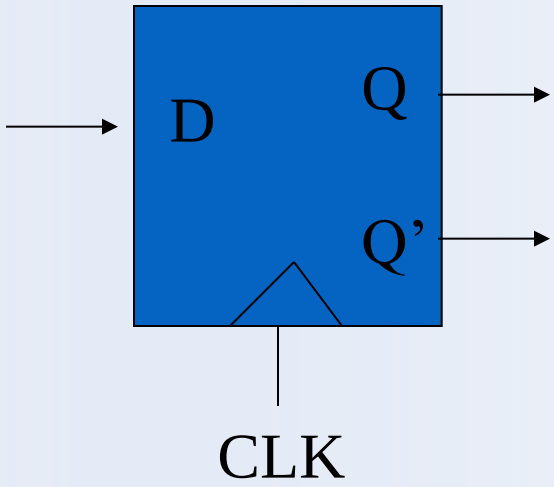
Digital Logic Design

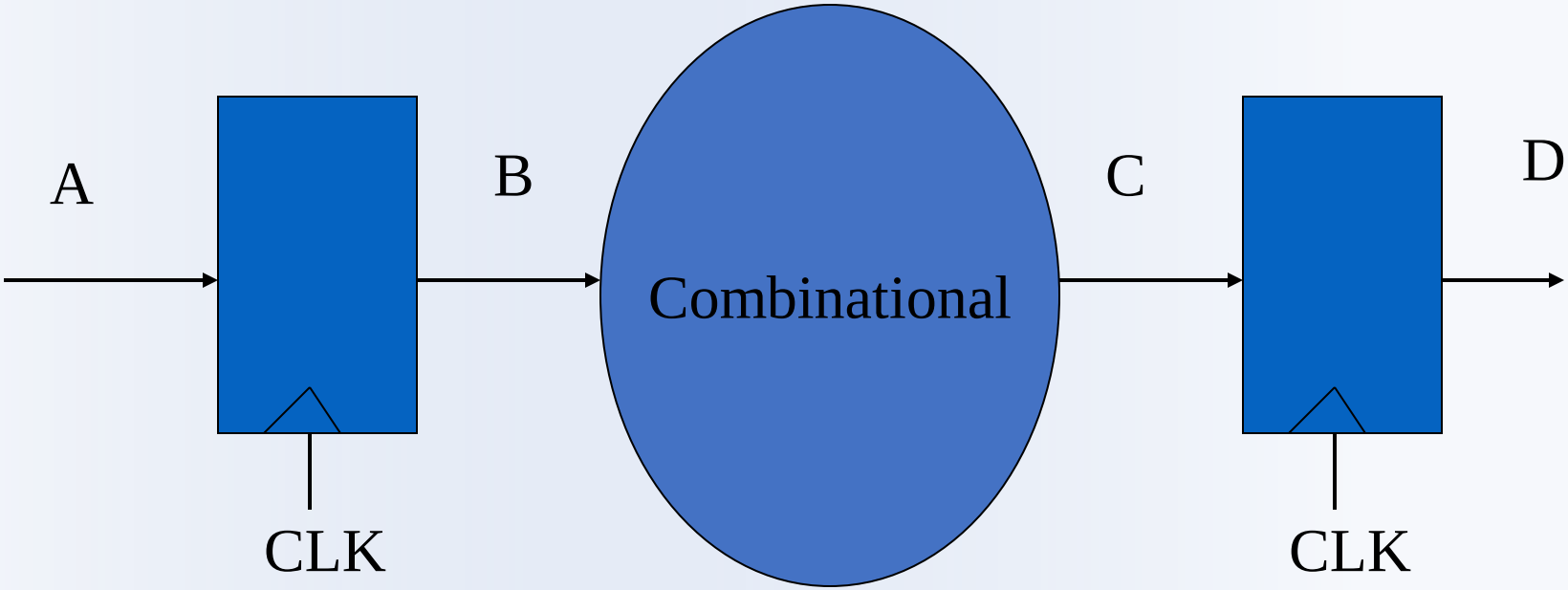
Timing in Sequential Circuits

Sequential Networks

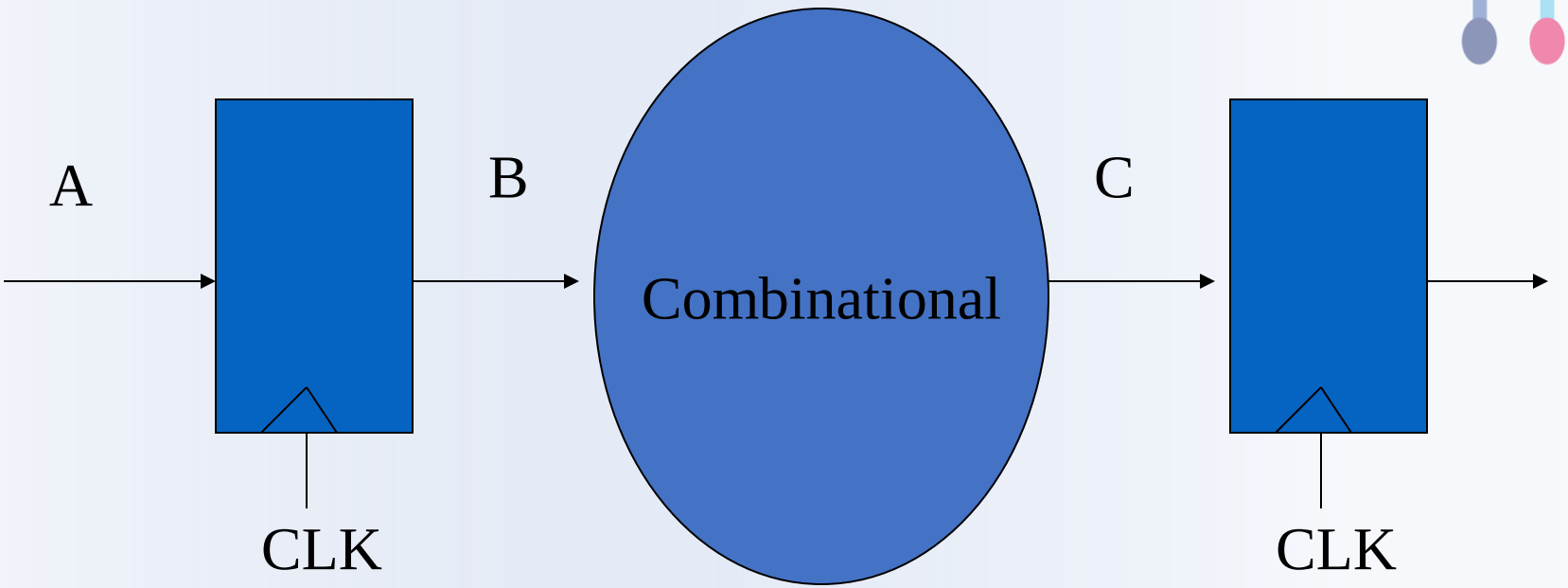
Timing: Setup Time and Hold Time Constraints



Sequential Networks



A typical sequential network has both a combinational circuit and flip-flips.



$$t_{cq} + t_{comb} + t_{setup} \leq T$$

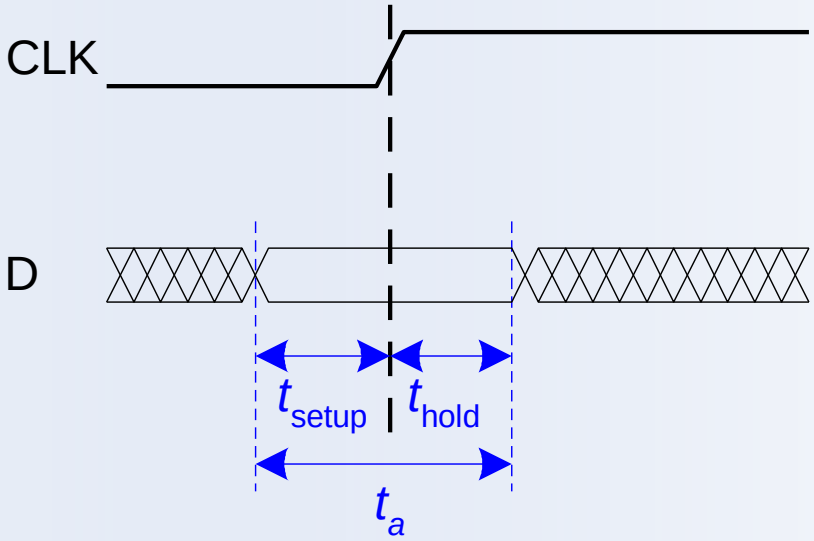
Clock period

$$t_{hold} < t_{cq} + t_{comb}$$

Shortest path

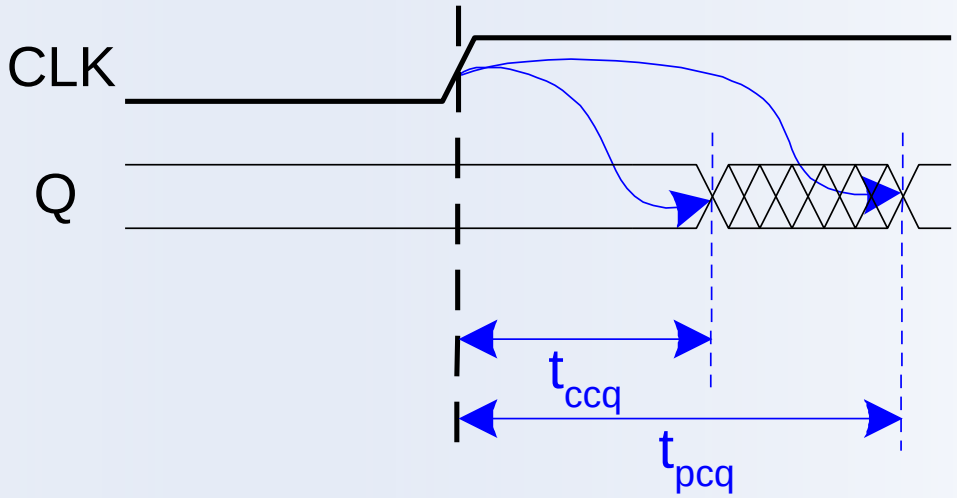
Input Timing Constraints

- 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 ($t_a = t_{setup} + t_{hold}$)



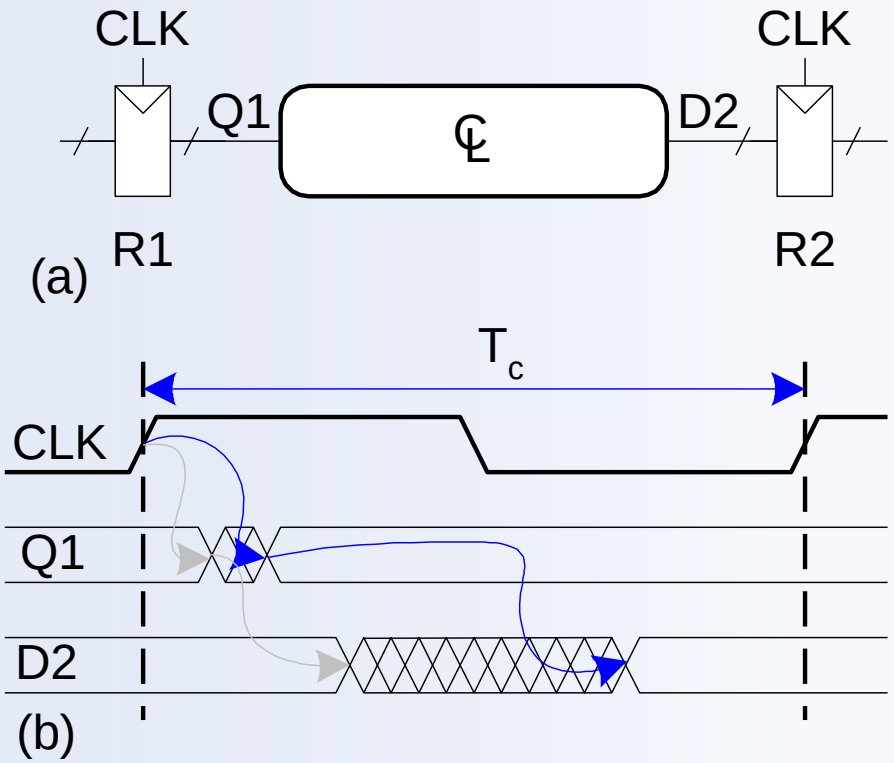
Output Timing Constraints

- 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)



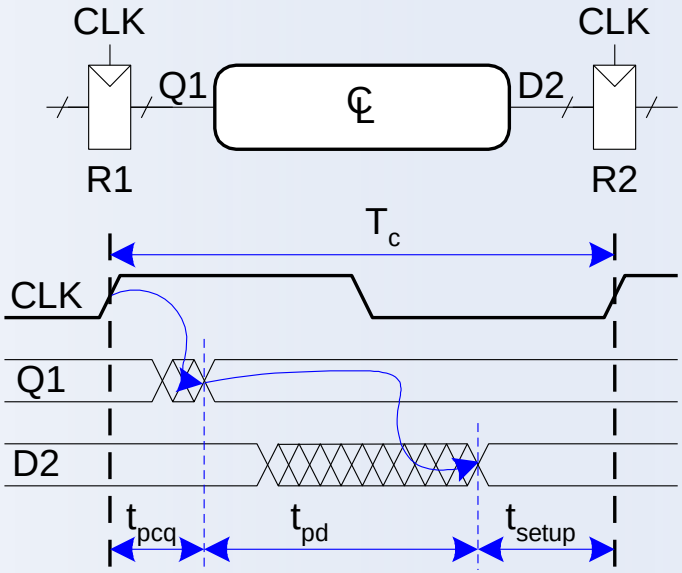
Dynamic Discipline

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



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.

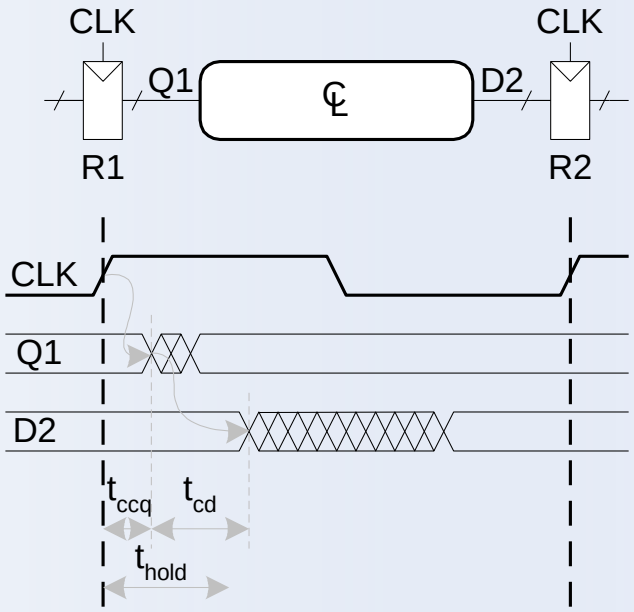


$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

$$t_{pd} \leq T_c - (t_{pcq} + t_{setup})$$

Hold Time Constraint

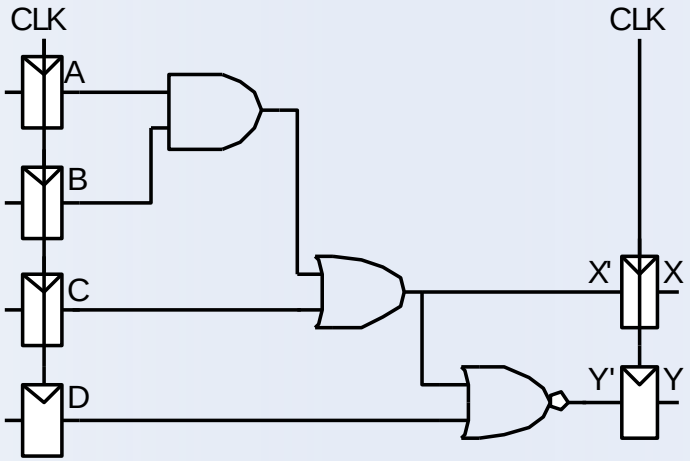
- The hold time constraint depends on the **minimum** delay from register R1 through the combinational logic.
- The input to register R2 must be stable for at least t_{hold} after the clock edge.



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

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

Timing Analysis



t_{pd} =

t_{cd} =

Setup time constraint:

$T_c \geq$

$f_c = 1/T_c =$

Timing Characteristics

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

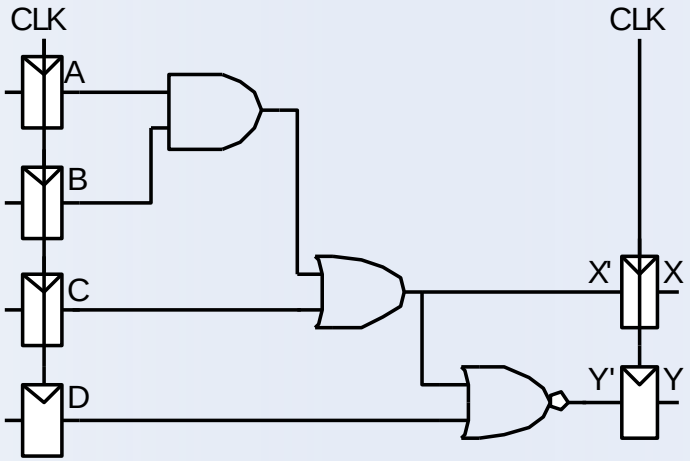
t_{pd} = 35 ps

t_{cd} = 25 ps

Hold time constraint:

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

Timing Analysis



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

Timing Characteristics

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

t_{pd} = 35 ps

t_{cd} = 25 ps

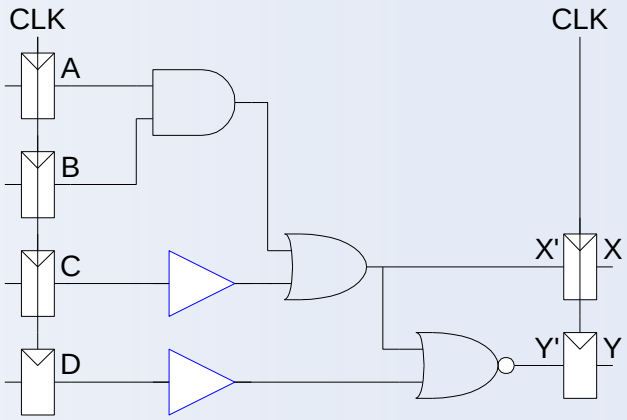
Hold time constraint:

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

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

Fixing Hold Time Violation

Add buffers to the short paths:



t_{pd} =

t_{cd} =

Setup time constraint:

$T_c \geq$

f_c =

Timing Characteristics

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

t_{pd} = 35 ps

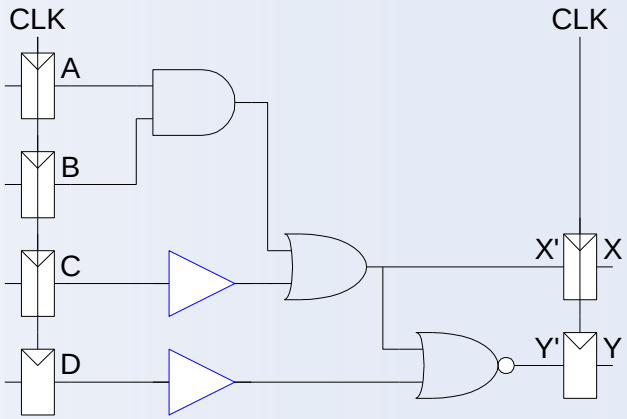
t_{cd} = 25 ps

Hold time constraint:

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

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

t_{pd} = 35 ps

t_{cd} = 25 ps

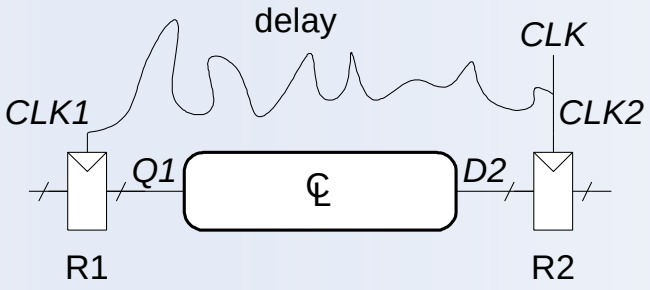
Hold time constraint:

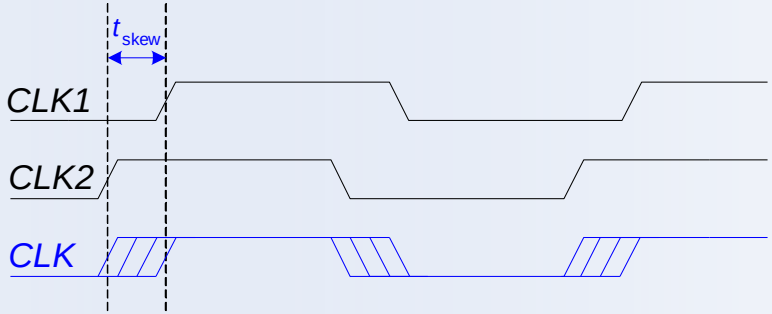
$t_{ccq} + t_{pd} > t_{hold}$?

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

Clock Skew

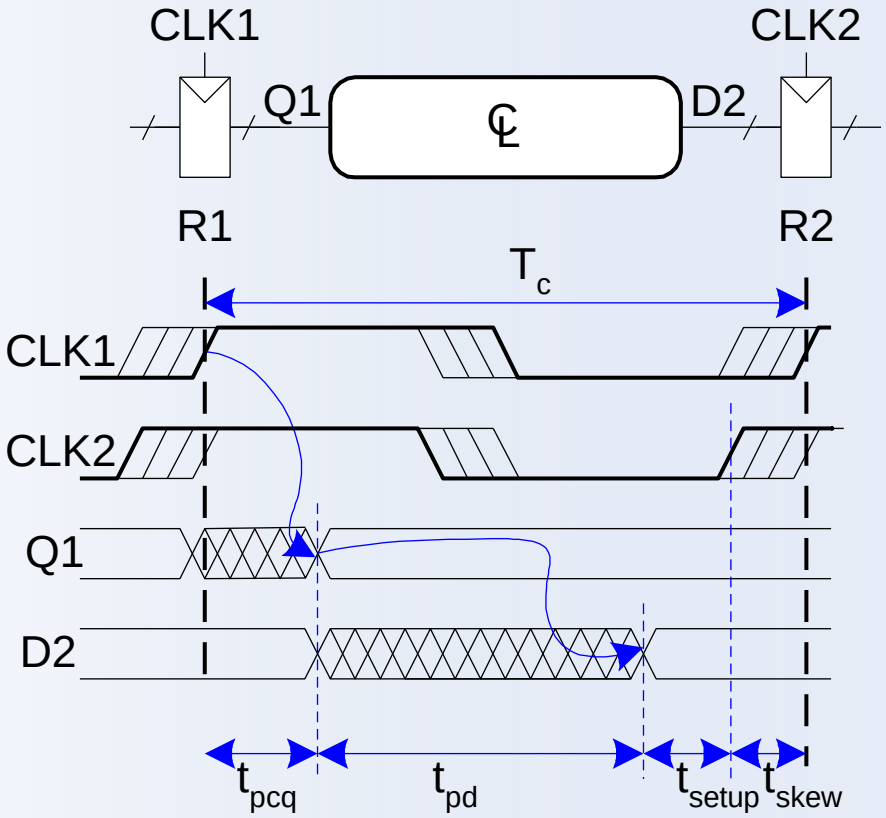
- The clock doesn't arrive at all registers at the same time
- Skew is the difference between two clock edges
- Examine the worst case to guarantee that the dynamic discipline is not violated for any register – many registers in a system!





Setup Time Constraint with Clock Skew

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

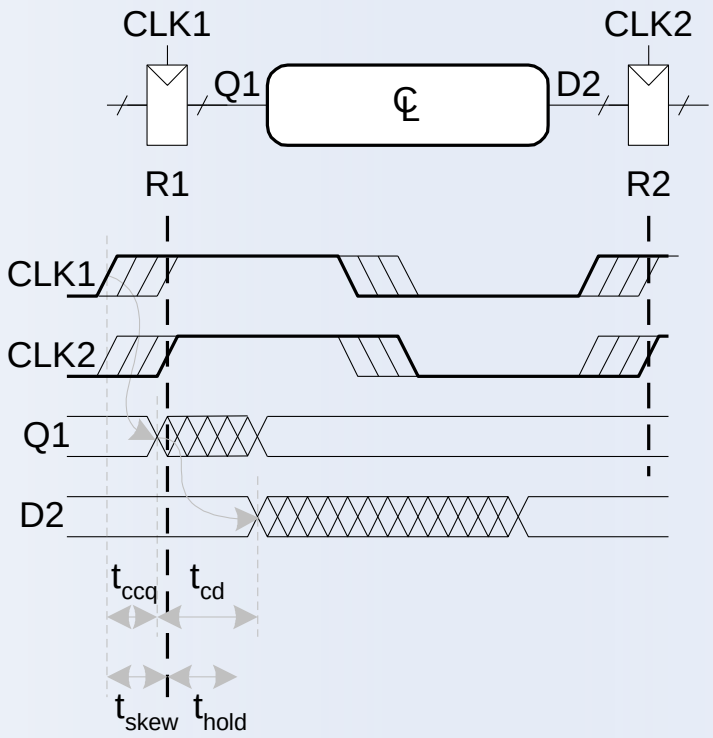


$$T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}} + t_{\text{skew}}$$

$$t_{pd} \leq T_c - (t_{pcq} + t_{\text{setup}} + t_{\text{skew}})$$

Hold Time Constraint with Clock Skew

- In the worst case, CLK2 is later than CLK1



$$t_{ccq} + t_{cd} > t_{\text{hold}} + t_{\text{skew}}$$

$$t_{cd} > t_{\text{hold}} + t_{\text{skew}} - t_{ccq}$$

Timing and Retiming

- Retiming: Adjust the clock skew so that the clock period can be reduced.
- Add a few more examples on timing and retiming.

Fixing Setup Time Violation

1. Increase clock period
2. Optimize datapath
    a. Pipeline
    b. Logic optimization
    c. Buffer removal
    d. Routing optimization
3. Use faster cells
4. Relax constraints for non-critical paths

Fixing Hold Time Violations

1. Add delay to datapath
    a. Insert buffers
    b. Interconnect delay
2. Reduce cock skew
    a. Clock tree optimization
3. Increase hold time requirements
    a. Use of retiming buffers
        i. Retiming involves repositioning registers in the circuit to balance the timing constraints. By shifting flip-flops, it's possible to meet the required hold time without impacting overall circuit performance.
    b. Insert hold buffers
4. Adjust operating conditions
    a. Lowering voltage
    b. Temperature adjustment
5. Change Flip-flop type