Sequential Circuits: Timing

Digital Design and Computer Architecture
Mohammad Sadrosadati
Frank K. Gürkaynak

http://safari.ethz.ch/ddca

Adapted from Digital Design and Computer Architecture, David Money Harris & Sarah L. Harris ©2007 Elsevier

What will we learn?

- **Timing of Sequential Logic**
  - Setup time
  - Hold time
- **Clock frequency**
- **Metastability and prevention**

Timing

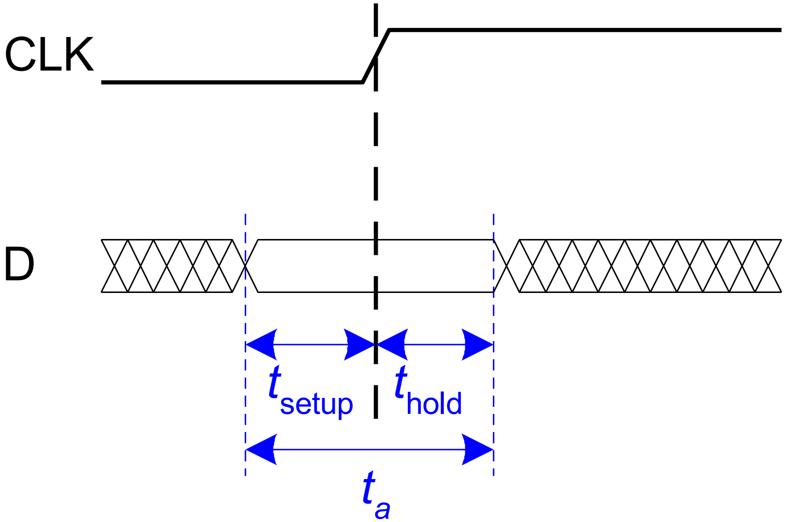
- **Flip-flop samples D at clock edge**
- **D must be stable when it is sampled**
- **Similar to a photograph, D must be stable around the clock edge**
- **If D is changing when it is sampled, *metastability* can occur**
  - Recall that a flip-flop copies the input D to the output Q on the rising edge of the clock. This process is called sampling D on the clock edge. If D is stable at either 0 or 1 when the clock rises, this behavior is clearly defined. But what happens if D is changing at the same time the clock rises?

Input Timing Constraints

- *Setup time*: t_{setup} = time before the clock edge that data must be stable (i.e. not changing)

- *Hold time*: t_{hold} = time after the clock edge that data must be stable

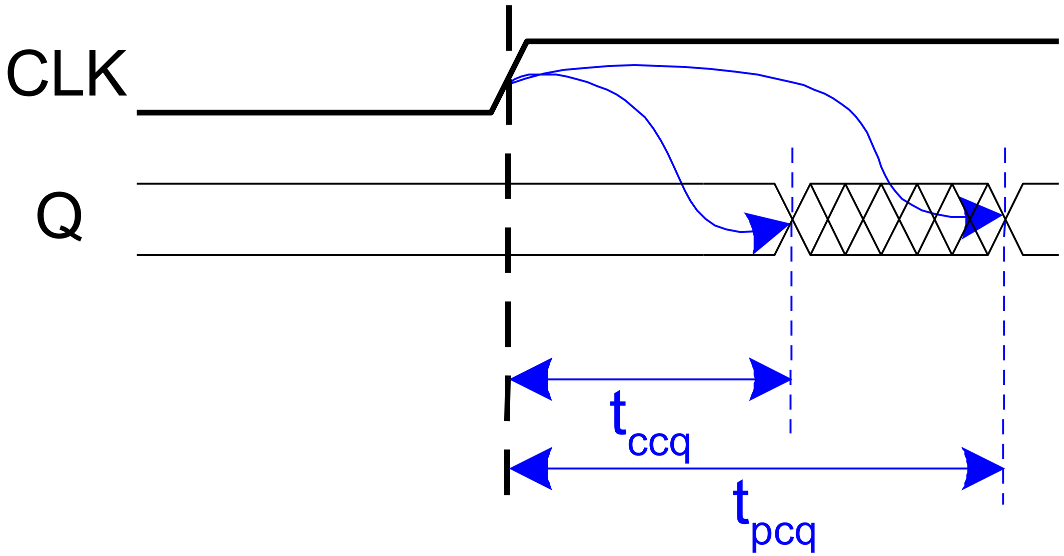
- *Aperture time*: t_a = time around clock edge that data must be stable ($t_a = t_{setup} + t_{hold}$)



Output Timing Constraints

- ***Propagation delay***: t_{pcq} = time after clock edge that the output Q is guaranteed to be stable (i.e., to stop changing)

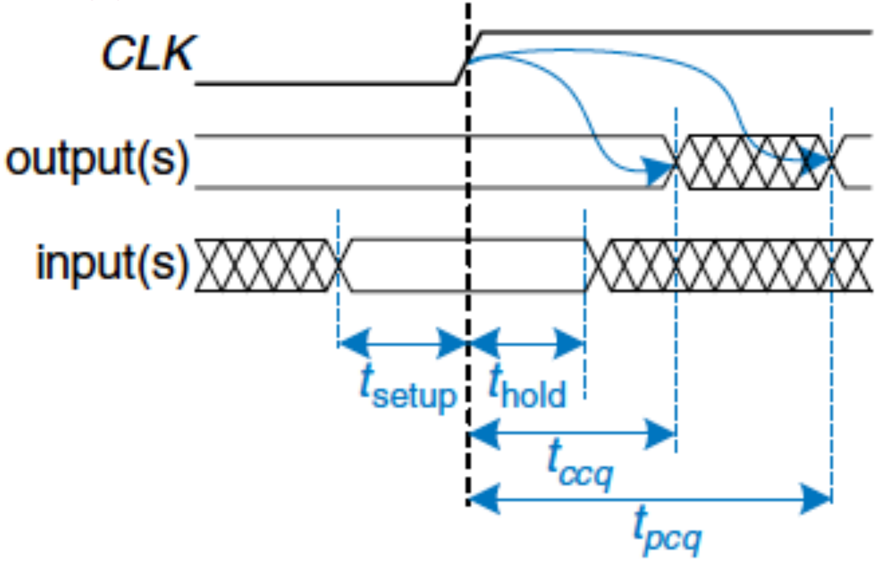
- ***Contamination delay***: t_{ccq} = time after clock edge that Q might be unstable (i.e., start changing)



Dynamic Discipline

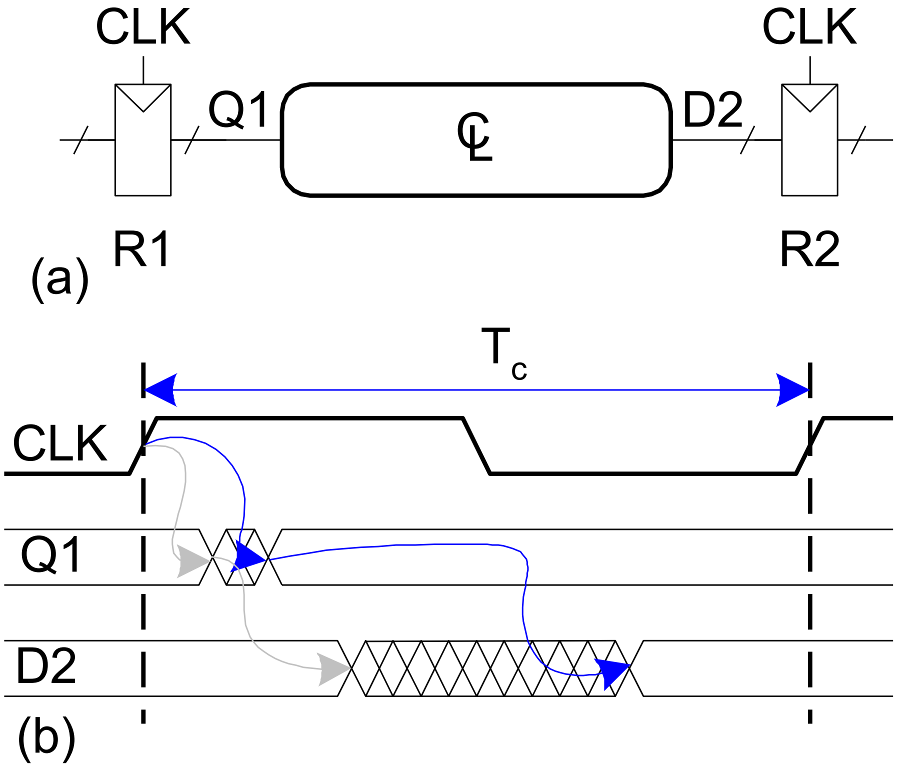
- **The input to a synchronous sequential circuit must be stable during the aperture (setup and hold) time around the clock edge.**

- **Specifically, the input must be stable**
  - at least t_{setup} **before** the clock edge
  - at least until t_{hold} **after** the clock edge



Dynamic Discipline

- **The delay between registers has a minimum and maximum delay, dependent on the delays of the circuit elements**



Setup Time Constraint

- **The clock period or cycle time, T_c, is the time between rising edges of a repetitive clock signal. Its reciprocal, $f_c=1/T_c$, is the clock frequency.**

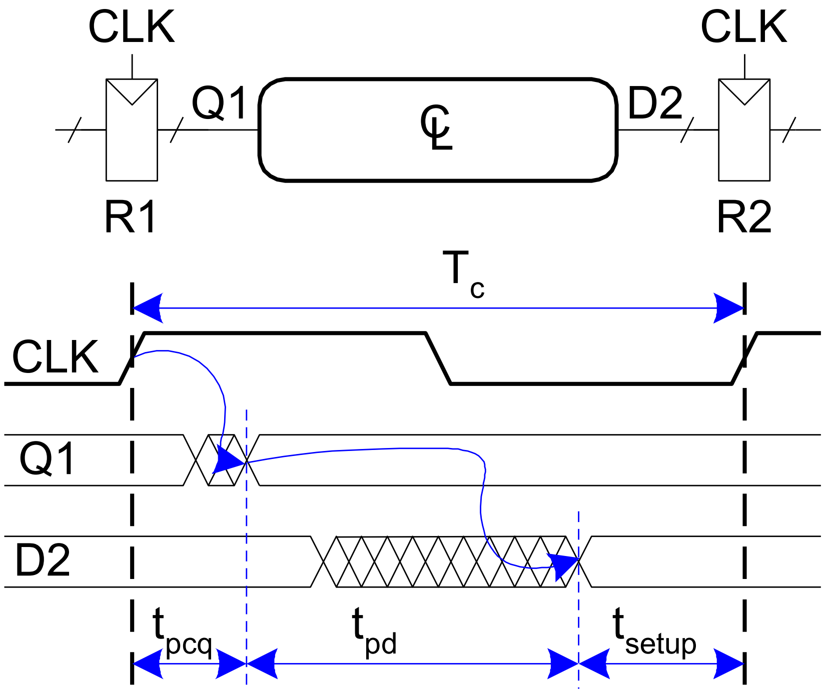
- **All else being the same, increasing the clock frequency increases the work that a digital system can accomplish per unit time.**

- **Frequency is measured in units of Hertz (Hz), or cycles per second:**
  - 1 megahertz (MHz) 10^6 Hz
  - 1 gigahertz (GHz) 10^9 Hz.

Setup Time Constraint

- The setup time constraint depends on the maximum delay from register R1 through the combinational logic.

- The input to register R2 must be stable at least t_{setup} before the clock edge.

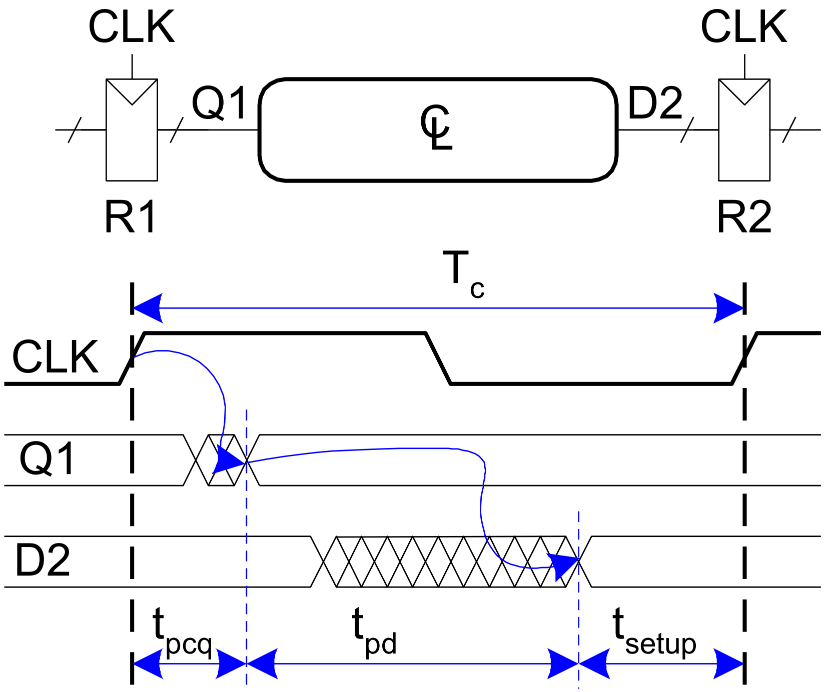


$T_c >=$

Setup Time Constraint

- The setup time constraint depends on the maximum delay from register R1 through the combinational logic.

- The input to register R2 must be stable at least t_{setup} before the clock edge.



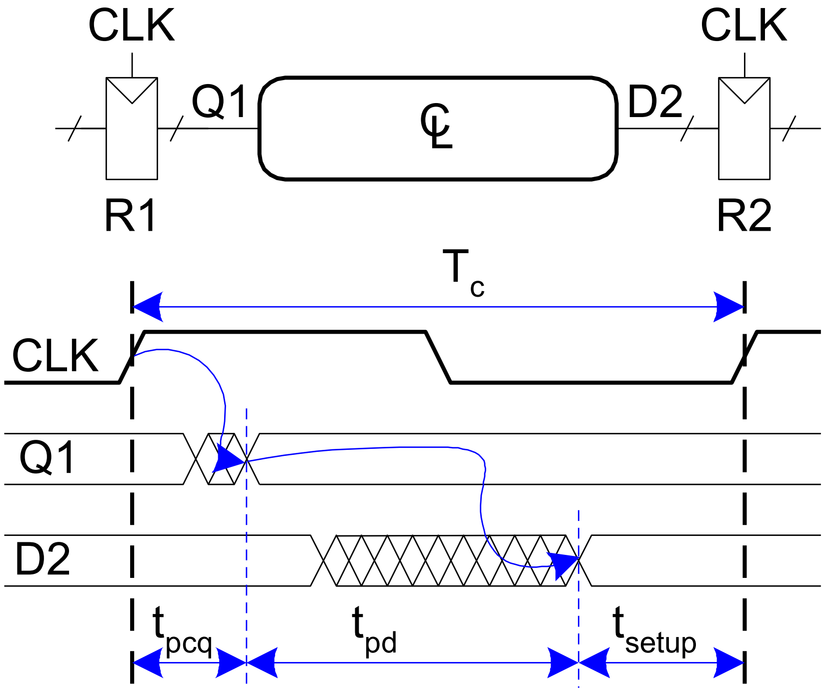
$T_c >= t_{pcq} + t_{pd} + t_{setup}$

$t_{pd} <=$

Setup Time Constraint

- The setup time constraint depends on the maximum delay from register R1 through the combinational logic.

- The input to register R2 must be stable at least t_{setup} before the clock edge.



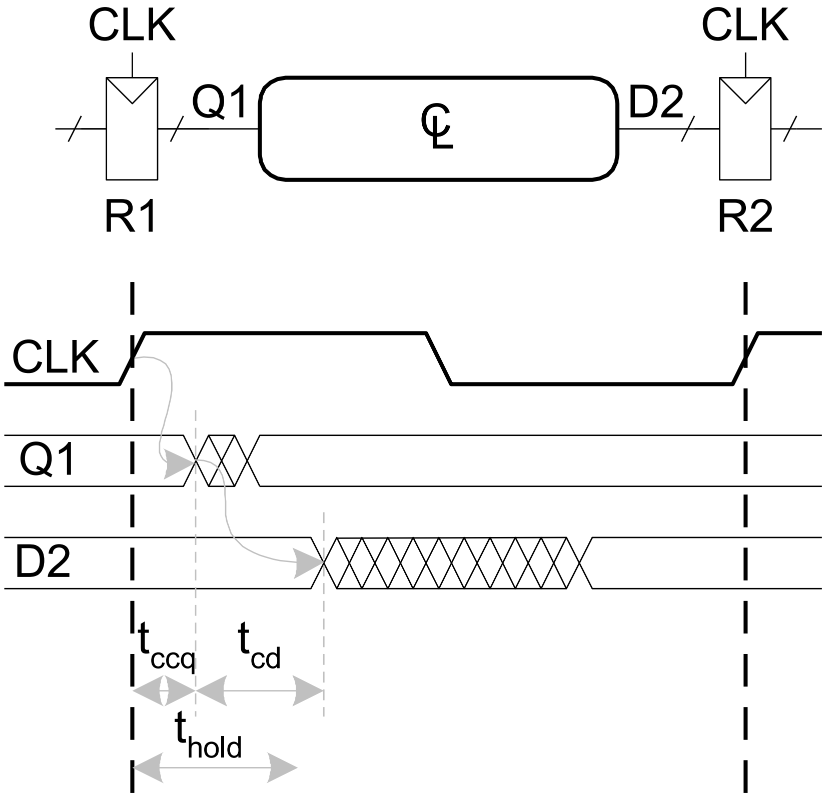
$$T_c >= t_{pcq} + t_{pd} + t_{setup}$$

$$t_{pd} <= T_c - (t_{pcq} + t_{setup})$$

Hold Time Constraint

- The hold time constraint depends on the minimum delay from register R1 through the combinational logic.

- The input to register R2 must be stable for at least t_{hold} after the clock edge.

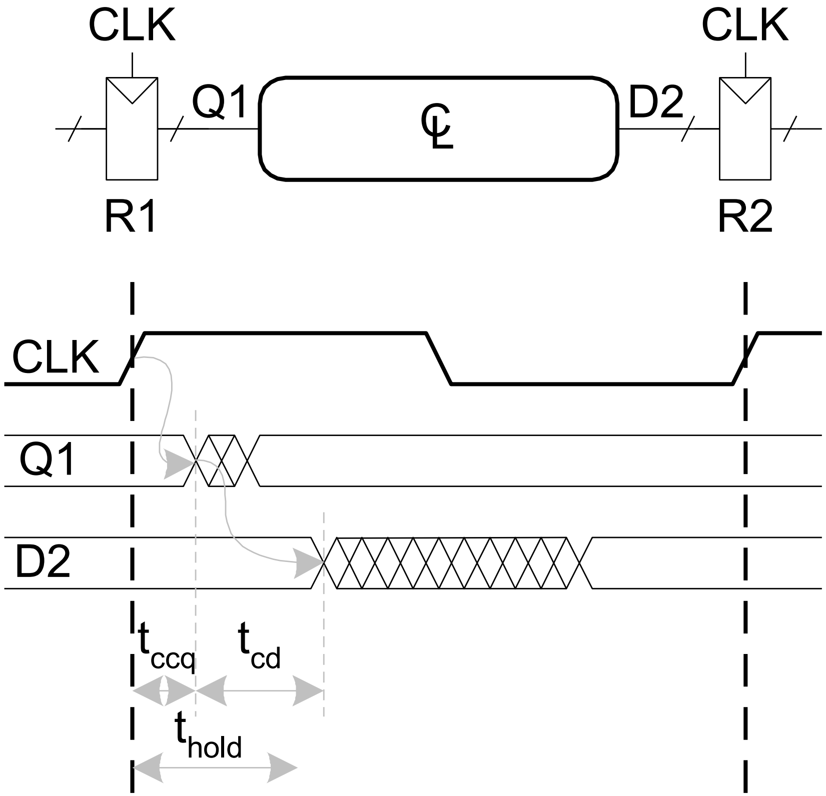


$t_{hold} <$

Hold Time Constraint

- **The hold time constraint depends on the minimum delay from register R1 through the combinational logic.**

- **The input to register R2 must be stable for at least t_{hold} after the clock edge.**



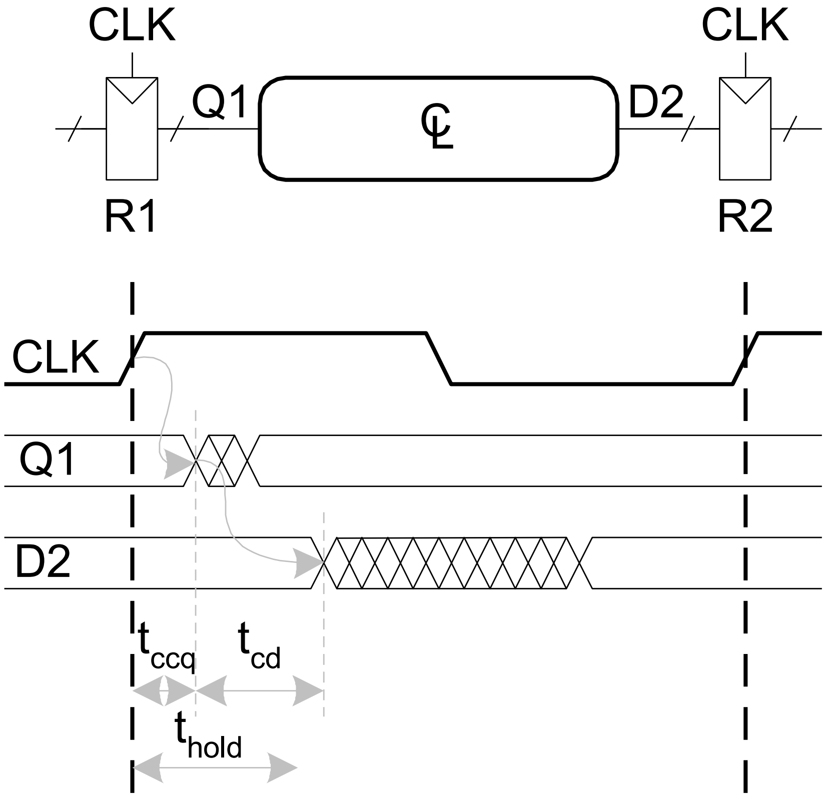
$$t_{hold} < t_{ccq} + t_{cd}$$

$$t_{cd} >$$

Hold Time Constraint

- **The hold time constraint depends on the minimum delay from register R1 through the combinational logic.**

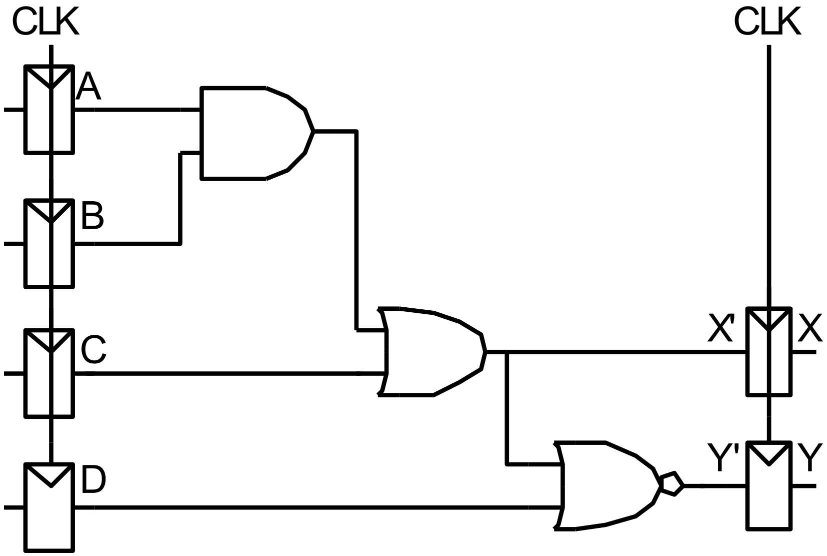
- **The input to register R2 must be stable for at least t_{hold} after the clock edge.**



$$t_{hold} < t_{ccq} + t_{cd}$$

$$t_{cd} > t_{hold} - t_{ccq}$$

Timing Analysis



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps

per gate

t_{pd}	= 35 ps
t_{cd}	= 25 ps

t_{pd} =

t_{cd} =

Setup time constraint:

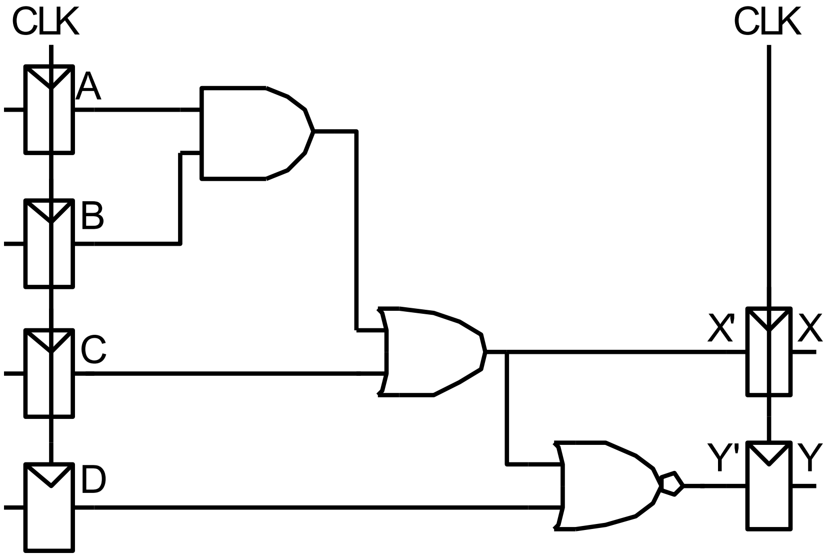
$T_c \geq$

$f_c = 1/T_c =$

Hold time constraint:

$t_{ccq} + t_{cd} > t_{hold}$?

Timing Analysis



Timing Characteristics

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

per gate

t_{pd} = 35 ps

t_{cd} = 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 25 ps

Setup time constraint:

$T_c \geq (50 + 105 + 60)$ ps = 215 ps

$f_c = 1/T_c$ = 4.65 GHz

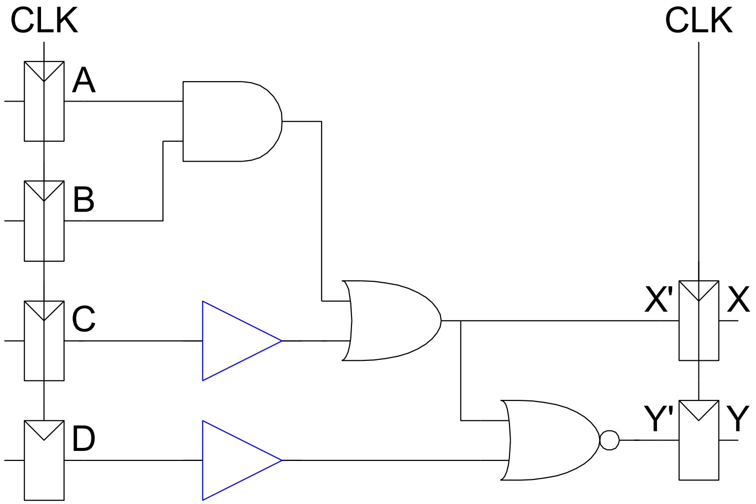
Hold time constraint:

$t_{ccq} + t_{cd} > t_{hold}$?

(30 + 25) ps > 70 ps ? **No!**

Fixing Hold Time Violation

Add buffers to the short paths:



t_{pd} =

t_{cd} =

Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps

per gate

t_{pd}	= 35 ps
t_{cd}	= 25 ps

Setup time constraint:

$T_c \geq$

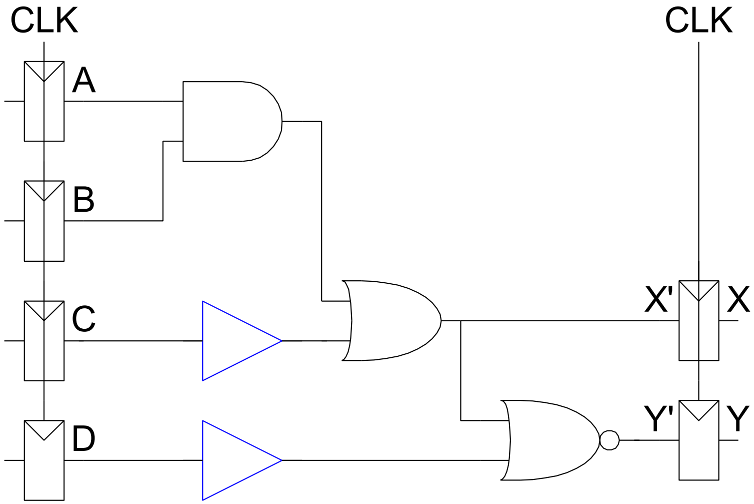
f_c =

Hold time constraint:

$t_{ccq} + t_{cd} > t_{hold}$?

Fixing Hold Time Violation

Add buffers to the short paths:



t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 2 x 25 ps = 50 ps

Setup time constraint:

$T_c \geq (50 + 105 + 60)$ ps = 215 ps

$f_c = 1/T_c$ = 4.65 GHz

Timing Characteristics

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

per gate

t_{pd} = 35 ps

t_{cd} = 25 ps

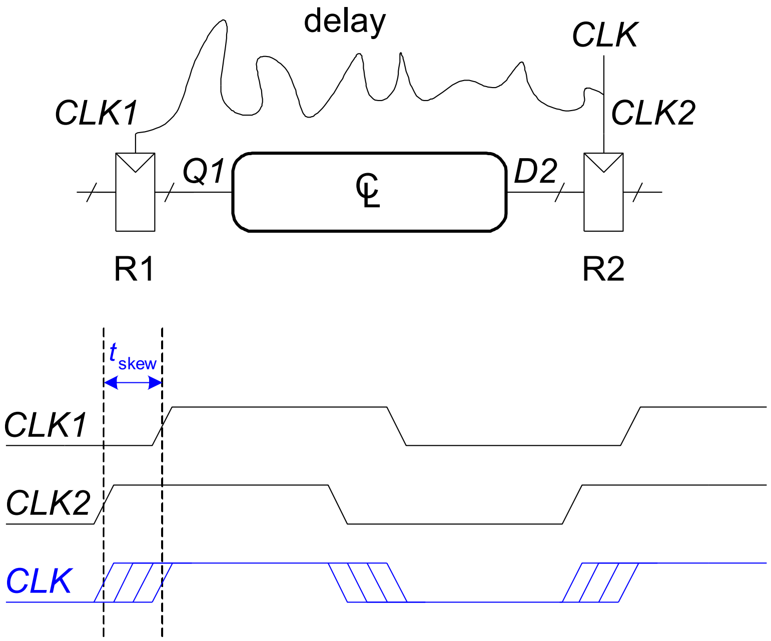
Hold time constraint:

$t_{ccq} + t_{cd} > t_{hold}$?

(30 + 50) ps > 70 ps ? **Yes!**

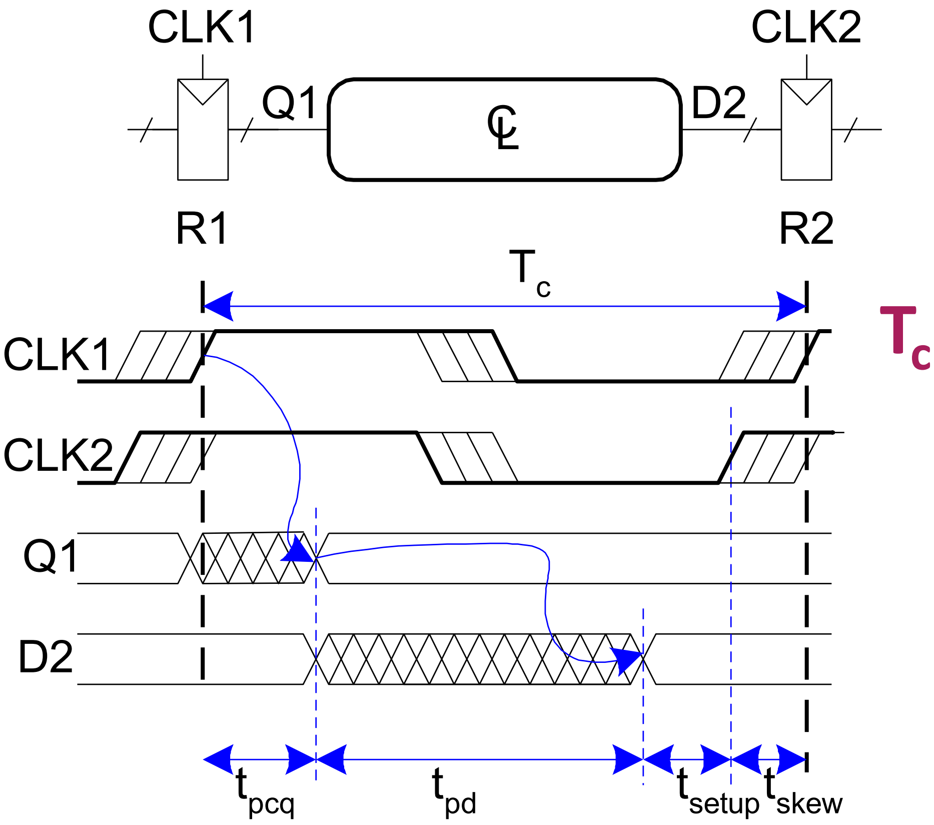
Clock Skew

- The clock doesn't arrive at all registers at the same time
- Skew is the difference between two clock edges
- Examine the worst case to guarantee that the dynamic discipline is not violated for any register – many registers in a system!



Setup Time Constraint with Clock Skew

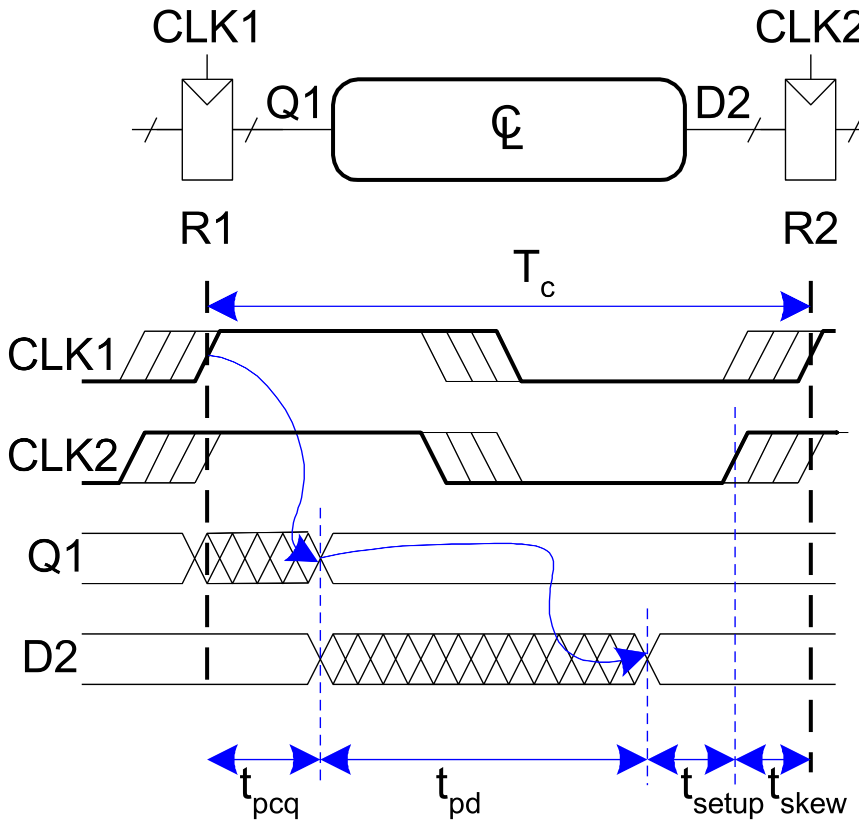
- In the worst case, the CLK2 is earlier than CLK1



$T_c >=$

Setup Time Constraint with Clock Skew

- In the worst case, the CLK2 is earlier than CLK1

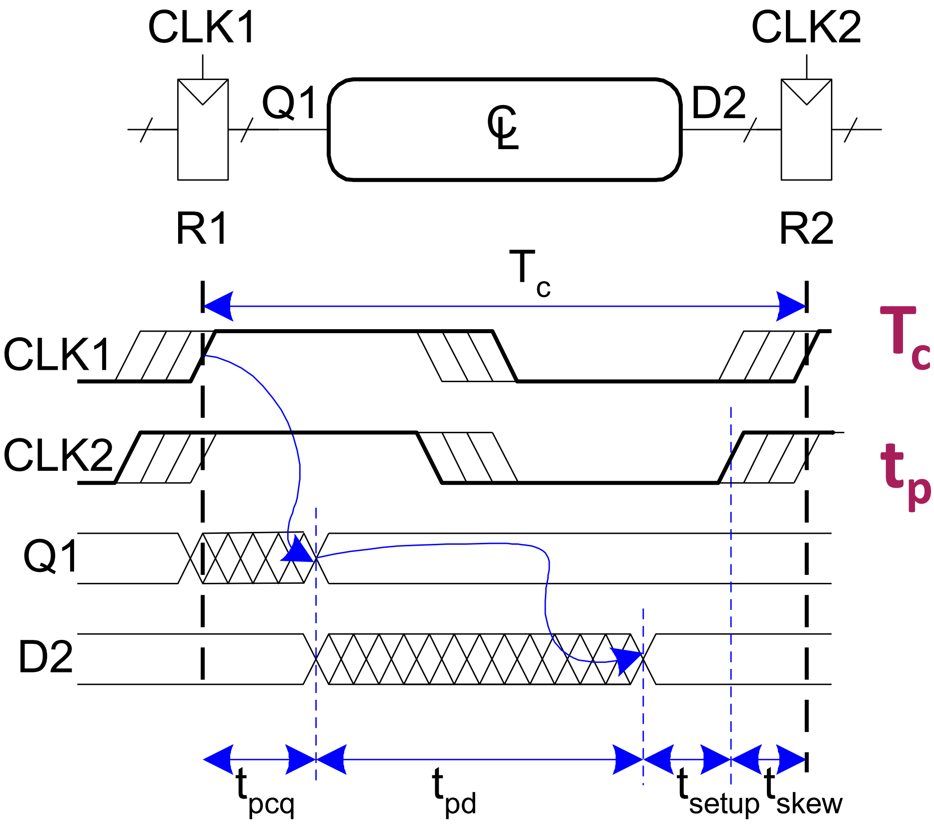


$T_c >= t_{pcq} + t_{pd} + t_{setup} + t_{skew}$

$t_{pd} <=$

Setup Time Constraint with Clock Skew

- In the worst case, the CLK2 is earlier than CLK1

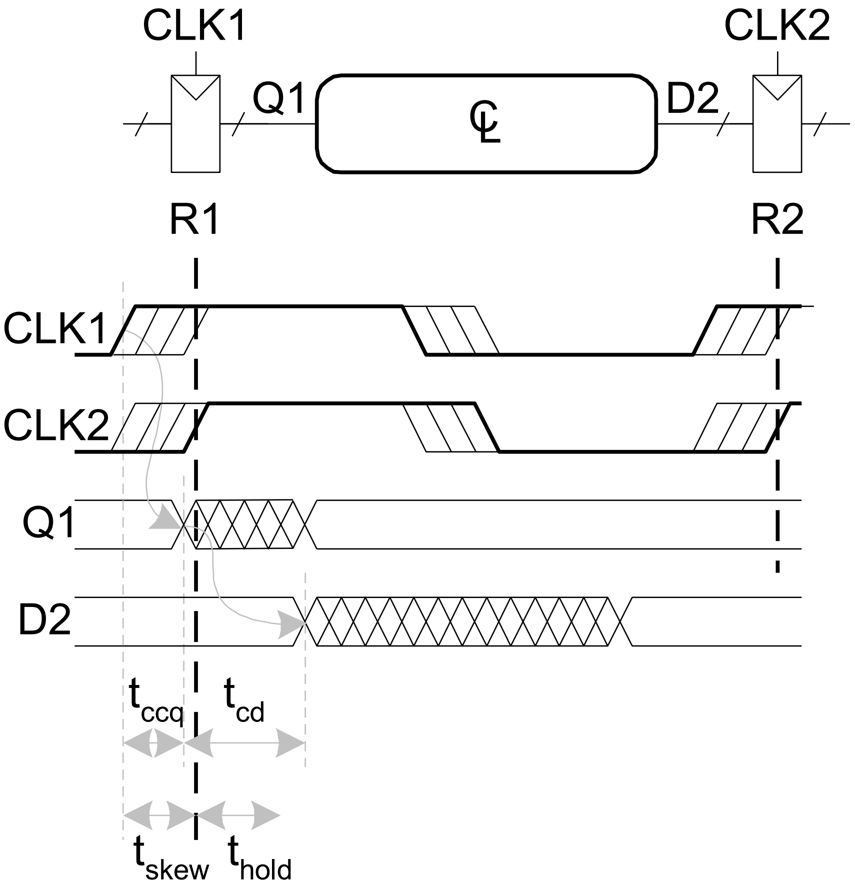


$T_c >= t_{pcq} + t_{pd} + t_{setup} + t_{skew}$

$t_{pd} <= T_c – (t_{pcq} + t_{setup} + t_{skew})$

Hold Time Constraint with Clock Skew

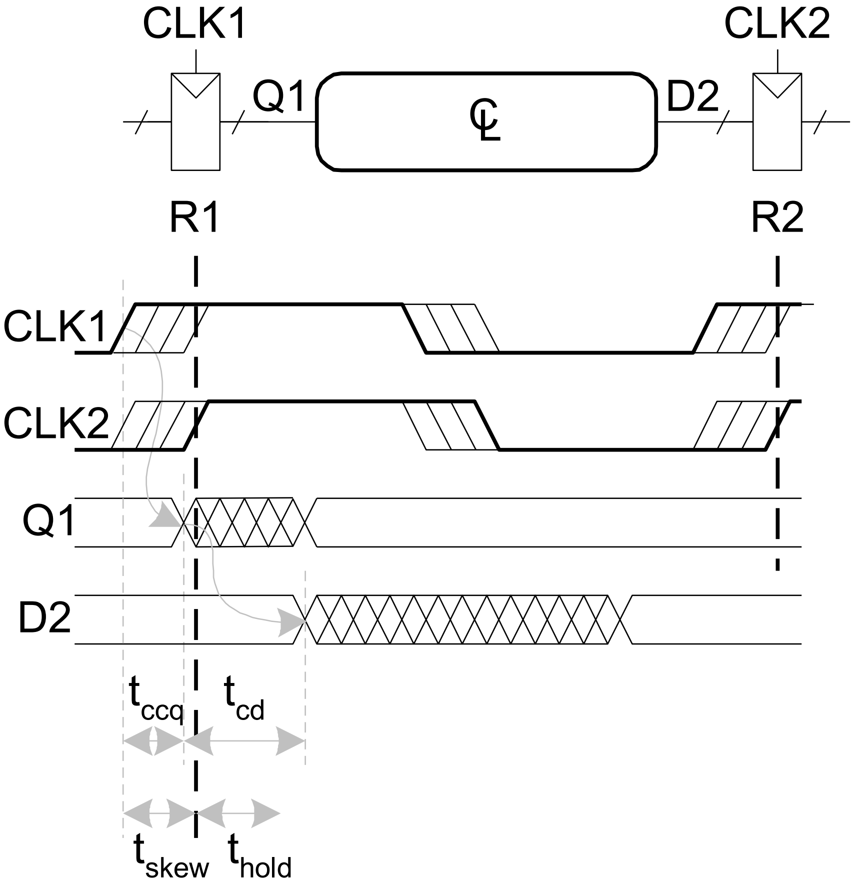
- In the worst case, CLK2 is later than CLK1



$t_{hold} + t_{skew}$ =<

Hold Time Constraint with Clock Skew

- In the worst case, CLK2 is later than CLK1

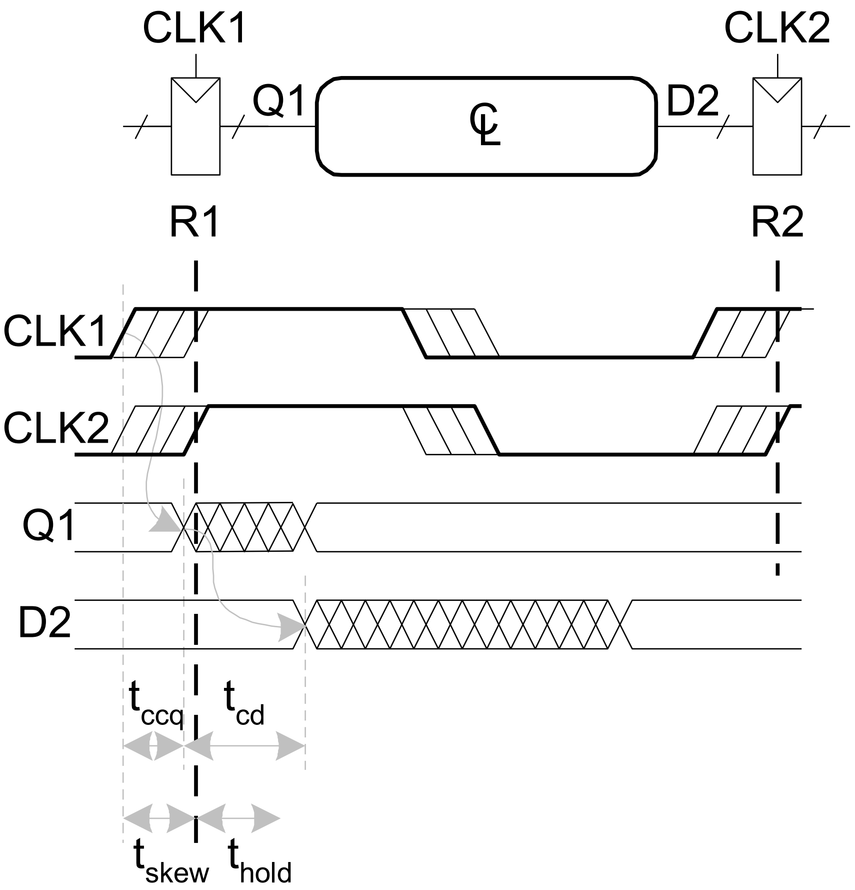


$t_{hold} + t_{skew} =< t_{ccq} + t_{cd}$

$t_{cd} >=$

Hold Time Constraint with Clock Skew

- In the worst case, CLK2 is later than CLK1

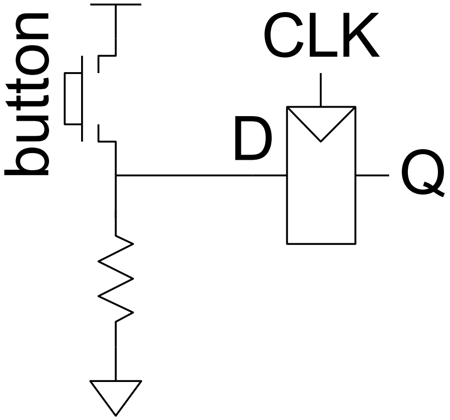


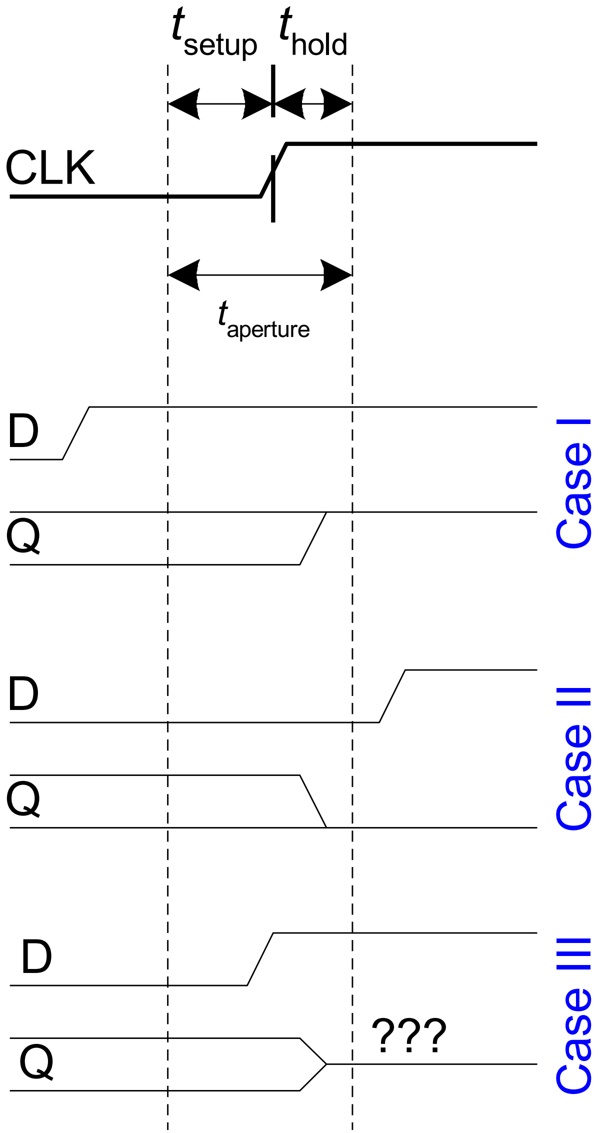
$$t_{hold} + t_{skew} =< t_{ccq} + t_{cd}$$

$$t_{cd} >= t_{hold} + t_{skew} - t_{ccq}$$

Violating the Dynamic Discipline

- **Asynchronous (for example, user) inputs might violate the dynamic discipline**



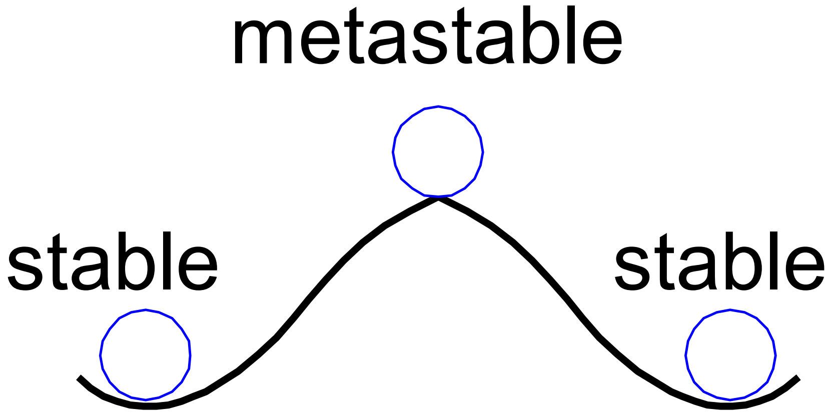


Metastability

- **Any bi-stable device has two stable states and a metastable state between them**

- **A flip-flop has two stable states (1 and 0) and one metastable state**

- **If a flip-flop lands in the metastable state, it could stay there for an undetermined amount of time**



Metastability

- T_0/T_c describes the probability that the input changes at a bad time, i.e., during the aperture time

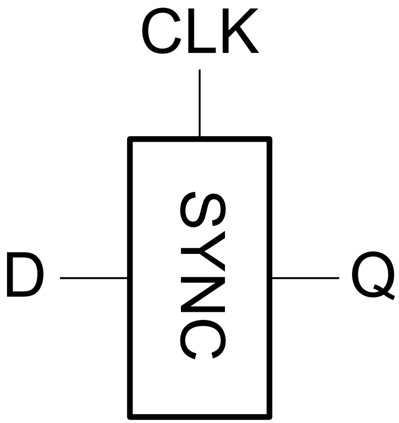
$$P(t_{res} > t) = (T_0/T_c\,)\, e^{-t/\tau}$$

- τ is a time constant indicating how fast the flip-flop moves away from the metastable state; it is related to the delay through the cross-coupled gates in the flip-flop

- In short, if a flip-flop samples a metastable input, if you wait long enough (t), the output will have resolved to 1 or 0 with high probability.

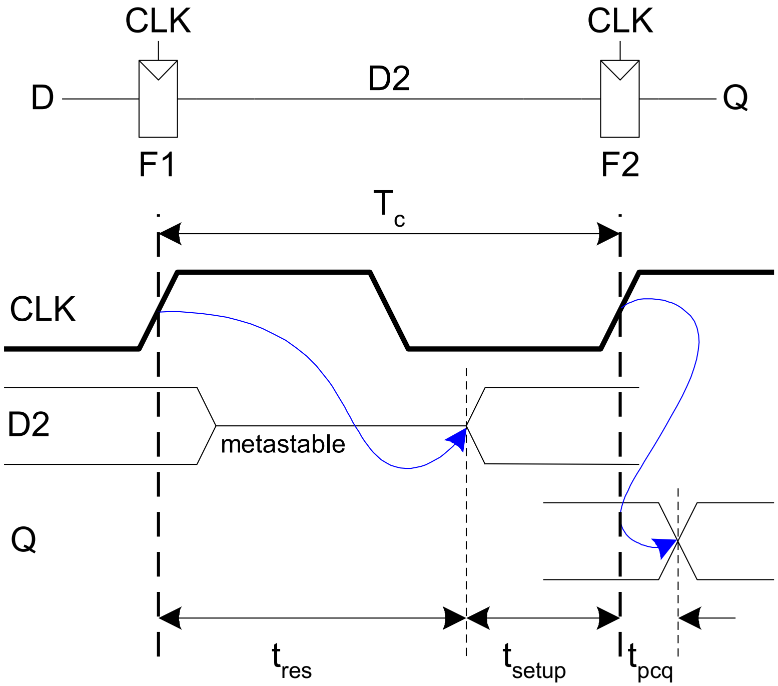
Synchronizers

- Asynchronous inputs (D) are inevitable (user interfaces, systems with different clocks interacting, etc.).
- The goal of a synchronizer is to make the probability of failure (the output Q still being metastable) low.
- A synchronizer cannot make the probability of failure 0.



Synchronizer Internals

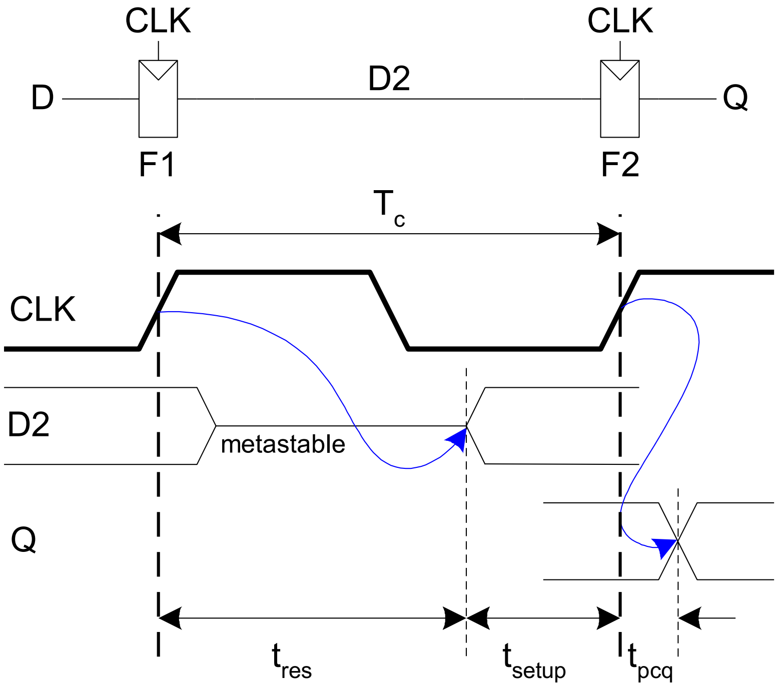
- A synchronizer can be built with two back-to-back flip-flops.
- Suppose the input D is changing when it is sampled by F1.
- Internal signal D2 has (T_c - t_{setup}) time to resolve a 1 or 0.



Synchronizer Probability of Failure

- **For each sample, the probability of failure of this synchronizer is:**

$$P(\text{failure}) = \frac{T_0}{T_c} e^{-\frac{T_c - t_{setup}}{\tau}}$$



Synchronizer Mean Time Before Failure

- If the asynchronous input changes once per second, the probability of failure per second of the synchronizer is simply P(failure).

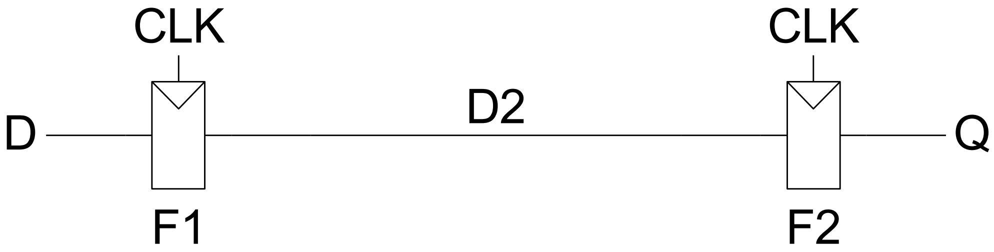
- In general, if the input changes N times per second, the probability of failure per second of the synchronizer is:

$$P(\text{failure})/\text{sec} = N\frac{T_0}{T_c}e^{-\frac{T_c - t_{\text{setup}}}{\tau}}$$

- Thus, the synchronizer fails, on average,1/[P(failure)/second] This is called the mean time between failures, MTBF:

$$MTBF = \frac{1}{P(\text{failure})/\text{sec}} = \frac{T_c\, e^{\frac{T_c - t_{\text{setup}}}{\tau}}}{NT_0}$$

Example Synchronizer



Suppose: T_c = 1/500 MHz = 2 ns   τ = 200 ps

T_0 = 150 ps t_{setup} = 100 ps

N = 10 events per second

What is the probability of failure? MTBF?

P(failure) =

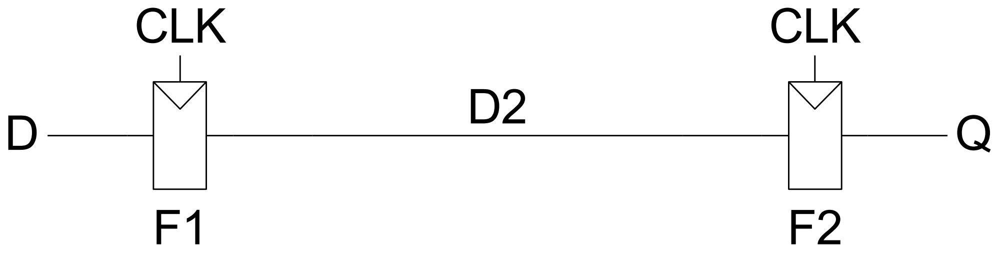
=

P(failure)/second =

=

MTBF =

Example Synchronizer



Suppose: T_c = 1/500 MHz = 2 ns　　τ = 200 ps

T_0 = 150 ps　　t_{setup} = 100 ps

N = 10 events per second

What is the probability of failure? MTBF?

P(failure) = (150 ps/2 ns) $e^{-(1.9\text{ ns})/200\text{ ps}}$

= 5.6×10^{-6}

P(failure)/second = $10 \times (5.6 \times 10^{-6})$

= 5.6×10^{-5} / second

MTBF = 1/[P(failure)/second] ≈ 5 hours

What did we learn ?

- **Timing constraints for sequential circuits**
  - ***Setup time***, time needed for the inputs to be present before clock
  - ***Hold time***, time needed after the clock where inputs should not change
  - ***Aperture time***, time around clock event where inputs should stay stable

- **Problems related to clock skew**

- **Metastability and Synchronization**