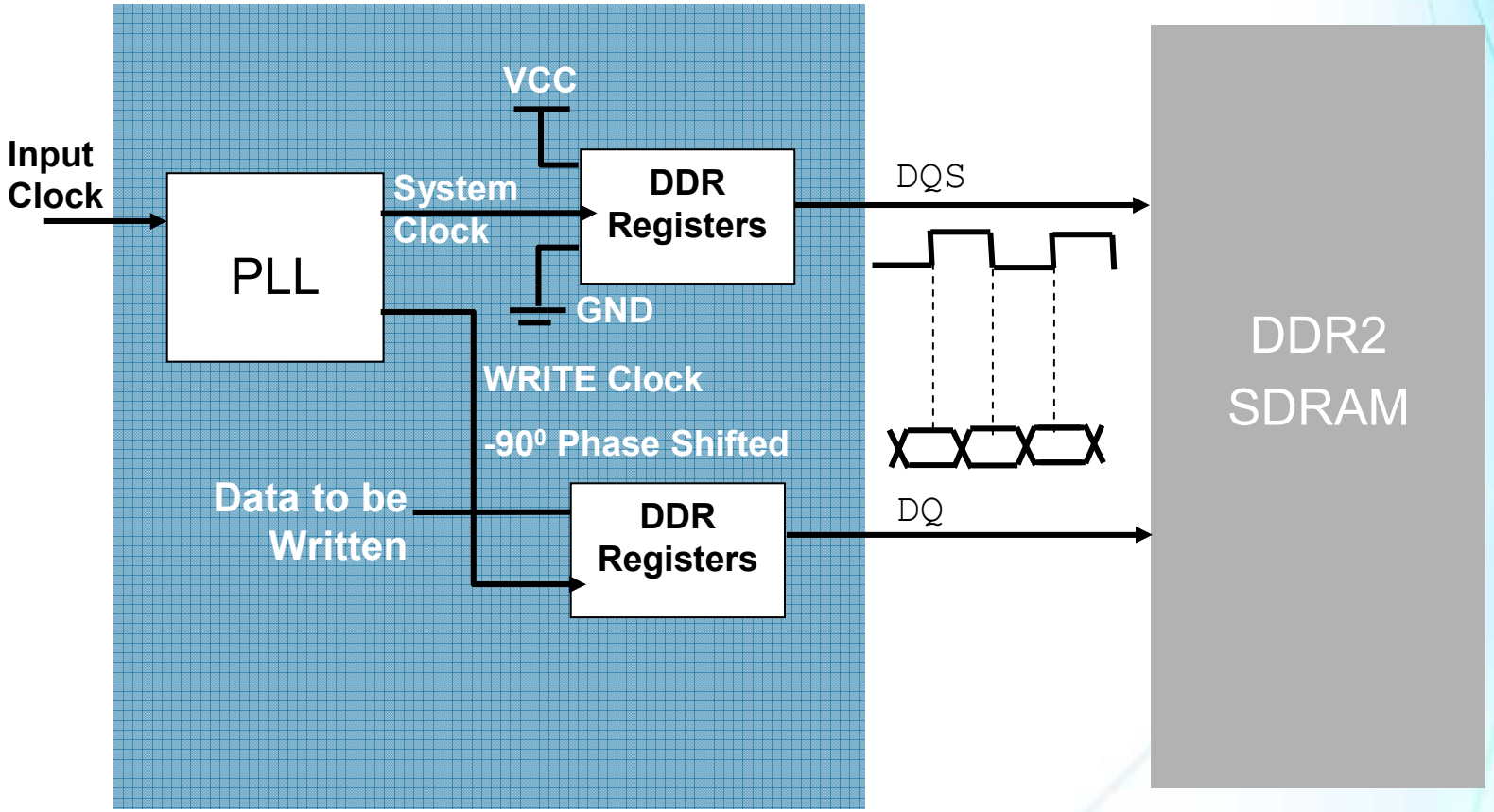
Constraining & Analyzing Source Synchronous Interface

Agenda

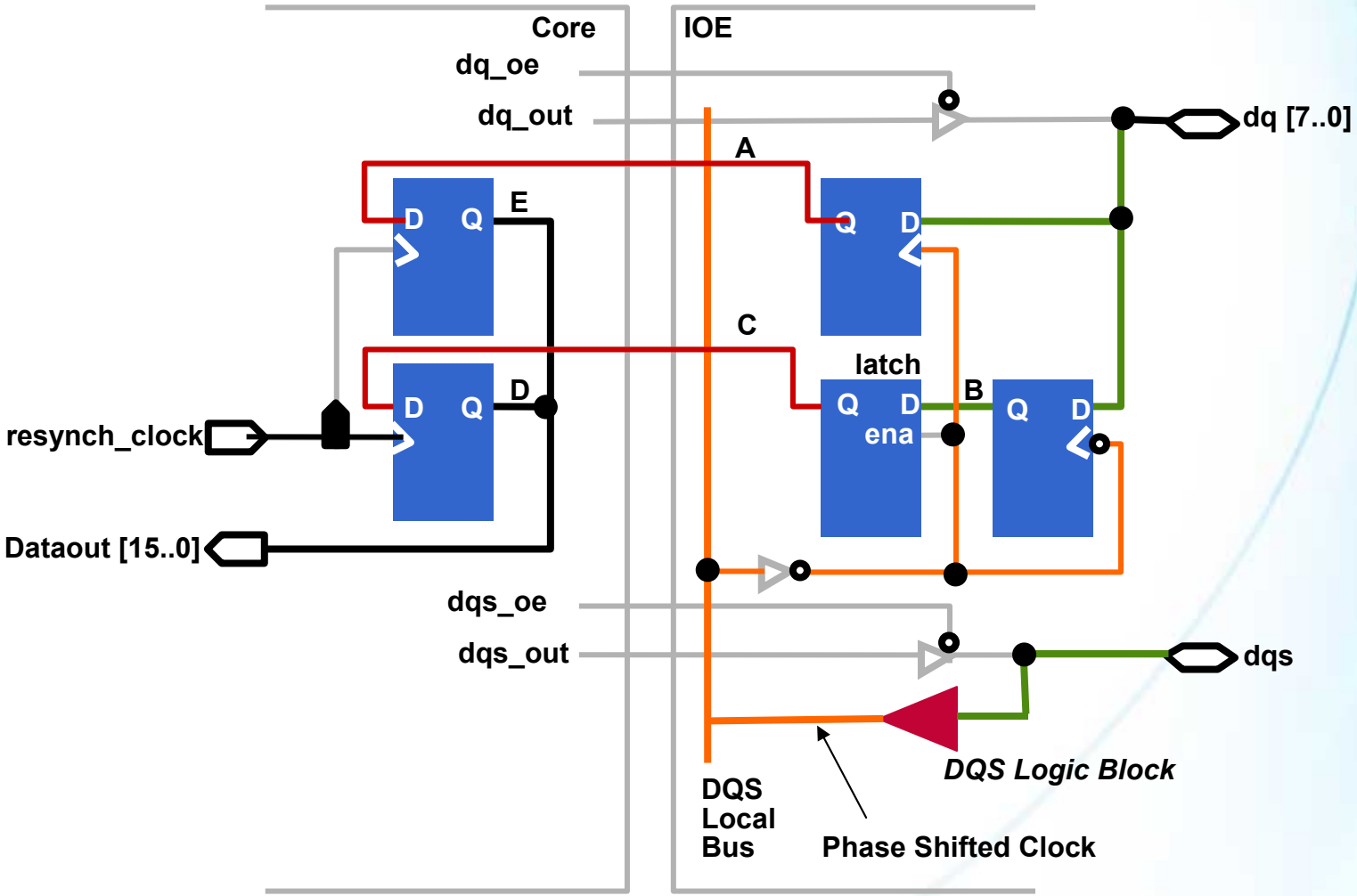
- Brief introduction to write/read structure
- Source Synchronous Introduction
- Write constraints
- Read constraints
- More constraints
- Timing Analysis

DDR2 SDRAM Write Interface

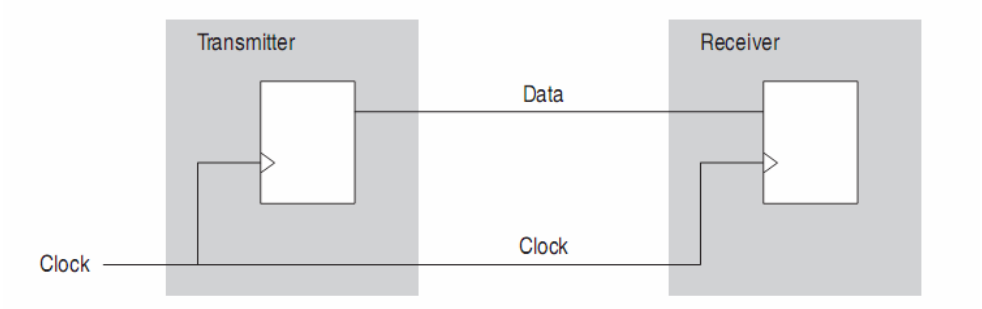


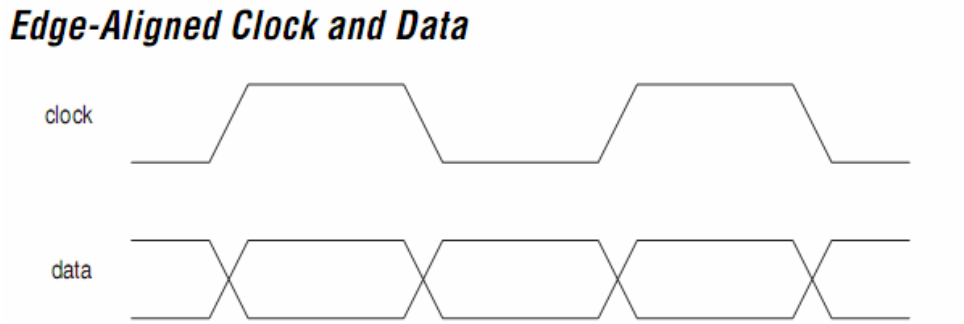
PLL Used to Generate the 90 Deg Phase Shift Between DQ and DQS Signals

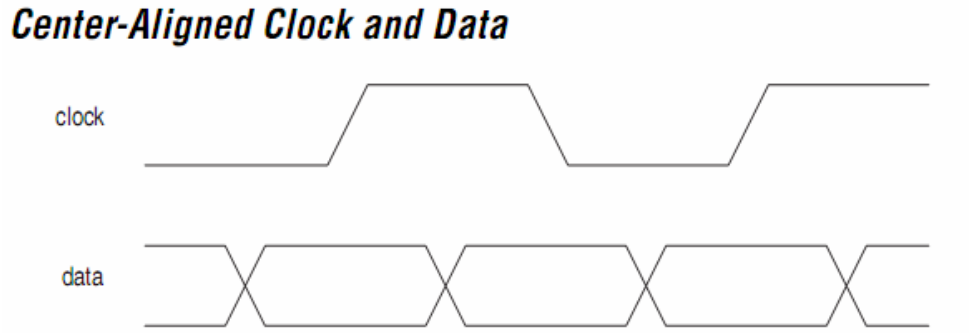
DDR2 SDRAM Read Interface



Basic Source Synchronous Interface







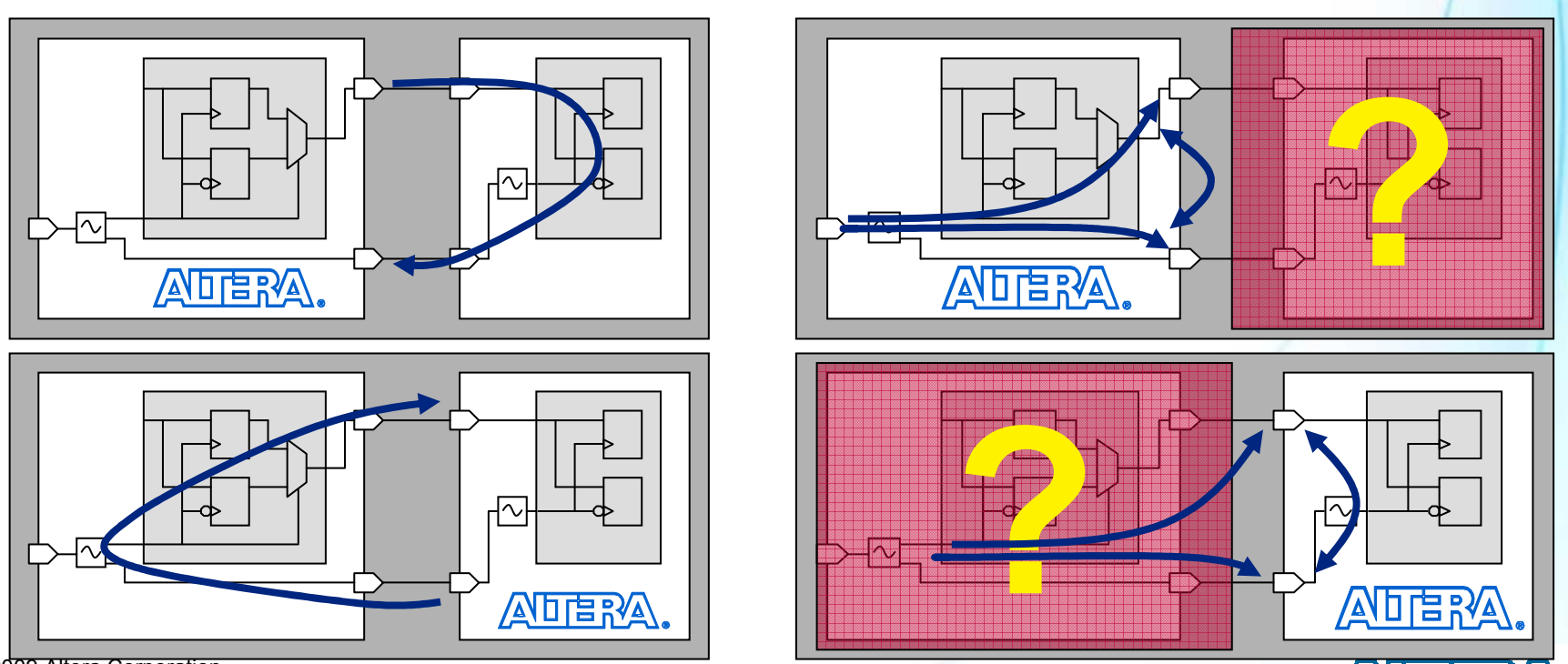
Source Synchronous interfaces are typically used for high speed data transfer.
i.e. DDR memory, HyperTransport bus and SPI

Interface constraints

- Clock constraints
  - Define the clocks used in the interface. Clock constraints define the period and other clock characteristics. i.e. offset and uncertainty
- Input or output delay constraints
  - Describe the required times for data to be valid at the interface. The input and output delay constraints are derived from timing parameters such as tSU, tH or tCO.
- Timing exceptions
  - Control launch and latch edges used in timing analysis. Timing exceptions ensure that only valid timing paths are analyzed.

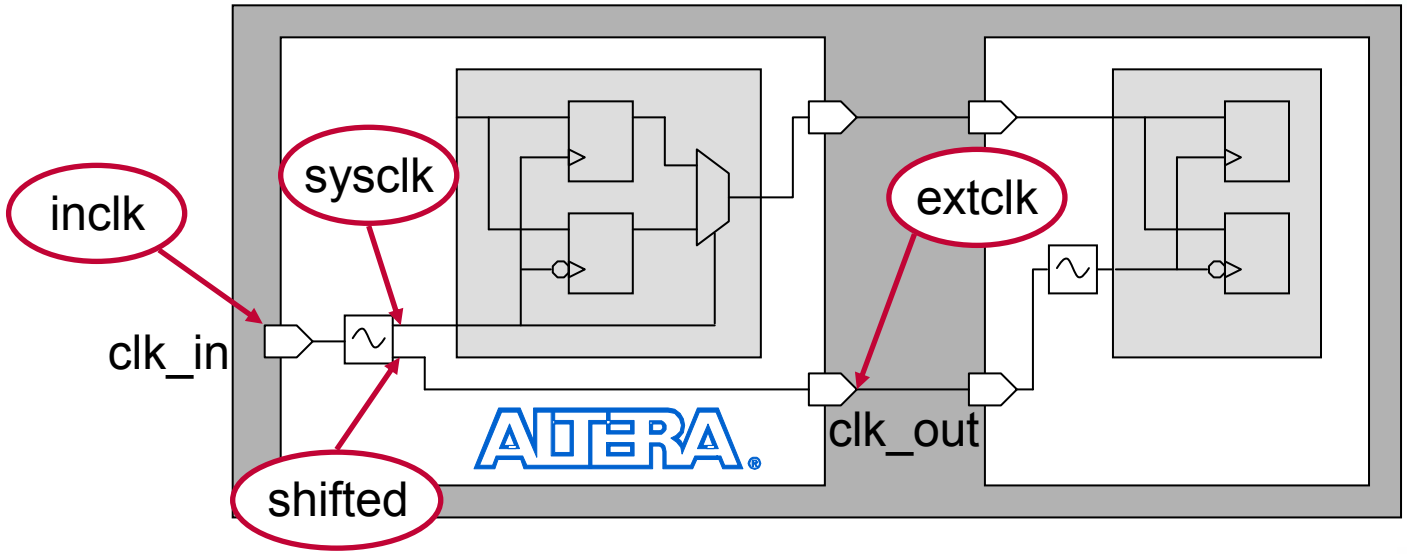
Input and Output Delay Constraints

- System-centric method
  - Trace delay, tSU, tH, tCO and tCOmin
- FPGA-centric method
  - skew



Write Constraints

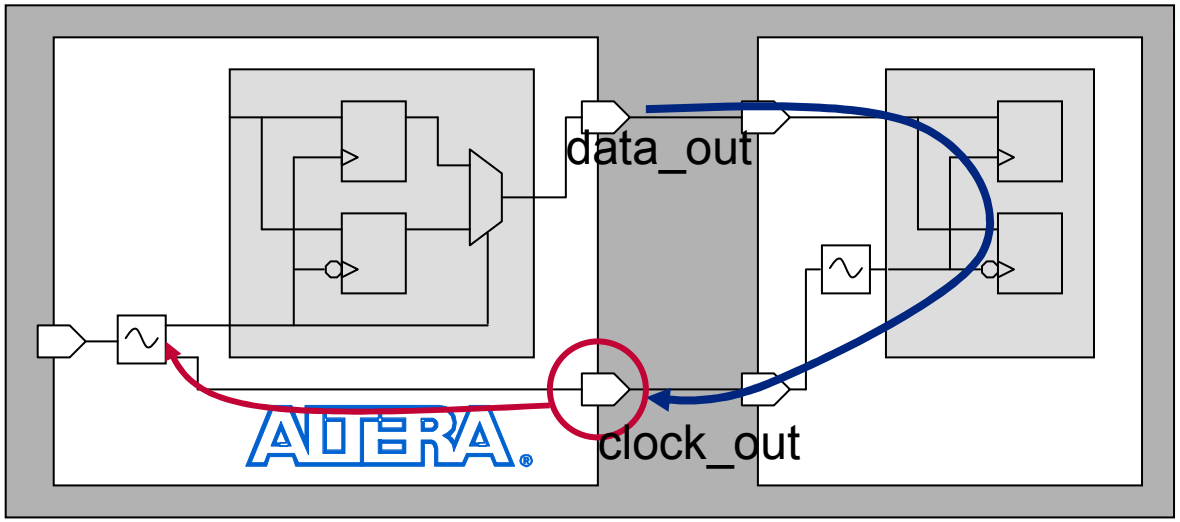
Output Clock Constraints



- Use same clocks for system-centric and FPGA-centric approaches
- Create generated clock on output clock port

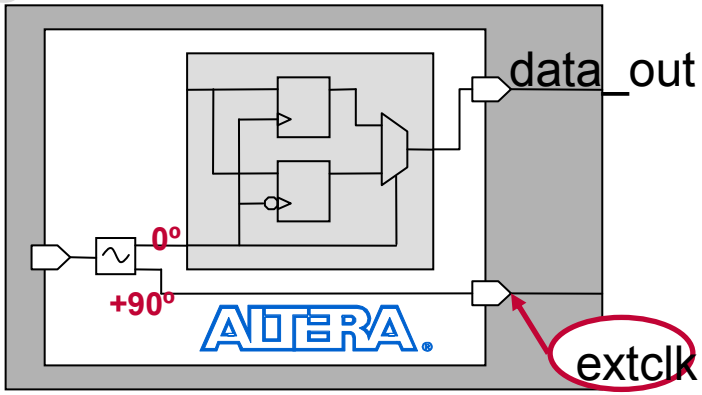
```
create_clock -name inclk -period 10 [get_ports clk_in]
create_generated_clock -name sysclk -source \
       [get_pins PLL|inclk[0]] [get_pins PLL|clk[0]]
create_generated_clock -name shifted -phase <shift> -source \
       [get_pins PLL|inclk[0]] [get_pins PLL|clk[1]]
create_generated_clock -name extclk -source \
       [get_pins PLL|clk[1]] [get_ports clk_out]
```

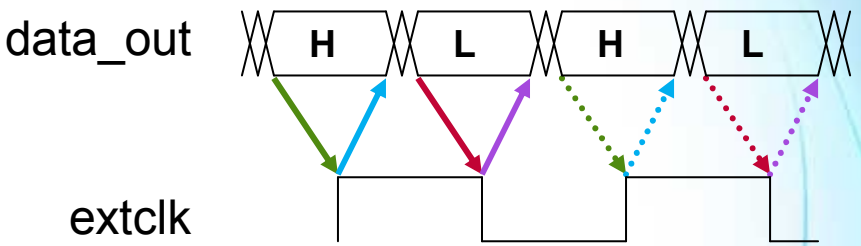
Output Delay Constraints-System Centric



- Create a generated clock on the output clock port
  - Accounts for delay to output clock port
- Specify output delays relative to the generated clock
  - Output max delay = max(board_data) + tSU(DDR) - min(board_clk)
  - Output min delay = min(board_data) - tH(DDR) - max(board_clk)
  - May need to compensate for default setup relationships (more on this later)
- Duplicate output delays to constrain data for falling clock edge
  - Add -clock_fall and -add_delay options

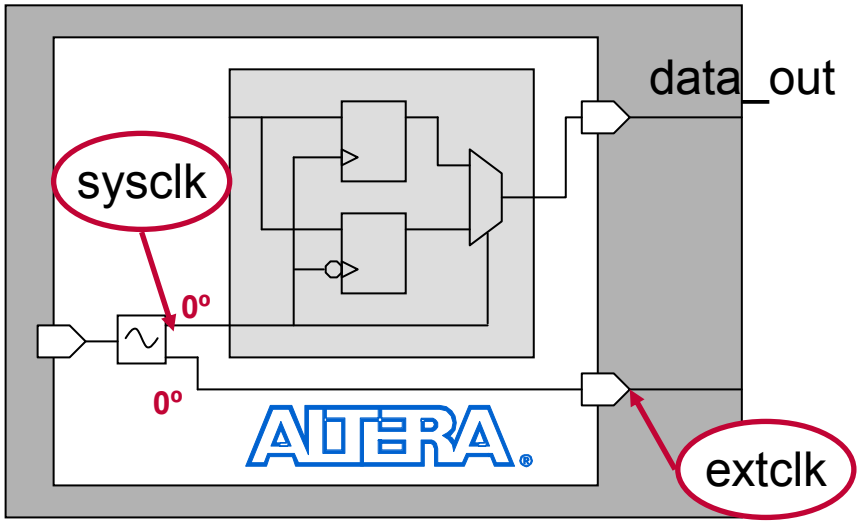
System Centric – Center Aligned

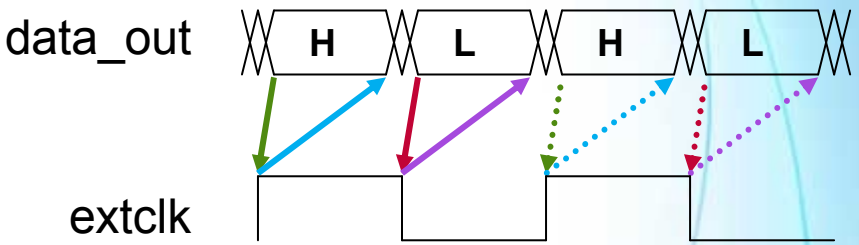





```
set_output_delay -max <output maximum delay> -clock \
        [get_clocks extclk] [get_ports data_out]

set_output_delay -min <output minimum delay value> -clock \
        [get_clocks extclk] [get_ports data_out]

set_output_delay -max <output maximum delay> -clock \
        [get_clocks extclk] -clock_fall [get_ports data_out] -add_delay

set_output_delay -min <output minimum delay value> -clock \
        [get_clocks extclk] -clock_fall [get_ports data_out] -add_delay
```

System Centric – Edge Aligned

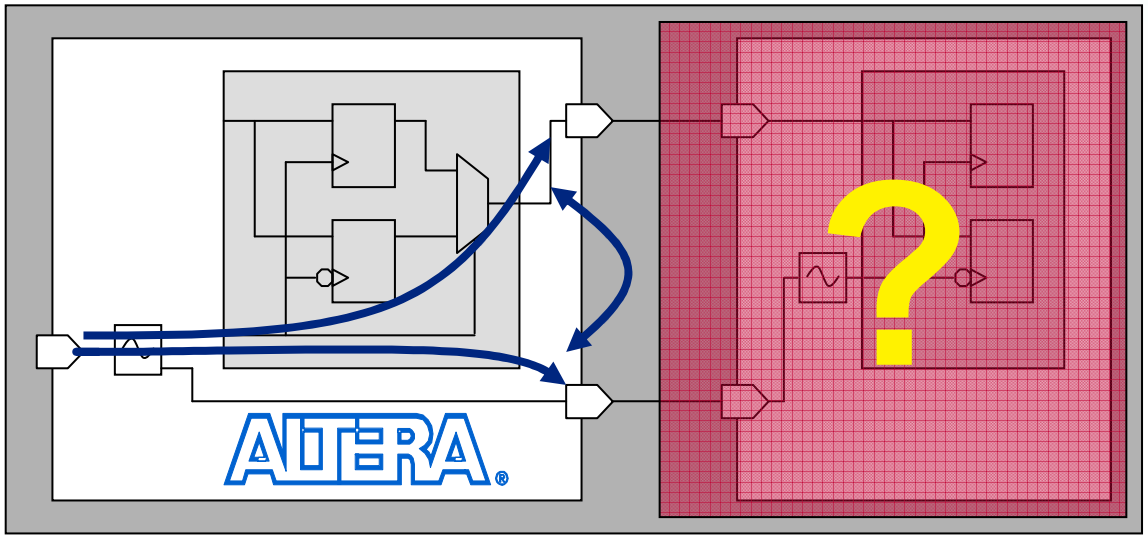




- Same output delay constraints as center-aligned case
- Requires timing exceptions
  - Setup check is to next clock edge, by default
  - Edge-aligned output requires same-edge capture

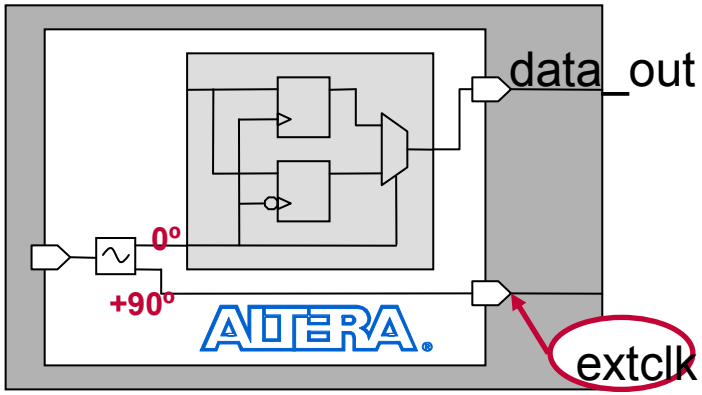
```
set_multicycle_path 0 -setup -end -rise_from \
      [get_clocks sysclk] -rise_to [get_clocks extclk]
set_multicycle_path 0 -setup -end -fall_from \
      [get_clocks sysclk] -fall_to [get_clocks extclk]
```

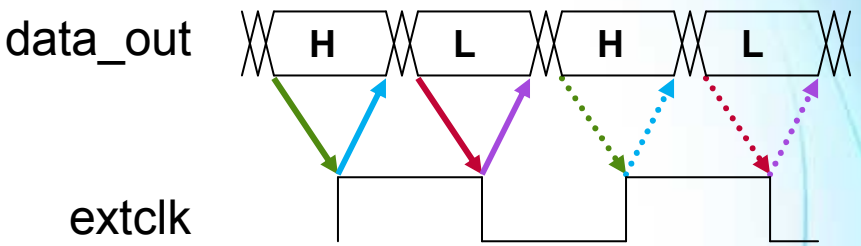
Output Delay Constraints – FPGA Centric



- Create a generated clock on the output clock port
  - Accounts for delay to output clock port
- Specify output delays relative to the generated clock
  - Output max delay = (latch – launch) - skew
  - Output min delay = (latch – launch) + skew
- You will need to adjust default setup/hold relationships
  - More on this later
- Duplicate output delays to constrain data for falling clock edge
  - Add -clock_fall and -add_delay options

FPGA Centric – Center Aligned

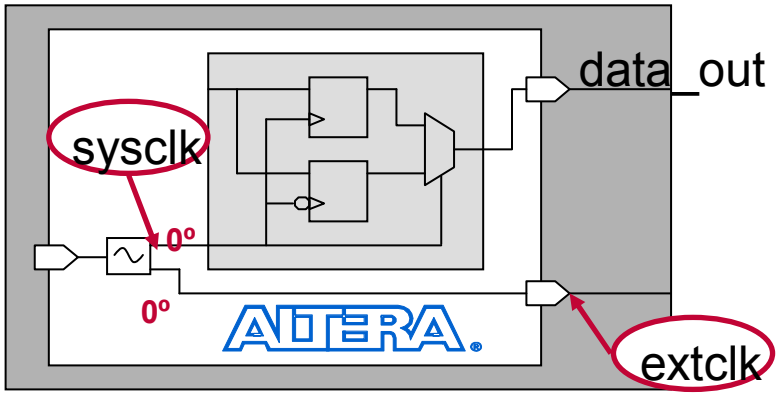


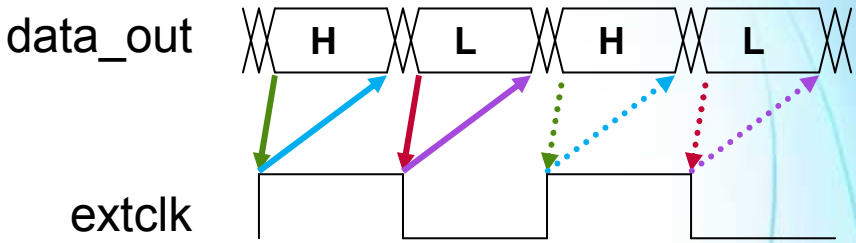


Skew = 200 ps period = 10 ns

```
set_output_delay -max [expr 2.5 - 0.2] -clock \
        [get_clocks extclk] [get_ports data_out]

set_output_delay -min [expr -2.5 + 0.2] -clock \
        [get_clocks extclk] [get_ports data_out]

set_output_delay -max [expr 2.5 – 0.2] -clock \
        [get_clocks extclk] -clock_fall [get_ports data_out] -add_delay

set_output_delay -min [expr -2.5 + 0.2] -clock \
        [get_clocks extclk] -clock_fall [get_ports data_out] -add_delay
```

FPGA Centric – Edge Aligned



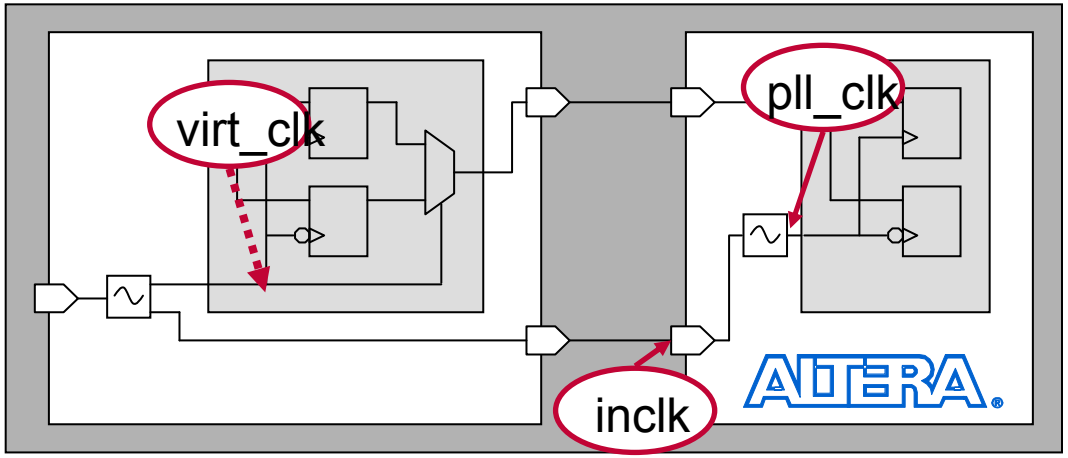


Skew = 200 ps period = 10 ns

```
set_output_delay -max [expr 0 - 0.2] -clock \
        [get_clocks extclk] [get_ports data_out]
set_output_delay -min [expr -5 + 0.2] -clock \
        [get_clocks extclk] [get_ports data_out]

set_output_delay -max [expr 0 – 0.2] -clock \
        [get_clocks extclk] -clock_fall [get_ports data_out] -add_delay
set_output_delay -min [expr -5 + 0.2] -clock \
        [get_clocks extclk] -clock_fall [get_ports data_out] -add_delay
```

Read Constraints

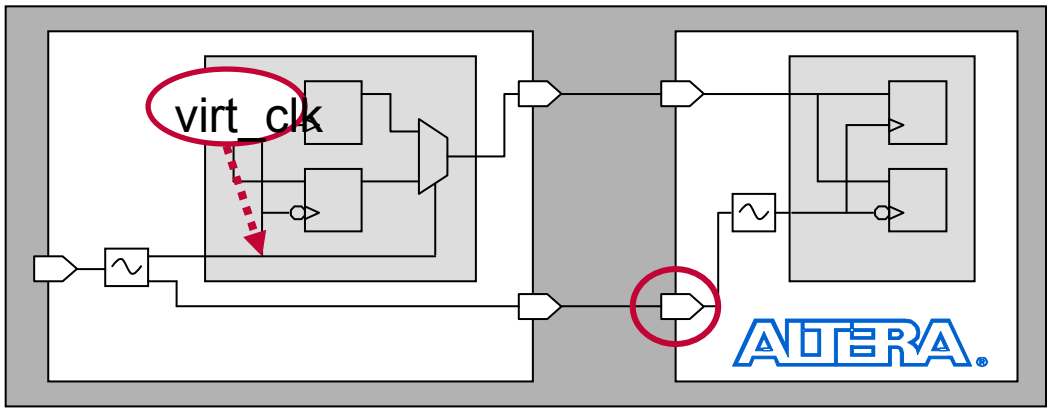
Input Clock constraints



- Use same clocks for system-centric and FPGA-centric approaches
- Virtual clock for represents external device
  - Use as clock for input delay constraints

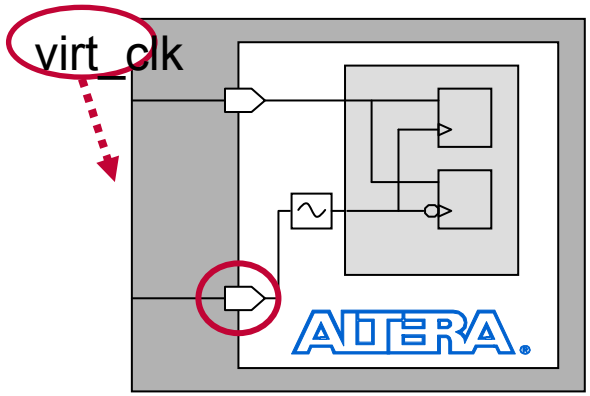
```
create_clock -name virt_clk -period 10
create_clock -name inclk -period 10 [get_ports clk_in] \
       -waveform { <rise time> <fall time> }
create_generated_clock -name pll_clk -phase <phase shift> \
       -source [get_pins PLL|inclk[0]] [get_pins PLL|clk[0]]
```

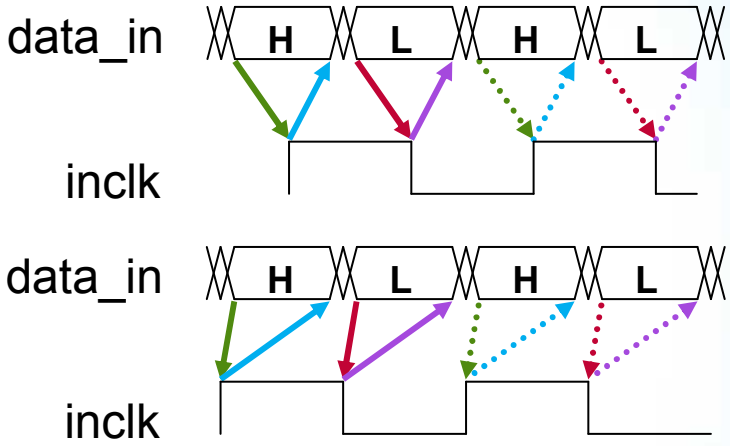
Input Delay Constraints – System Centric



- Create a virtual clock for input delay constraints
- Create a base clock on the input clock port
- Specify input delays relative to the virtual clock
  - Input max delay = max(board_data) + tCO - min(board_clk)
  - Input min delay = min(board_data) + tCOmin - max(board_clk)
- Duplicate input delays to constrain data for falling clock edge
  - Add -clock_fall and -add_delay options

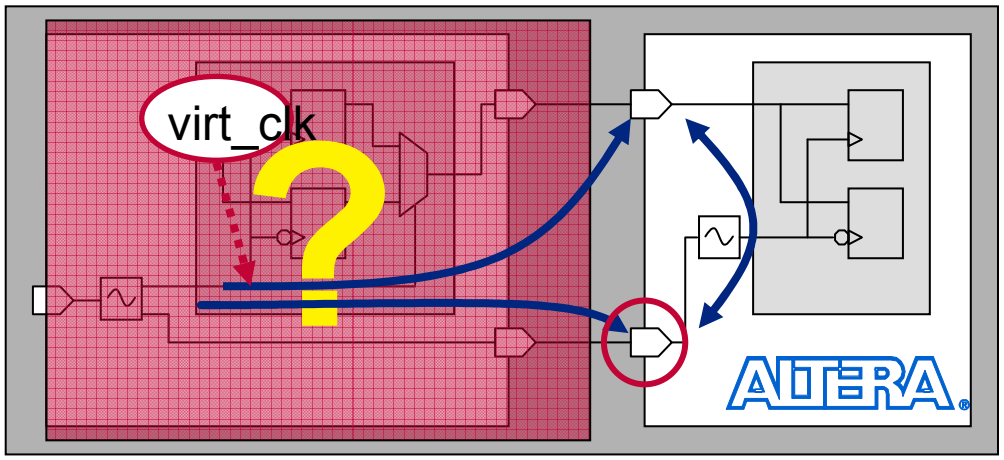
System Centric – Center or Edge Aligned





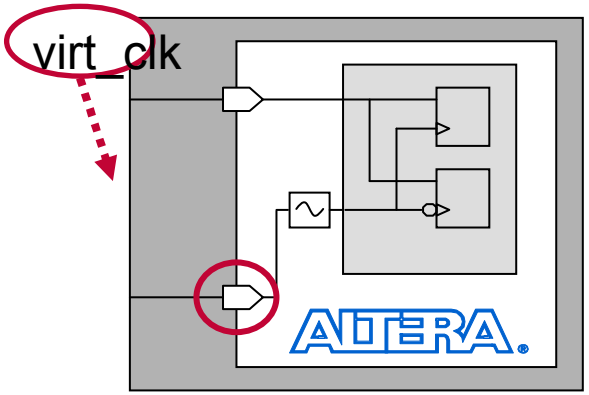

```
set_input_delay -clock [get_clocks virt_clk] -max <max> \
      [get_ports data_in*] -add_delay
set_input_delay -clock [get_clocks virt_clk] -min <min> \
      [get_ports data_in*] -add_delay
set_input_delay -clock [get_clocks virt_clk] -clock_fall -max \
      <max> [get_ports data_in*] -add_delay
set_input_delay -clock [get_clocks virt_clk] -clock_fall -min \
      <min> [get_ports data_in*] -add_delay
```

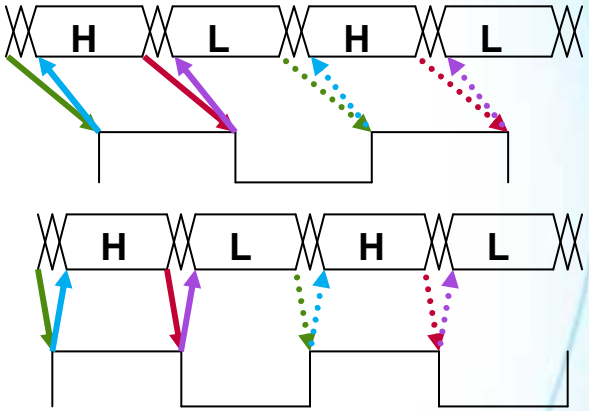
Input Delay Constraints - FPGA Centric



- Create a virtual clock for input delay constraints
- Specify input delays relative to the virtual clock
  - Input max delay = <skew>
  - Input min delay = -<skew>
- You may need to compensate for default setup/hold relationships
  - More on this later
- Duplicate input delays to constrain data for falling clock edge
  - Add -clock_fall and -add_delay options

FPGA Centric – Center or Edge Aligned






```
set_input_delay -clock [get_clocks virt_clk] -max <skew> \
      [get_ports data_in*] -add_delay
set_input_delay -clock [get_clocks virt_clk] -min -<skew> \
      [get_ports data_in*] -add_delay
set_input_delay -clock [get_clocks virt_clk] -clock_fall -max \
      <skew> [get_ports data_in*] -add_delay
set_input_delay -clock [get_clocks virt_clk] -clock_fall -min \
      -<skew> [get_ports data_in*] -add_delay
```

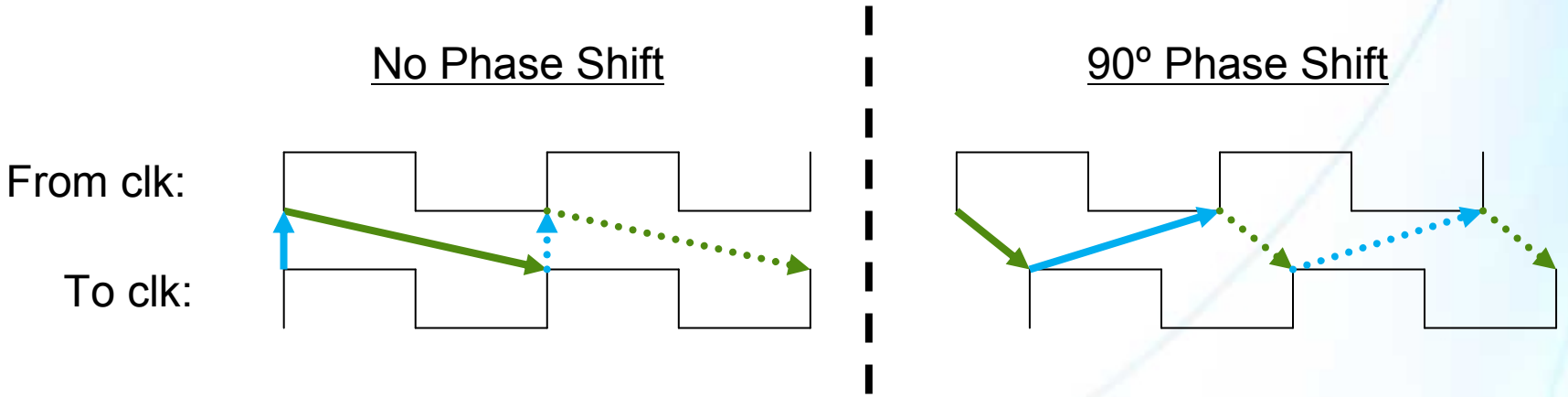
More Constraints

Double Data Rate Complexities

- Source synchronous interfaces typically launch and latch data on same clock edge
  - Can require adjustment for correct setup/hold analysis
- PLL usage
  - Center or edge align clock and data (phase shift)
  - Improve setup/hold margin (delay)
- Additional setup/hold analysis on falling clock edge
- Use existing solutions for DDR memory
  - High performance auto-calibrating ALTMEMPHY controller
  - Legacy data path and controller with DDR Timing Wizard (DTW)

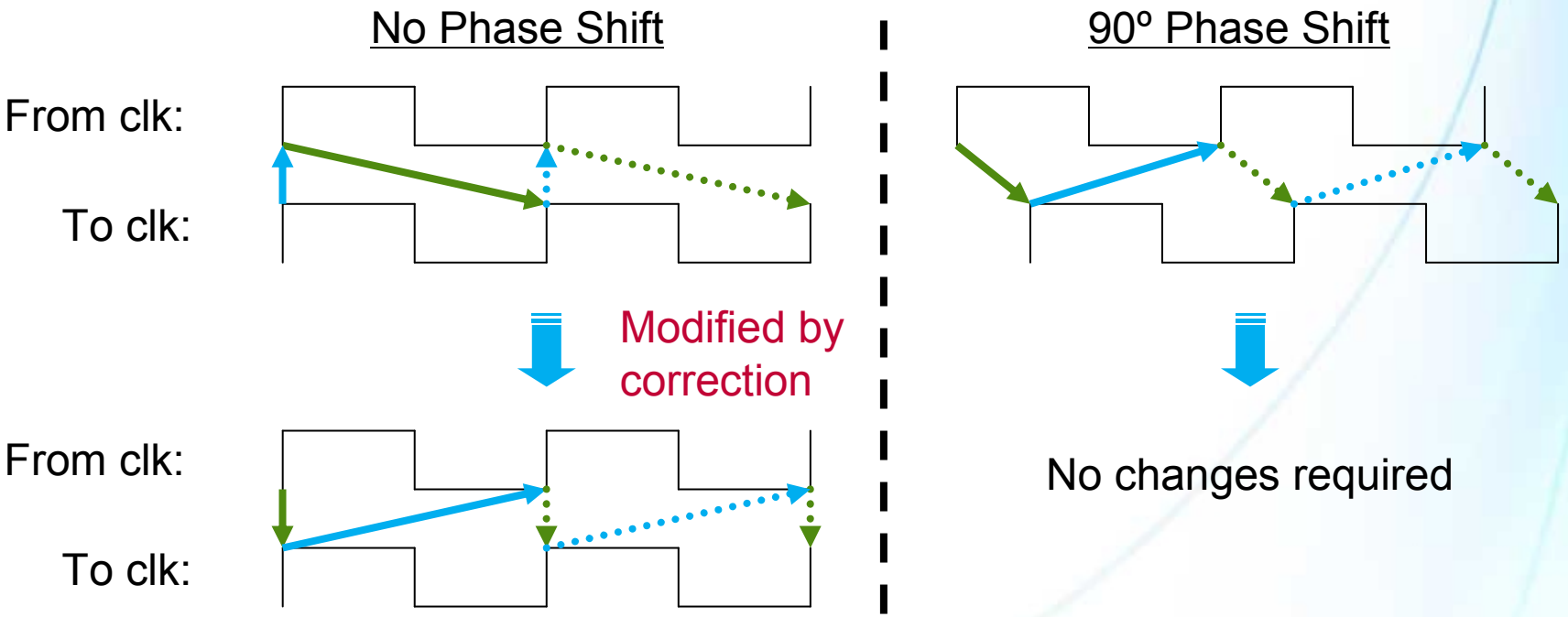
PLLs & Phase Shifts

- A PLL phase shift affects the setup/hold relationships (latch - launch)
  - Phase shift is not a delay
- What are the default setup/hold relationships:



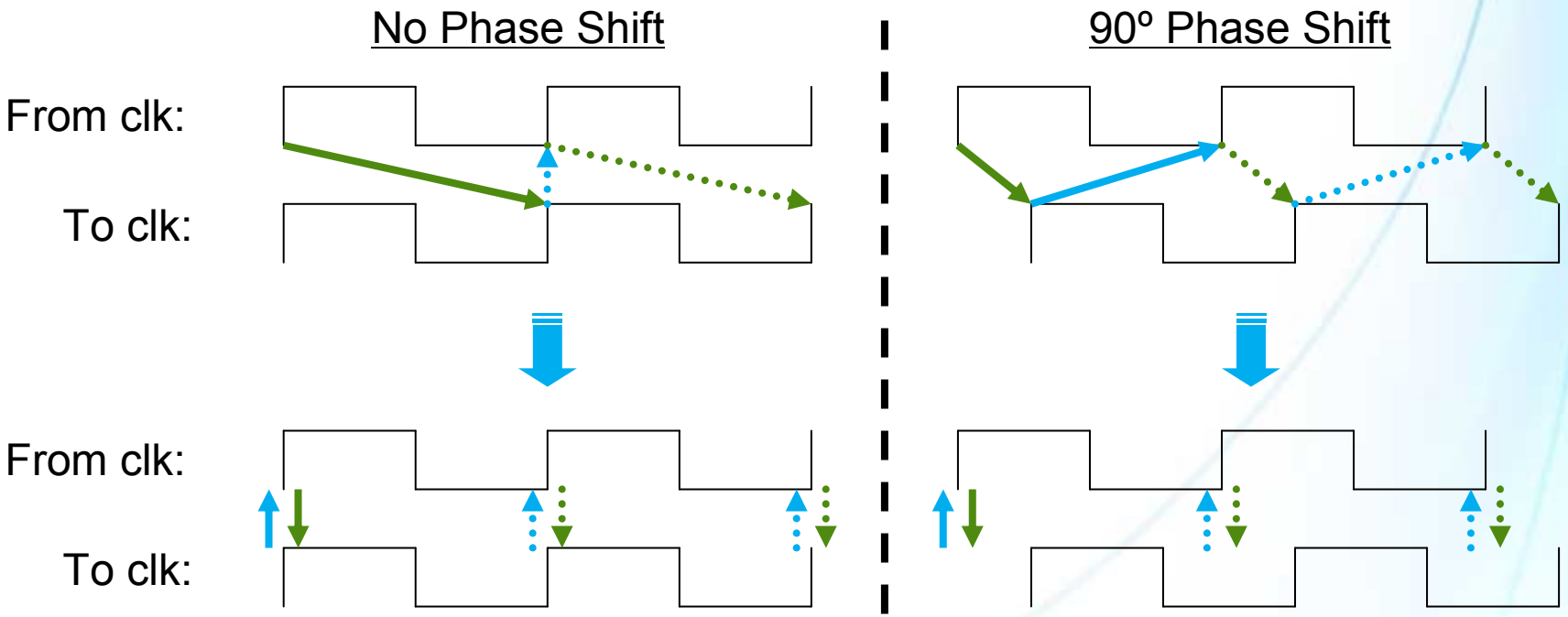
Correcting Setup/Hold (System)

- The same clock edge launches and latches data
- If external clock is phase shifted > 0º, no changes are required
- If external clock is phase shifted <= 0º, correct it
  - Use a set_multicycle_path exception with value 0
  - Or add one clock period (Tclk) to your output max & min delays



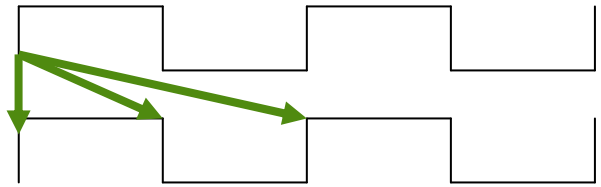
Correcting Setup/Hold (Skew)

- Move the “latch” edge to the “launch” edge for both setup & hold
- If external clock is not phase shifted
  - Add one clock period (Tclk) to your output max delay (and not the min delay)
- If external clock is phase shifted +90º
  - Use set_max_delay 0, set_min_delay 0
  - Or add (¼ Tclk) to the output max delay and (-3/4 Tclk) to the min delay
- When in doubt: add (latch - launch) to the output delay (max & min)



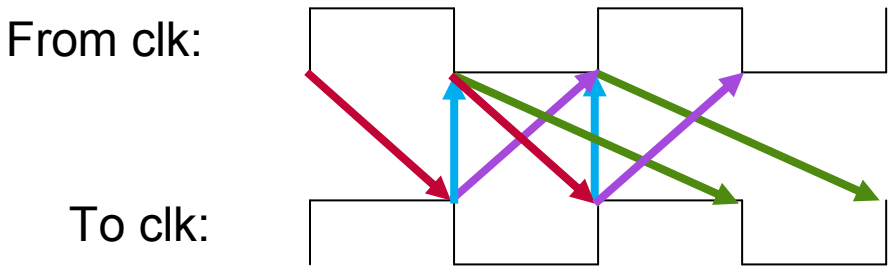
Correcting Setup/Hold – More Details

- What is the required operation of your interface?
  - Same edge launches/latches
  - Rise edge launches, fall edge latches
  - One cycle launches, another cycle latches



- Add any exceptions necessary to change the default analysis to match the desired analysis for the required operation of your interface
  - False paths
  - Multicycles

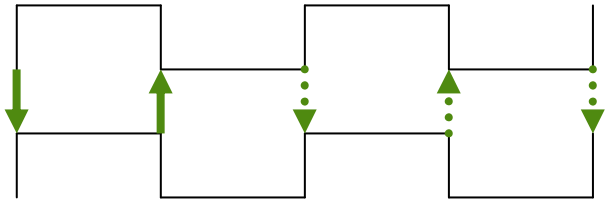
Default Analysis - Edge Aligned DDR Interface



From	To	Abbr.	Setup	Hold
Rise	Rise	RR	Period	0
Fall	Rise	FR	Period/2	-Period/2
Fall	Fall	FF	Period	0
Rise	Fall	RF	Period/2	-Period/2

Example

- Edge aligned interface
  - 0 degree phase shift
- Same edge launches and latches

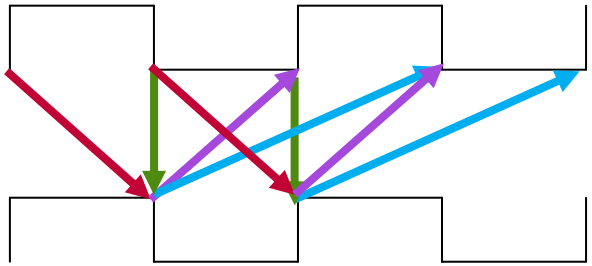


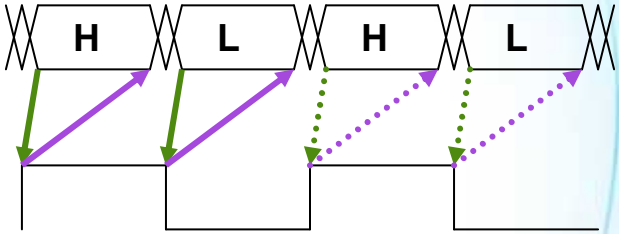
- Determine what setup/hold analysis should be
- Create false path and multicycle exceptions to ensure correct analysis

Example – Same Edge Launches/Latches

From clk:

To clk:





	Default		Desired	
Edges	Setup	Hold	Setup	Hold
RR	Period	0	0	False path
FR	Period/2	-Period/2	False path	-Period/2
FF	Period	0	0	False path
RF	Period/2	-Period/2	False path	-Period/2

Example – Constraints

```
set_multicycle_path -rise_from <from clk> -rise_to <to clk> \
      -setup 0
set_multicycle_path -fall_from <from clk> -fall_to <to clk> \
      -setup 0
set_false_path -rise_from <from clk> -rise_to <to clk> -hold
set_false_path -rise_from <from clk> -fall_to <to clk> -setup
set_false_path -fall_from <from clk> -fall_to <to clk> -hold
set_false_path -fall_from <from clk> -rise_to <to clk> -setup
```

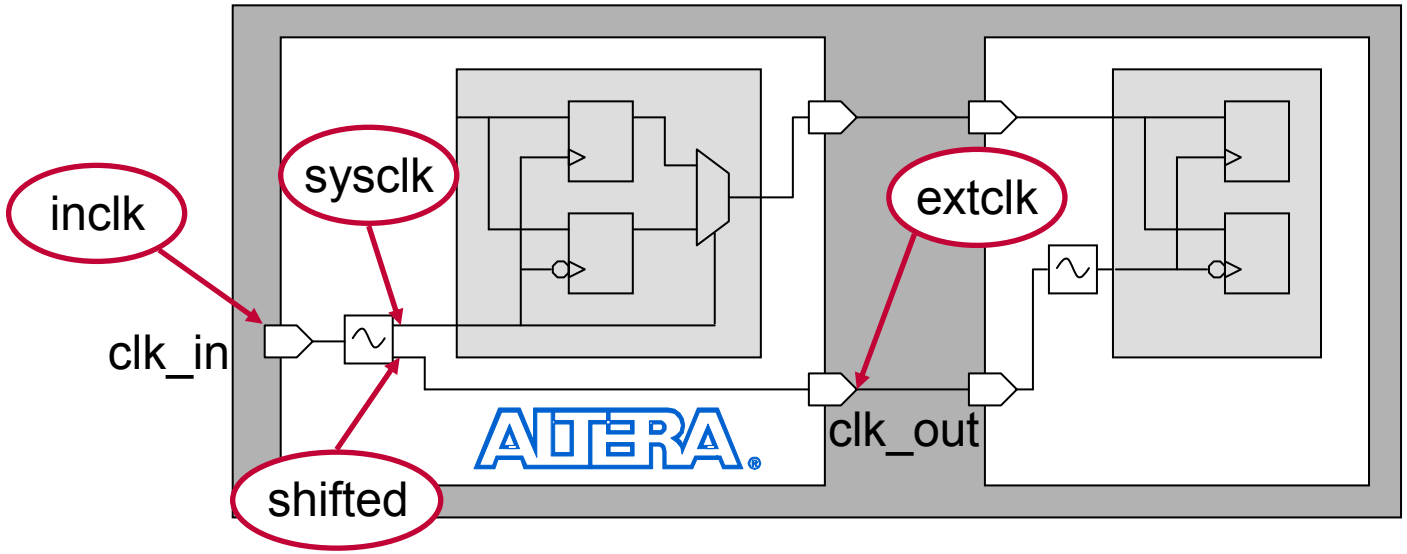
	Default		Desired	
Edges	Setup	Hold	Setup	Hold
RR	Period	0	0	False path
FR	Period/2	-Period/2	False path	-Period/2
FF	Period	0	0	False path
RF	Period/2	-Period/2	False path	-Period/2

Timing Analysis

Timing Analysis

- Use report_timing command to verify constraints are met
  - Report setup and hold timing
- Perform timing analysis at slow and fast corners
- Balance slack values to ensure largest margin

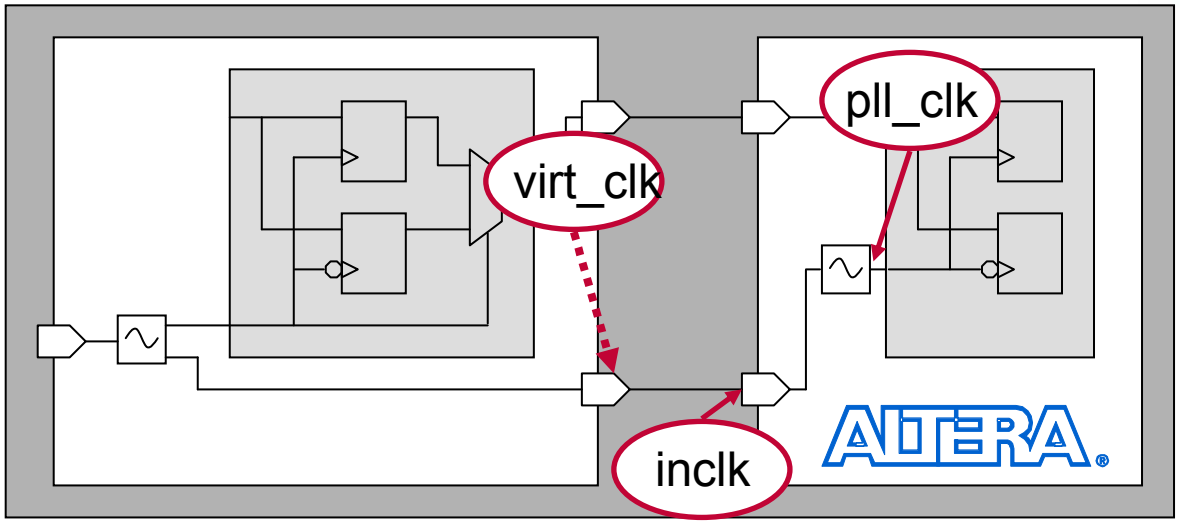
Output Timing Reporting



- Choose clock transfers to report
  - From clock driving output registers (sysclk)
  - To generated clock on output clock port (extclk)

```
report_timing -from_clock [get_clocks sysclk] -to_clock \
       [get_clocks extclk] -setup
report_timing -from_clock [get_clocks sysclk] -to_clock \
       [get_clocks extclk] -hold
```

Input Timing Reporting



- Choose clock transfers to report
  - From virtual clock (virt_clk)
  - To clock driving input clock port (inclk)

```
report_timing -from_clock [get_clocks virt_clk] -to_clock \
       [get_clocks pll_clk] -setup
report_timing -from_clock [get_clocks virt_clk] -to_clock \
       [get_clocks pll_clk] -hold
```