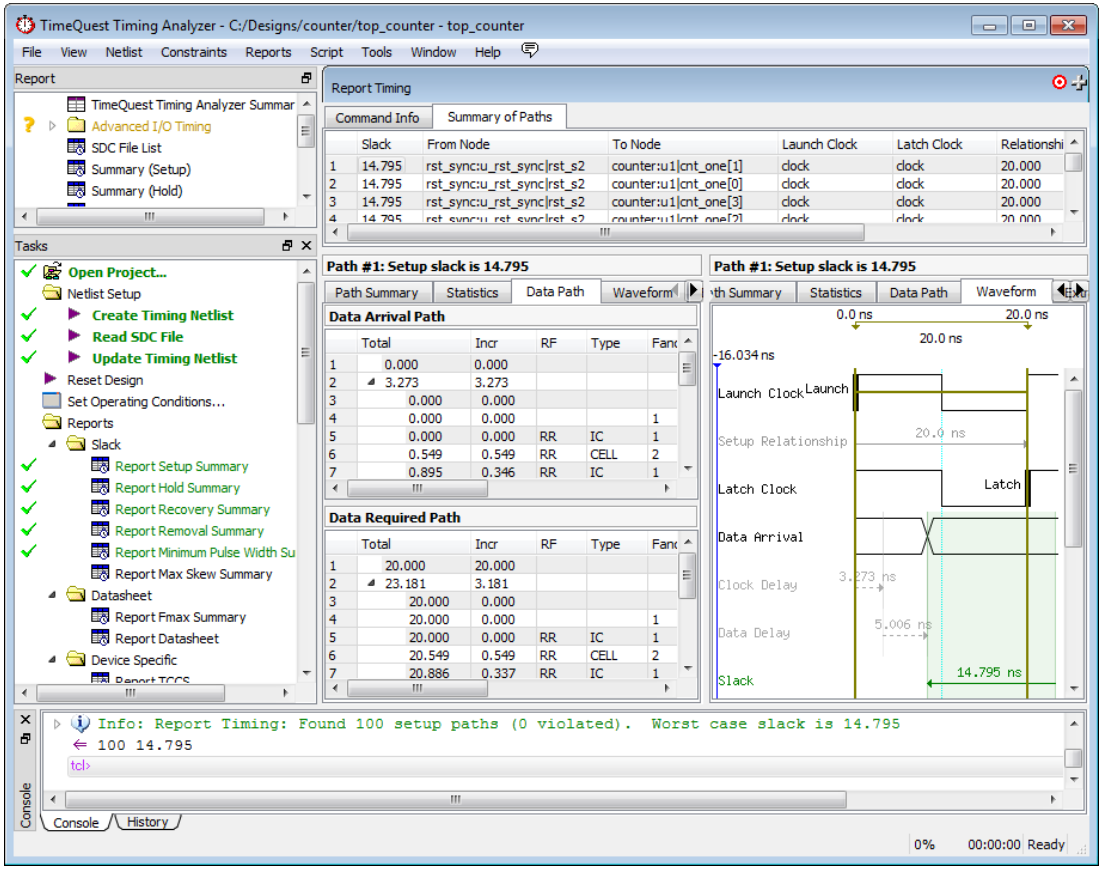
Quartus® II Software Design Series: Timing Analysis

TimeQuest Basics

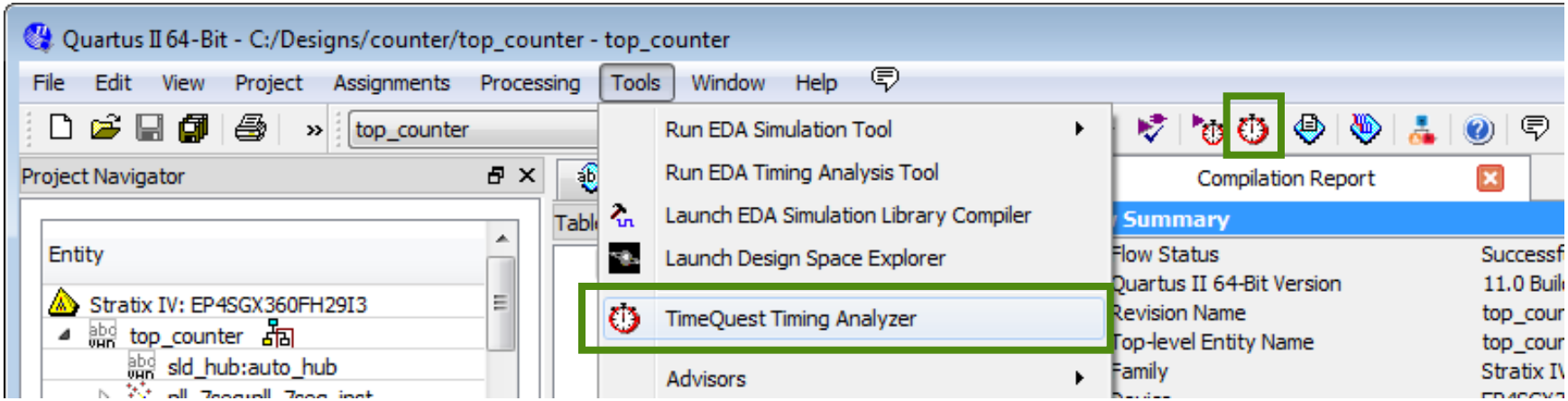
TimeQuest Timing Analyzer

- Timing analysis engine in Quartus II software
- Provides timing analysis solution for all levels of experience
- Features
  - Synopsys Design Constraints (**SDC**) support
    - Standardized constraint methodology
  - Easy-to-use interface
    - Constraint entry
    - Standard, on-the-fly reporting
  - Scripting emphasis
    - Presentation focuses on using GUI

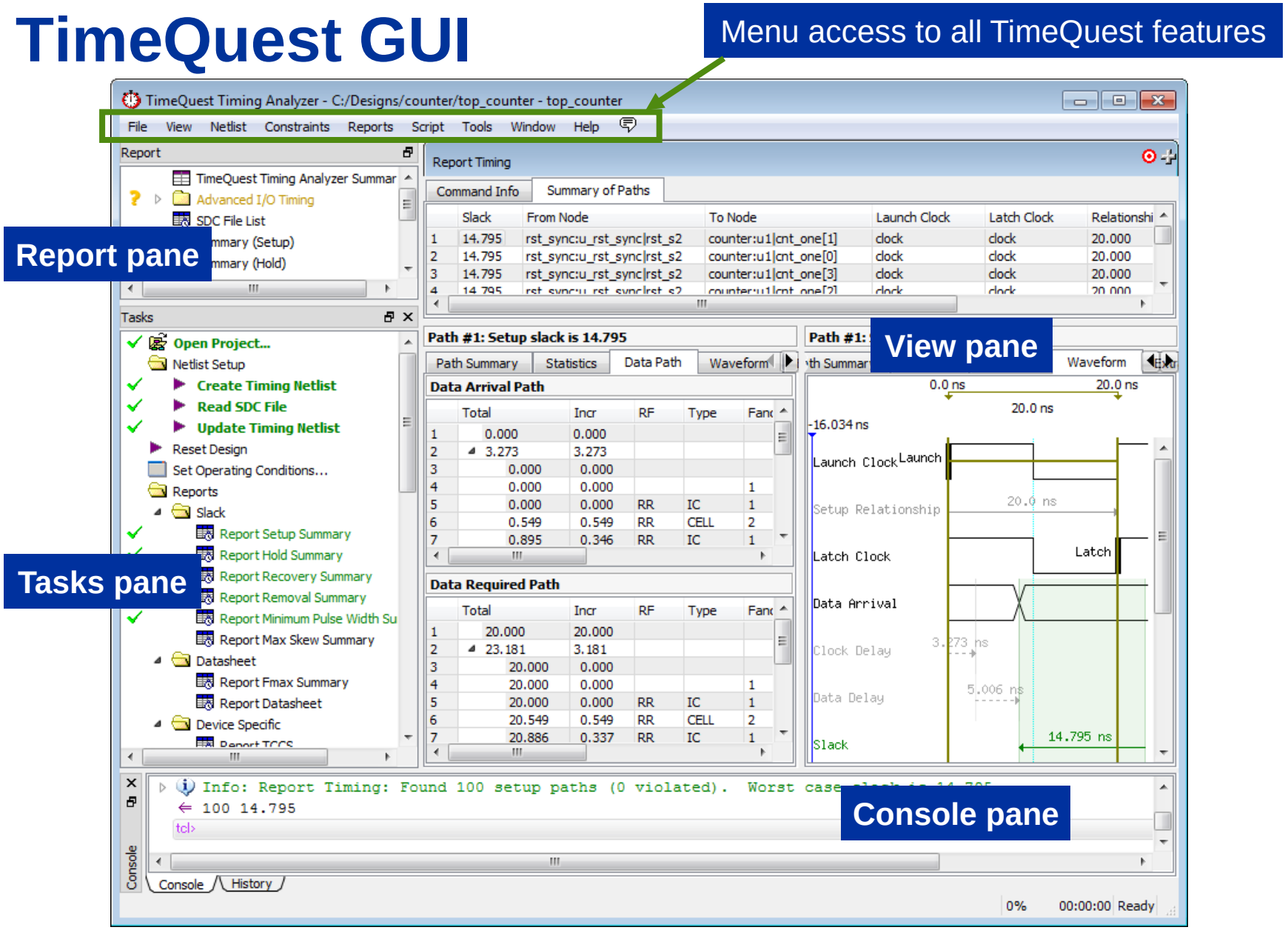


Opening the TimeQuest Interface

- Toolbar button 
- **Tools** menu
- **Tasks** window
- Stand-alone mode
  - **`quartus_staw`**
- Command line

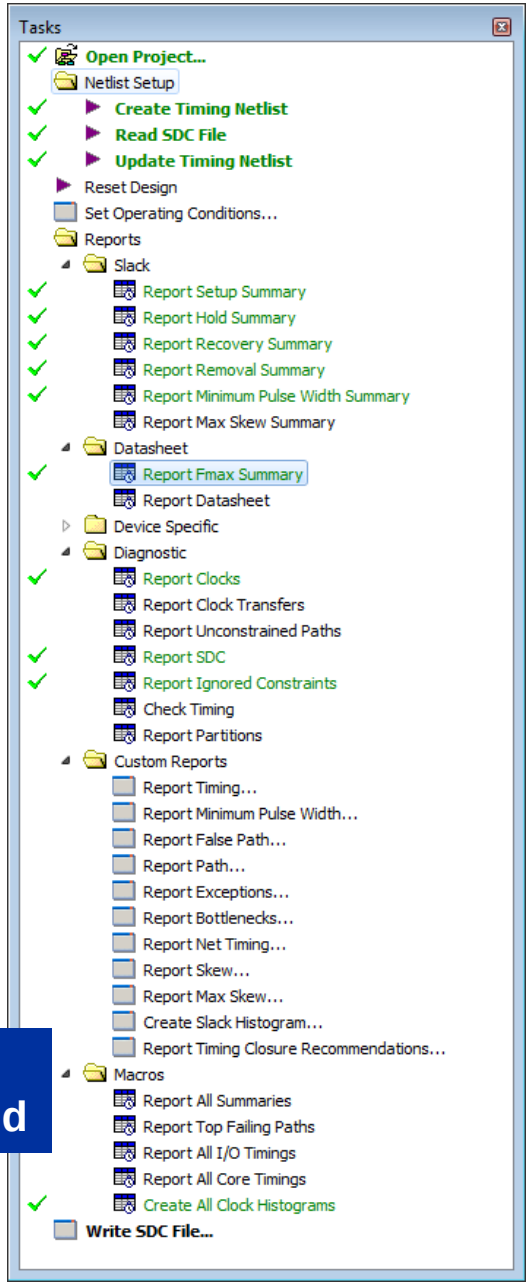


TimeQuest GUI



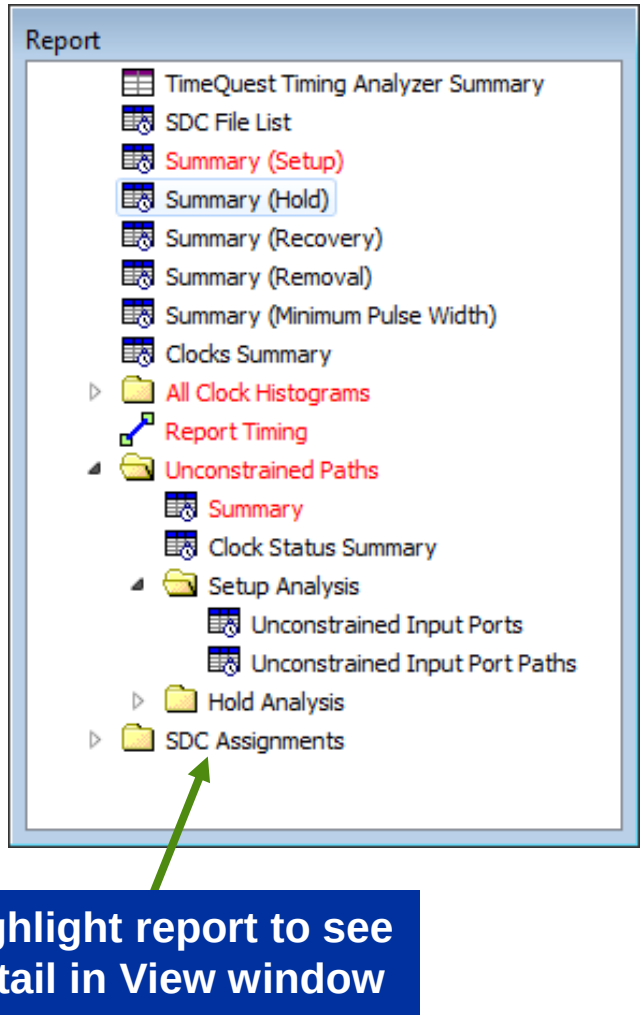
Tasks Pane

- Provides quick access to common operations
  - Command execution
  - Report generation
- Executes most commands with ***default*** settings
- Use menus for non-default settings



Double-click to execute any command

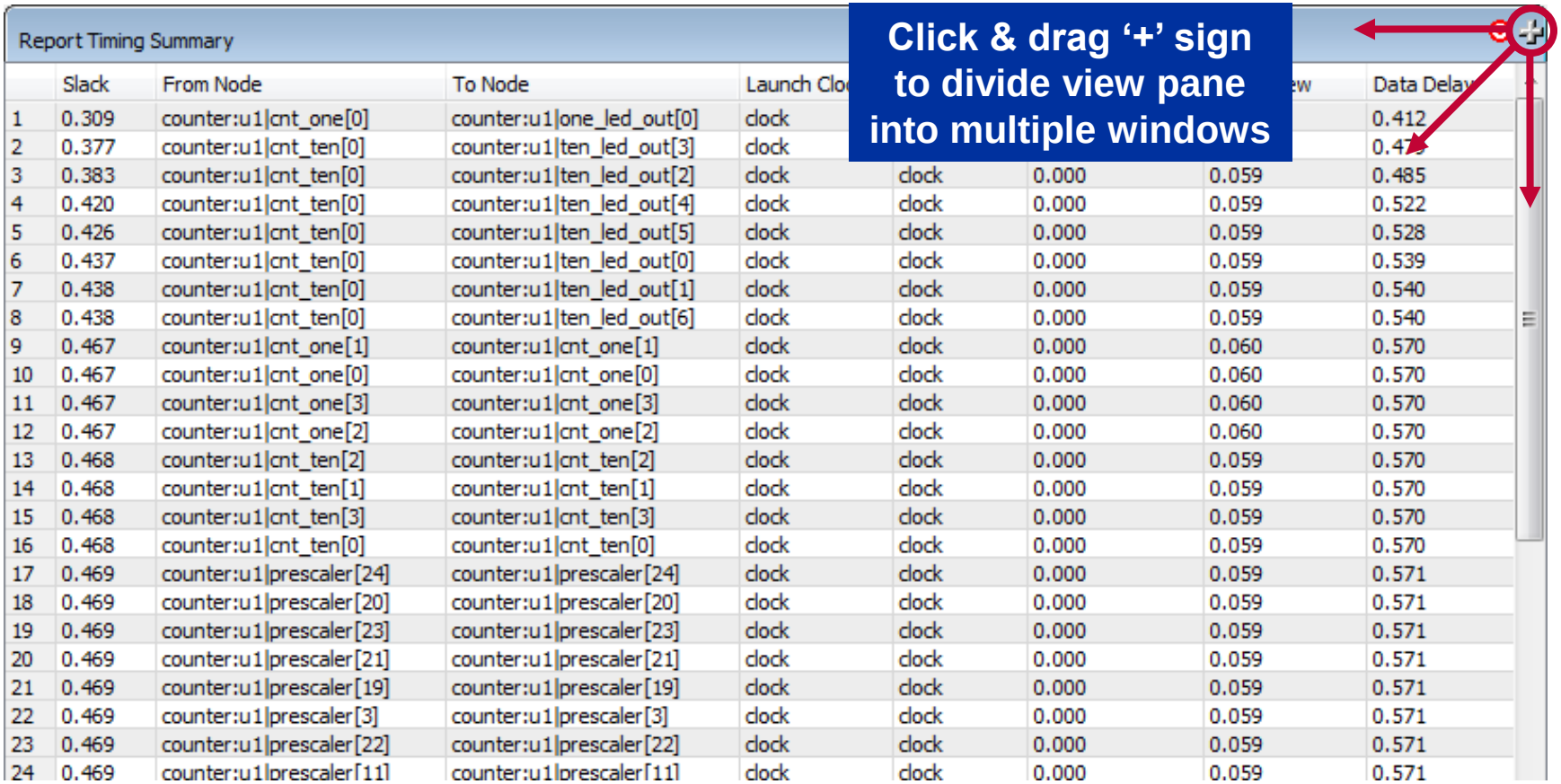
Report Pane



Highlight report to see detail in View window

- Displays list of generated reports currently available for viewing
  - Reports generated by Tasks pane
  - Reports generated using report commands

Viewing Multiple Reports



Report Timing Summary

	Slack	From Node	To Node	Launch Clo				Data Delay
1	0.309	counter:u1|cnt_one[0]	counter:u1|one_led_out[0]	clock				0.412
2	0.377	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[3]	clock				0.47
3	0.383	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[2]	clock	clock	0.000	0.059	0.485
4	0.420	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[4]	clock	clock	0.000	0.059	0.522
5	0.426	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[5]	clock	clock	0.000	0.059	0.528
6	0.437	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[0]	clock	clock	0.000	0.059	0.539
7	0.438	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[1]	clock	clock	0.000	0.059	0.540
8	0.438	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[6]	clock	clock	0.000	0.059	0.540
9	0.467	counter:u1|cnt_one[1]	counter:u1|cnt_one[1]	clock	clock	0.000	0.060	0.570
10	0.467	counter:u1|cnt_one[0]	counter:u1|cnt_one[0]	clock	clock	0.000	0.060	0.570
11	0.467	counter:u1|cnt_one[3]	counter:u1|cnt_one[3]	clock	clock	0.000	0.060	0.570
12	0.467	counter:u1|cnt_one[2]	counter:u1|cnt_one[2]	clock	clock	0.000	0.060	0.570
13	0.468	counter:u1|cnt_ten[2]	counter:u1|cnt_ten[2]	clock	clock	0.000	0.059	0.570
14	0.468	counter:u1|cnt_ten[1]	counter:u1|cnt_ten[1]	clock	clock	0.000	0.059	0.570
15	0.468	counter:u1|cnt_ten[3]	counter:u1|cnt_ten[3]	clock	clock	0.000	0.059	0.570
16	0.468	counter:u1|cnt_ten[0]	counter:u1|cnt_ten[0]	clock	clock	0.000	0.059	0.570
17	0.469	counter:u1|prescaler[24]	counter:u1|prescaler[24]	clock	clock	0.000	0.059	0.571
18	0.469	counter:u1|prescaler[20]	counter:u1|prescaler[20]	clock	clock	0.000	0.059	0.571
19	0.469	counter:u1|prescaler[23]	counter:u1|prescaler[23]	clock	clock	0.000	0.059	0.571
20	0.469	counter:u1|prescaler[21]	counter:u1|prescaler[21]	clock	clock	0.000	0.059	0.571
21	0.469	counter:u1|prescaler[19]	counter:u1|prescaler[19]	clock	clock	0.000	0.059	0.571
22	0.469	counter:u1|prescaler[3]	counter:u1|prescaler[3]	clock	clock	0.000	0.059	0.571
23	0.469	counter:u1|prescaler[22]	counter:u1|prescaler[22]	clock	clock	0.000	0.059	0.571
24	0.469	counter:u1|prescaler[11]	counter:u1|prescaler[11]	clock	clock	0.000	0.059	0.571

Viewing Multiple Reports Example

View pane split into four windows

Report Timing Summary

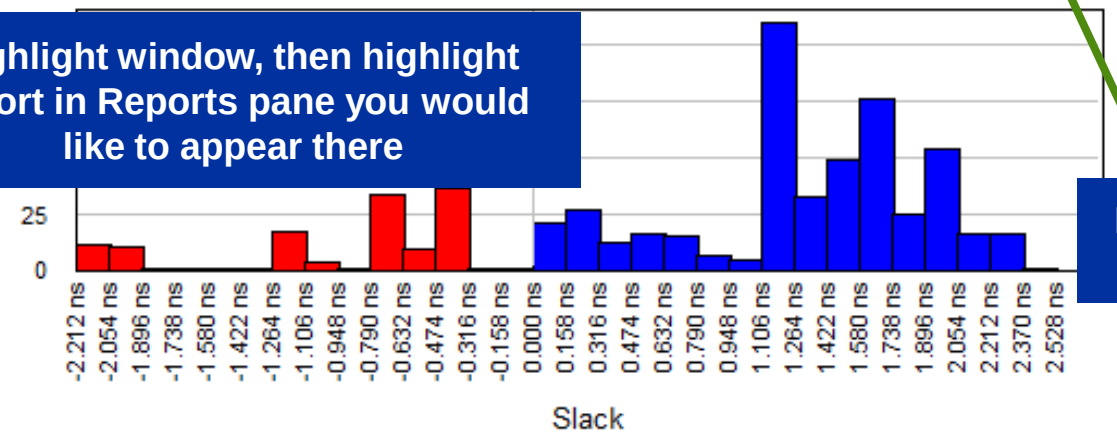
	Slack	From Node	To Node	Launch Clock	Latch Clock
1	0.309	counter:u1|cnt_one[0]	counter:u1|one_led_out[0]	clock	clock
2	0.377	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[3]	clock	clock
3	0.383	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[2]	clock	clock
4	0.420	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[4]	clock	clock
5	0.426	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[5]	clock	clock
6	0.437	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[0]	clock	clock
7	0.438	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[1]	clock	clock
8	0.438	counter:u1|cnt_ten[0]	counter:u1|ten_led_out[6]	clock	clock
9	0.467	counter:u1|cnt_one[1]	counter:u1|cnt_one[1]	clock	clock
10	0.467	counter:u1|cnt_one[0]	counter:u1|cnt_one[0]	clock	clock

Summary (Setup)

	Clock	Slack	End P
1	clock	-2.205	-30.4
2	altera_reserved_tck	48.146	0.000
3	seg_clk	29975.850	0.000

Use Target Pane button to force a selected report to appear in a pane

Slack Histogram (clock)



Highlight window, then highlight report in Reports pane you would like to appear there

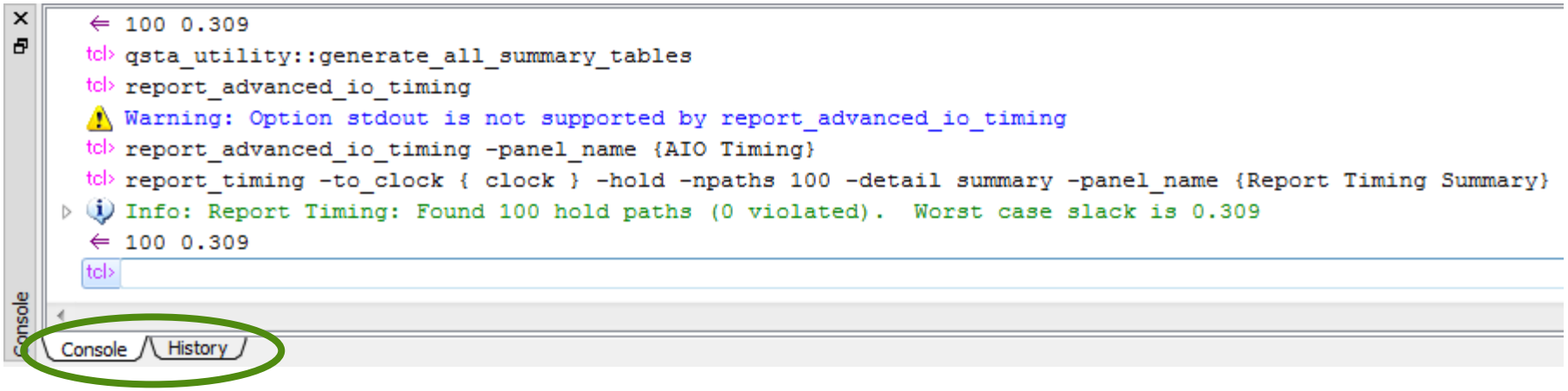
Drag bars to edges to remove splits

Summary (Hold)

	Clock	Slack	End Point
1	altera_reserved_tck	0.285	0.000
2	clock	0.309	0.000
3	seg_clk	0.340	0.000

Console Pane

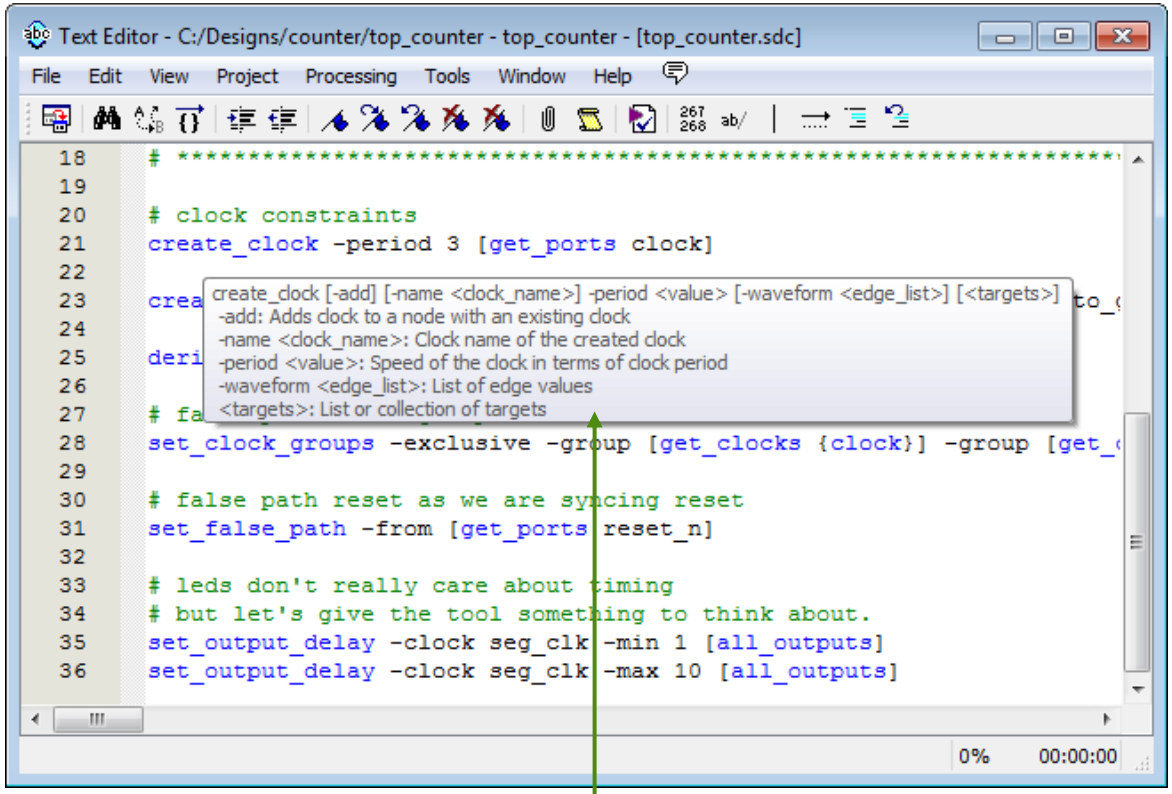
- Allows direct entry and execution of SDC & Tcl commands
  - Displays equivalent of command executed by GUI
- Displays TimeQuest output messages
- History tab records all executed SDC & Tcl commands
  - Copy & paste to create scripts or SDC files



SDC File Editor = Quartus II Text Editor

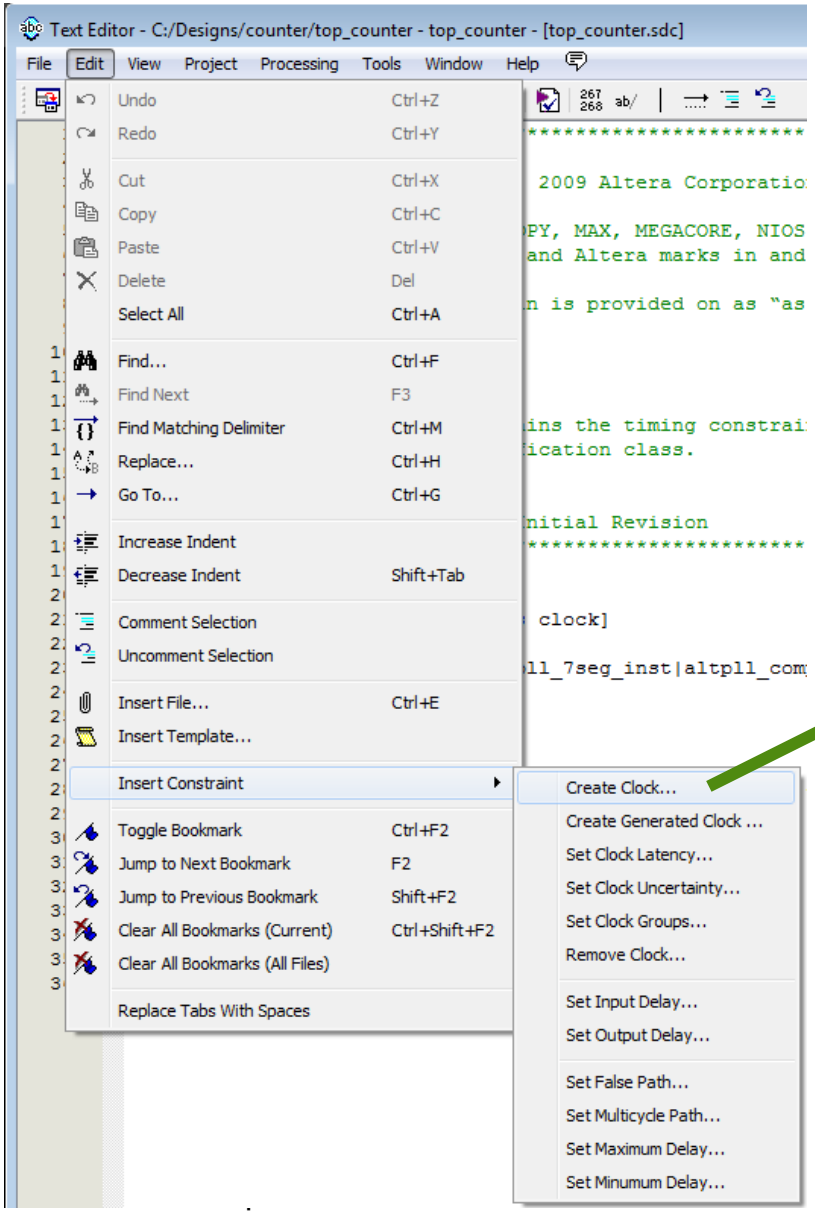
- Use Quartus II editor to create and/or edit SDC
- SDC editing unique features (for .sdc files)
  - Access to GUI dialog boxes for constraint entry (**Edit ⇒ Insert Constraint**)
  - Syntax coloring
  - Tooltip syntax help
  - SDC templates

TimeQuest File menu ⇒ Open/New SDC File
Quartus II File menu ⇒ New ⇒ Other Files

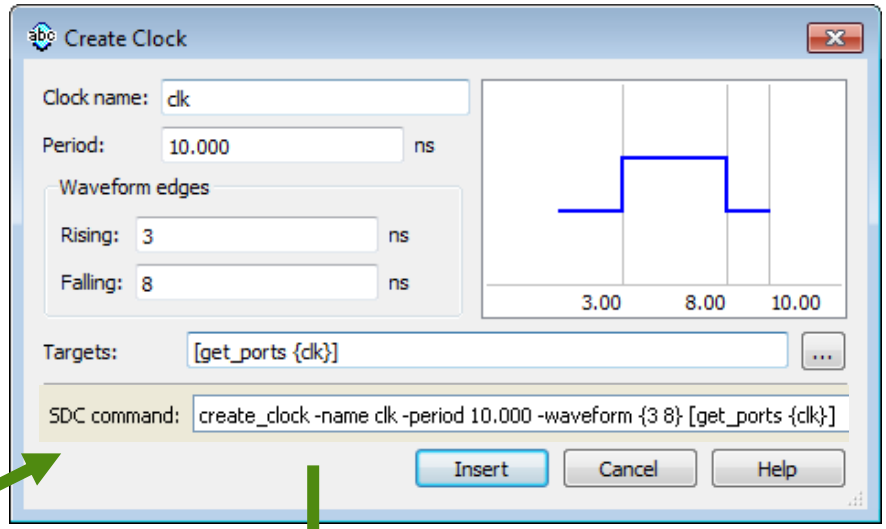


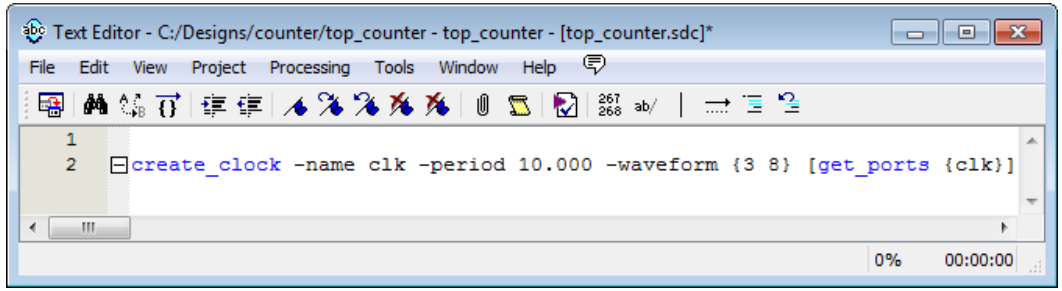
Place cursor over command to see tooltip

SDC File Editor (cont.)



Construct an SDC file using TimeQuest graphical constraint creation tools

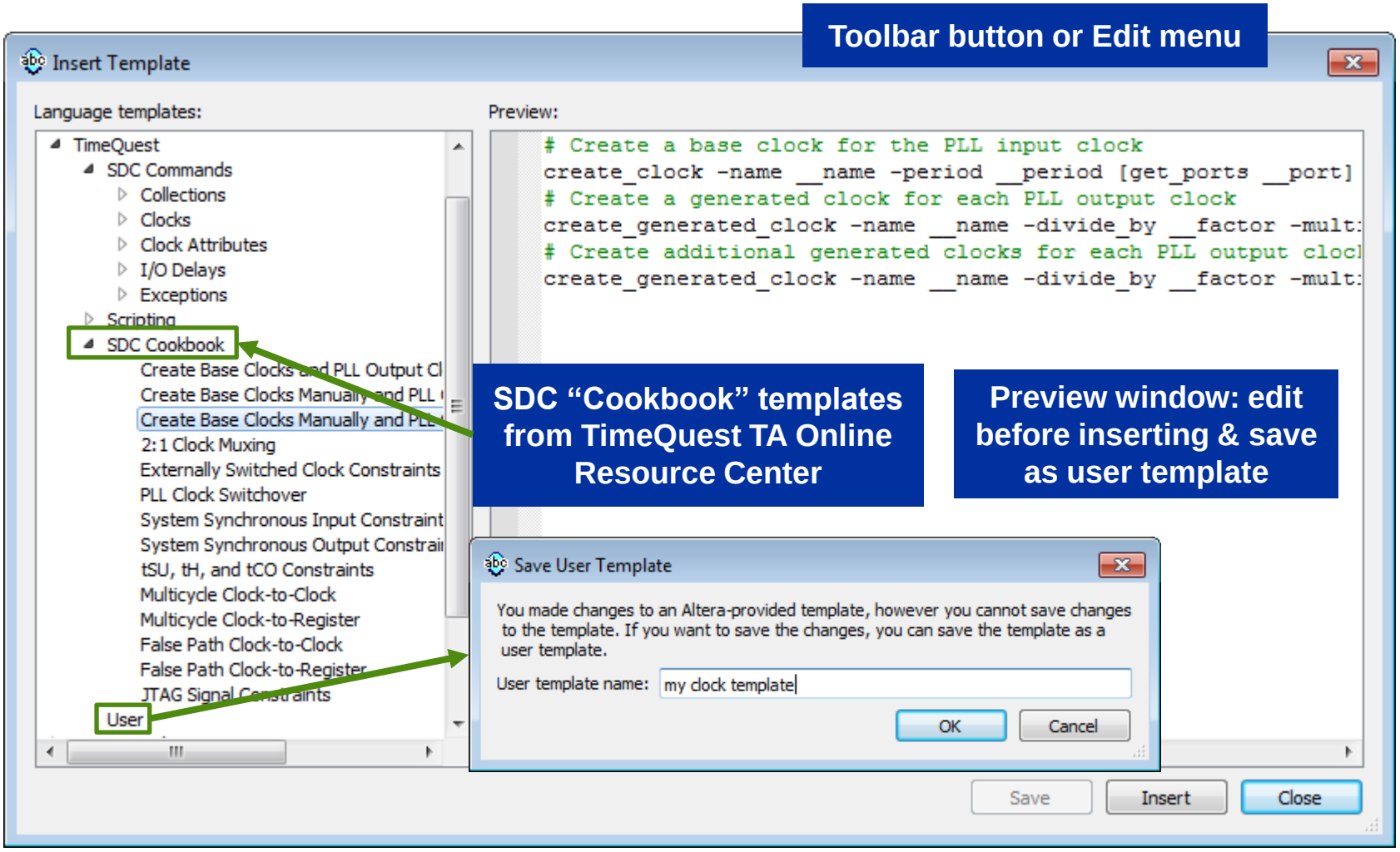




Constraints inserted at cursor location

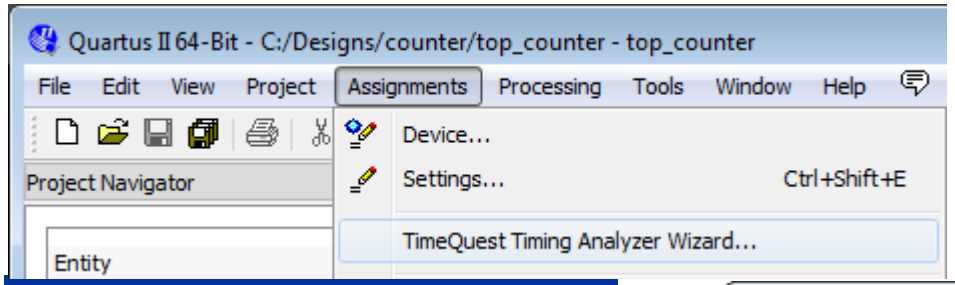
SDC Templates

- Quickly add customized constraint templates

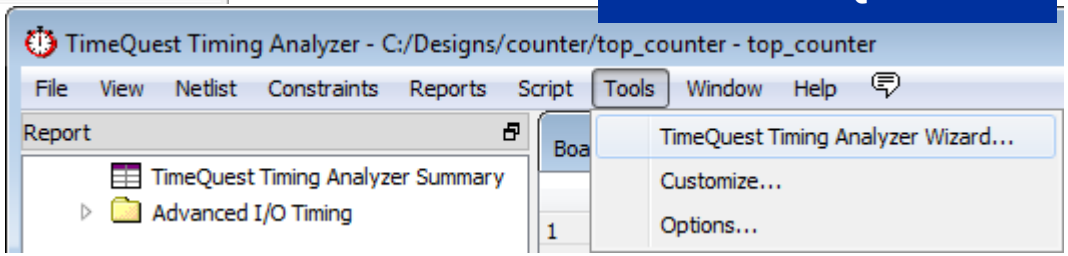


TimeQuest Timing Analyzer Wizard

- Easily create complete SDC file by entering:
  - Clock constraints
  - T_{su} and T_h requirements for input pins
  - $T_{co(max\ and\ min)}$ requirements for output pins
  - $T_{pd(max\ and\ min)}$ pin-to-pin delay requirements for combinatorial logic
- For quickly creating a simple SDC file and for those familiar with the Classic Timing Analyzer



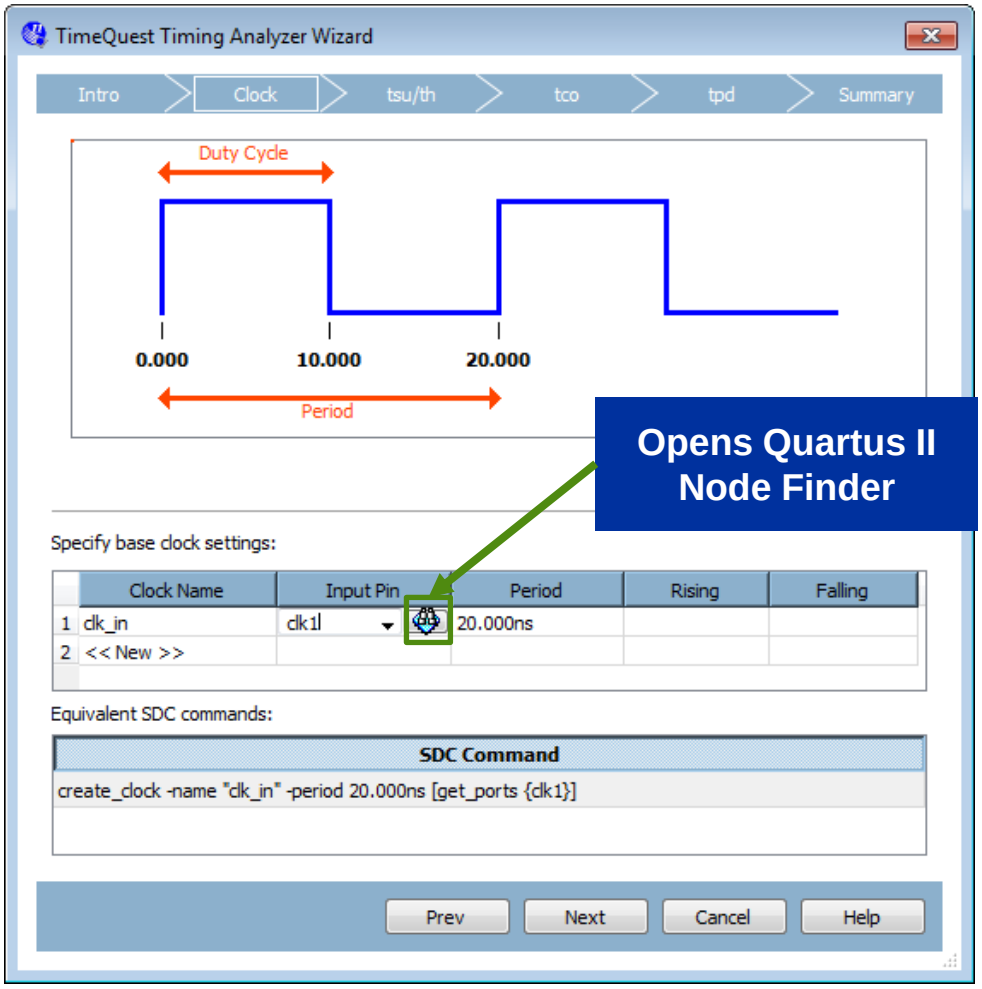
From main Quartus II window



From TimeQuest TA

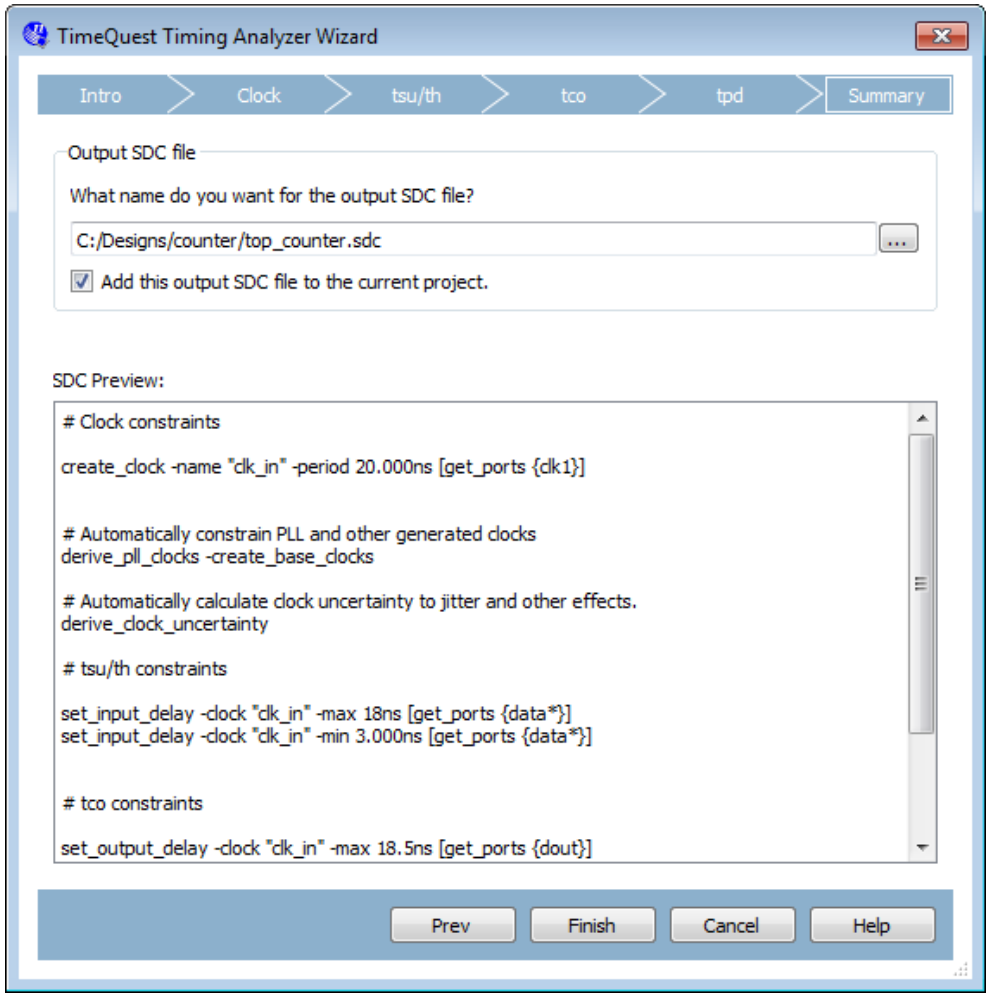
Using the Wizard

- Double-click cells to edit
- Enter names, signals, and constraint values, similar to text editor GUI dialog boxes
- Waveforms and equivalent SDC update on-the-fly
- Clocks: PLL-generated clocks constrained automatically



Using the Wizard (cont.)

- Enter constraints for I/O and propagation through device
- Summary tab displays completed SDC file
- Write out file and add to project, if desired



Basic Steps to Using TimeQuest TA

1. Generate timing netlist
2. Enter SDC constraints
   a) Create and/or read in SDC file (recommended method)

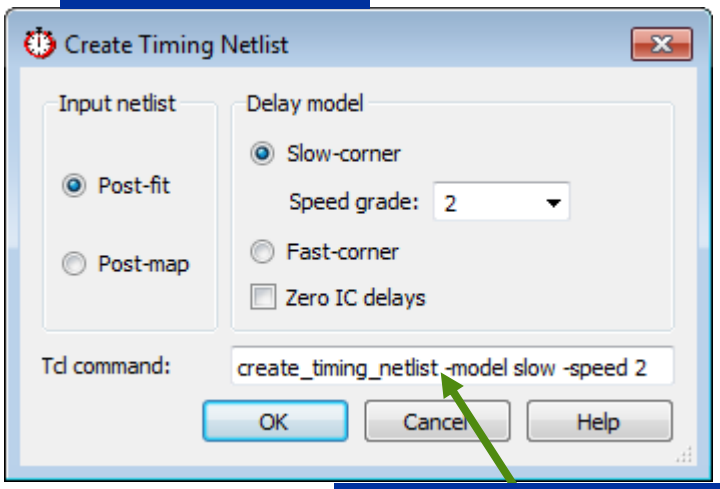
   or

   b) Constrain design directly in console
3. Update timing netlist
4. Generate timing reports
5. Save timing constraints (optional)

1) Generate Timing Netlist

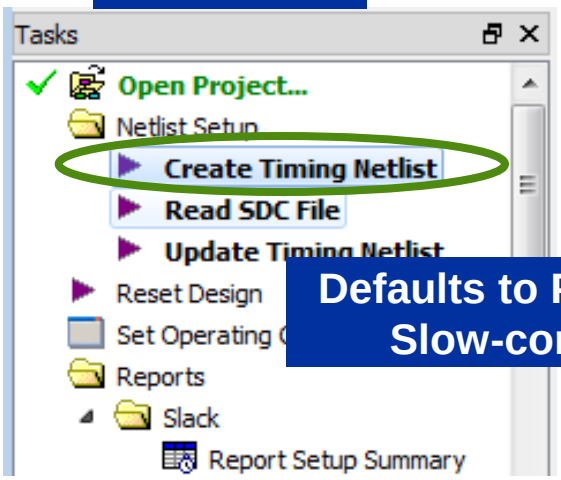
- Create a timing netlist (i.e. database) based on compilation results
  - **Post-map** (post-synthesis) or **post-fit** (if design already fully compiled)
  - Worst-case (slow; maximum operating temperature), best-case (fast; minimum operating temperature) timing models
  - Set custom operating conditions (65 nm technology devices; military; industrial, etc.)
- To execute:

Netlist menu



Tcl equivalent of command

Tasks pane

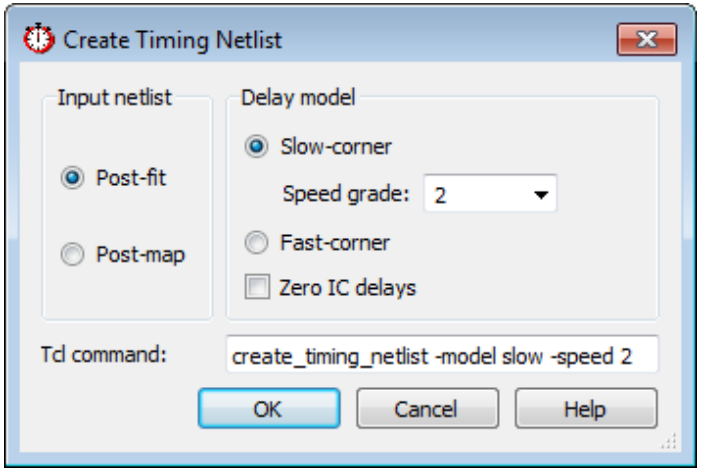


Defaults to Post-fit, Slow-corner

Tcl: create_timing_netlist

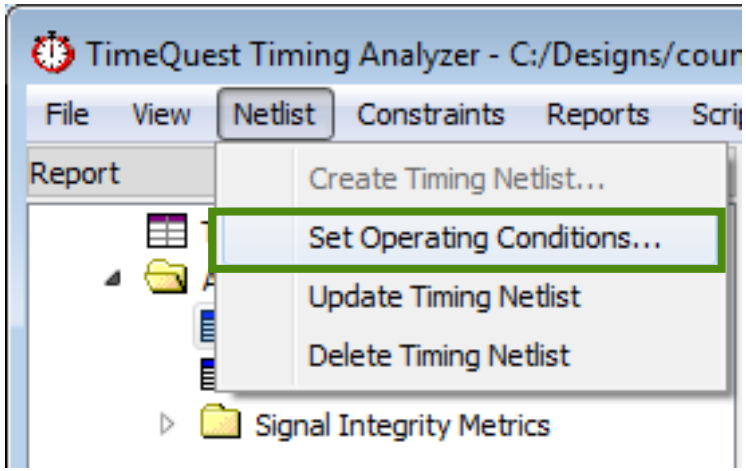
Generating Fast/Slow Netlist

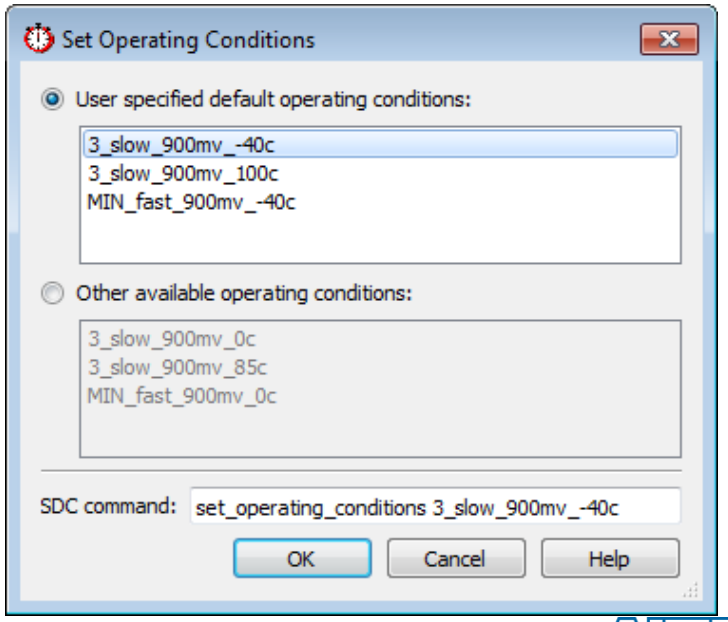
- Specify one of the timing models to be used when creating your netlist
- Default is the slow timing netlist
- To specify fast timing netlist:
  - Use `-model fast` option with `create_timing_netlist` command
  - Choose **Fast-corner** in GUI when executing **Create Timing Netlist** from **Netlist** menu
  - CANNOT select fast corner from Tasks Pane



Specifying Operating Conditions

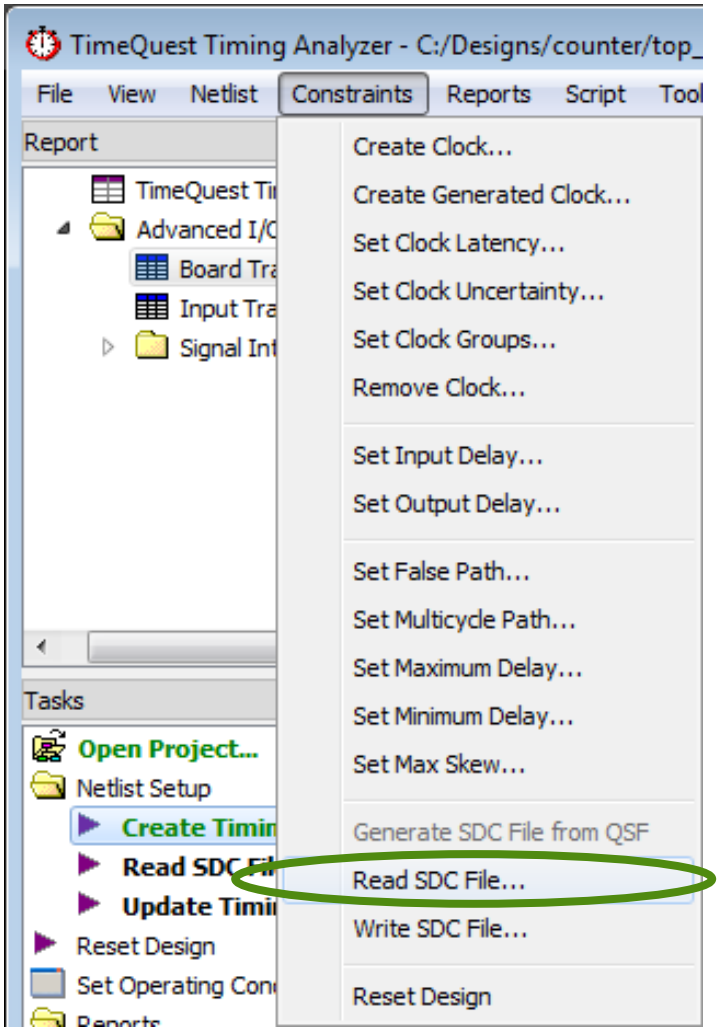
- Perform timing analysis for different delay models without recreating the existing timing netlist
- Takes precedence over already generated netlist
- Required for selecting slow, min. temp. model and other models (industrial, military, etc.) depending on device
- Use `get_available_operating_conditions` in Console or scripts to see available conditions for target device





2a) Create or Read in SDC File

- Create SDC file using SDC file editor
  - ***Don't*** enter constraints using **Constraints** menu
- Read in constraints & exceptions from existing SDC file(s)
- Execution
  - **Read SDC File** (**Tasks** pane or **Constraints** menu)
- File precedence (if no filename specified) in order (high → low)
  - Files specifically added to Quartus II project
  - <*current_revision*>.sdc (if it exists in project directory)

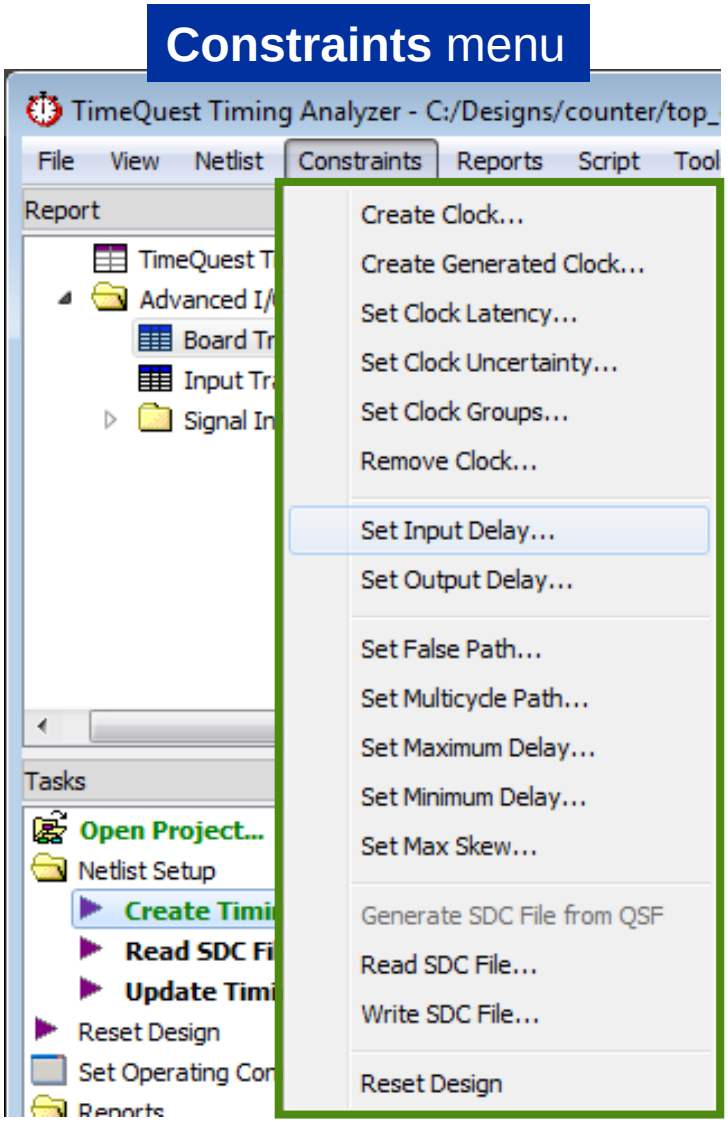


Tcl: read_sdc [<filename>]

2b) Constrain Directly in Console

- Apply new constraints directly to netlist with console SDC commands or from the **Constraints** menu
  - Not automatically added to SDC file
  - Not needed if all constraints in SDC file
- Use **`remove_`**<*command*> to remove applied constraints
  - Only **`remove_clock`** is in GUI
- *Recommend using SDC file (step 2a) instead to ease management and storage of constraints*

Using GUI to Enter Constraints Directly



- Most common constraints can be accessed from the **Constraints** menu
- Same as **Edit** menu ⇒ **Insert Constraints** in SDC file editor
- Use if unfamiliar with SDC syntax

Constraining

- User MUST enter constraints for all paths to fully analyze design
  - Timing analyzer only performs slack analysis on constrained design paths
  - Constraints guide the fitter to place & route design in order to meet timing requirements
  - Recommendation: Constrain ***all*** paths (at least clocks & I/O)
- Not as difficult a task as it may sound
  - Wildcards
  - Single, generalized constraints cover many paths, even all paths in an entire clock domain

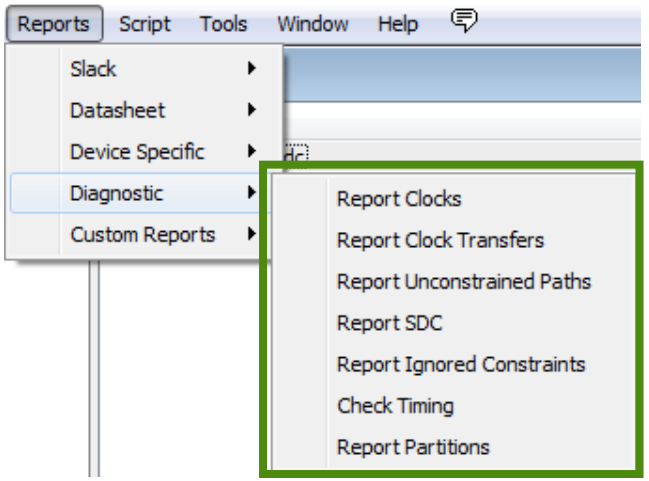
3) Update Timing Netlist

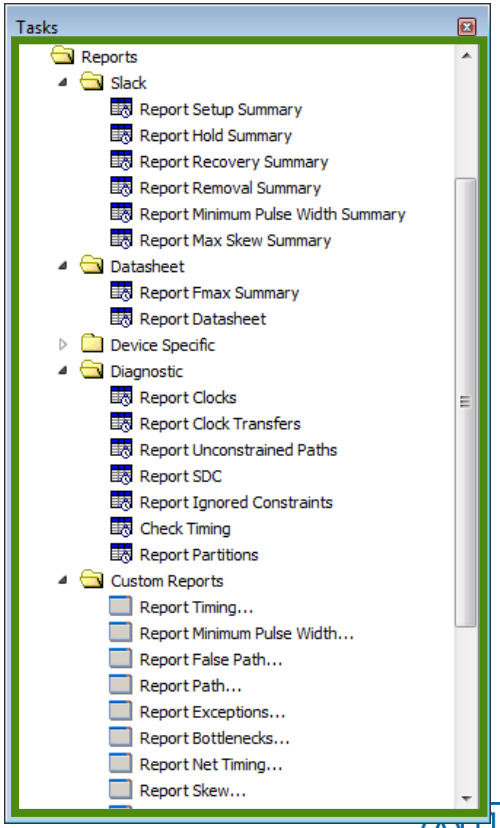
- Apply SDC constraints/exceptions to current timing netlist
- Generates warnings
  - Undefined clocks
  - Partially defined I/O delays
  - Combinational loops
- Update timing netlist after adding any new constraint
- Execution
  - **Update Timing Netlist** (**Tasks** pane or **Netlist** menu)

Tcl: update_timing_netlist

4) Generate Timing Reports

- Verify timing requirements and locate violations
- Check for fully constrained design or ignored timing constraints
- Two methods
  - **Tasks** pane
    - *Shortcut*: Automatically creates/updates netlist & reads default SDC file if needed
  - **Reports** menu
    - Must have valid netlist to access

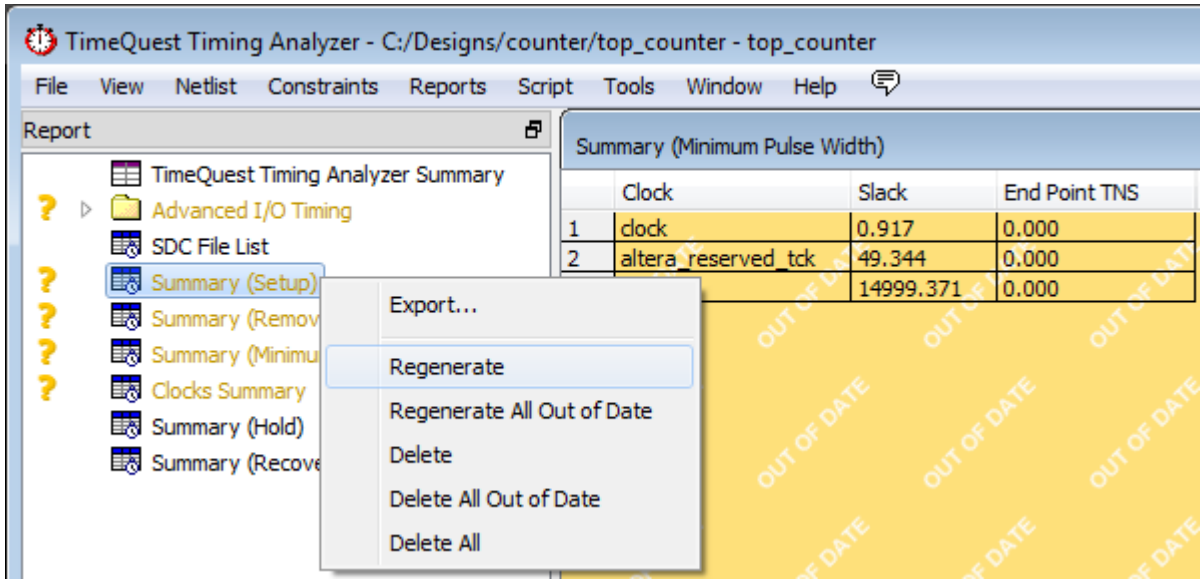




Double-click to create individual report

“Out of Date” Reports

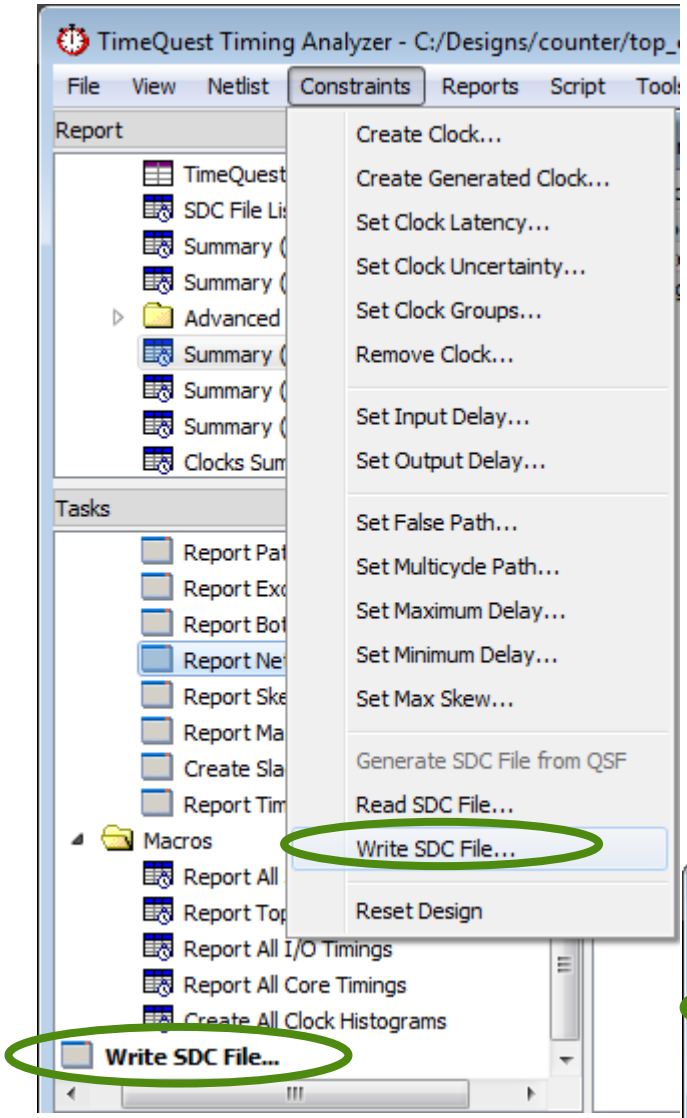
- Adding new constraints interactively in console causes current reports to be “out of date”
- Update timing netlist & regenerate reports (Report pane right-click menu)
- No such warning when using SDC file



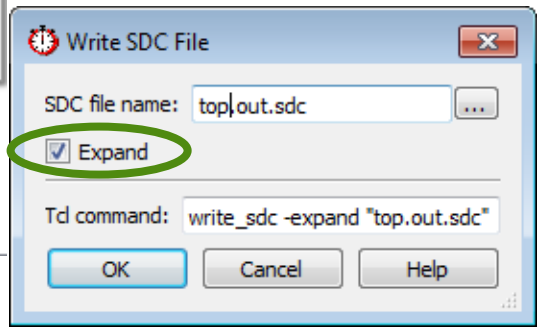
Reset Design Command

- **Tasks** pane or **Constraints** menu
- Flushes all timing constraints from current timing netlist and recreates original netlist
  - Functional Tcl equivalent: `delete_timing_netlist` command followed by `create_timing_netlist`
- Uses
  - "Re-starting" timing analysis on same timing netlist applying different constraints or SDC file
  - Starting analysis over if results seem to be unexpected

5) Save Timing Constraints (Optional)



- `write_sdc` command
  - Saves all constraints & exceptions applied to current netlist into SDC file
  - Use if constraints added during TimeQuest session in console instead of SDC file
- Notes
  - SDC files generated by TimeQuest TA only if requested
  - Use `-expand` option to convert Altera-specific SDC commands *(discussed later)* into standard SDC
  - Run `report_sdc` command (console, Tasks pane, or **Report** menu) to see what will get written to SDC file



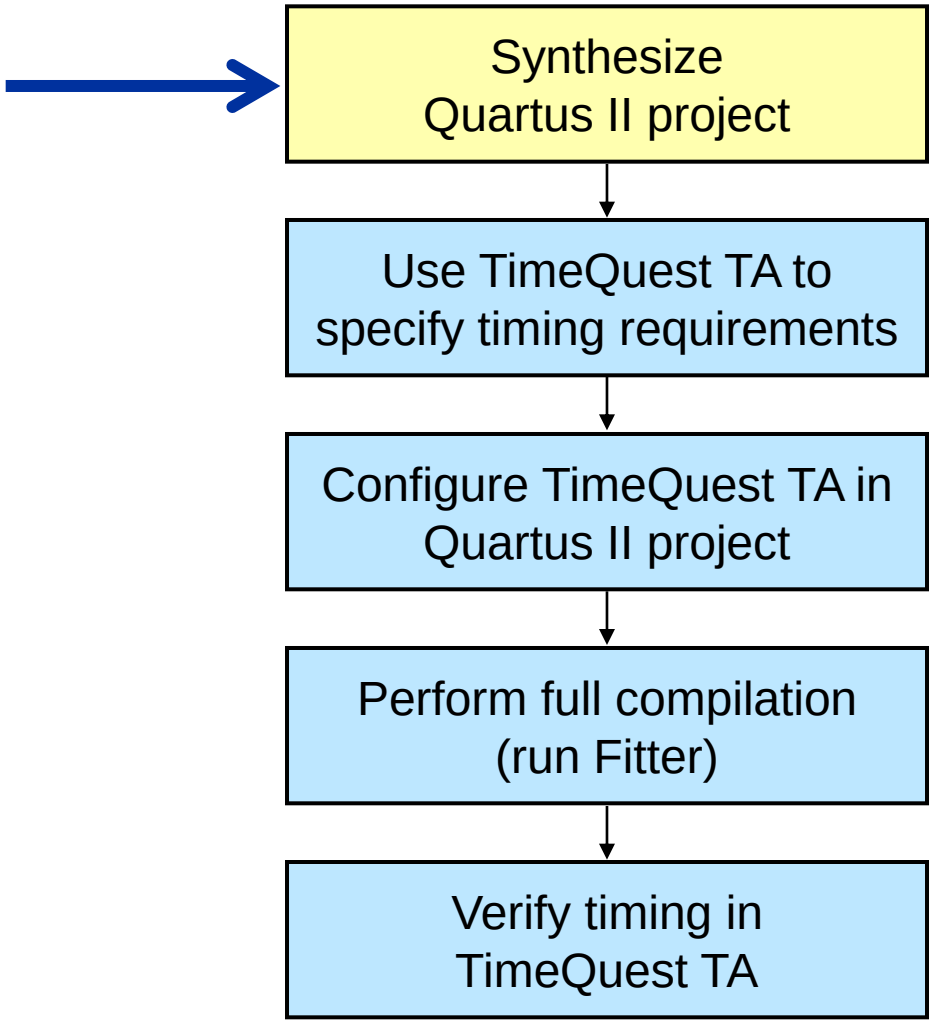
Basic Steps to Using TimeQuest TA (Review)

1. Generate timing netlist
2. Enter SDC constraints
   a) Create and/or read in SDC file (recommended method)

   or

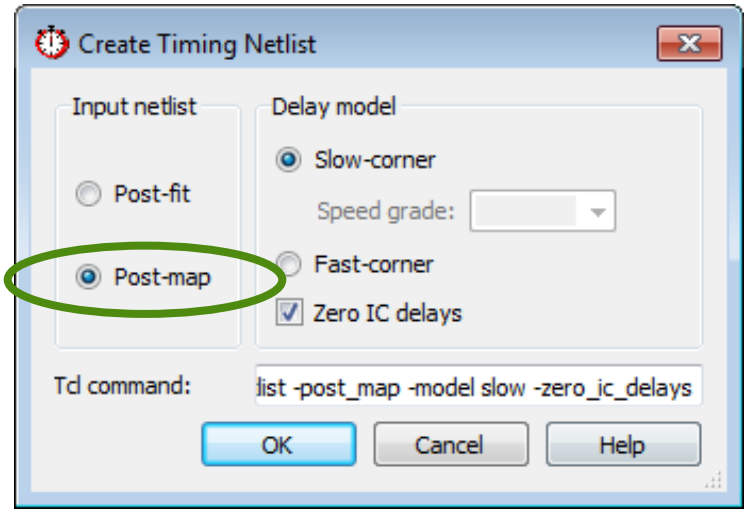
   b) Constrain design directly in console
3. Update timing netlist
4. Generate timing reports
5. Save timing constraints (optional)

Using TimeQuest TA in Quartus II Flow

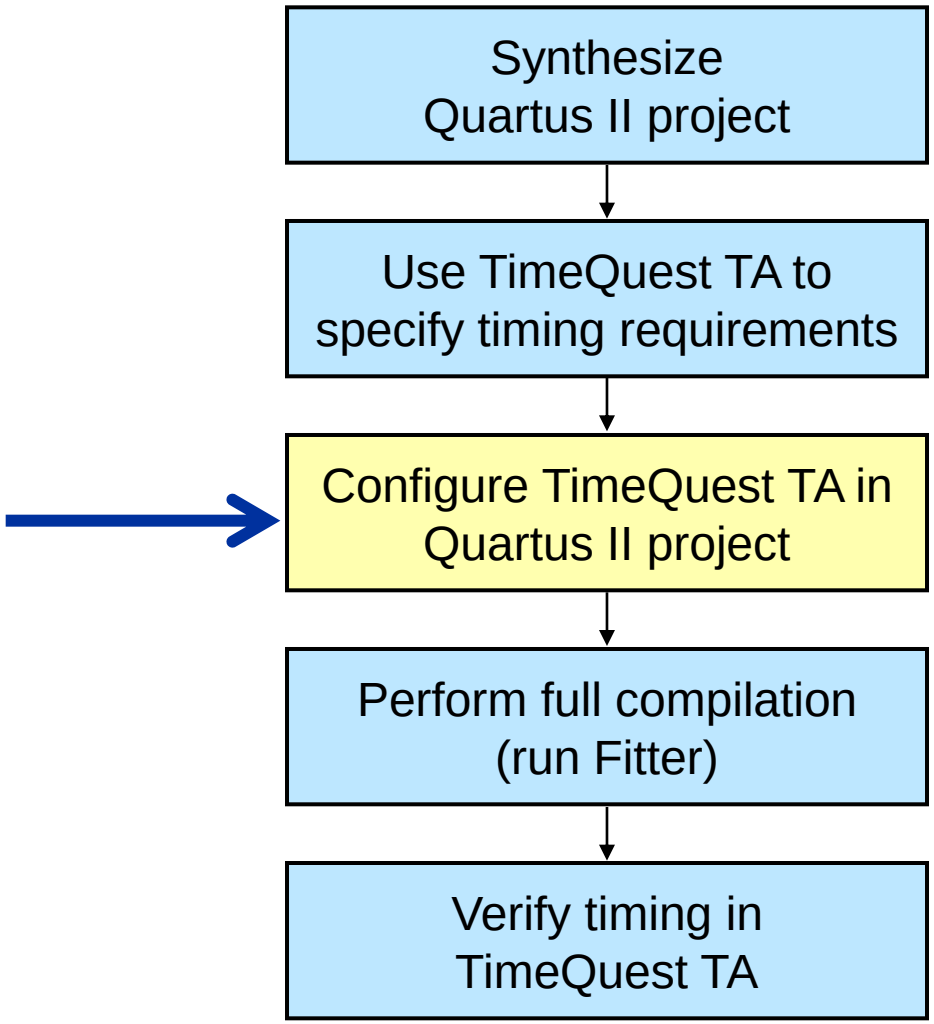


Timing Requirements: Create Post-Map Netlist

- Follow TimeQuest flow
- Use `-post_map` argument for synthesis (mapping) only netlist
  - If design already fully compiled, choose `-post_fit` (default)
- Tasks list command defaults to post-fit, so must use **Netlist** menu in GUI
- **Zero IC delays** auto-enabled with **Post-map**
  - Assumes no interconnect delays to determine if it will be possible to meet timing



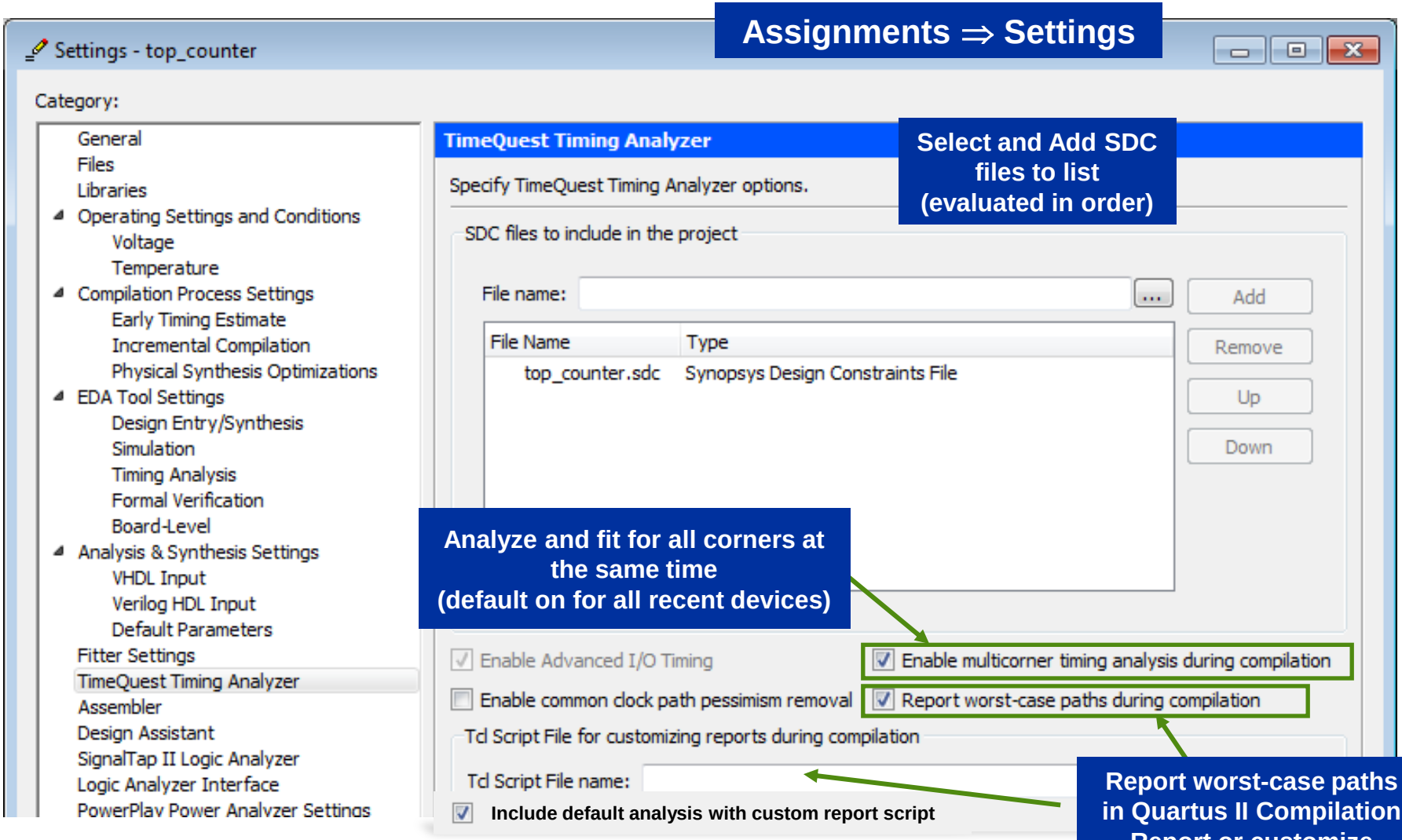
Using TimeQuest TA in Quartus II Flow



Configure TimeQuest TA in Quartus II Software

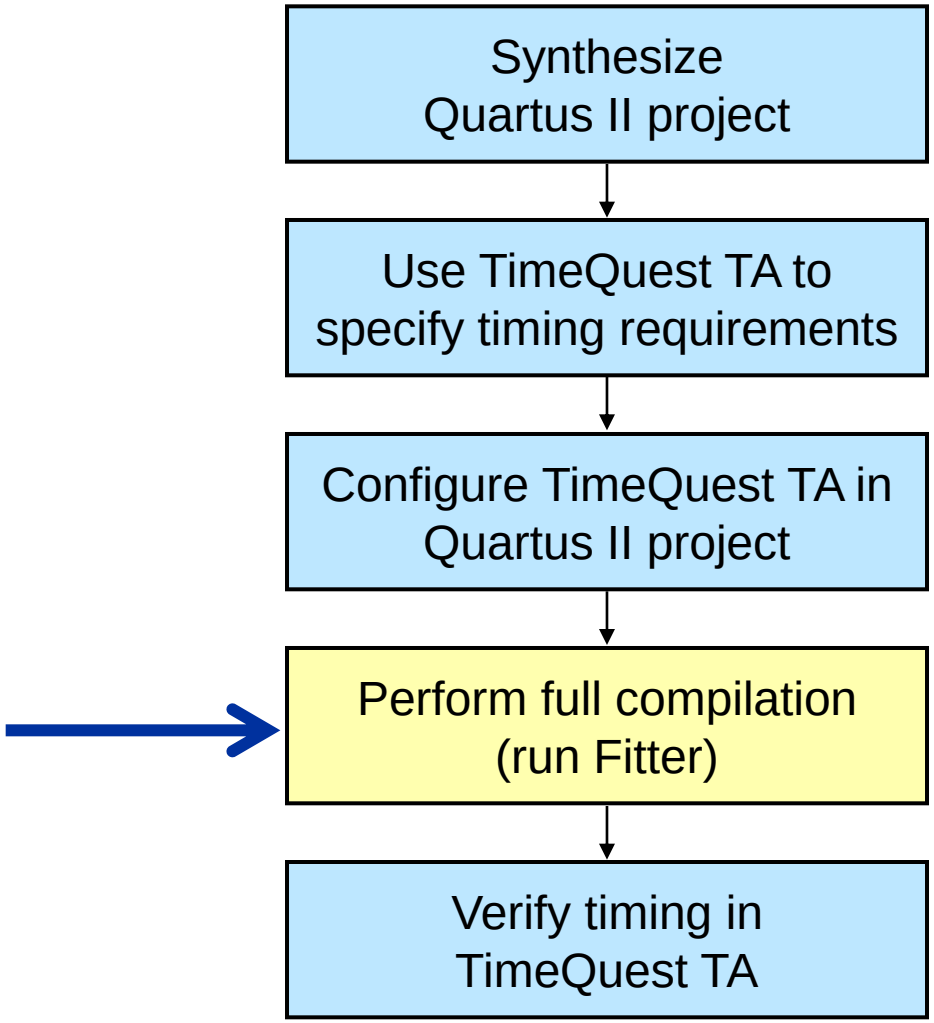
- Tells the Quartus II software to use SDC constraints during fitting
- File order precedence (high → low)
  - Any SDC files manually added to Quartus II project (in the order they appear in the files list)
  - <*current_revision*>.SDC located in project directory

Quartus II TimeQuest Settings



Note: Advanced I/O Timing & Common Clock Path Pessimism (CCPP) removal discussed later

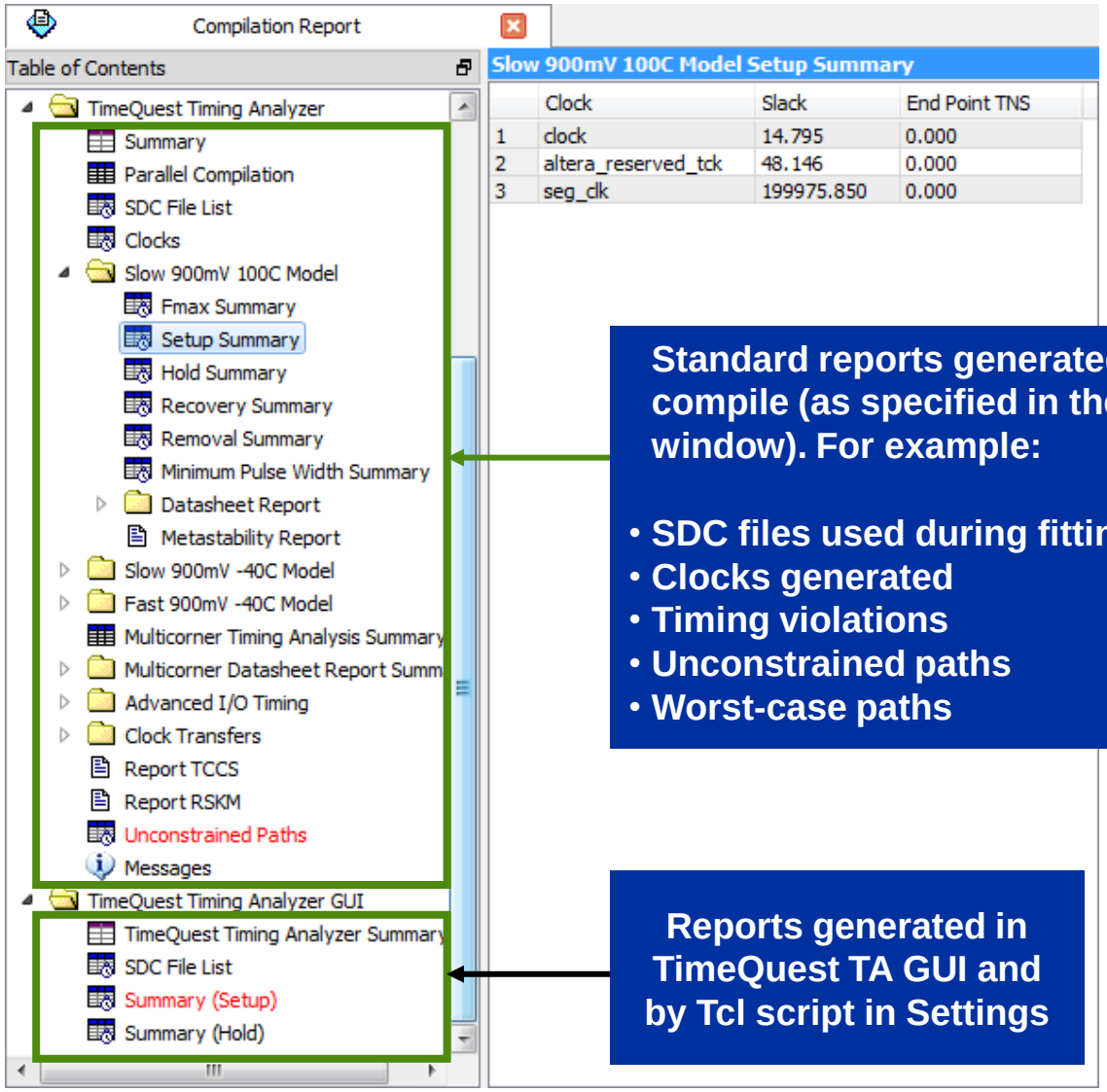
Using TimeQuest TA in Quartus II Flow



Verifying Timing Requirements

- View TimeQuest summary information directly in Quartus II Compilation Report
- Open TimeQuest TA for more thorough analysis
  - Follow TimeQuest flow, selecting **Post-fit** netlist
    - Optional: Enable **Zero IC Delays** to see if there is any chance of meeting timing without having to enable optimization options
  - Run TimeQuest easy-to-use reporting capabilities (**Tasks** pane)
  - Many different reporting options available
  - Place Tcl reporting commands into script file
    - Easy repetition
- ***Verify whether Fitter was able to meet timing requirements***

TimeQuest Reports in Compilation Report



Standard reports generated during compile (as specified in the Settings window). For example:

- **SDC files used during fitting**
- **Clocks generated**
- **Timing violations**
- **Unconstrained paths**
- **Worst-case paths**

Reports generated in TimeQuest TA GUI and by Tcl script in Settings

Please go to Exercise 1

Quartus® II Software Design Series: Timing Analysis

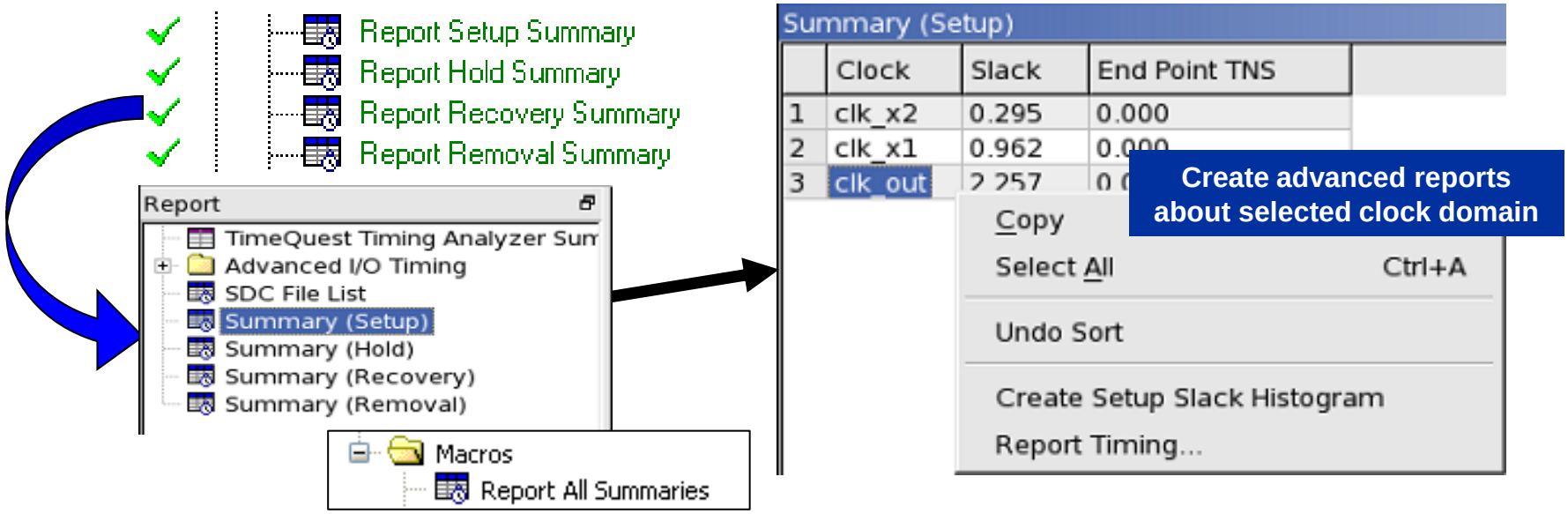
Timing Reports

Timing Reports

- Timing results available in both the Quartus II Compilation Report and TimeQuest GUI
- TimeQuest TA includes more extensive reporting capabilities
- Create reports while creating constraints (**post-map** netlist) before fitting to see if design can meet timing requirements
- Create reports after fitting (**post-fit** netlist) to verify that placed & routed design meets timing requirements

Summary Reports

- Simplest, most common type of timing report
- Each row reports on a clock domain in the design
  - Worst case (positive or negative slack) listed first
  - If negative, total negative slack (TNS) on all edges in clock domain
- Command: `create_timing_summary`
  - `-setup`, `-hold`, `-recovery`, `-removal`: create report for selected analysis type



Detailed Slack/Path Analysis

- Create more specific/detailed reports
  - Ex. Details on a specific clock domain
  - Ex. View timing paths between particular I/O & registers
- Create using Tcl commands or GUI
  - Use GUI to see report immediately
  - Use Tcl file for repeatability

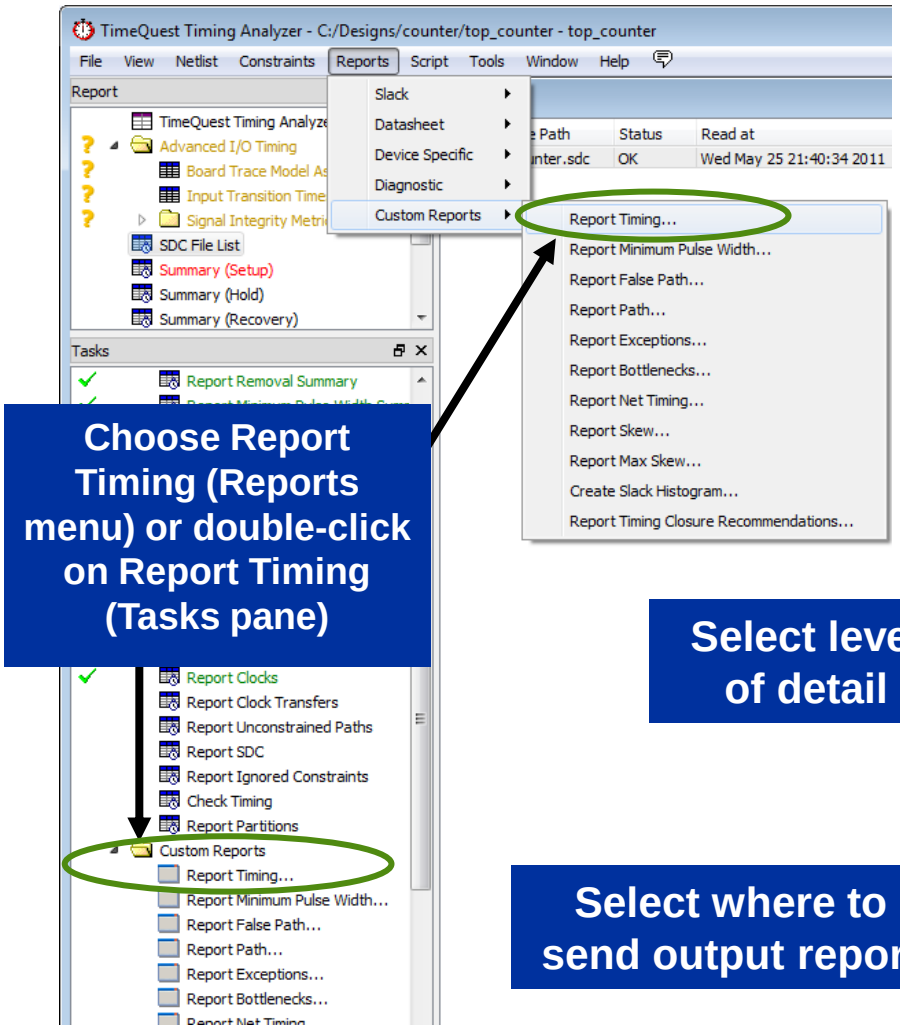
Advanced Reporting: Report Timing

```
report_timing
        -from <source_nodes>
        -from_clock <source_clock_names>
        -rise_from_clock <source_clock_names>
        -fall_from_clock <source_clock_names>
        -through <thru_node>
        -to <destination_nodes>
        -to_clock <destination_clock_names>
        -rise_to_clock <destination_clock_names>
        -fall_to_clock <destination_clock_names>
        -setup|-hold|-recovery|-removal
        -detail <summary|path_only|path_and_clock|full_path>
        -file <file_name>
        -append
        -panel_name <report_name>
        -stdout
        -less_than_slack <slack_limit>
        -npaths <#_of_paths_to_display>
        -nworst <max_#_of_paths_per_endpoint>
        -false_path
        -pairs_only
        -show_routing
```

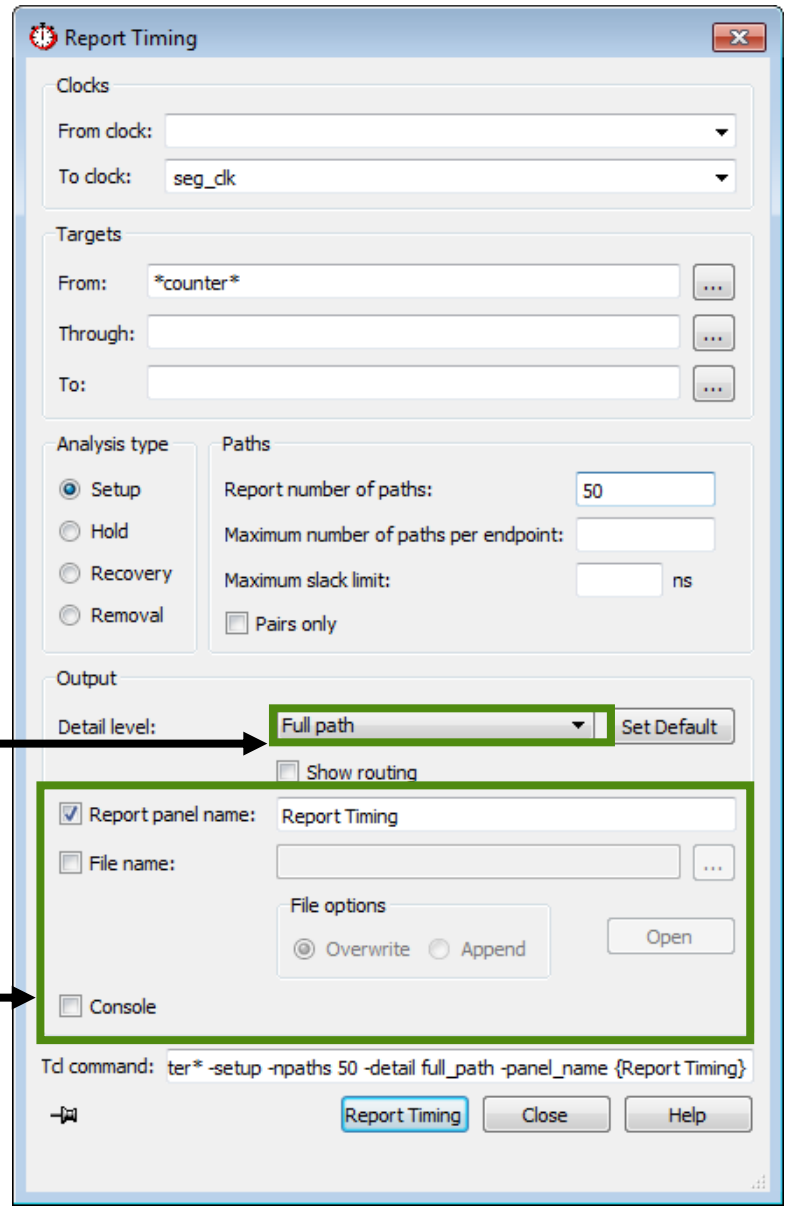
`report_timing` Arguments

- `-setup|hold|recovery|removal` are mutually exclusive
  - Default is `-setup`
- `-detail <option>` selects report clock path detail level
  - `path_only`: lumps clock network delay together
  - `summary`: lists individual path (condense path report)
  - `path_and_clock`: shows clock network delay in detail
  - `full_path`: shows full clock and data networks in more detail, particularly generated clock (default option)
- `-npaths`: number of paths to report; one path shown if not specified

Report Timing (GUI)



Choose Report Timing (Reports menu) or double-click on Report Timing (Tasks pane)



Select level of detail

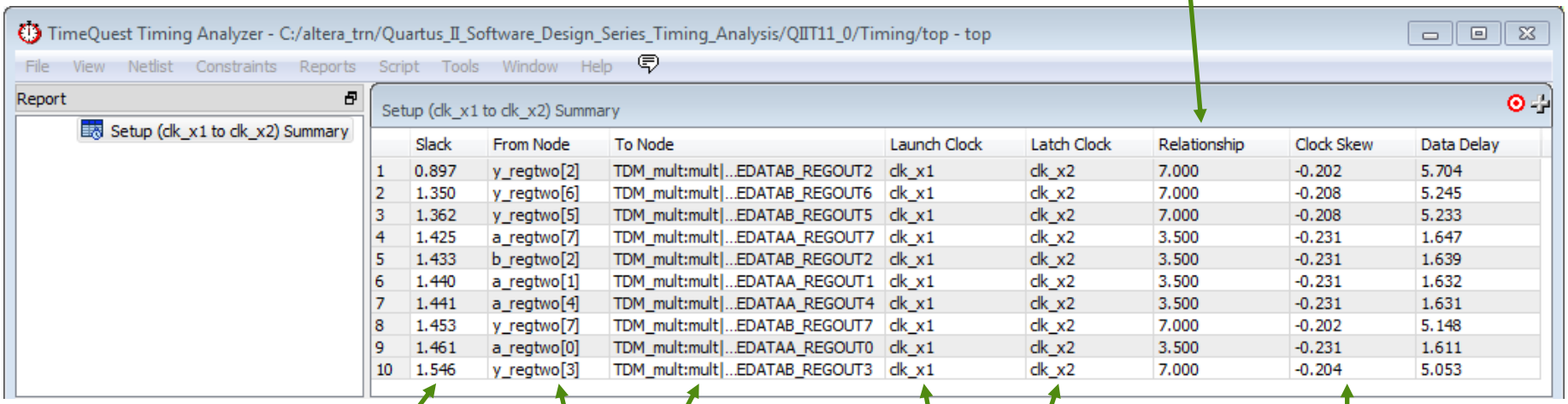
Select where to send output report

Tcl: report_timing

Summary Slack/Path Report

```
report_timing -from_clock clk_x1 -to_clock clk_x2 \
        -setup -npaths 10 -detail summary \
        -panel_name "Setup (clk_x1 to clk_x2) Summary"
```

Analysis relationship (may be adjusted by exceptions discussed later)



	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	0.897	y_regtwo[2]	TDM_mult:mult|...EDATAB_REGOUT2	clk_x1	clk_x2	7.000	-0.202	5.704
2	1.350	y_regtwo[6]	TDM_mult:mult|...EDATAB_REGOUT6	clk_x1	clk_x2	7.000	-0.208	5.245
3	1.362	y_regtwo[5]	TDM_mult:mult|...EDATAB_REGOUT5	clk_x1	clk_x2	7.000	-0.208	5.233
4	1.425	a_regtwo[7]	TDM_mult:mult|...EDATAA_REGOUT7	clk_x1	clk_x2	3.500	-0.231	1.647
5	1.433	b_regtwo[2]	TDM_mult:mult|...EDATAB_REGOUT2	clk_x1	clk_x2	3.500	-0.231	1.639
6	1.440	a_regtwo[1]	TDM_mult:mult|...EDATAA_REGOUT1	clk_x1	clk_x2	3.500	-0.231	1.632
7	1.441	a_regtwo[4]	TDM_mult:mult|...EDATAA_REGOUT4	clk_x1	clk_x2	3.500	-0.231	1.631
8	1.453	y_regtwo[7]	TDM_mult:mult|...EDATAB_REGOUT7	clk_x1	clk_x2	7.000	-0.202	5.148
9	1.461	a_regtwo[0]	TDM_mult:mult|...EDATAA_REGOUT0	clk_x1	clk_x2	3.500	-0.231	1.611
10	1.546	y_regtwo[3]	TDM_mult:mult|...EDATAB_REGOUT3	clk_x1	clk_x2	7.000	-0.204	5.053

Calculated slack

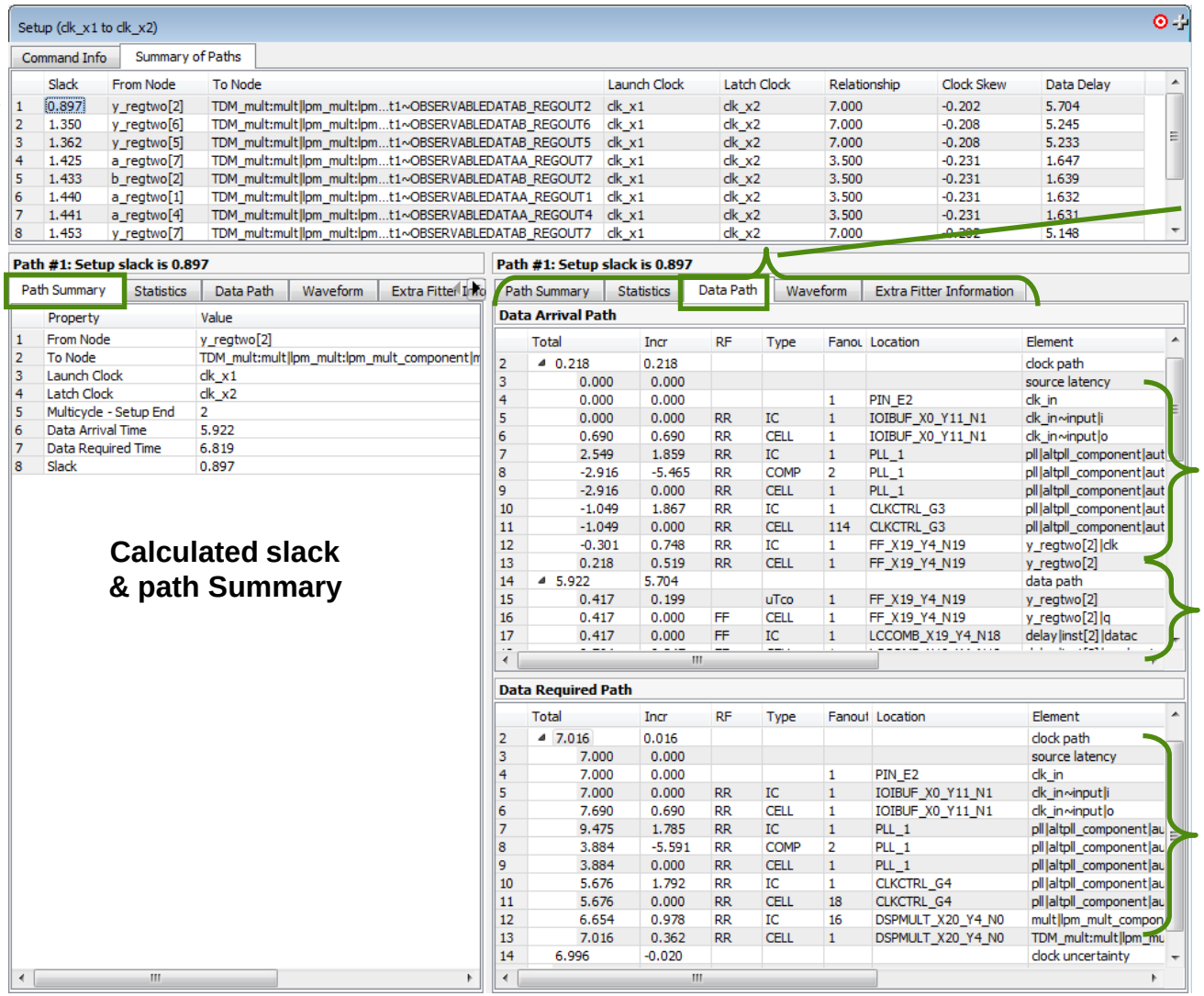
Source & destination nodes

Source (launch) & destination (latch) clocks

Skew between clock arrival at source and destination

Detailed Slack/Path Report

```
report_timing -from_clock clk_x1 -to_clock clk_x2 -setup -npaths 10 \
        -detail full_path -panel_name "Setup (clk_x1 to clk_x2)"
```



5 detailed views of path available

Calculated slack & path Summary

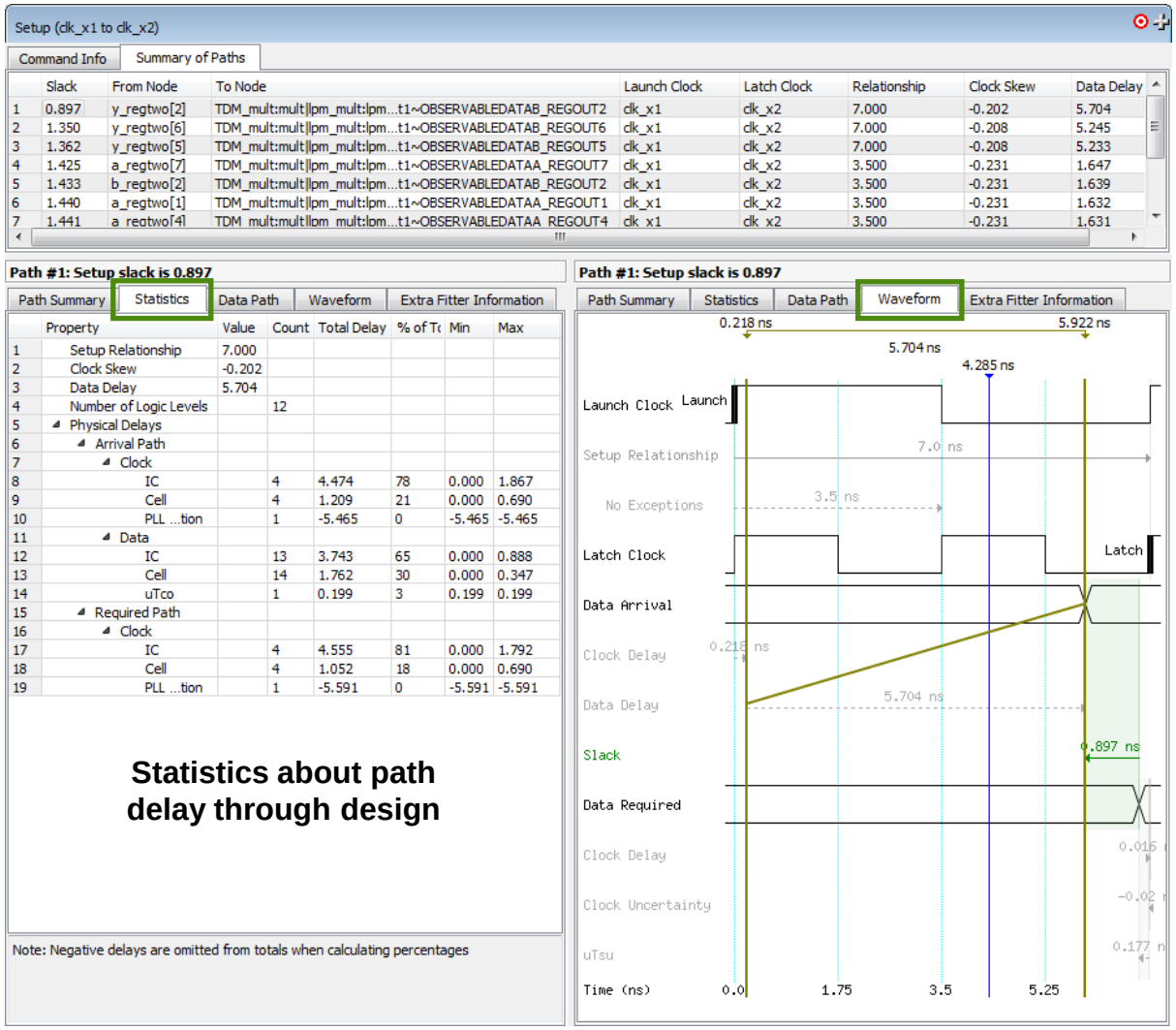
Clock path "hops" to launch register

Data arrival path "hops" to latch register

Clock path "hops" to latch register

Detailed Slack/Path Report (cont.)

```
report_timing -from_clock clk_x1 -to_clock clk_x2 -setup -npaths 10 \
    -detail full_path -panel_name "Setup (clk_x1 to clk_x2)"
```



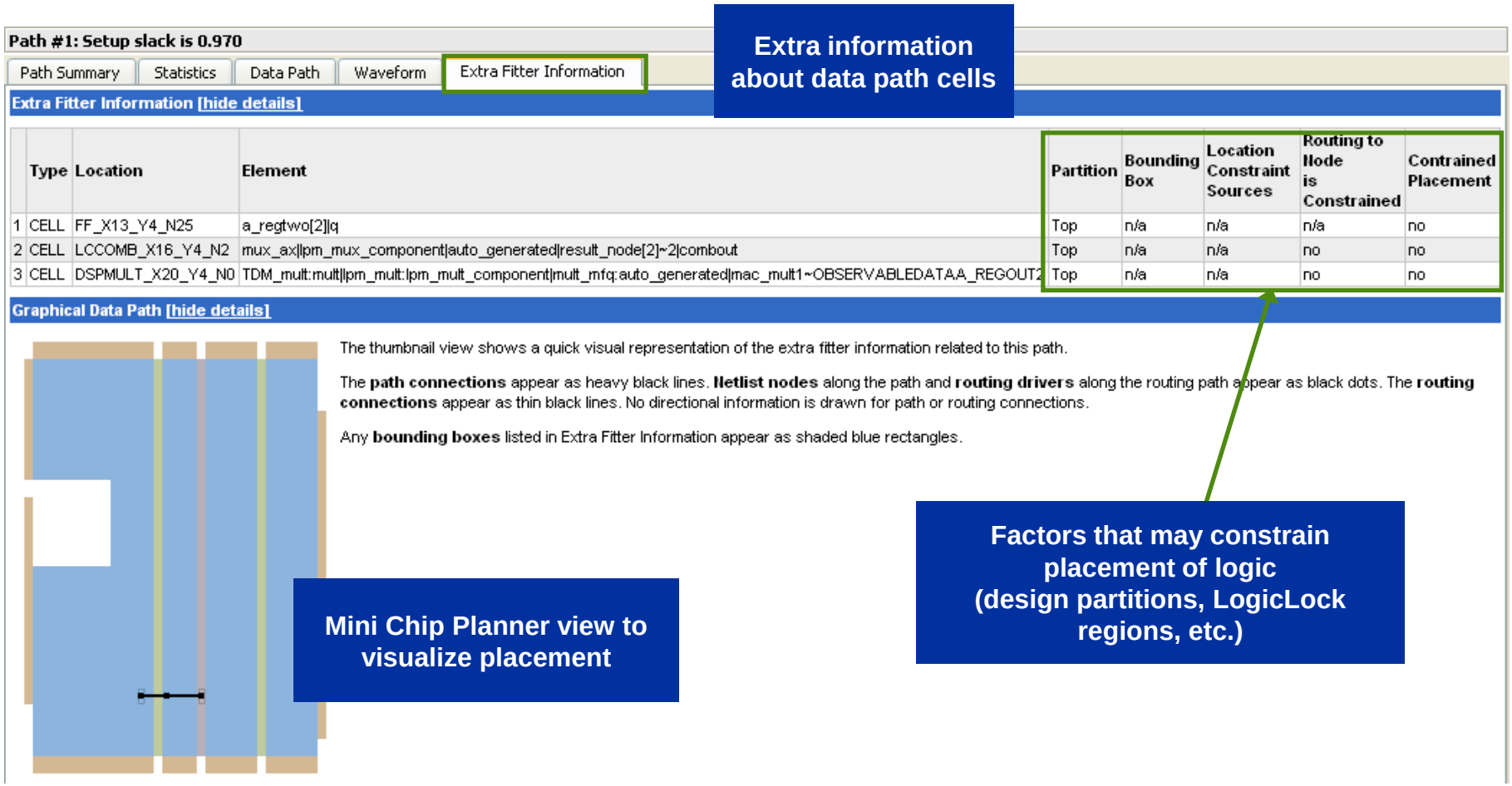
Statistics about path delay through design

Waveform visualizes TimeQuest slack calculations

Click and drag cursors that "snap" to path timing events

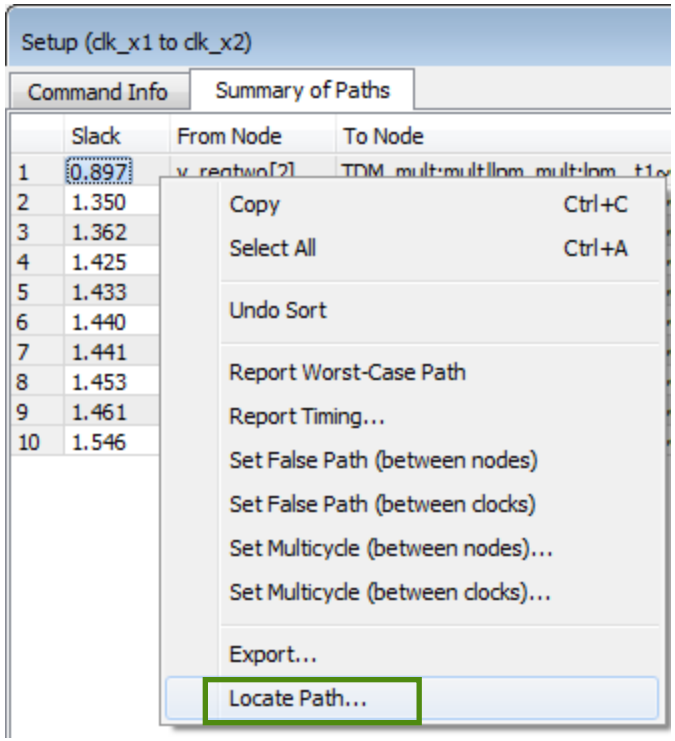
Detailed Slack/Path Report (cont.)

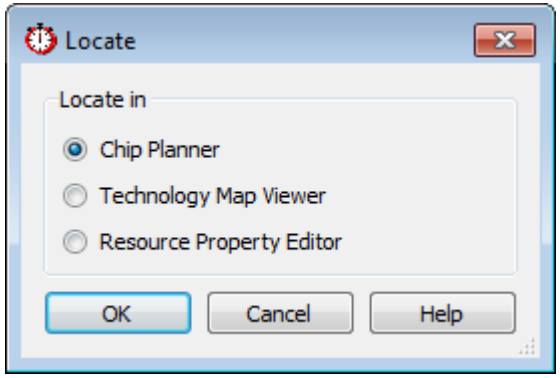
```
report_timing -from_clock clk_x1 -to_clock clk_x2 -setup -npaths 10 \
        -detail full_path -panel_name "Setup (clk_x1 to clk_x2)"
```



Further Path Analysis

- Right-click path(s) to cross-probe to other Quartus II tools or design files
- `locate` command in Console

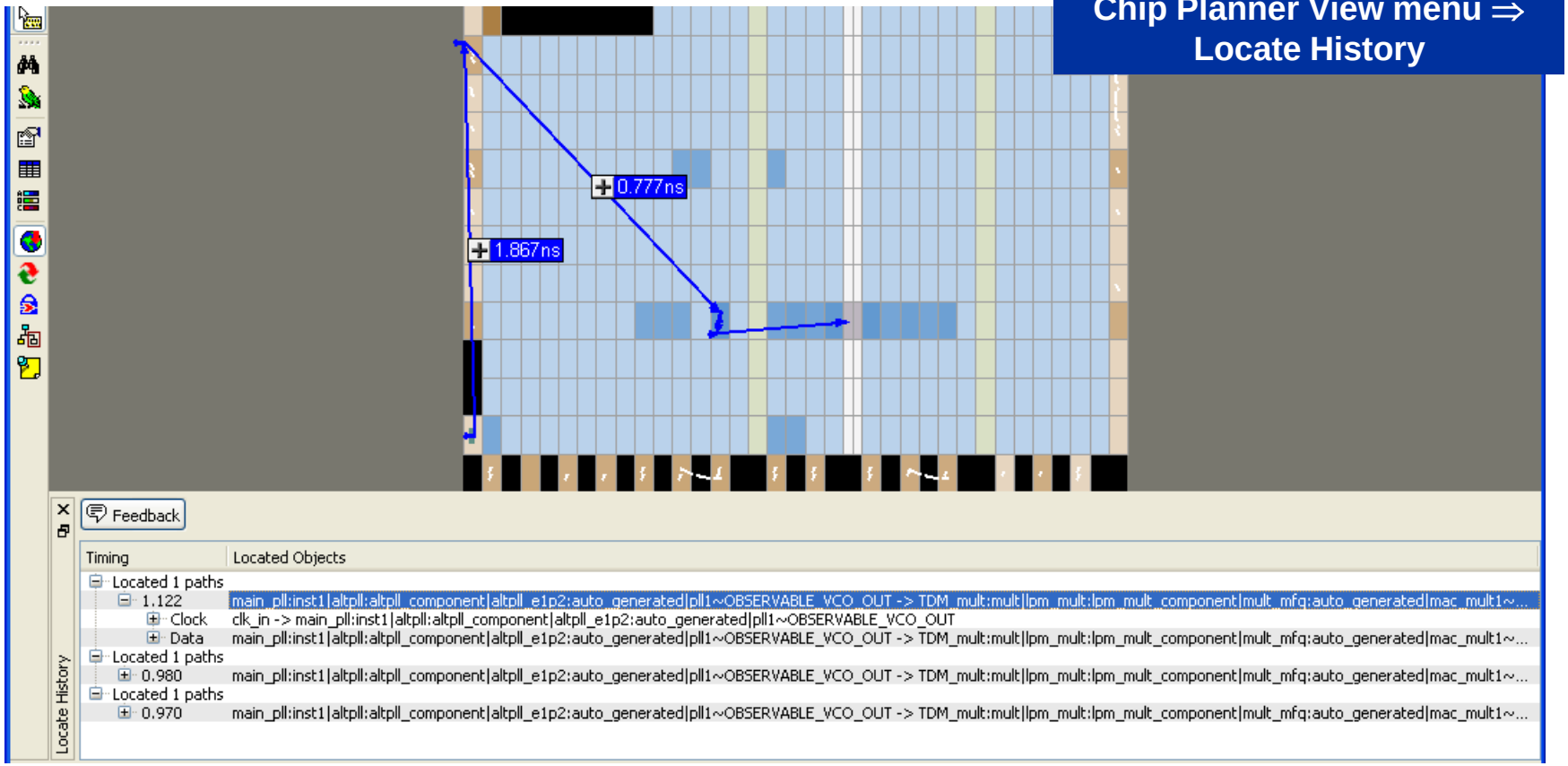




Locate History in Chip Planner

- Keep track of paths analyzed in Chip Planner
- Quickly highlight or re-highlight whole, clock, or data portions of Data Arrival Path

Chip Planner View menu ⇒ Locate History



Timing Closure Recommendations

- Heuristically-derived recommendations for design or settings adjustments on failing paths in order to meet timing (*for setup analysis only*)
- Filter analysis with dialog similar to **Report Timing**
- Good starting point for closing timing