information in the shell. This Tcl procedure is available in your lab setup. The following shows an example of using this Tcl procedure.

```
pt_shell> report_noise_bumps -bump_greater_than 0.5
```

4. Quit PrimeTime.

Congratulations! This concludes Lab-9

Answers / Solutions

Question 1. List at least three steps in this run script that are important or relevant to noise analysis?

Following are four steps that are relevant to noise.

- Turn on SI analysis with the variable **`si_enable_analysis`**. [step common to SI delay analysis]
- Use libraries that include characterization for noise.
- Apply noise derating factors to scale the calculated noise bump height and width.
- Read net parasitics and keep the coupling capacitances. [step common to SI delay analysis]
- Execute **`update_noise`** to initiate a noise update. You could also execute **`report_noise`** which will both initiate a noise update and report noise violations.

Question 2. Check the appropriate boxes to indicate what is turned on by default during a noise analysis (controlled by the command **set_noise_parameters**)?

□ Beyond rail analysis ☒ Between-the-rails analysis

□ Noise propagation ☒ Timing windows are used to filter aggressors

Question 3. Which command initiated a timing update (look for message **UITE-214**)?

The command **`check_timing`** initiated a timing update.

Question 4. Does the timing update include analysis for crosstalk delays?

Yes. This is apparent from the `XTALK-xxx` messages.

Question 5. Which command initiated a noise update?

The command **`update_noise`** initiated a noise update. These messages are turned on by the application variable **`si_noise_update_status_level`**.

Question 6. From lecture, what information is used from the timing update for noise analysis?

The arrival windows for the aggressor nets are used for filtering. Only aggressor nets that overlap are defined as active and will contribute to the calculated noise bump.

Question 7. Does this design contain any noise violations?

No, there are no noise violations in this design.

Question 8. What is the worst slack for noise in area units and describe the relevant region (above_low or below_high)?

The worst slack is 153.8006 for the region above_low.

Question 9. How many "effective" aggressor nets are listed for this "worst" victim net?

There is one effective aggressor net: n25422

Question 10. Identify the following information for this victim net: the driving cell pin name, the load pin name, the net name?

The victim net name is n25449, the driving cell pin name is T3_reg_190/QN and the load pin name is U11405/A1.

Question 11. Describe the meaning of the term "Propagated" in this report and offer one reason why this row is empty (filled with dashes)?

This row is for propagated noise generated by noise on the preceding net and propagated through the driving cell. This row may be empty because noise propagation is not turned on (as in this analysis) and/or because noise propagation is not supported by the library (this is also true for the library used in this lab).

Question 12. Describe what is meant by the line "Steady state resistance source: library set iv curve"?

This identifies the model used for the steady-state I-V characteristics of the driver output to calculate the noise bump height. Other options could be, for example, "user set value" set by the command `set_steady_state_resistance`, or "estimation set value" if PrimeTime estimates the linear resistance in the absence of any other information.

Question 13. Were any noise derating factors applied to this net and does this match your expectations from viewing the run script?

Yes, there is a noise height and width scaling factor of 1.5 which was applied to the design in the run script.

Question 14. Describe what is meant by "noise effort threshold"?

Nets whose noise bump height is larger than is 0.2 (also the default) times the rail-to-rail voltage will be calculated using a more complex calculator based on Arnoldi order reduction.

Question 15. What command allows a user to exclude specific aggressor nets from analysis or define them as having an infinite timing window?

The command `set_si_noise_analysis`.

Question 16. Describe what is meant by the line "Constraint type: library immunity table"?

This describes the model used to calculate the noise constraint (the required or allowed noise bump height). Other options could be, for example, "user margin" set by the command `set_noise_margin`, or "none" if a noise constraint is not available in the library nor by the user.

Question 17. What is the noise constraint (i.e. the required noise bump height in library voltage units)?

The noise constraint is 0.6009.

Question 18. The slack "area" calculation is different from the noise bump "area" in the previous section. Can you describe how they are different?

The noise bump area is ½ height * width. The slack area is height * width. Use the Tcl command **expr** in the Primetime shell if you need a calculator to explore.

Question 19. Describe "how far off from the constraint" the worst flip-flop data pins are (i.e. lots of margin or little margin)?

Slack for above low: 58.27% (pin: T1_reg_120/SD)

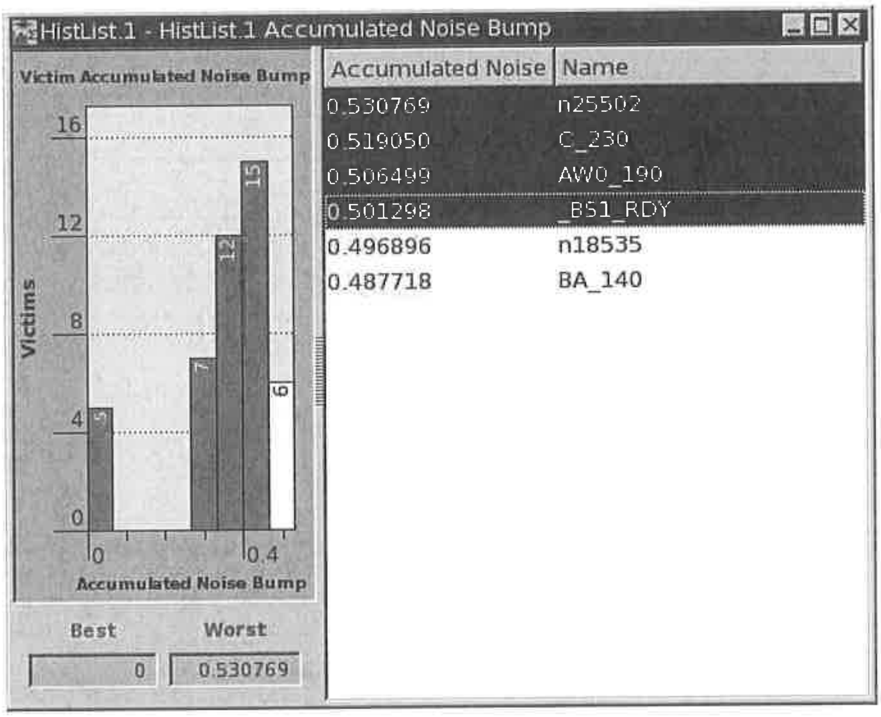
Slack for below high: INFINITY (pin: T1_reg_120/SD)

Question 20. Describe what is meant by a slack of infinity?

If the calculated noise bump is zero, the slack is defined as "INFINITY". The calculated noise bump could be zero because there are no effective aggressors for that net (as in this case) or all effective aggressors have been screened.

Question 21. How many nets have a noise bump height larger than 0.5?

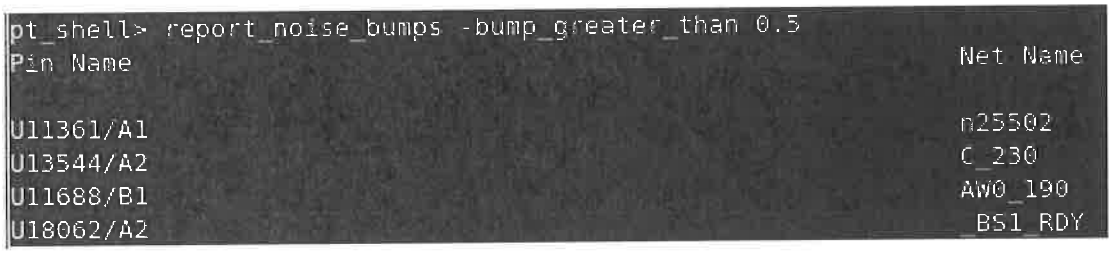
There are four nets in this design with a noise bump height greater than 0.5 in the below_high region.



HistList.1 - HistList.1 Accumulated Noise Bump

Accumulated Noise	Name
0.530769	n25502
0.519050	C_230
0.506499	AW0_190
0.501298	_BS1_RDY
0.496896	n18535
0.487718	BA_140

Best: 0 Worst: 0.530769



```
pt_shell> report_noise_bumps -bump_greater_than 0.5
Pin Name                                                    Net Name

U11361/A1                                                   n25502
U13544/A2                                                   C_230
U11688/B1                                                   AW0_190
U18062/A2                                                    _BS1_RDY
```

This page was intentionally left blank.

Summary Reports for All Violations

```
report_analysis_coverage          Total violations
      -status violated            Details on violations
      -check {setup hold}         For these timing checks
      -sort_by slack              Default
      -sort_by check_type
report_constraint -all            Sort violations by clock group
report_global_timing
report_qor -only_violated
```

report_timing Switches: Control Which and How Many

-nworst	Number of paths to consider per endpoint.
-max_paths	Number of paths to generate per design.
-delay	Setup (max) or hold (min) with a specific end point data transition (max_rise, max_fall, min_rise, min_fall).
-group	Focus on this path group (e.g. capture clock name).
-group [get_path_group *]	
-to -rise_to -fall_to	
-from -rise_from -fall_from	
-through -rise_through -fall_through	
-exclude -rise_exclude -fall_exclude	
-slack_lesser_than	
-slack_greater_than	

report_timing Switches: Control Details

-input_pins	Show input pins and separate net and cell delays.
-nets	Show net names and fanout.
-derate	Show derate factor for each delay.
-path full_clock	Show cells on clock network for propagated clocks.
-path short	Remove cells on data path.
-path end	Report endpoint, data arrival, required and slack only.

Helpful PrimeTime Commands

```
restore_session <unix_dir>
# Save the session before restoring
save_session <unix_dir>
report_timing -help
help -verbose report_timing
man report_timing
help report*; # Find commands
printvar *sig*; # Find variables
apropos clock ; # Search command DB
page_on; # An alias
page_off; # An alias
history
list_libraries; list_designs
report_lib <lib_name>; # Time units
remove_design -all
remove_lib -all
sh_list_key_bindings
set_app_var sh_line_editing_mode emacs
```

PrimeTime
JOB AID
Timing Reports

Tcl Examples

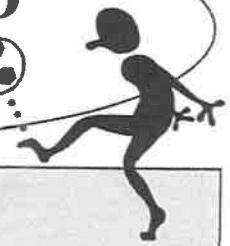
```
is is a comment
# No need to type entire command or switch names
# Use backslash to continue lines
# Use square brackets to embed commands
# Use get_clocks to refer to clock objects
report_timing -fall_from [get_clocks PCI_CLK] \
     -delay min -max 4
# Use set_app_var to modify variables
set_app_var sh_enable_page_mode true
```

```
# .synopsys_pt.setup
alias page_on {set_app_var sh_enable_page_mode true}
alias page_off {set_app_var sh_enable_page_mode false}
alias un_on {set_app_var timing_report_unconstrained_paths true}
alias un_off {set_app_var timing_report_unconstrained_paths false}
suppress_message {ENV-003 PTSR-004}
suppress_message CMD-029
history keep 200
set_app_var sh_enable_line_editing true
foreach _file [glob -nocomplain ./tcl_procs/*.tcl] {source $_file}
```

Tcl Procedures		SolvNet Doc ID
aa;	# always ask for both commands and variables	012959
view;	# Pop-up window for long reports	014947

PT JOB AID

Run Scripts



```
# variables.tcl
set_app_var report_default_significant_digits 4
set_app_var link_create_black_boxes false
set_app_var sh_source_uses_search_path true
set_app_var sh_script_stop_severity E
set_app_var timing_update_status_level high
```

Find Non-Default Application Variables

```
report_app_var -only_changed_vars
```

Path-Based Analysis

```
report_timing -pba_mode path …
report_timing -pba_mode exhaustive …
```

```
# RUN.tcl
start_profile
set_app_var search_path {. ./scripts ./libs ./designs}
lappend link_path mytech_lib.db RAM_lib.db
source -echo -verbose ./scripts/variables.tcl
read_verilog {top.v A.v B.v}
link_design TOP
read_parasitics ORCA.SPEF.gz
report_annotated_parasitics
source -echo -verbose constraints.tcl
redirect -tee ./EW.log {check_timing}
report_analysis_coverage
save_session top_savesession
redirect -tee -append ./EW.log {print_message_info}
redirect -tee -append ./EW.log {quit}
```

```
unix% pt_shell -f RUN.tcl | tee -i run.log
# If the script terminates, fix errors
# If script does not terminate, search run.log for Warnings
# Generate man pages on message ID for further information
```

PT JOB AID

Clocks and More

```
# Get to know the design clocks
report_clock
sizeof_collection [all_clocks]
sizeof_collection [get_generated_clocks *]
rpt_clock_ports; Tcl procedure in lab, Identify ports with defined clocks
report_clock -skew SD_DDR_CLK; # Reports generated clock source latency
report_timing -path full_clock_expanded; # Expands generated clock source latency
check_timing -verbose -override clock_crossing
set_app_var timing_all_clocks_propagated true; # Clocks created as propagated
# Identify clock network start points
reset_design
clk_strt -verbose; # Tcl procedure in lab
# Flag multiple clocks propagating (suggests a missing clock controlling signal)
lappend timing_check_defaults ideal_clocks; # For check_timing
report_case_analysis; # Report all case values set in design
```

Tcl Procedure	SolvNet Doc ID
rpt_case_value	009400
get_clock_ports	012739

```
# Back Annotation and Analysis Types
report_design; # Validate the analysis type
read_parasitics -syntax_only
report_annotated_parasitics
set_annotated_delay;
remove_annotated_delay; remove_annotated_check
set_load; set_resistance; # set net RC
report_delay_calculation
set_app_var timing_remove_clock_reconvergence_pessimism true
```

```
Symbol    Annotation
------    ----------
 H        Hybrid annotation
 *        SDF back-annotation
 &        RC network back-annotation
 $        RC pi back-annotation
 +        Lumped RC
<none>    Wire-load model or none
```

```
# Other
report_timing -group **async_default**; # Recovery/Removal
report_timing -group **clock_gating_default**
report_clock_timing -type skew
report_exceptions <-ignored>
report_min_pulse_width
report_timing -trace_latch_borrow
```

Navigation

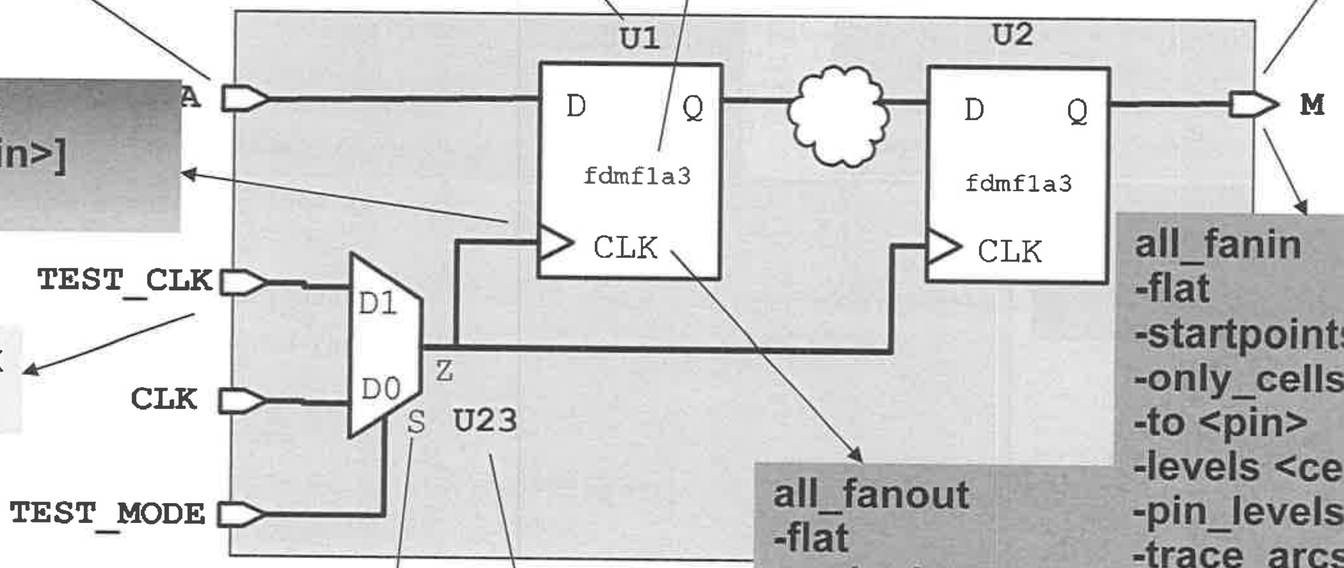
```
all_inputs -clock CLK1 ;# Input ports clocked by CLK1
get_pins -of_object U1 ;# Pin names of U1
all_connected U1/Z ;# Nets connected to pin
all_registers -level_sensitive ; # Latches
```

10-I-034-SLG-015

PT JOB AID

Debugging Constraints

check_timing
report_analysis_coverage



report_port
-input_delay
<port>

report_cell
-connections
-verbose
<cell>

report_lib
-timing_arcs
<library>
<lib cell>

report_port
-output_delay
<port>

get_attribute
[get_pins <pin>]
clocks

report_clock
<clock>

all_fanin
-flat
-startpoints
-only_cells
-to <pin>
-levels <cell#>
-pin_levels <pin#>
-trace_arcs enabled

all_fanout
-flat
-endpoints
-only_cells
-from <pin>
-levels <cell#>
-pin_levels <pin#>
-trace_arcs enabled

report_case_propagation
<pins or ports>

Tcl procedure available
on SolvNet

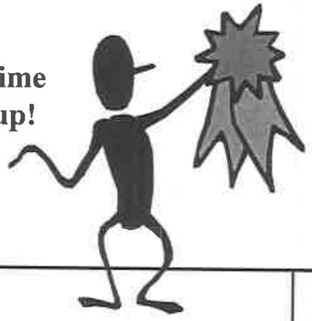
report_disable_timing
<cells>

PrimeTime SI Crosstalk Delay

Key Analysis Reports

You want to report . . .	Fixing Solution	PrimeTime Command
Total # violations for each timing check.		report_analysis_coverage <-status_details violated>
Detailed stage delay, delta delay, delta slew.	Use non-SI fixing techniques if violation is not due to crosstalk.	report_timing -crosstalk -transition -nets <-to pin>
Weakest victim nets with the worst delta delay on violating paths.	Reduce Cc by net isolation.	report_si_bottleneck -cost_type delta_delay \ <-slack_lesser_than 0> <-max> <-min>
Most significant victim nets with a large ratio delta delay/stage delay on violating paths.	Upsize victim driver or insert a buffer.	report_si_bottleneck -cost_type delta_delay_ratio \ <-slack_lesser_than 0> <-max> <-min>
Strongest aggressor nets with largest sum of switching bumps on all violating victims.	Isolate aggressor nets or downsize the aggressors to fix multiple victims.	report_si_bottleneck -cost_type delay_bump_per_aggressor \ <-slack_lesser_than 0> <-max> <-min>
Highly coupled victim nets with largest sum of switching bumps (independent of delta delay) and a large # of aggressors.	Potential noise issues or indication of implementation problems.	report_si_bottleneck -cost_type total_victim_delay_bump \ <minimum_active_aggressors 100> \ <-slack_lesser_than 10> <-max> <-min>
Effect of crosstalk on clock skew for each clock domain.	Indication of clock tree implementation problems.	report_clock_timing -verbose -crosstalk -type skew

Ensure layout tool and PrimeTime SI use the same inputs and setup!



PrimeTime SI Crosstalk Delay & Noise

Complete Your Inputs

You want to check that . . .	PrimeTime Command
Analysis is performed in a single PVT corner and the analysis type is on_chip_variation.	`report_design` `list_lib`
All input ports (including the clocks) have a driving cell defined.	`check_timing`

All input ports have min and max input delays defined.	`check_timing`
	Warning! – this will not check for min/max numbers that are equivalent (which will still result in zero arrival windows). Use the following instead.
	`report_port -input_delay <specific port>`

All clocks are propagated.	`check_timing -override ideal_clocks`
	`report_clock`

Asynchronous clocks are specified using `set_clock_group`.
Physically exclusive clocks (e.g. test and functional clocks) are analyzed separately using inactive clocks or by not creating them together during the same STA run.
	First, check which clocks are interacting in your design.
	`check_timing -override clock_crossing -verbose`
	The GUI window "Clock Domain Matrix" has the same information as a graphical matrix.
	Then, report asynchronous, exclusive groups and active/inactive clocks.
	`report_clock -groups -attributes`

There is no missing parasitic data.
There are no associated RC-* DES-* PARA-* warnings.
	`report_annotated_parasitics <-list_not_annotated>`
	`print_message_info; # Must be done in run script`

Constant signals have case values defined.	`report_case_analysis` Requires design knowledge to confirm this information.

SolvNet articles	Doc ID
An easy way to generate delay calculation reports on nets	013383
How can an effective aggressor be screened due to bump size	014179
Accurate sign-off analysis with PrimeTime's path-based analysis	012134
View long reports in the shell with a pop-up window	014947
Search Synopsys commands and variables with Tcl procedure aa	012959

PrimeTime SI Crosstalk Delay
Fine Tuning for Timing Closure

Control a Specific Net Coupling Information

```
set_si_delay_analysis
```

Path-Based Analysis Related Commands

```
set viol_path [get_timing_paths -slack_lesser 0]
set path [get_timing_paths -pba_mode exhaustive $viol_path]
report_timing -crosstalk $pathset set
report_timing -pba_mode path …
report_timing -pba_mode exhaustive …
```

What-If Analysis Related Commands

```
size_cell
report_lib tech_lib buf*
insert_buffer
remove_buffer
set_coupling_separation
set eco_instance_name_prefix
set eco_net_name_prefix
estimate_eco
fix_eco_timing
fix_eco_drc
write_changes
```

```
# Variables to control victim reselection
printvar si_xtalk_reselect*
# Variables to control exit criteria
printvar si_xtalk_exit*
# Variable to control electrical filtering
printvar si_filter*
```

PrimeTime SI Crosstalk Noise
Essential Commands

SolvNet articles	Doc ID
Checking the Library for Noise Characterization	037963
Details on report_noise_calculation	014387
Application note on incomplete library noise characterization	013888
Checking Consistency of Pin-Based CCS Noise Models	2285479
What is si_noise_limit_propagation_ratio?	013301

Incomplete Noise Characterization Library

```
set_steady_state_resistance
set_input_noise
set_noise_immunity_curve
set_noise_margin
set si_noise_nmos_threshold_ratio 0.2
set si_noise_pmos_threshold_ratio 0.2
```

Report Noise Violations and Details

```
report_noise [-nworst_pins] [-all_violators] [-verbose] [-clock_pins] [async_pins] [-slack_type] object_list
report_noise_calculation [-from pin] [-to pin]
```

Report noise bump heights using the GUI “Composite Noise Bump Histogram”

Invoking and Controlling Noise Analysis

```
update_noise
set si_noise_update_status_level high; # Turn on informational messages during a timing update
set_noise_parameters [-ignore_arrival] [-include_beyond_rails] [-enable_propagation]
set si_noise_limit_propagation_ratio 0.75; # Scale propagated noise for violating nets
set_si_noise_analysis; # Exclude nets from analysis or define infinite timing windows for aggressor nets
set_noise_derate; # Define noise scaling factors
```

PrimeTime SI ECO

Essential Commands

ECO Commands	Purpose	PrimeTime support
`fix_eco_drc`	max_transition, max_capacitance, and max_fanout fixing	PrimeTime-SI
`fix_eco_timing`	Setup and hold fixing	PrimeTime-SI
`fix_eco_power`	Leakage recovery by Vt Cell Swap	PrimeTime-ADV
`set_eco_options`	Configure physically aware fixing	PrimeTime-ADV
`write_changes`	Create ECO change file	PrimeTime-SI
`report_cell_usage`	Reports Vt cell usage	PrimeTime-SI

Important variables:	Purpose
`eco_report_unfixed_reason_max_endpoints`	Displays unfixable violations with the reasons
`eco_alternative_area_ratio_threshold`	Limit the cell size increase for better closure with P&R
`read_parasitics_load_locations`	Read design parasitics with location for on route buffering

This page was intentionally left blank.