Static Timing Analysis

NOTE: sta_tut.tcl is just a sample script file. You have to modify this file in order to make it appropriate for your design.

Source the setup file in the sta folder.

Check if PrimeTime is installed on your account
which pt_shell

You should get the following:
/usr/local/synopsys//PrimeTime/default/bin/pt_shell

Start the PrimeTime shell at `~/sta/` directory.
`pt_shell`

Souce the .synopsys_pt.setup file.

PrimeTime displays the version number and copyright notice, followed by the pt_shell prompt.

Read the Design and Library Files

```
pt_shell> read_verilog ./designs/example_design.syn.v
Loading verilog file ' .../sta/designs/example_design.syn.v'
1
```

The 1 returned by the command indicates successful completion.

NOTE: example_design.syn.v is your synthesized verilog (output of design compiler) which has to be copied in "`~/sta/designs/`" directory.

Specifying the target design:

Use the following command to specify your top module as the target design for linking:

```
pt_shell> current_design example_design
{"example_design"}
```

Note: you can see the list of all loaded modules (designs), using get_designs command:

```
pt_shell> get_designs
```

Link the design:

Link the design to the library.

```
pt_shell> link_design
Loading db file ' ... /tsmc018/osu018_stdcells.db'
Linking design example_design ...
Designs used to link example_design:
```

```
<None>
Libraries use to link example_design:
tut_lib
Design 'example_design' was successfully linked.
```

Specify the Timing Constraints

Before you can analyze a design, you need to specify the timing constraints, including the clocks, input delays, and output delays.

1. To check the design for timing setup problems:

```
pt_shell> check_timing Information: Checking
'no_clock'. Information: Checking
'no_input_delay'. Information: Checking
'partial_input_delay'. Information: Checking
'no_driving_cell'.
Information: Checking 'unconstrained_endpoints'.
Warning: There are 7 endpoints which are not constrained for
maximum delay.
Information: Checking 'unexpandable_clocks'.
Information: Checking 'generic'.
Information: Checking 'latch_fanout'.
Information: Checking 'loops'. Information:
Checking 'generated_clocks'.
0
```

The `check_timing` command reports registers that are not clocked and timing paths that are not constrained at their endpoints. This is because you have not yet defined the clocks and input/output timing constraints. The 0 reported at the end indicates that the design did not pass the timing setup check.

2. To define a 500 MHz clock:

```
pt_shell> create_clock -period 2.0 -name CLK [get_ports clk1]
1
```

This `create_clock` command defines a clock named CLK having a period of 2.0ns (time unit). The time unit size is defined in the technology library which is nanoseconds, in our OSU library. The duty cycle is 50 percent by default.
The command `[get_ports clk]` is nested within the `create_clock` command. It looks for a port named *clk*, then passes the port as an argument to the `create_clock` command. This argument specifies the source of the clock (the place in the design where the clock signal exists).

3. To set the clock latency:

```
pt_shell> set_clock_latency 0.5 CLK
1
```

The `set_clock_latency` command specifies the total amount of delay from a

CLK clock edge to the arrival of the clock edge at sequential devices inside the design.
Note that pt_shell is case sensitive (CLK and clk are not equal). To define the input delay and output delay for each port:

```
pt_shell> set_input_delay 0.5 -clock CLK [all_inputs]
1
pt_shell> set_output_delay 0.5 -clock CLK [get_ports Y*]
1
```

The `set_input_delay` command specifies the amount of delay from an edge of the CLK clock to the arrival of data on all input pins. It establishes a timing constraint at those inputs and specifies the arrival time of data with respect to the CLK clock.
The `set_output_delay` command specifies the amount of delay from outputs with name Y*, such as Y[0] and Y[1], to the external device that captures the output data. It establishes a timing constraint at the outputs and specifies the required time for output data with respect to the CLK clock.

4. Check the design constraints with the following commands:

```
pt_shell> report_clock -skew
...
pt_shell> report_port -input_delay
...
pt_shell> report_port -output_delay
...
pt_shell> check_timing
...
check_timing succeeded.
1
```

This time, the `check_timing` command does not report any setup errors because the design is fully constrained.

NOTE: you can write all the commands that you want to use in a script file and then source that script file. This way you do not need to do same things over and over again which will save your time.
You can start with the sample script file, sta_tut.tcl, and modify it (refer to page1 of this document). You can then run your script file that you have prepared by typing the following:

```
%source ./scripts/sta_tut.tcl
```

End the Session

When you are finished with a project, you can remove the designs and libraries that have been read into PrimeTime.
To remove the loaded design from PrimeTime:

```
pt_shell> remove_design -all
Removing design 'example_design' ...
1
```

To remove the loaded library from PrimeTime memory:

```
pt_shell> remove_lib -all
Removing library '/.../tutpt/libs/tut_lib.db' ...
1
```

At this point, with all the design and library removed, you could start work on a new project.

To end the PrimeTime session:

```
pt_shell> exit
Number of timing updates: ...
Maximum memory usage for this session: ...
CPU usage for this session: ...
Diagnostics summary: ...
Thank you for using pt_shell!
...
```