

DESIGN OF DIGITAL CIRCUITS AND SYSTEMS

Static Timing Analysis

Instructor: Justin Hsia

Teaching Assistants:

Colton Carroll

Grace Zhou

Hemil Patel

Quinlyn Donohue

Rasya Fawwaz

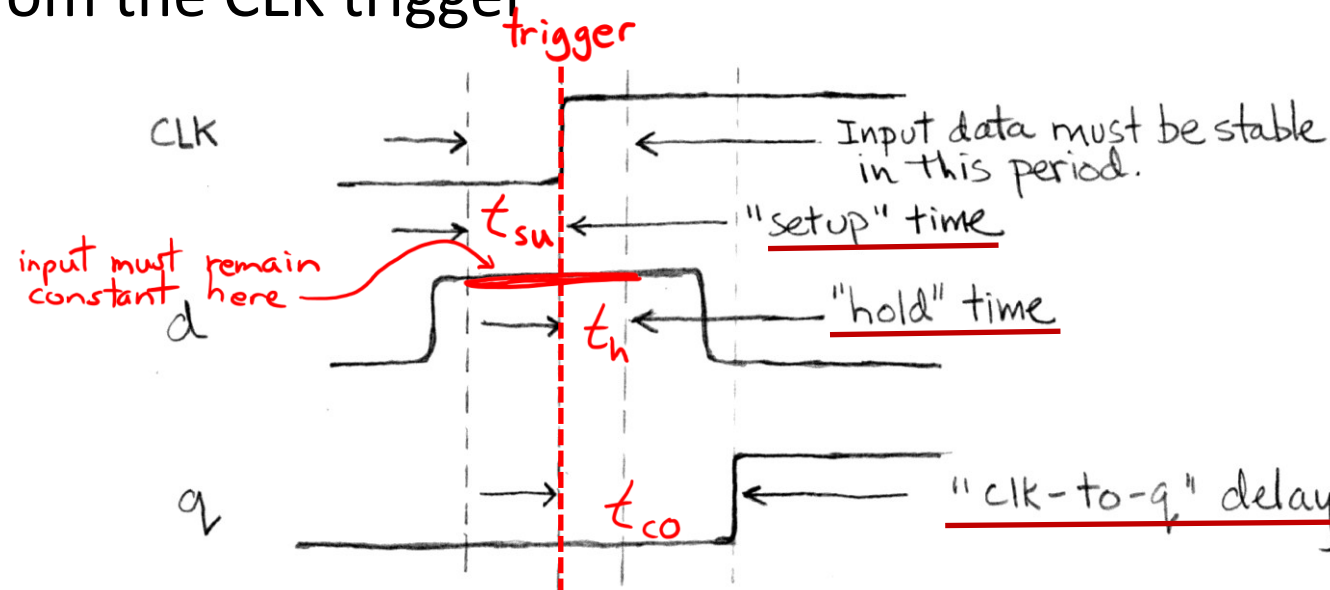
Rose Maresh

Relevant Course Information

- ❖ Lab 4 due Friday (5/8)
- ❖ Quiz 3 is this Thursday at **11:50** am (30 min)
 - ASM and ASMD charts
- ❖ Homework 5 (5/15) and Lab 5 (5/22) will be released on Thursday
 - Lab 5 is the longest “regular” lab, mostly because of debugging – careful with number representation
- ❖ Rest of course material will NOT show up in labs

Review: Sequential Timing Constraints

- ❖ **Setup Time (t_s or t_{su}):** How long the input must be stable *before* the CLK trigger for proper input read
- ❖ **Hold Time (t_h):** How long the input must be stable *after* the CLK trigger for proper input read
- ❖ **"CLK-to-Q" Delay (t_{c2Q} or t_{co}):** How long it takes the output to change, measured from the CLK trigger

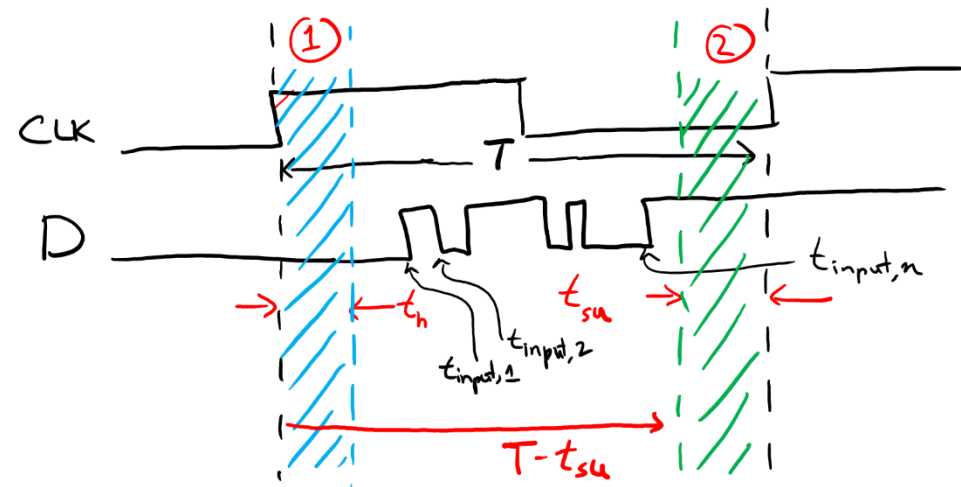


When Can the Input Change?

- ❖ When a register input changes shouldn't violate hold time (t_h) or setup time (t_{su}) constraints within a clock period (T)
- ❖ Let $t_{input,i}$ be the time it takes for the input of a register to change for the i -th time in a single clock cycle, measured from the CLK trigger:
 - Then we need $t_h \stackrel{\textcircled{1}}{\leq} t_{input,i} \stackrel{\textcircled{2}}{\leq} T - t_{su}$ for all i
 - Two separate constraints!

$$\textcircled{1} \quad t_{input,1} \geq t_h$$

$$\textcircled{2} \quad t_{input,n} \leq T - t_{su}$$

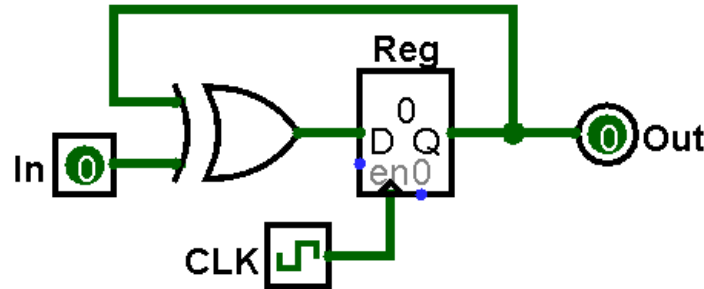


Timing Constraint Questions

- 1) What are you solving for? Which constraint/inequality is relevant?
 - t_h → hold time constraint
 - t_{su} or T → setup time constraint
 - Combinational logic delay → “max” means setup time constraint, “min” means hold time constraint
- 2) Find the most relevant pathway to a register input
 - Hold time constraint → shortest pathway
 - Setup time constraint → longest pathway
- 3) Solve the inequality for chosen path using the given constants

Practice Timing Constraint Question

- ❖ Let $T = 300$ ps,
 $t_{su} = 80$ ps, $t_h = 50$ ps,
 $t_{CO} = 100$ ps.



- 1) If In changes exactly on clock triggers, what are the limits on the XOR gate delay that ensure proper behavior?

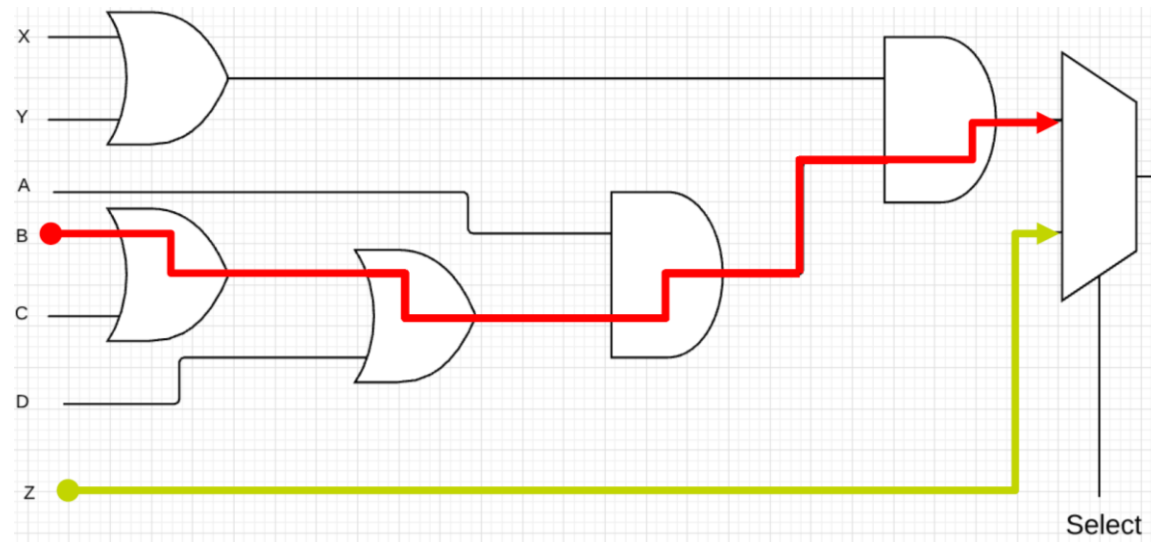
$$t_{XOR} \in (\underline{\quad}, \underline{\quad}) \text{ ps}$$

- 2) Now let $t_{XOR} = 100$ ps and In change a fixed t_{in} after each clock trigger, what ranges of t_{in} will result in proper behavior? Answer within a single clock cycle:

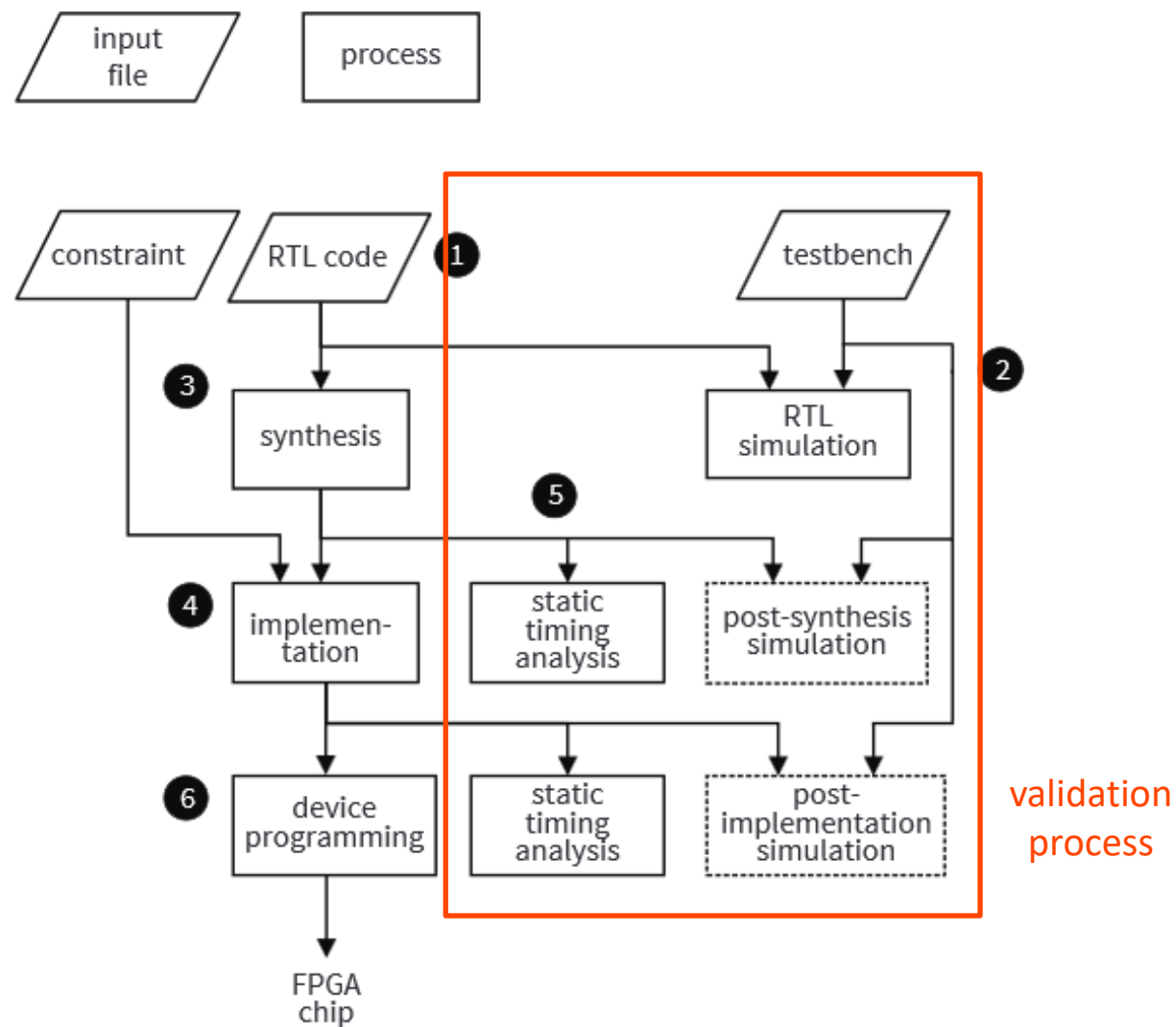
$$t_{in} \in [0, \underline{\quad}) \cup (\underline{\quad}, 300] \text{ ps}$$

What Affects Circuit Timing?

- ❖ A more realistic picture:
 - **Logic depth of circuit pathways**
 - Length, resistance, and capacitance of wire
 - Size of the transistors
 - Operating conditions: Voltage & Temperature



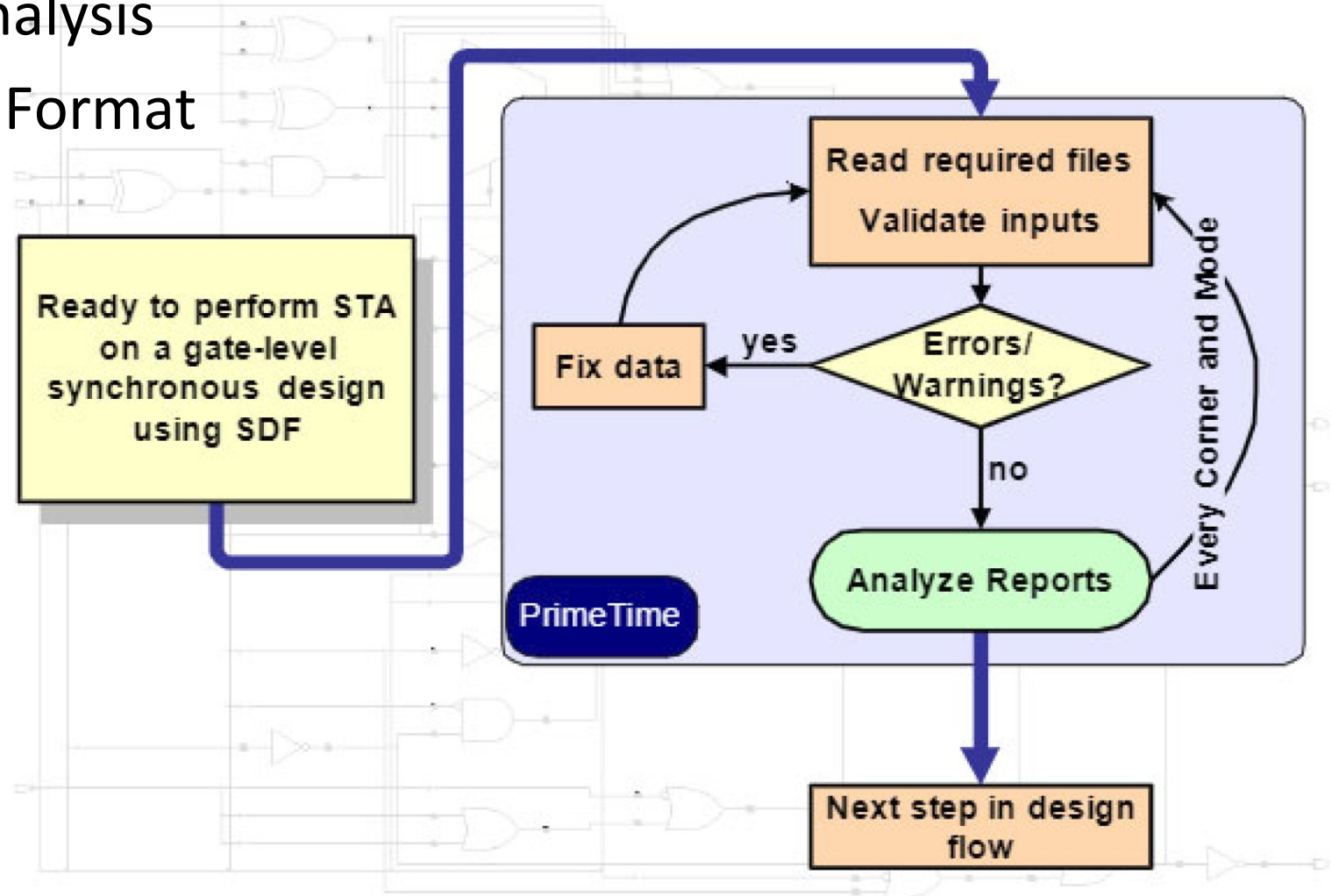
Review: FPGA-Based Design Flow



Source: Figure 2.3 from "FPGA Prototyping" by P. Chu

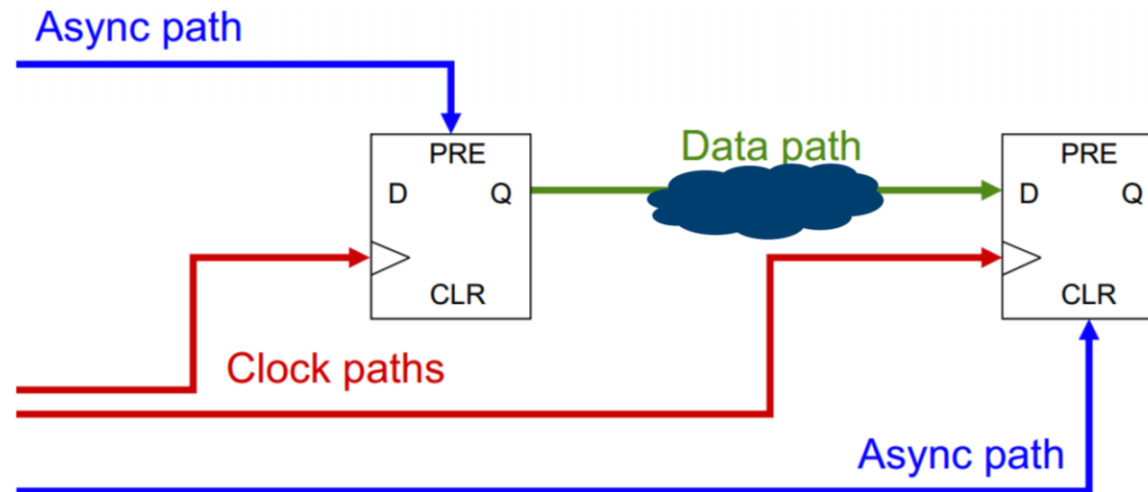
Static Timing Analysis Flow

- ❖ STA = Static Timing Analysis
- ❖ SDF = Standard Delay Format



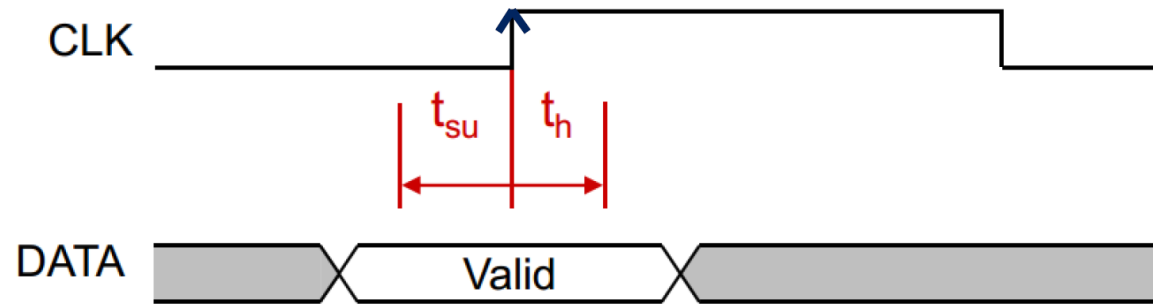
Circuit Path Categorization

- ❖ **Data paths** are between inputs, sequential elements, and outputs
- ❖ **Clock paths** are from device ports or internally-generated clocks to the clock pins of sequential elements
- ❖ **Asynchronous paths** are between inputs and asynchronous set and clear pins of sequential elements



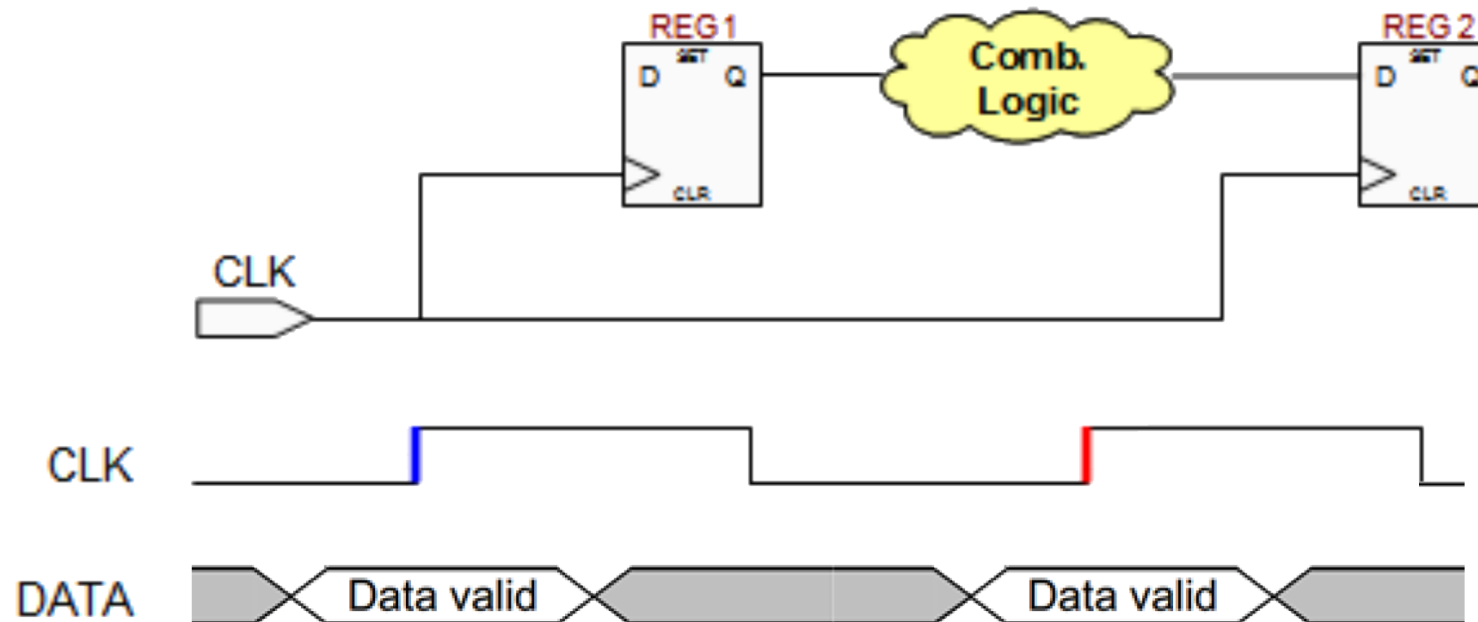
Data Path Validity

- ❖ A data signal is considered **valid** when it has finished computing and is stable
 - Note that this remains true until the next round of computation begins, which can cross clock cycle boundaries
- ❖ For proper behavior (*i.e.*, correct reads), the input to a register must remain valid throughout the setup and hold time constraints



Clock Edges

- ❖ Looking at a single clock cycle
 - **Launch Edge**: the clock edge that activates/launches the **source register**
 - **Latch Edge**: the clock edge that latches the data into the **destination register**



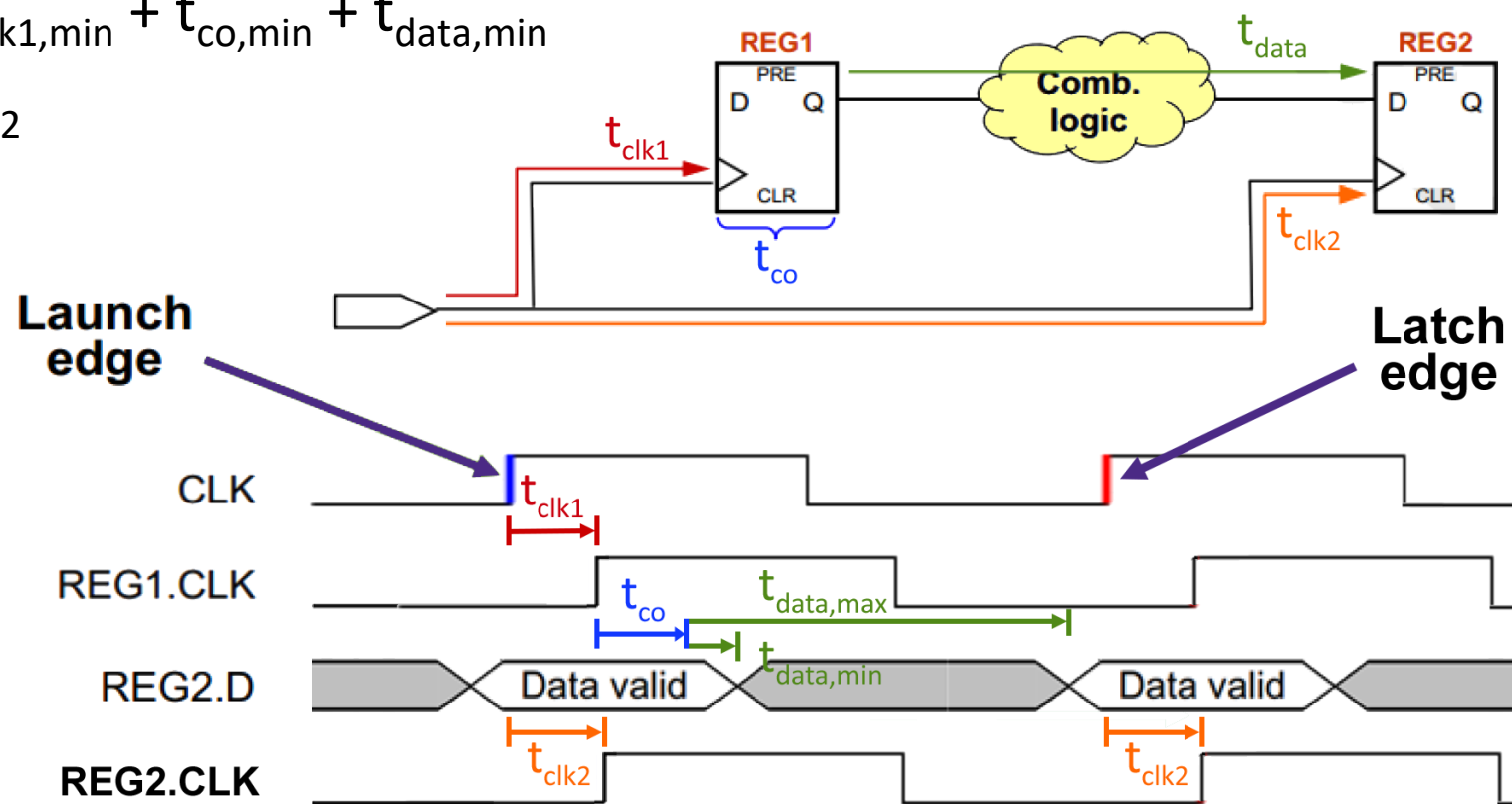
Timing Delays and Points of Interest: Definitions

- ❖ **Clock Network Delay** (t_{clk})
 - How long it takes for changes in the clock signal to arrive at the register – the cause of *clock skew*
- ❖ **Clock-to-Q or Clock-to-out Delay** (t_{C2Q} , t_{CQ} , or t_{CO})
 - How long it takes the output to change, measured from the register's *received* clock edge – a property of the FF/register
- ❖ **Data Arrival Time** (DAT)
 - The time at which the destination register's input is settled
- ❖ **Clock Arrival Time** (CAT)
 - The time at which a clock edge arrives at the destination register

Timing Delays and Points of Interest: Formulas

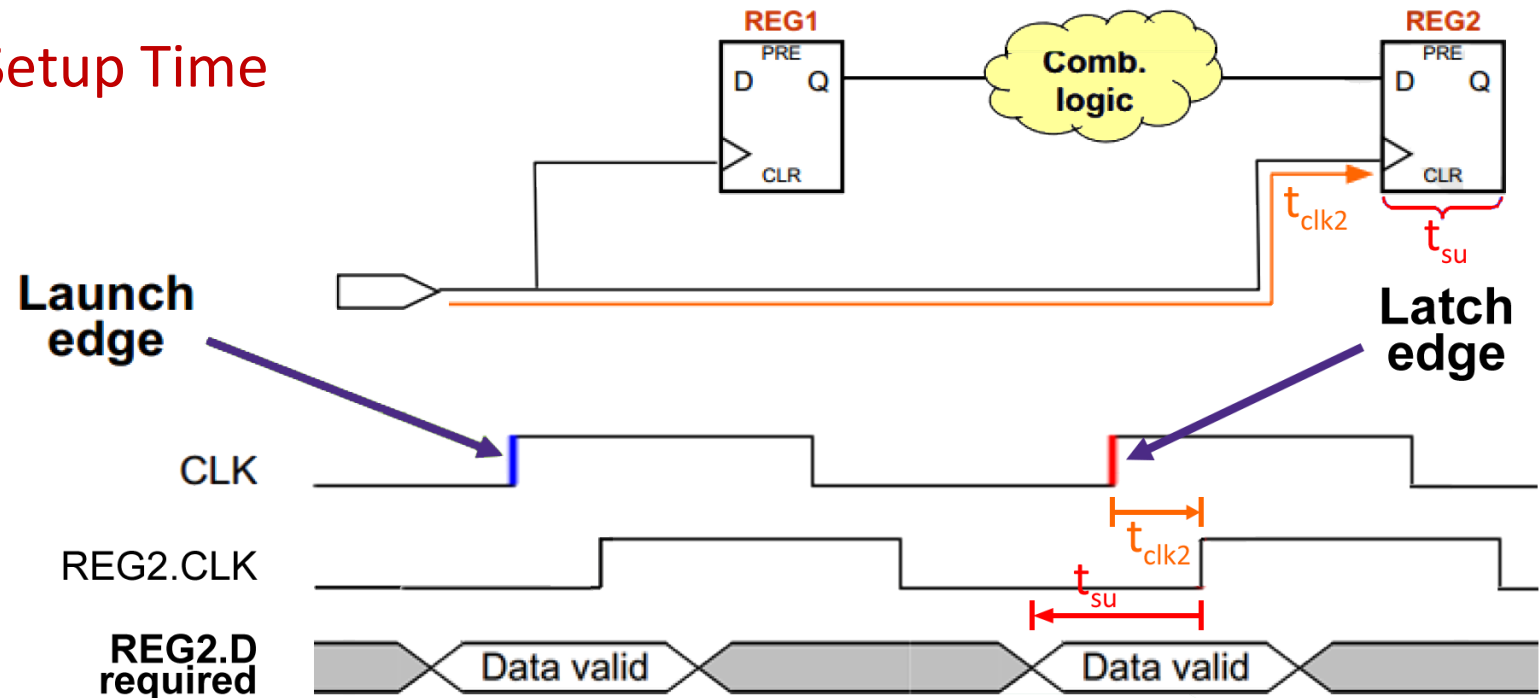
❖ Data Arrival Times (DAT) and Clock Arrival Times (CAT)

- $DAT_{su} = launch\ edge + t_{clk1,max} + t_{co,max} + t_{data,max}$
- $DAT_h = launch\ edge + t_{clk1,min} + t_{co,min} + t_{data,min}$
- $CAT_{latch/launch} = edge + t_{clk2}$



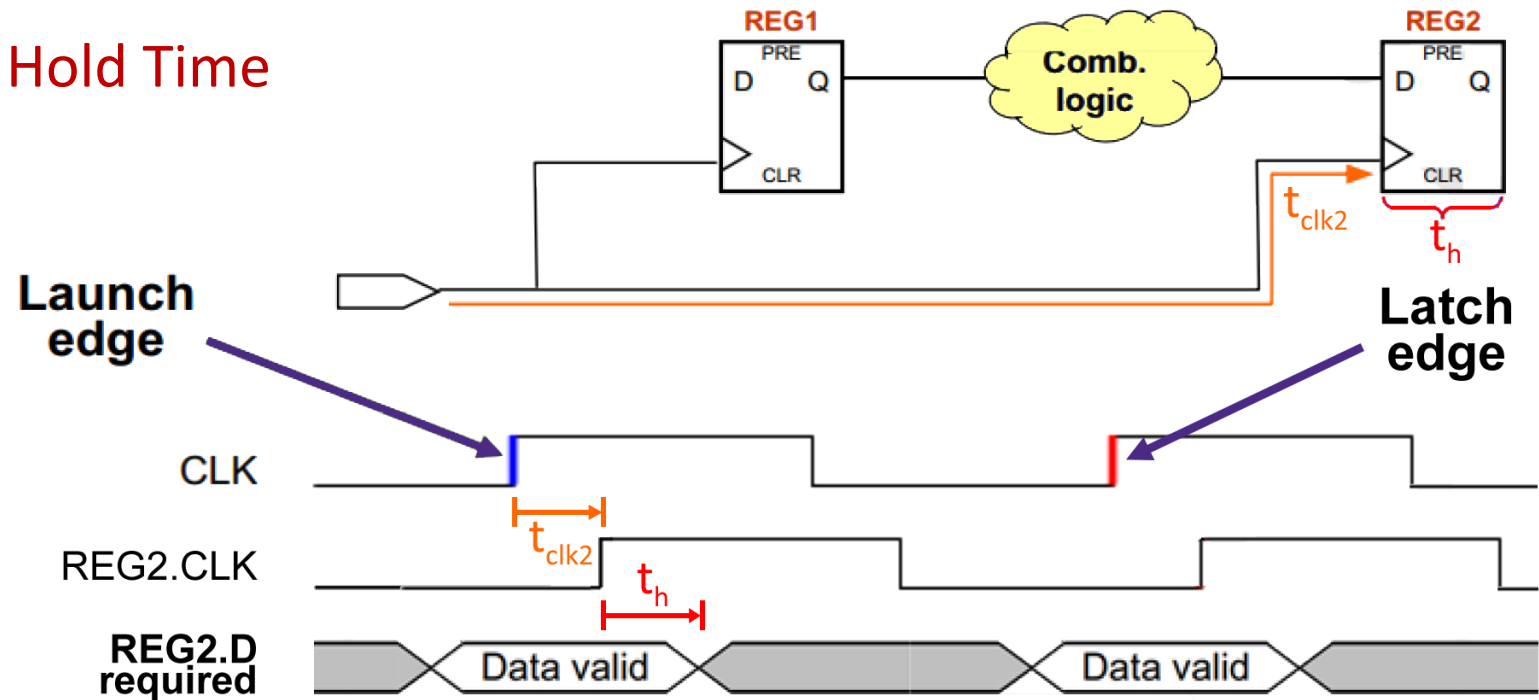
Data Required Time (DRT): Setup

- ❖ Data Required Time for Setup (DRT_{su})
 - The minimum time required *before* the *latch* edge for data to get latched into the destination register
 - Clock Arrival Time_{latch} – Setup Time



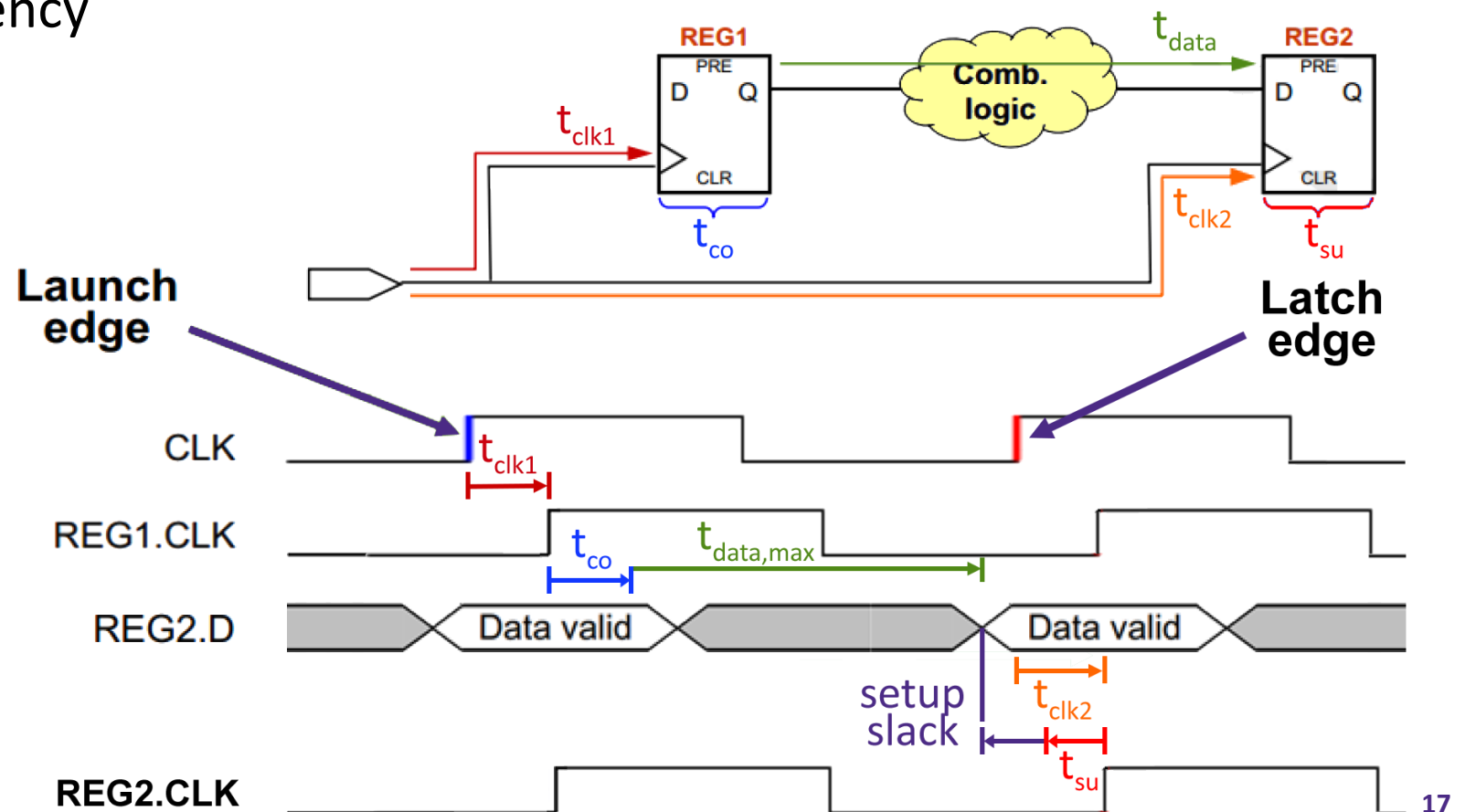
Data Required Time (DRT): Hold

- ❖ Data Required Time for Hold (DRT_h)
 - The minimum time required *after* the *launch* edge for the data to remain valid for successful latching
 - Clock Arrival Time_{launch} + Hold Time



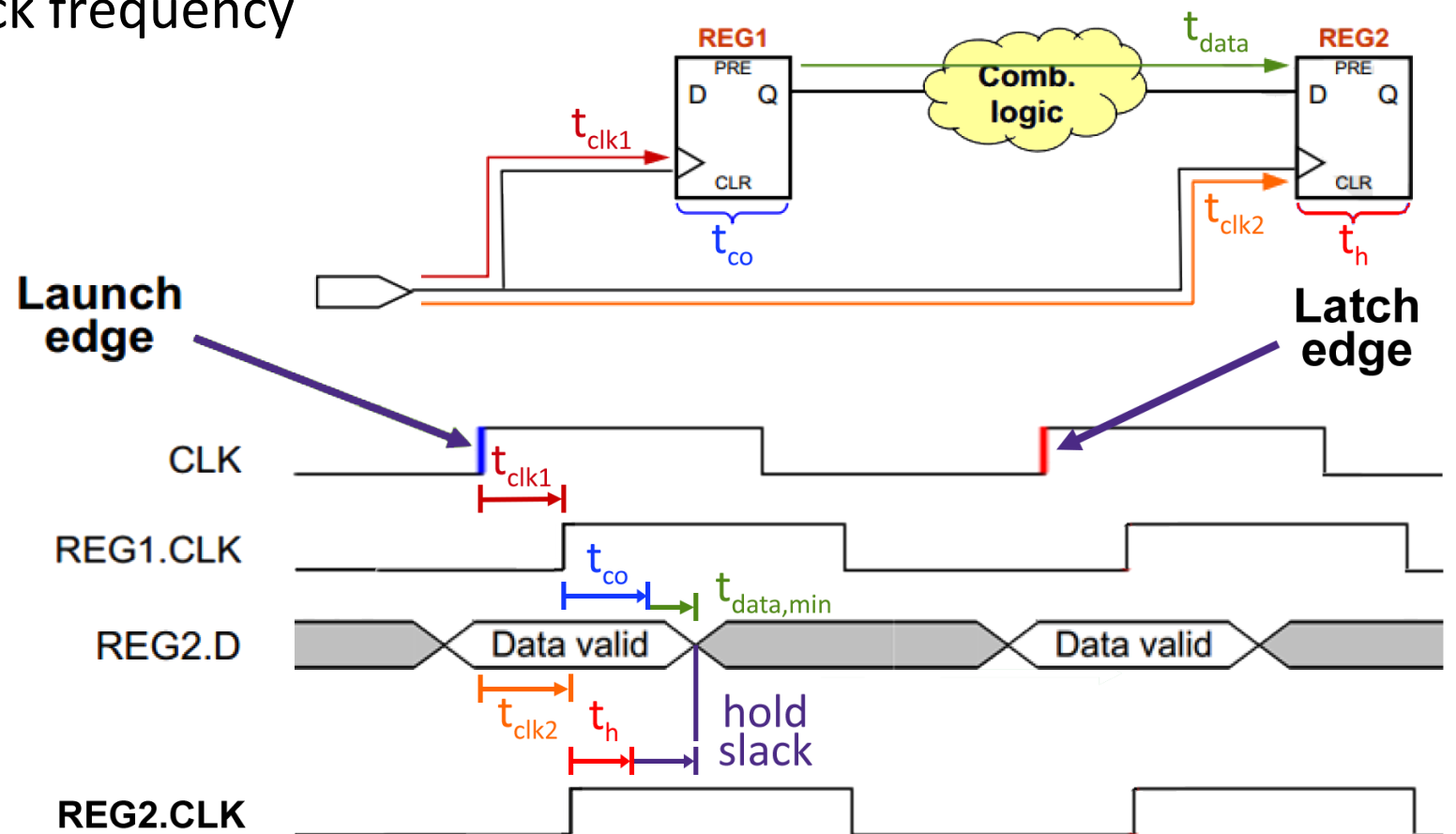
Setup Slack

- ❖ Wiggle room for setup time violations
 - Setup slack = $DRT_{su} - DAT_{su}$
 - Depends on clock frequency



Hold Slack

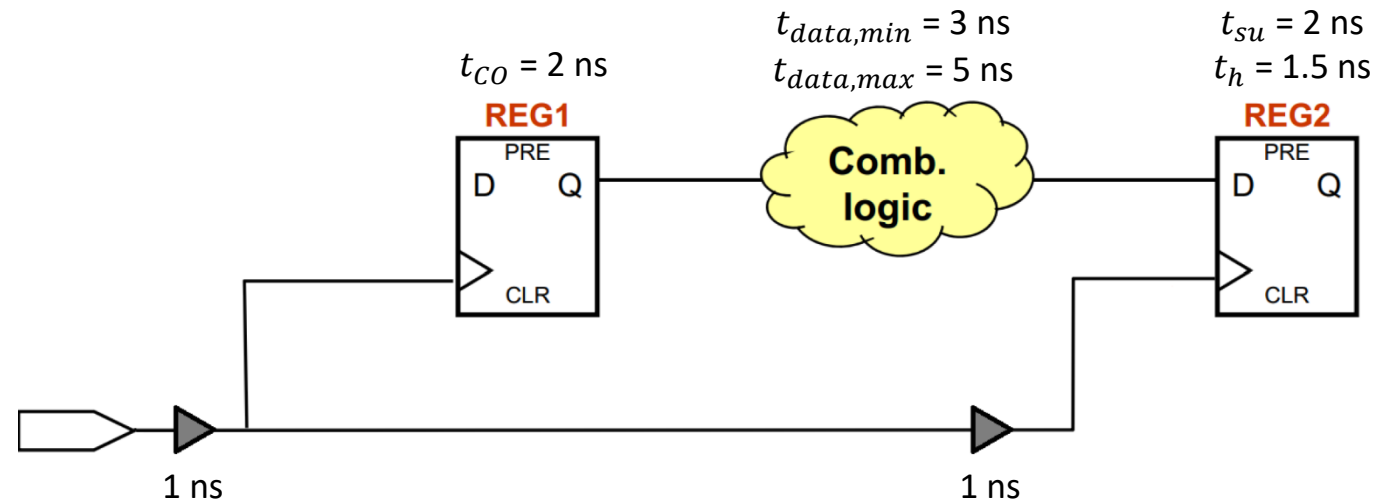
- ❖ Wiggle room for hold time violations
 - Hold slack = $DAT_h - DRT_h$
 - Does not depend on clock frequency



TECHNOLOGY

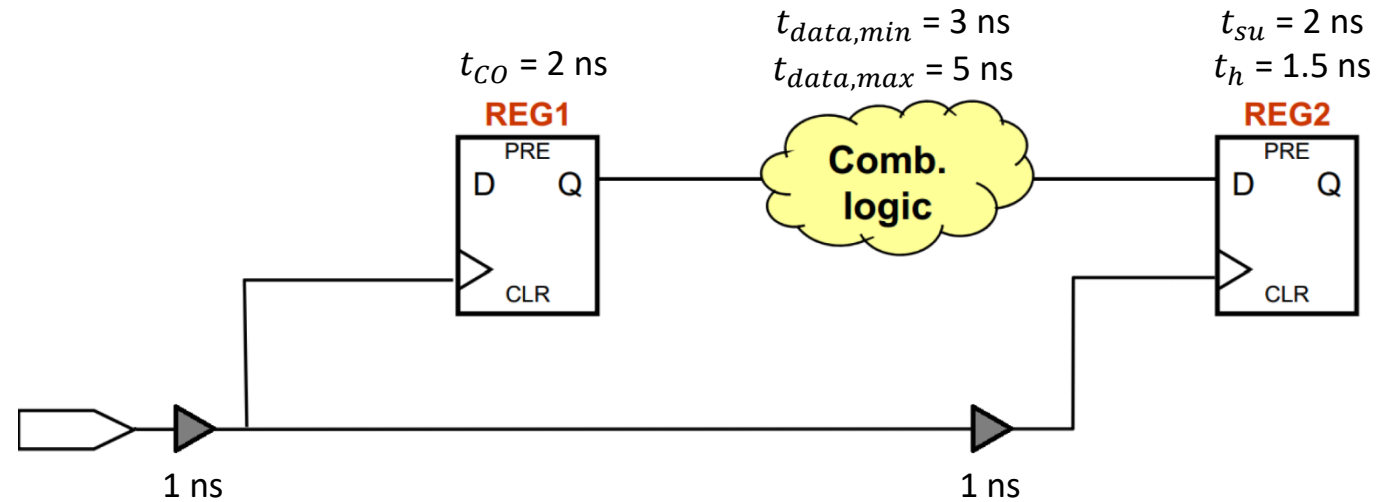
BREAK

Setup Slack Example



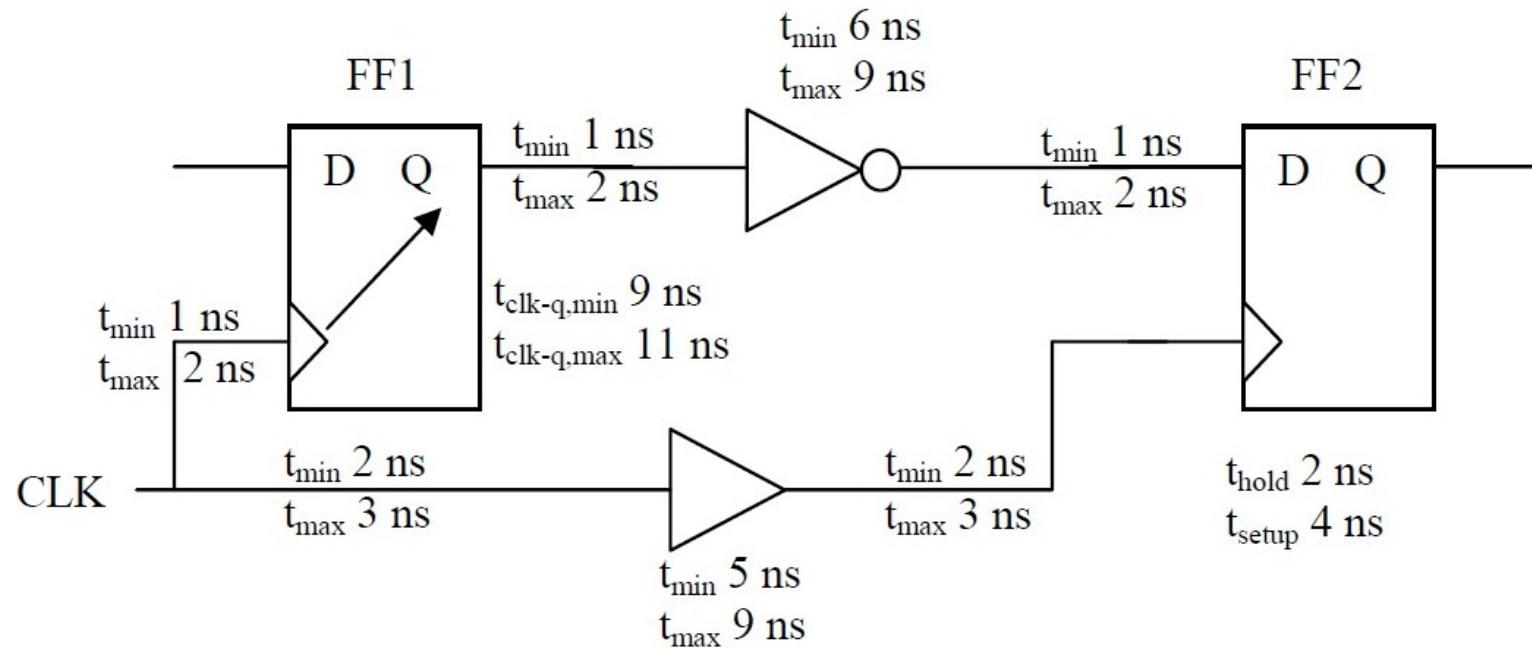
- ❖ Assume 100 MHz clock; clock period $T = 10$ ns
- ❖ Setup slack = $DRT_{su} - DAT_{su}$
 $= \text{min clock path} - \text{max data path}$

Hold Slack Example



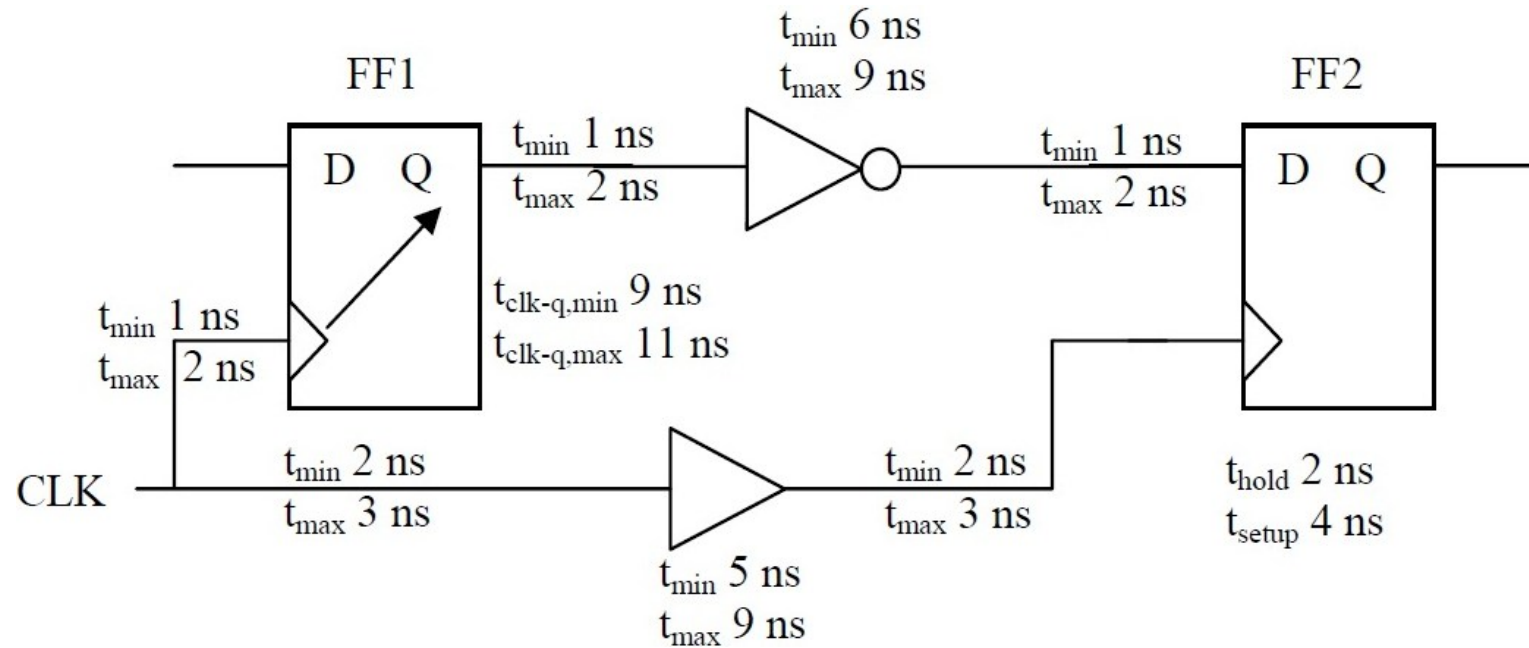
- ❖ Assume 100 MHz clock; clock period $T = 10 \text{ ns}$
- ❖ Hold slack = $DAT_h - DRT_h$
 = *min data path* - *max clock path*

Slack Exercise (1/2)



- ❖ Calculate the setup and hold slacks:
 - Assume 50 MHz clock; clock period $T = 20 \text{ ns}$
 - setup slack = *min* clock path – *max* data path
 - hold slack = *min* data path – *max* clock path

Slack Exercise (2/2)



- ❖ What is the fastest clock we can use with this circuit?

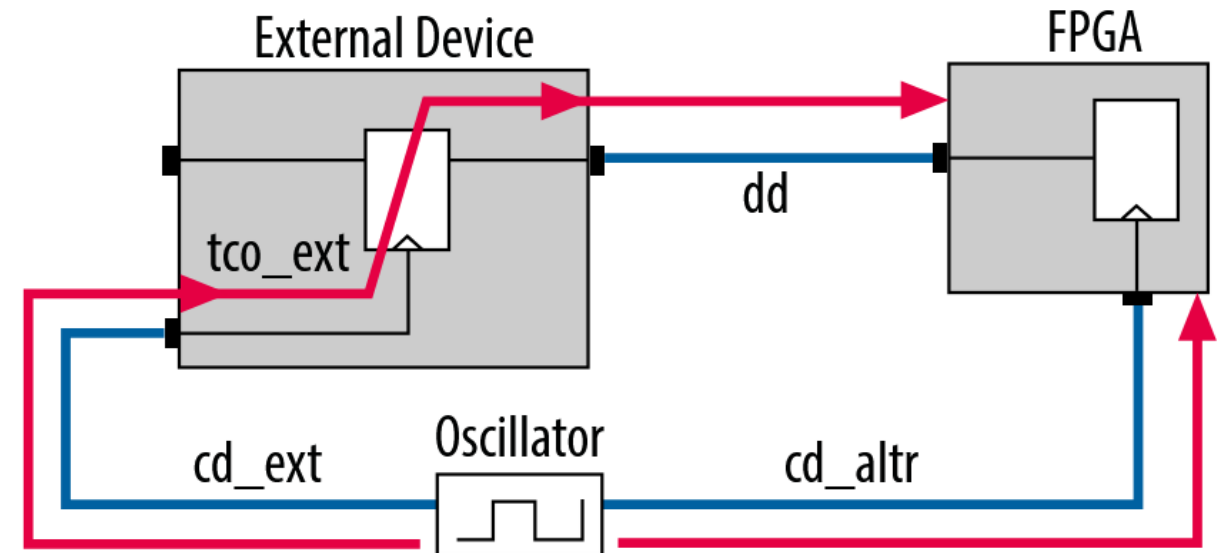
Synopsys Design Constraints (SDC)

- ❖ Synopsys Design Constraints describe timing constraints and exceptions
 - Quartus uses `.sdc` files to define clocks, I/O constraints, and other information that the fitter needs to make the best decisions
 - You can make and edit these using the TimeQuestGUI or by editing the `.sdc` file in a text editor (see Homework 5)

I/O Constraints: Input Delays

❖ set_input_delay

- Command to specify external delays feeding into the FPGA's input ports
- <https://docs.altera.com/r/docs/683243/25.3/quartus-prime-pro-edition-user-guide-timing-analyzer/input-constraints-set-input-delay>

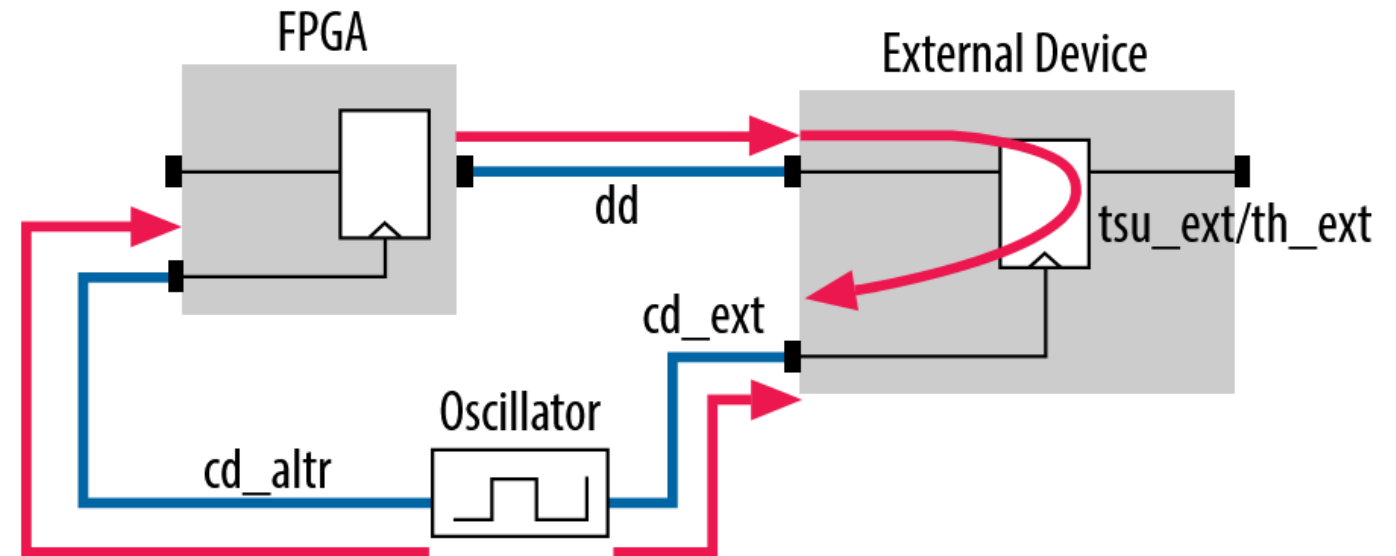


❖ Applied to this setup:

- $\text{input delay}_{\min} = (cd_ext_{\min} - cd_altr_{\max}) + tco_ext_{\min} + dd_{\min}$
- $\text{input delay}_{\max} = (cd_ext_{\max} - cd_altr_{\min}) + tco_ext_{\max} + dd_{\max}$
 - “cd” = clock delay, “dd” = data delay, “altr” = Altera FPGA

I/O Constraints: Output Delays

- ❖ `set_output_delay`
 - Command to specify external delays leaving the FPGA's output ports
 - https://docs.altera.com/r/docs/683243/25.3/quartus-prime-pro-edition-user-guide-timing-analyzer/output-constraints-set_output_delay



- ❖ Applied to this setup:
 - $\text{output delay}_{\min} = (\text{cd_altr}_{\min} - \text{cd_ext}_{\max}) + \text{dd}_{\min} - \text{th_ext}_{\max}$
 - $\text{output delay}_{\max} = (\text{cd_altr}_{\max} - \text{cd_ext}_{\min}) + \text{dd}_{\max} + \text{tsu_ext}_{\max}$
 - “cd” = clock delay, “dd” = data delay, “altr” = Altera FPGA

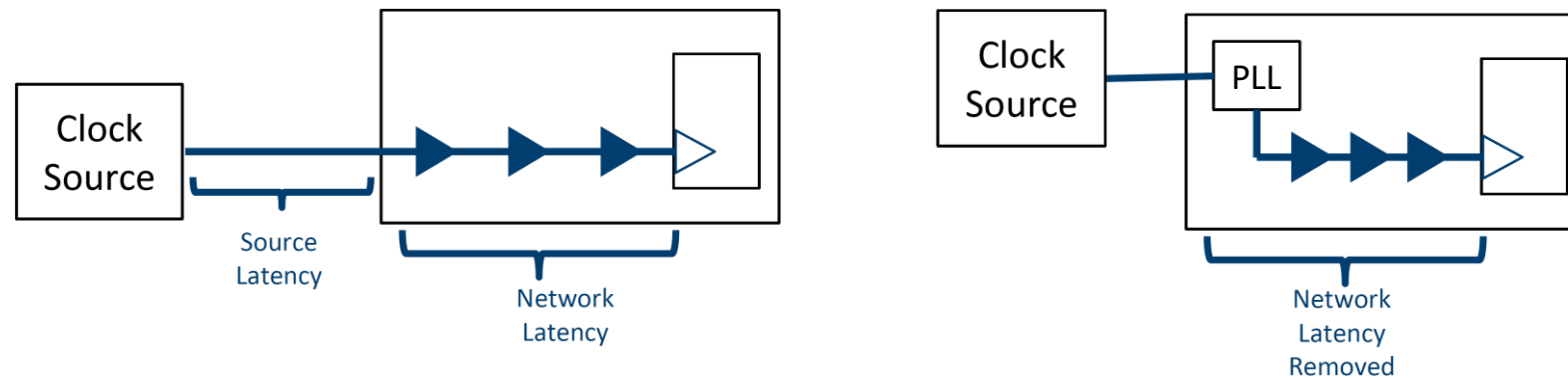
I/O Constraints: Datasheets

❖ Check the datasheet!

READ/WRITE CYCLE SWITCHING CHARACTERISTICS (Over Operating Range)						
Symbol	Parameter	-166		-133		Unit
		Min.	Max.	Min.	Max.	
f _{MAX} ⁽³⁾	Clock Frequency	—	166	—	133	MHz
t _{CC} ⁽³⁾	Cycle Time	6	—	7.5	—	ns
t _{KH}	Clock High Time	2.4	—	2.8	—	ns
t _{KL} ⁽³⁾	Clock Low Time	2.4	—	2.8	—	ns
t _{CA} ⁽³⁾	Clock Access Time	—	3.5	—	4	ns
t _{COX} ⁽¹⁾	Clock High to Output Invalid	3	—	3	—	ns
t _{COLZ} ^(1,2)	Clock High to Output Low-Z	0	—	0	—	ns
t _{COHZ} ^(1,2)	Clock High to Output High-Z	1.5	3.5	1.5	3.5	ns
t _{OEQ} ⁽³⁾	Output Enable to Output Valid	—	3.5	—	3.8	ns
t _{EOX} ⁽¹⁾	Output Disable to Output Invalid	0	—	0	—	ns
t _{ELZ} ^(1,2)	Output Enable to Output Low-Z	0	—	0	—	ns
t _{EHZ} ^(1,2)	Output Disable to Output High-Z	2	4.5	2	5	ns
t _{AS} ⁽³⁾	Address Setup Time	2.1	—	2.1	—	ns
t _{SS} ⁽³⁾	Address Status Setup Time	1.5	—	1.5	—	ns
t _{WS} ⁽³⁾	Write Setup Time	1.5	—	1.5	—	ns
t _{CS} ⁽³⁾	Chip Enable Setup Time	1.5	—	1.5	—	ns
t _{AVS} ⁽³⁾	Address Advance Setup Time	1.5	—	1.5	—	ns
t _{AH} ⁽³⁾	Address Hold Time	1.0	—	1.0	—	ns
t _{SH} ⁽³⁾	Address Status Hold Time	0.5	—	0.5	—	ns
t _{WH} ⁽³⁾	Write Hold Time	0.5	—	0.5	—	ns
t _{CEH} ⁽³⁾	Chip Enable Hold Time	0.5	—	0.5	—	ns
t _{AVH} ⁽³⁾	Address Advance Hold Time	0.5	—	0.5	—	ns

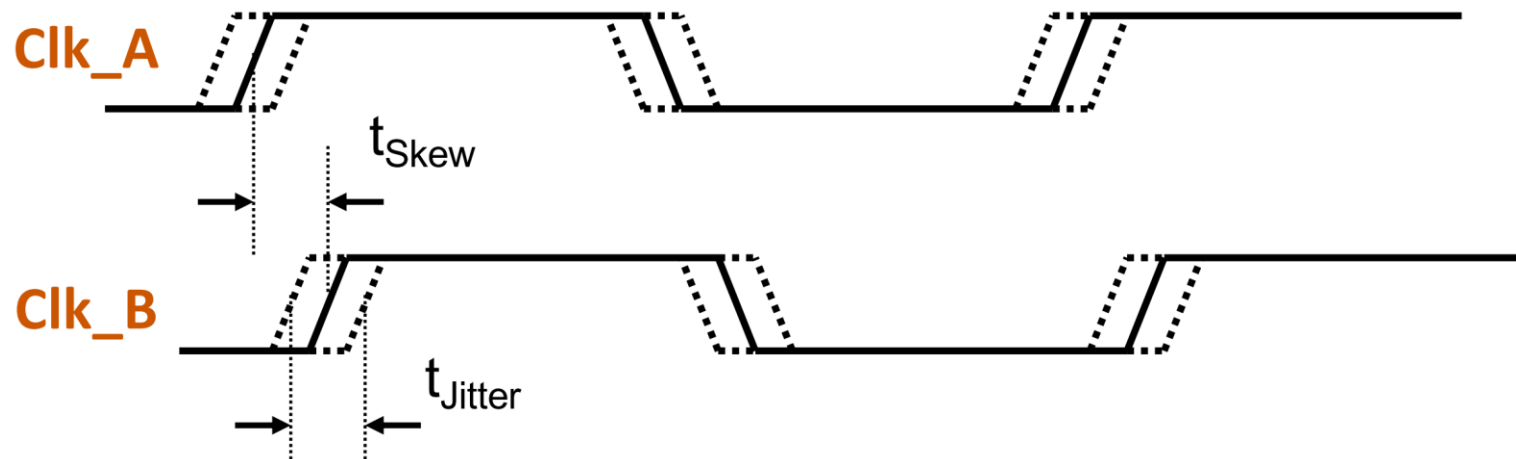
Clock Effects: Latency

- ❖ **Clock Latency:** Delay for clock signal to reach components
 - Source latency: The delay between the clock definition and its source
 - Network latency: The delay between the clock definition and register clock pins
 - [extra] Often dealt with using an onboard phase-locked loop (PLL) – see `derive_pll_clocks` command



Clock Effects: Uncertainty

- ❖ **Clock Uncertainty:** Clock signal does not arrive at every clock pin simultaneously
 - Skew: Differences in arrival time of a clock signal to different components
 - Jitter: Deviations from defined period
 - Synchronous clock domain crossings (future lecture)



Timing Closure

- ❖ Timing constraints need to be met *simultaneously*
 - Sometimes fixing one violation will introduce another 🤪
- ❖ **Hold violations:** Fast data path and/or destination register's clock latency
 - Fix: Add delay in data path with buffers or pairs of inverters (Quartus does this automatically!)
- ❖ **Setup violations:** Data arrives too late compared to the destination register's clock speed
 - Slow down the clock (undesirable)
 - Tell fitter to try harder or confine logic to a smaller area
 - Rewrite code to simplify logic
 - Add *pipelining* (next lecture)