

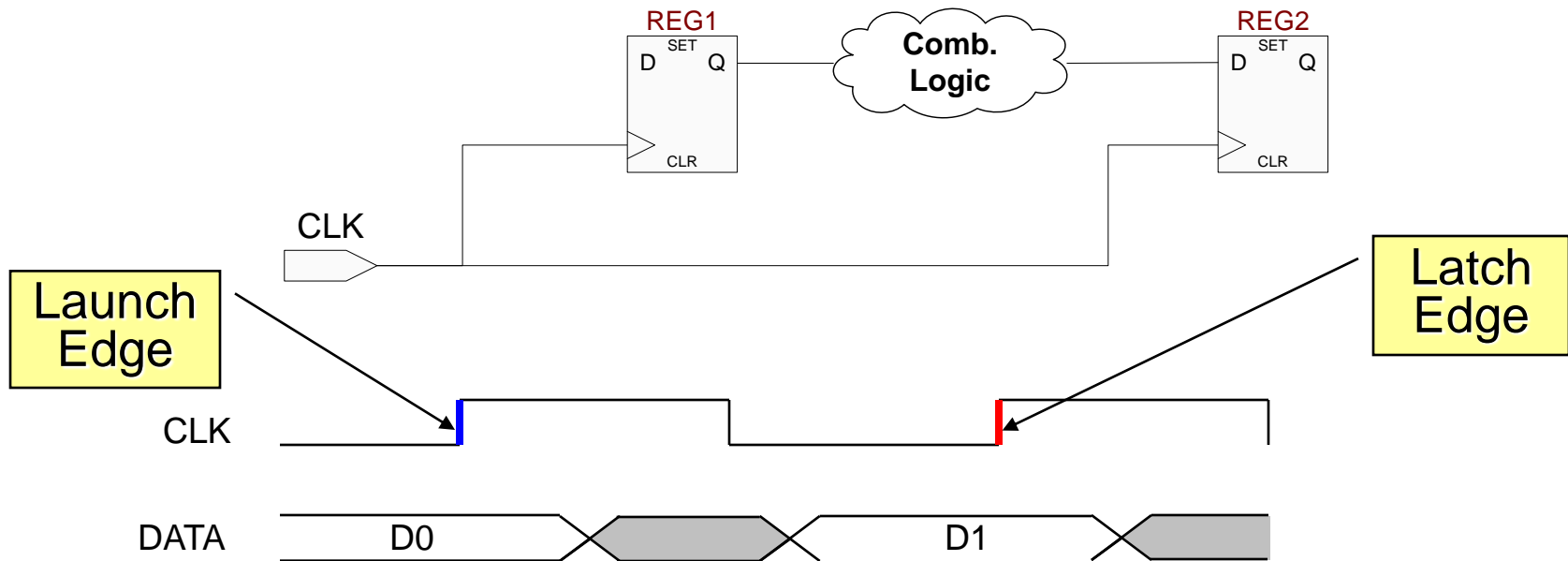
# Timing Analysis with Timequest

# Timing Analysis Basics

# Timing Analysis Basics Objectives

- Define the basic timing parameters used in timing analysis
- Understand the calculations performed by the timing analyzer for reporting timing results

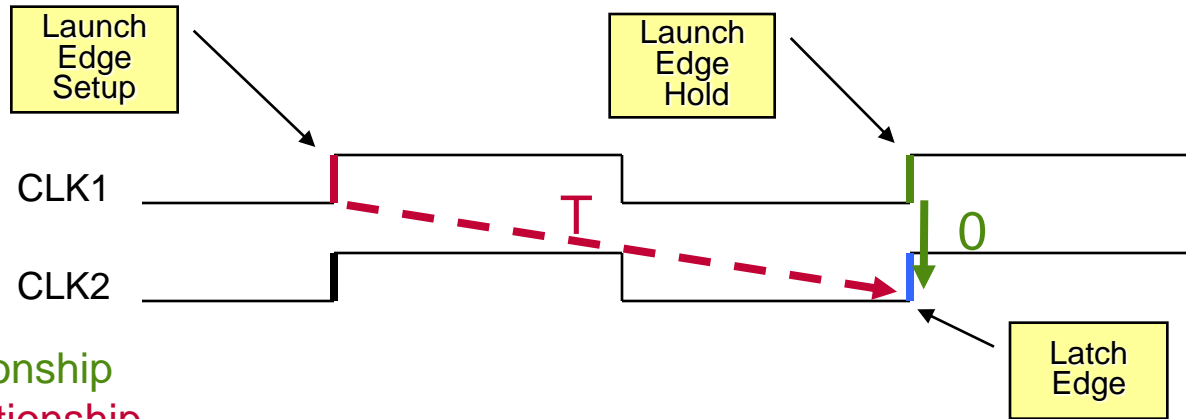
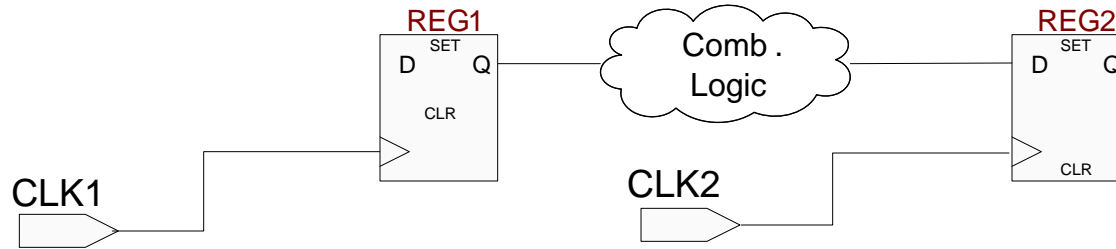
# Launch & Latch Edges



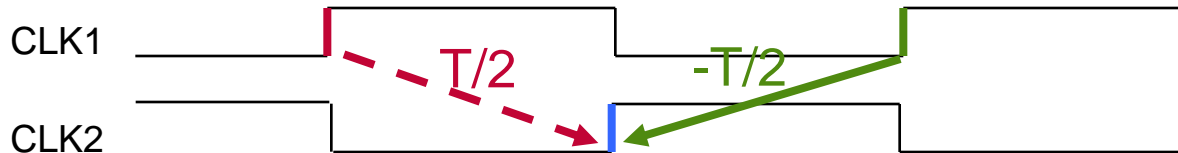
**Launch Edge:** the edge which “launches” the data from source register

**Latch Edge:** the edge which “latches” the data at destination register (with respect to the launch edge)

# Setup & Hold relationships

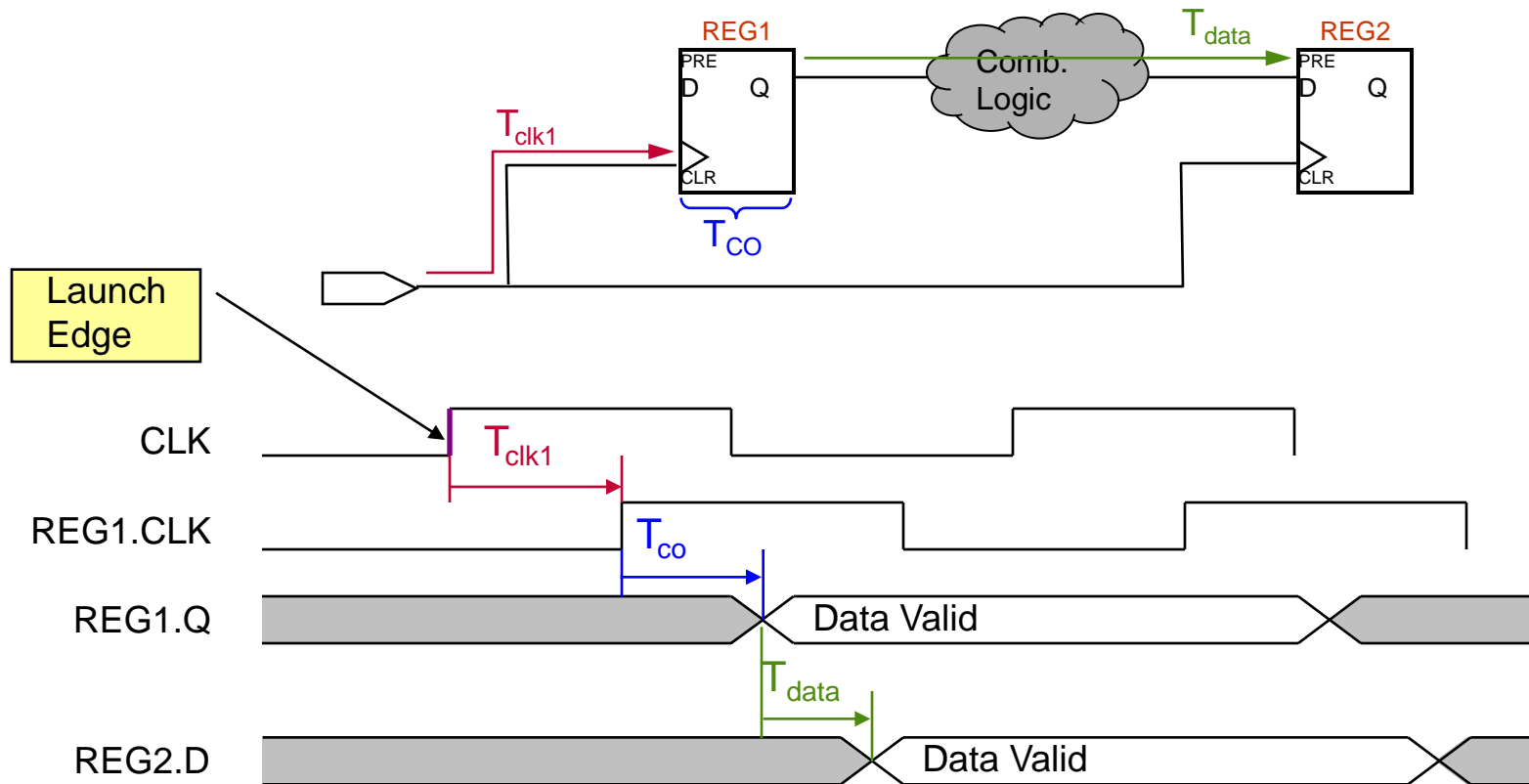


- Hold relationship
- - Setup relationship



# Data Arrival Time

- The time for data to arrive at destination register's D input



$$\text{Data Arrival Time} = \text{launch edge} + T_{clk1} + T_{co} + T_{data}$$

© 2010 Altera Corporation—Confidential

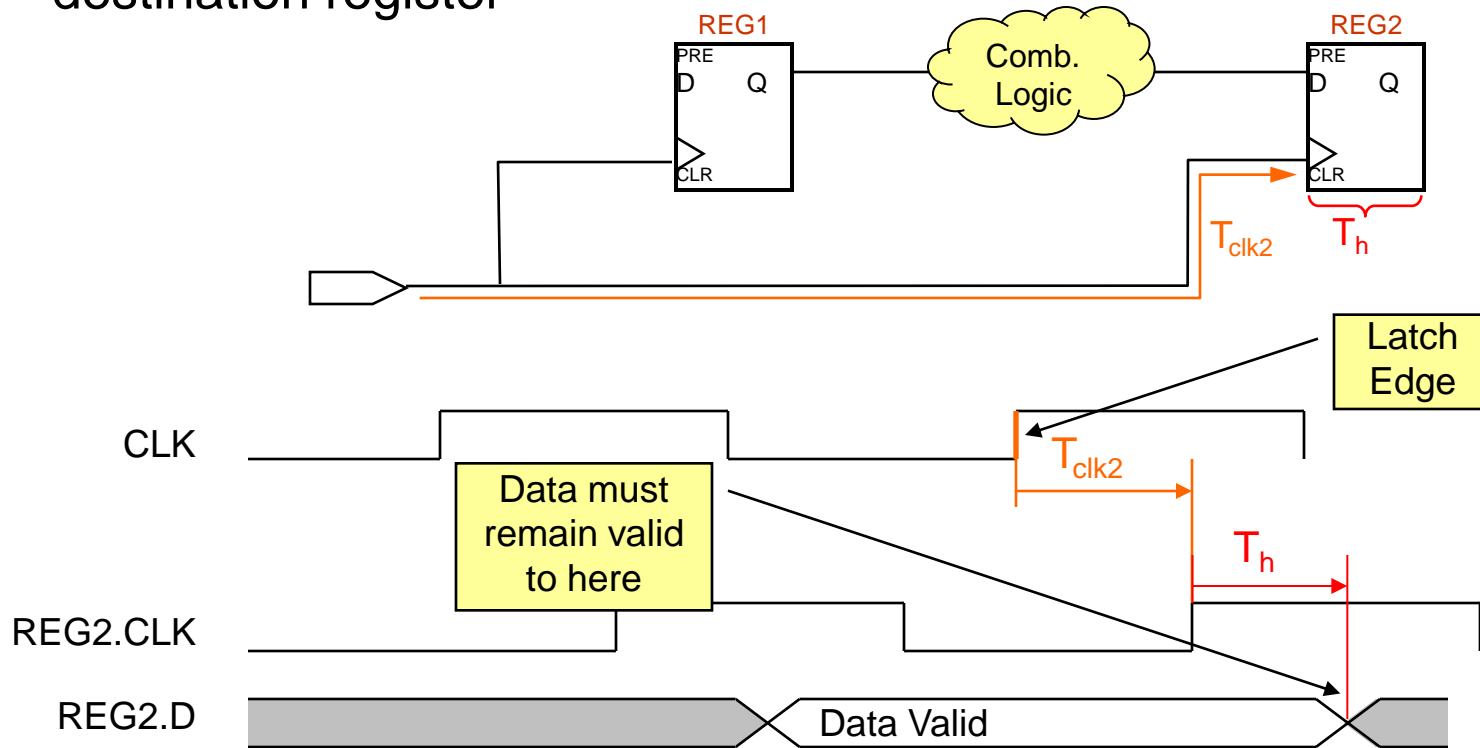
ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.





# Data Required Time - Hold

- The minimum time required for the data to get latched into the destination register

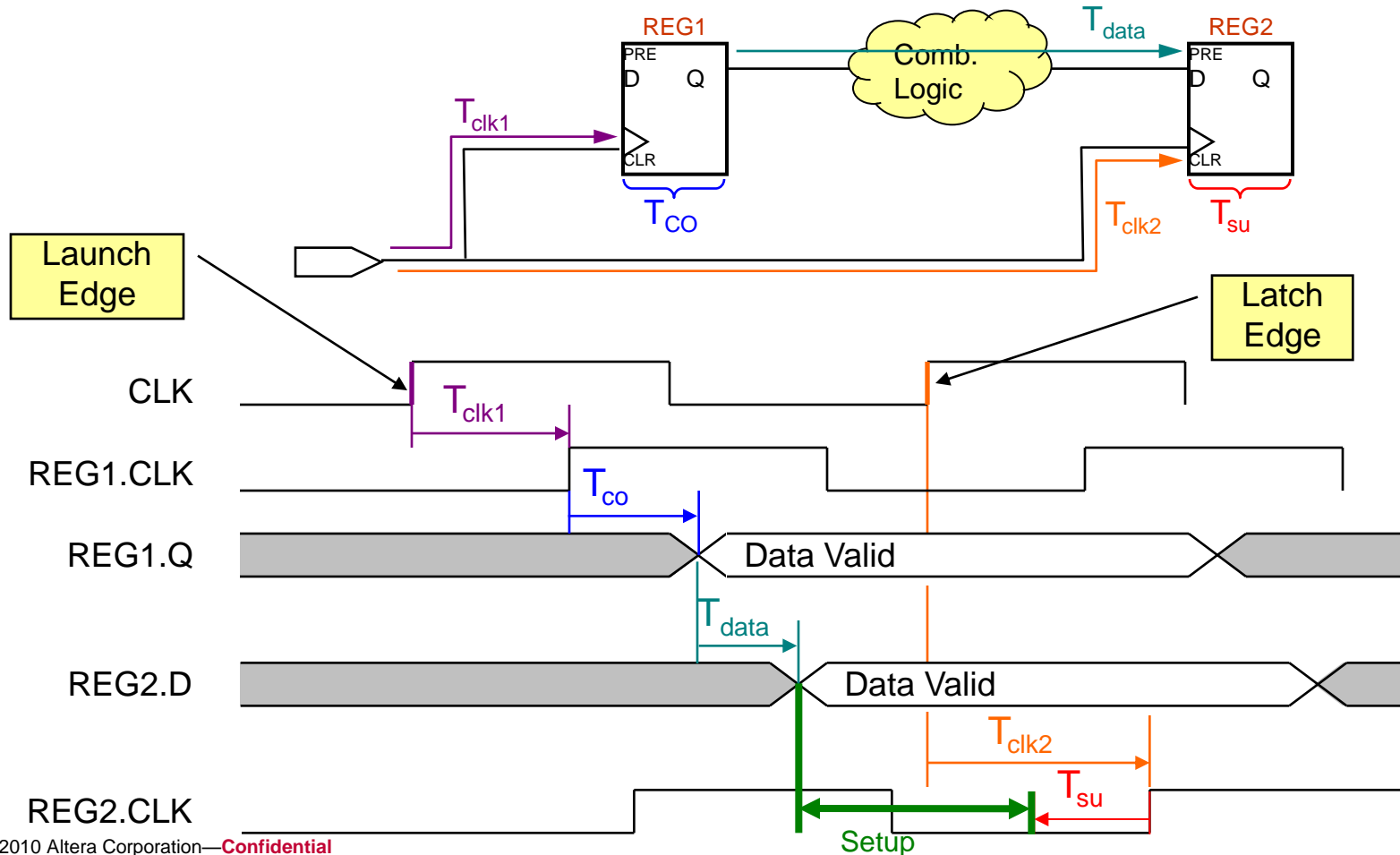


Data Required Time = Clock Arrival Time +  $T_h$  + Hold Uncertainty



# Setup Slack

- The margin by which the setup timing requirement is met. It ensures launched data arrives in time to meet the latching requirement.



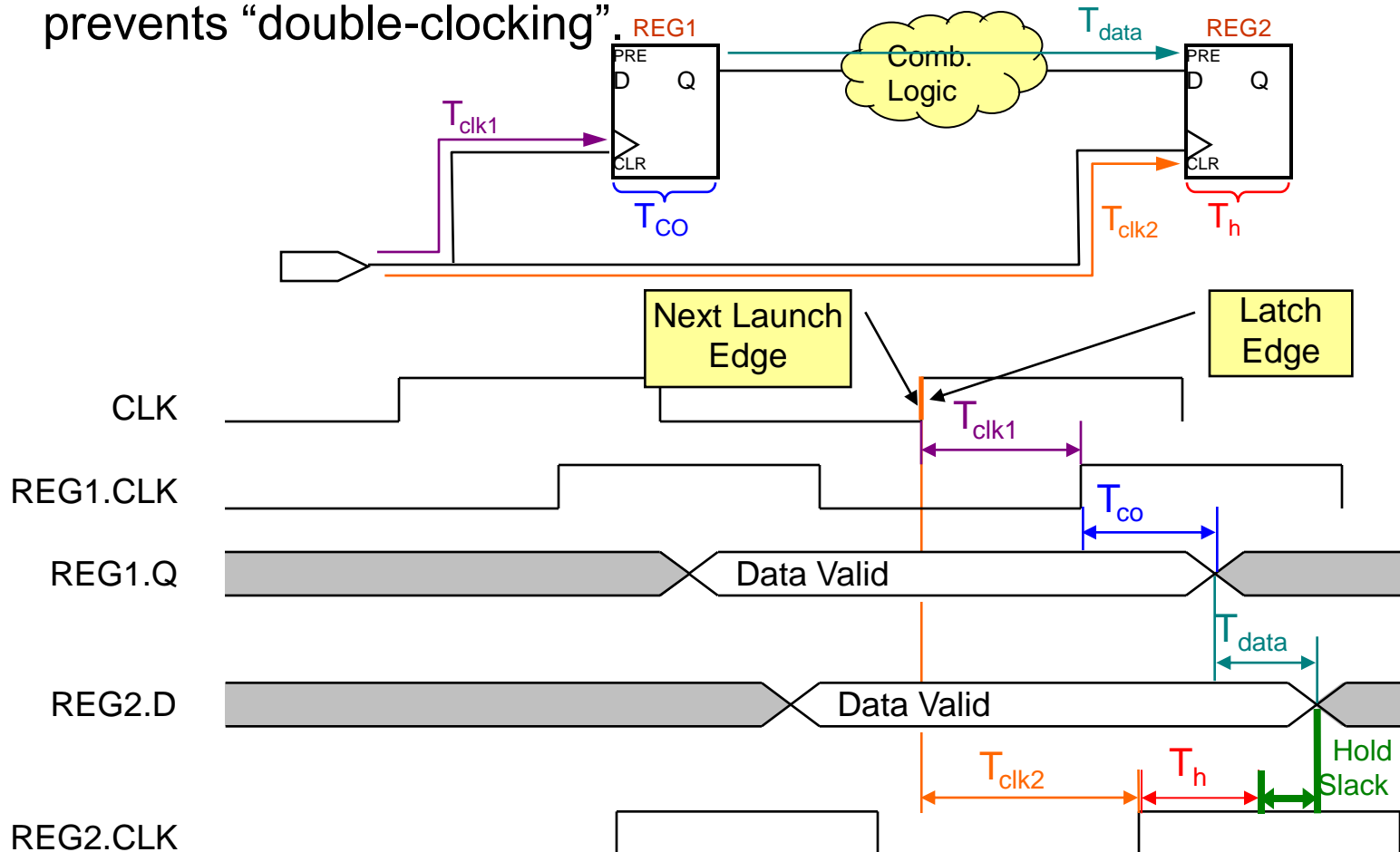
© 2010 Altera Corporation—Confidential

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Registered Pat. & Tm. Off. and Altera marks in and outside the U.S.



# Hold Slack

- The margin by which the hold timing requirement is met. It ensures latch data is not corrupted by data from another launch edge. It also prevents “double-clocking”

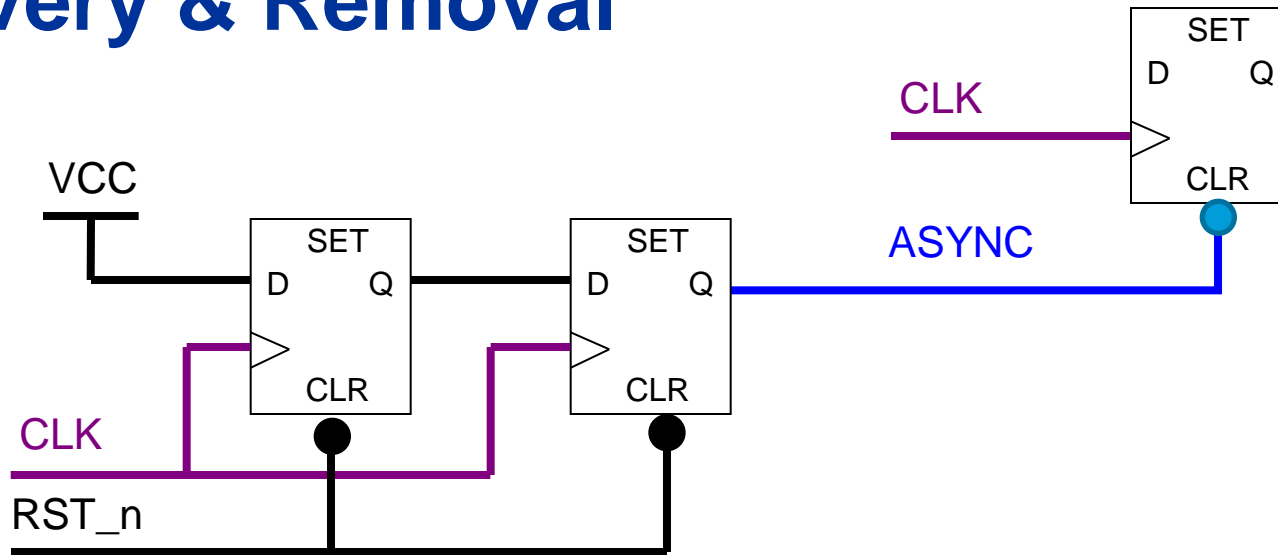


© 2010 Altera Corporation—Confidential

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.

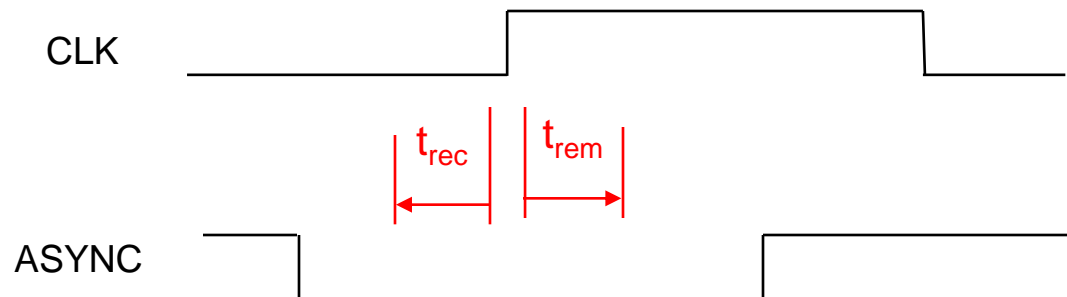


# Recovery & Removal



Recovery: The minimum time an asynchronous signal must be de-asserted BEFORE clock edge

Removal: The minimum time an asynchronous signal must be de-asserted AFTER clock edge



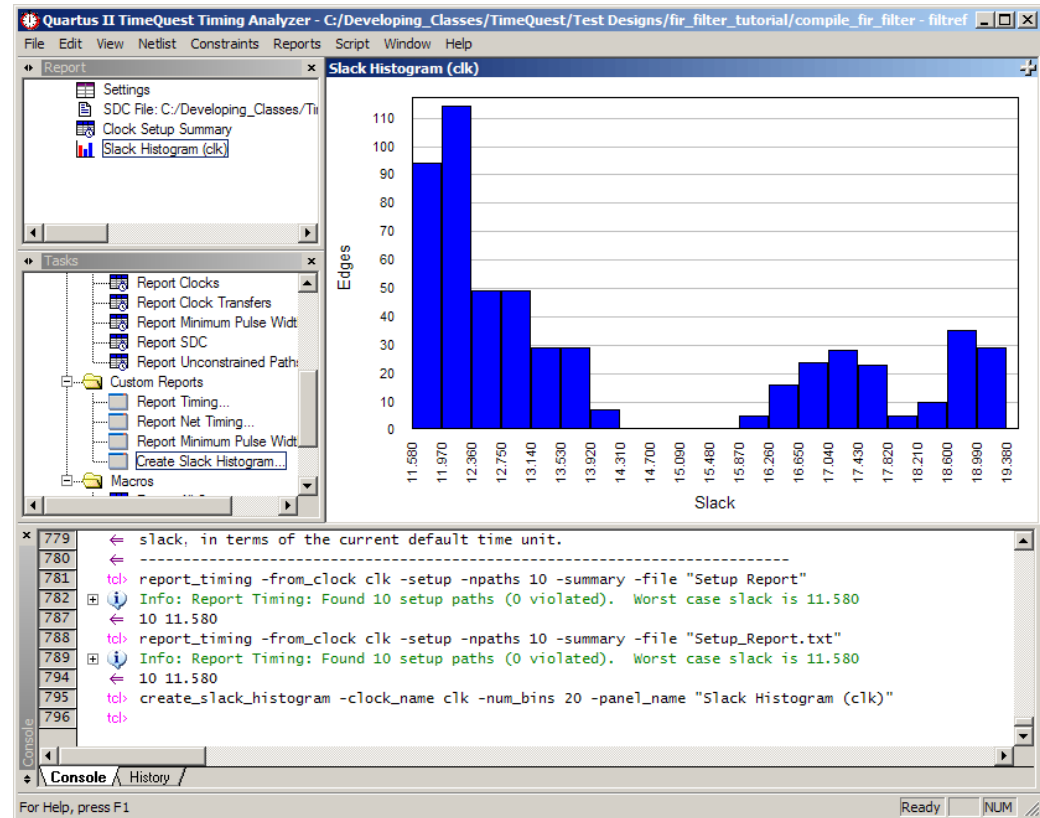
# Timing Models in Detail

- Quartus II software models device timing at multiple process voltage temperature (PVT) conditions by default
  - **Slow corner** model
    - Indicates slowest possible performance for any single path
    - Timing for slowest device at maximum operating temperature and VCCMIN
  - **Fast corner** model
    - Indicates fastest possible performance for any single path
    - Timing for fastest device at minimum operating temperature and VCCMAX
  - **2<sup>nd</sup> slow and 2<sup>nd</sup> fast** models (temperature inversion phenomenon)
    - Slow timing at minimum operating temperature
    - Fast timing at maximum operating temperature
      - Available only for Stratix® V devices
- Why analyze for multiple corner timing models?
  - Ensure **setup** timing is met in the **slow** models
  - Ensure **hold** timing is met in **fast** model
    - Essential for source synchronous interfaces
- Read the following white paper for more information
  - <http://www.altera.com/literature/wp/wp-01139-timing-model.pdf>

# Timing Analysis with Timequest

# TimeQuest Timing Analyzer

- New timing 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 reporting
  - Scripting emphasis
    - Presentation focuses on using GUI



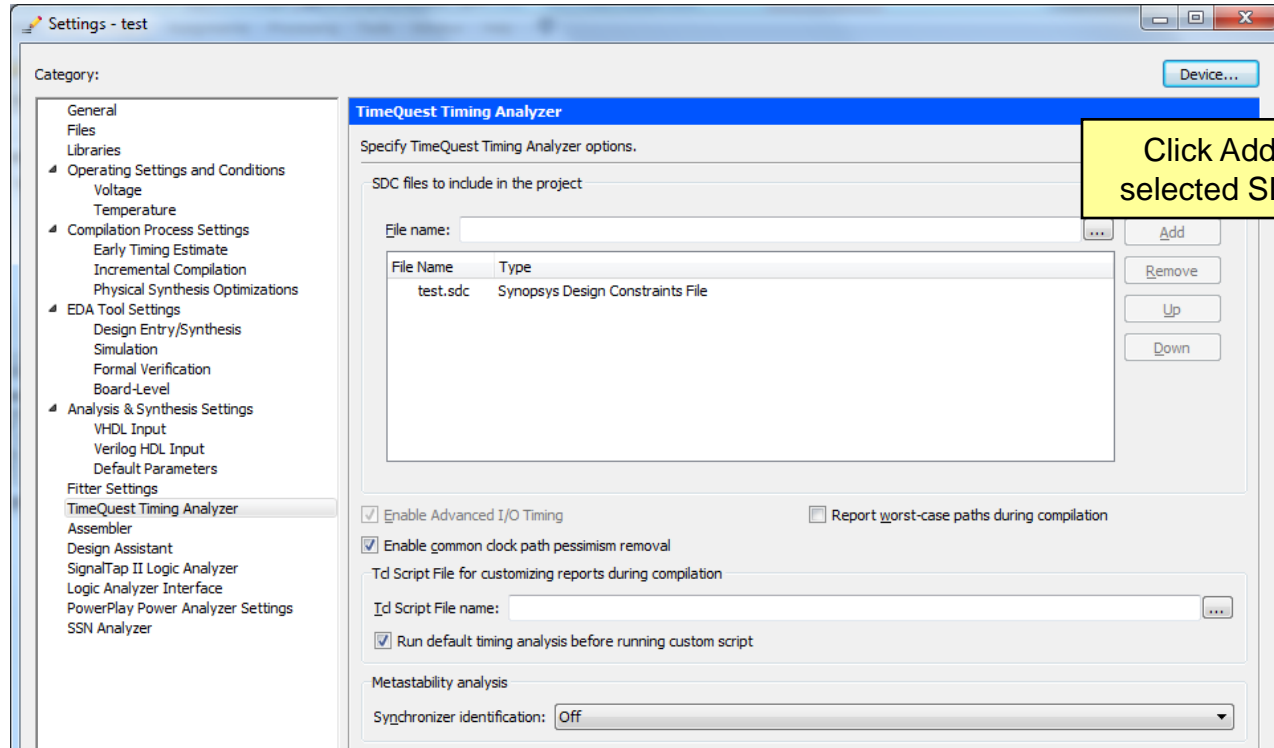
# Timing Analysis Agenda

- TimeQuest flow
- Timing constraints
- Timing reports



# Quartus II TimeQuest Settings

- Add SDC files to TimeQuest Timing Analyzer page of Settings dialog box
- Multicorner analysis checks all process corners in one analysis
- Report worst-case paths in each clock domain
- Select Tcl script to customize report generation




© 2010 Altera Corporation—**Confidential**

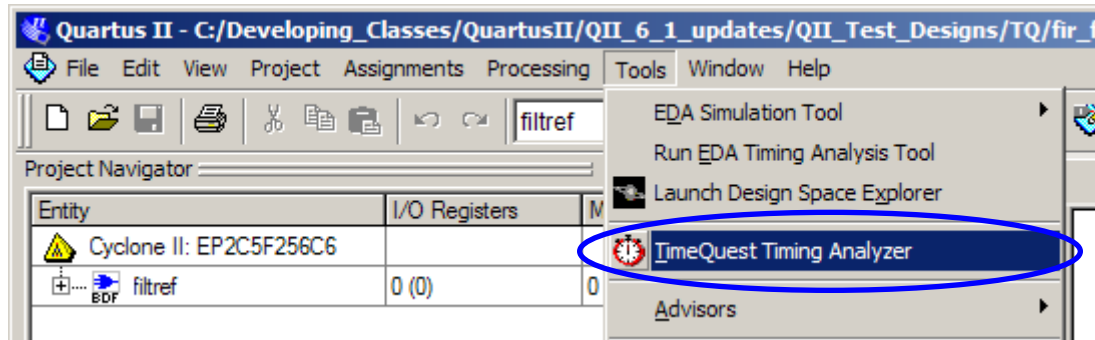
ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.





# Opening TimeQuest

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



# TimeQuest GUI

Menu access to all TimeQuest features

Report Pane

View Pane

Tasks Pane

Console Pane

The screenshot shows the TimeQuest GUI with several panes highlighted by yellow boxes:

- Report Pane:** Located on the left, it contains a tree view of reports including "TimeQuest Timing Analyzer Summary", "SDC File List", "Summary (Setup)", "Summary (Hold)", and "Report Timing".
- View Pane:** The central pane showing a "Report Timing" window. It includes a "Command Info" window with a table of paths, a "Path #1: Setup slack is 1.045" window with a "Data Arrival Path" table, and a "Data Required Path" table. The "Data Arrival Path" table is as follows:
 

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					...ge tir
2	1.297	1.297	R				...k del
3	1.547	0.250		uTco	1	Unassigned	inst2[1
4	1.547	0.000	RR	CELL	1	Unassigned	...]reg
5	1.547	0.000	RR	IC	1	Unassigned	...]dat
6	4.255	2.708	RR	CELL	0	Unassigned	q[15]
- Tasks Pane:** Located on the left, it shows a list of tasks such as "Report RSKM", "Report DDR", "Report SDC", "Report Unconstrained Paths", "Report Ignored Constraints", "Report Datasheet", "Check Timing", "Custom Reports", "Report Timing...", "Report Net Timing...", "Report Path...", "Report Minimum Pulse Width...", and "Create Slack Histogram...".
- Console Pane:** Located at the bottom, it shows a command prompt with the following text:
 

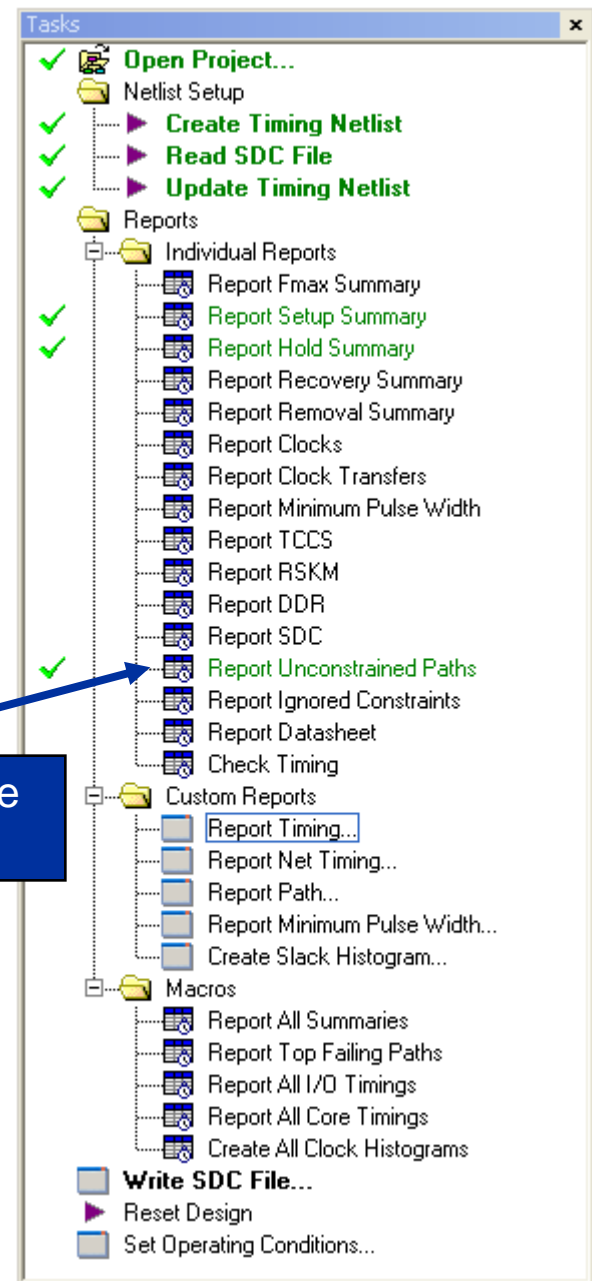
```

22 tcl> read_sdc "C:/altera_trn/Quartus_II_Software_Design_Series_Foundation/QIIF7_2/Solutions/Final_projects/Schematic/pipemult.sdc"
23 Info: Reading SDC File: 'C:/altera_trn/Quartus_II_Software_Design_Series_Foundation/QIIF7_2/Solutions/Final_projects/Schematic/pipemult.sdc'
24 tcl> update_timing_netlist;
25 tcl> create_timing_summary -setup -panel_name "Summary (Setup)"
26 tcl> create_timing_summary -hold -panel_name "Summary (Hold)"
27 tcl> report_timing -to_clock clk1 -setup -npaths 10 -detail path_only -panel_name (Report Timing)
28 Info: Report Timing: Found 10 setup paths (0 violated). Worst case slack is 1.045
33 10 1.045
34 tcl>
      
```

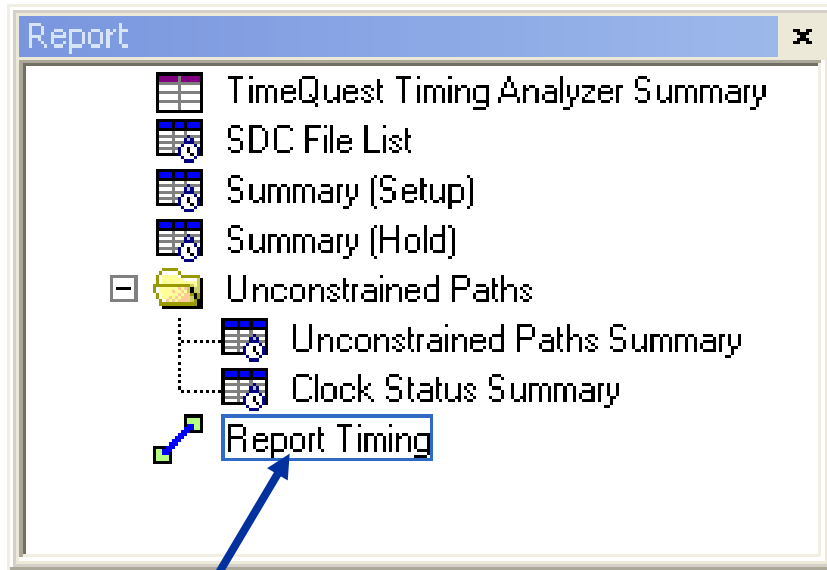
# Tasks Pane

- 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

# View Pane

- Main viewing area that displays report table contents & graphical results

### Report Timing Summary

	Slack	From Node	To Node	Launch Clock	Latch Clock
1	1.045	inst2[15]	q[15]	clk1	clk1
2	1.045	inst2[13]	q[13]	clk1	clk1
3	1.045	inst2[9]	q[9]	clk1	clk1
4	1.045	inst2[1]	q[1]	clk1	clk1
5	1.045	inst2[2]	q[2]	clk1	clk1
6	1.045	inst2[10]	q[10]	clk1	clk1
7	1.045	inst2[3]	q[3]	clk1	clk1
8	1.045	inst2[4]	q[4]	clk1	clk1
9	1.045	inst2[14]	q[14]	clk1	clk1
10	1.045	inst2[11]	q[11]	clk1	clk1
11	1.045	inst2[5]	q[5]	clk1	clk1
12	1.045	inst2[6]	q[6]	clk1	clk1
13	1.045	inst2[12]	q[12]	clk1	clk1
14	1.045	inst2[7]	q[7]	clk1	clk1
15	1.045	inst2[8]	q[8]	clk1	clk1
16	1.045	inst2[0]	q[0]	clk1	clk1
17	2.471	dataa[6]	...m_mult:lpm_m	clk1	clk1
18	2.471	dataa[7]	...m_mult:lpm_m	clk1	clk1
19	2.471	dataa[4]	...m_mult:lpm_m	clk1	clk1

### Slack Histogram (clk1)

### Report Timing

	Slack	From Node	To Node	Launch Clock	Latch Clock
1	1.045	inst2[15]	q[15]	clk1	clk1
2	1.045	inst2[13]	q[13]	clk1	clk1
3	1.045	inst2[9]	q[9]	clk1	clk1
4	1.045	inst2[1]	q[1]	clk1	clk1
5	1.045	inst2[2]	q[2]	clk1	clk1

Path Slack Report

Timing Summary Table

Timing Histogram

### Path #5: Setup slack is 1.045

Total	Incr	RF	Type	Fanout	Location	Element
1	0.000					launch edge time
2	1.297	1.297	R			clock network delay
3	1.547	0.250	uTco	1	Unassigned	inst2[2]
4	1.547	0.000	RR	CELL	1	inst2[2]regout
5	1.547	0.000	RR	IC	1	Unassigned
6	4.255	2.708	RR	CELL	0	Unassigned

### Data Required Path

Total	Incr	RF	Type	Fanout	Location	Element
1	6.000	6.000				latch edge time
2	6.000	0.000	R			clock network delay
3	5.300	-0.700	R	oExt	0	Unassigned

### Path #5: Setup slack is 1.045

# Viewing Multiple Reports

Setup: clk\_x2

Command Info | Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Setup Time	Hold Time	Delay
1	0.295	TDM_...OUT0	TDM_mu...c_out2	clk_x2	clk_x2			
2	0.295	TDM_...OUT0	TDM_m...AOUT1	clk_x2	clk_x2			
3	0.295	TDM_...OUT0	TDM_m...AOUT2	clk_x2	clk_x2			
4	0.295	TDM_...OUT0	TDM_m...AOUT3	clk_x2	clk_x2			
5	0.295	TDM_...OUT0	TDM_m...AOUT4	clk_x2	clk_x2	3.500	-0.388	2.620
6	0.295	TDM_...OUT0	TDM_m...AOUT5	clk_x2	clk_x2	3.500	-0.388	2.620
7	0.295	TDM_...OUT0	TDM_m...AOUT6	clk_x2	clk_x2	3.500	-0.388	2.620
8	0.295	TDM_...OUT0	TDM_m...AOUT7	clk_x2	clk_x2	3.500	-0.388	2.620
9	0.295	TDM_...OUT0	TDM_m...AOUT8	clk_x2	clk_x2	3.500	-0.388	2.620

Click & drag '+' sign to divide view pane into multiple windows

**Path #1: Setup slack is 0.295**

Path Summary | Statistics | Data Path | Waveform

**Data Arrival Path**

	Total	Incr	RF	Type	Fanout	Location
1	0.000	0.000				
2	0.404	0.404				
3	0.000	0.000				
4	0.000	0.000			1	PIN F?

**Data Required Path**

	Total	Incr	RF	Type	Fanout	Location
1	3.500	3.500				
2	3.516	0.016				
3	3.500	0.000				
4	3.500	0.000			1	PIN F?

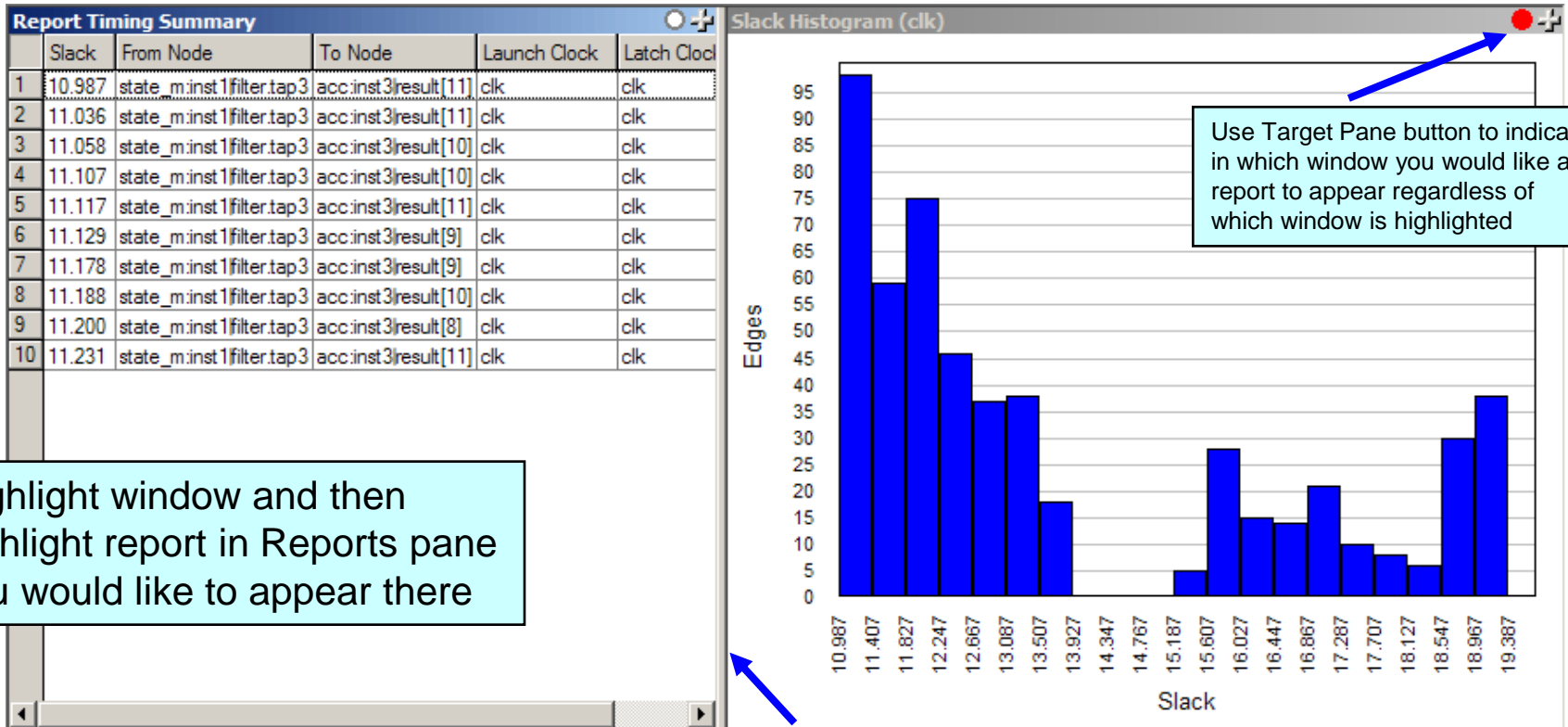
**Path #1: Setup slack is 0.295**

Path Summary | Statistics | Data Path | Waveform

The timing diagram illustrates the relationship between the Launch Clock and Latch Clock. The Launch Clock has a period of 3.5 ns. The Setup Relationship is shown as a horizontal arrow indicating a 3.5 ns interval. The Latch Clock is shown as a square wave. The Data Arrival is shown as a signal that arrives at the Latch Clock. The Clock Delay is shown as a horizontal arrow indicating a 0.404 ns interval.

# Viewing Multiple Reports Example

View pane split into two side-by-side windows

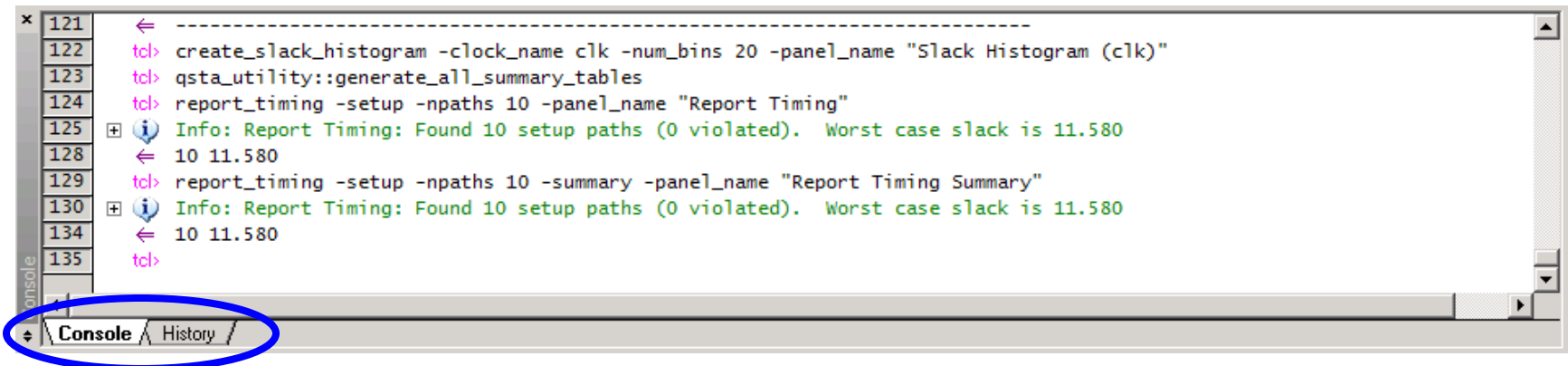


Highlight window and then highlight report in Reports pane you would like to appear there

Drag bar to left or right to remove split

# 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



The screenshot shows a console window with a list of lines on the left and a text area on the right. The text area contains the following text:

```
-----  
tcl> create_slack_histogram -clock_name clk -num_bins 20 -panel_name "Slack Histogram (clk)"  
tcl> qsta_utility::generate_all_summary_tables  
tcl> report_timing -setup -npaths 10 -panel_name "Report Timing"  
Info: Report Timing: Found 10 setup paths (0 violated). Worst case slack is 11.580  
10 11.580  
tcl> report_timing -setup -npaths 10 -summary -panel_name "Report Timing Summary"  
Info: Report Timing: Found 10 setup paths (0 violated). Worst case slack is 11.580  
10 11.580  
tcl>
```

At the bottom of the console window, there are two tabs: "Console" and "History". The "Console" tab is selected and circled in blue.

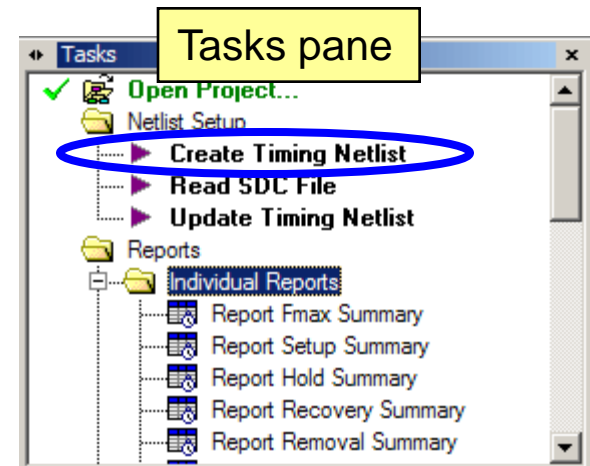
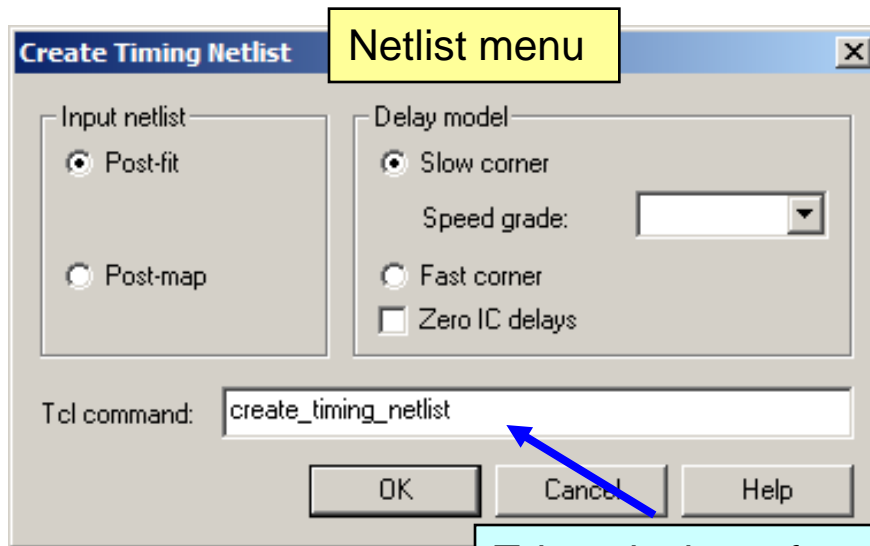


# Steps to Using TimeQuest

1. Generate timing netlist
2. Read SDC file
3. Update timing netlist
4. Generate timing reports

# 1) Generate Timing Netlist

- Creates timing netlist (i.e. database) based on compilation results
  - Post-synthesis (mapping) or post-fit
  - Worst-case (slow) or best-case (fast) timing model
- To execute:

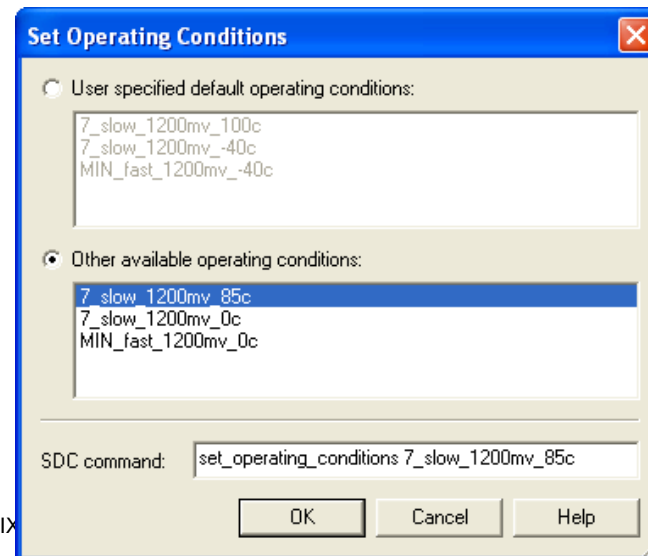
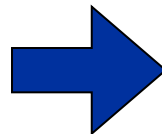
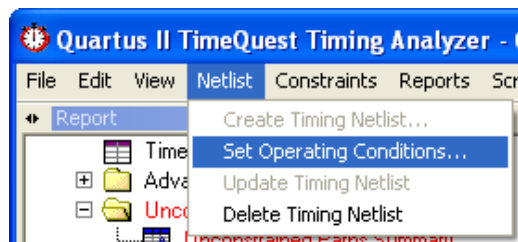


Tcl equivalent of command

Tcl: create\_timing\_netlist

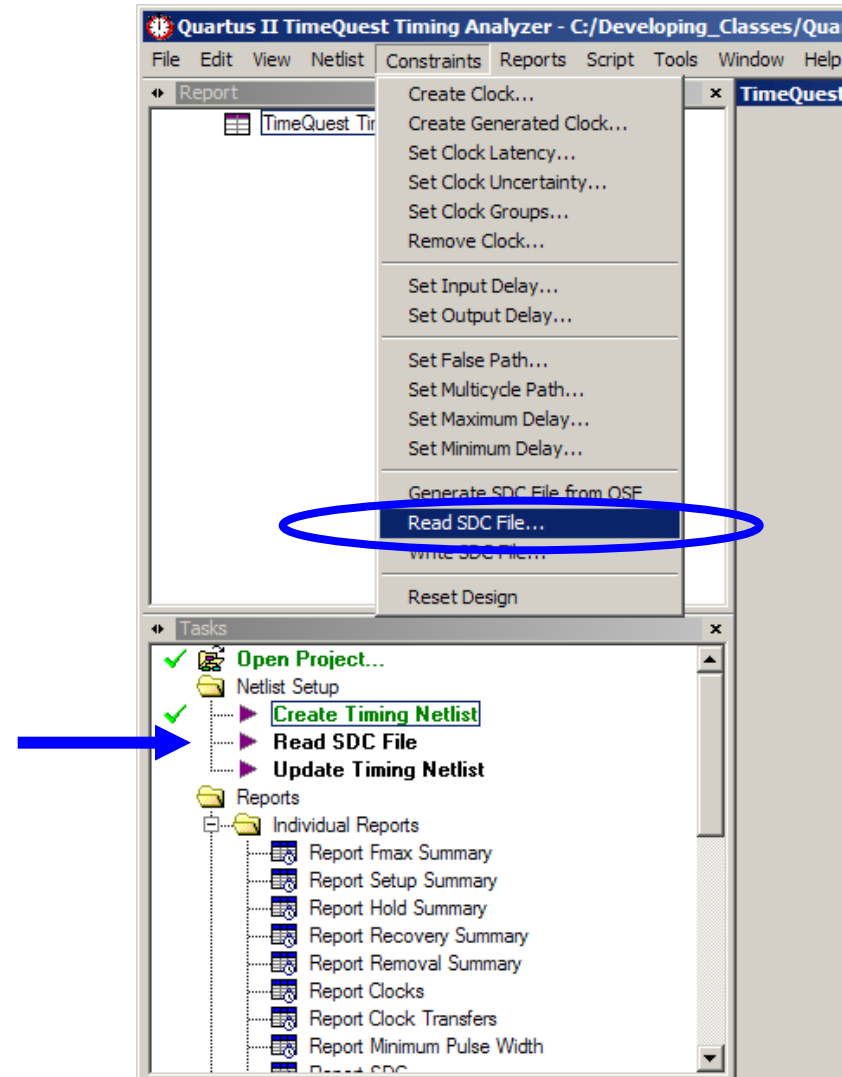
# Specifying Operating Conditions

- 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` to see available conditions for target device



## 2) Read SDC File

- Reads constraints & exceptions from SDC file
  - Skip if no SDC file
- Execution
  - Read SDC File (Tasks pane or Constraints menu)
- File Precedence (if no filename specified)
  - Files specifically added to Quartus II project
  - <current\_revision>.sdc (if it exists)

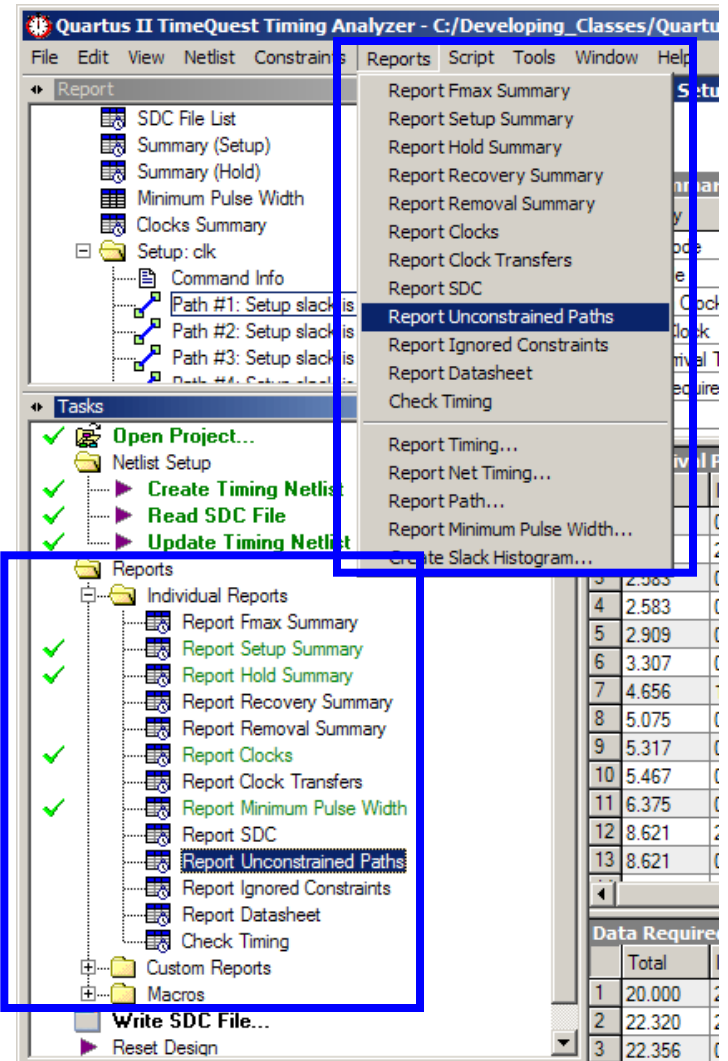


### 3) Update Timing Netlist

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

# 4) Generate Timing Reports

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



Double-click on individual report

# Reset Design Command

- Located in Tasks pane
- Flushes all timing constraints from current timing 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

# Steps to Using TimeQuest (Review)

1. Generate timing netlist
2. Read SDC file
3. Update timing netlist
4. Generate timing reports

(Optionally)

5. Modify SDC file
6. Reset Design and go back to step 2



# Timing Analysis Agenda

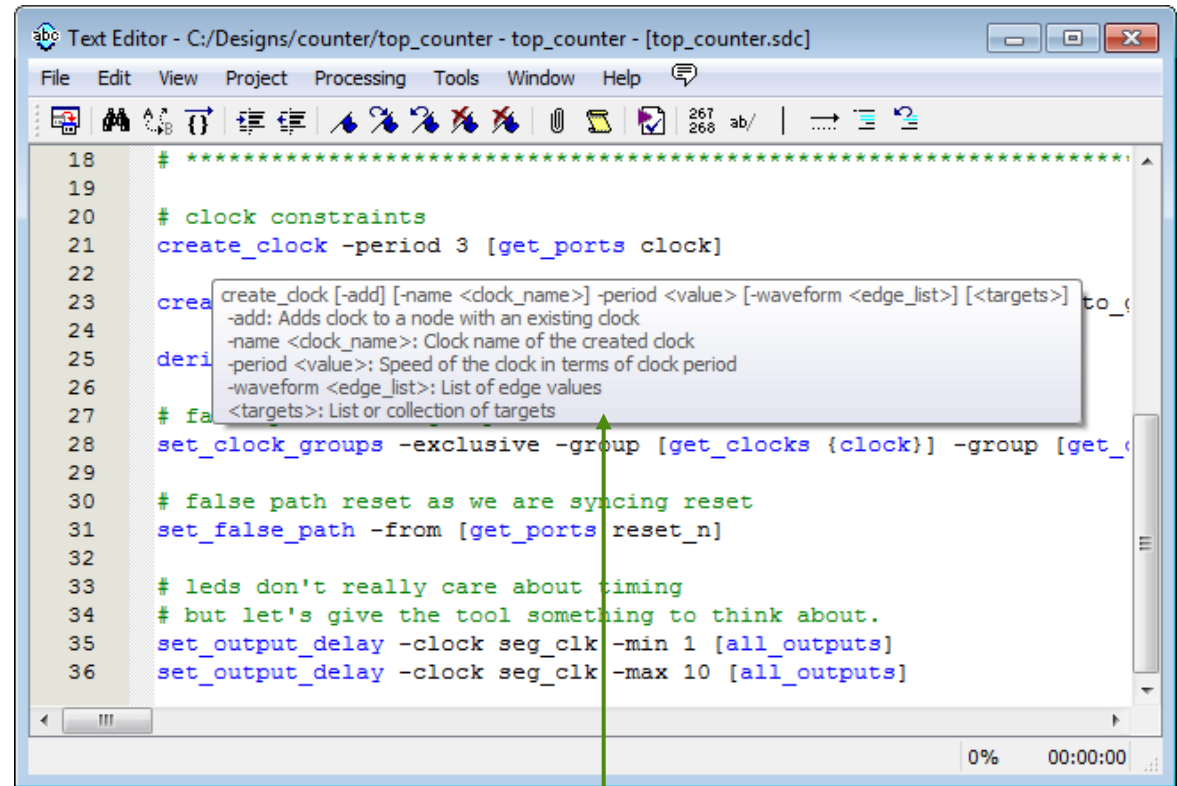
- TimeQuest flow
- Timing constraints
- Timing reports



# SDC File Editor = Quartus II Text Editor

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

- 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



The screenshot shows a text editor window titled "Text Editor - C:/Designs/counter/top\_counter - top\_counter - [top\_counter.sdc]". The editor displays SDC code with syntax coloring. A tooltip is visible over the "create\_dock" command on line 23. The tooltip text is:

```
create_dock [-add] [-name <clock_name>] -period <value> [-waveform <edge_list>] [<targets>] to_  
-add: Adds dock to a node with an existing dock  
-name <clock_name>: Clock name of the created dock  
-period <value>: Speed of the dock in terms of dock period  
-waveform <edge_list>: List of edge values  
<targets>: List or collection of targets
```

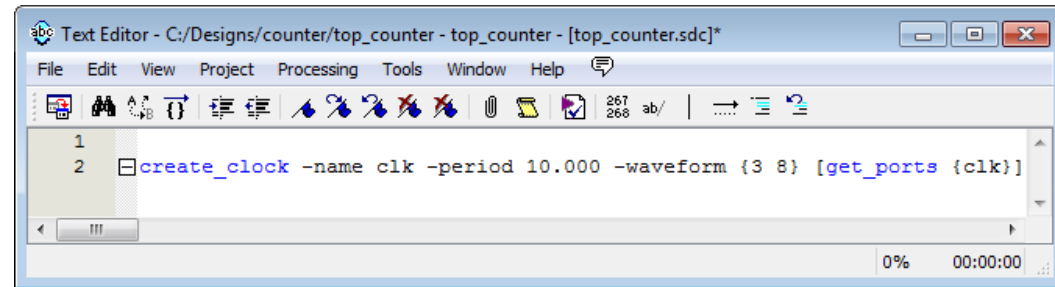
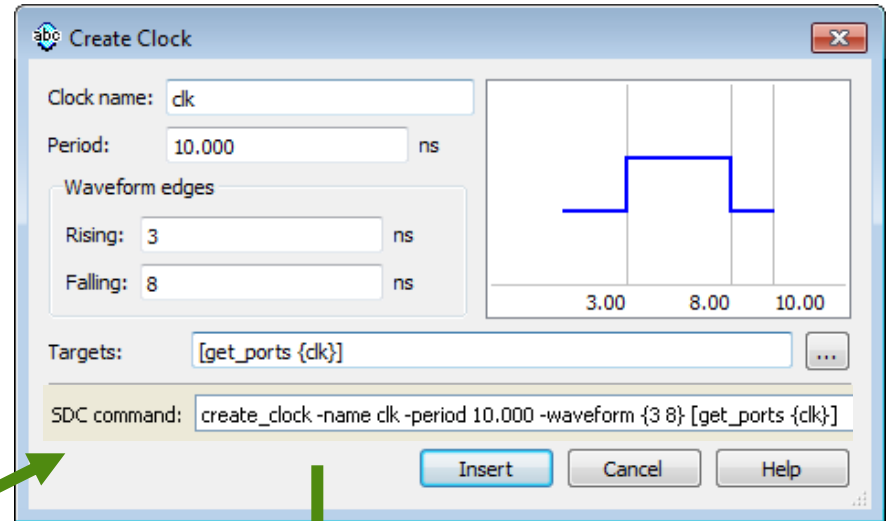
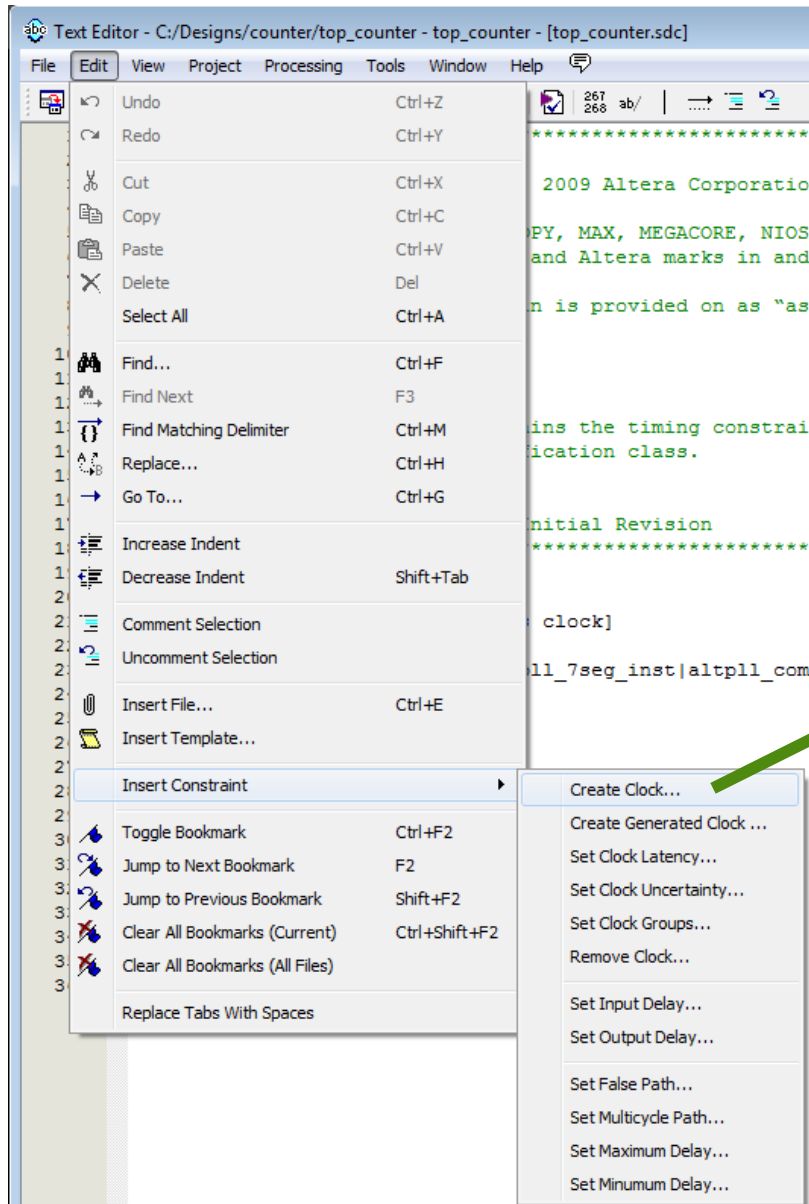
The code in the editor includes:

```
18 # *****  
19  
20 # clock constraints  
21 create_clock -period 3 [get_ports clock]  
22  
23 create_dock [-add] [-name <clock_name>] -period <value> [-waveform <edge_list>] [<targets>] to_  
24 -add: Adds dock to a node with an existing dock  
25 -name <clock_name>: Clock name of the created dock  
26 -period <value>: Speed of the dock in terms of dock period  
27 -waveform <edge_list>: List of edge values  
28 -targets>: List or collection of targets  
28 set_clock_groups -exclusive -group [get_clocks {clock}] -group [get_  
29  
30 # false path reset as we are syncing reset  
31 set_false_path -from [get_ports reset_n]  
32  
33 # leds don't really care about timing  
34 # but let's give the tool something to think about.  
35 set_output_delay -clock seg_clk -min 1 [all_outputs]  
36 set_output_delay -clock seg_clk -max 10 [all_outputs]
```

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

TUS & STRATIX are Reg. U.S. Pat. & Trademark Off.

location



# SDC Templates



- Quickly add customized constraint templates

Toolbar button or Edit menu

Language templates:

- TimeQuest
  - SDC Commands
    - Collections
    - Clocks
    - Clock Attributes
    - I/O Delays
    - Exceptions
    - Scripting
    - SDC Cookbook**
      - Create Base Clocks and PLL Output Cl
      - Create Base Clocks Manually and PLL
      - Create Base Clocks Manually and PLL
      - 2:1 Clock Muxing
      - Externally Switched Clock Constraints
      - PLL Clock Switchover
      - System Synchronous Input Constraint
      - System Synchronous Output Constrai
      - tSU, th, and tCO Constraints
      - Multicycle Clock-to-Clock
      - Multicycle Clock-to-Register
      - False Path Clock-to-Clock
      - False Path Clock-to-Register
      - JTAG Signal Constraints

Preview:

```
# Create a base clock for the PLL input clock
create_clock -name __name -period __period [get_ports __port]
# Create a generated clock for each PLL output clock
create_generated_clock -name __name -divide_by __factor -mult:
# Create additional generated clocks for each PLL output clock
create_generated_clock -name __name -divide_by __factor -mult:
```

SDC "Cookbook" templates from TimeQuest TA Online Resource Center

Preview window: edit before inserting & save as user template

Save User Template

You made changes to an Altera-provided template, however you cannot save changes to the template. If you want to save the changes, you can save the template as a user template.

User template name: my clock template

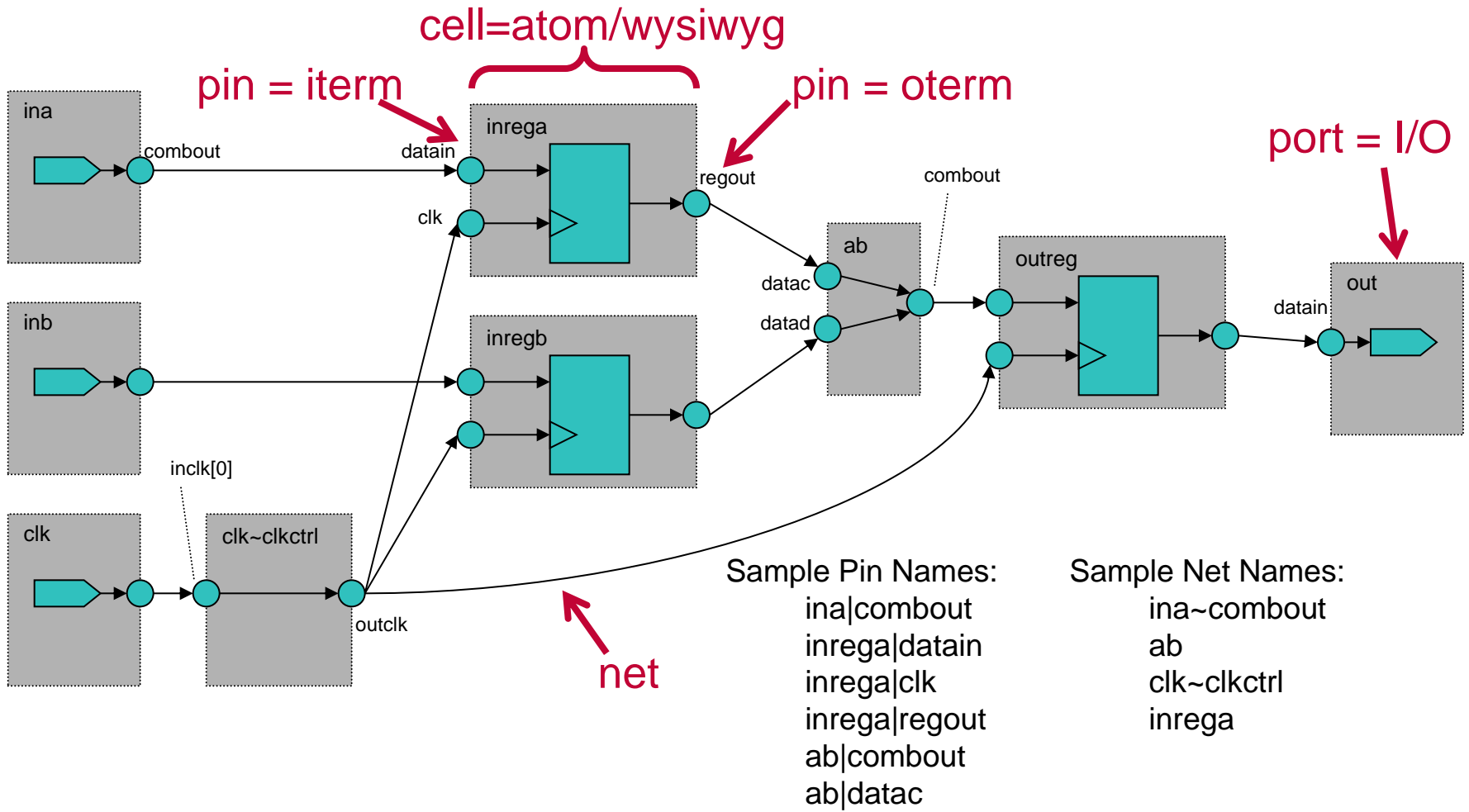
OK Cancel

Save Insert Close

# SDC Netlist Terminology

Term	Definition
Cell	Device building blocks (e.g. look-up tables, registers, embedded multipliers, memory blocks, I/O elements, PLLs, etc.)
Pin	Input or outputs of cells
Net	Connections between pins
Port	Top-level inputs and outputs (e.g. device pins)

# SDC Netlist Example



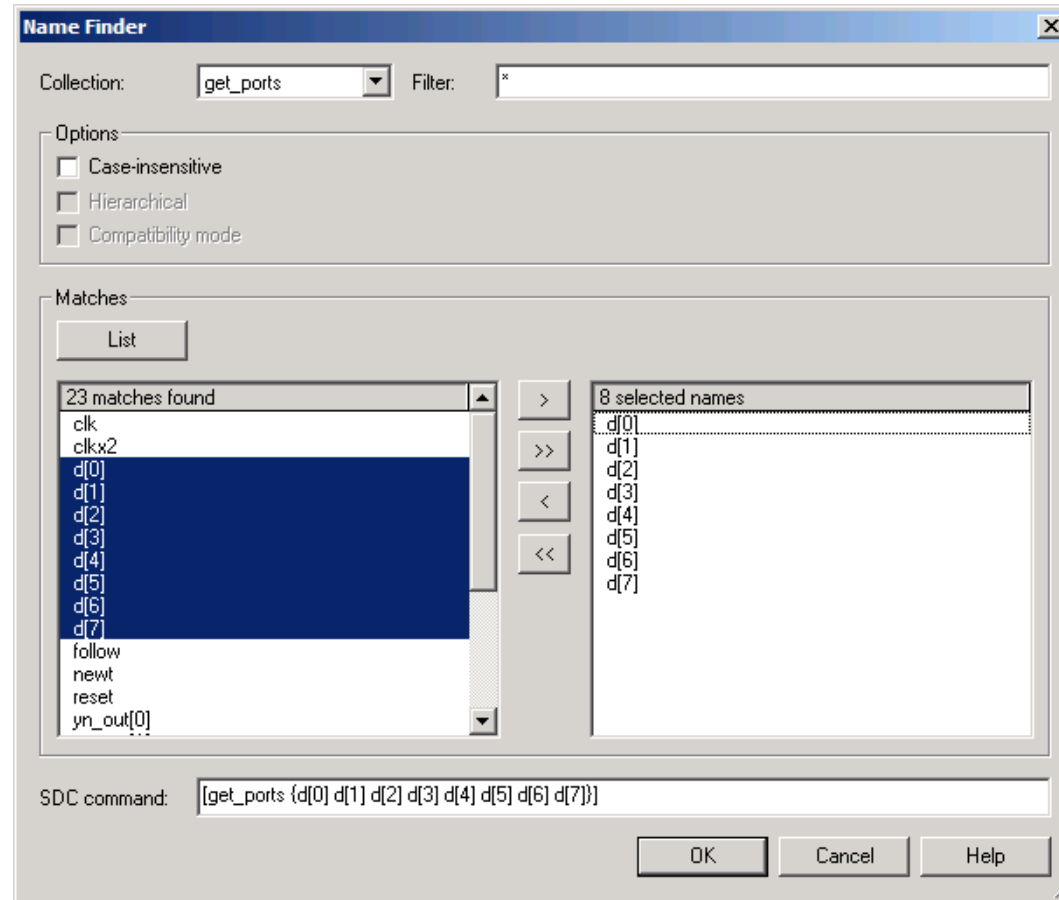
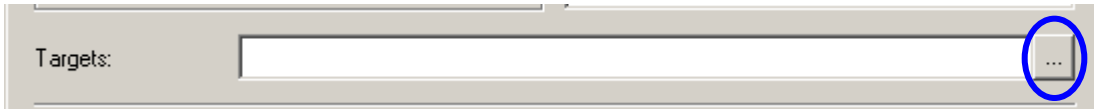
# Collections

- Searches and returns from the design netlist with a list of names meeting criteria
- Used in SDC commands
  - Some collections searched automatically during commands usage and may not need to be specified
- Examples
  - `get_ports`
  - `get_pins`
  - `get_clocks`
  - `all_clocks`
  - `all_registers`
  - `all_inputs`
  - `all_outputs`

*See “TimeQuest Timing Analyzer” chapter of the Quartus II Software Handbook for a complete list & description of each*

# Name Finder

Clicking on Browse button opens Name Finder allowing you to search netlist for node names






# Name Finder Search Options

- All options off
  - Hierarchy levels in **Filter** match results except for \*
  - \* finds all names in all levels of hierarchy in selected collection
  - \* | **data**\* finds names starting with **data** at second level *only*
- Case-insensitive (all collections)
  - Names match **Filter** ignoring capitalization
- Hierarchical (get\_pins; get\_cells only)
  - **Filter** must be just cell name or in form of <cell> | <pin>
  - **foo** | \* finds all pins on cell named **foo**
  - \* | **data**\* finds all pins starting with **data** at *any* level of hierarchy
- Compatibility mode (get\_pins; get\_cells only)
  - Always searches entire hierarchy
  - \* | **data**\* finds all pins starting with **data** at *any* level of hierarchy
  - \* | \* | **data**\* performs the same search; extra \* | not required

# SDC Timing Constraints

- Clocks 
- I/O
- False paths
- Multicycle paths
- Absolute delays

# Clocks in SDC

## ■ Two types

### – Clock

- Absolute or base clock

### – Generated Clock

- Timing derived from another clock in design
  - Must have defined relation with source clock
- Apply to output of logic function that modifies clock input
  - PLLs, clock dividers, output clocks, ripple clocks, etc.
  - Clock inversions automatically detected unless derived from more complex logic structure

## ■ All are related by default

- Cross-domain transfers analyzed

# Clock Constraints

- Create Clock
- Create Generated Clock
- Clock Uncertainty
- Clock Latency
- PLL clocks

# Creating a Clock

- Command: `create_clock`

- Options

  - `[-name <clock_name>]`

  - `-period <time>`

  - `[-waveform {<rise_time> <fall_time>}]`

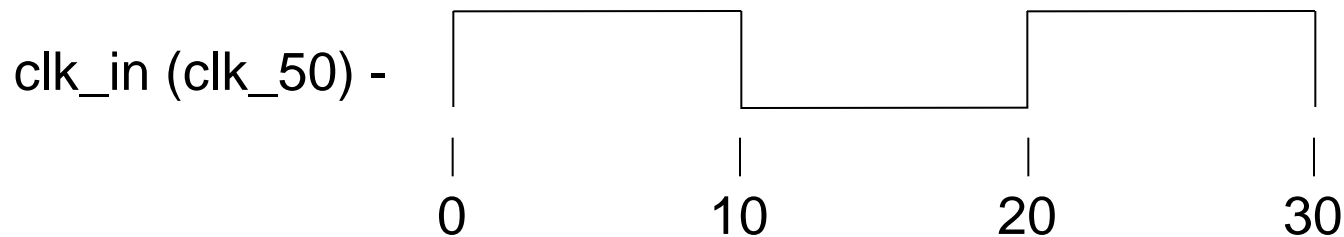
  - `[<targets>]`

  - `[-add]`

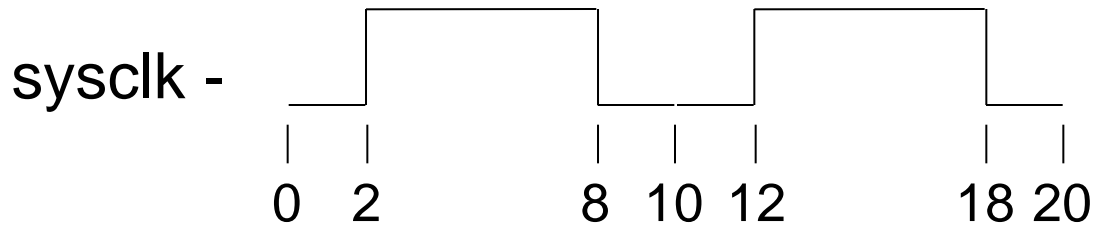
`[]` = optional

# create\_clock Examples

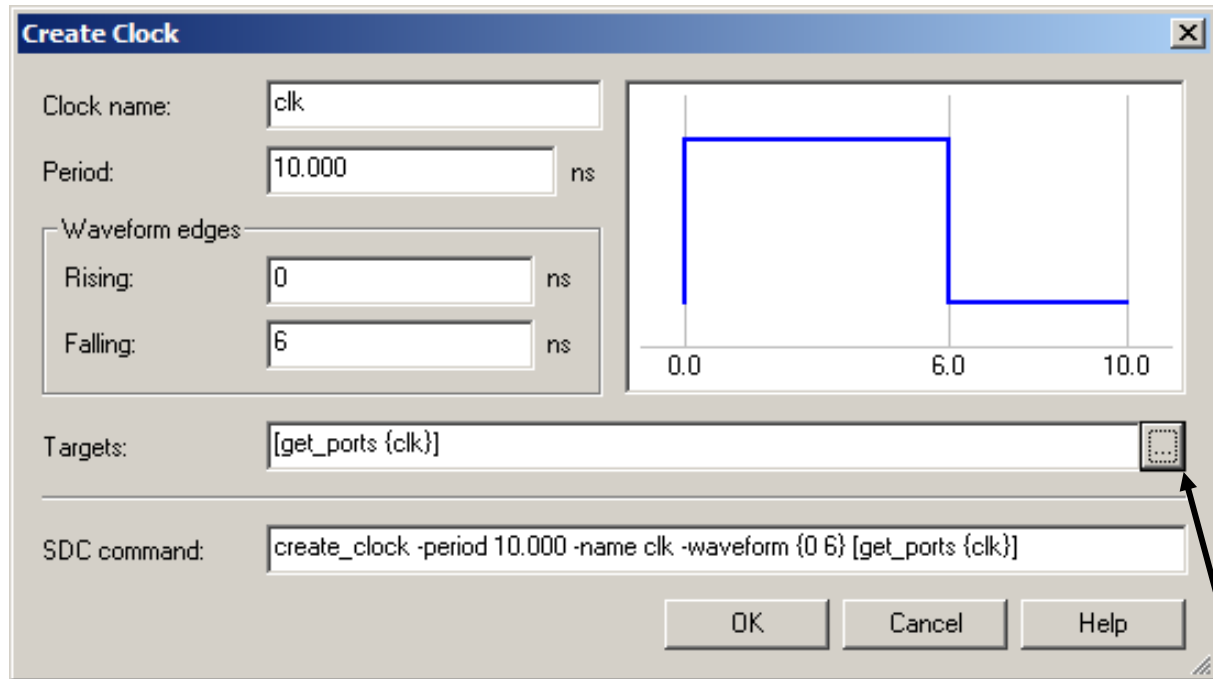
```
create_clock -period 20.0 -name clk_50 [get_ports clk_in]
```



```
create_clock -period 10.0 -waveform {2.0 8.0} [get_ports sysclk]
```



# Create Clock using GUI



Use Name Finder to search collections

# Creating a Generated Clock

- **Command:** `create_generated_clock`

- **Options**

- `[-name <clock_name>]`
- `-source <master_pin>`
- `[-master_clock <clock_name>]`
- `[-divide_by <factor>]`
- `[-multiply_by <factor>]`
- `[-duty_cycle <percent>]`
- `[-invert]`
- `[-phase <degrees>]`
- `[-edges <edge_list>]`
- `[-edge_shift <shift_list>]`
- `[<targets>]`
- `[-add]`



# Create Generated Clock using GUI

**Create Generated Clock**

Clock name:

Source:

Relationship to source

Based on period

Divide by:

Multiply by:

Duty cycle:

Phase:

Offset:

Based on waveform

Edge list:

Edge shift list:  ns  ns  ns

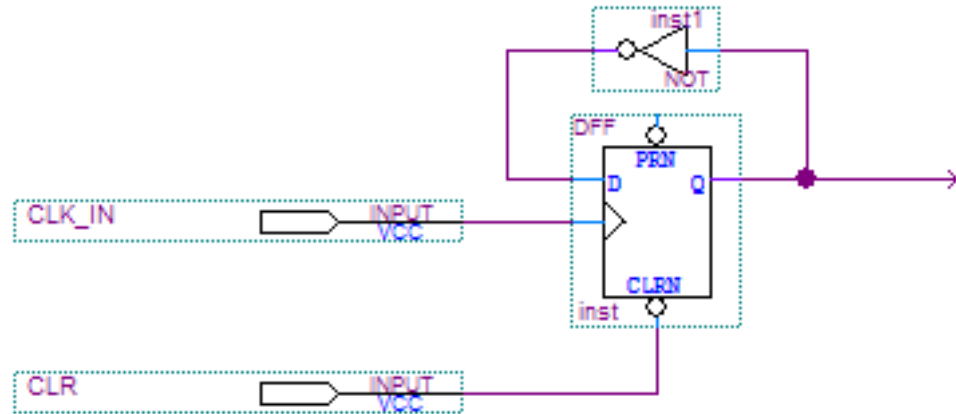
Invert waveform

Targets:

SDC command:

OK Cancel Help

# Generated Clock Example



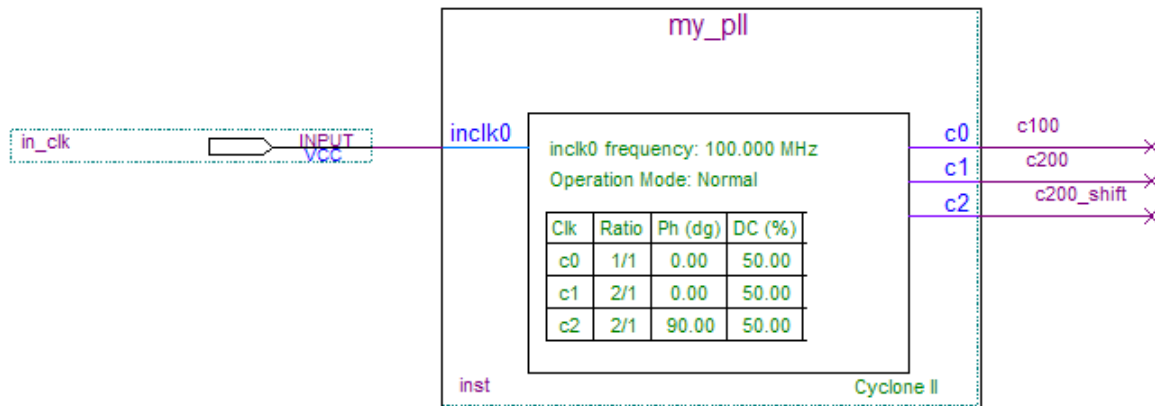
```
create_clock -period 10 [get_ports clk_in]
```

```
create_generated_clock -name clk_div \  
    -source [get_ports clk_in] \  
    -divide_by 2 \  
    [get_pins inst|regout]
```

# PLL Clocks

- Command: `derive_pll_clocks`
- Use to create generated clocks on all PLL outputs
  - Based on input clock & PLL settings
- Requires defining PLL input as clock

# derive\_pll\_clocks Example



## Using generated clock commands

```
create_clock -period 10.0 [get_ports in_clk]
create_generated_clock -name {inst|altpll_component|pll|clk[0]} \
  -source [get_pins {inst|altpll_component|pll|inclk[0]}] \
  -divide_by 1 \
  [get_pins {inst|altpll_component|pll|clk[0]}]
create_generated_clock -name {inst|altpll_component|pll|clk[1]} \
  -source [get_pins {inst|altpll_component|pll|inclk[0]}] \
  -multiply_by 2 \
  [get_pins {inst|altpll_component|pll|clk[1]}]
create_generated_clock -name {inst|altpll_component|pll|clk[2]} \
  -source [get_pins {inst|altpll_component|pll|inclk[0]}] \
  -multiply_by 2 \
  -phase 90 \
  [get_pins {inst|altpll_component|pll|clk[2]}]
```


## Using derive\_pll command

```
create_clock -period 10.0 \
  [get_ports in_clk]

derive_pll_clocks

# Note the clock names for
# the generated clocks
# will be the names of
# the PLL output pins
```

# SDC Timing Constraints

- Clocks
- I/O ← 
- False paths
- Multicycle paths
- Absolute delays

# I/O Constraining

- Specify system-level timing constraints
- Settings
  - Input maximum delay
  - Input minimum delay
  - Output maximum delay
  - Output minimum delay

# Input Min/Max Delay Definition

- External delays added to the input data arrival path
- Max value used in Setup and Min in Hold

Input Maximum Delay = 2ns

Path #1 Setup slack is 6.374

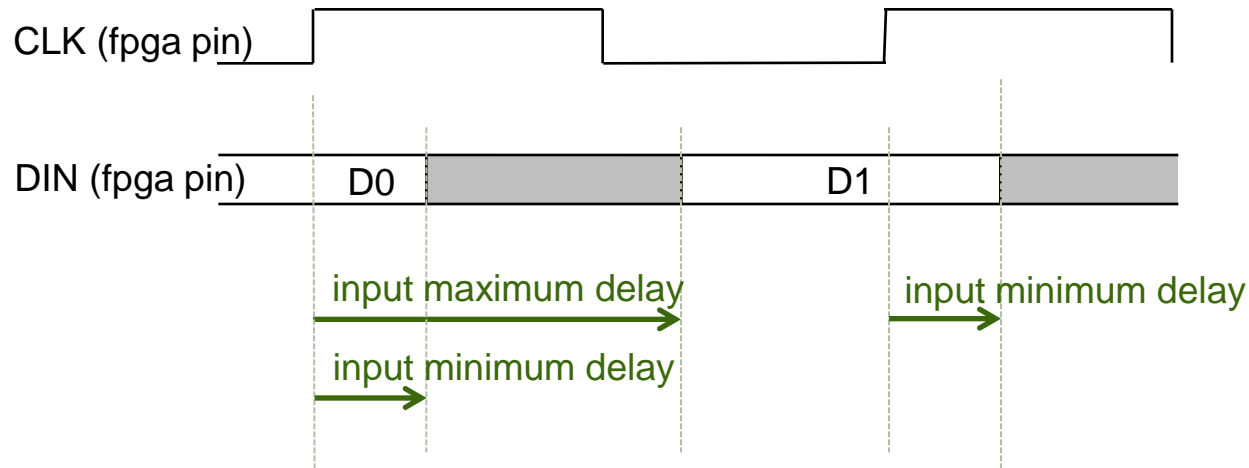
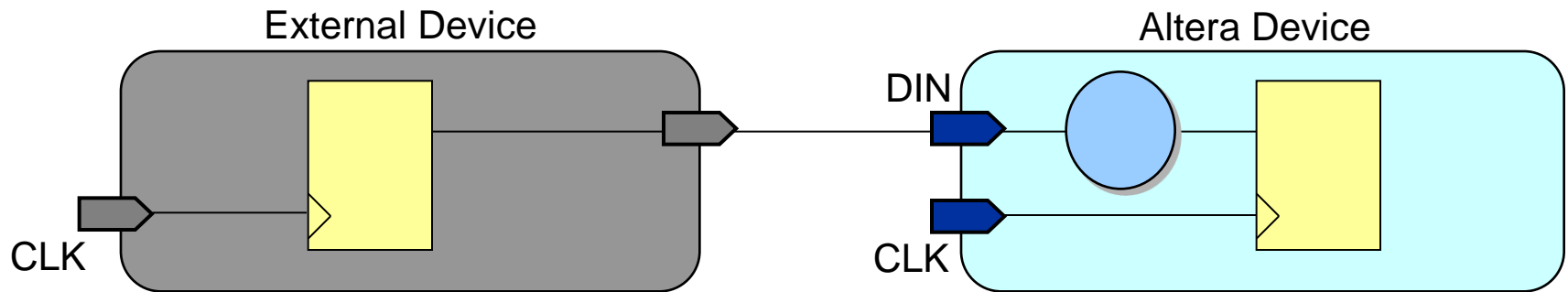
Path Summary		Statistics	Data Path	Waveform	Extra Fitter Information		
<b>Data Arrival Path</b>							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	0.000	0.000					clock path
1	0.000	0.000	R				clock network delay
3	2.000	2.000	F	Ext	1	PIN_B6	din
4	5.698	3.698					data path
1	2.000	0.000	FF	IC	1	IOIBUF_X14_Y31_N8	din~input i
2	2.681	0.681	FF	CELL	1	IOIBUF_X14_Y31_N8	din~input o
3	5.161	2.480	FF	IC	1	FF_X14_Y31_N10	inst d
4	5.698	0.537	FF	CELL	1	FF_X14_Y31_N10	inst
<b>Data Required Path</b>							
	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	12.181	2.181					clock path
1	10.000	0.000					source latency
2	10.000	0.000			1	PIN_J7	clk
3	10.000	0.000	RR	IC	1	IOIBUF_X16_Y0_N15	clk~input i
4	10.527	0.527	RR	CELL	1	IOIBUF_X16_Y0_N15	clk~input o
5	10.901	0.374	RR	IC	1	CLKCTRL_G17	clk~inputclkctrl jndk[0]
6	10.901	0.000	RR	CELL	3	CLKCTRL_G17	clk~inputclkctrl outclk
7	11.816	0.915	RR	IC	1	FF_X14_Y31_N10	inst clk
8	12.181	0.365	RR	CELL	1	FF_X14_Y31_N10	inst
3	12.161	-0.020					clock uncertainty
4	12.072	-0.089		uTsu	1	FF_X14_Y31_N10	inst

Input Minimum Delay = 1ns

Path #: Hold slack is 1.811

Path Summary		Statistics	Data Path	Waveform	Extra Fitter Information		
<b>Data Arrival Path</b>							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	0.000	0.000					clock path
1	0.000	0.000	R				clock network delay
3	1.000	1.000	R	Ext	1	PIN_B6	din
4	4.133	3.133					data path
1	1.000	0.000	RR	IC	1	IOIBUF_X14_Y31_N8	din~input i
2	1.507	0.507	RR	CELL	1	IOIBUF_X14_Y31_N8	din~input o
3	3.619	2.112	RR	IC	1	FF_X14_Y31_N10	inst d
4	4.133	0.514	RR	CELL	1	FF_X14_Y31_N10	inst
<b>Data Required Path</b>							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					latch edge time
2	2.250	2.250					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_J7	clk
3	0.000	0.000	RR	IC	1	IOIBUF_X16_Y0_N15	clk~input i
4	0.527	0.527	RR	CELL	1	IOIBUF_X16_Y0_N15	clk~input o
5	0.917	0.390	RR	IC	1	CLKCTRL_G17	clk~inputclkctrl jndk[0]
6	0.917	0.000	RR	CELL	3	CLKCTRL_G17	clk~inputclkctrl outclk
7	1.870	0.953	RR	IC	1	FF_X14_Y31_N10	inst clk
8	2.250	0.380	RR	CELL	1	FF_X14_Y31_N10	inst
3	2.270	0.020					clock uncertainty
4	2.322	0.052		uTh	1	FF_X14_Y31_N10	inst

# Input Min/Max Delay in Waveform



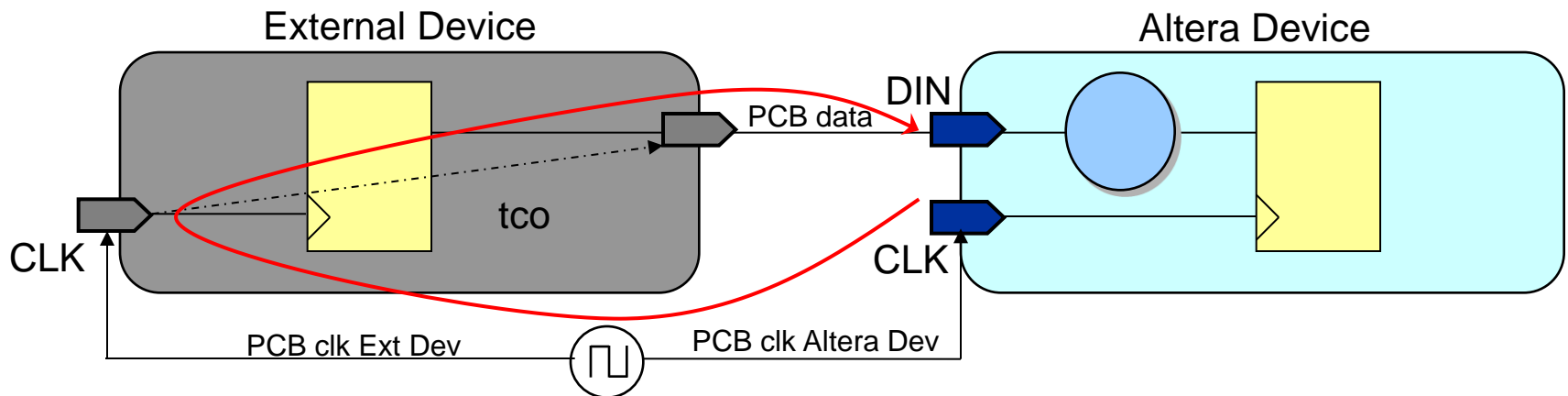
$$T_{\text{setup(Altera)}} = \text{Setup Relationship (=Tck)} - \text{Input Maximum Delay}$$

$$T_{\text{hold(Altera)}} = - \text{Hold Relationship (=0)} + \text{Input Minimum Delay}$$



# Input Maximum Delay Calculation (Example)

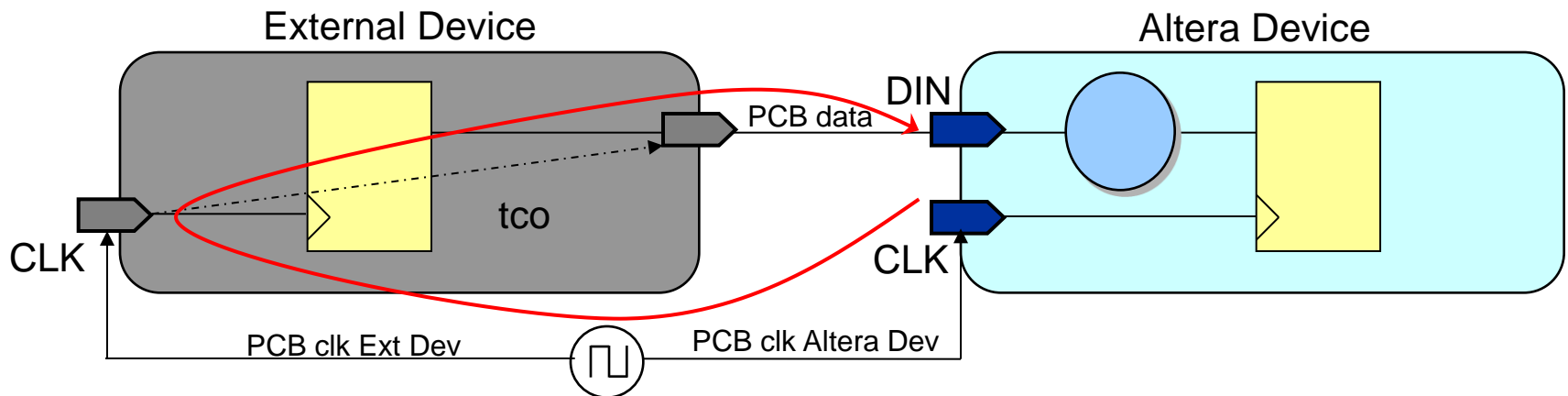
- Constraining registered input path in setup



- Input Maximum Delay =
  - PCB clk(Altera Dev) + PCB clk(Ext Dev)
  - + Tco max(Ext Dev)
  - + PCB data max

# Input Minimum Delay Calculation (Example)

- Constraining registered input path in hold



- Input Minimum Delay =
  - PCB clk(Altera Dev) + PCB clk(Ext Dev)
  - +  $T_{co \text{ min}}(\text{Ext Dev})$
  - + PCB data min

# Output Min/Max Delay Definition

- External delays added to the output data arrival path
  - in fact, subtracted to the output data required path !
- Max value used in Setup and Min in Hold analysis

Output Maximum Delay = 2ns

Path #1 Setup slack is 2.145

Data Arrival Path							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	▲ 2.332	2.332					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_J7	clk
3	0.000	0.000	RR	IC	1	IOIBUF_X16_Y0_N15	clk~input i
4	0.527	0.527	RR	CELL	1	IOIBUF_X16_Y0_N15	clk~input o
5	0.917	0.390	RR	IC	1	CLKCTRL_G17	clk~inputclkctrl indk[0]
6	0.917	0.000	RR	CELL	3	CLKCTRL_G17	clk~inputclkctrl outclk
7	1.870	0.953	RR	IC	1	DDIOOUTCELL_X12_Y31_N4	inst2 clk

Data Required Path							
	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	▲ 10.000	0.000					clock path
1	10.000	0.000	R				clock network delay
3	9.980	-0.020					clock uncertainty
4	7.980	-2.000	R	oExt	0	PIN_A7	dout

Output Minimum Delay = -1ns

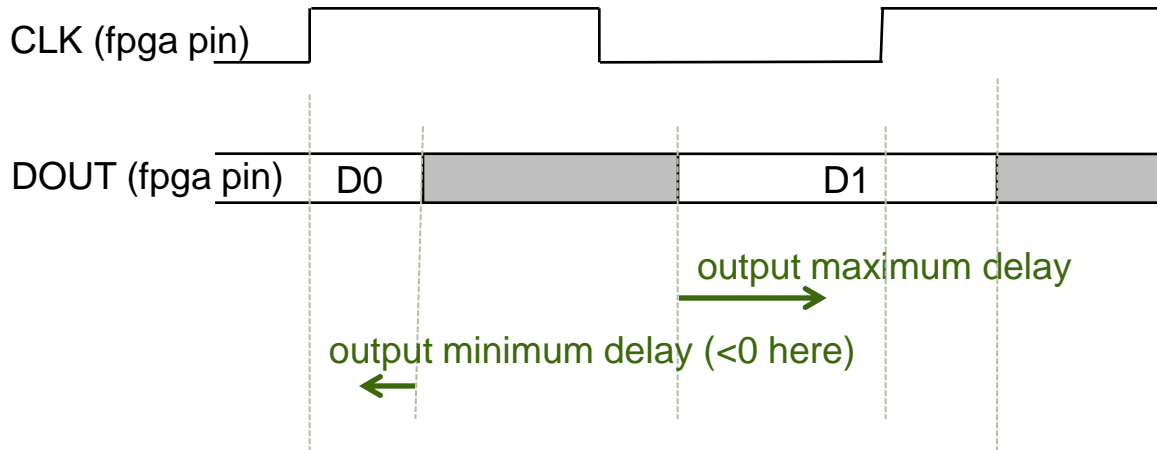
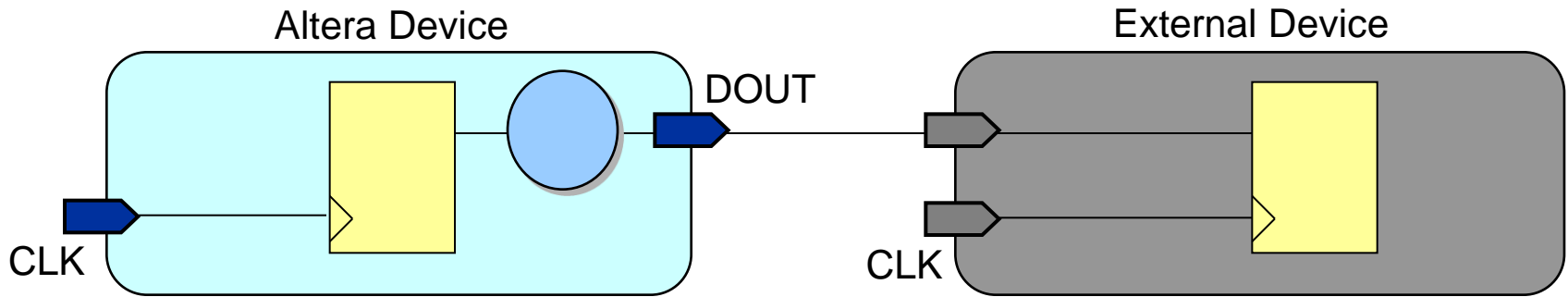
Path #1 Hold slack is 4.444

Data Arrival Path							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	▲ 2.260	2.260					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_J7	clk
3	0.000	0.000	RR	IC	1	IOIBUF_X16_Y0_N15	clk~input i
4	0.527	0.527	RR	CELL	1	IOIBUF_X16_Y0_N15	clk~input o
5	0.901	0.374	RR	IC	1	CLKCTRL_G17	clk~inputclkctrl indk[0]
6	0.901	0.000	RR	CELL	3	CLKCTRL_G17	clk~inputclkctrl outclk
7	1.816	0.915	RR	IC	1	DDIOOUTCELL_X12_Y31_N4	inst2 clk

Data Required Path							
	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					latch edge time
2	▲ 0.000	0.000					clock path
1	0.000	0.000	R				clock network delay
3	0.020	-0.020					clock uncertainty
4	1.020	1.000	F	oExt	0	PIN_A7	dout

# Output Min/Max Delay in Waveform

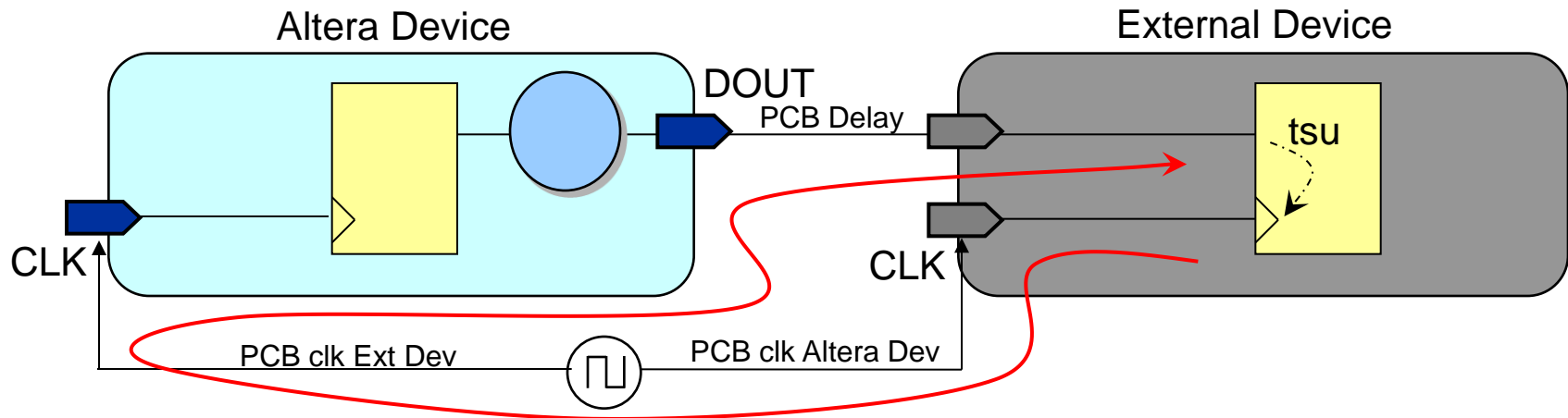


$$T_{co \max}(\text{Altera}) + \text{Output Maximum Delay} \leq \text{Setup Relationship } (=T_{ck})$$

$$T_{co \min}(\text{Altera}) + \text{Output Minimum Delay} \geq \text{Hold Relationship } (=0)$$

# Output Maximum Delay Calculation (Example)

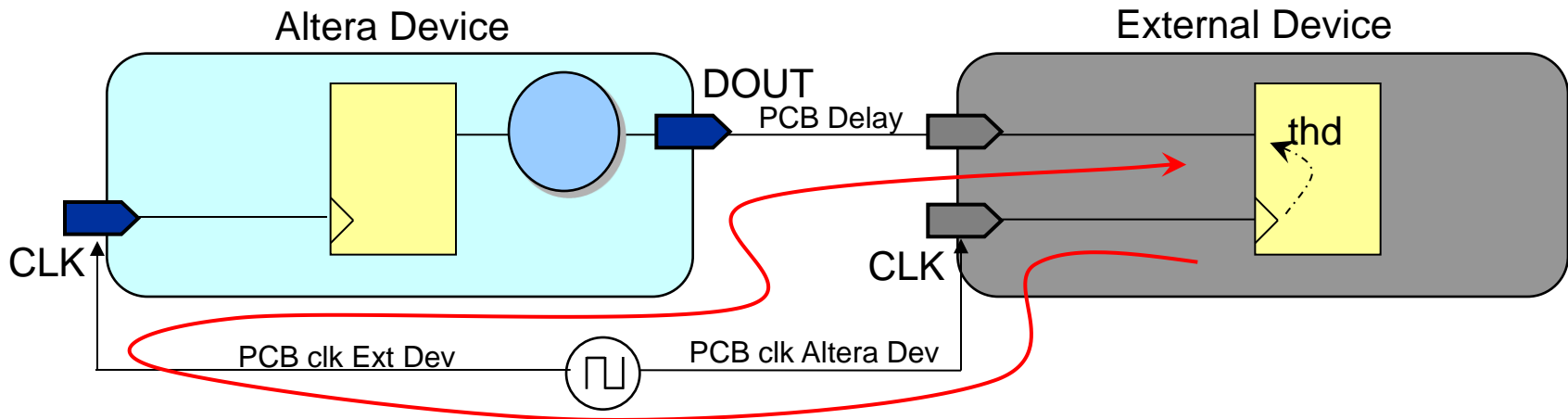
- Constraining registered output path in setup



- Output Maximum Delay =
  - PCB clk(Ext Dev) + PCB clk(Altera Dev)
  - + PCB data max
  - + Tsu(Ext Dev)

# Output Minimum Delay Calculation (Example)

- Constraining registered output path in hold



- Output Minimum Delay =
  - PCB clk(Ext Dev) + PCB clk(Altera Dev)
  - + PCB data min
  - Thd(Ext Dev)

# set\_input\_delay Command

- Constrains input pins by specifying external device timing parameters

- Options

```
-clock <clock_name>  
[-clock_fall]  
[-rise | -fall]  
[-max | -min]  
[-add_delay]  
[-reference_pin <target>]  
[-source_latency_included]  
<delay value>  
<targets>
```

# set\_output\_delay Command

- Constrains output pins by specifying external device timing parameters

- Options

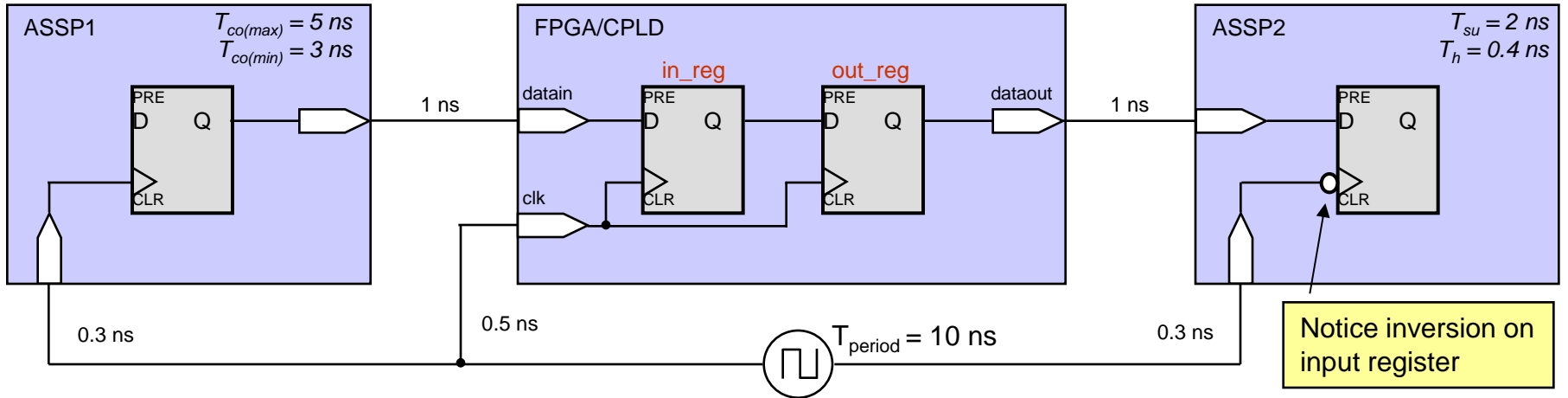
```
-clock <clock_name>  
[-clock_fall]  
[-rise | -fall]  
[-max | -min]  
[-add_delay]  
[-reference_pin <target>]  
<delay value>  
<targets>
```



# Input/Output Delays (GUI)



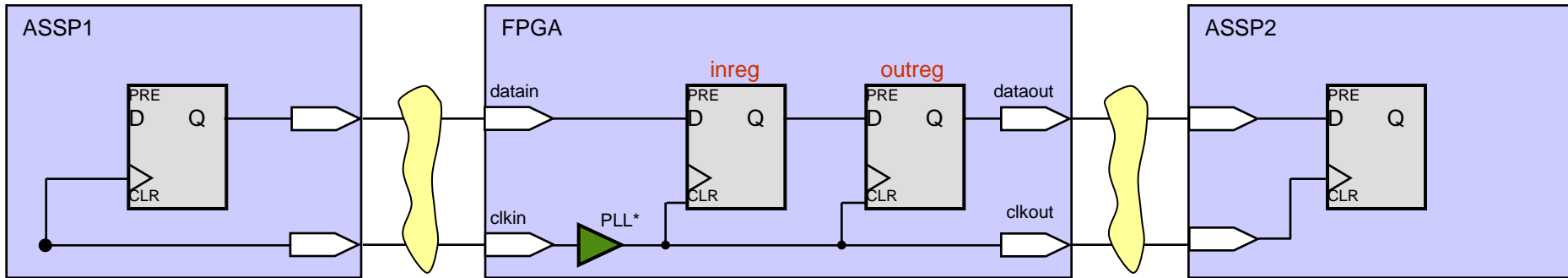
# Synchronous I/O Example



```
set_input_delay -clock clk -max [expr -0.5 + 0.3 + 5 + 1] datain
set_input_delay -clock clk -min [expr -0.5 + 0.3 + 3 + 1] datain
```

```
set_output_delay -clock clk -max [expr -0.3 + 0.5 + 1 + 2 ] -clock_fall dataout
set_output_delay -clock clk -min [expr -0.3 + 0.5 + 1 -0.4] -clock_fall dataout
```

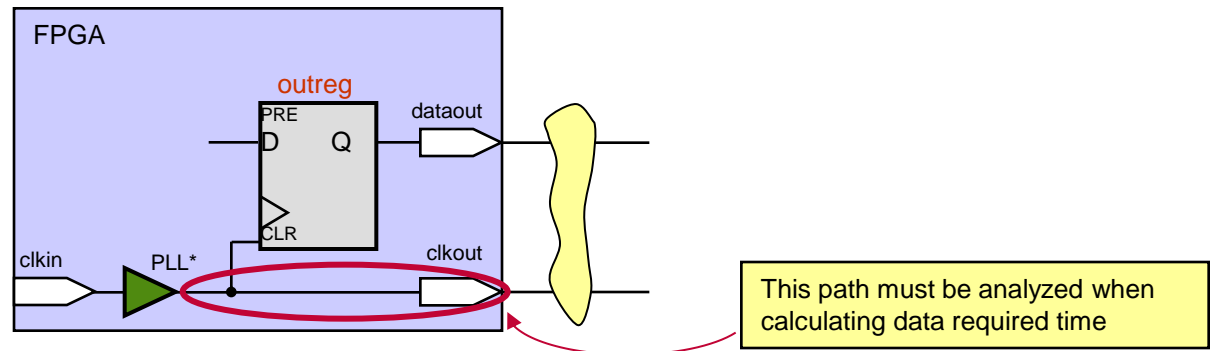
# Source-Synchronous Interfaces



- Both data & clock transmitted by host device with designated phase relationship (e.g. edge or center-aligned)
  - No clock tree skew included in calculation
  - Target device uses transmitted clock to sample incoming data
- Data & clock routed identically to maintain phase relationship at destination device
  - Board delay not included in external delay calculations
    - Clock trace delay (data required time) & Data trace delay (data arrival time) are equal and offset
  - Enables higher interface speeds (compared to using system clock)

*\*PLL, represented by 1 symbol, could be multiple PLLs or generating multiple clock outputs*

# Using SDC with Source-Synch



- Must tell TimeQuest to analyze path from clock source to output clock pin during analysis
- Two Methods using `set_output_delay` command
  - Method 1
    - Add `-reference_pin` argument to output delay assignment
      - `-reference_pin` is a TimeQuest SDC extension
  - Method 2
    1. Assign generated clock on output clock pin
    2. Use `-clock` argument in output delay assignment to associate output clock to output data bus

# Method 1 Example

```
create_clock 5 -name clkin \  
    [get_ports clkin]  
create_generated_clock -name pllclk divide_by 1 \  
    -source [get_ports clkin] \  
    [get_pins inst|altp11_component|pll|clk[0]]  
  
# Constrain dataout with an external tsu of 0.5 ns  
# and th of 0.5 ns using clkout as reference pin  
set_output_delay -clock pllclk \  
    -max 0.500 \  
    -reference_pin clkout \  
    [get_ports dataout]  
set_output_delay -clock pllclk \  
    -min -0.500 \  
    -reference_pin clkout \  
    [get_ports dataout]
```

# Method 2 Example

```
create_clock 5 -name clkin \  
    [get_ports clkin]  
create_generated_clock -name pllclk divide_by 1 \  
    -source [get_ports clkin] \  
    [get_pins inst|altp11_component|pll|clk[0]]
```

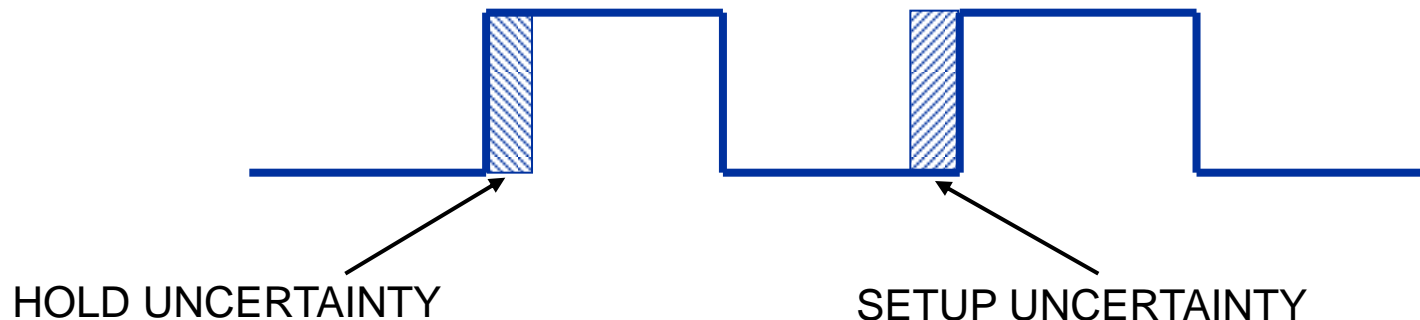
```
# Place clock on external clock output  
create_generated_clock -name clkout \  
    -source [get_pins inst|altp11_component|pll|clk[0]] \  
    [get_ports clkout]
```

```
# Constrain dataout with an external tsu of 0.5 ns  
# and th of 0.5 ns using clkout as clock  
set_output_delay -clock clkout \  
    -max 0.500 \  
    [get_ports dataout]  
set_output_delay -clock clkout \  
    -min -0.500 \  
    [get_ports dataout]
```

# Virtual Clocks

# Clock Uncertainty

- Setup uncertainty decreases setup required time
- Hold uncertainty increases hold required time





# Clock Uncertainty

- Command: `set_clock_uncertainty`
- Use to model jitter, guard band, or skew
  - Allows generation of clocks that are non-ideal
- Options
  - `[-setup | -hold]`
  - `[-fall_from <fall_from_clock>]`
  - `[-fall_to <fall_to_clock>]`
  - `[-from <from_clock>]`
  - `[-rise_from <rise_from_clock>]`
  - `[-rise_to <rise_to_clock>]`
  - `[-to <to_clock>]`
  - `<value>`

# SDC Enhancement - *derive\_clock\_uncertainty*

- New SDC constraint - *derive\_clock\_uncertainty*
  - Computes all the required clock uncertainty for every clock transfer (e.g. PLL settings)
- Advantage in using *derive\_clock\_uncertainty*
  - Automatically applies SDC constraint, *set\_clock\_uncertainty*
  - Provides flexibility for accounting system considerations
    - Optionally, modify generated uncertainty values to account for external factors (e.g. board skew, external PLL jitter, etc.)
    - Optionally, disregard if accounted for in timing constraints and/or in controlled environment

The screenshot shows the SDC editor with the following code:

```
# Original Clock Setting
create_clock -period 8.0
             mac_clk
# -----
derive_pll_clocks
derive_clock_uncertainty
```

The console output shows the following information:

```
Info: Deriving Clock Uncertainty
Info: set_clock_uncertainty -from i_clkkin_13ns -to i_clkkin_13ns -setup 0.090
Info: set_clock_uncertainty -from i_clkkin_13ns -to i_clkkin_13ns -hold 0.050
```

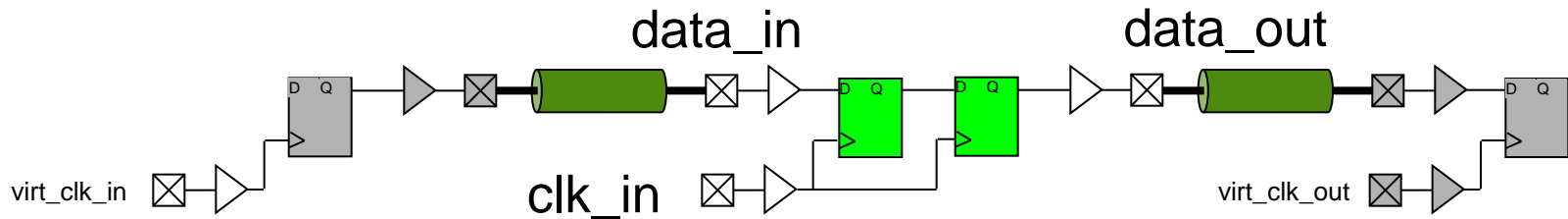
The resulting timing report table is as follows:

5	11.039	3.730	FF	IC	3	PLL_6	PACE PLL1_2 altpll_component U1 pll inclk[0]
6	3.420	-7.619	FF	COMP	2	PLL_6	PACE PLL1_2 altpll_component U1 pll clk[0]
7	4.723	1.303	FF	IC	1	CLKCTRL_G4	PACE PLL1_2 altpll_component U1 _clk0~clkctrl inclk[0]
8	4.723	0.000	FF	CELL	2	CLKCTRL_G4	PACE PLL1_2 altpll_component U1 _clk0~clkctrl outclk
9	5.366	0.643	FF	IC	1	LCFF_X15_Y1_N21	clk13ns_tail_m_Zclk
10	5.994	0.619	FF	CELL	1	LCFF_X15_Y1_N21	clk13ns_tail_m
11	5.894	-0.090					clock uncertainty
12	5.994	0.998			1	LCFF_X15_Y1_N21	clk13ns_tail_m

# Virtual Clocks

- Virtual clocks could be defined for all input clocks and output clocks in current design
  - Virtual input clock for external launch clock in the constraint for the input case
  - Virtual output clock for external latch clock in the constraint for the output case.
- This will separate the clock domains for the IO and core.
- This is required to set the correct clock uncertainty numbers in the clock uncertainty constraints, which differ for the core and the IO transfers
- Not used for output source synchronous interface (output generated clock)

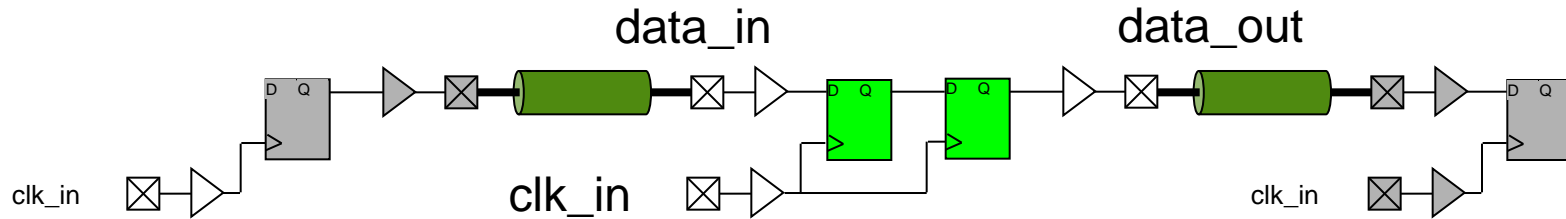
# Virtual Clocks



- Traditionally, you'd set input delay and output delay with `clk_in` as the reference clock
  - This intra-clock transfer would appear the same as the IO clock transfer
  - Incorrect clock uncertainty would be applied for IO transfers
- Virtual clocks for input transfer and output transfer provide separate and distinct clock transfers from `clk_in` core intra-clock transfer.

# Virtual Clocks

## Case A: BAD!



- **Old Constraints:**

```
create_clock -period 10 -name clk_in [get_ports {clk_in}]
set_input_delay -clock [get_clocks {clk_in}] -max 2 [get_ports {data_in}]
set_input_delay -clock [get_clocks {clk_in}] -min 0 [get_ports {data_in}]
set_output_delay -clock [get_clocks {clk_in}] -max 2 [get_ports {data_out}]
set_output_delay -clock [get_clocks {clk_in}] -min 0 [get_ports {data_out}]
```

- **Clock Transfers**

From `clk_in` to `clk_in` only

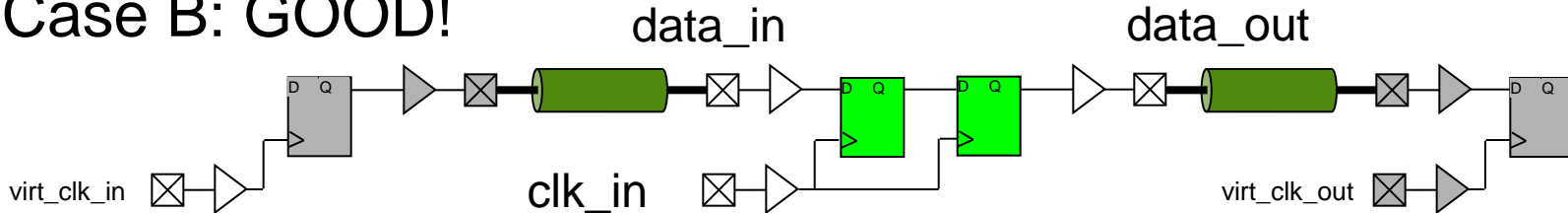
- **Resultant Clock Uncertainty**

From `clk_in` to `clk_in`

**Setup: 150ps Hold: 50ps**

# Virtual Clocks

## Case B: GOOD!



### ■ New Constraints:

```
create_clock -period 10 -name clk_in [get_ports {clk_in}]
create_clock -period 10 -name virt_clk_in
create_clock -period 10 -name virt_clk_out
set_input_delay -clock [get_clocks {virt_clk_in}] -max 2 [get_ports {data_in}]
set_input_delay -clock [get_clocks {virt_clk_in}] -min 0 [get_ports {data_in}]
set_output_delay -clock [get_clocks {virt_clk_out}] -max 2 [get_ports {data_out}]
set_output_delay -clock [get_clocks {virt_clk_out}] -min 0 [get_ports {data_out}]
```


### ■ Clock Transfers

From `clk_in` to `clk_in`  
From `virt_clk_in` to `clk_in`  
From `clk_in` to `virt_clk_out`

### ■ Resultant Clock Uncertainty

From `clk_in` to `clk_in`                      **Setup: 150ps Hold: 50ps**  
From `virt_clk_in` to `clk_in`                    **Setup: 130ps Hold: 130ps**  
From `clk_in` to `virt_clk_out`                   **Setup: 130ps Hold: 130ps**

# SDC Timing Constraints

- Clocks
- I/O
- False paths 
- Multicycle paths
- Absolute delays

# False Paths

## ■ Logic-based

- Paths not relevant during for circuit operations
- e.g. Test logic, static or quasi-static registers

## ■ Timing-based

- Paths intentionally not analyzed by designer
- e.g. Bridging asynchronous clock domains using synchronizer circuits

## ■ Must be marked by constraint to tell TimeQuest to ignore them



# Two Methods to Create False Paths

- `set_false_path` **command**
  - Use when particular nodes are involved
  - Examples
    - All paths from an input pin to a set of registers
    - All paths from a register to another clock domain
  
- `set_clock_groups` **command**
  - Use when just clock domains are involved

# set\_false\_path Command

- Indicates paths that should be ignored during fitting and timing analysis

- Options

- `[-fall_from <names>]`

- `[-rise_from <names>]`

- `[-from <names>]`

- `[-through <names>]`

- `[-to <names>]`

- `[-fall_to <names>]`

- `[-rise_to <names>]`

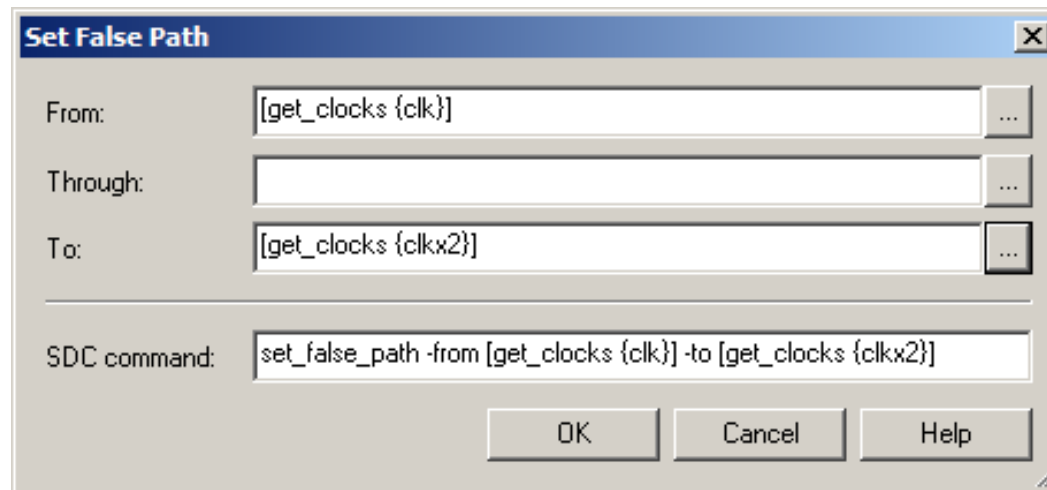
- `[-setup]`

- `[-hold]`

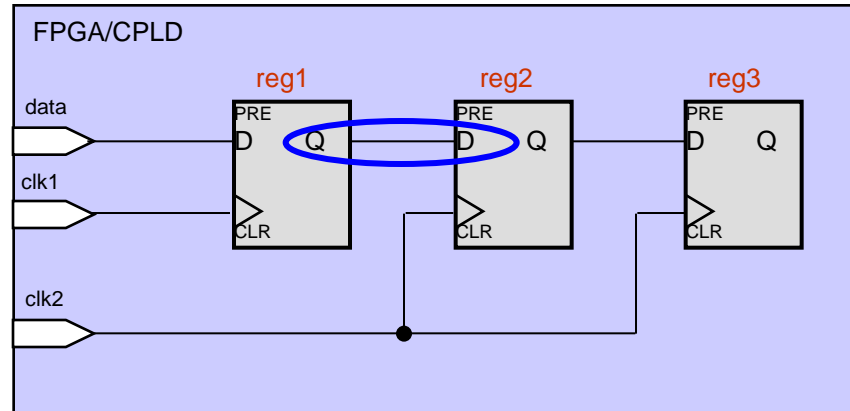
- `<targets>`

# Set False Path (GUI)

Constraints menu



# False Path Example



Simple synchronizer circuit between two asynchronous clock domains

```
Set_false_path -from [get_pins reg1|regout] \  
-to [get_pins reg2|datain]
```

# set\_clock\_groups Command

- Tells fitter and TimeQuest to ignore **ALL** paths between specified clock domains
  - Great for clock muxes
  - Equivalent to setting false paths (`-from & -to`) on all paths between domains

## ■ Options

`[-asynchronous | -exclusive]`

`-group <clock name>`

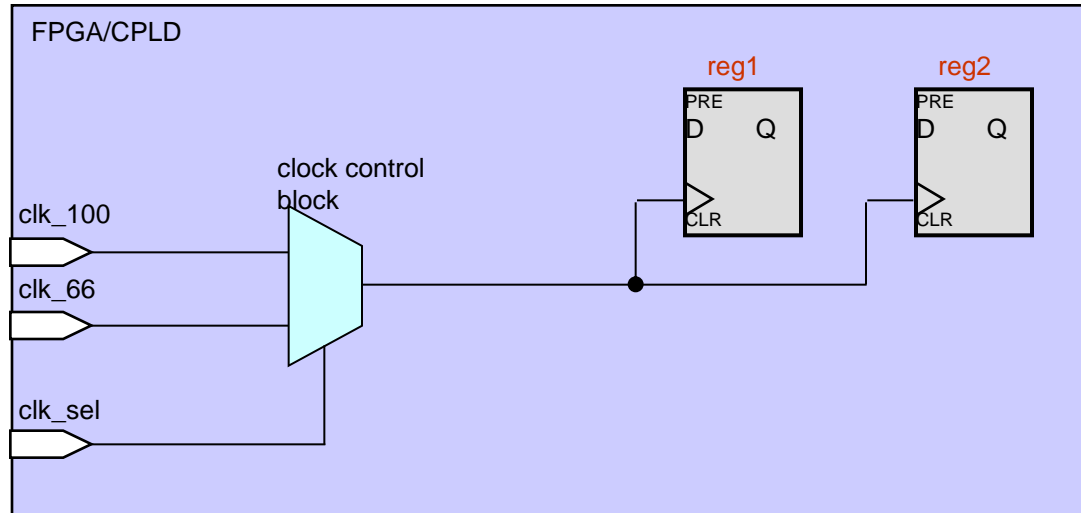
`-group <clock_name>`

`[-group <clock name>]...`

# Asynchronous clock domains example

```
set_clock_groups -asynchronous \
\
-group { clkin1 \
    inst1|altp11_component|pll|clk[0] \
    inst1|altp11_component|pll|clk[1] } \
\
-group { clkin2 \
    inst2|altp11_component|pll|clk[0] \
    inst2|altp11_component|pll|clk[1] \
    inst2|altp11_component|pll|clk[2] } \
\
-group {PCI_CLOCK }
```

# Clock Mux Example



```
create_clock -period 10.0 [get_ports clk_100]
create_clock -period 15.0 [get_ports clk_66]
```

```
set_clock_groups -exclusive -group {clk_100} -group {clk_66}
# Since clocks are muxed, TimeQuest should not analyze
# cross-domain paths as only one clock will be driving the
# registers at any one time.
```

# SDC Timing Constraints

- Clocks
- I/O
- False paths
- Multicycle paths
- Absolute delays





# When to Use Multicycle Paths

- Design does not require single cycle to transfer data
  - Otherwise needlessly over-constrain paths
- Clocks are integer multiples of each other with or without offset
- Clock enables ensuring register(s) not sampling data every clock edge

# Multicycle Types (1)

## ■ Destination

- Constraint based on destination clock edges
- Moves latch edge backward to relax required setup/hold time
- Used in most multicycle situations

## ■ Source

- Constraint based on source clock edges
- Moves launch edge forward to relax required setup/hold time
- Useful when source clock is at higher frequency than destination

# Multicycle Types (2)

## ■ Setup

- Increases the number of cycles for setup analysis
- Default is 1

## ■ Hold

- Increases the number of cycles for hold analysis
- Default is 0\*

*\*Note: Subtract 1 from the Classic Timing Analyzer hold multicycle value to convert to SDC*

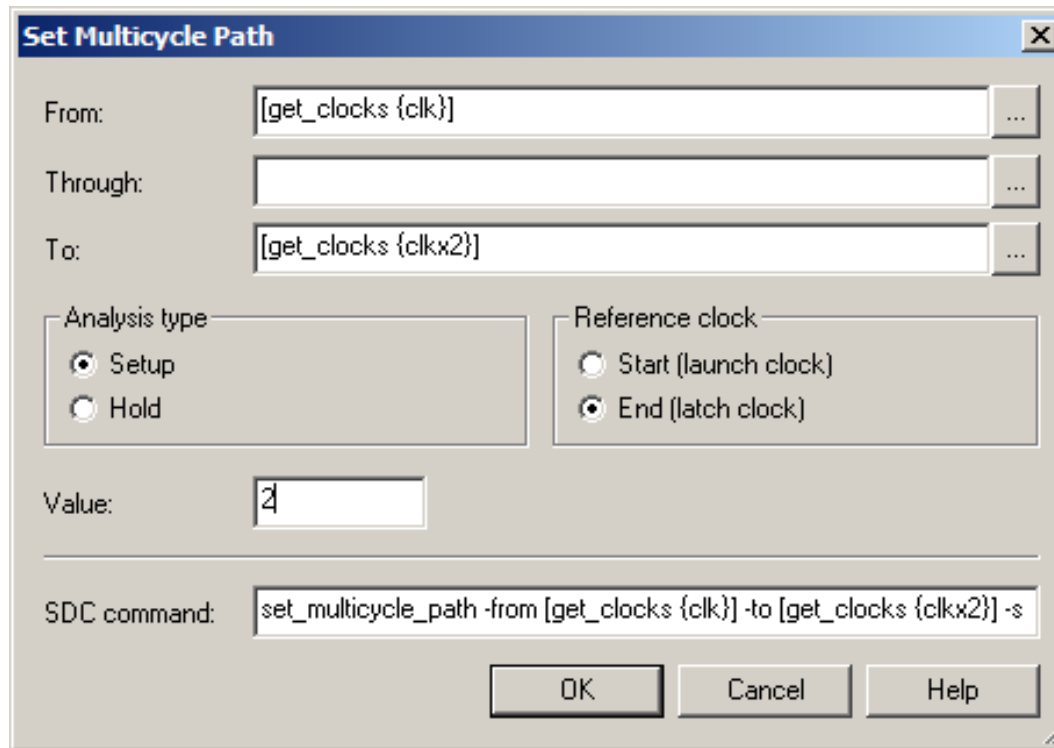
# set\_multicycle\_path Command

- Indicates by how many cycles the required time (setup or hold) should be extended from defaults

- Options

```
[-start | -end]  
[-setup | -hold]  
[-fall_from <names>]  
[-rise_from <names>]  
[-from <names>]  
[-through <names>]  
[-to <names>]  
[-fall_to <names>]  
[-rise_to <names>]  
<targets>  
<value>
```

# Set Multicycle Path (GUI)



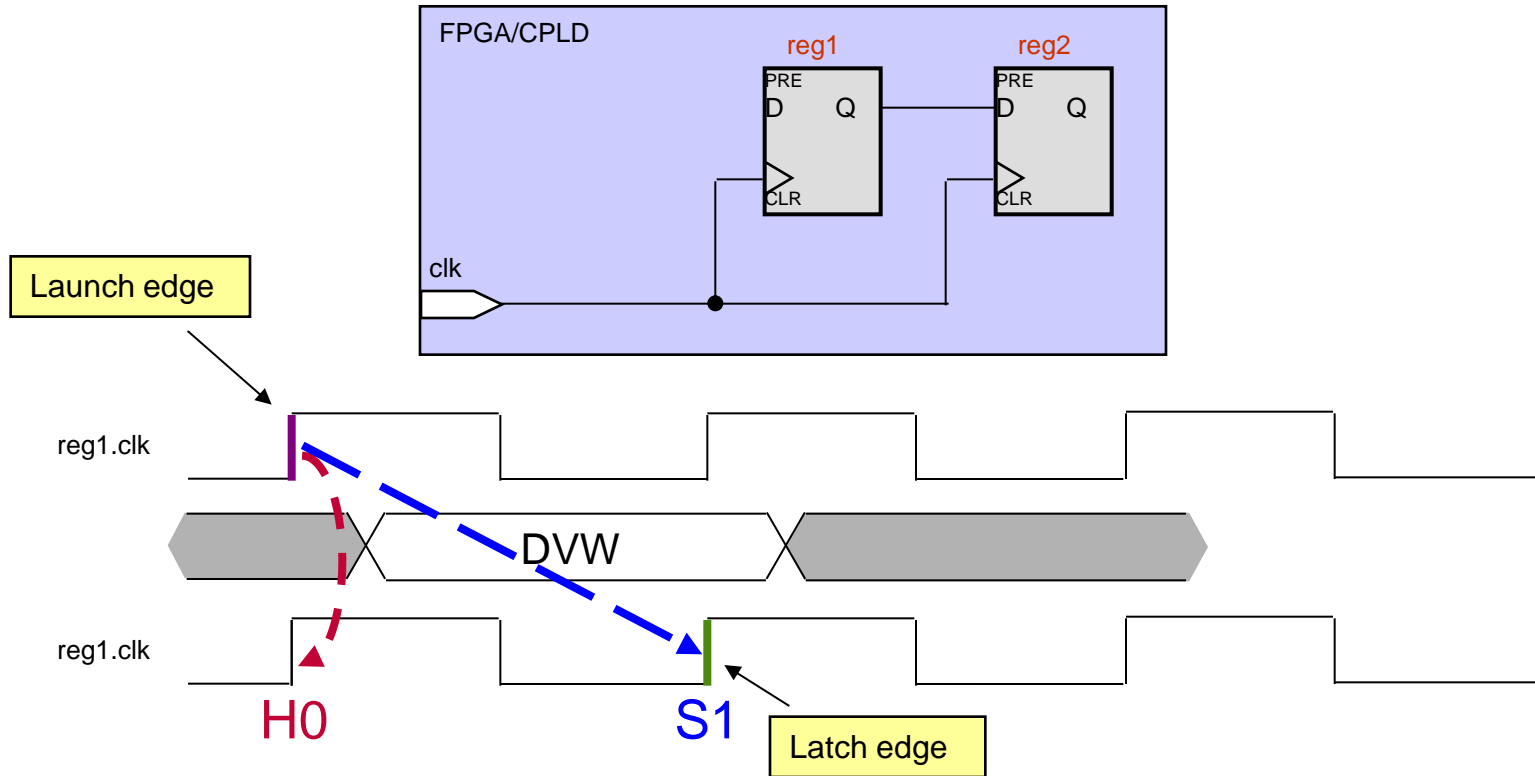
The screenshot shows a dialog box titled "Set Multicycle Path" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- From:** A text field containing the text `[get_clocks {clk}]` and a browse button (...).
- Through:** An empty text field and a browse button (...).
- To:** A text field containing the text `[get_clocks {clkx2}]` and a browse button (...).
- Analysis type:** A group box containing two radio buttons:  Setup and  Hold.
- Reference clock:** A group box containing two radio buttons:  Start (launch clock) and  End (latch clock).
- Value:** A text field containing the number `2`.
- SDC command:** A text field containing the command `set_multicycle_path -from [get_clocks {clk}] -to [get_clocks {clkx2}] -s`.

At the bottom of the dialog are three buttons: **OK**, **Cancel**, and **Help**.

# Understanding Multicycle (1)

## Standard single-cycle register transfer



— — • Multicycle Setup = 1 (Default)

- - - - Multicycle Hold = 0 (Default)\*

© 2010 Altera Corporation—Confidential

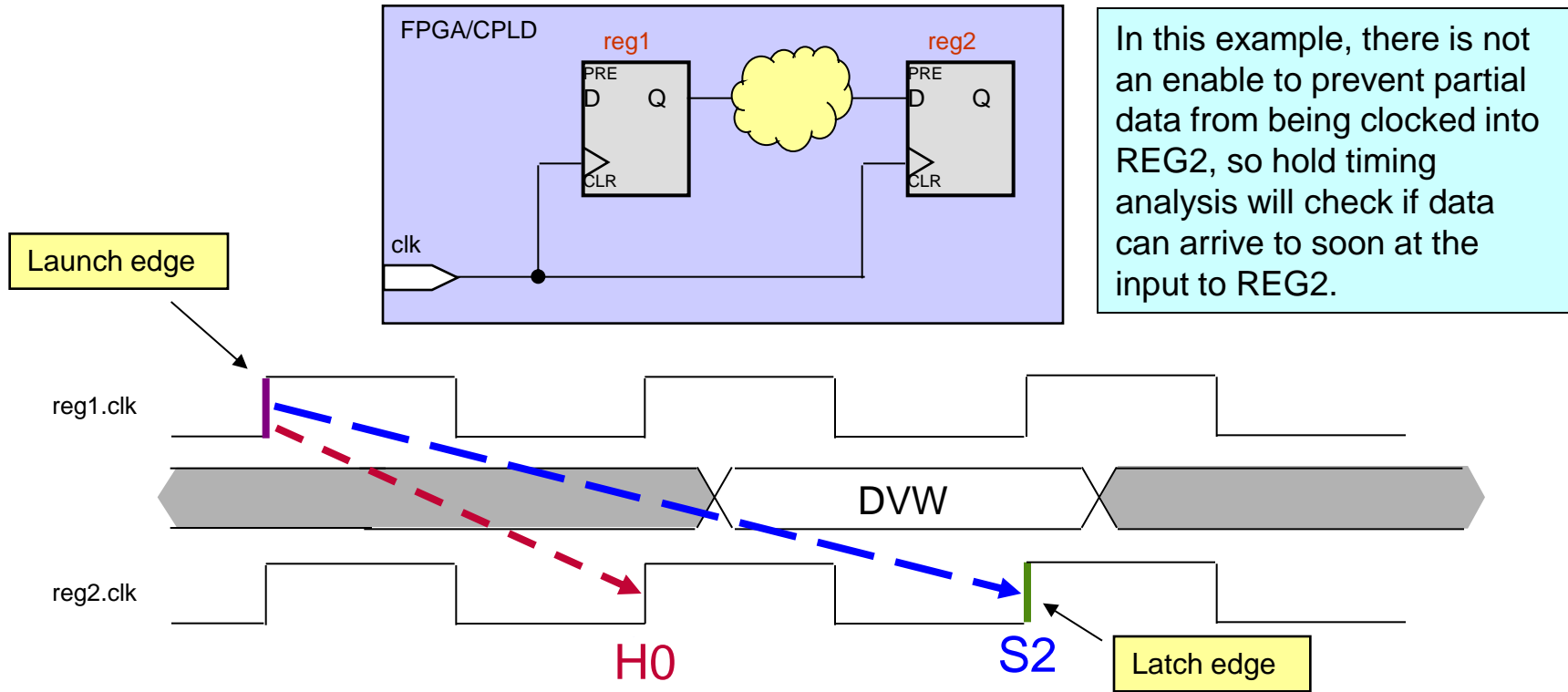
ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.

*Default hold edge is one edge before/after setup edge*

ALTERA®

# Understanding Multicycle (2)

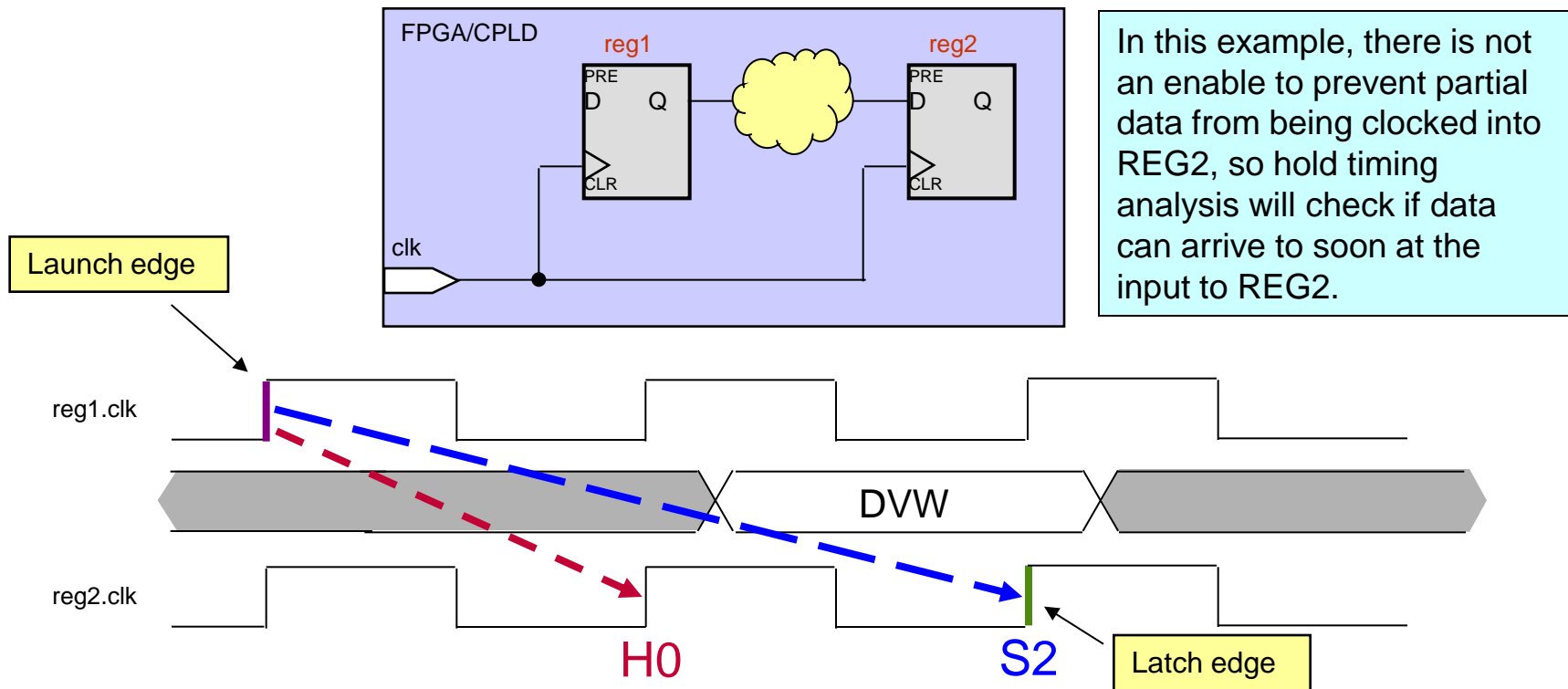
Change to a *two cycle setup*; *single cycle hold* transfer



- — • Multicycle Setup = 2
- - - - Multicycle Hold = 0 (Default)

# Understanding Multicycle (2) (cont.)

Change to a *two cycle setup*; *single cycle hold* transfer

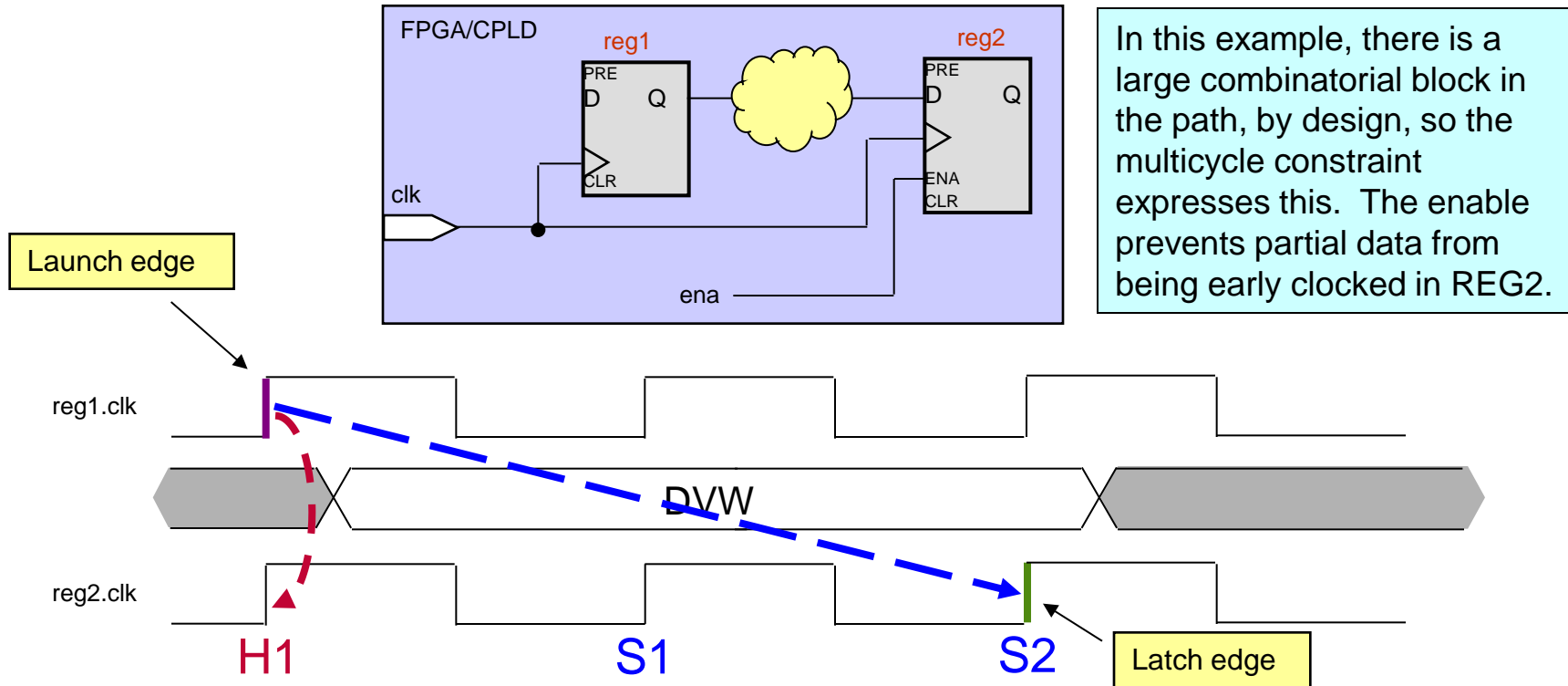


```
Set_multicycle_path -from [get_pins reg1|regout] -to [get_pins reg2|datain] \  
-end -setup 2
```



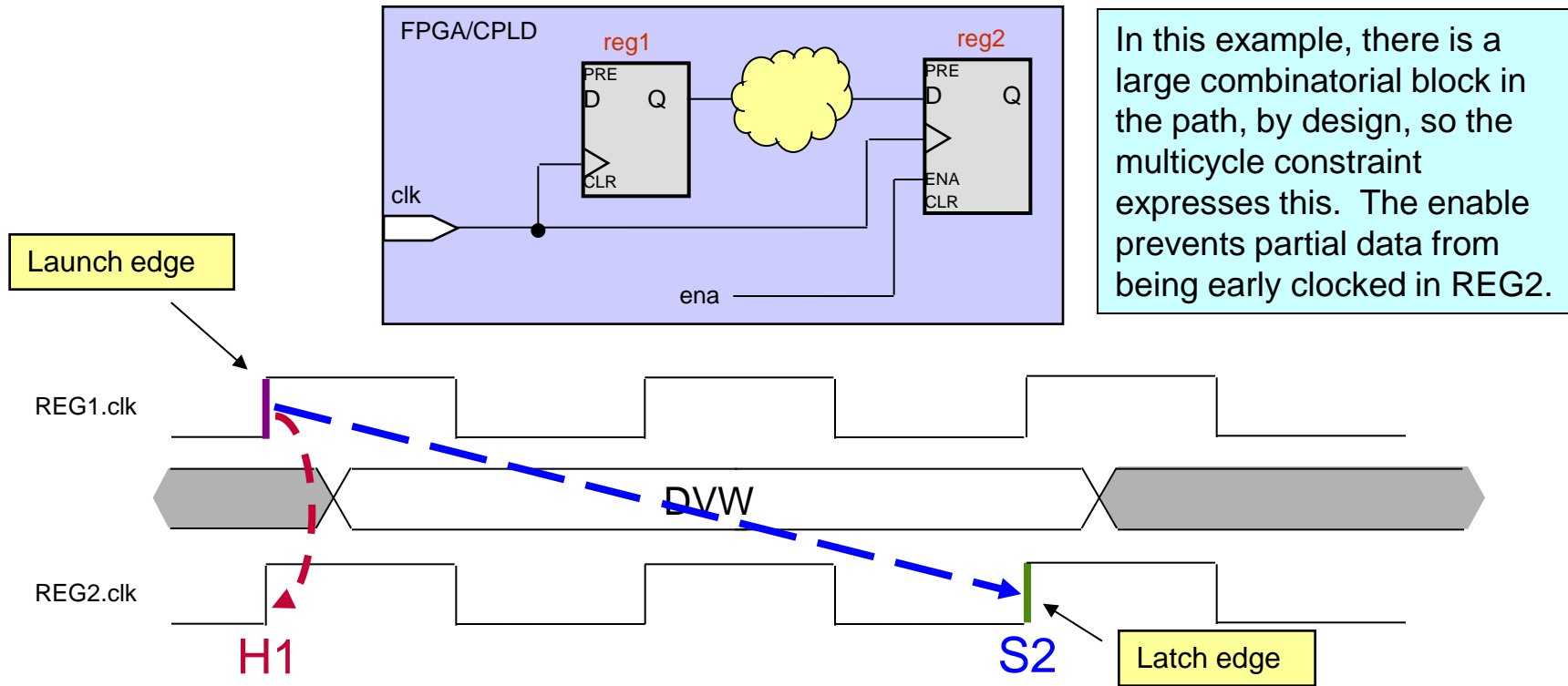
# Understanding Multicycle (3)

Change to a *two cycle setup*; *two cycle hold* transfer



# Understanding Multicycle (3) (cont.)

Change to a *two cycle setup*; *two cycle hold* transfer



```
Set_multicycle_path -from [get_pins reg1|regout] -to [get_pins reg2|datain] \
    -setup 2
Set_multicycle_path -from [get_pins reg1|regout] -to [get_pins reg2|datain] \
    -hold 1
```

# Reporting Multicycles

**Report Timing: set\_multicycle\_path**

Command Info | Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock
1	2.643	y_regtwo[2]	...OBSERVABLEDATAB_REGOUT2	c100	c200

Same path with Setup Multicycle = 2

---

**Path #1: Setup slack is 2.643**

Path Summary | Statistics | Data Path | Waveform

**Data Arrival Path**

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	0.091	0.091	R				clock network delay
3	0.341	0.250		uTco	1	LCCFF_X27_Y7_N7	y_regtwo[2]
4	0.341	0.000	RR	CELL	1	LCCFF_X27_Y7_N7	y_regtwo[2]regout
5	0.341	0.000	RR	IC	1	LCCOMB_X27_Y7_N6	inst24[inst2]datab
6	0.664	0.323	RR	CELL	1	LCCOMB_X27_Y7_N6	inst24[inst2]combout
7	0.909	0.245	RR	IC	1	LCCOMB_X27_Y7_N0	inst24[inst11[2]]datab
8	1.058	0.149	RR	CELL	1	LCCOMB_X27_Y7_N0	inst24[inst11[2]]combout
9	1.303	0.245	RR	IC	1	LCCOMB_X27_Y7_N26	inst24[inst12[2]]datab
10	1.452	0.149	RR	CELL	1	LCCOMB_X27_Y7_N26	inst24[inst12[2]]combout
11	1.700	0.248	RR	IC	1	LCCOMB_X27_Y7_N10	inst24[inst13[2]]datab

**Data Required Path**

	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	10.136	0.136	R				clock network delay
3	10.089	-0.047		uTsu			...LEDATAB_REGOUT2

Latch edge extended by one destination clock cycle

---

**Path #1: Setup slack is 2.643**

Path Summary | Statistics | Data Path | Waveform

	Property	Value
1	From Node	y_regtwo[2]
2	To Node	...ABLEDATAB_REGOUT2
3	Launch Clock	c100
4	Latch Clock	c200
5	Multicycle - Setup End	2
6	Data Arrival Time	7.446
7	Data Required Time	10.089
8	Slack	2.643

# SDC Timing Constraints

- Clocks
- I/O
- False paths
- Multicycle paths
- Absolute delays



# Absolute Delays

- Applies a timing value to a particular path
- Overrides the current setup/hold information for the path derived from clock and I/O constraints
- Apply `set_max_delay` & `set_min_delay` constraints to paths

# Absolute Delay Example

- Specify a input port-to-register or register-to-output port constraint without using input & output delays
- Use `-rise_from/-fall_from` & `-rise_to/-fall_to` to restrict timing value to only registers responding to a rising or falling edge transition

Ex. DDR input

```
# Apply a 2ns max delay for an input port only to nodes clocked by  
# the rising edge of clock CLK  
set_max_delay -from [get_ports in[*]] -rise_to [get_clocks CLK] 2.000
```

# Timing Analysis Agenda

- TimeQuest flow
- Timing constraints
- Timing reports



# Verifying Clocks & I/O Timing

- Use Clock Setup & Hold summary reports to check worst slack for each clock

*“Did I make it or did I not make it?”*

- Positive slack displayed in **black**, negative in **red**
- Obtaining summary reports
  - Use `create_timing_summary` Tcl command
  - TimeQuest folder of Compilation Report
  - Run Report Setup Summary & Report Hold Summary reports from Tasks pane or Reports menu
- For detailed slack/path analysis
  - Run Report Timing from Tasks pane or Constraints menu
  - Use `report_timing` command



# TimeQuest Reports in Compilation Report

	Clock	Slack	End Point TNS
1	clk_x2	0.295	0.000
2	clk_x1	0.846	0.000
3	clk_out	2.256	0.000

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

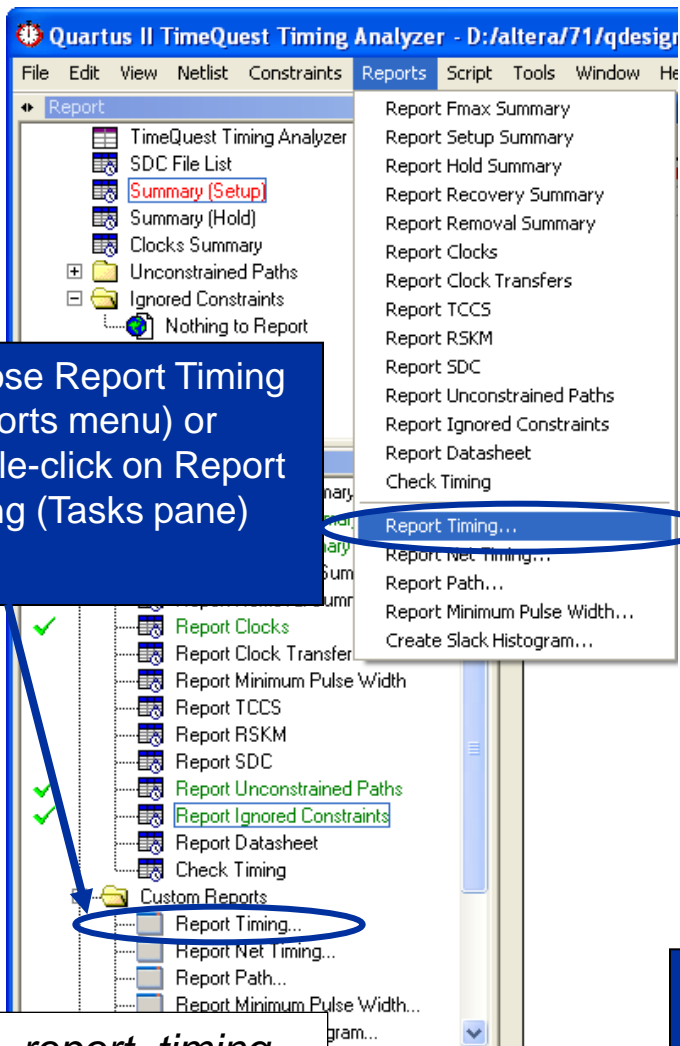
- SDC files used during fitting
- Clocks generated
- Timing violations
- Unconstrained paths
- Worst-case paths

Reports generated in TimeQuest GUI

# 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
    - Recommendation: Do not place reporting commands in SDC file

# Report Timing (GUI)

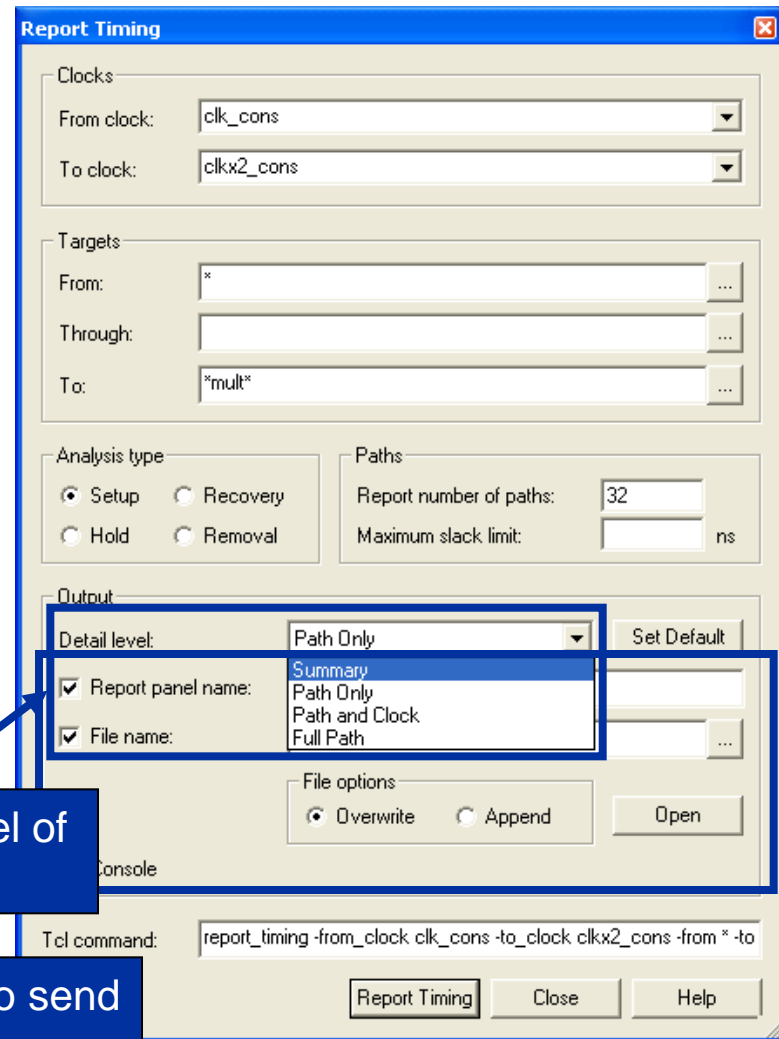


Choose Report Timing (Reports menu) or double-click on Report Timing (Tasks pane)

Tcl: `report_timing`

© 2010 Altera Corporation—Confidential

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.



Select level of detail

Select where to send output report

# Summary Slack/Path Report

```
report_timing -from_clock c100 -to_clock c200 \
  -setup -npaths 10 -detail summary \
  -panel_name "Setup (c100 to c200) Summary"
```

	Slack	From Node	To Node	Launch Clock	Latch Clock
1	1.748	b_regtwo[3]	...mult1~OBSERVABLEDATAB_REGOUT3	c100	c200
2	2.071	b_regtwo[2]	...mult1~OBSERVABLEDATAB_REGOUT2	c100	c200
3	2.141	inst1 altpll_component pll clk[0]	...mult1~OBSERVABLEDATAB_REGOUT3	c100	c200
4	2.141	inst1 altpll_component pll clk[0]	...mult1~OBSERVABLEDATAB_REGOUT3	c100	c200
5	2.236	b_regtwo[4]	...mult1~OBSERVABLEDATAB_REGOUT4	c100	c200
6	2.286	y_regtwo[3]	...mult1~OBSERVABLEDATAB_REGOUT3	c100	c200
7	2.463	inst1 altpll_component pll clk[0]	...mult1~OBSERVABLEDATAB_REGOUT2	c100	c200
8	2.463	inst1 altpll_component pll clk[0]	...mult1~OBSERVABLEDATAB_REGOUT2	c100	c200
9	2.487	b_regtwo[5]	...mult1~OBSERVABLEDATAB_REGOUT5	c100	c200
10	2.511	y_regtwo[2]	...mult1~OBSERVABLEDATAB_REGOUT2	c100	c200

# Detailed Slack/Path Report

```
report_timing -from_clock c100 -to_clock c200 \
  -setup -npaths 10 -detail path_only \
  -panel_name "Setup (c100 to c200)"
```

The screenshot shows the 'Setup (c100 to c200)' report window. A red arrow points to the first row of the 'Summary of Paths' table. A blue bracket indicates that four detailed views are available for each path. Two detailed views for Path #1 are shown below, with yellow callouts highlighting specific sections.

Slack	From Node	To Node	Launch Clock	Latch Clock
1 1.748	b_regtwo[3]	...EDATAB_REGOUT3	c100	c200
2 2.071	b_regtwo[2]	...EDATAB_REGOUT2	c100	c200
3 2.141	...ll_componentpllclk[0]	...EDATAB_REGOUT3	c100	c200
4 2.141	...ll_componentpllclk[0]	...EDATAB_REGOUT3	c100	c200
5 2.236	b_regtwo[4]	...EDATAB_REGOUT4	c100	c200
6 2.286	y_regtwo[3]	...EDATAB_REGOUT3	c100	c200
7 2.463	...ll_componentpllclk[0]	...EDATAB_REGOUT2	c100	c200
8 2.463	...ll_componentpllclk[0]	...EDATAB_REGOUT2	c100	c200
9 2.487	b_regtwo[5]	...EDATAB_REGOUT5	c100	c200
10 2.511	y_regtwo[2]	...EDATAB_REGOUT2	c100	c200

Property	Value
1 From Node	b_regtwo[3]
2 To Node	...~OBSERVABLEDATAB_REGOUT3
3 Launch Clock	c100
4 Latch Clock	c200
5 Data Arrival Time	3.330
6 Data Required Time	5.078
7 Slack	1.748

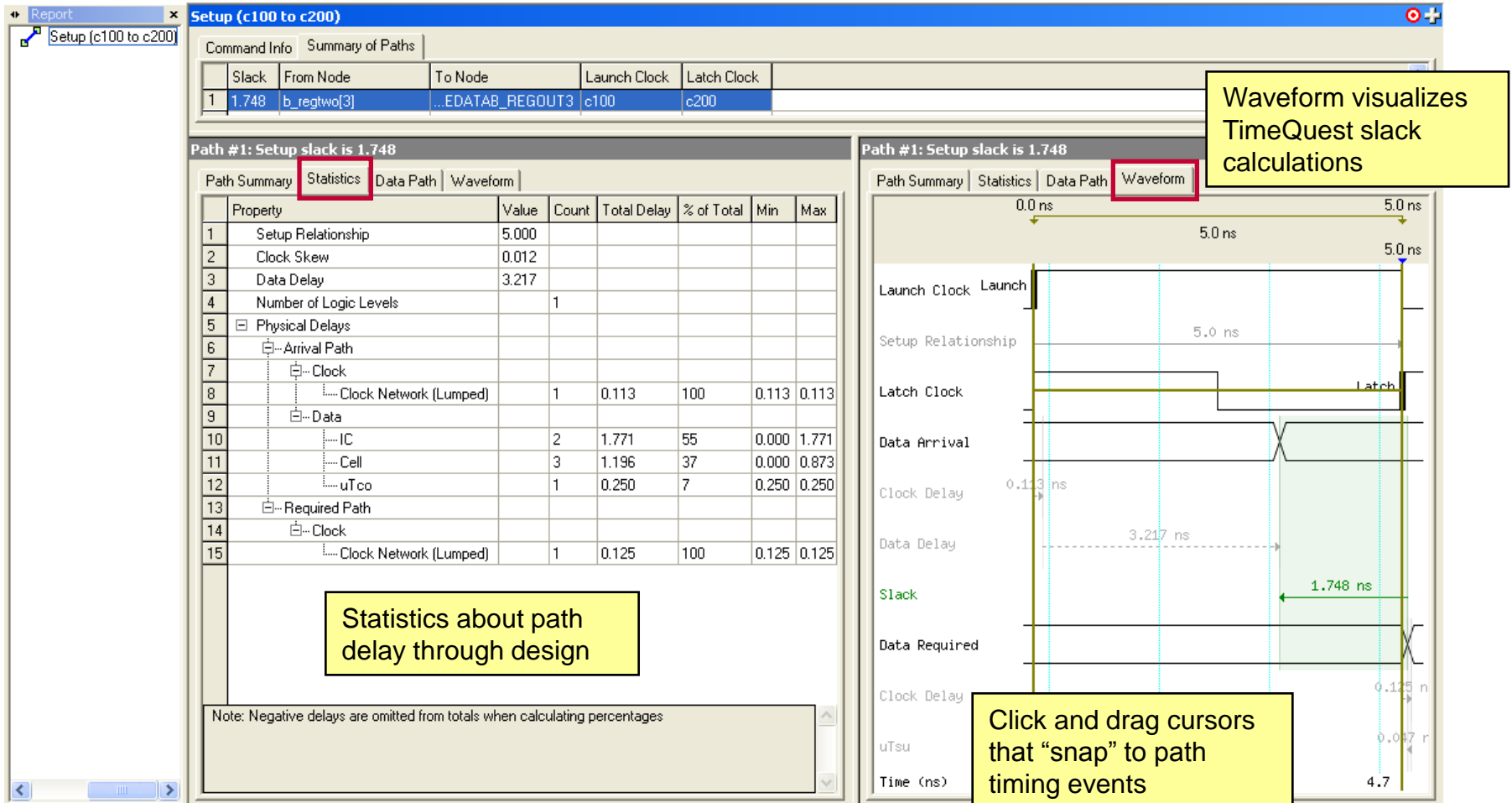
Data Arrival Path						
	Total	Incr	RF	Type	Fanout	Location
1	0.000	0.000				
2	0.113	0.113	R			clock network delay
3	0.363	0.250		uTco	1	LCFF_X26_Y1_N9
4	0.363	0.000	RR	CELL	1	LCFF_X26_Y1_N9
5	0.363	0.000	RR	IC	1	LCCOMB_X26_Y1_N8
6	0.686	0.323	RR	CELL	1	LCCOMB_X26_Y1_N8
7	2.457	1.771	RR	IC	1	DSPMULT_X16_Y8_N0
8	3.330	0.873	RR	CELL	13	DSPMULT_X16_Y8_N0

Data Required Path						
	Total	Incr	RF	Type	Fanout	Location
1	5.000	5.000				latch edge time
2	5.125	0.125	R			clock network delay
3	5.078	-0.047		uTsu	13	DSPMULT_X16_Y8_N0

# Detailed Slack/Path Report (cont.)

```
report_timing -from_clock c100 -to_clock c200 \
  -setup -npaths 10 -detail path_only \
  -panel_name "Setup (c100 to c200)"
```



# Report Path

- Report arbitrary point-to-point path (does not need to be constrained)

## Nodes

Specifies what you are looking for

The screenshot shows the 'Report Path' dialog box with the following details:

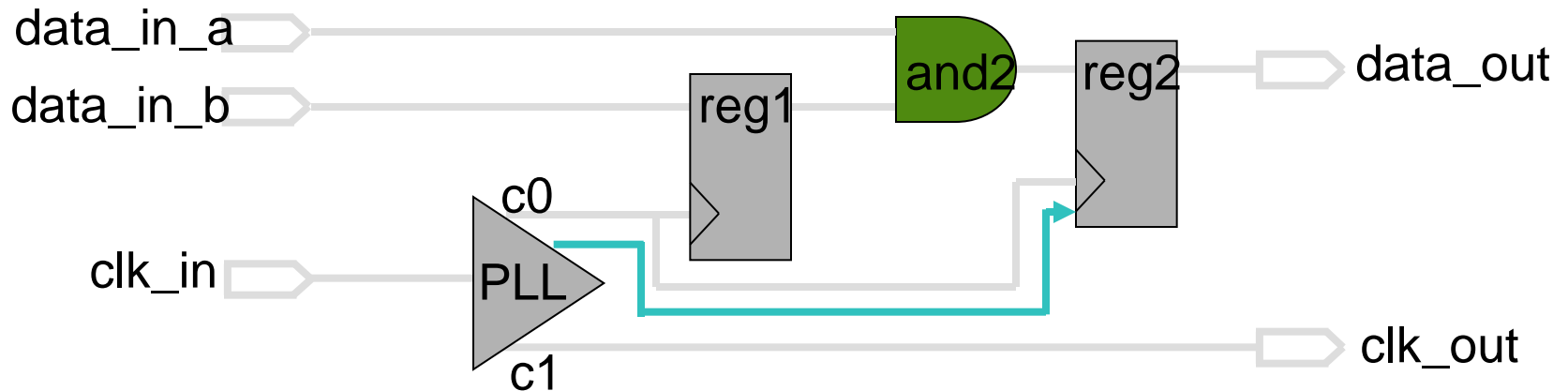
- Targets:** Three input fields for 'From:', 'Through:', and 'To:'.
- Paths:** A field for 'Report number of paths:' with the value '1'.
- Output:** A checked checkbox for 'Report panel name:' with the value 'Report Path'. An unchecked checkbox for 'File name:' is present. Below these are 'File options' with radio buttons for 'Overwrite' and 'Append', and an 'Open' button.
- Console:** An unchecked checkbox.
- Tcl command:** A text field containing the command: `report_path -npaths 1 -panel_name "Report Path"`.
- Buttons:** 'Report Path', 'Close', and 'Help' buttons at the bottom.

## Output options

Specifies output customizations

- Does not go flow through keepers

# Report Path



```
>report_path -from [get_pins {inst1|altpll_component|pll|clk[0]}] -to [get_pins \
{reg2|clk}] -npaths 1
```

Info: Path #1: Delay is 1.974

```
Info: =====
Info: From Node   : inst1|altpll_component|pll|clk[0]
Info: To Node     : reg2|clk
Info:
Info: Path:
Info:
Info: Total (ns)   Incr (ns)   Type   Node
Info: =====
Info: 0.000        0.000                       inst1|altpll_component|pll|clk[0]
Info: 1.303        1.303        RR IC  inst1|altpll_component|_clk0~clkctrl|inclk[0]
Info: 1.303        0.000        RR CELL inst1|altpll_component|_clk0~clkctrl|outclk
Info: 1.974        0.671        RR IC  reg2|clk
Info:
Info: Total Path Delay : 1.974
Info: =====
```



# Report Clocks (report\_clocks)

- List details about the properties of constrained clocks

Clock properties

	Clock Name	Type	Period	Frequency	Rise	Fall	Duty Cycle	Divide by	Multiply by	Phase	Inverted	Master	Source	Targets
1	c100	Generated	10.000	100.0 MHz	0.000	5.000	50.00	1	1		false	clk_in_100mhz	inst1 altpll_component pll inclk[0]	{ inst1 altpll_component pll clk[0] }
2	c100_out	Generated	10.000	100.0 MHz	-3.610	1.390	50.00	1	1	-130.0	false	clk_in_100mhz	inst1 altpll_component pll inclk[0]	{ inst1 altpll_component pll clk[2] }
3	c200	Generated	5.000	200.0 MHz	0.000	2.500	50.00	1	2		false	clk_in_100mhz	inst1 altpll_component pll inclk[0]	{ inst1 altpll_component pll clk[1] }
4	clk_in_100mhz	Base	10.000	100.0 MHz	0.000	5.000								{ clk_in_100mhz }
5	clkout	Generated	10.000	100.0 MHz	6.390	11.390		1	1		false	c100_out	inst1 altpll_component pll clk[2]	{ clkout }

Clock names  
(-name argument or default name)

# Unconstrained Path Report

Quartus II TimeQuest Timing Analyzer - C:/altera\_trn/Quartus\_II\_Software\_Design\_Ser

File Edit View Netlist Constraints Reports Script Tools Window Help

Report

- TimeQuest Timing Analyzer Summary
  - Unconstrained Paths
    - Unconstrained Paths Summary**
    - Clock Status Summary
  - Setup Analysis
  - Hold Analysis

Property	Setup	Hold
1 Illegal Clocks	0	0
2 Unconstrained Clocks	1	1
3 Unconstrained Input Ports	34	34
4 Unconstrained Input Port Paths	51	51
5 Unconstrained Output Ports	32	32
6 Unconstrained Output Port Paths	32	32

Unconstrained Paths Summary Report indicates how many clock nodes are unconstrained (along with other unconstrained paths)

Clock Status Summary Report lists each clock found and whether it was constrained

Quartus II TimeQuest Timing Analyzer - C:/altera\_trn/Quartus\_II\_Software\_D

File Edit View Netlist Constraints Reports Script Tools Window Help

Report

- TimeQuest Timing Analyzer Summary
  - Unconstrained Paths
    - Unconstrained Paths Summary
    - Clock Status Summary**
  - Setup Analysis
  - Hold Analysis

Clock	Type	Status
1 clk_test	Base	Constrained
2 clk_in_100mhz	Base	Unconstrained

# Verifying False Paths & Groups

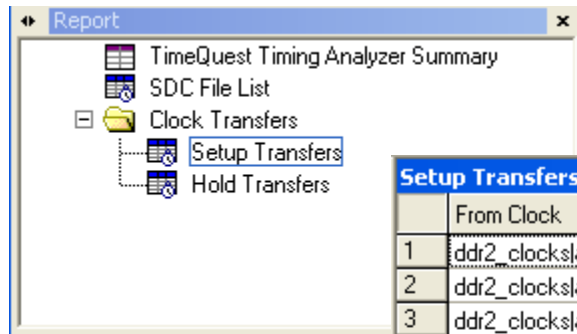
## ■ False paths

- Check ignored constraints
- Perform report timing on specified paths to ensure no results are returned

## ■ Clock groups

- Check clock transfers to ensure no paths are returned

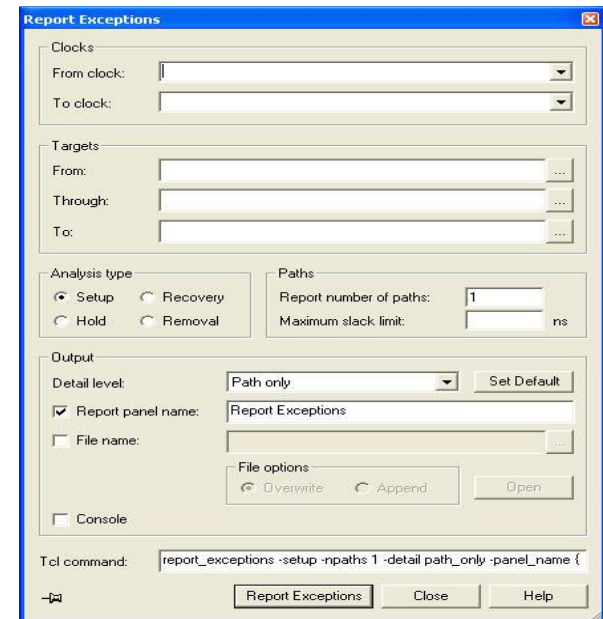
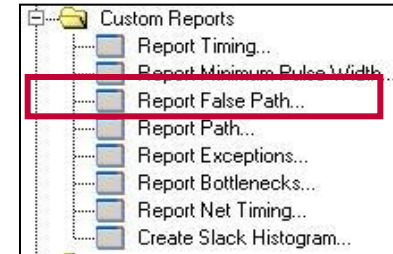
# Report Clock Transfers



Setup Transfers						
	From Clock	To Clock	RR Paths	FR Paths	RF Paths	FF Paths
1	ddr2_clocks\altpll_component\pllclk[0]	ddr2_clocks\altpll_component\pllclk[0]	26198	0	19	0
2	ddr2_clocks\altpll_component\pllclk[3]	ddr2_clocks\altpll_component\pllclk[0]	64	0	0	0
3	ddr2_clocks\altpll_component\pllclk[0]	ddr2_clocks\altpll_component\pllclk[1]	108	0	0	0
4	ddr_ck0_nddr_dqs[0]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
5	ddr_ck0_nddr_dqs[1]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
6	ddr_ck0_nddr_dqs[2]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
7	ddr_ck0_nddr_dqs[3]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
8	ddr_ck0ddr_dqs[0]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
9	ddr_ck0ddr_dqs[1]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
10	ddr_ck0ddr_dqs[2]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
11	ddr_ck0ddr_dqs[3]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
12	ddr_ck1_nddr_dqs[0]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
13	ddr_ck1_nddr_dqs[1]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
14	ddr_ck1_nddr_dqs[2]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
15	ddr_ck1_nddr_dqs[3]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
16	ddr_ck1ddr_dqs[0]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
17	ddr_ck1ddr_dqs[1]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
18	ddr_ck1ddr_dqs[2]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
19	ddr_ck1ddr_dqs[3]	ddr2_clocks\altpll_component\pllclk[3]	0	16	0	0
20	dtw_read_ddr_dqs[0]	ddr2_clocks\altpll_component\pllclk[3]	0	false path	0	0
21	dtw_read_ddr_dqs[1]	ddr2_clocks\altpll_component\pllclk[3]	0	false path	0	0
22	dtw_read_ddr_dqs[2]	ddr2_clocks\altpll_component\pllclk[3]	0	false path	0	0

# Report Exceptions

- Debug potential timing failures due to unintended or conflicting timing exceptions
  - Wildcards and Tcl lists are very convenient and commonly used to easily constrain parts of the design (e.g. `set_false_path -from {sig[*] sig_ena sig_overflow} -to [all_outputs]`)
  - Users can easily miss paths, incorrectly relax paths, or accidentally override other exceptions
- Includes all exceptions that are set by the SDC constraints
  - `set_false_path`, `set_multicycle`, `set_min_delay`, and `set_max_delay`



Report Exceptions Summary						
	Status	Exception	Setup Slack	Hold Slack	Recovery Slack	Removal Slack
186	Complete	...R2_90_phy_alt_mem_phy_inst clk pll_reconfig_reset_ams_n clm }	-3.492	n/a	n/a	n/a
187	Complete	... quartusDDR2_90_phy_alt_mem_phy_inst clk pfs_shift_busy_siild }	-3.492	n/a	n/a	n/a
188	Invalid	... } {mem_ba[0]} {mem_ba[1]} mem_cas_n mem_ras_n mem_we_n }	n/a	n/a	n/a	n/a
189	Partially Overridden	... {mem_ba[0]} {mem_ba[1]} mem_cas_n mem_ras_n mem_we_n } 2	4.997	n/a	n/a	n/a

# Timequest resource center

[www.altera.com/support/software/timequest/sof-qts-timequest.html](http://www.altera.com/support/software/timequest/sof-qts-timequest.html)

## TimeQuest Timing Analyzer Resource Center

[Home](#) > [Support](#) > [Design Software](#) > [TimeQuest Timing Analyzer](#)

The TimeQuest timing analyzer is an ASIC-strength static timing analyzer that supports the industry-standard Synopsys Design Constraints (SDC) format. This page provides links to resources where you can learn more about the TimeQuest analyzer.

For resources on the TimeQuest analyzer, see the following:

- [TimeQuest Analyzer Documentation](#)
- [TimeQuest Analyzer Training and Demonstrations](#)
- [Design Examples](#)

For a brief overview of the TimeQuest timing analyzer, refer to the TimeQuest Timing Analyzer section on the [Verification and Board Level](#) product feature page.

To search for known TimeQuest issues and technical support solutions, use Altera's [Knowledge Database](#). You can also visit the [Altera® Forum](#) to connect to and discuss technical issues with other Altera users.

For further technical support, use [mySupport](#) to create, view, and update service requests.

[www.alterawiki.com/wiki/TimeQuest\\_User\\_Guide](http://www.alterawiki.com/wiki/TimeQuest_User_Guide)

A TimeQuest User Guide to help users understand the details of SDC constraints and using TimeQuest for static timing analysis

**Thank You**