## Agenda: Day Two

### Unit 2: I/O Paths and Exceptions

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td>Constraining I/O Interface Paths</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Specifying Timing Exceptions</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Introduction to Timing Models (QTM)</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Performing STA</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Summary</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Customer Support</td>
</tr>
</tbody>
</table>

---

**Creating Timing Models (QTM)**
After completing this unit, you should be able to:

- Create a quick timing model (QTM) of a block given a block specification
The Inputs and Outputs of PrimeTime

- Gate-Level Netlist
- Constraints
- Exceptions
- Technology Libraries
- SDF
- Reports
- Log, Script Files

Our Focus

Creating Timing Models (QTM)

PrimeTime: Introduction to Static Timing Analysis
QTM is typically used as a place holder when some blocks are not yet available in the form of synthesized gates, but you still want to perform STA on the Functional Core for early estimation.

Here are some other Timing Models supported by PT:
ILM: Interface Logic Model (requires Gate level netlist).
ETM: Extracted Timing Model (requires Gate level netlist).

These 2 models (created more accurately using Post Layout data) are covered in the PrimeTime: Chip Level Timing Analysis workshop.
Quick Timing Models (QTM)

- A QTM is typically used to create a timing model for a block in which the gate level netlist does not yet exist

- **Normally used for early floorplanning and top-level timing analysis:**
  - Before HDL has been written
  - Before 3rd party macros (or their timing models) have been obtained

- A QTM is a “library cell” with setup, hold and delay arcs:
  - Created using a PrimeTime script of QTM commands

- A QTM is as accurate as your description of the interface timing specification of the design

To create QTM, you need to write a PT script containing QTM commands. Fortunately, all qtm commands contain the “qtm” string in them, thus you can obtain a list of all the QTM commands with the “help *qtm*” command.
The flow shows a role of QTM in the early floor planning stage when no blocks have been realized into gates. PT is used to create block level QTMs, which are used by a Top level Floor planner (CHIP Architect) to estimate the placement (and extract the net parasitics between them).

FlexRoute is used as the Top level router in this flow. Once you have the DB files for the block level QTMs and the Top level net parasitics between them, PT can perform STA and help derive block level constraints. Block level synthesis, (using DC) therefore, has a better starting point with more accurate block constraints because it took into account top level net parasitics.
Creating QTMs

- A QTM is created by writing a **PrimeTime** script containing QTM commands

- **Commands will indicate:**
  - Port name, directions and input to output delays
    - Input Port
      - setup and hold constraints
      - capacitive loading
    - Output Port
      - Clock-to-output delays
      - Drive Strength

- Delay, load and drive are specified in technology library units

- Setup, hold, clk_to_output parameters can be specified as a constant value or derived from a library flip flop
Simple QTM Example

Your job is to design a 4-bit synchronous adder/subtractor

Before writing HDL code, a QTM will be written to:

- Formalize the design’s specification
- Allow the system engineer(s) to perform top-level routing and timing analysis
The block-level specification is as follows:

- Clock Frequency = 250 Mhz
- Inputs shall perform desired mathematical operation within 2 ns
- Input pin capacitance shall not be more than 0.05 fF
- Clock-to-output delay shall be less than 0.5 ns
- Output drive resistance shall be less than 0.05 mΩ
Basic QTM Flow - Define Ports

1. State the model name
2. Declare name & direction of IO ports

# Tell PT we’re defining a QTM
create_qtm_model ADDSUB

# Define IO Ports
create_qtm_port {Clk} -type clock
create_qtm_port {A[3:0] B[3:0] add_subN} -type input
create_qtm_port {Y[3:0] carry_borrow} -type output
**Define input port characteristics:**

- Timing
- Capacitive Loading

```plaintext
# Define input port setup time
set_qtm_global_parameter -param setup -value 0.0
create_qtm_constraint_arc -setup -from Clk \

# Define input port capacitive loading
    -value 0.05
```

`set_qtm_global_parameter` specifies one of the 3 global parameters: setup, hold or clk_to_output. The value specified (in this case 0.00) is added to the arcs (in this case setup). Consider this as a way of adding a known setup time from library flop, (OR) you can combine the setup time altogether with the constraint arc.

You can set a global parameter extracted from your library flop-flop, for example:

```
Set_qtm_global_parameter –param setup –lib_cell fd1fa1 –pin D.
```

Constraint arc is defined (in reverse) from the Clk to data as in the `create_qtm_constraint_arc`, since it is a setup check as opposed to propagation delay.
Define output port characteristics:

- Timing
- Drive Strengths

```csharp
# Define output port clock-to-output timing
set_qtm_global_parameter -param clk_to_output \\n  -value 0.0
create_qtm_delay_arc -from Clk \n  -to {Y[3:0] carry_borrow} -value 0.5 -edge rise

# Define output port drive strengths
set_qtm_port_drive {Y[3:0] carry_borrow} -value 0.05
```

In this case, the global parameter clk_to_output value (0.00) is added to the delay arc. “clk_to_output” can be specified as a fixed value (in this case, 0.00) or can be derived from that of a library flip-flop. Example: set_qtm_global_parameter –param clk_to_output –lib_cell fd1fa1 –pin QN.
Issue a QTM model report to ensure the model was entered correctly

Save the model in “.db” format for use by Synopsys tools

```
# Before saving, report model & check results
redirect qtm.rpt report_qtm_model

# Save timing model
save_qtm_model
```
Saving QTM

pt_shell> report_lib ADDSUB_lib

Library Cells:

Attributes:
  b - black box (function unknown)
  ....other attributes...
  Q - Quick timing model (QTM)

<table>
<thead>
<tr>
<th>Lib Cell</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDSUB</td>
<td>b,Q</td>
</tr>
</tbody>
</table>

By default, the base name (or design name) for the QTM files are specified to the create_qtm_model command (ADDSUB).

save_qtm_model creates a library DB file, i.e., basename_lib.db (in our case, ADDSUB_lib.db).

You can use the “-output” option with save_qtm_model to change the name of the file.
Using QTM

pt_shell> lappend link_path ADDSUB_lib.db
pt_shell> read_verilog top.v
pt_shell> current_design TOP
pt_shell> link_design

```bash
pt_shell> lappend link_path ADDSUB_lib.db
pt_shell> read_verilog top.v
pt_shell> current_design TOP
pt_shell> link_design
```
pt_shell> report_cell

Attributes:
n - noncombinational
...other attributes...
E - extracted timing model
S - Stamp timing model
Q - Quick timing model (QTM)

<table>
<thead>
<tr>
<th>Cell</th>
<th>Reference</th>
<th>Library</th>
<th>Area</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>ADDSUB</td>
<td>ADDSUB_lib</td>
<td>0.00</td>
<td>n,Q</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

Zero Area
QTM Parameters in the Timing Report

pt_shell> report_timing –from [all_inputs]

clock Clk (rise edge) 10.00 10.00
clock uncertainty -0.10 9.90
U1/Clk (ADDSUB) 9.90 r
library setup time -6.00 3.90
data required time 3.90

-----------------------------------------------

data required time 3.90
data arrival time -3.25

-----------------------------------------------

slack (MET) 0.65

Test For Understanding

1. Which of the following is (are) NOT specified when creating a QTM:
   a) Design input, output port names
   b) Timing from an input to output
   c) Functionality of an output in terms of input(s)
   d) Input port load, output port drive

2. Under what circumstance(s) is a QTM used? Circle all that apply:
   a) The gate level netlist of a block is too large for PrimeTime
   b) The gate level netlist of a block has been verified for timing and stable
   c) Inter block net parasitics are to be extracted for a more accurate block synthesis
   d) RTL code for one of the blocks in Functional core is not yet available
   e) 3rd party IP is not yet available

3. Circle whichever is (are) correct:
   a) A QTM library file can be read into PrimeTime memory using read_verilog
   b) A QTM library file is specified in the link_path variable
You will be given a design specification, from which you will create a quick timing model.