ECE 225
High-Speed Digital IC Design

Lecture 14

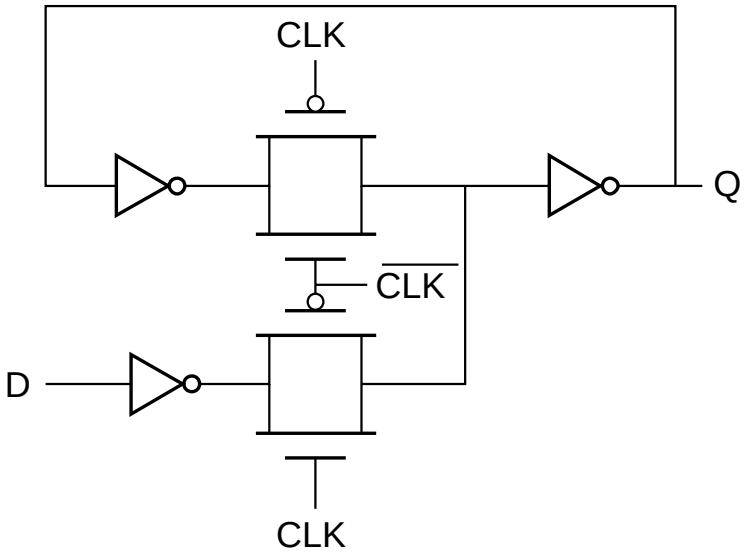
Sequential Logic Circuits-II, Timing, Clocking….

Prof. Kaustav Banerjee
Electrical and Computer Engineering
E-mail: kaustav@ece.ucsb.edu

Storage Mechanisms

A Static Latch

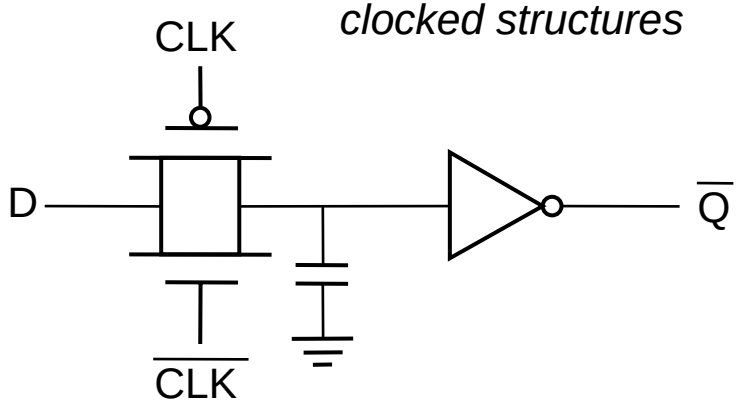
Uses a bistable memory device (FF), more complex



$$Q = \overline{Clk} \,.\, Q + Clk \,.\, D$$

Dynamic Latch (charge-based)

Useful for frequently clocked structures



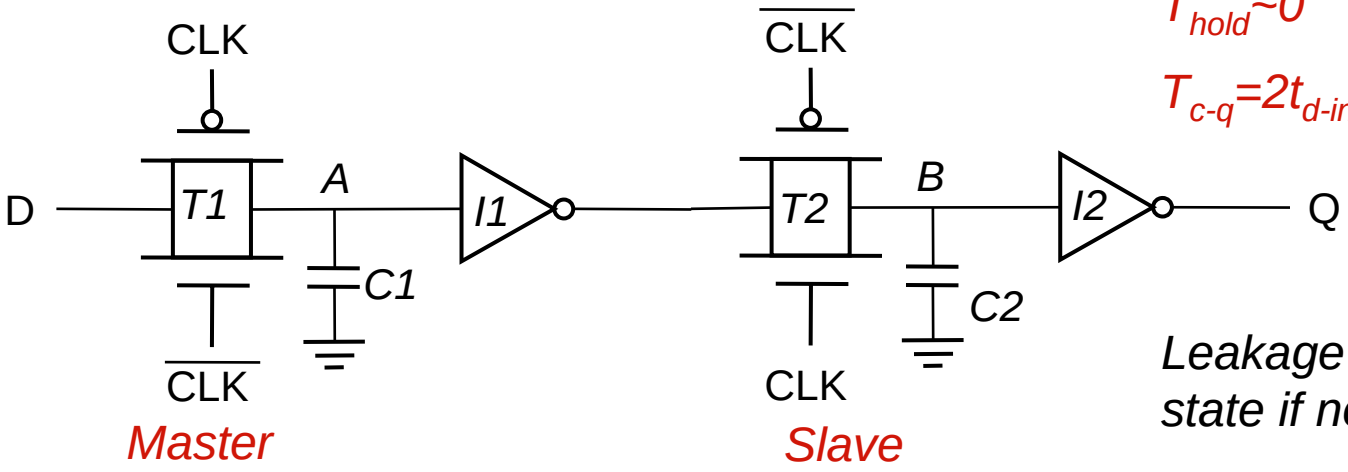
Stored value can remain for a limited time….few ms….need to refresh periodically

Dynamic X-gate Edge-Triggered Registers

$T_{su} = t_{dx\text{-}T1}$

$T_{hold} \sim 0$

$T_{c\text{-}q} = 2t_{d\text{-}inv} + t_{dx\_T2}$



Leakage will destroy the state if not refreshed

CLK=0: *D input is sampled at storage node 1, Slave stage in hold mode with node 2 in high-impedance state*

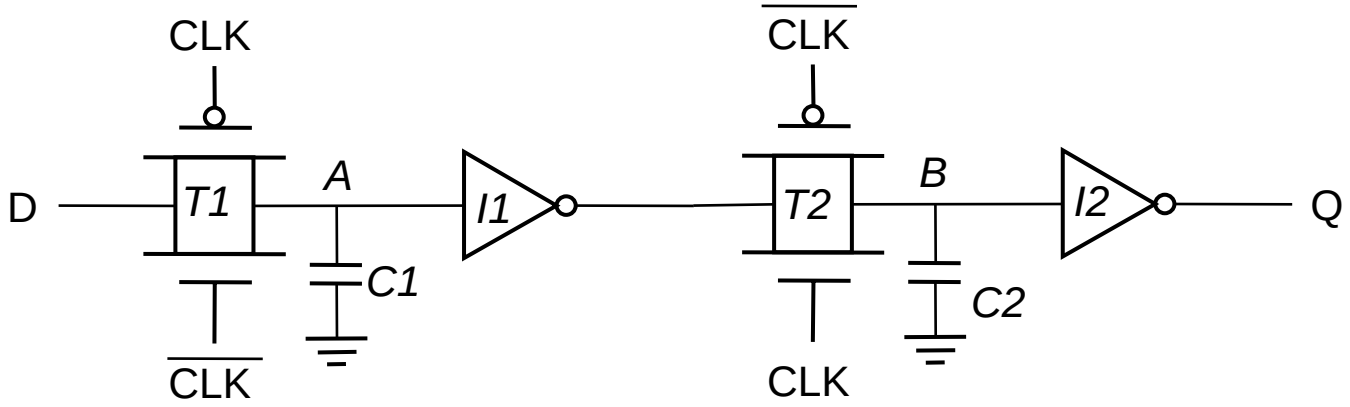
At the rising edge of CLK: *T2 is turned on and value sampled at node 1 right before the rising edge is copied to Q*

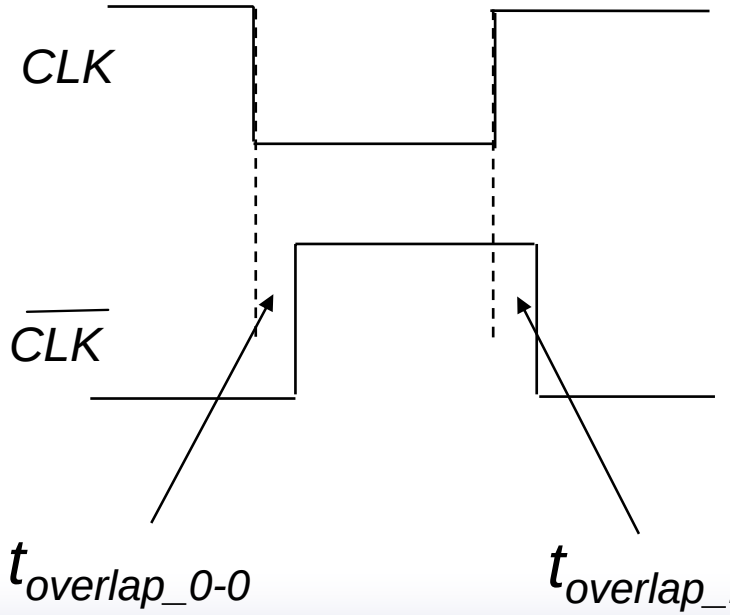
Very efficient: *requires only 8 transistors, can be made even simpler (6 transistors) using NMOS-only pass transistors*

Race condition *can occur due to CLK overlaps!*

Robustness *is also an issue….(state at the nodes can be distorted by injected noise)*

Impact of Non-overlapping Clocks…on +ve Edge Triggered Register





$$t_{overlap\_0\text{-}0} < t_{T1} + t_{I1} + t_{T2}$$

So that output Q doesn't change on the falling CLK edge….

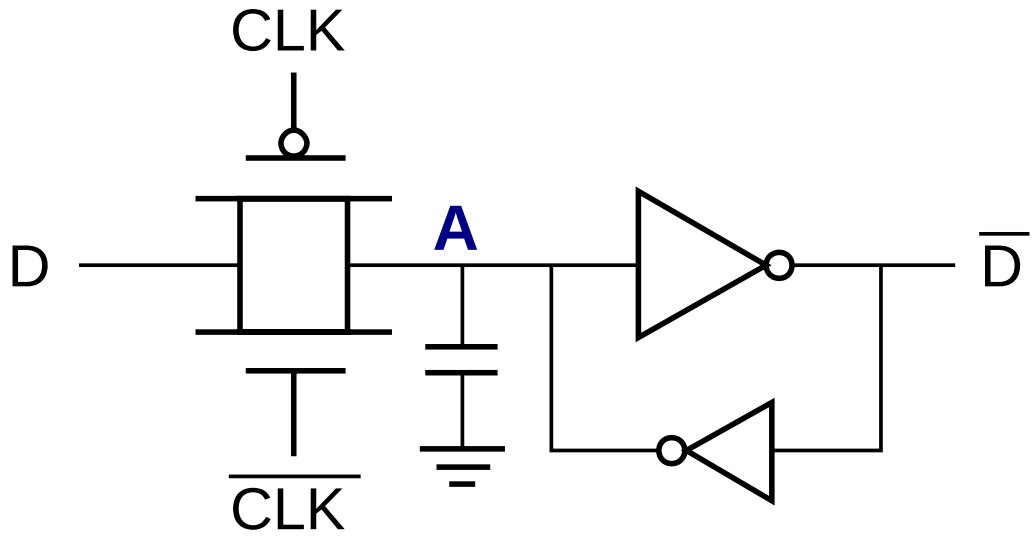
$$t_{hold} > t_{overlap\_1\text{-}1}$$

data must remain stable during the high overlap period

Making a Dynamic Latch Pseudo-Static

Problems with Dynamic Latches: *1) any signal net that is capacitively coupled to the internal storage node (such as A…) can inject significant noise and destroy the state 2) Leakage..during CLK gating or slowdown 3) internal dynamic node doesn't track variations in Vdd*

Problems of the dynamic register can be overcome by adding a weak feedback INV



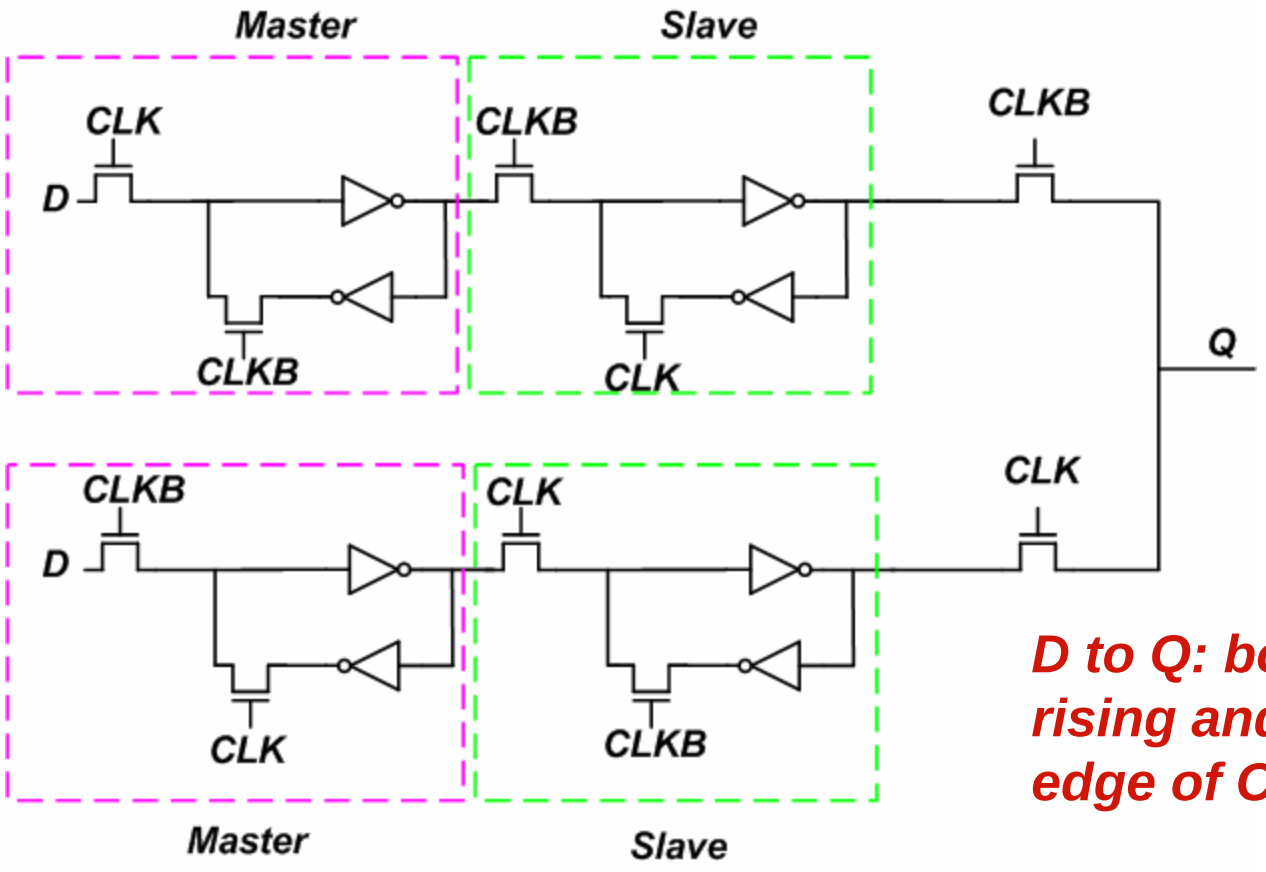
Increases delay slightly…but improves noise immunity

Most registers should be pseudo-static or static unless used in high-performance datapaths

Dual Edge-Triggered Master-Slave Flip-Flop

- Higher throughput
- Data is sampled both in positive-edge and negative edge of the clock
- Duty cycle of the clock should be 50%
- Setup time and hold time is important in both edges of the clock

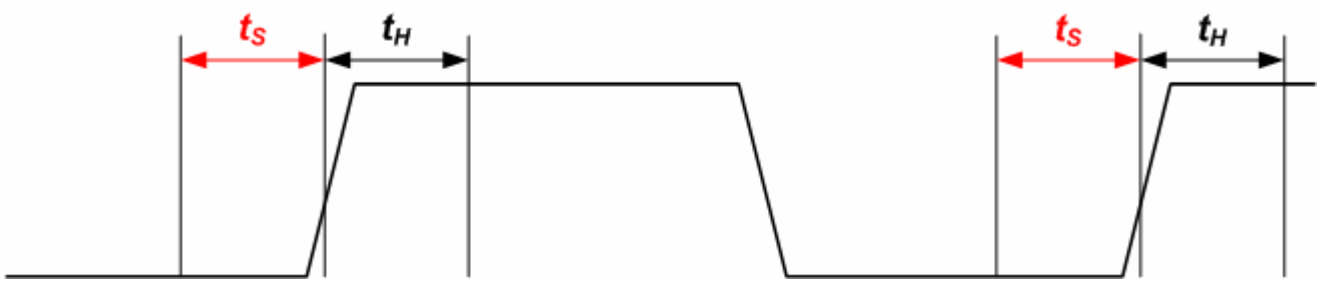
Dual Edge-Triggered Master-Slave Flip-Flop



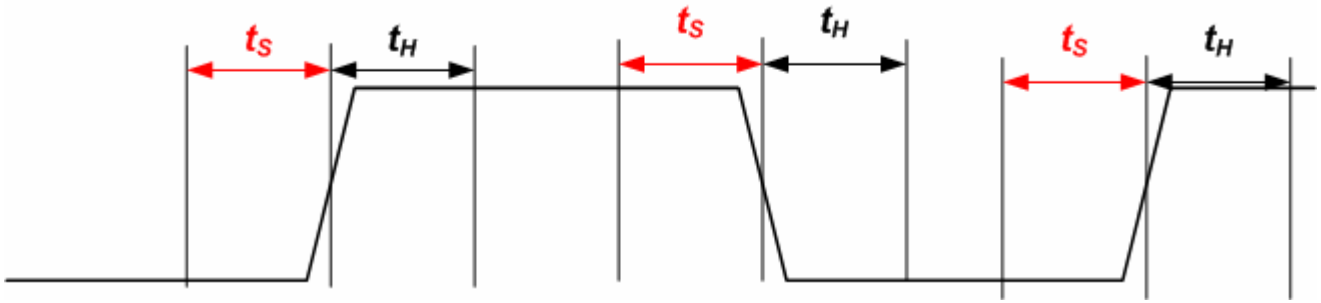
D to Q: both at the rising and falling edge of CLK

Single vs Dual Edge Triggered FF Timing

- Single Edge Triggered: $(T/2)_{CLK} > Max(t_S, t_H)$



- Double Edge Triggered: $(T/2)_{CLK} > t_S + t_H$



Note: lower frequency CLK giving same throughput, good for power savings

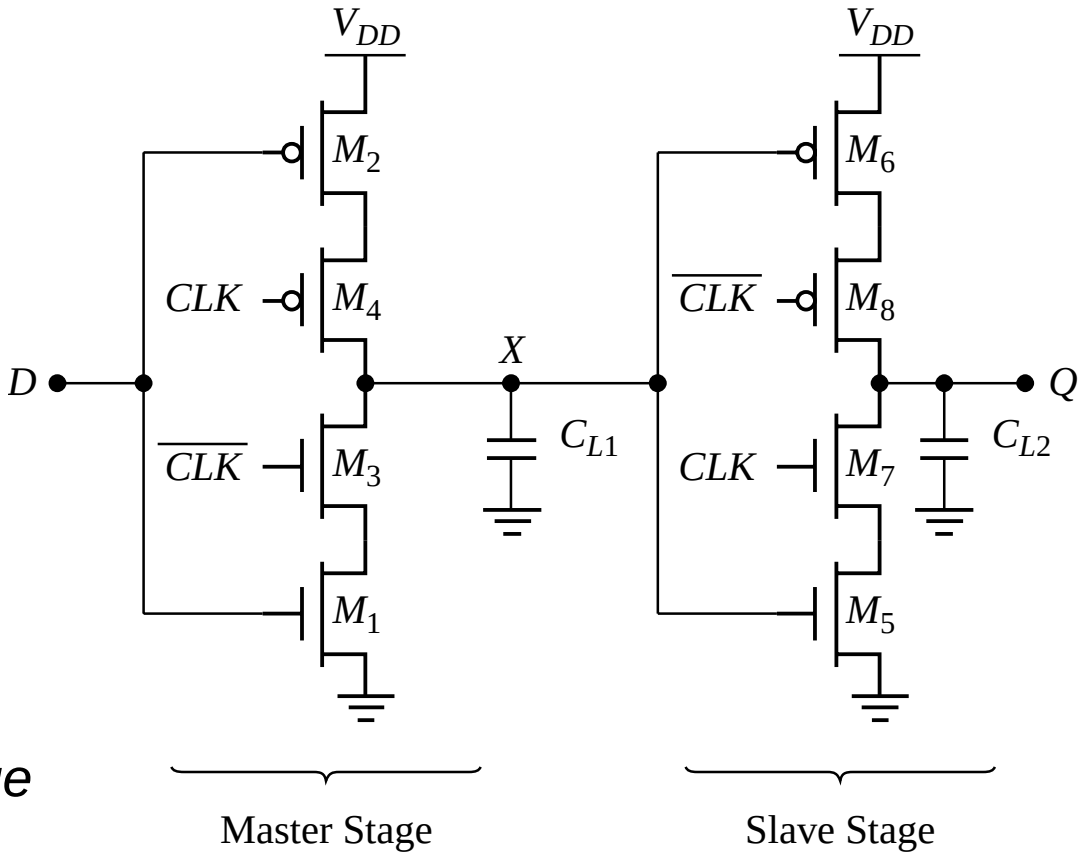
Other Latches/Registers: C²MOS

***C**locked **C**MOS* Register: +ve edge triggered Master-Slave FF

CLK=0:

first tri-state buffer turns on….and Master samples the inverted version of D

Q retains previous value on C_{L2}



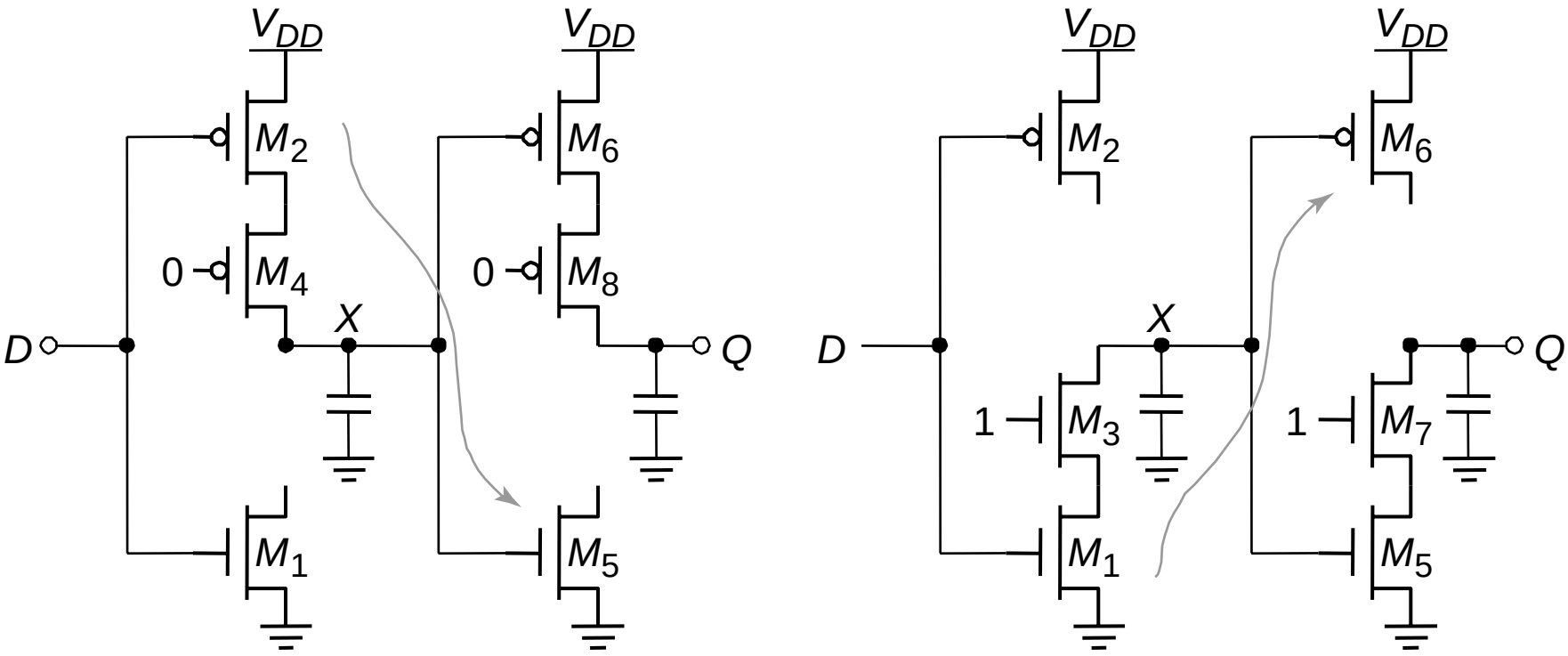
CLK=1:

Master is in hold mode…second stage turns on and C_{L1} propagates to the output Q

"Keepers" can be added to make circuit pseudo-static

C^2MOS: Insensitive to Clock-Overlap

Either PUN or PDN is activated by overlaps…but not both simultaneously….



(a) (0-0) overlap

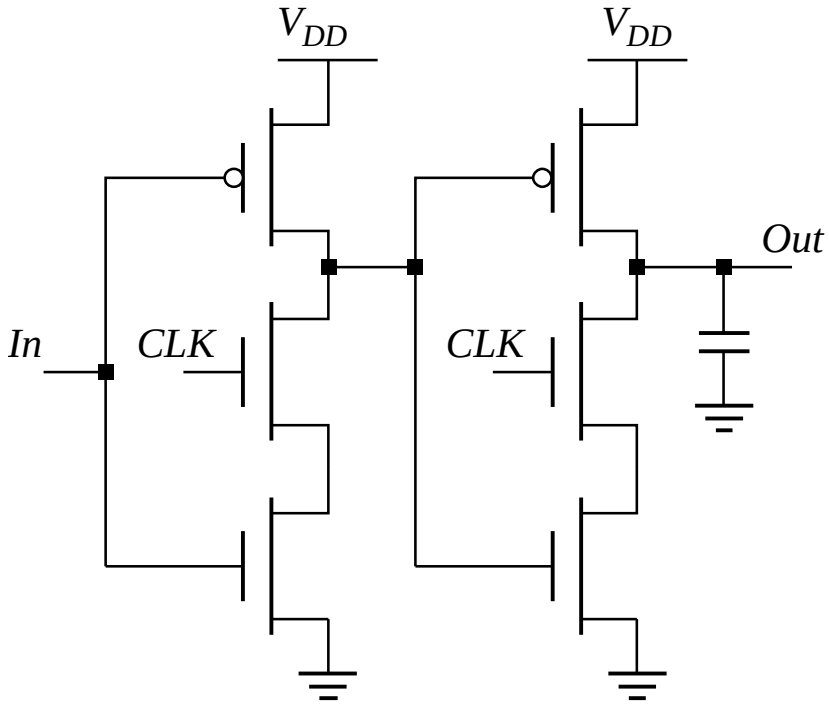
(b) (1-1) overlap

….hence any change in stored value at X cannot propagate to Q

Other Latches/Registers: TSPCR

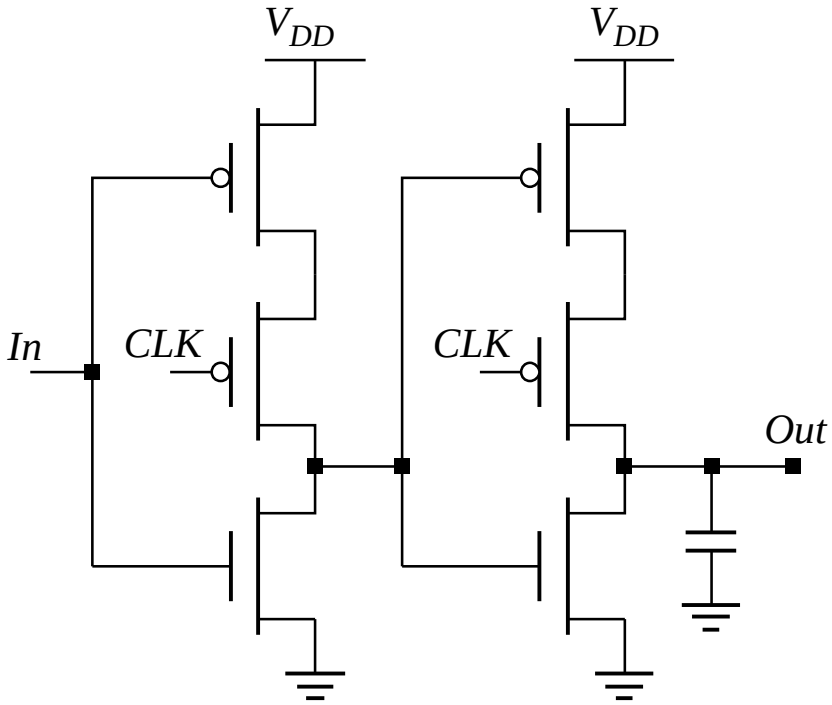
(True Single-Phase Clocked Register)

No signal can ever propagate from input to output…because of the dual stage approach….slightly increases number of transistors (=12)



Positive latch

(transparent when CLK= 1, when CLK=0, both inverters are disabled...latch is in hold mode)

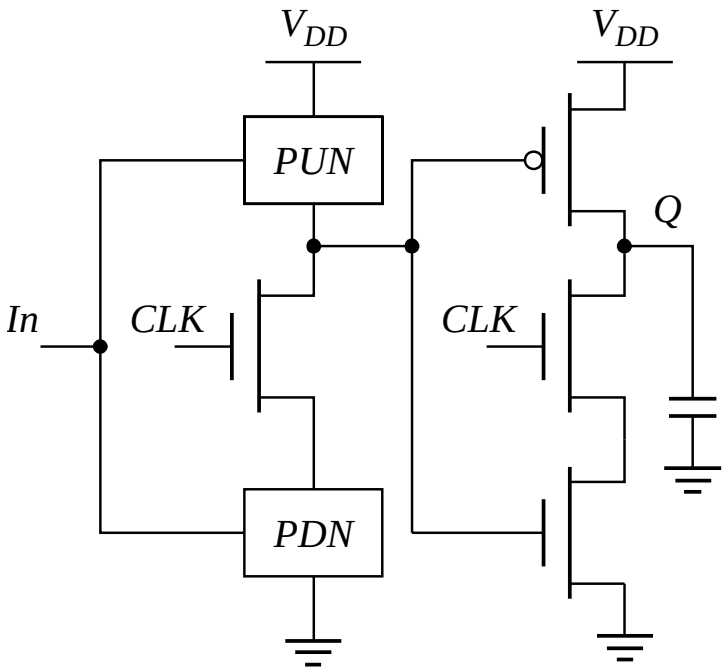


Negative latch

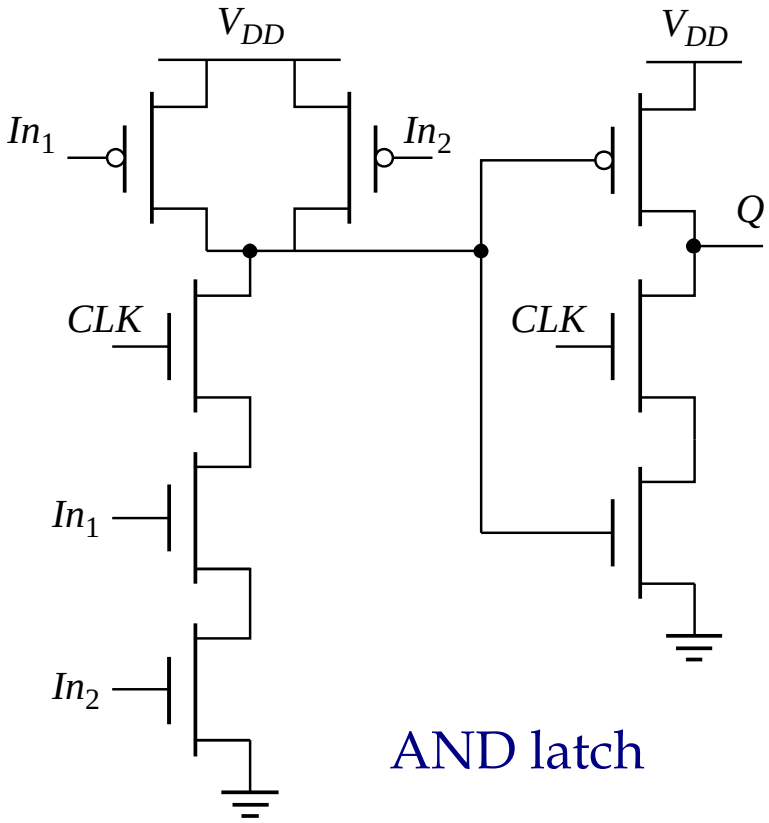
(transparent when CLK= 0)

Including Logic in TSPC

Reduces delay overhead associated with the latches…..



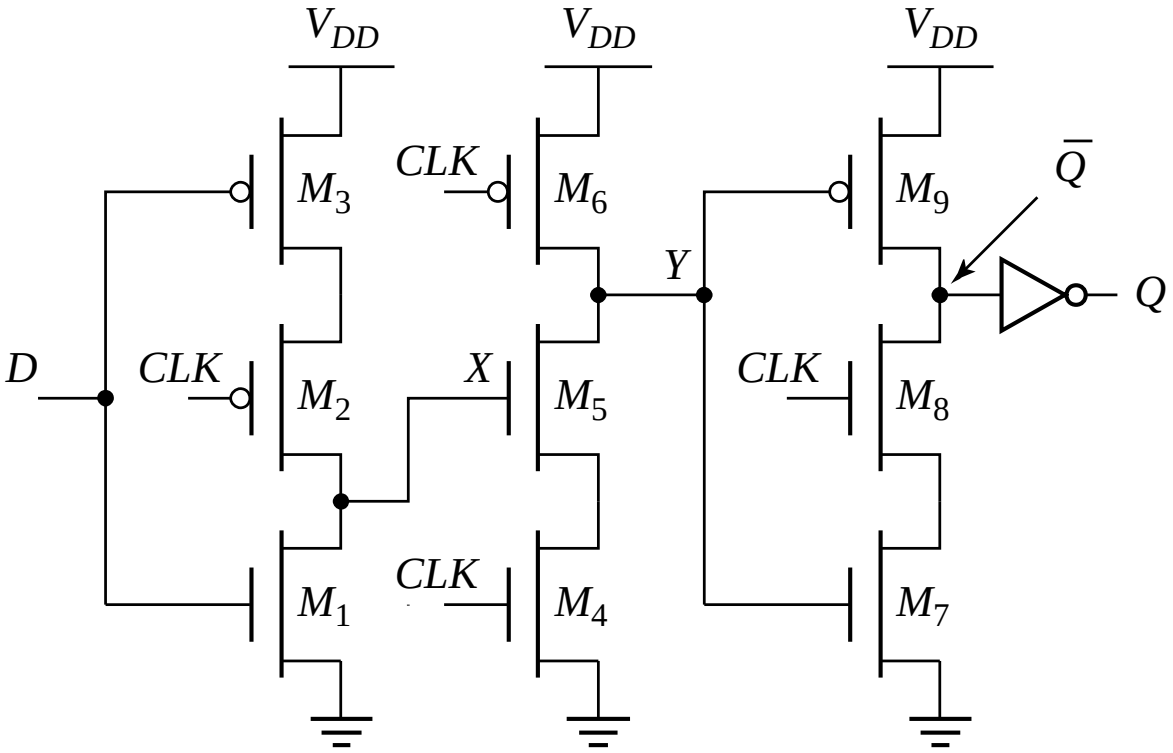
Example: logic inside the latch



AND latch

Note: set-up time increases a bit…but overall performance (T_{CLK}) is improved….

A Specialized TSPC Edge-Triggered Register



First inverter: *samples inverted version of D at X …..for CLK=0*

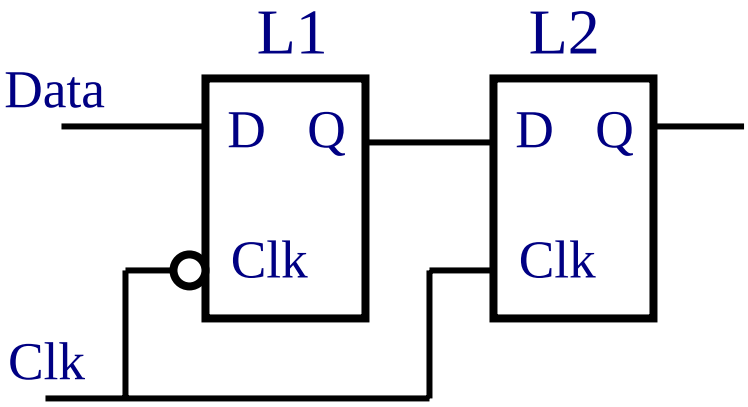
Second inverter: *is in pre-charge mode…M6 pull up Y*

Third inverter: *is in hold mode….M8 and M9 are off*

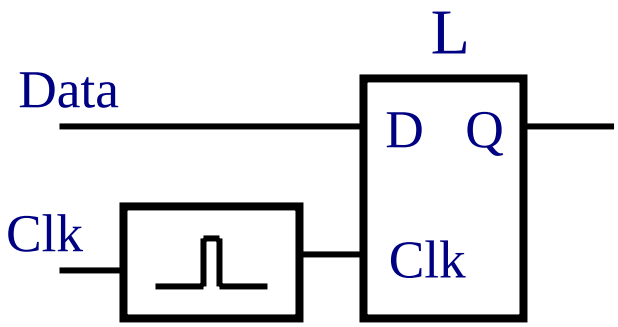
Pulse-Triggered Latches
An Alternative Approach

Ways to design an edge-triggered sequential cell:

Master-Slave Latches



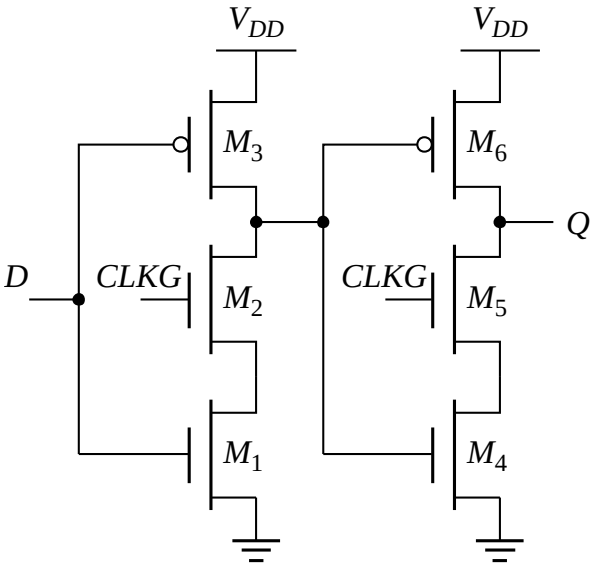
Pulse-Triggered Latch



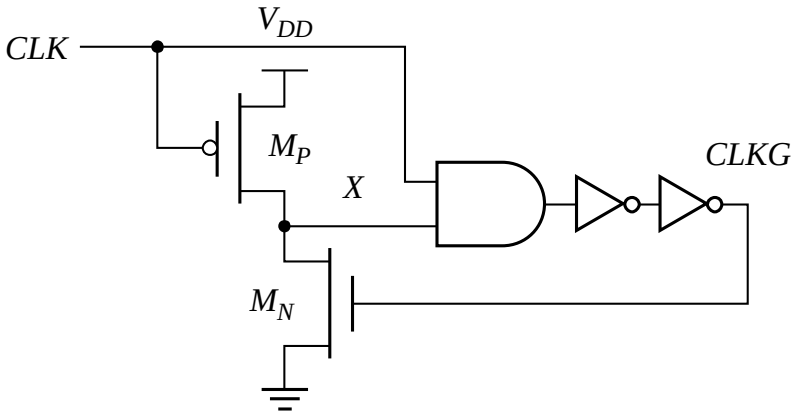
Pulsed Latches

A short-pulse is constructed around the rising (or falling) edge of the CLK….

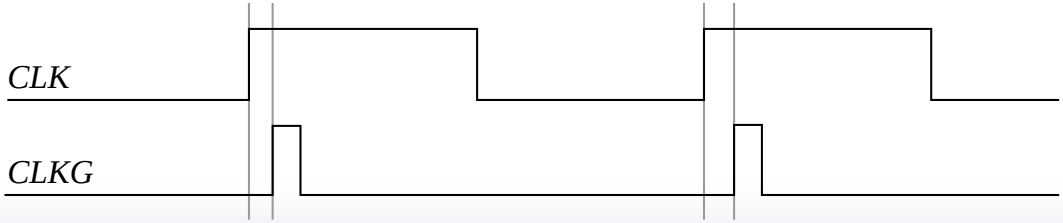
Pulse acts as the CLK input to a latch….sampling the input only in a small window…avoids RACE conditions



(a) register



(b) glitch generation



(c) glitch clock

TIMING……is everything!!

Timing Classification of Digital Systems

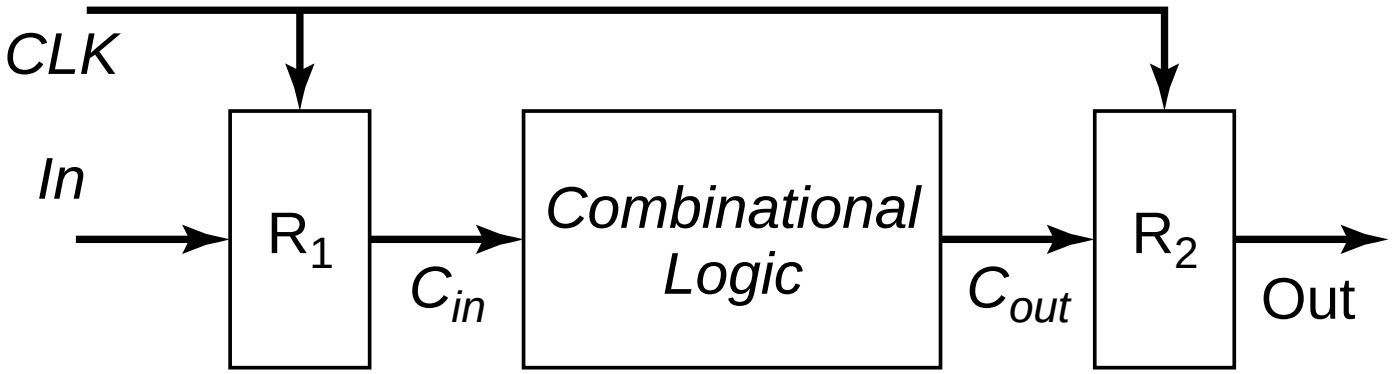
Synchronous: A signal that has exactly the same frequency as the local CLK and has a known fixed phase offset to that CLK

Mesochronous: A signal that has the same frequency as the local CLK but has an unknown phase offset w.r.t. that CLK.

Plesiochronous: A signal that has a “slightly different” frequency compared to that of the local CLK. This causes the phase difference to drift in time.

Asynchronous: A signal that has no fixed relationship w.r.t. that of the local CLK. Asynchronous signals can transition any time.

Synchronous Timing



The certainty period of the signal C_{out}---the period during which data are valid is synchronized with the system CLK. Hence, R2 can sample the data with confidence.

Mesochronous Timing

- If data are being passed between two different CLK domains, the data signal transmitted from the first module can have an unknown phase relationship to the CLK of the receiving module
- Not possible to directly sample the output data at the receiving module because of uncertainty in the phase offset
- A synchronizer is needed to synchronize the data signal with the receiving CLK

Plesiochronous Timing

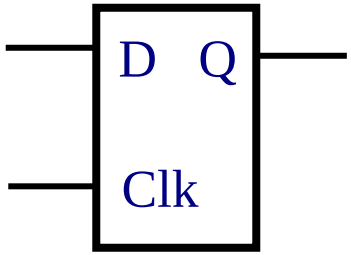
- Signal has slightly different frequency as compared to the local CLK---phase difference drifts over time
- Can arise due to the use of different CLK generators
- Practically, plesiochronous timing occurs in large distributed systems that involves long distance communications
- A timing recovery circuit is needed to ensure that all data are received

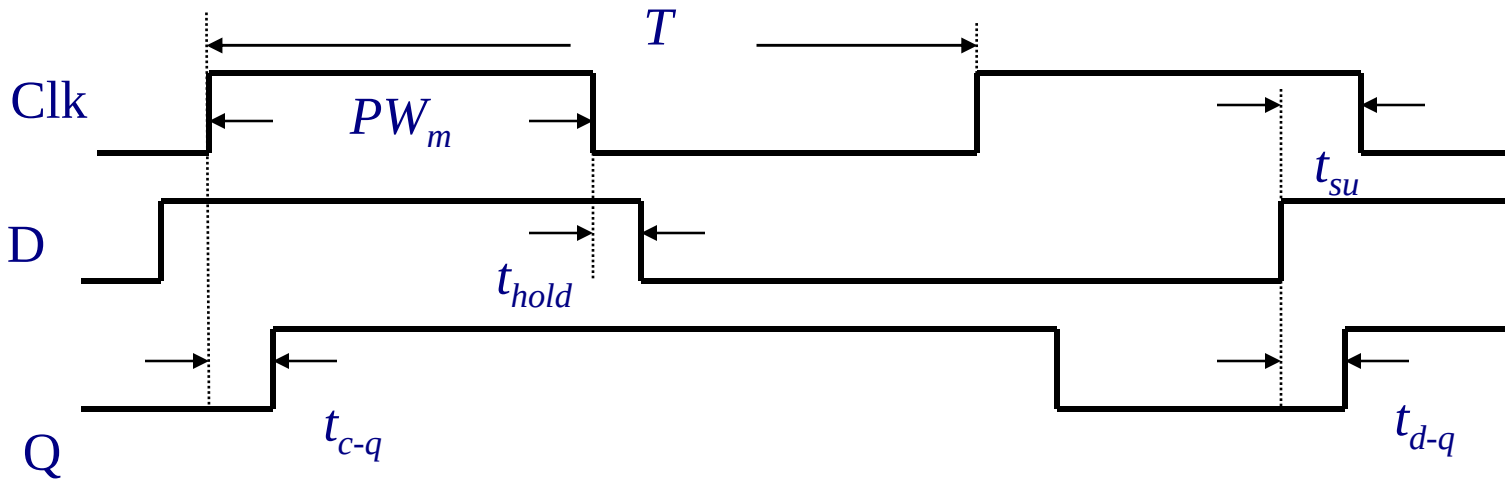
Asynchronous Timing

- Signals transition at arbitrary times
- Can synchronize asynchronous signals by detecting events and by introducing latencies into the data stream synchronized to a local CLK
- Alternatively, a self-timed asynchronous design approach may be adopted—communication between modules achieved not through a CLK but through a handshaking protocol that ensures proper ordering of operation
- Computations can be performed at speeds determined solely by the latency of the logic---no need to manage CLK skew!
- Increased complexity + overhead in communication, impacts performance. Also, CAD methodology is more complex.

Timing Definitions…

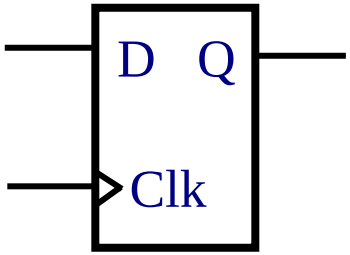
Latch Parameters

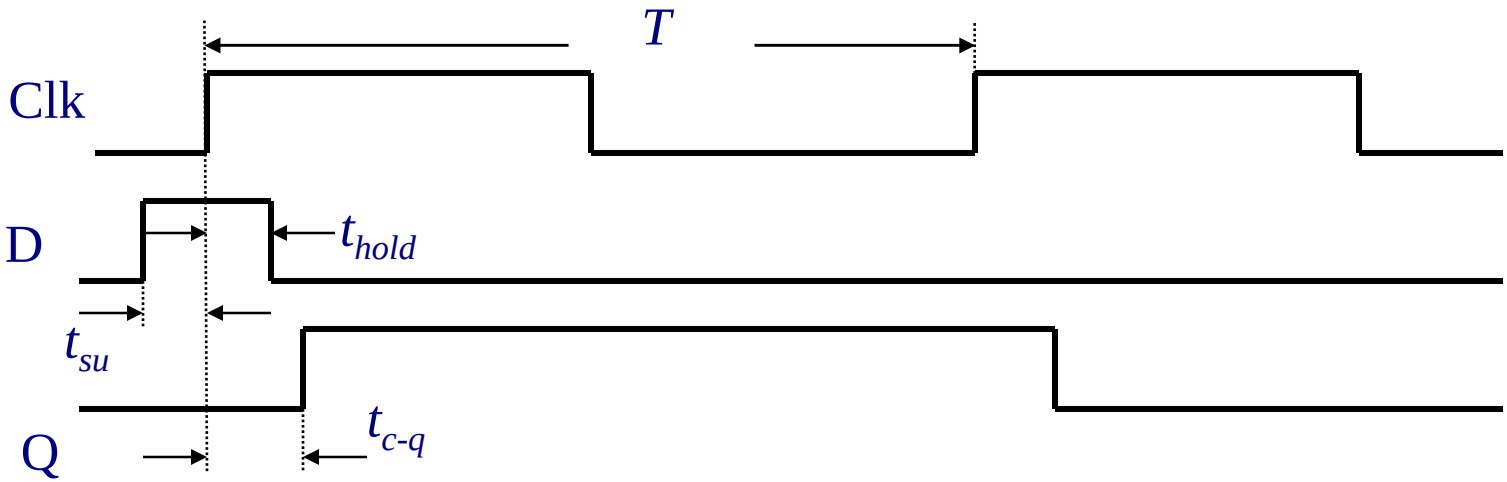




Delays can be different for rising and falling data transitions

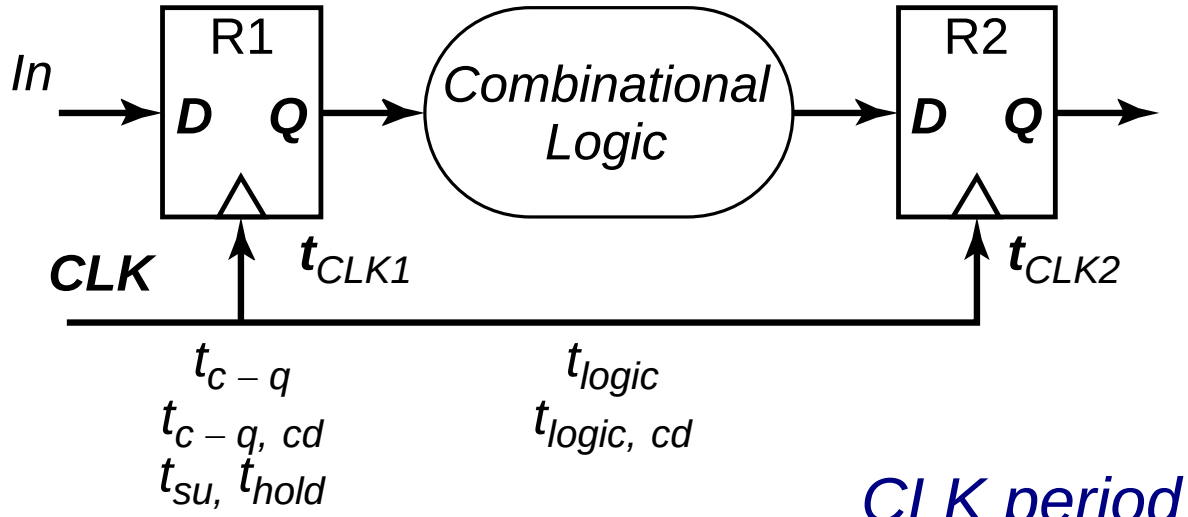
Register Parameters





Delays can be different for rising and falling data transitions

Timing Constraints



CLK period constraint:

$$T_{CLK} > t_{c\text{-}q} + t_{logic} + t_{su}$$

Hold time constraint:

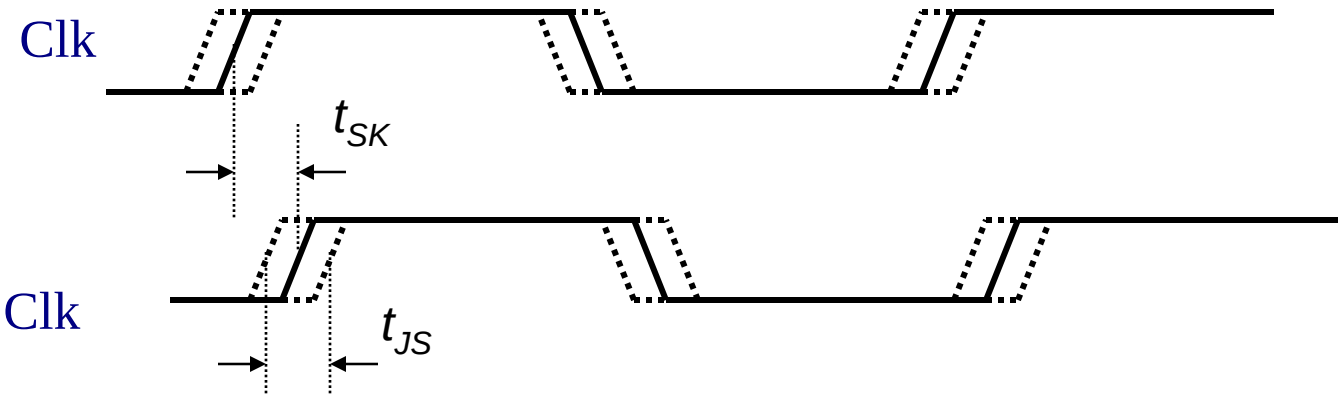
$$t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} > t_{hold}$$

Worst case is when receiving edge arrives late
Race between data and clock

Clock Nonidealities

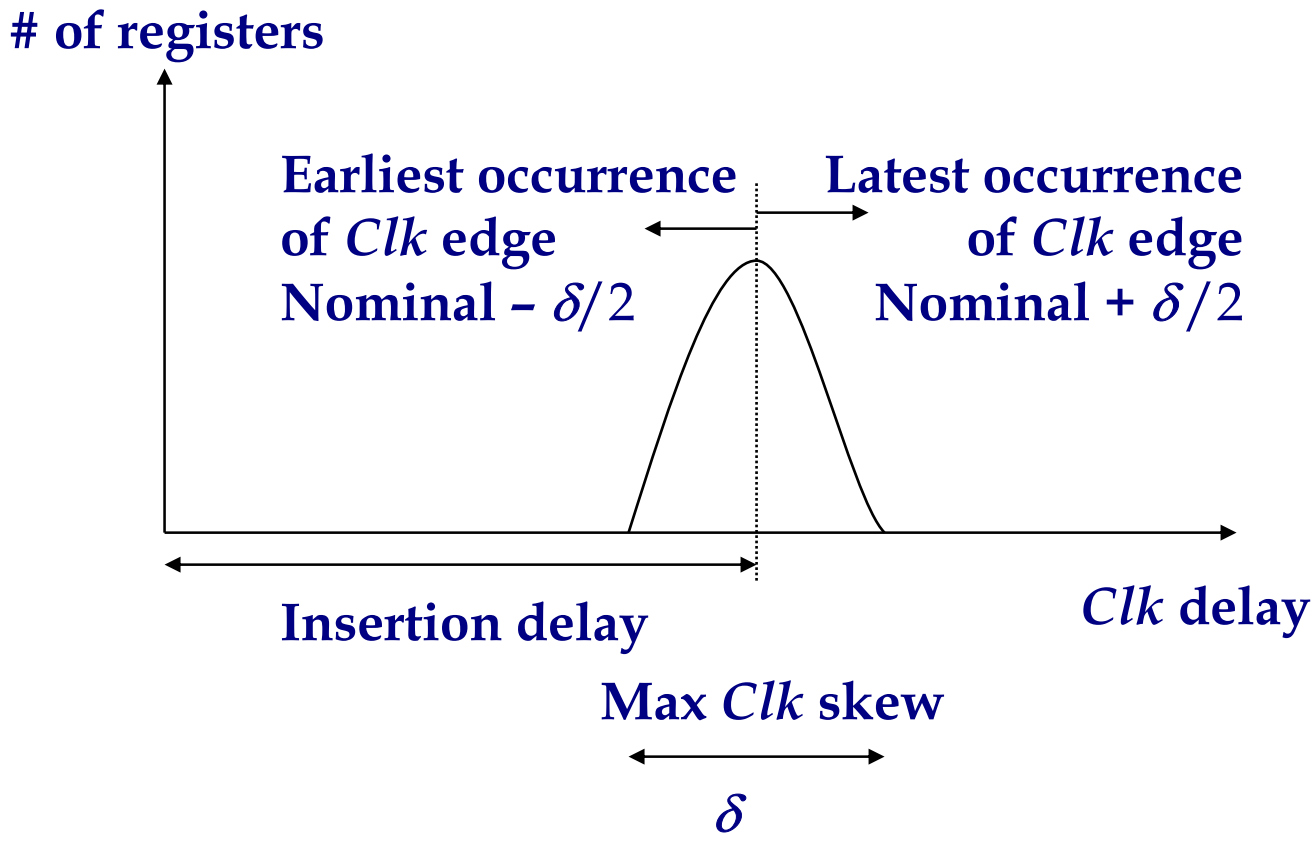
- **Clock skew**
  - Spatial variation in temporally equivalent clock edges; deterministic + random, t_{SK}
- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) t_{JS}
  - Long term t_{JL}
- **Variation of the pulse width**
  - Important for level sensitive clocking

Clock Skew and Jitter

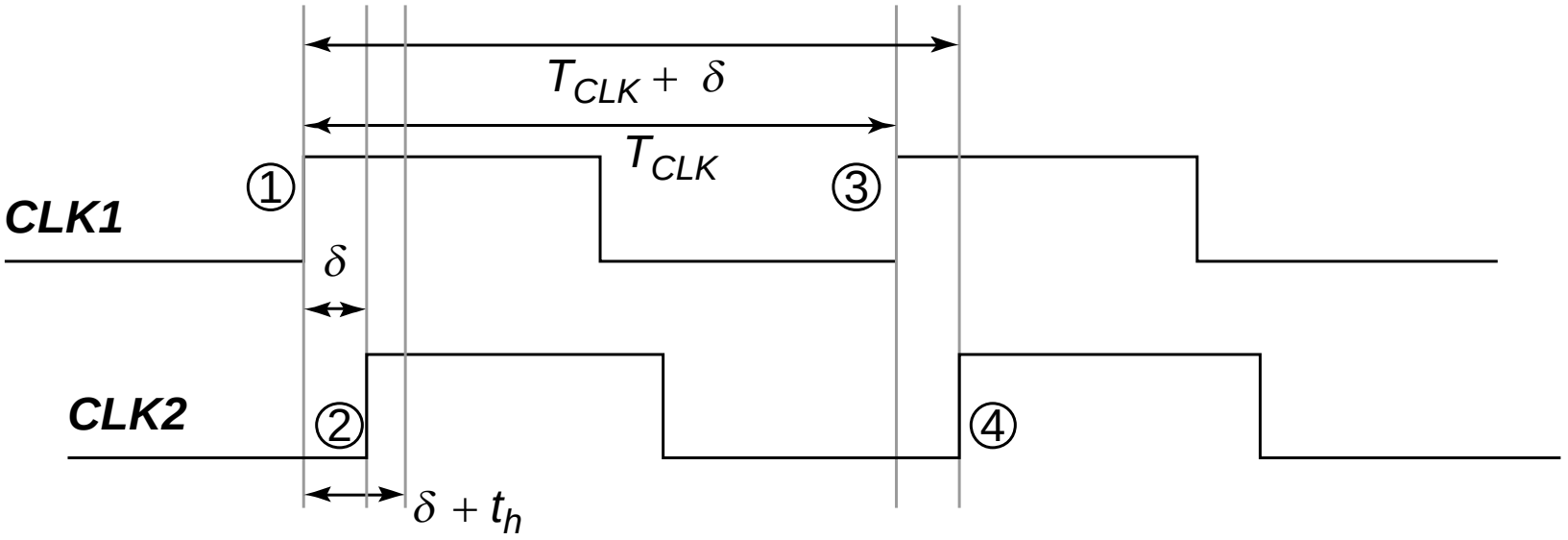


- Both skew and jitter affect the effective cycle time
- Only skew affects the race margin

Clock Skew

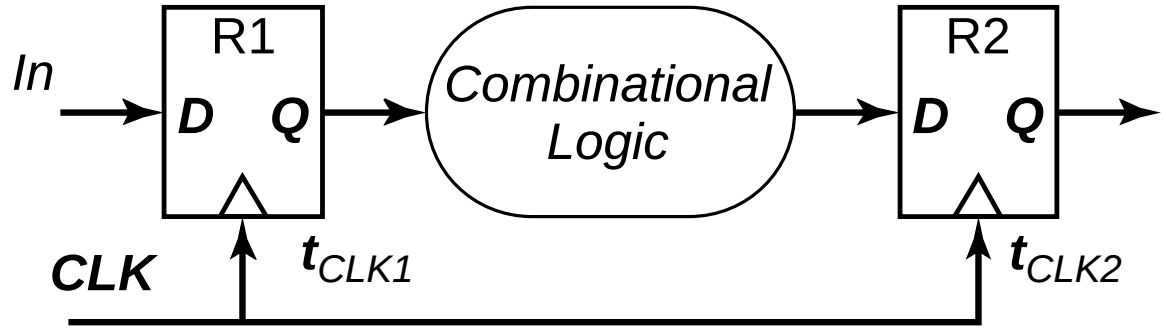

# of registers
Earliest occurrence of *Clk* edge Nominal – $\delta/2$
Latest occurrence of *Clk* edge Nominal + $\delta/2$
Insertion delay
Clk delay
Max *Clk* skew
δ

Positive Skew



Launching edge arrives before the receiving edge

Timing Constraints: +ve Skew



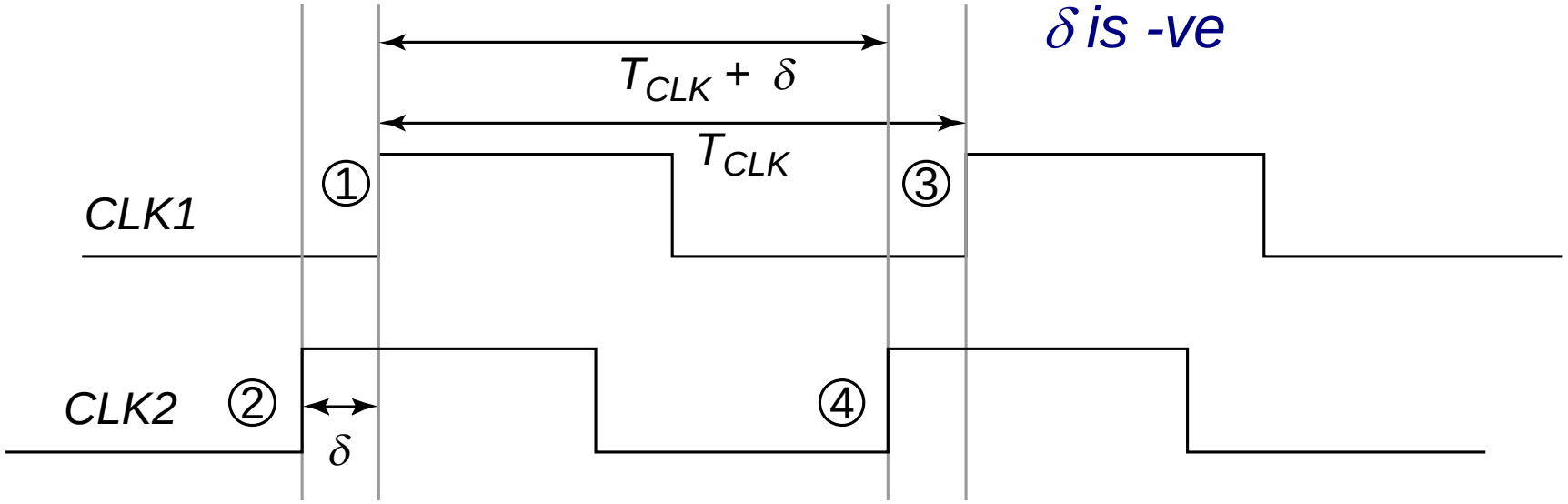
Minimum cycle time:

$$T_{CLK} + \delta = t_{c\text{-}q} + t_{su} + t_{logic}$$

CLK skew can improve performance?
Watch out for RACE problems….(if minimum logic delay is small, inputs to R2 can change before the CLK2 edge at ②)

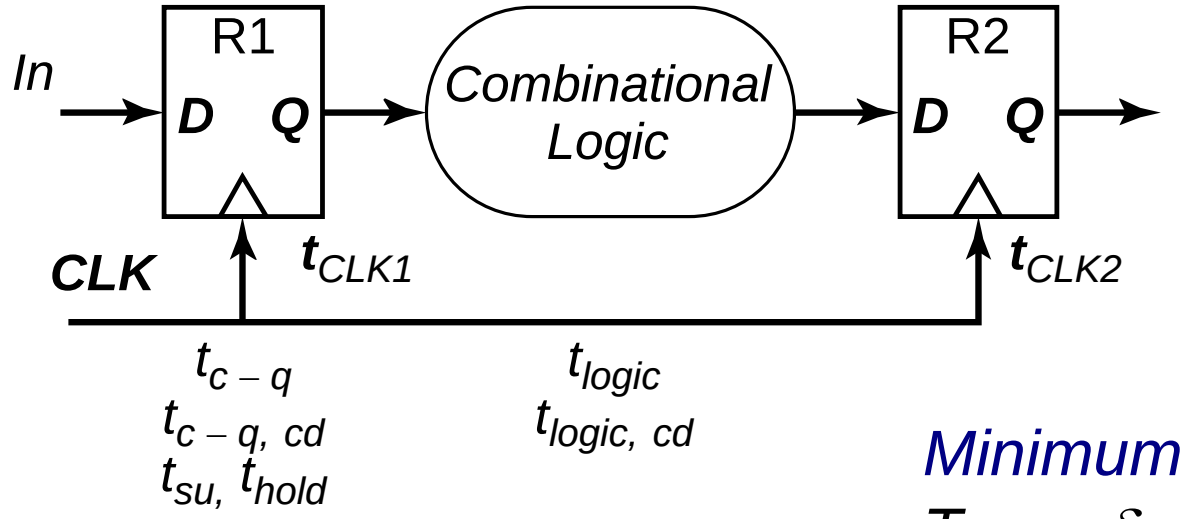
Minimum delay constraint: $\delta + t_{hold} < t_{c\text{-}q,cd} + t_{logic,cd}$

Negative Skew



Receiving edge arrives before the launching edge

Timing Constraints: -ve Skew



Minimum cycle time:

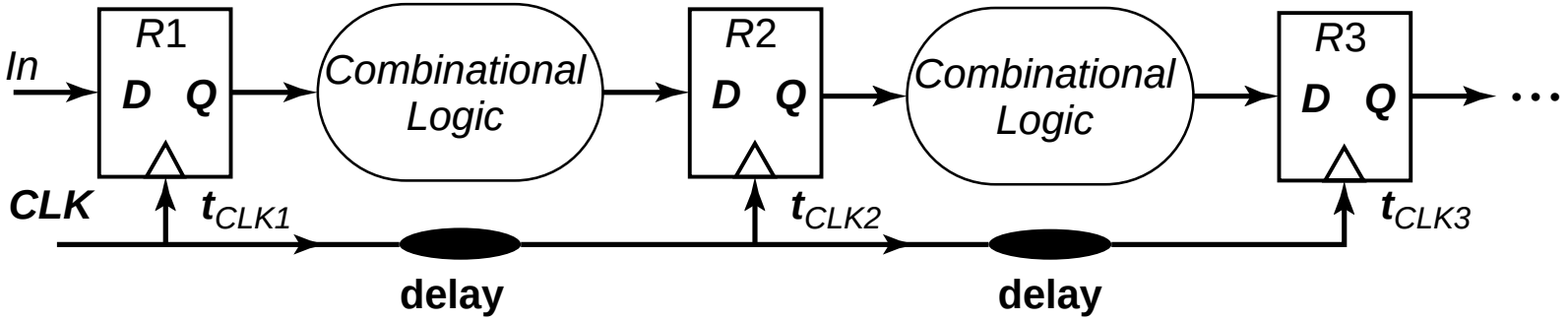
$$T_{CLK} - \delta = t_{c\text{-}q} + t_{su} + t_{logic}$$

-ve skew adversely affects the performance….

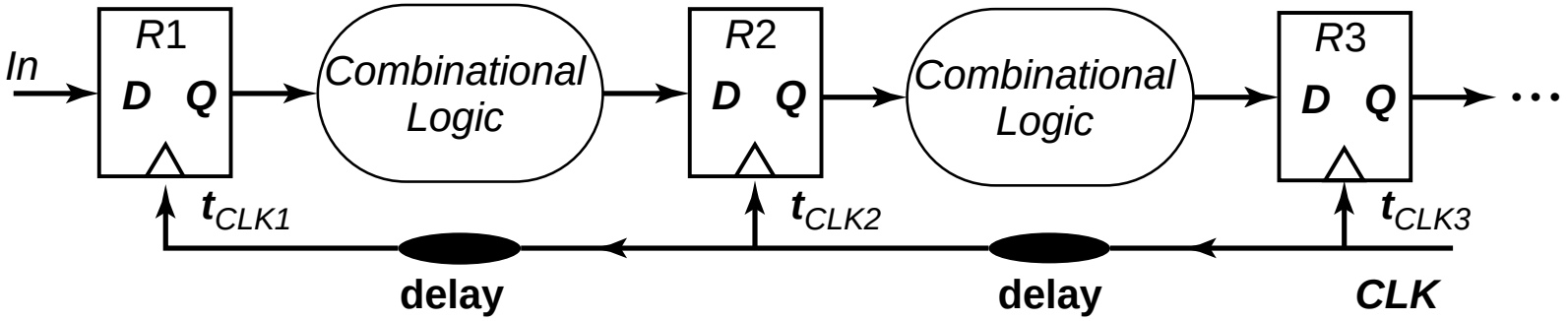
Minimum delay constraint: $\delta + t_{hold} < t_{c\text{-}q,cd} + t_{logic,cd}$*: eliminates RACE*

Positive and Negative Skew

CLK and data in the same direction



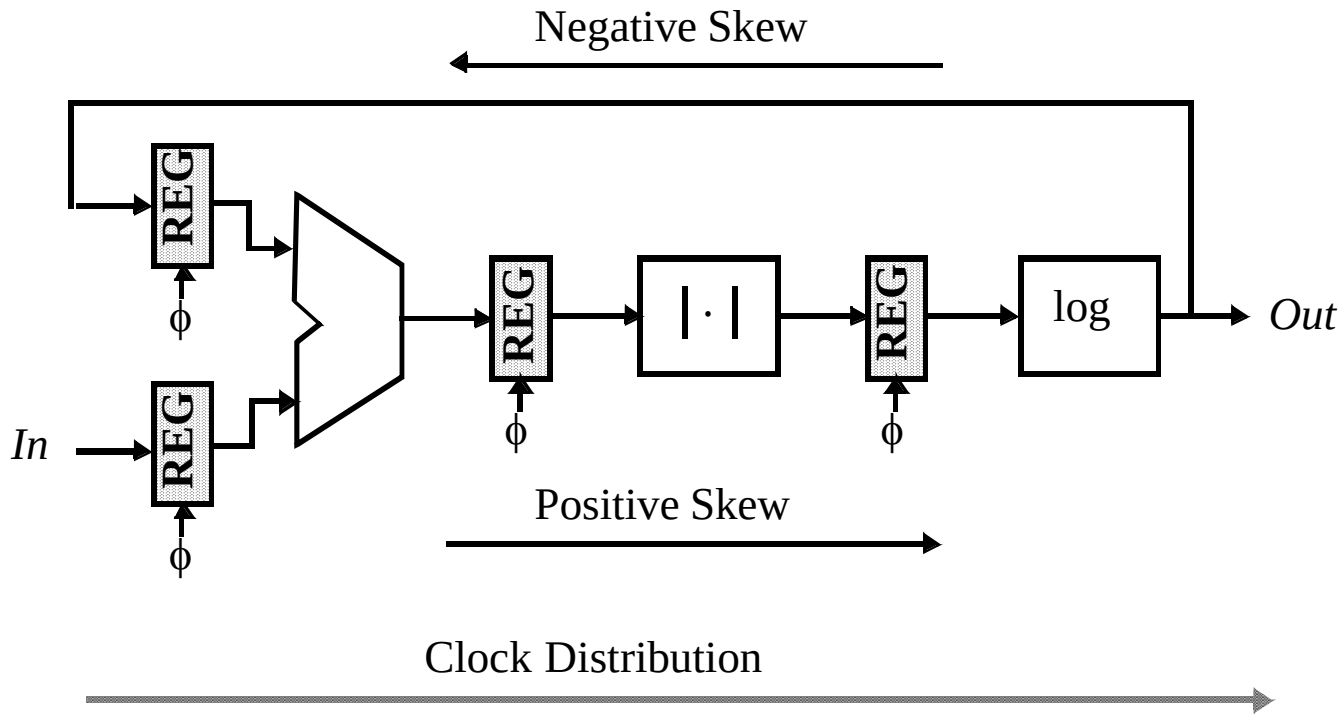
(a) Positive skew



(b) Negative skew

CLK and data in the opposite direction

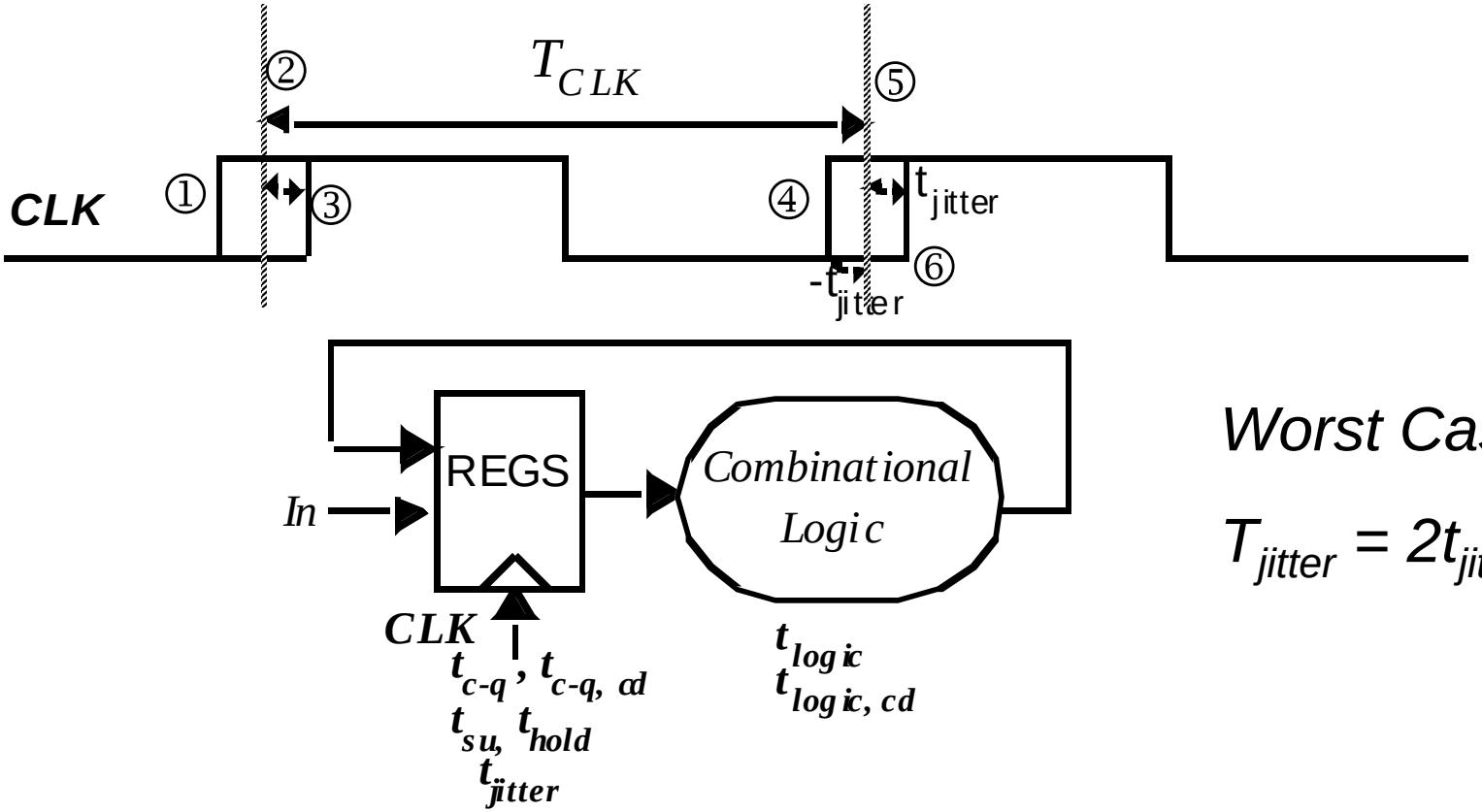
How to counter Clock Skew?



Data and Clock Routing

Must account for the worst case skew…

Impact of Jitter



Worst Case:

$T_{jitter} = 2t_{jitter}$

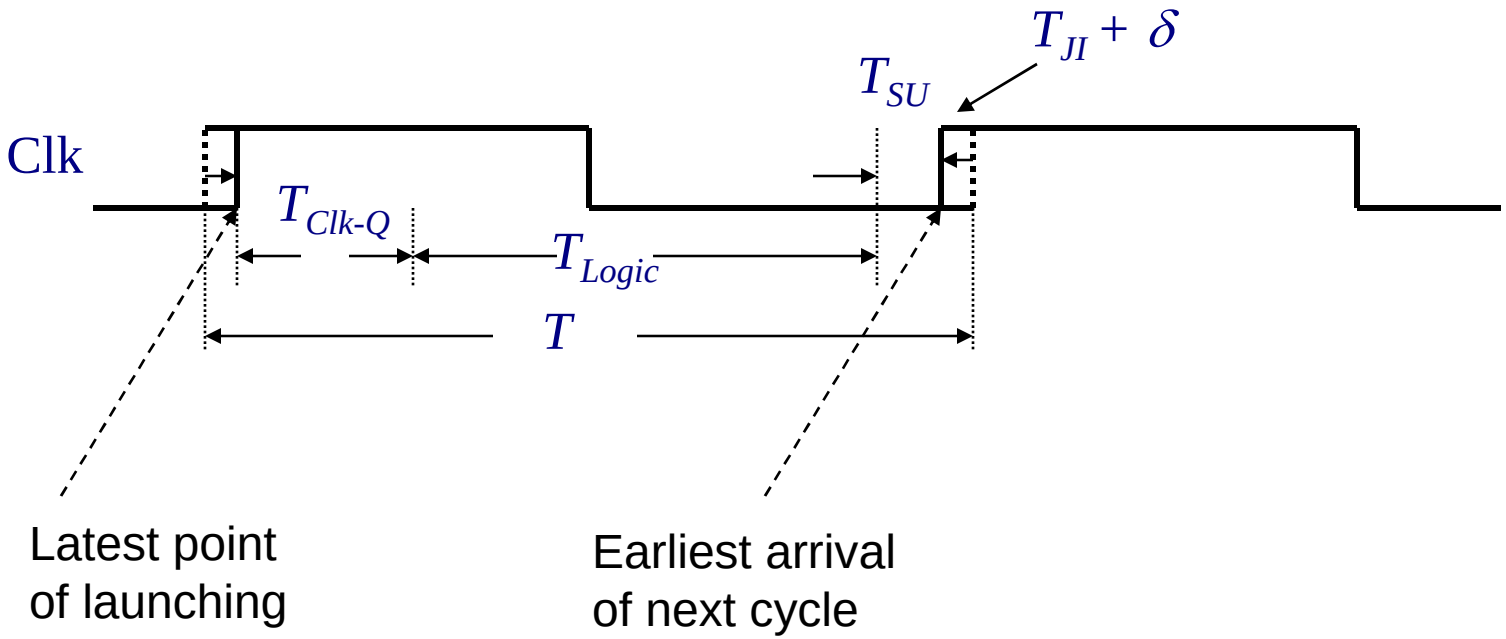
Absolute jitter = t_{jitter} = worst case variation of a CLK edge at a given location w.r.t. an ideally periodic reference CLK edge

Cycle-to-Cycle Jitter = T_{jitter} = time varying deviations of a single clock period relative to an ideal reference CLK:

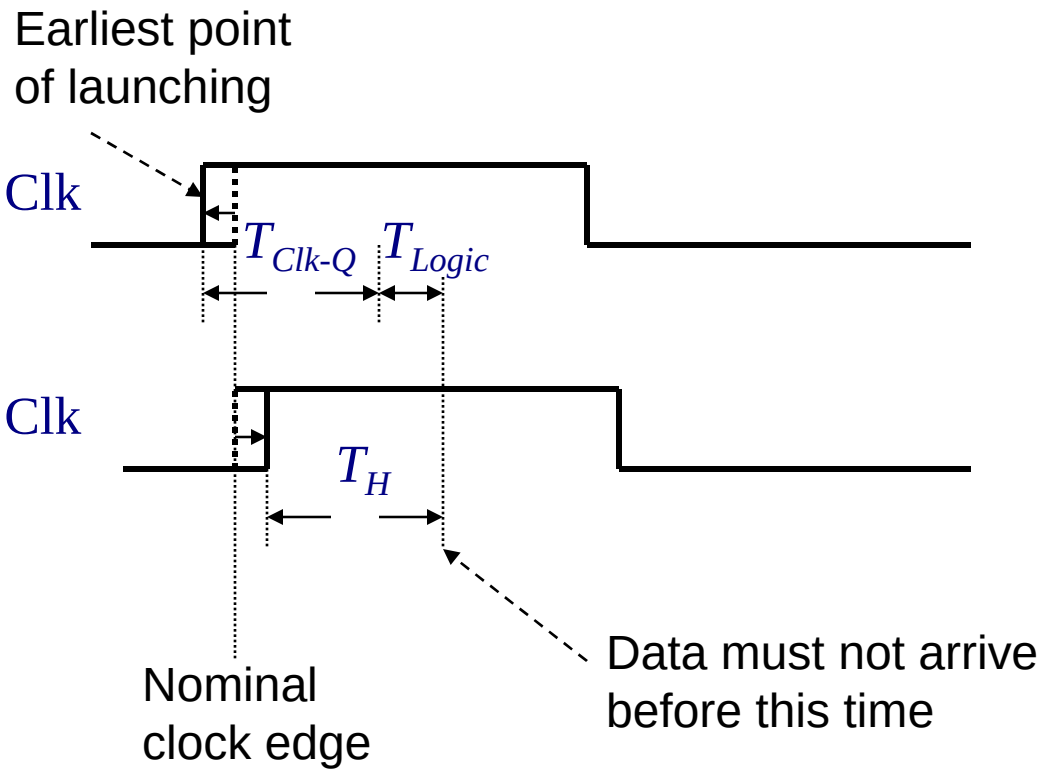
$$T^{i}_{jitter}(n) = t^{i}_{clk,n+1} + t^{i}_{clk,n} - T_{clk}$$

Arrival time of the nth CLK edge at i

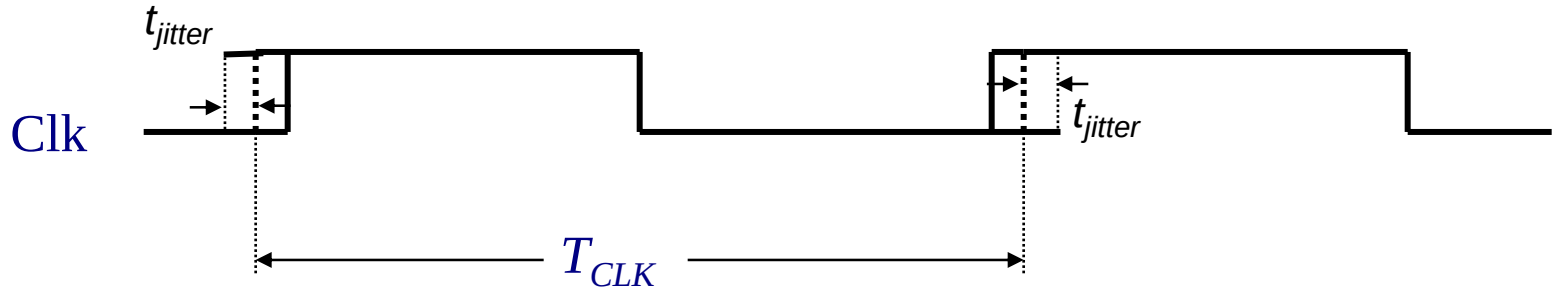
Longest Logic Path in Edge-Triggered Systems



Shortest Path



Impact of Jitter on Clock Constraints in Edge-Triggered Systems



Jitter directly impacts the performance of sequential systems:

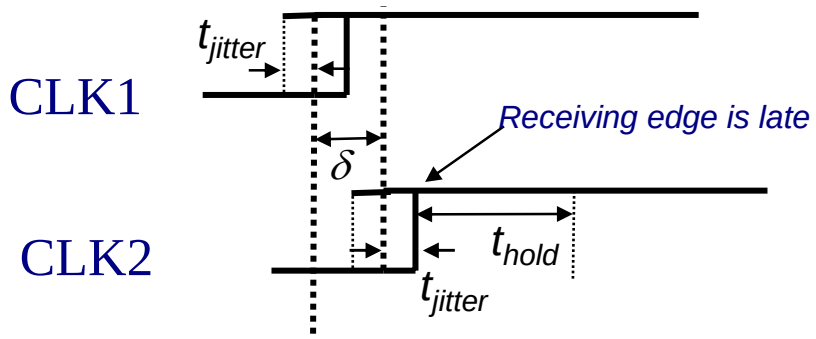
$$T_{CLK} - 2t_{jitter} > t_{c\text{-}q} + t_{logic} + t_{su}$$

Combined effects of Skew and Jitter:

$$T_{CLK} + \delta - 2\,t_{jitter} > t_{c\text{-}q} + t_{logic} + t_{su}$$

Skew can be either positive or negative. +ve skew improves performance, jitter always degrades performance

Impact of Jitter and Skew on Minimum Delay Constraints in Edge-Triggered Systems



If launching edge (CLK1) is early and receiving edge (CLK2) is late:

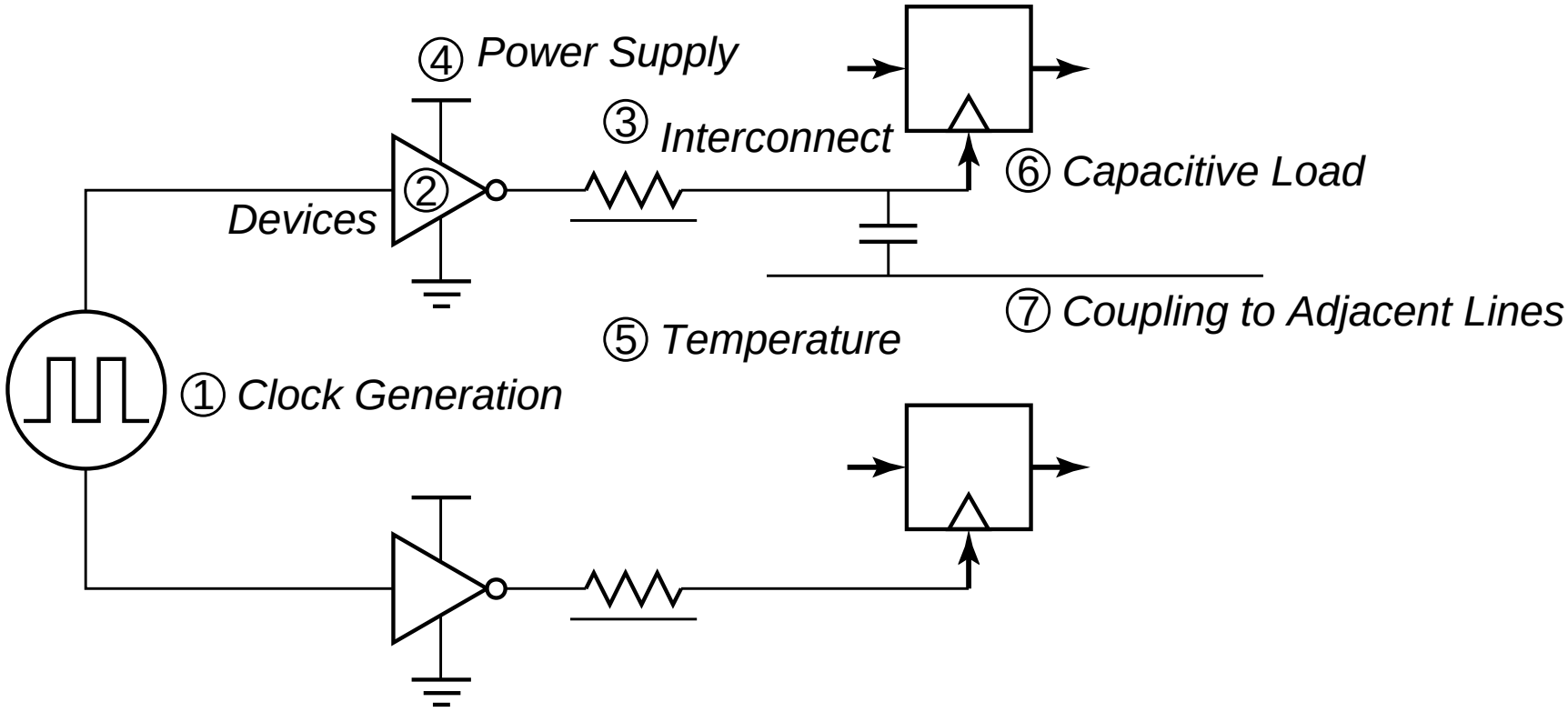
$$t_{hold} + \delta + 2t_{jitter} < t_{c\text{-}q,\,cd} + t_{logic,\,cd}$$

Minimum logic delay constraint:

$$\delta < t_{c\text{-}q,cd} + t_{logic,cd} - t_{hold} - 2t_{jitter}$$

Acceptable skew is reduced by the jitter of the two signals

Clock Uncertainties



Sources of clock uncertainty