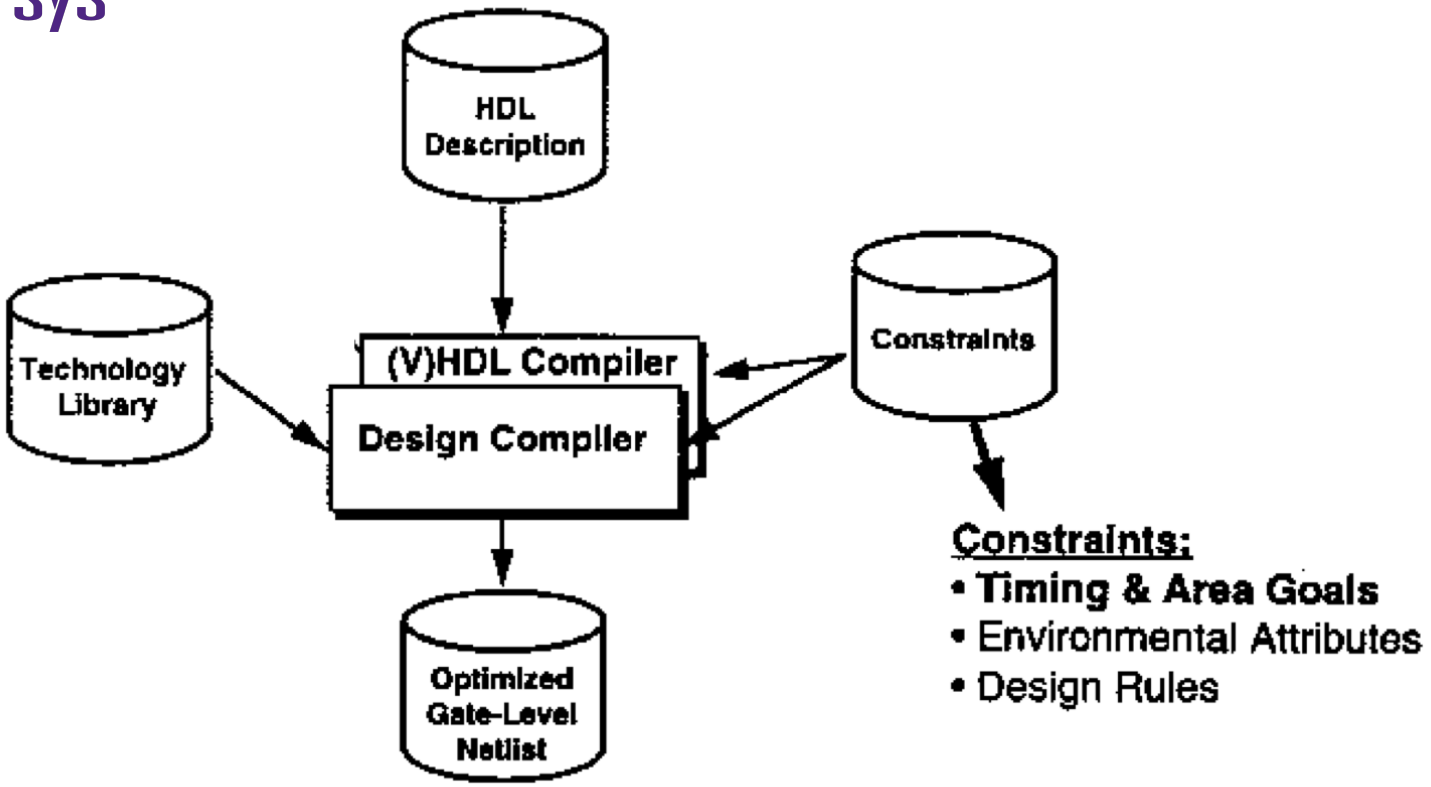
GONZAGA UNIVERSITY

Synthesis: Timing Constraints

GONZAGA UNIVERSITY

Synthesizing a Design

SYNOPSYS®



By properly constraining your design, you guide the synthesis tool towards achieving your design goals.

GONZAGA UNIVERSITY

Synthesizing a Design

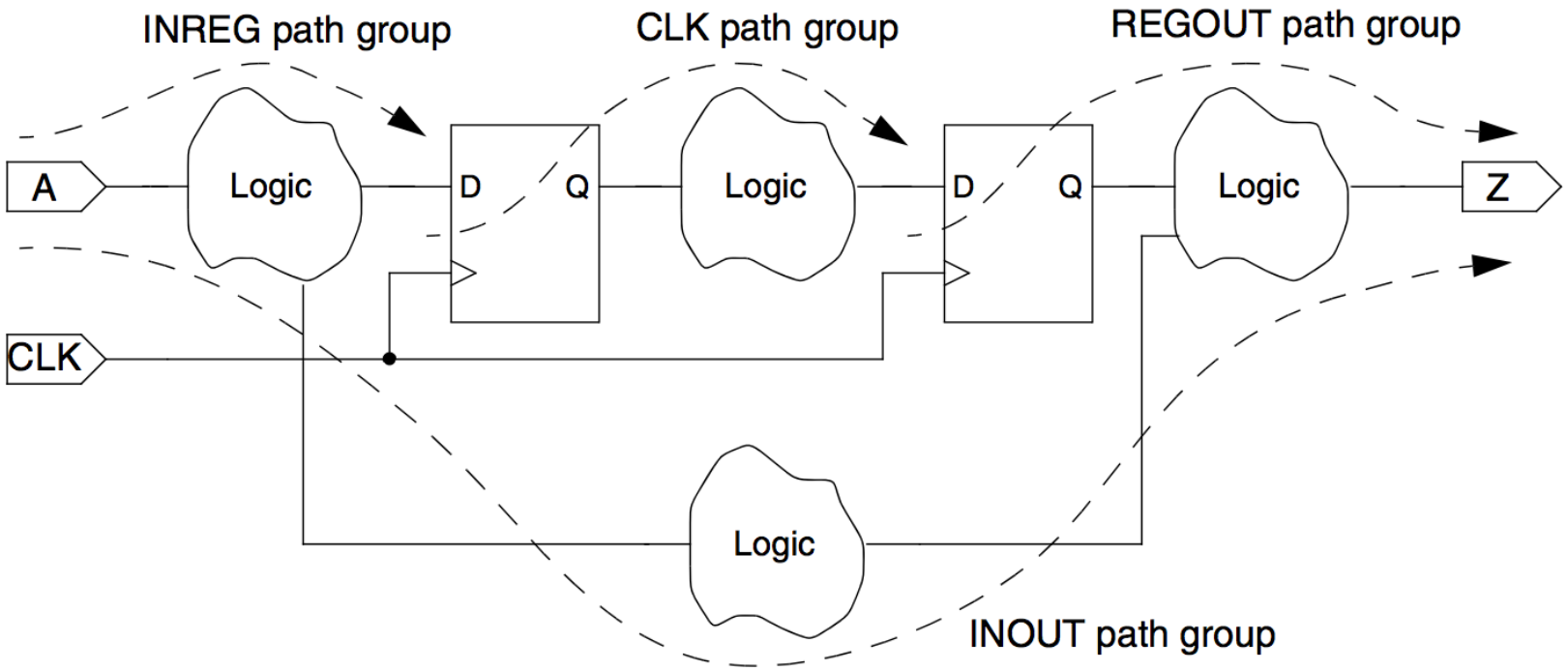
- Recommended readings for in depth understanding of how to constrain and synthesize a design:
  - Timing Constraints and Optimization User Guide (by Synopsys)
  - Using the Synopsys Design Constraints Format Application Note (by Synopsys)
  - SDC and TimeQuest API Reference Manual (by Altera)

Static Timing Analysis

- Static timing analysis is a method of validating the timing performance of a design by checking all possible paths for timing violations.
- It considers the "worst" possible delay through each logic element, but not the logical operation of the circuit
- In comparison to circuit simulation, static timing analysis is faster and more thorough.
- However, static timing analysis checks the design only for proper timing, not for correct logical functionality.

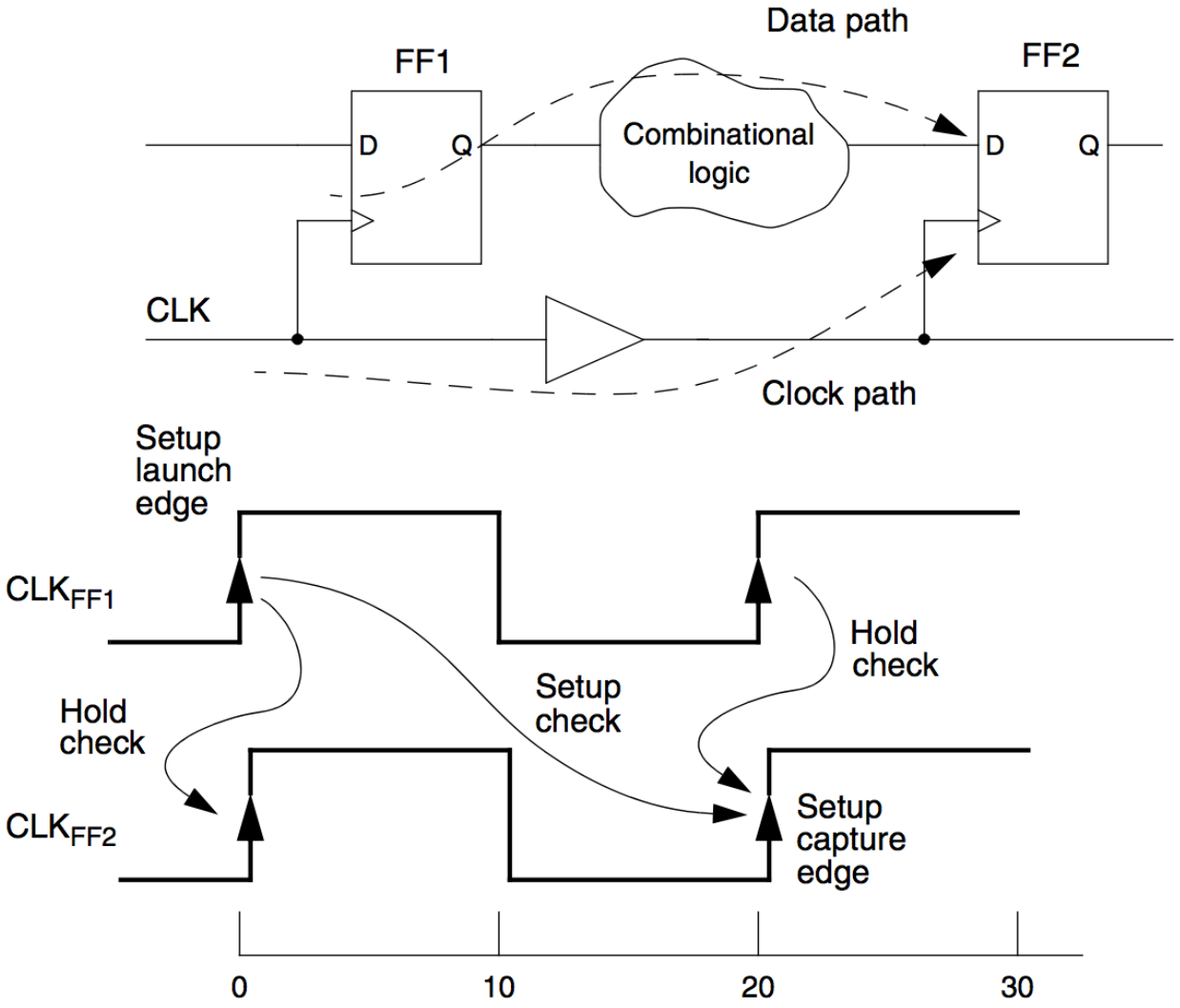
Timing Paths

Figure 1-19 Timing Path Groups



GONZAGA UNIVERSITY

Flip-Flop Timing Checks

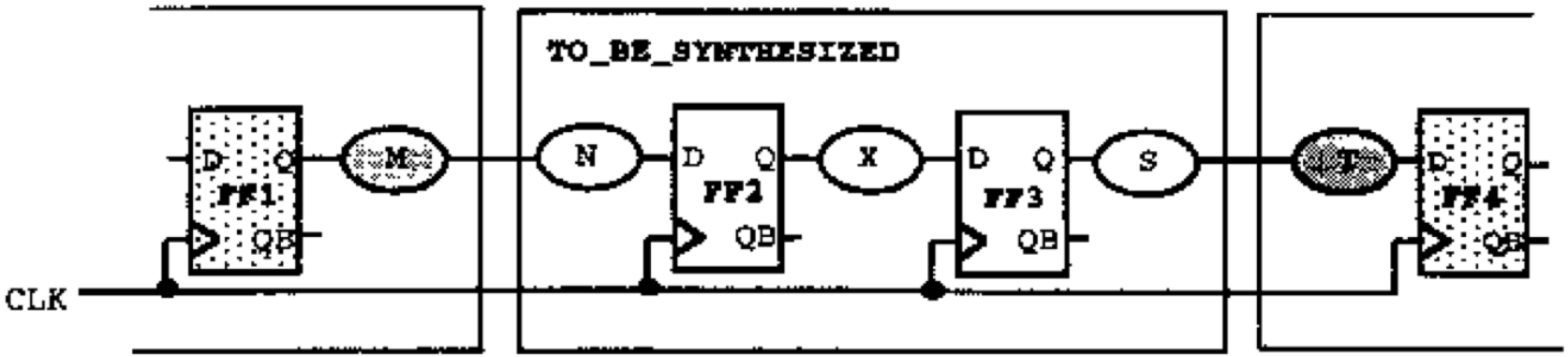


Defining Timing goals for Synchronous Designs

GONZAGA UNIVERSITY

Synchronous Designs:

- Data arrives from a clocked device.
- Data goes to a clocked device.

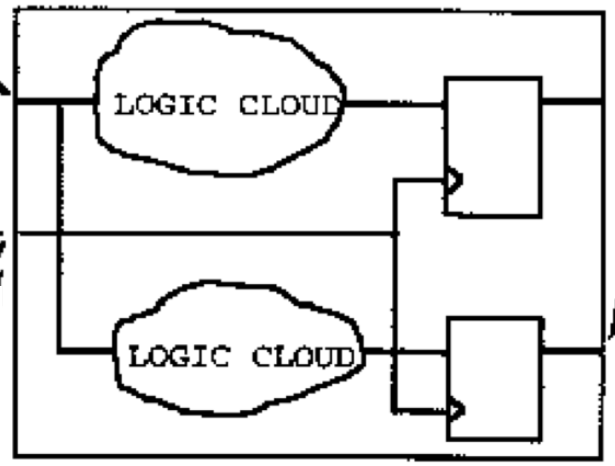


Methodology:

1. **Define the clocks.**
2. Define the I/O timing *relative* to the clocks.

Specifying Timing Goals

Specify *timing goals* by defining the clock and other related timing requirements of the design.



set_input_delay

- Defines input arrival times (relative to a clock)

create_clock

- Defines period and waveform of clock source

set_clock_skew

- Defines clock skew of clock source

set_output_delay

- Defines output timing requirements (relative to a clock)

Exceptions to Single Cycle Behavior:
set_multicycle_path
set_false_path

GONZAGA UNIVERSITY

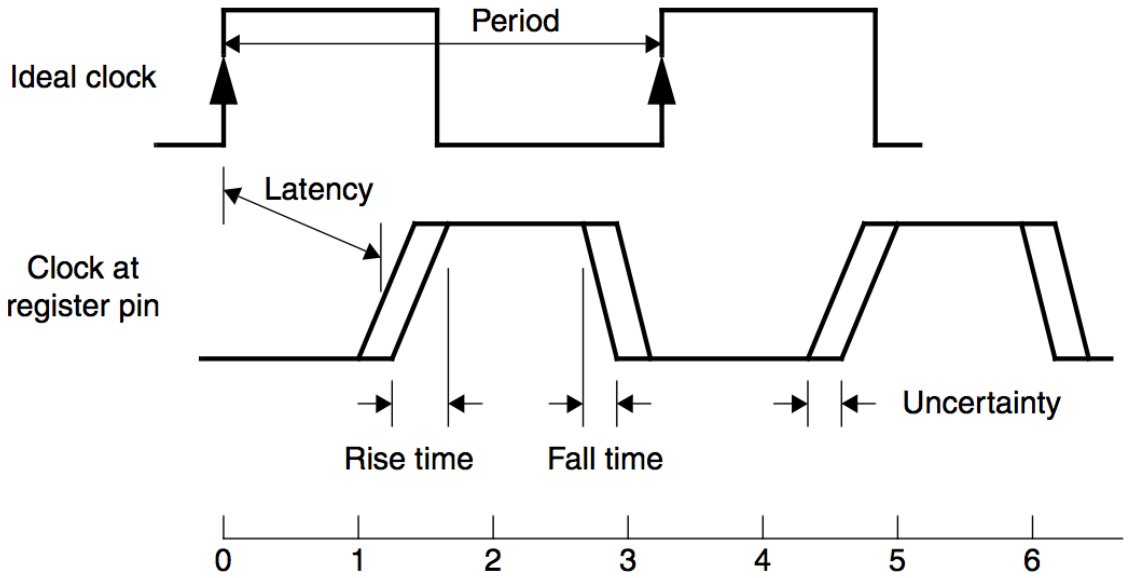
Defining Clocks

- Define:
  - Clock Source (port or pin)
  - Period
  - Duty Cycle
  - Clock Skew (uncertainty)
  - Clock Latency (propagation through clock tree)
  - Transition Time (rise/fall time)

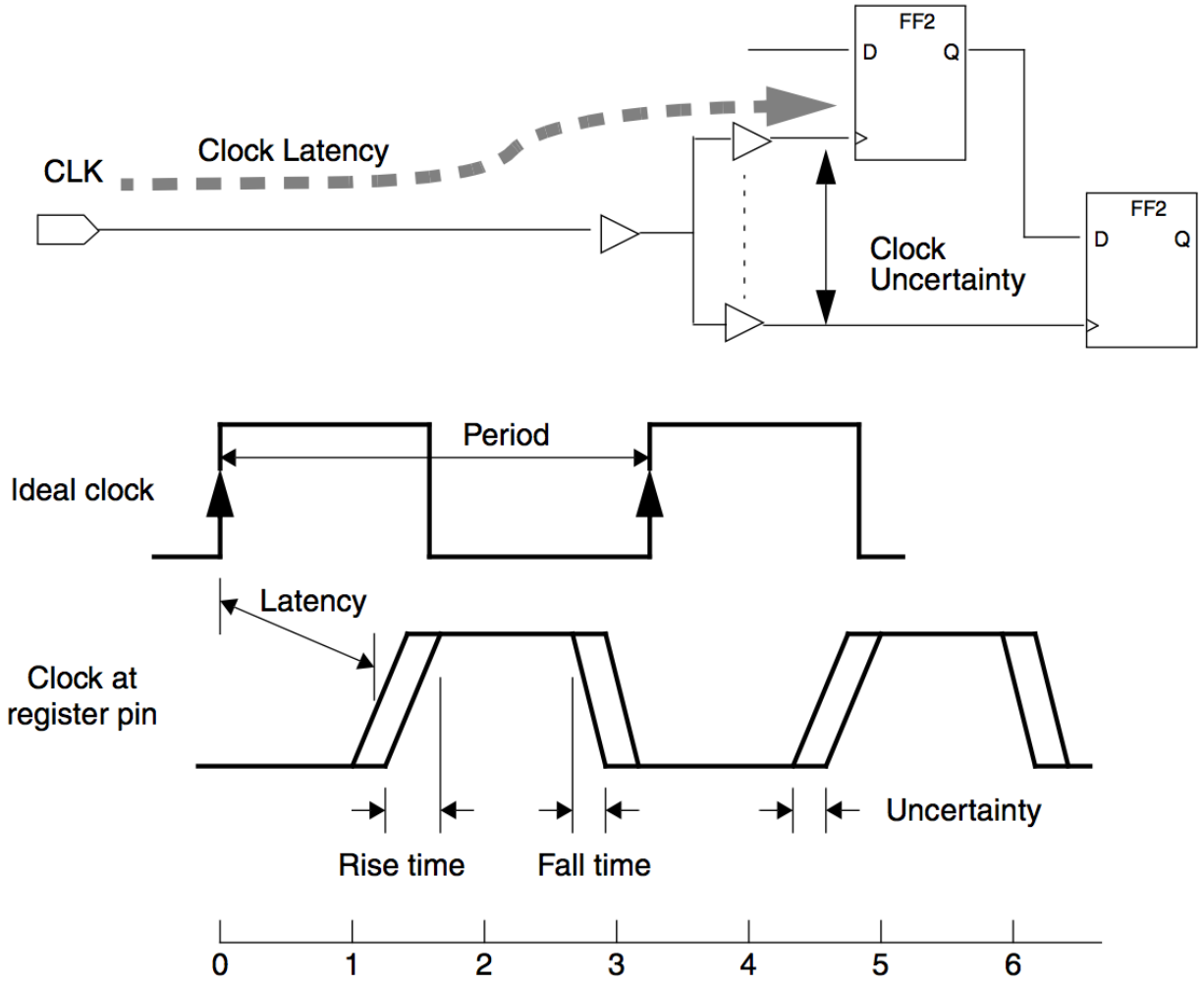
GONZAGA UNIVERSITY

Defining Clocks

- The command for specifying clocks is `create_clock`
- If multiple clocks are used in the design, you must specify the timing relationships between those clocks so that the analysis tool can check the timing of a path launched by one clock and captured by another



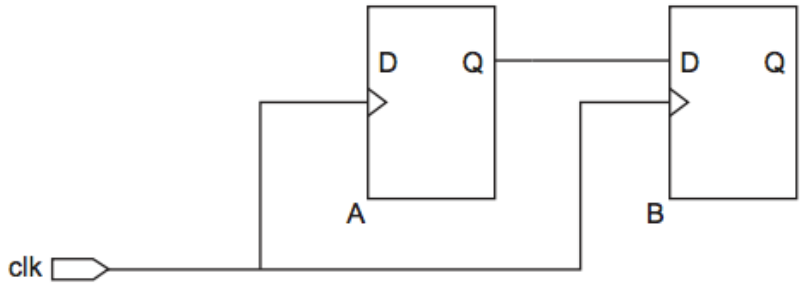
Clock Network Effects

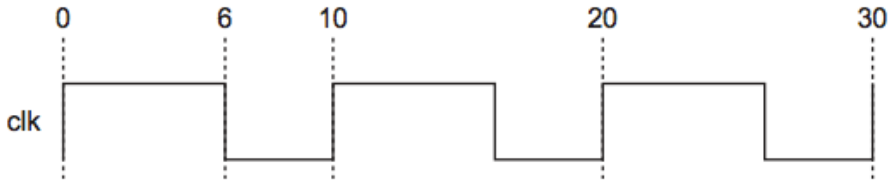


Example: `create_clock`

Example 1–1. Non-50/50 Duty Cycle Clock Constraints

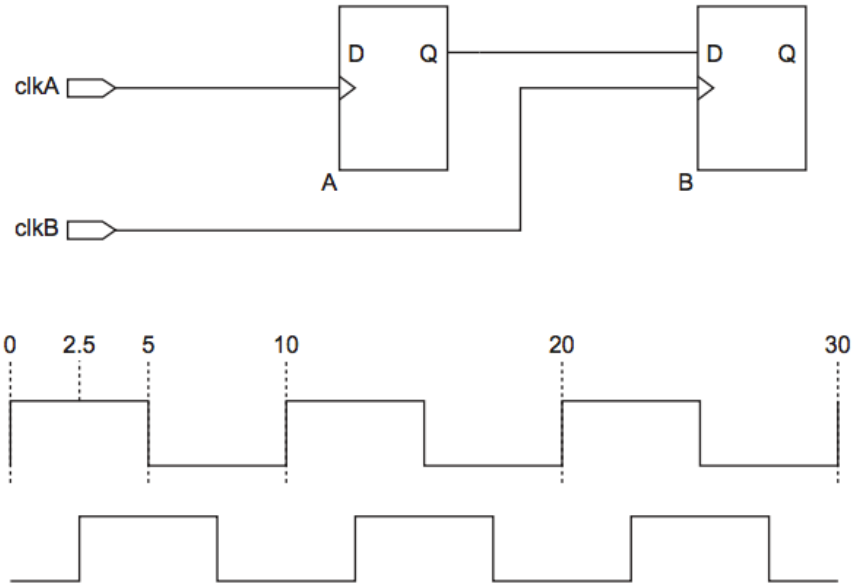
```
#60/40 duty cycle clock
create_clock \
    -period 10.000 \
    -waveform {0.000 6.000} \
    -name clk6040 [get_ports {clk}]
```





GONZAGA UNIVERSITY

Example: offset clocks



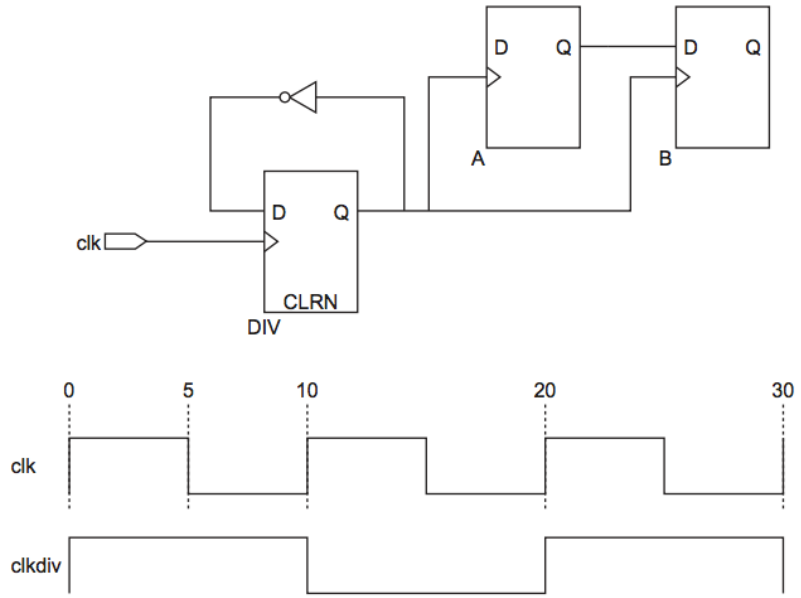
Example 1–2. Offset Clock Constraints

```
# -waveform defaults to 50/50 duty cycle
create_clock -period 10.000 \
    -name clkA \
    [get_ports {clkA}]

#create a clock with a 2.5 ns offset
create_clock -period 10.000 \
    -waveform {2.500 7.500} \
    -name clkB [get_ports {clkB}]
```

Example: `create_generated_clock`

Figure 1–3. Divide-by-Two Derived Clock



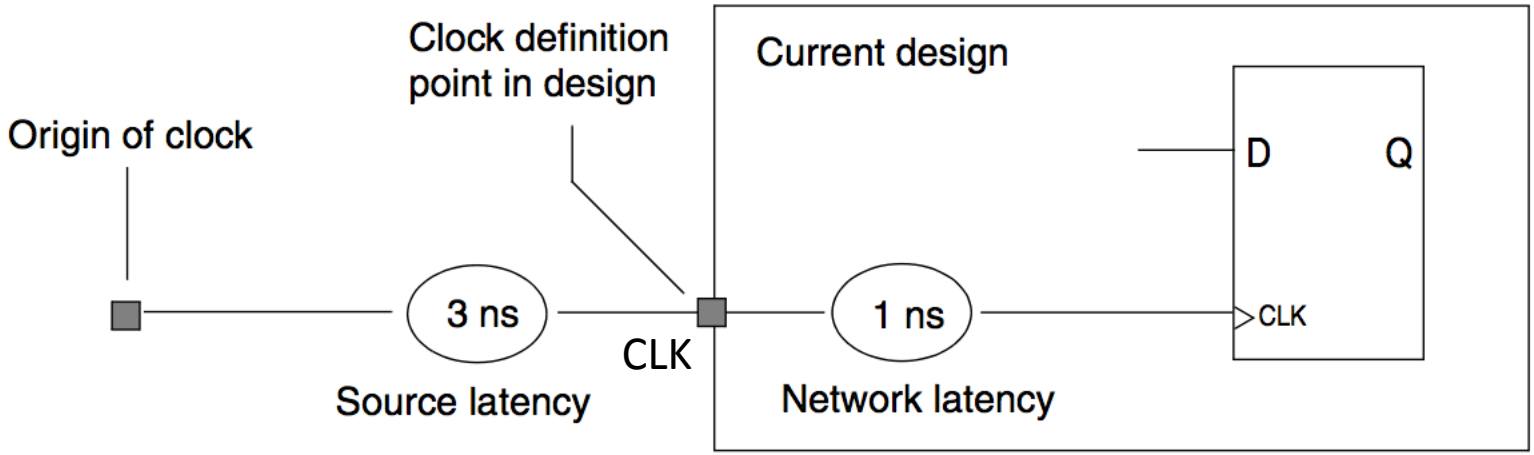
Example 1–3. Divide-by with -waveform Clock Constraints

```
create_clock -period 10.000 -name clk [get_ports {clk}]

# Using -divide_by option
create_generated_clock \
    -divide_by 2 \
    -source [get_ports {clk}] \
    -name clkdiv \
    [get_pins {DIV|q}]
```

Clock Latency

Figure 2-3 Source and Network Latency



NOTE:
Altera's TimeQuest analyzer automatically computes network latencies; therefore, you only can characterize source latency with the set_clock_latency command.

```
create_clock –name SYSCLK –period 10.0 [get_ports CLK]
set_clock_latency –source 3.000 [get_clocks SYSCLK]
```

Example: set_clock_latency

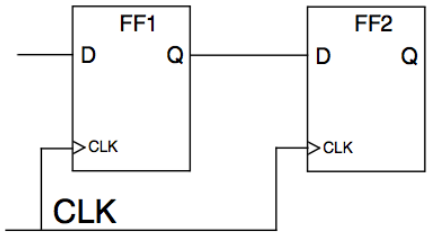
- Specifies additional delay (that is, latency) in a clock network. This delay value represents the external delay from a virtual (or ideal) clock through the longest (-late) or shortest (-early) path, with reference to the rising (-rise) or falling (-fall) clock transition. For setup analysis, the timing analyzer uses the late clock latency for the data arrival path and the early clock latency for the clock arrival path. For hold analysis, the timing analyzer uses the early clock latency for the data arrival time and the late clock latency for the clock arrival time.

```
create_clock –name SYSCLK –period 10.0 [get_ports CLK]
set_clock_latency -source -late -rise 0.800 [get_clocks SYSCLK]
set_clock_latency -source -late -fall 0.750 [get_clocks SYSCLK]
set_clock_latency -source -early -rise 0.500 [get_clocks SYSCLK]
set_clock_latency -source -early -fall 0.460 [get_clocks SYSCLK]
```

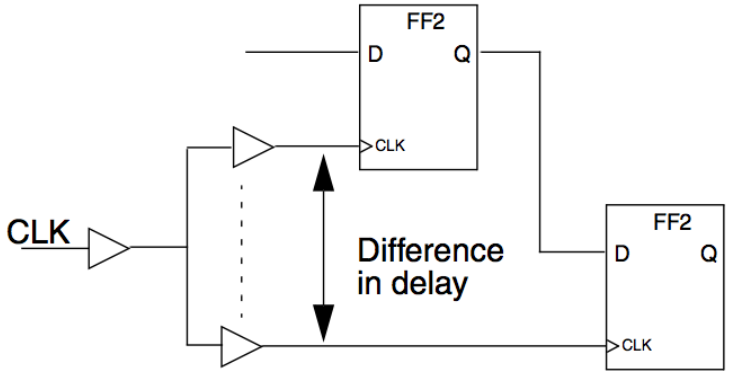
GONZAGA UNIVERSITY

Clock Uncertainty

Figure 2-6 Clock Uncertainty

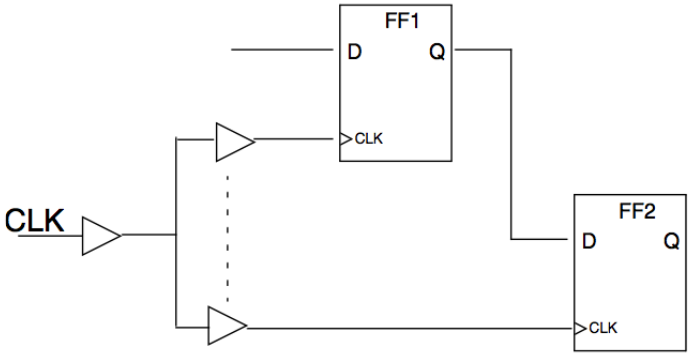


Circuit after logic synthesis

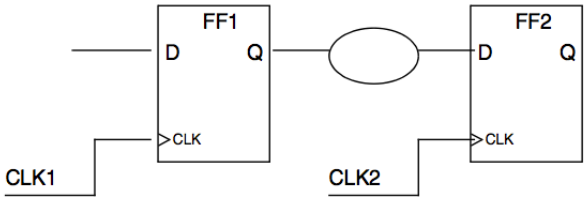


Circuit after clock tree synthesis and layout

Figure 2-7 Simple and Interclock Uncertainty



Simple clock uncertainty



Interclock uncertainty

Example: `set_clock_uncertainty`

- Allows you to specify the expected clock setup or hold uncertainty associated with jitter, skew, and a guard band when performing setup and hold checks for clocks or clock-to-clock transfers. You can specify separate clock uncertainty for setup (-setup) and hold (-hold).

```
set_clock_uncertainty -setup 0.21 [get_clocks SYSCLK]
set_clock_uncertainty -hold 0.33 [get_clocks SYSCLK]
```

Example: derive_clock_uncertanty

GONZAGA UNIVERSITY

This command calculates and applies setup and hold clock uncertainties for each clock-to-clock transfer found in the design. The calculation of the uncertainties is delayed until the next update_timing_netlist call.

```
# create a clock
create_clock –name SYSCLK -period 10.0 [get_ports CLK]

# call derive_clock_uncertainty.
# results will be calculated at the next
# update_timing_netlist call
derive_clock_uncertainty
update_timing_netlist
```

SYNOPSYS®

GONZAGA UNIVERSITY

Clock Transition

To specify a nonzero transition time for an ideal clock, use the `set_clock_transition` command

```
set_clock_transition
 transition_time
 [-rise] [-fall]
 [-min] [-max]
 clock_list
```

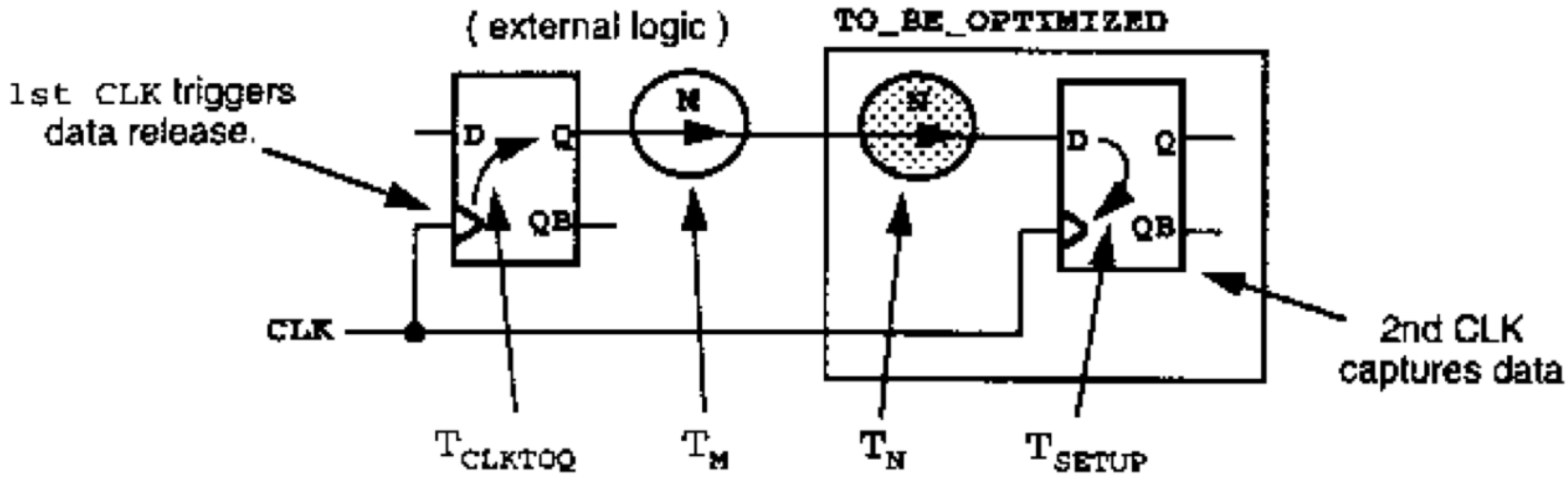
NOTE: This SDC is not supported by ALTERA®

Example:

```
set_clock_transition 0.64 -fall [get_clocks SYSCLK]
```

Use the -rise or -fall option to specify a separate transition time for only rising or only falling edges of the clock. Use the -min or -max option to specify the transition time for minimum operating conditions or maximum operating conditions.

Input Delay

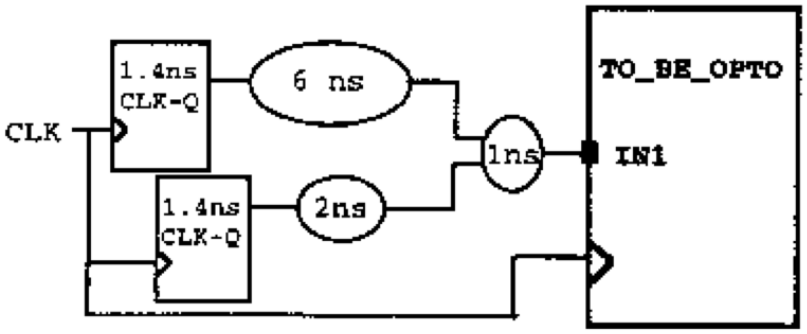


- At what time *after* the clock does the input data arrive ?

Answer: $T_{CLKTOQ} + T_M$: This is the amount to specify as the input delay.

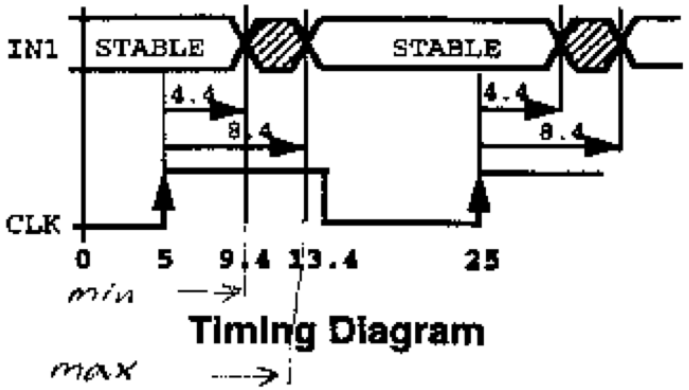
$T_N < T_{CLK} - (T_M + T_{CLKTOQ}) - T_{SETUP}$

Example: `set_input_delay`



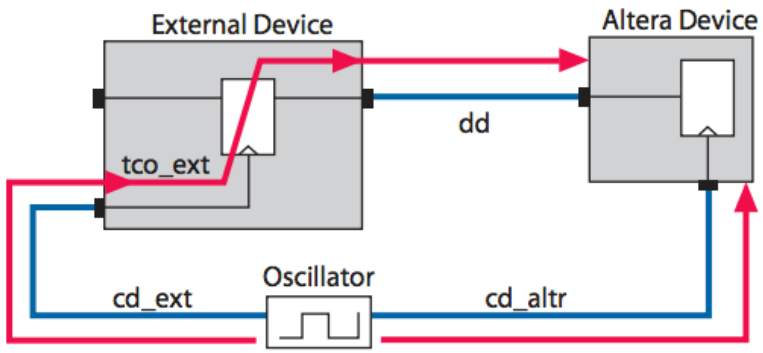
NOTE: outdated syntax

```
create_clock -period 20 -waveform { 5 15 } find (port CLK)
set_input_delay -clock CLK -max 8.4 find (port IN1)
set_input_delay -clock CLK -min 4.4 find (port IN1)
```



Input Constraint (Altera's jargon)

Figure 7–10. Input Delay

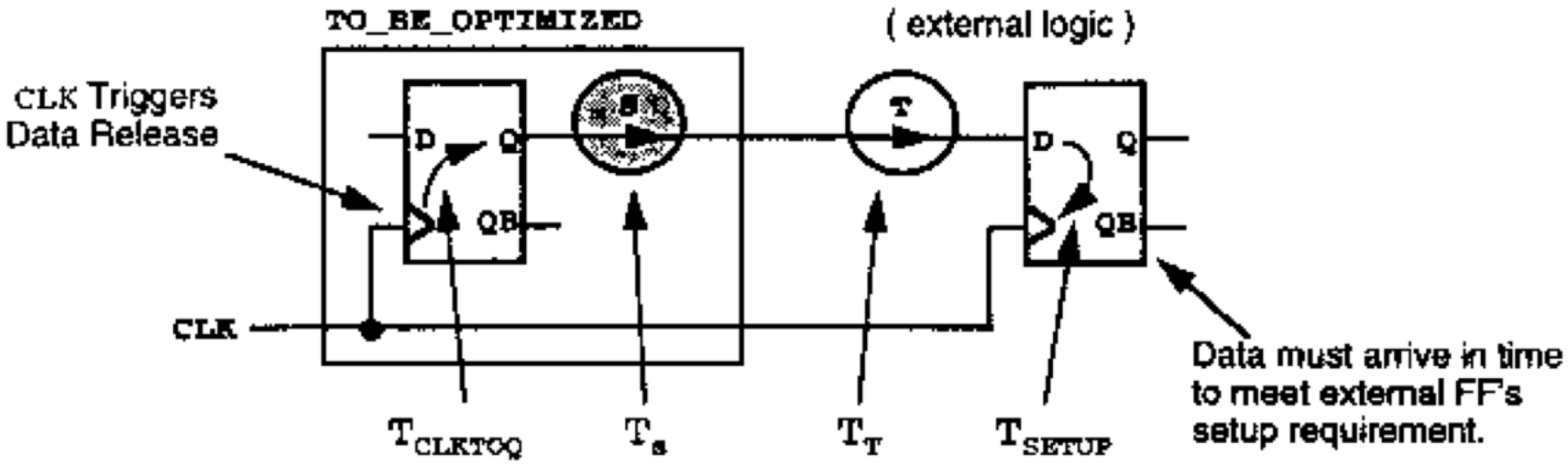


Equation 7–1. Input Delay Calculation

$$\text{input delay}_{MAX} = (\text{cd\_ext}_{MAX} - \text{cd\_altr}_{MIN}) + \text{tco\_ext}_{MAX} + \text{dd}_{MAX}$$

$$\text{input delay}_{MIN} = (\text{cd\_ext}_{MIN} - \text{cd\_altr}_{MAX}) + \text{tco\_ext}_{MIN} + \text{dd}_{MIN}$$

Output Delay

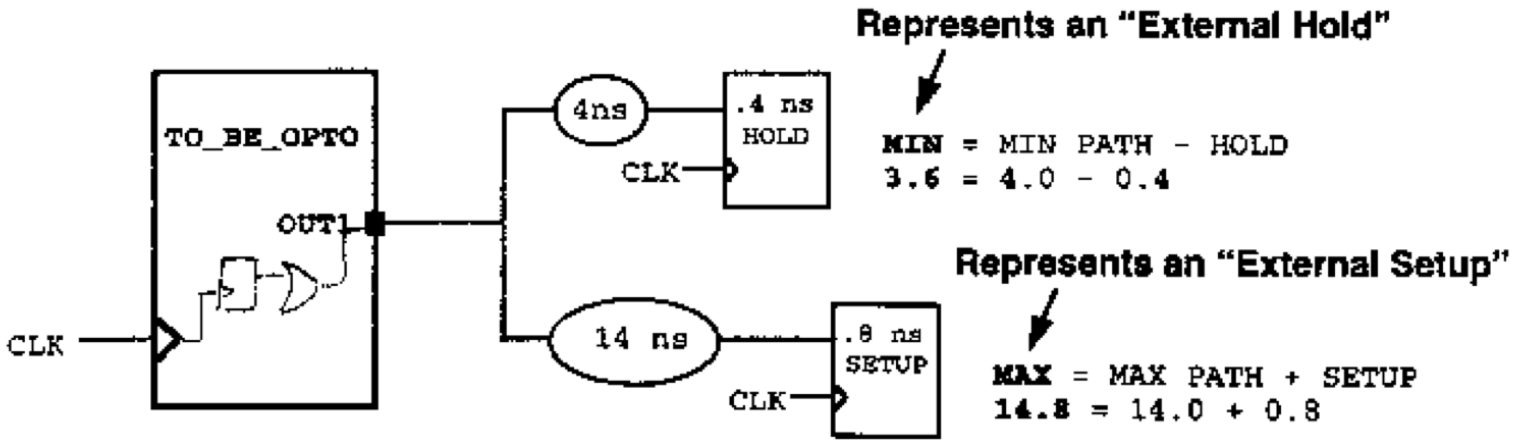


- What is the external logic's *setup* requirement relative to the clock?

Answer : $T_T + T_{SETUP}$ This is the amount to specify as the output delay.

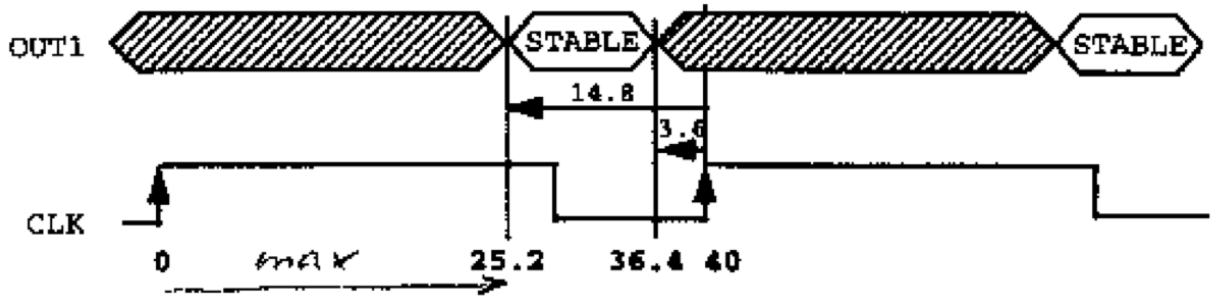
$T_S < T_{CLK} - (T_T + T_{SETUP}) - T_{CLKTOQ}$

Example: `set_output_delay`



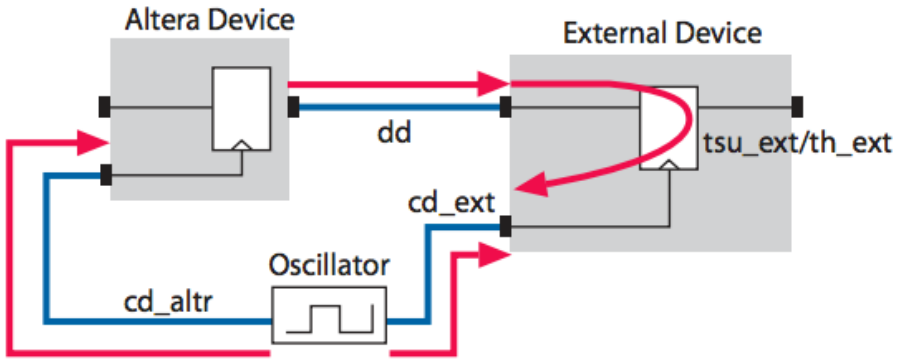

```
create_clock -period 40 -waveform { 0 30 } find (port CLK)
set_output_delay -clock CLK -min 3.6 find (port OUT1)
set_output_delay -clock CLK -max 14.8 find (port OUT1)
```

NOTE: outdated syntax



Output Constraint (Altera's jargon)

Figure 7–11. Output Delay



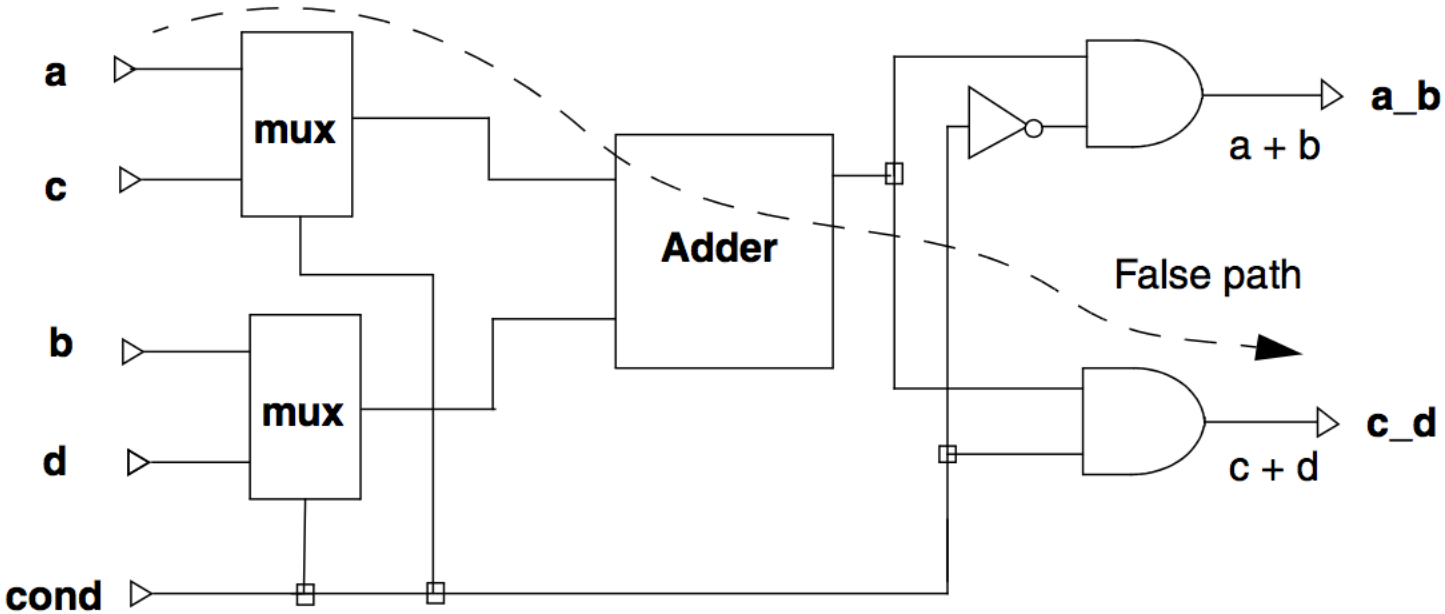
Equation 7–2. output Delay Calculation

$$\text{output delay}_{MAX} = dd_{MAX} + tsu\_ext + (cd\_altr_{MAX} - cd\_ext_{MIN})$$

$$\text{output delay}_{MIN} = (dd_{MIN} + th\_ext + (cd\_altr_{MIN} - cd\_ext_{MAX}))$$

False Path Example

Figure 3-9 False Path Example

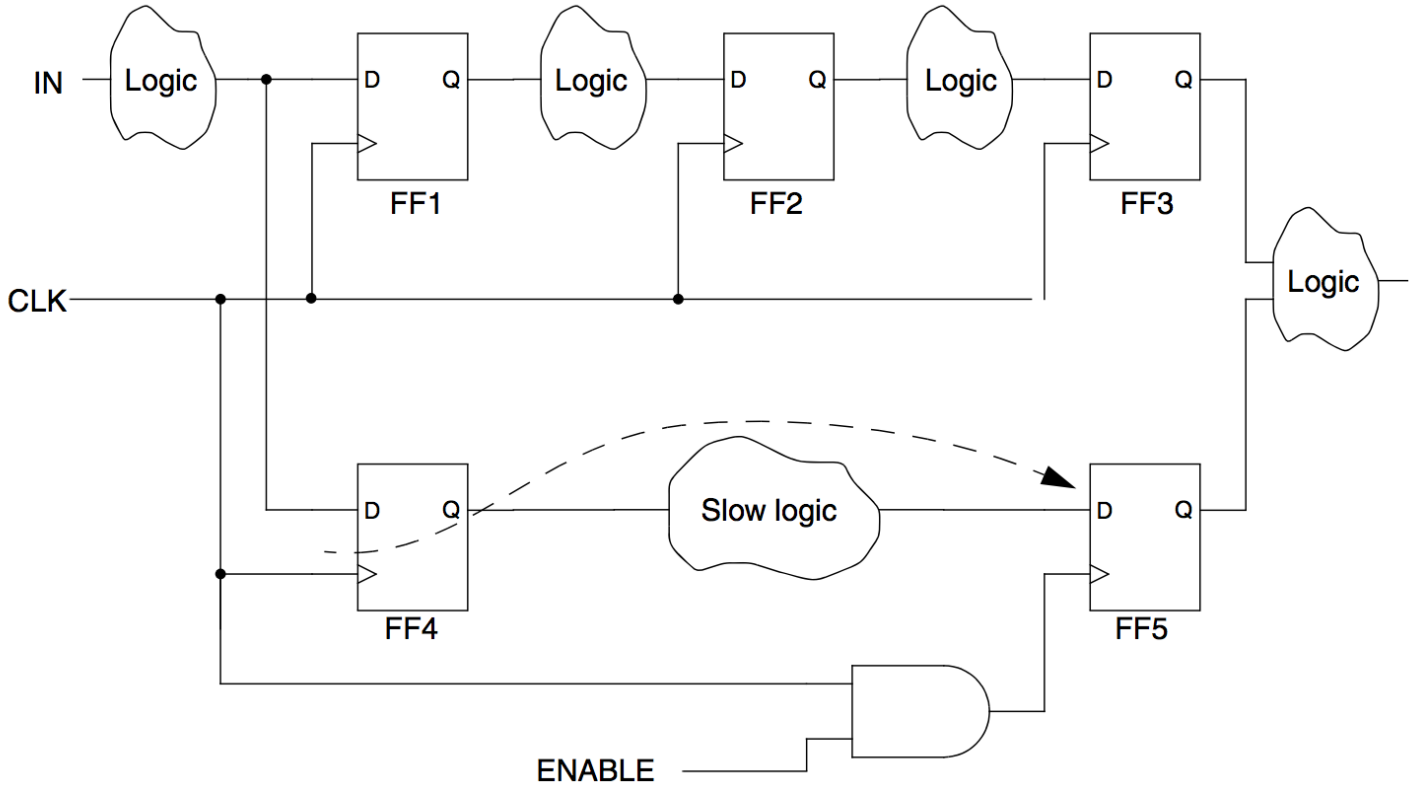



```
prompt> set_false_path -from [get_ports {a b}] -to [get_ports c_d]
prompt> set_false_path -from [get_ports {c d}] -to [get_ports a_b]
```

GONZAGA UNIVERSITY

Multicycle Path Example

Figure 3-10 Multicycle Path Example




```
prompt> set_multicycle_path -setup 2 \
          -from [get_cells FF4] -to [get_cells FF5]
prompt> set_multicycle_path -hold 1 \
          -from [get_cells FF4] -to [get_cells FF5]
```