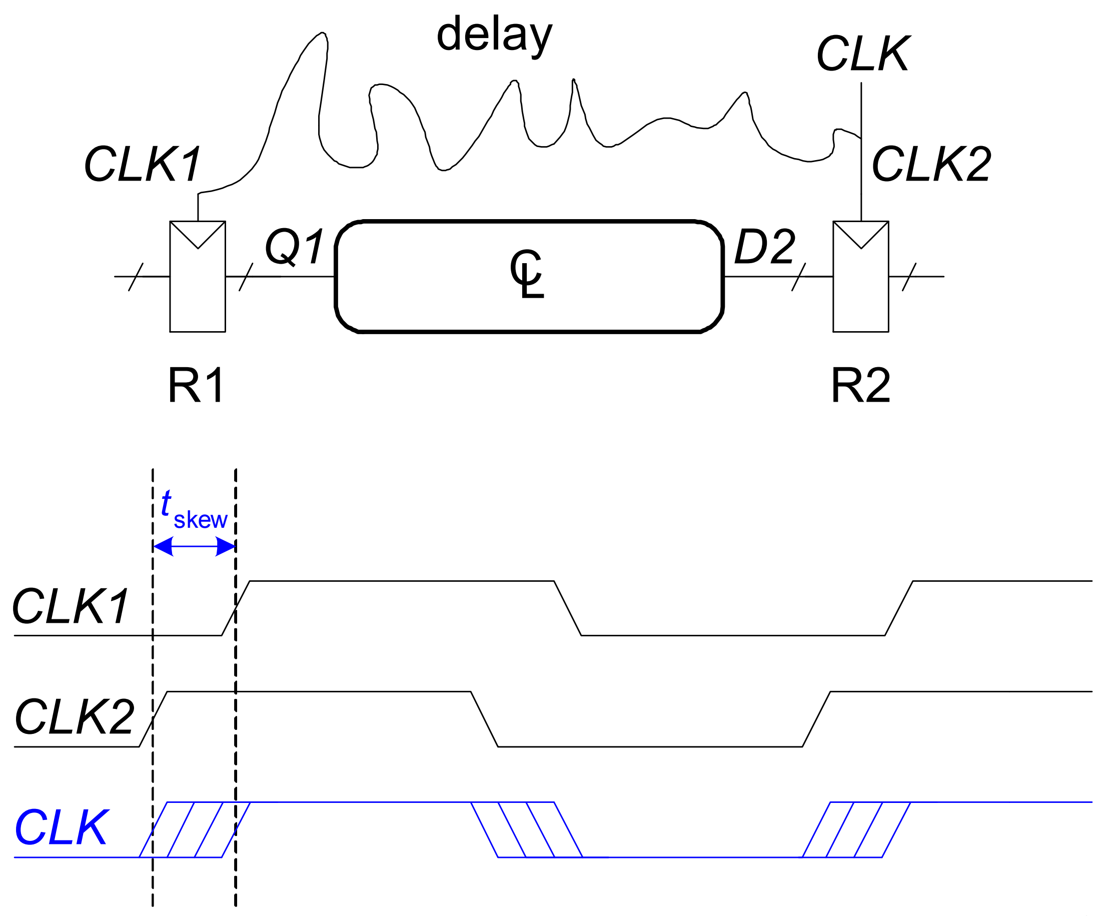
EECE 2322: Fundamentals of Digital Design and Computer Organization

Lecture 12_2: Timing of Digital Systems

Xiaolin Xu
Department of ECE
Northeastern University

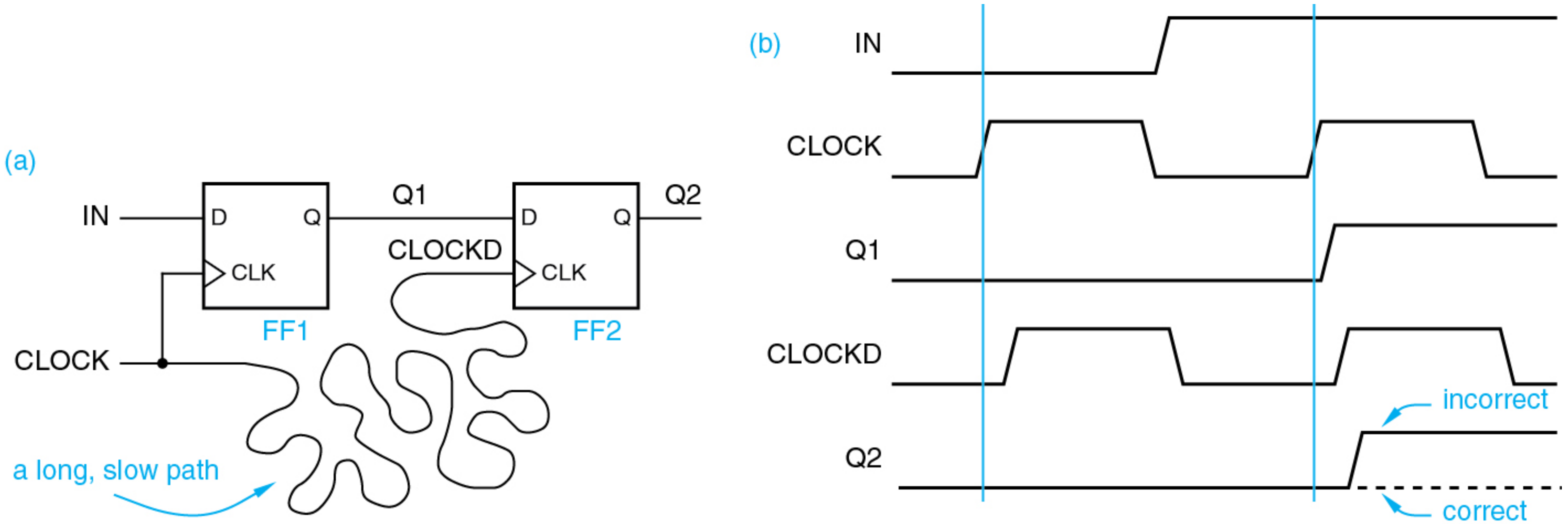
Clock Skew

- The clock doesn't arrive at all registers at same time
- **Skew:** difference between two clock edges
- Perform **worst case analysis**
  - Guarantee dynamic discipline is not violated for any register – many registers in a system!



Difficulties in Synchronous Design

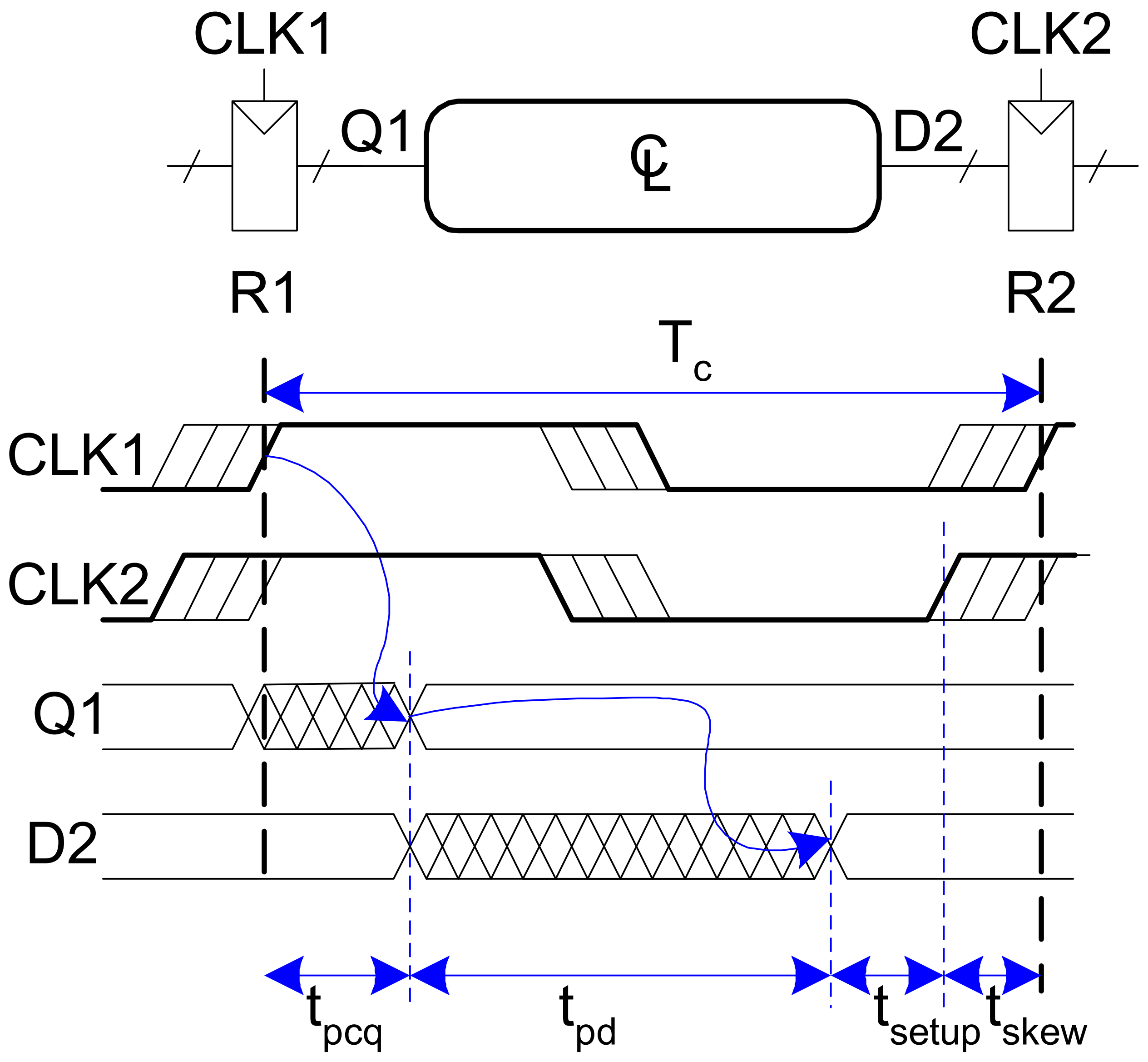
- Clock-Skew Example



$$t_{\text{ffpd(min)}} + t_{\text{comb(min)}} - t_{\text{hold}} - t_{\text{skew(max)}} > 0$$

Setup Time Constraint with Skew

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

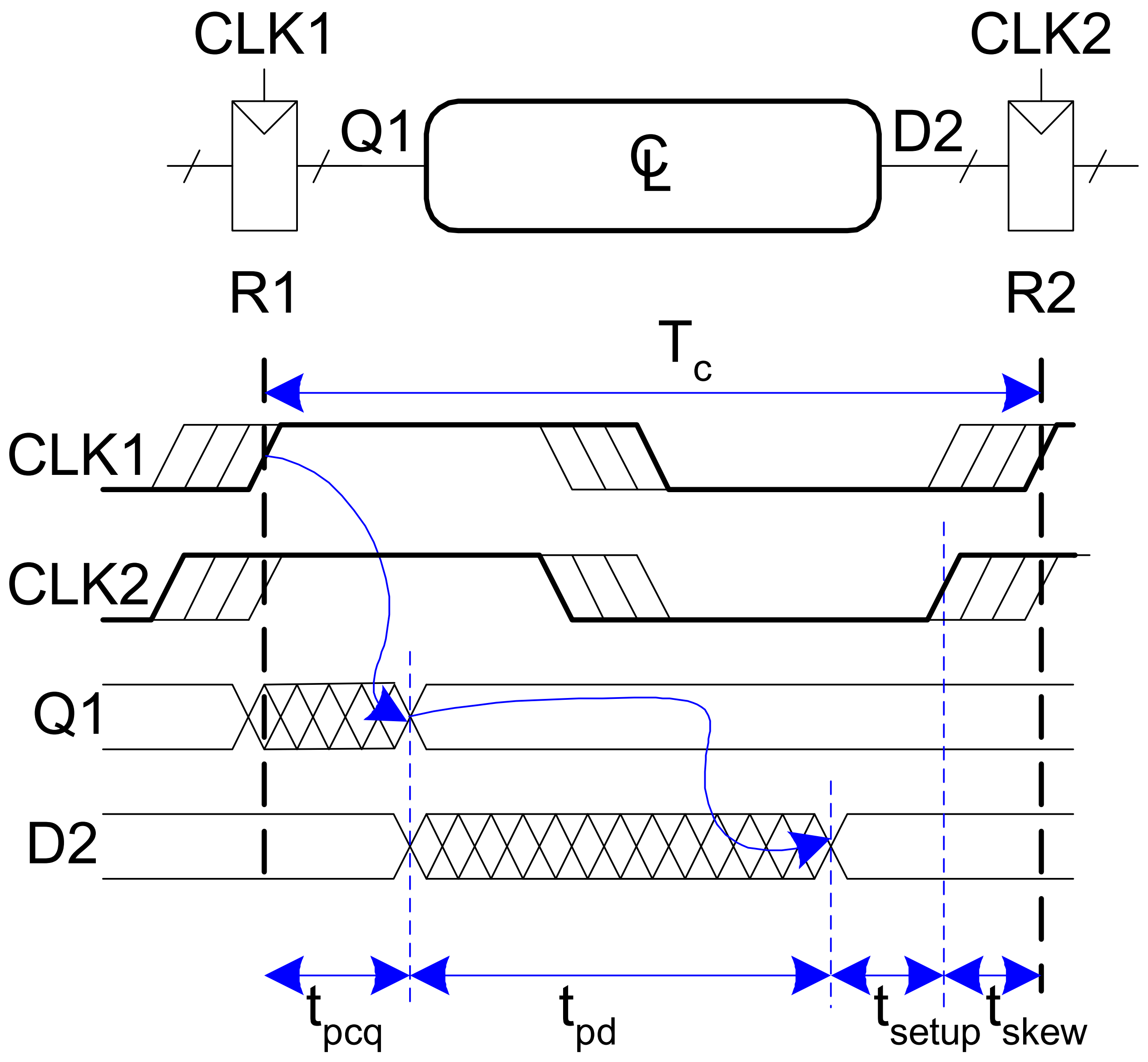


$T_c \geq$

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

Setup Time Constraint with Skew

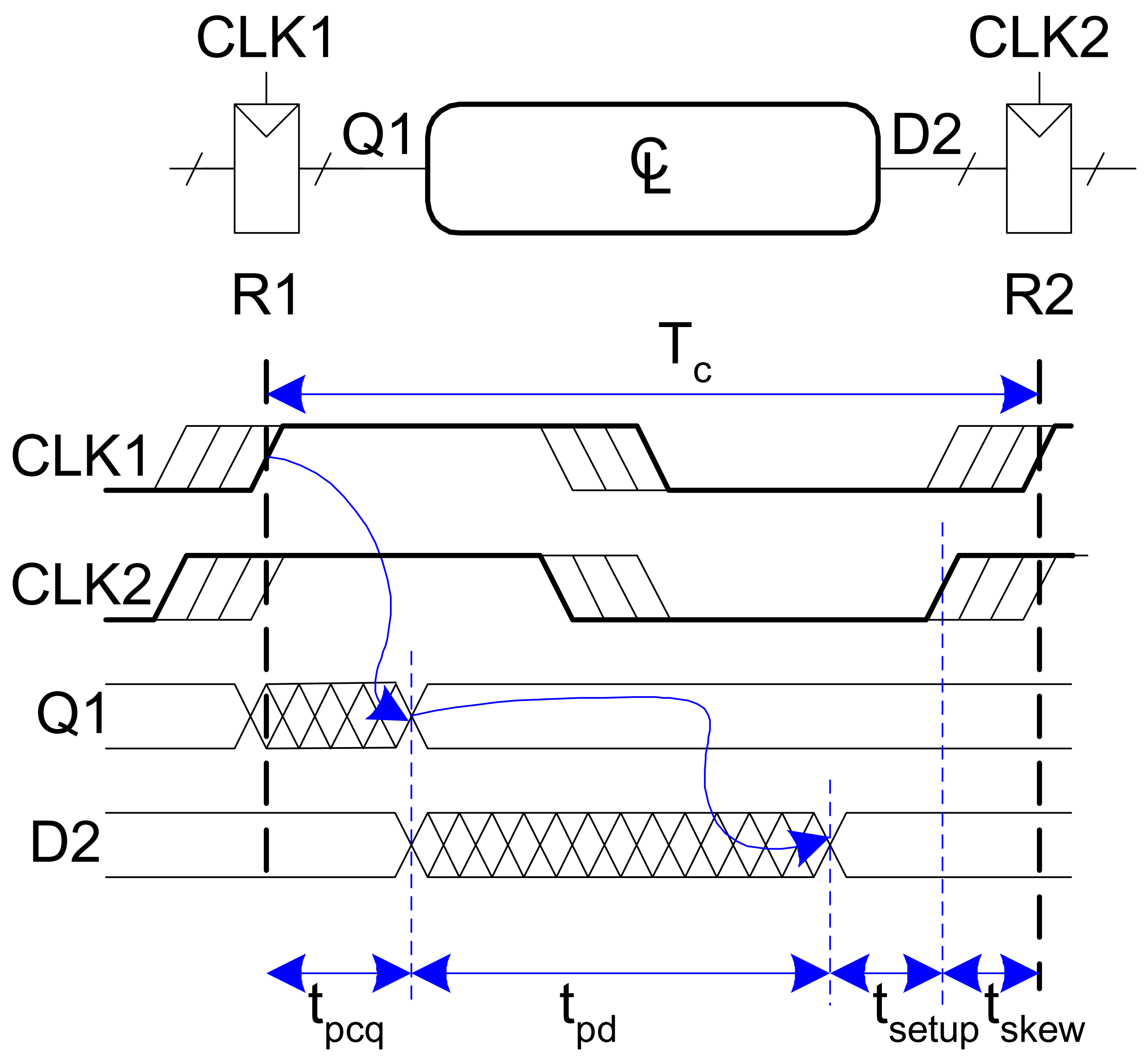
- In **the worst case**, CLK2 is earlier than CLK1



$T_c \geq$

Setup Time Constraint with Skew

- In the worst case, CLK2 is earlier than CLK1



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

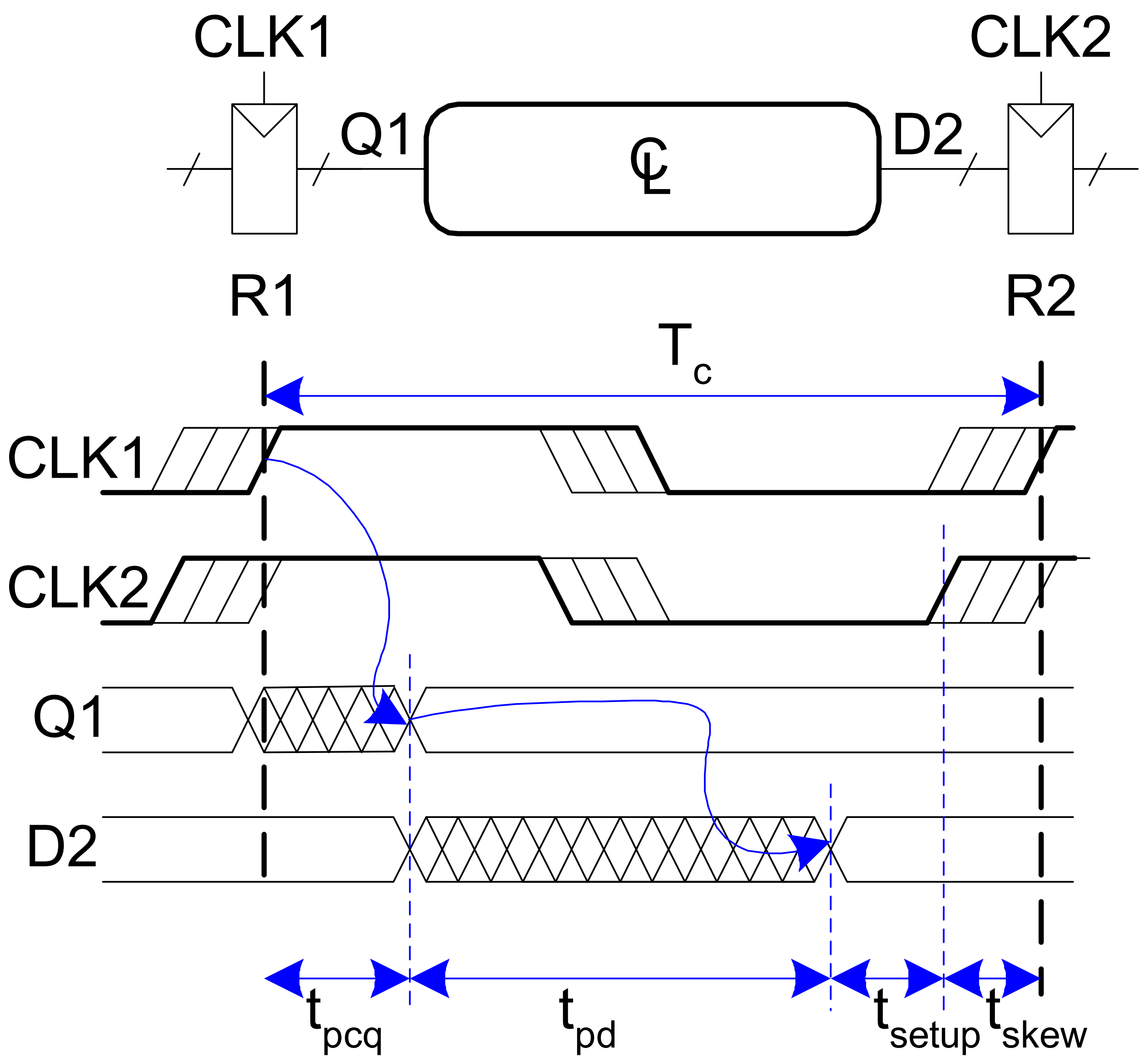
$$t_{pd} \leq$$

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

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

Setup Time Constraint with Skew

- In the worst case, CLK2 is earlier than CLK1

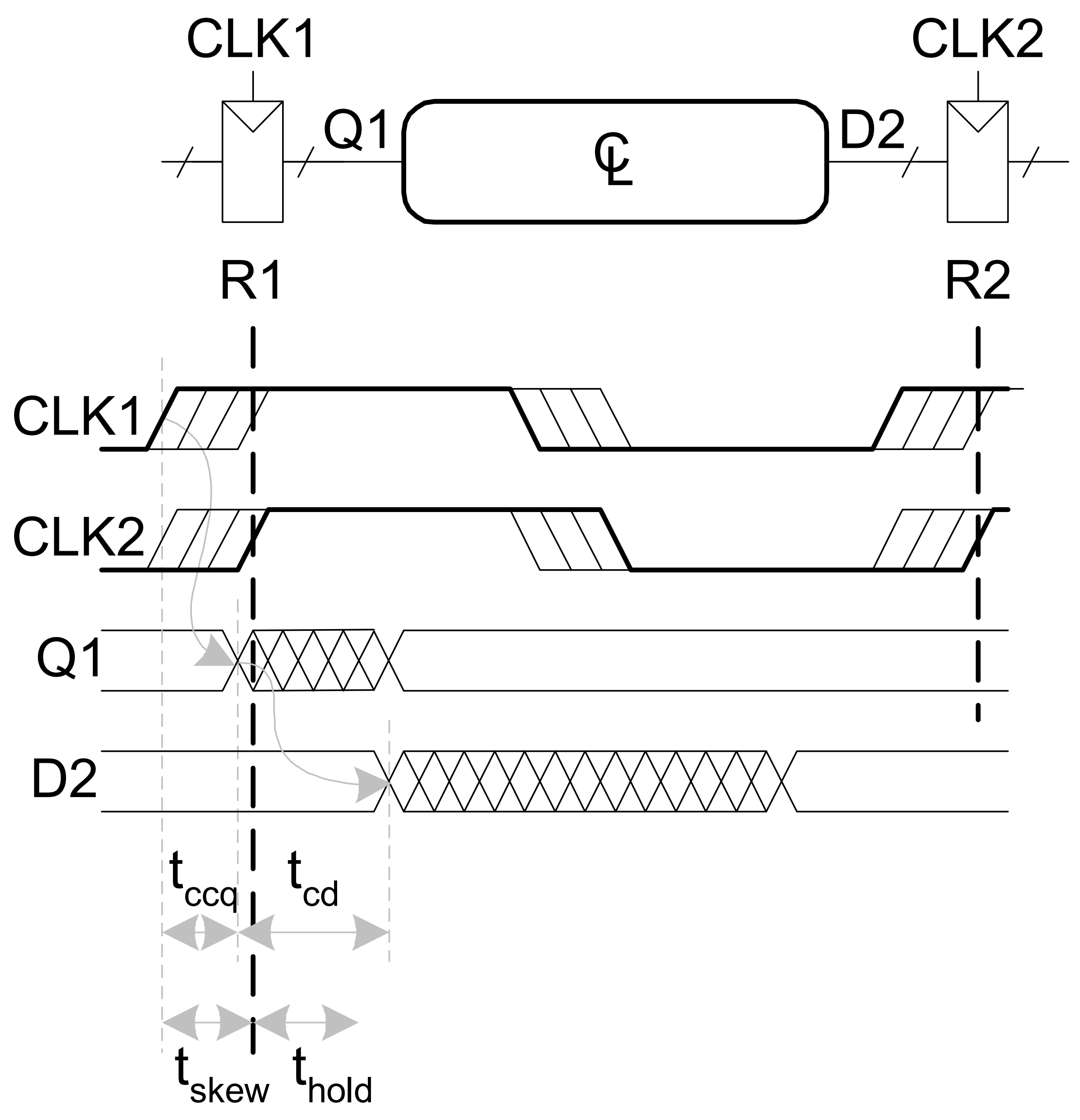


$$T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew}$$
$$t_{pd} \leq T_c - (t_{pcq} + t_{setup} + t_{skew})$$

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

Hold Time Constraint with Skew

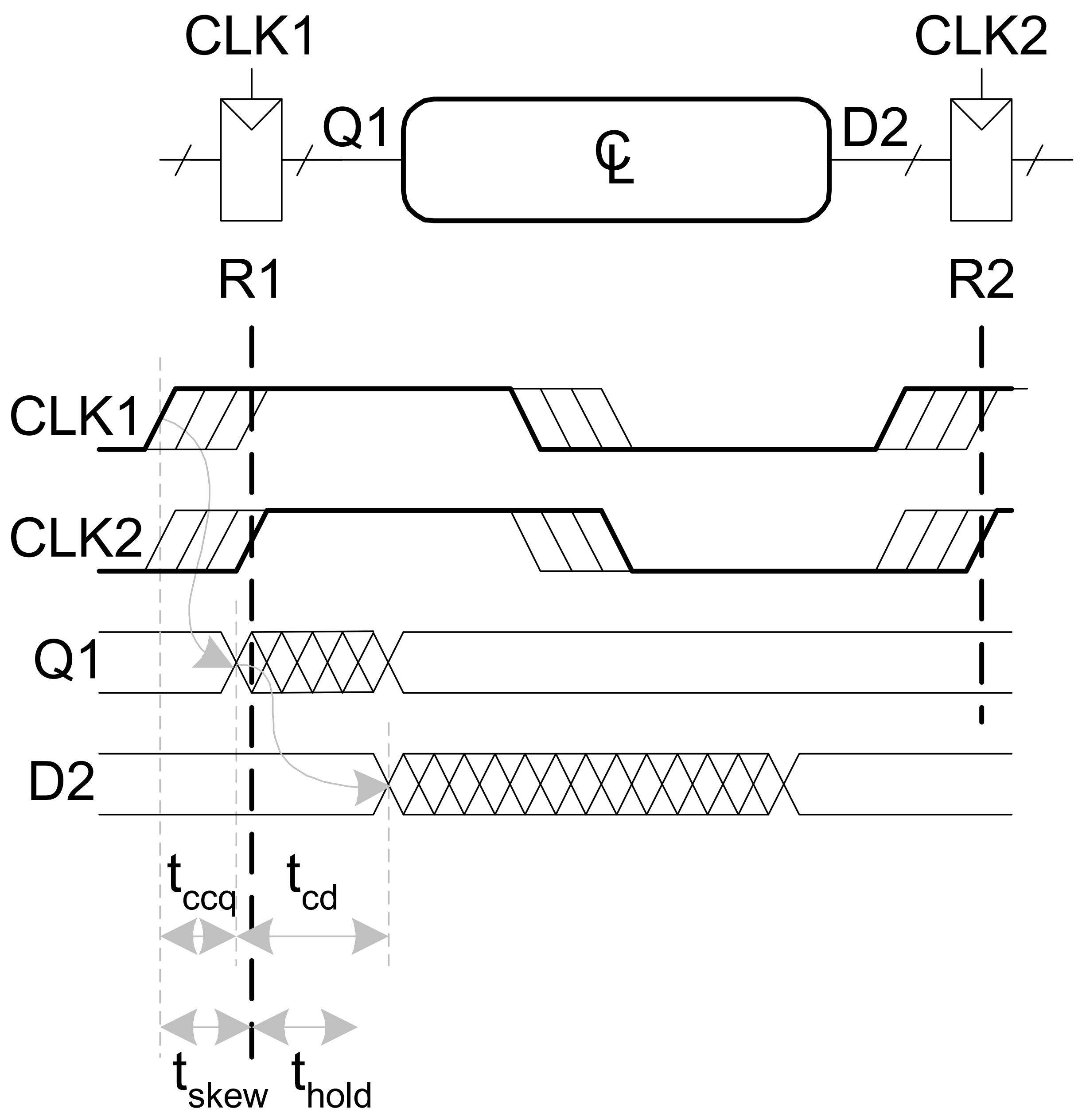
- In **the worst case**, CLK2 is later than CLK1



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

Hold Time Constraint with Skew

- In **the worst case**, CLK2 is later than CLK1

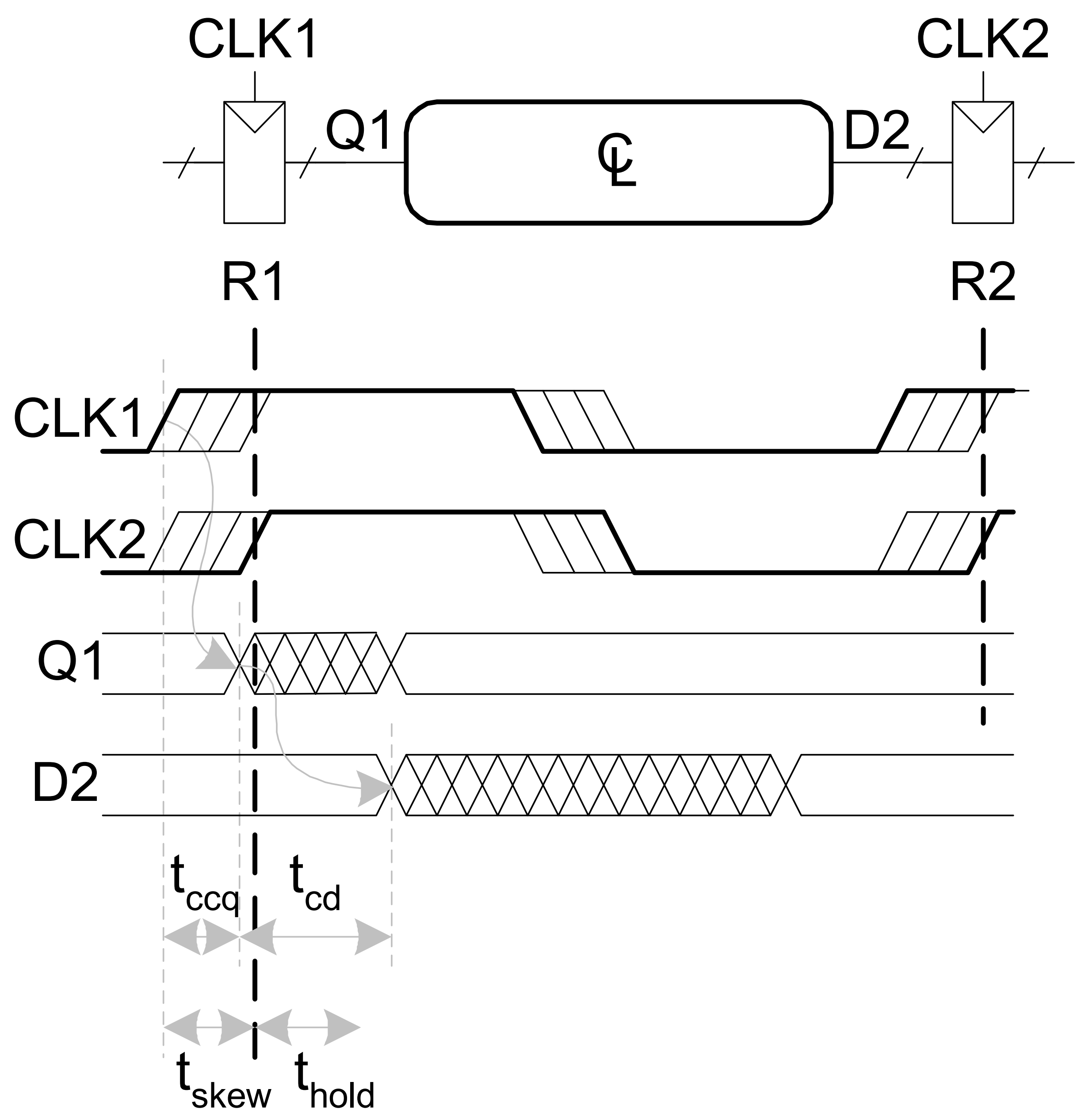


$$t_{ccq} + t_{cd} >$$

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

Hold Time Constraint with Skew

- In the worst case, CLK2 is later than CLK1



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

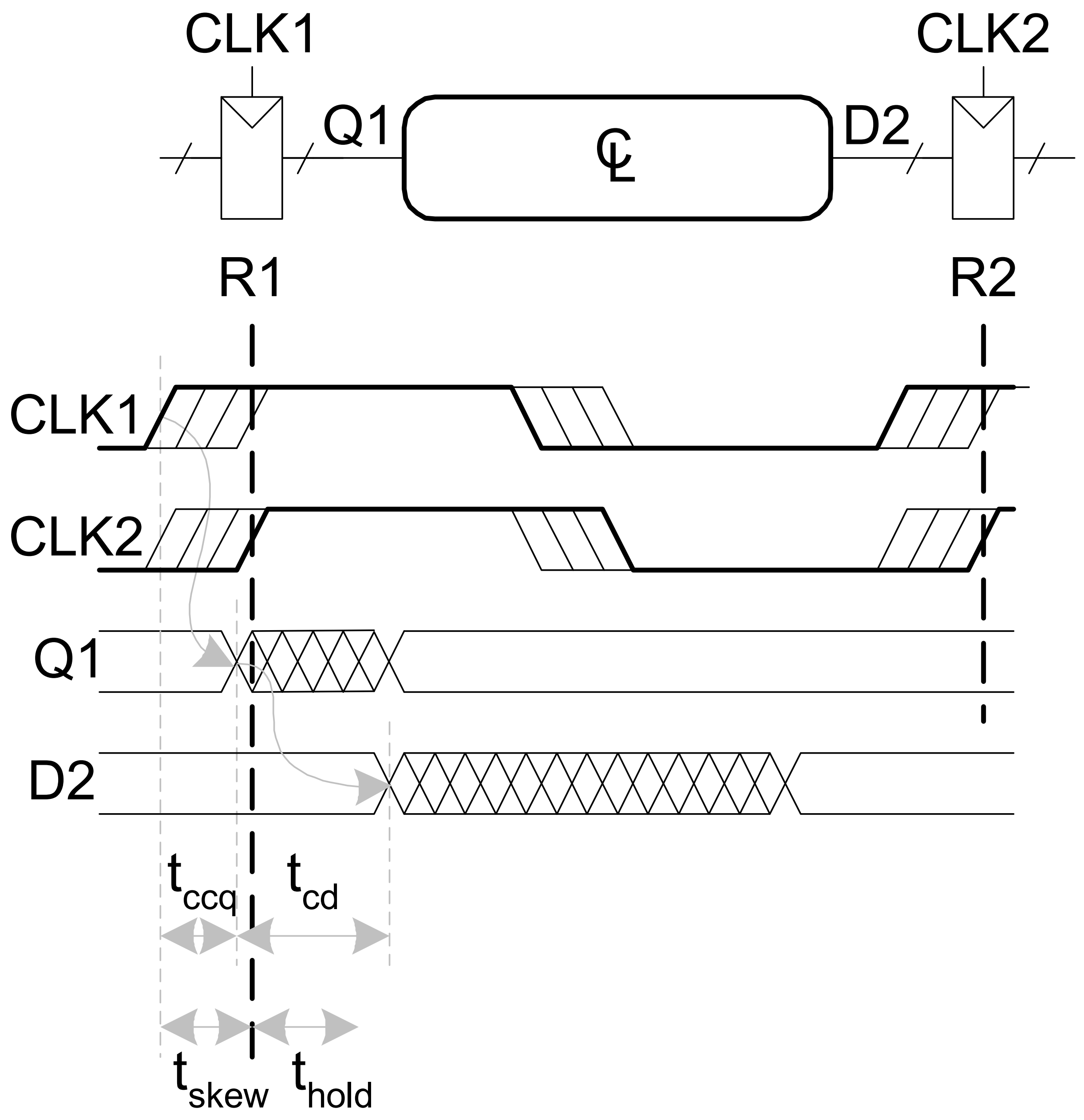
$$t_{cd} >$$

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

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

Hold Time Constraint with Skew

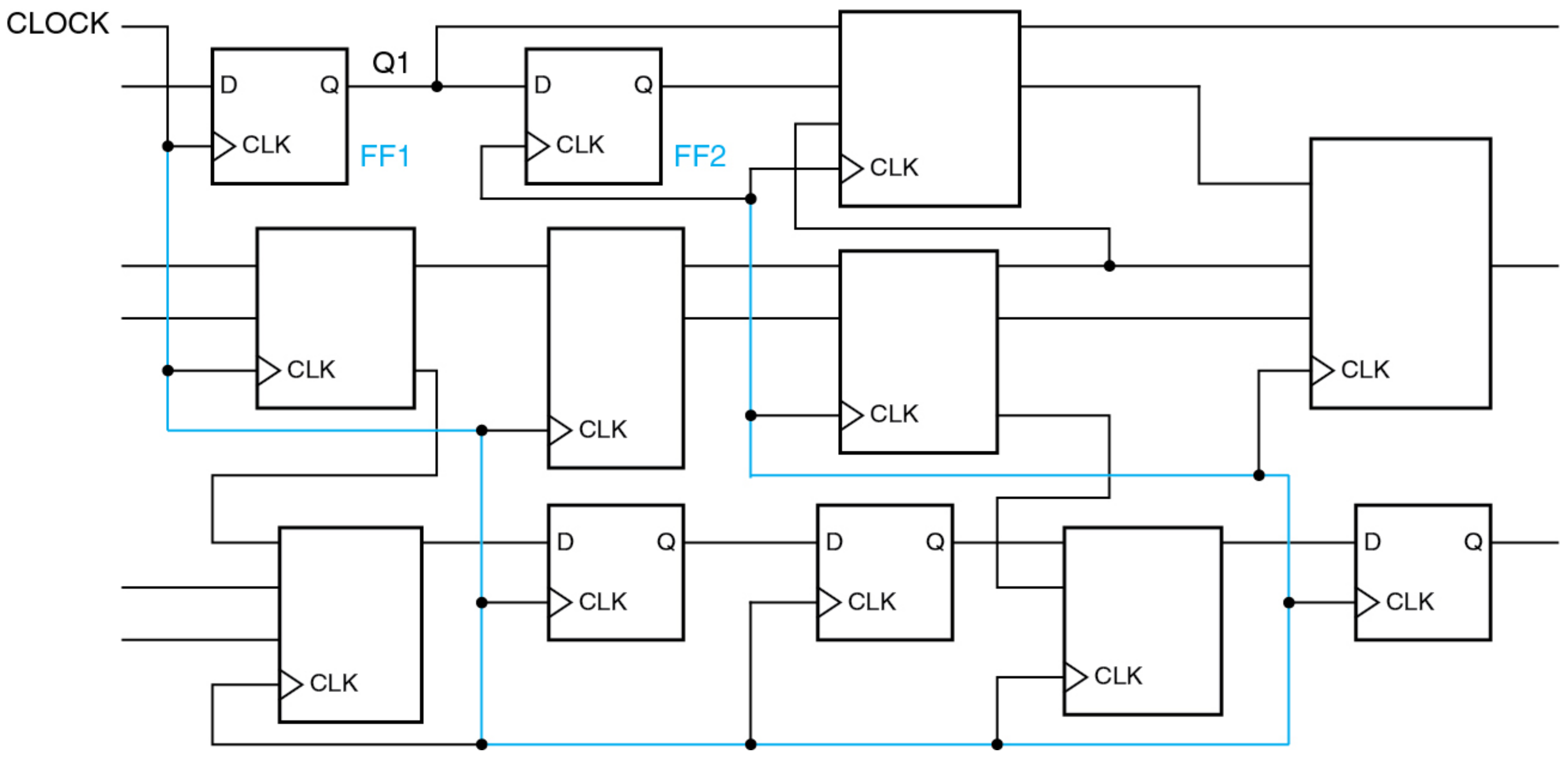
- In the worst case, CLK2 is later than CLK1



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

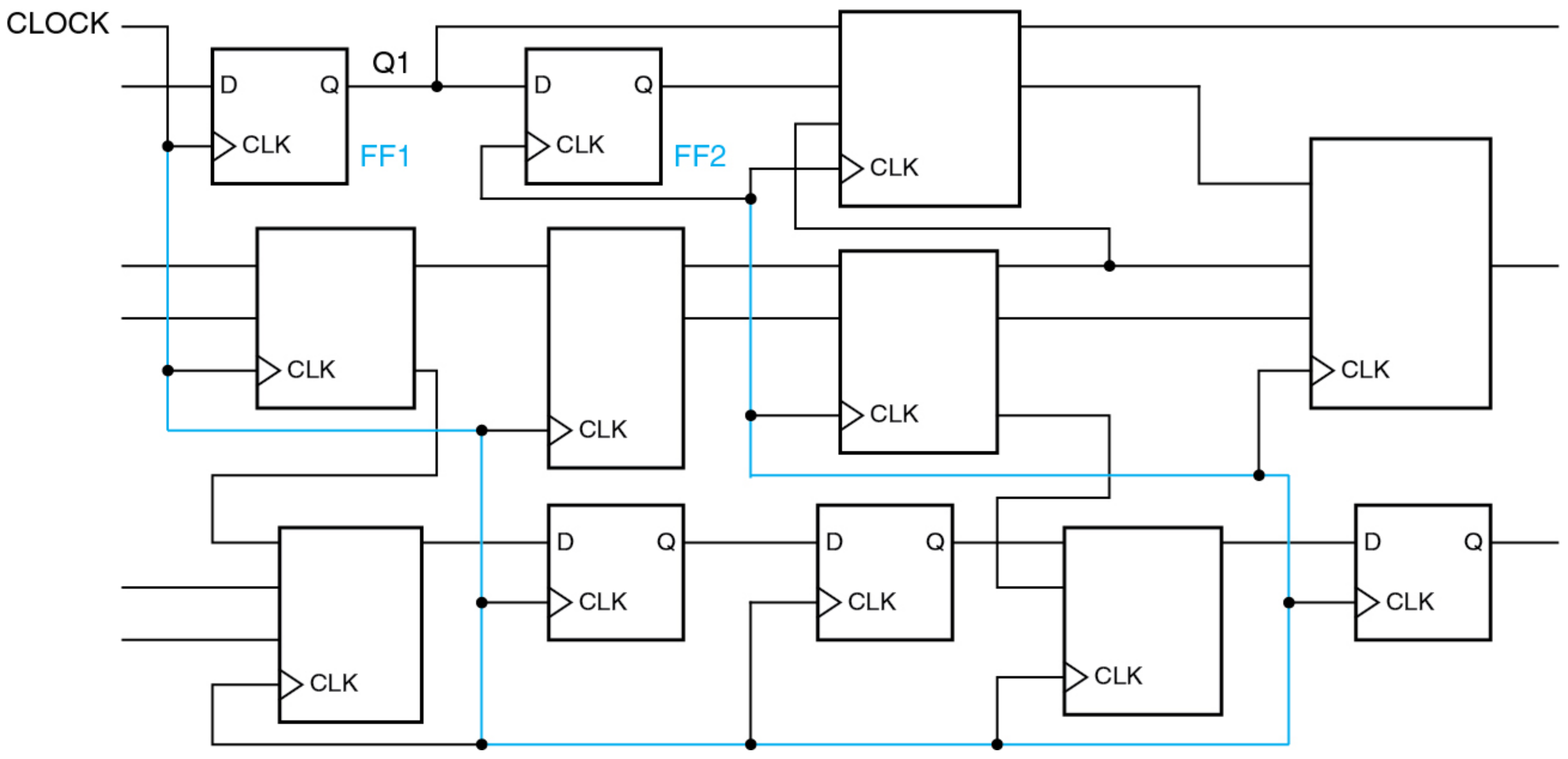
$$t_{hold} < t_{ccq} + t_{cd}$$
$$t_{cd} > t_{hold} - t_{ccq}$$

A Clock-Signal Path Leading to Excessive Skew



A Clock-Signal Path Leading to Excessive Skew

- Problem: clock signal may arrive FF2 **much later**!

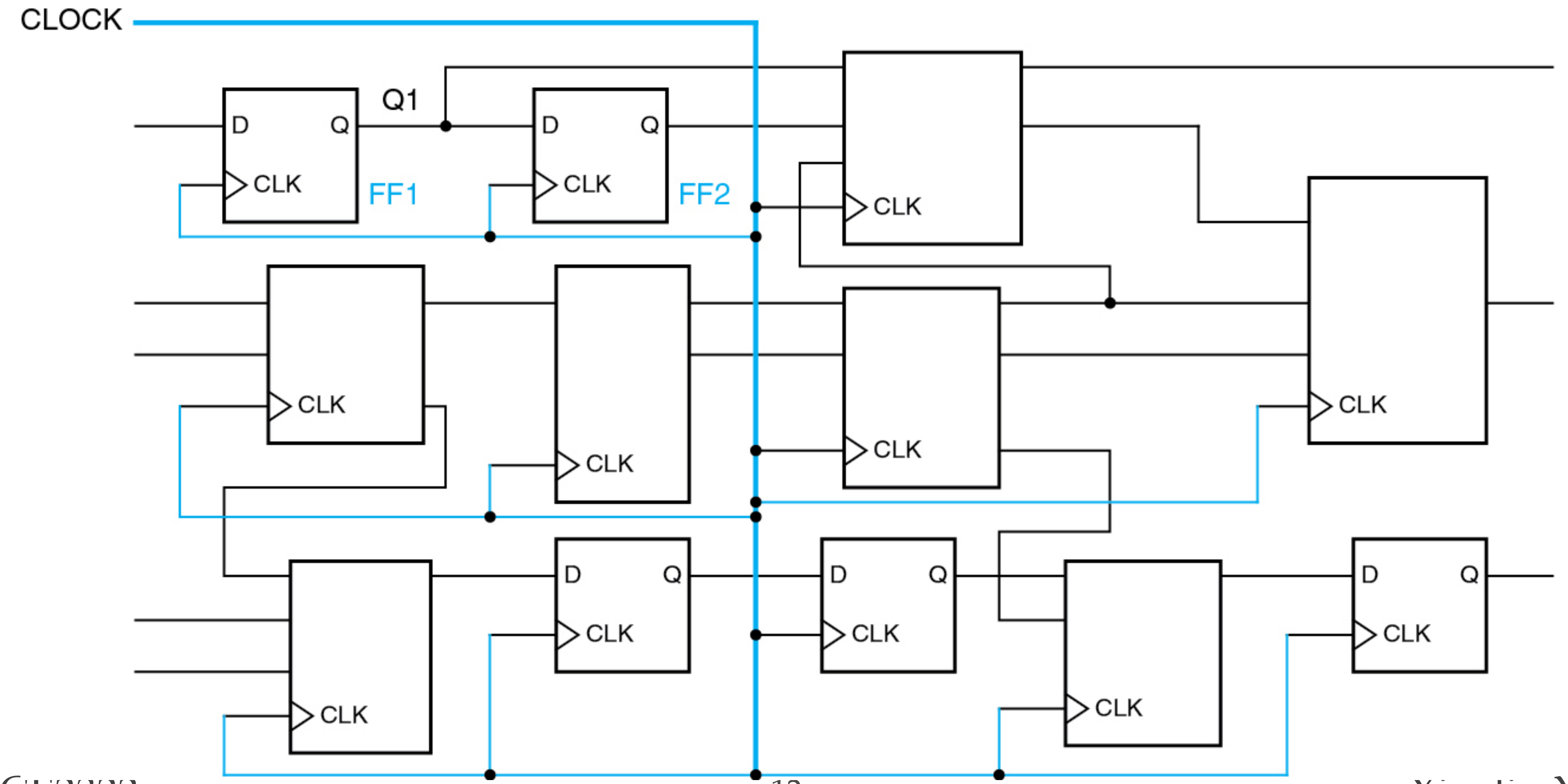


Clock-Signal Routing to Minimize Skew

- Distributed, balanced, tree-like structure
- Fastest type of wire connection

Clock-Signal Routing to Minimize Skew

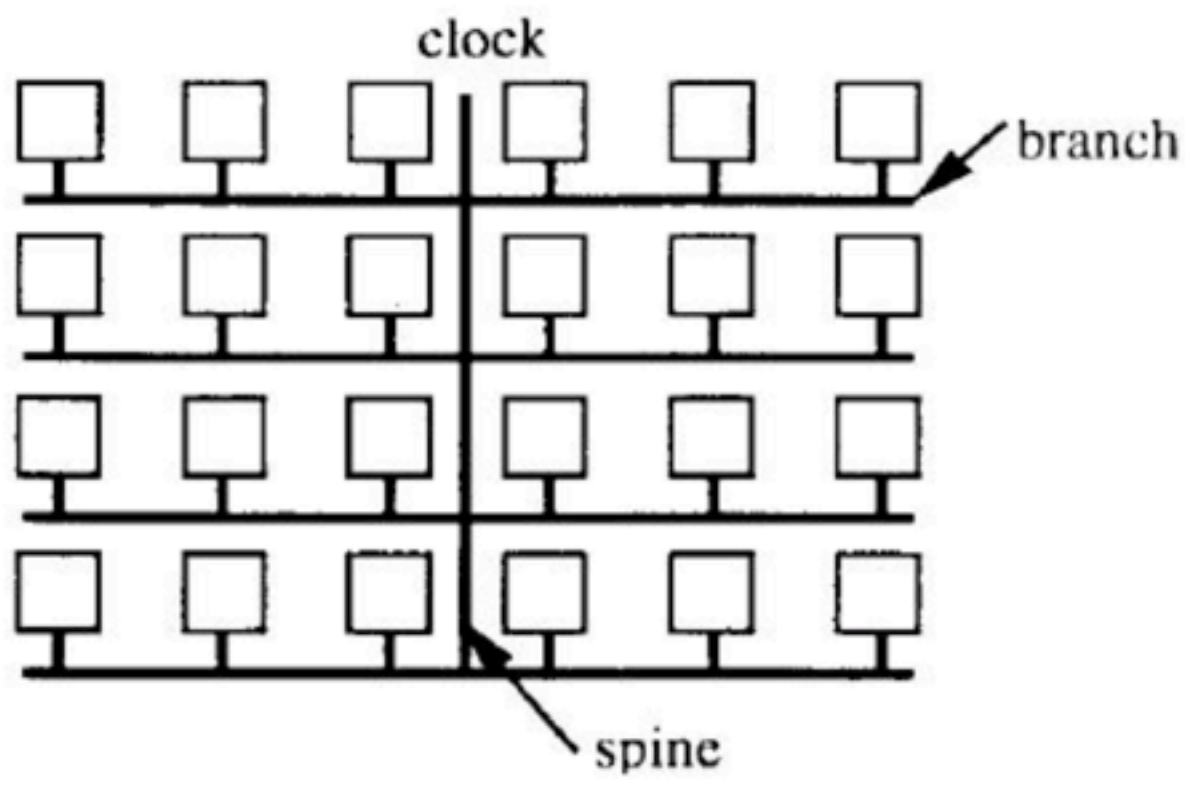
- Distributed, balanced, tree-like structure
- Fastest type of wire connection

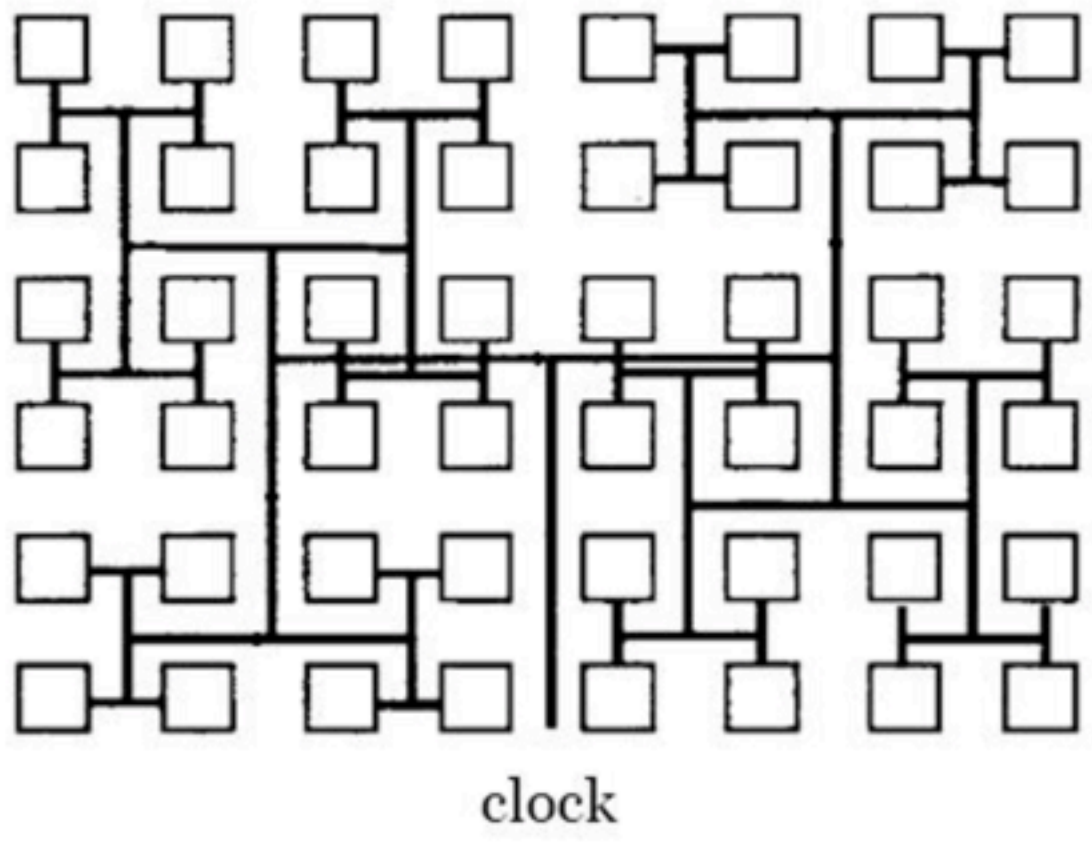


Clock-Signal Routing to Minimize Skew

Clock Tree Architecture

Comb	H-Tree
• Clock signal passed down the spine then spreads out through the branches • Does not have equal length traces	• All trace lengths are equal • Minimizes clock skew



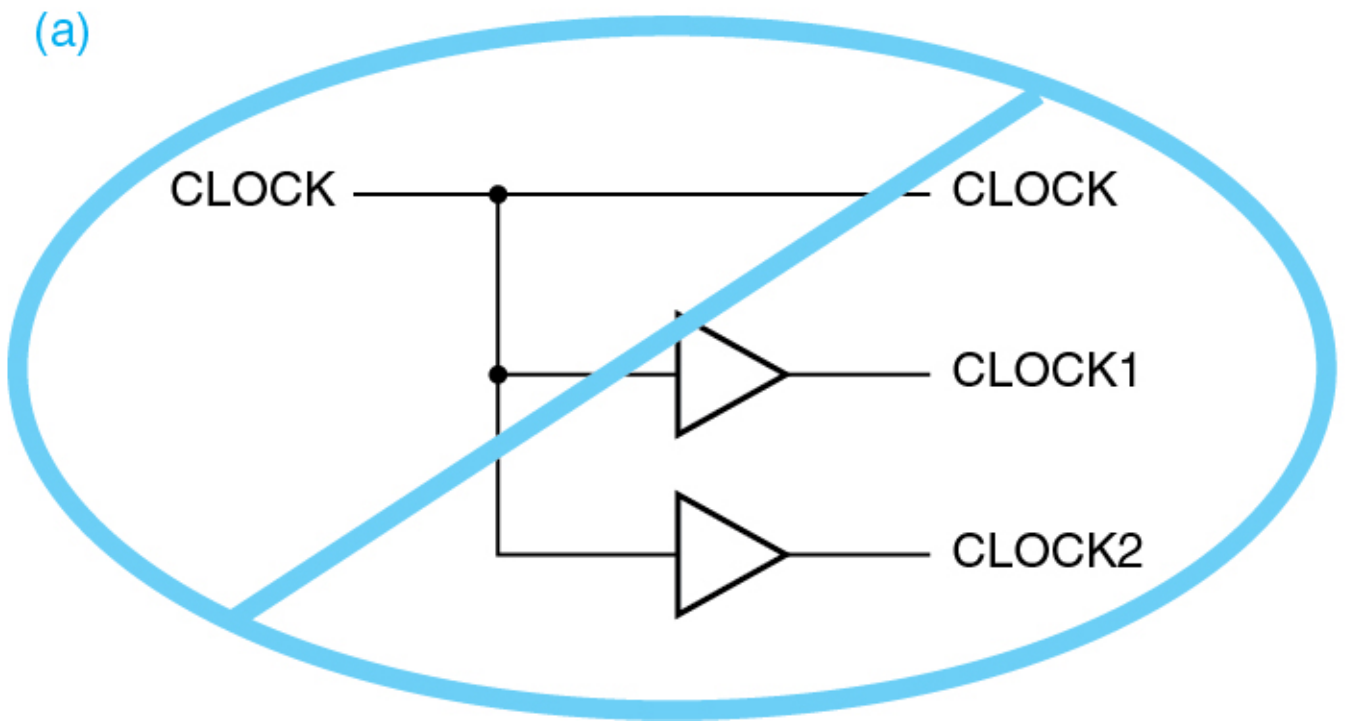


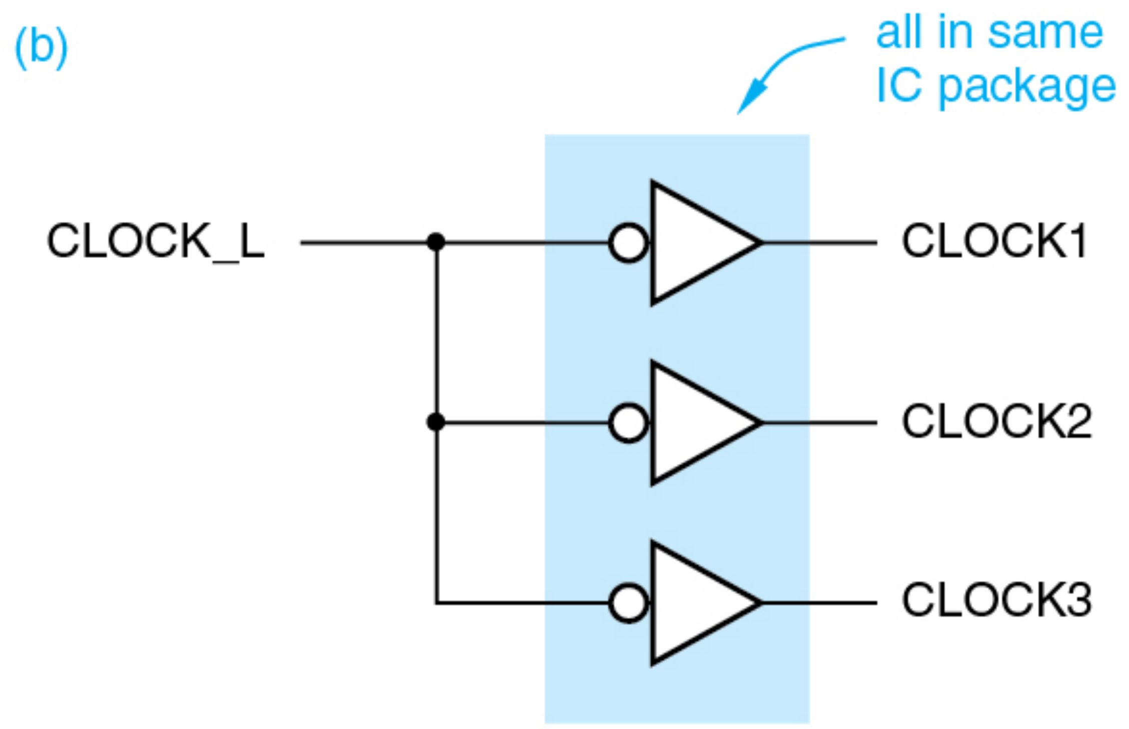
Clock-Signal Routing to Minimize Skew

- Receivers using the same clock signal
  - Should be placed equally close, inserted in a balanced way!
- Sometimes, designer may have to route the components by hand
- Modern EDA tools —> "clock tree synthesis (CTS)" algorithm
  - Back end design, after the logic components have been placed on the chip

Buffering the Clock

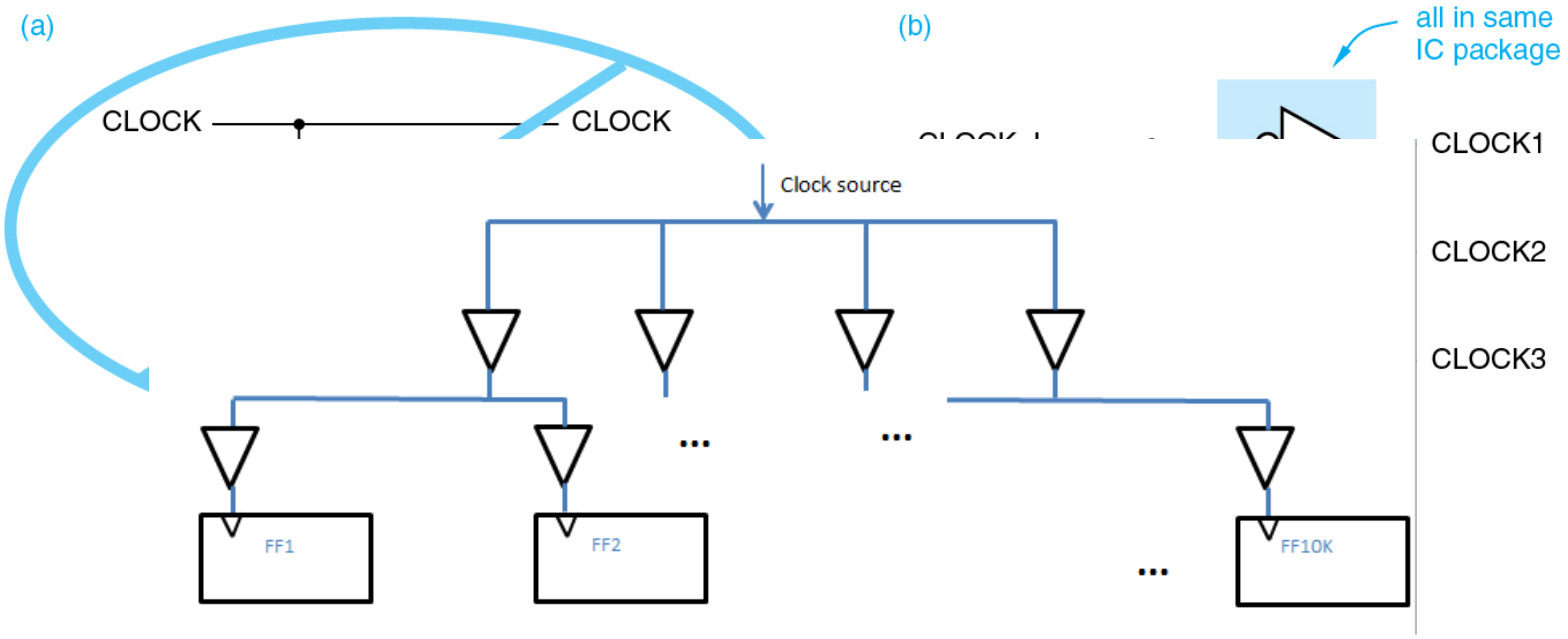
- (A) Excessive Clock Skew; (B) Controllable Clock Skew
- While more copies are needed?





Buffering the Clock

- (A) Excessive Clock Skew; (B) Controllable Clock Skew
- While more copies are needed?

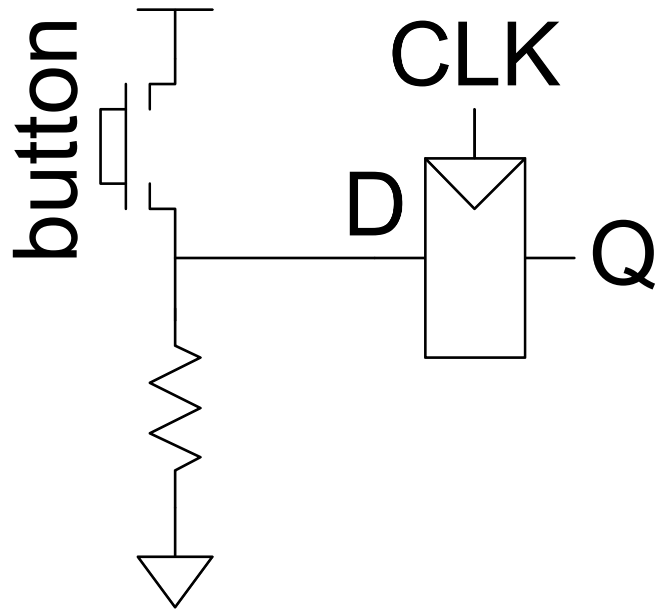


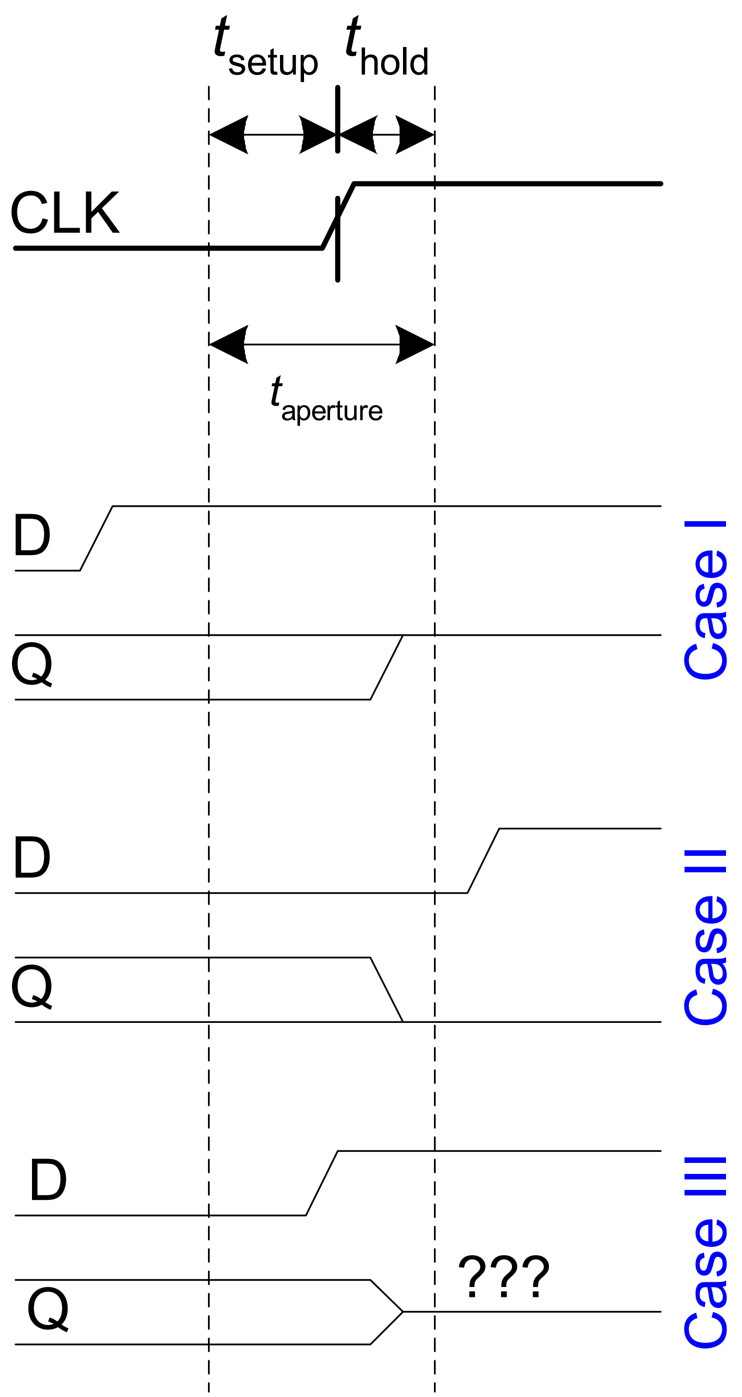
What if a Circuit DOES Violate the Timing Constraints

- For example, violating setup or hold time?
- Either due to circuit design
- Or
- Due to the external input!

Violating the Dynamic Discipline

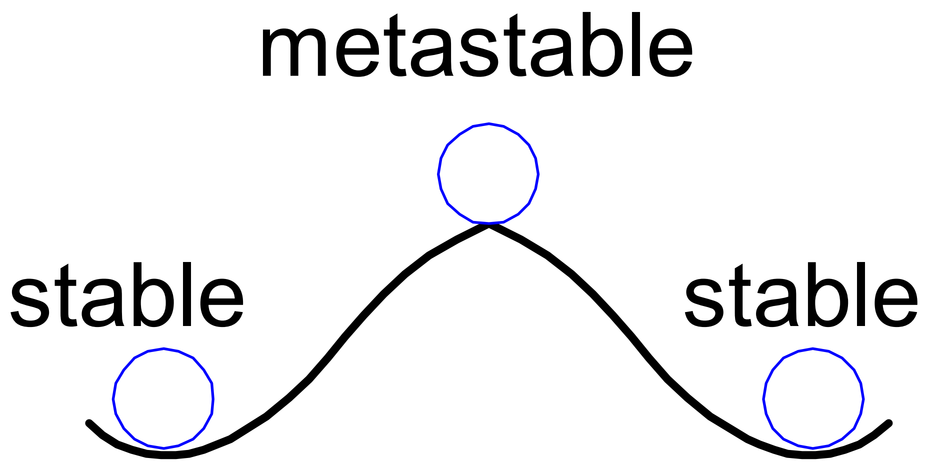
- Asynchronous (for example, user) inputs might violate the dynamic discipline





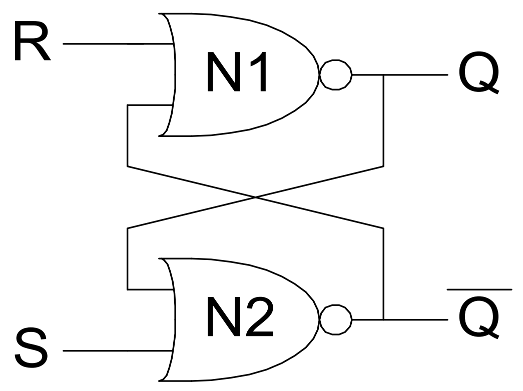
Metastability

- **Bistable devices:** two stable states, and a metastable state between them
- **Flip-flop:** two stable states (1 and 0) and one metastable state
- If flip-flop lands in metastable state, could stay there for an undetermined amount of time



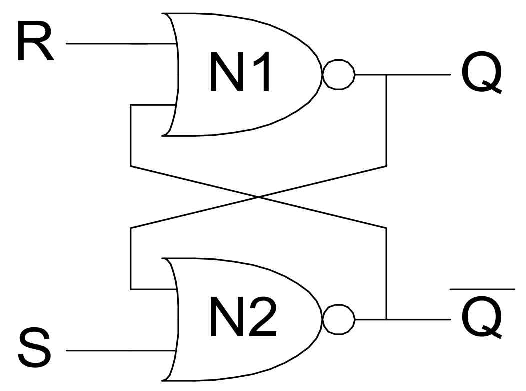
Flip-Flop Internals

- Flip-flop has **feedback**: if *Q* is somewhere between 1 and 0, cross-coupled gates drive output to either rail (1 or 0)



- **Metastable signal:** if it hasn't resolved to 1 or 0

Flip-Flop Internals



- If flip-flop input changes at random time, **probability that output *Q* is metastable** after waiting some time, *t*:

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

t_{res} :	time to resolve to 1 or 0
T_0, τ :	properties of the circuit

Metastability

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

t_{res} : time to resolve to 1 or 0

T_0, τ : properties of the circuit

- **Intuitively:**
  - ***T*₀/*T*c**: probability input changes at a bad time (during aperture)

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

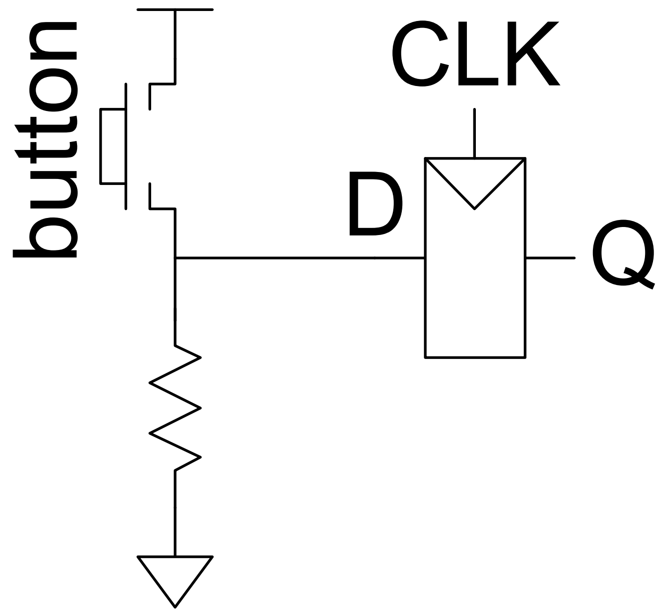
  - **τ**: time constant for how fast flip-flop moves away from metastability

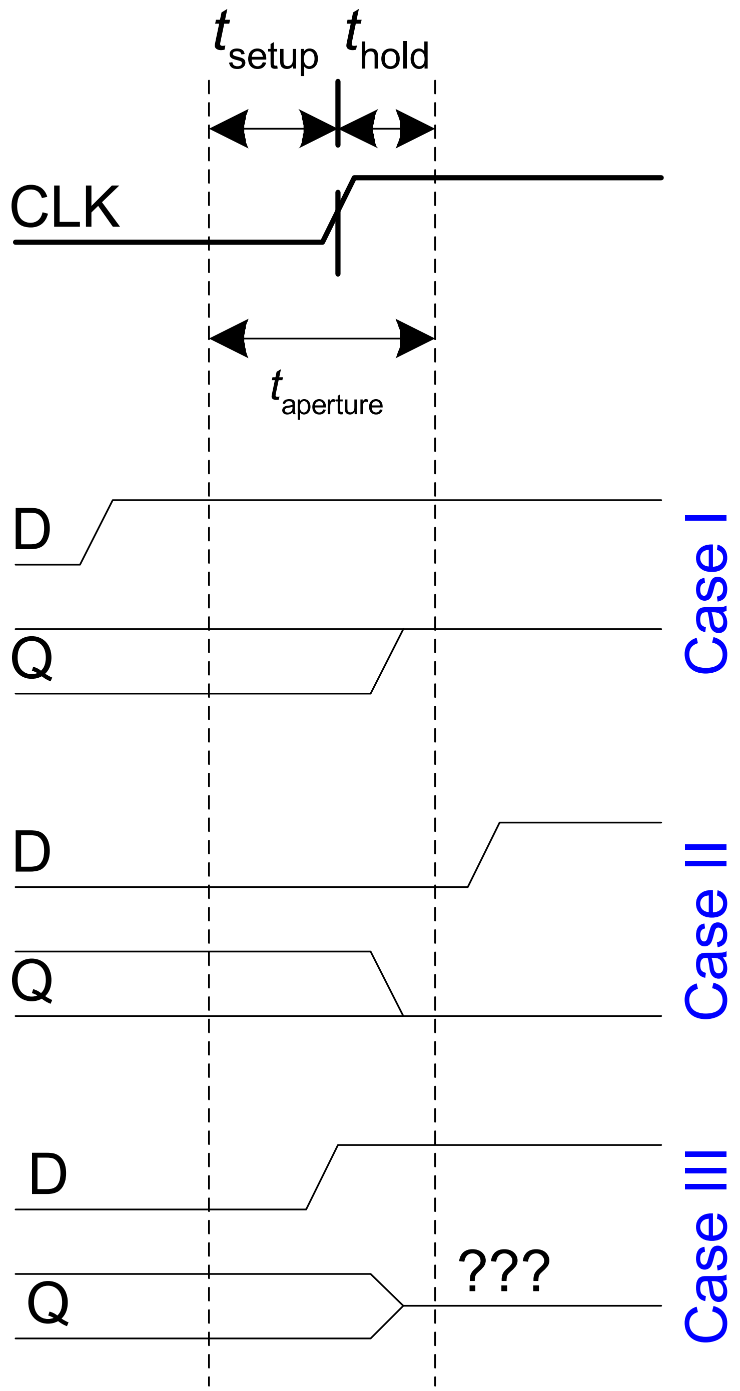
$$P(t_{res} > t) = (T_0/T_c)\, \boldsymbol{e^{-t/\tau}}$$

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

Violating the Dynamic Discipline

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





Synchronizers

- **Asynchronous inputs are inevitable** (user interfaces, systems with different clocks interacting, etc.)
- **Synchronizer goal:** make the probability of failure (the output Q still being metastable) low
- Synchronizer cannot make the probability of failure 0