EE 431: COMPUTER-AIDED DESIGN OF VLSI DEVICES

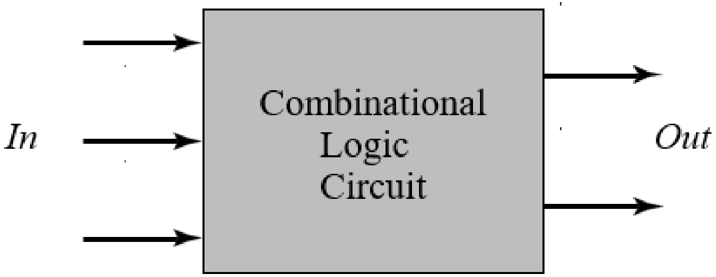
Sequential Circuits

Nishith N. Chakraborty

November, 2024

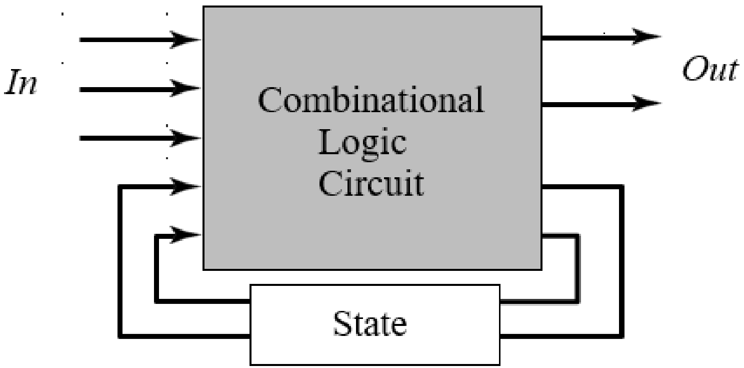
Adapted from: Dr. Garrett Rose, Dr. Ahmedullah aziz University of Tennessee

COMBINATIONAL VS SEQUENTIAL LOGIC



Combinational

Output = f(In)

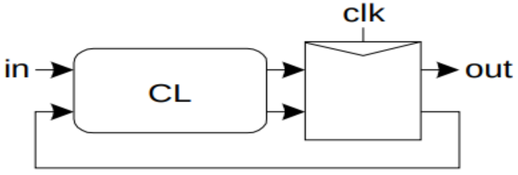


Sequential

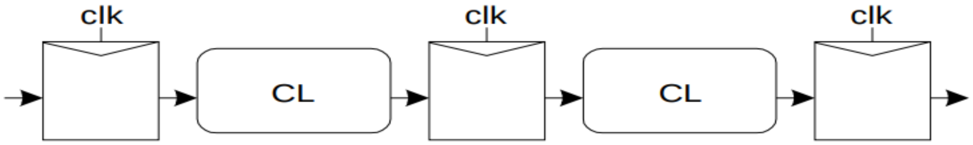
Output = f(In, Previous In)

SEQUENCING

- Combinational logic
  - Output depends on current inputs
- Sequential logic
  - Output depends on current and previous inputs
  - Requires separating previous, current, future
  - Called state or tokens
  - Ex: FSM, pipeline



Finite State Machine



Pipeline

SEQUENCING CONTD.

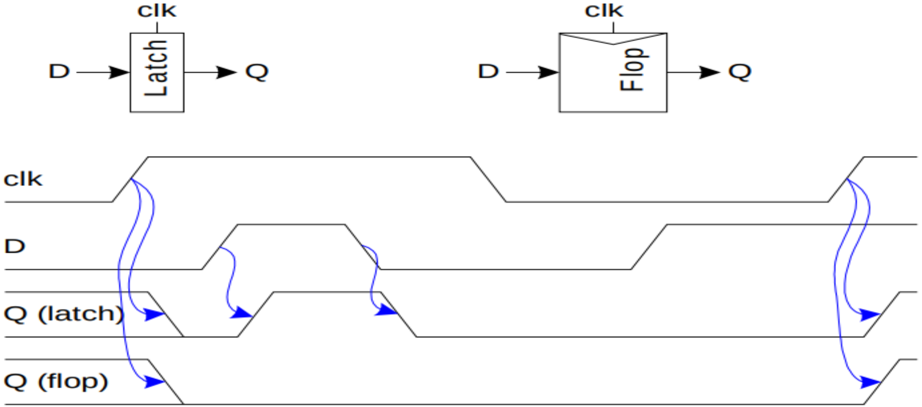
- If tokens moved through pipeline at constant speed, no sequencing elements would be necessary
- Ex: fiber-optic cable
  - Light pulses (tokens) are sent down cable
  - Next pulse sent before first reaches end of cable
  - No need for hardware to separate pulses
  - But dispersion sets min time between pulses
- This is called *wave pipelining* in circuits
- In most circuits, dispersion is high
  - Delay fast tokens so they don’t catch slow ones.

SEQUENCING OVERHEAD

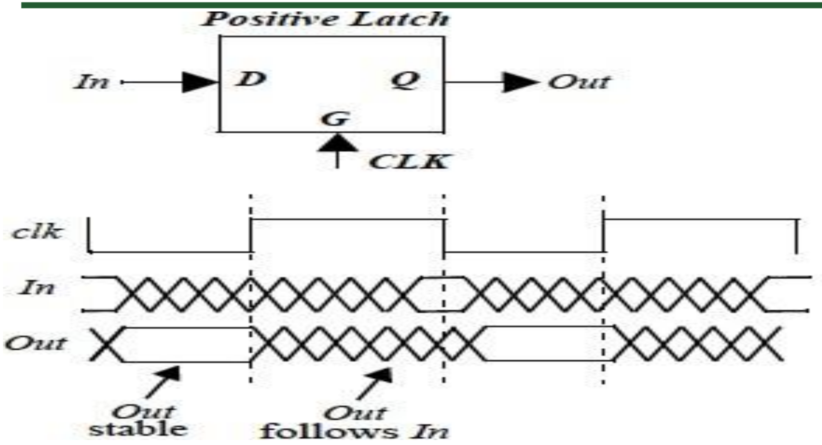
- Use flip-flops to delay fast tokens so they move through exactly one stage each cycle.
- Inevitably adds some delay to the slow tokens
- Makes circuit slower than just the logic delay
    - Called sequencing overhead
- Some people call this clocking overhead
    - But it applies to asynchronous circuits too
    - Inevitable side effect of maintaining sequence

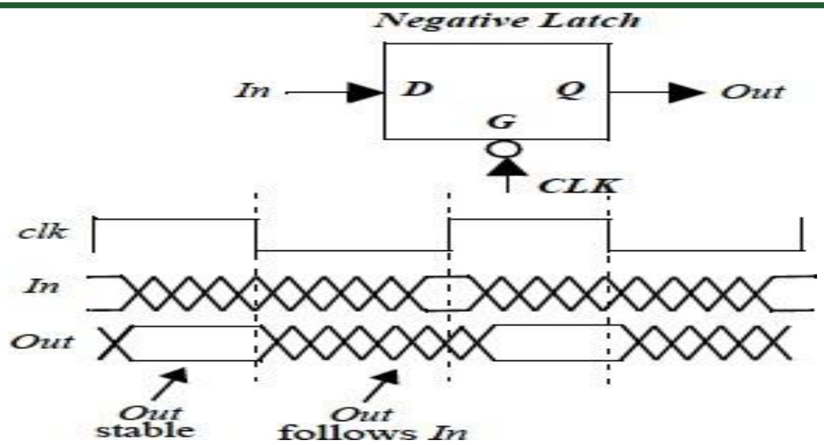
SEQUENCING ELEMENTS

- Latch: Level sensitive
  - a.k.a. transparent latch, D latch
- Flip-flop: edge triggered
  - A.k.a. master-slave flip-flop, D flip-flop, D register
- Timing Diagrams
  - Transparent
  - Opaque
  - Edge-trigger



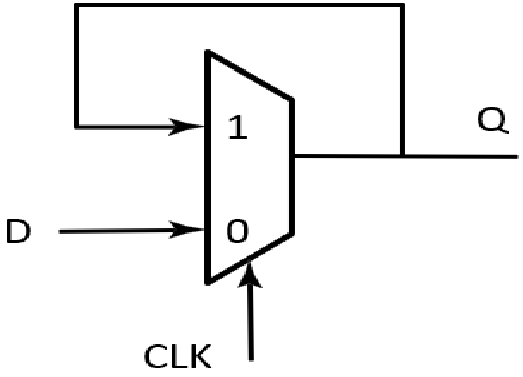
LATCH



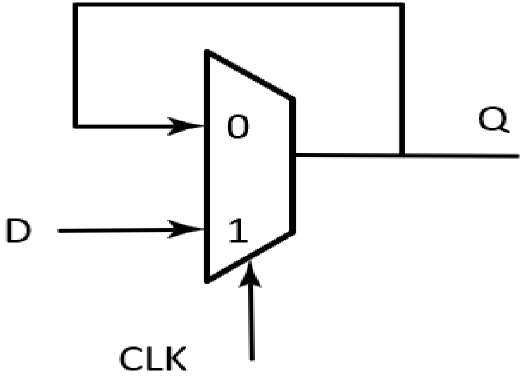


- A **Positive Latch** is a level-sensitive circuit that passes the D input to the Q output when the clock signal is high. While the clock is high, the Latch is said to be in **transparent mode**.
- When the clock is low, the input data sampled on the falling edge of the clock is held stable at the output for the entire phase, and the latch is in **hold mode**.
- A **Negative Latch** does the exact opposite (passes the D input to the Q output when the clock signal is low).

LATCH FUNCTIONALITY: LOGICAL VIEW

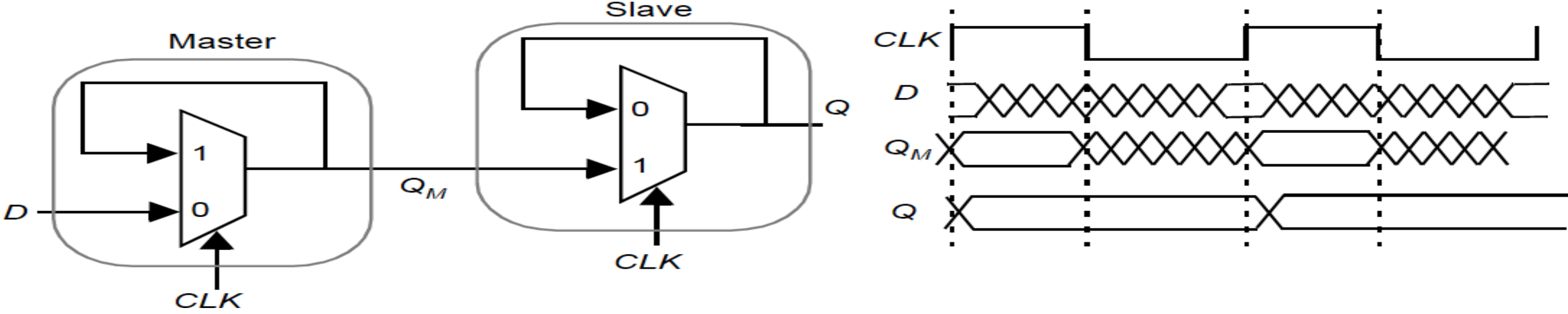


Negative latch
(transparent when CLK= 0)



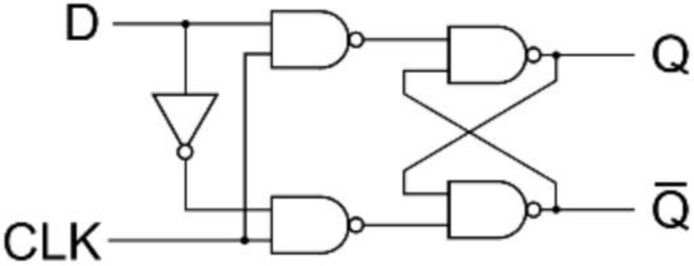
Positive latch
(transparent when CLK= 1)

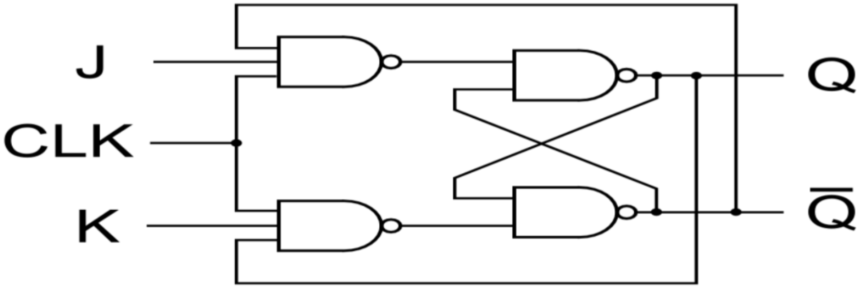
REGISTER/FLIP-FLOP

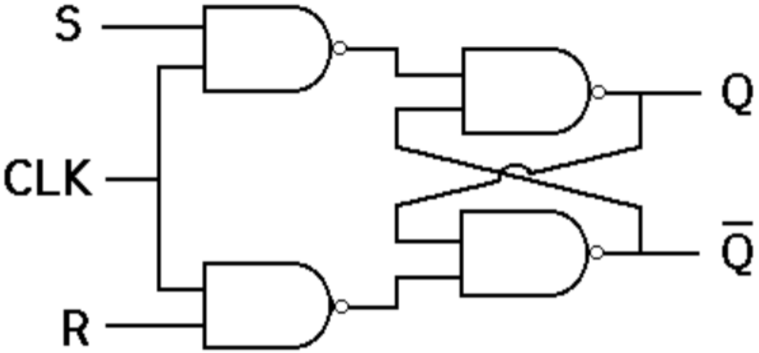


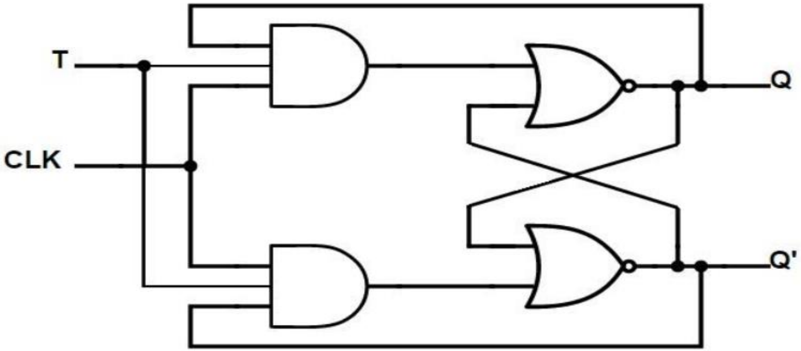
- A Register/Flip Flop is triggered by the clock edge. [N.B: Registers and Flip Flops have some differences]
- Contrary to level-sensitive latches, edge-triggered registers only sample the input on a clock transition : 0-to-1 for a **positive edge-triggered register**, and 1-to-0 for a **negative edge-triggered register**.
- A common register configuration is the master-slave structure that cascades a positive and negative latch.

DIFFERENT FLIP-FLOPS

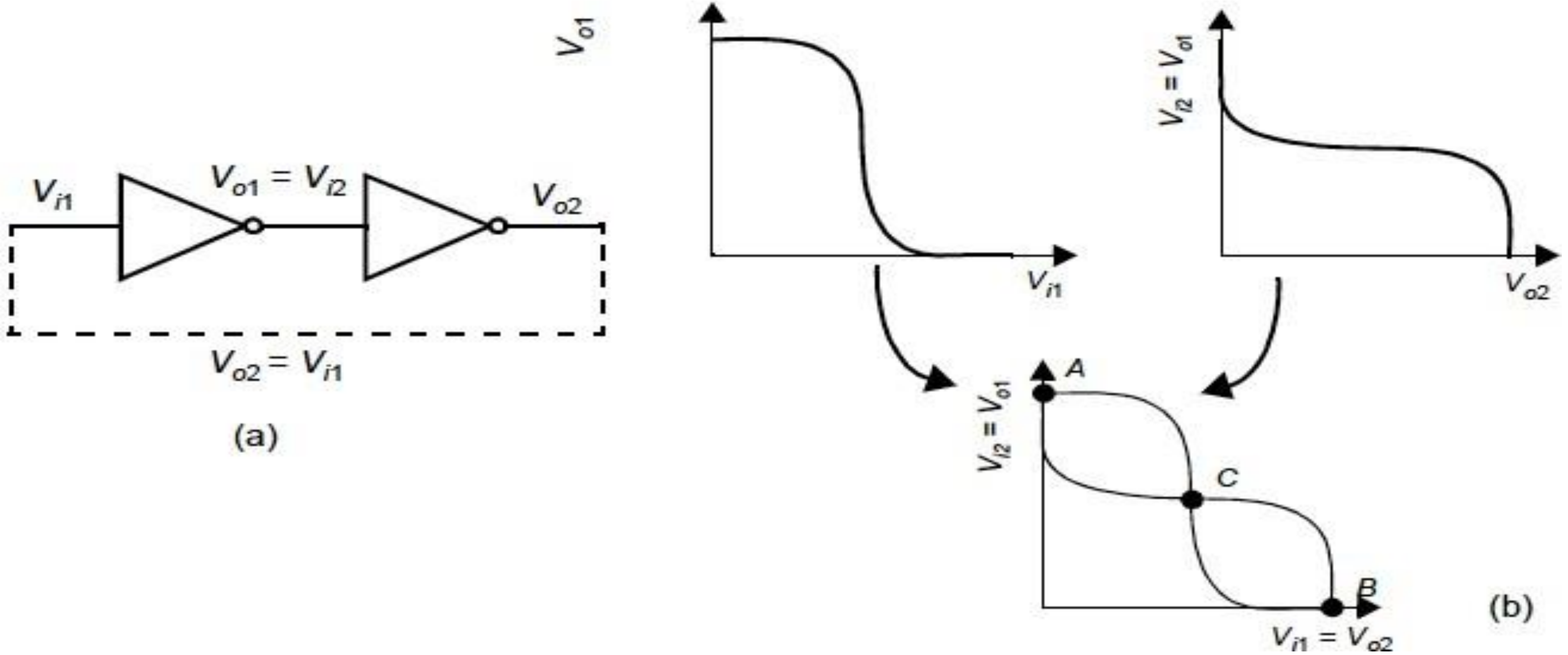




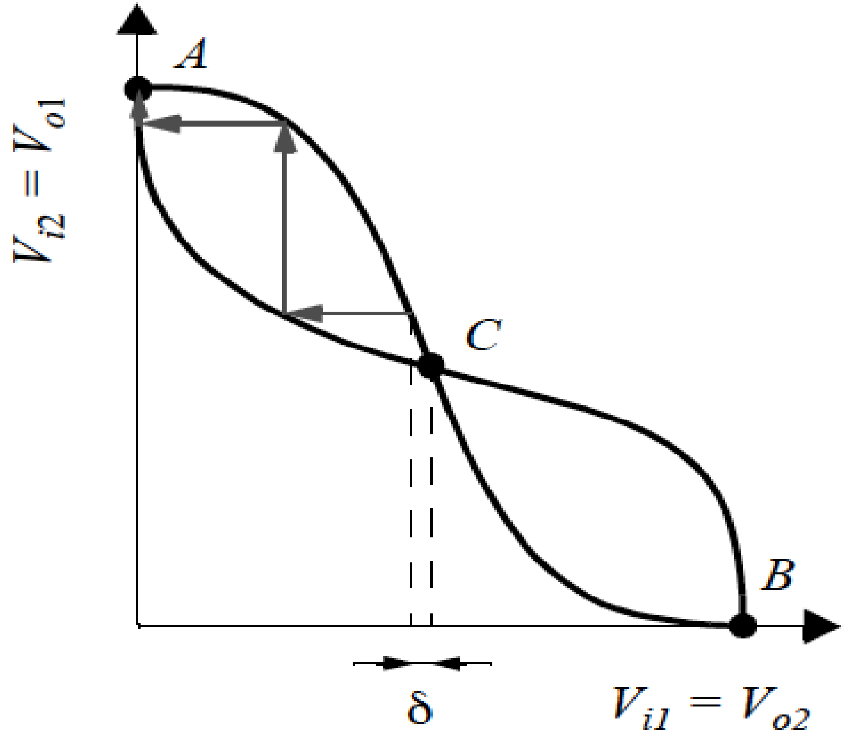


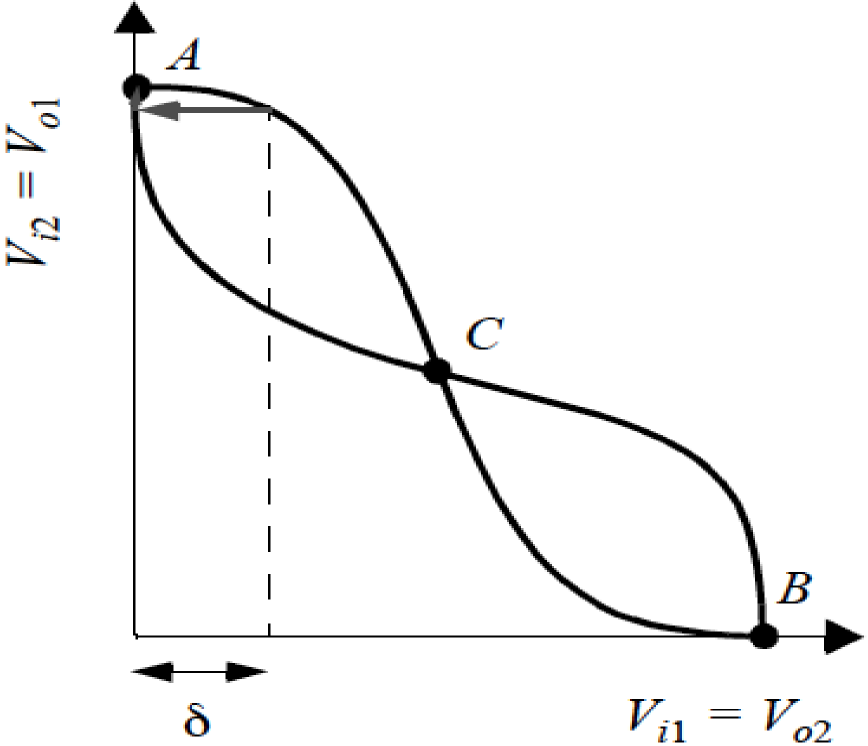


THE BI-STABILITY PRINCIPLE

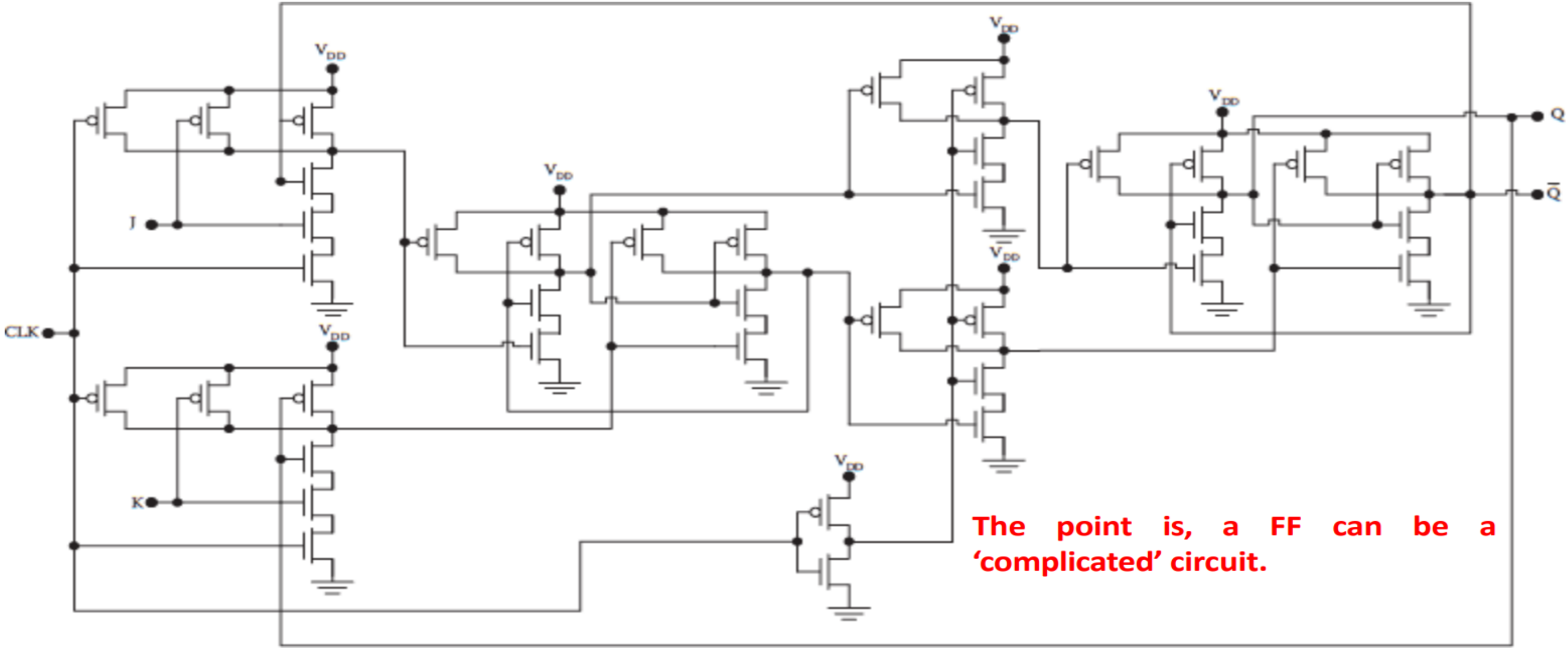


METASTABILITY





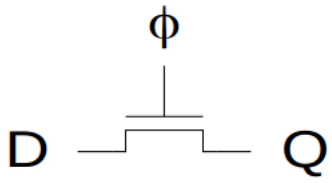
FLIP-FLOP: SNEAK PEEK



The point is, a FF can be a 'complicated' circuit.

LATCH DESIGN: PASS TRANSISTOR LATCH

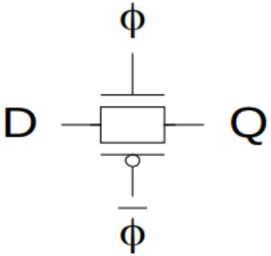
- Pros
  - Tiny
  - Low clock load
- Cons
  - V_t drop
  - Non-restoring
  - Backdriving
  - Output noise sensitivity
  - Dynamic
  - Diffusion input



Used in 1970's

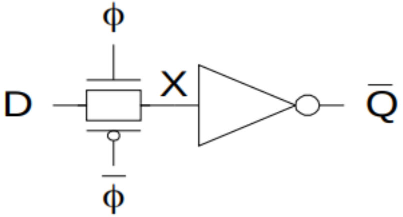
LATCH DESIGN: TRANSMISSION GATE LATCH

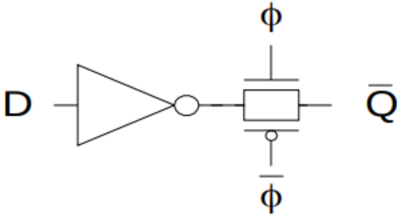
- Pros
  - No V_t drop
- Cons
  - Requires inverted clock
  - Backdriving



LATCH DESIGN: INVERTING BUFFER LATCH

- Pros
  - Restoring
  - No backdriving
  - Fixes either
    - Output noise sensitivity
    - Or diffusion input
- Cons
  - Inverted output

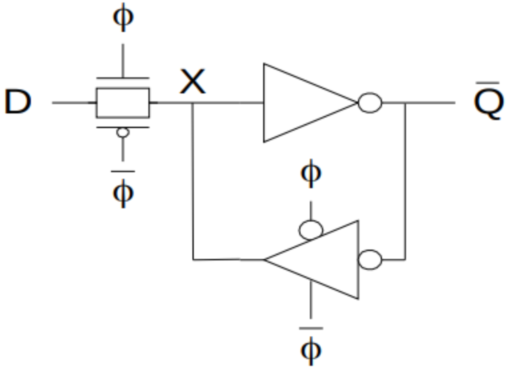




LATCH DESIGN: TRISTATE FEEDBACK LATCH

- Pros
  - Static
- Cons
  - Backdriving risk

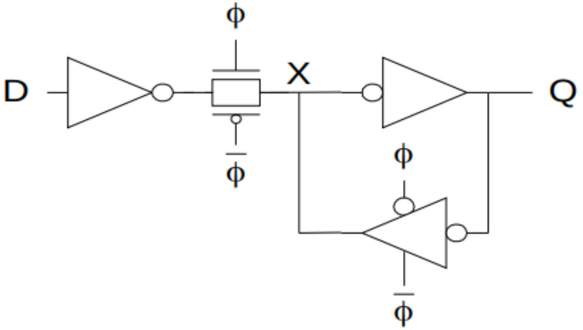
- Static latches are now essential



CAL POLY
SAN LUIS OBISPO

LATCH DESIGN: TRISTATE FEEDBACK LATCH

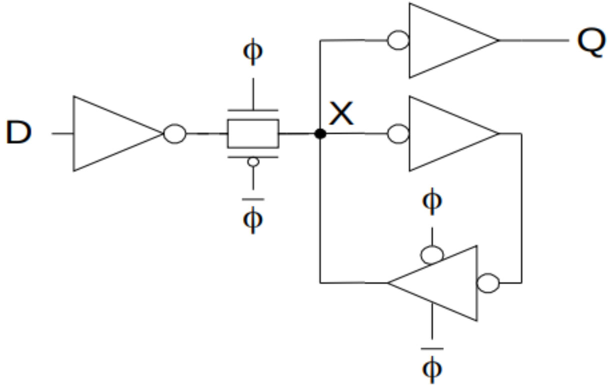
- Buffered input
- Pros
  - Fixes diffusion input
  - Noninverting



LATCH DESIGN: TRISTATE FEEDBACK LATCH

- Buffered output
- Pros
  - No backdriving

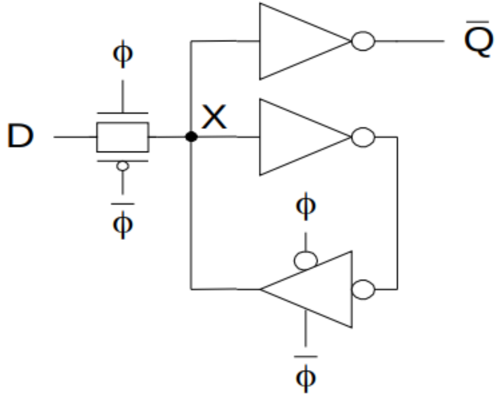
- Widely used in standard cells
  - Pro: Very robust (most important)
  - Con: Rather large
  - Con: Rather slow (1.5 – 2 FO4 delays)
  - Con: High clock loading



LATCH DESIGN: DATAPATH LATCH

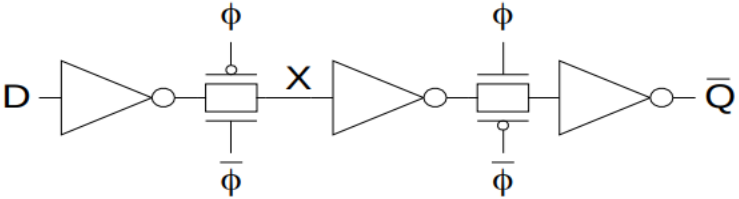
- Buffered output
- Pros
  - Smaller, faster

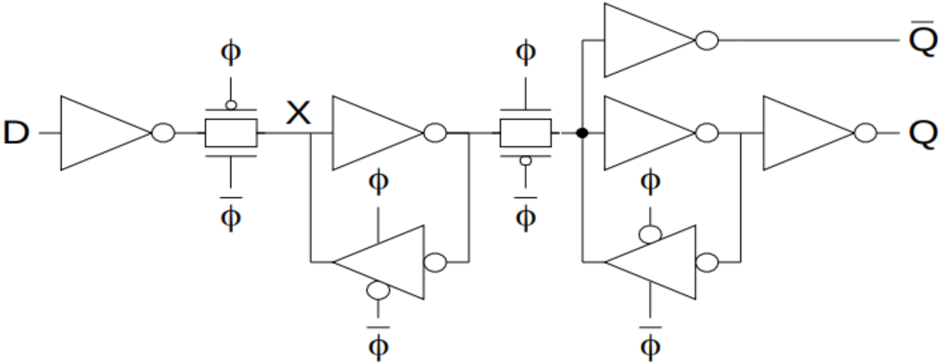
- Cons:
  - Unbuffered input



FLIP-FLOP DESIGN

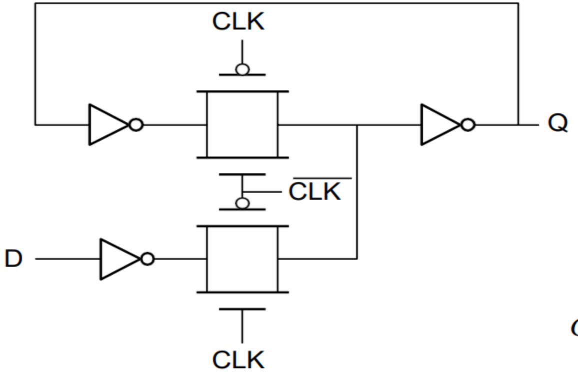
- Flip-flop is built as pair of back-to-back latches



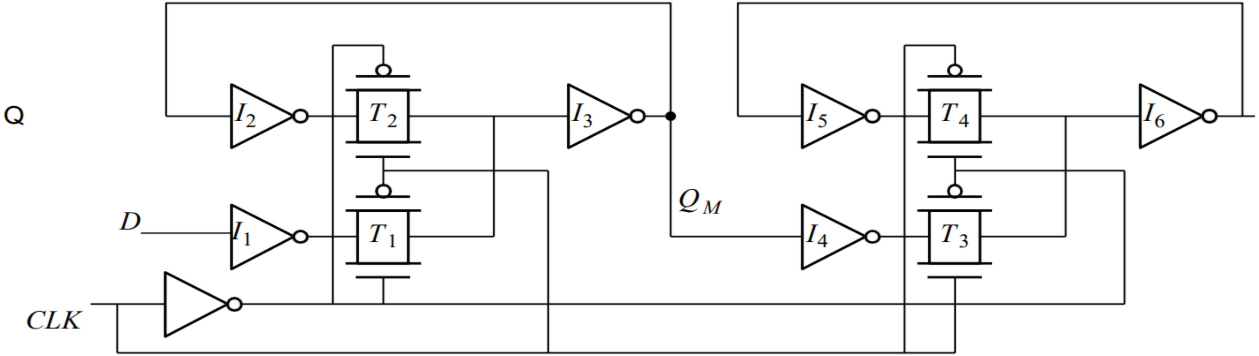


EXAMPLE: TRANSMISSION GATE BASED LATCH & FF

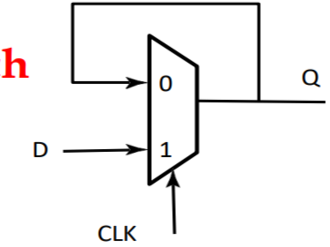
Positive latch
(transparent when CLK= 1)



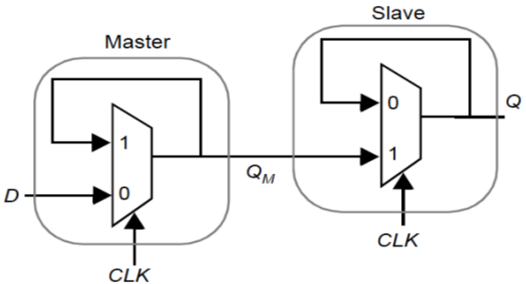
Positive (Rising) Edge Triggered Flip-Flop
(transparent when CLK= 1)



Latch

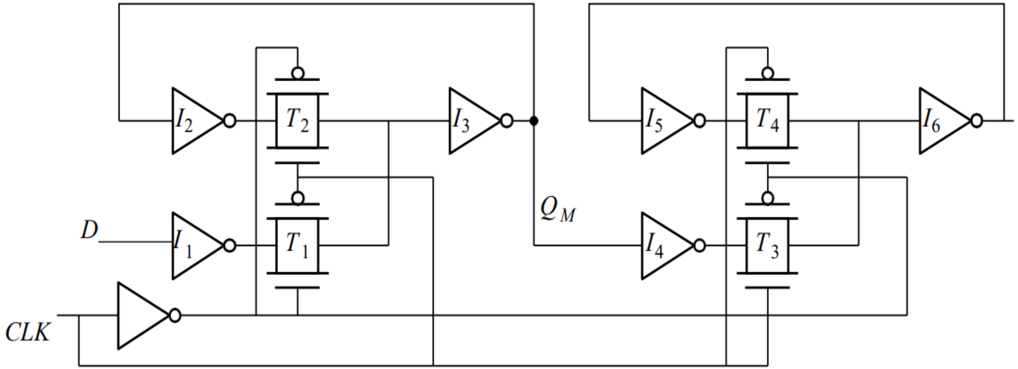


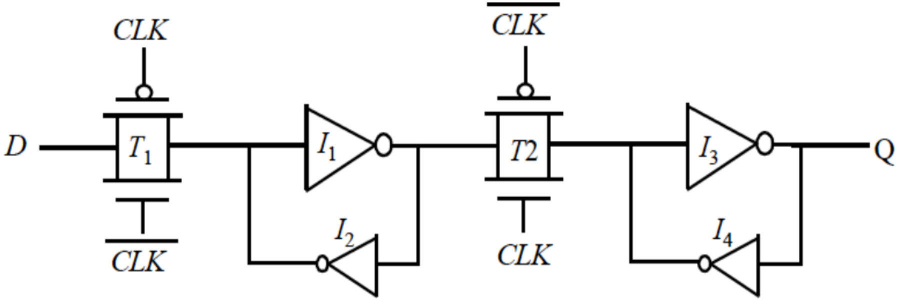
Flip-Flop



CLOCK LOAD REDUCTION

- The drawback of the transmission gate register is the high capacitive load presented to the clock signal. The clock load per register is important, since it directly impacts the power dissipation of the clock network.
- The penalty for the reduced clock load is increased design complexity. The transmission gate (T_1) and its source driver must overpower the feedback inverter (I_2) to switch the state of the cross-coupled inverter. So, proper sizing required.

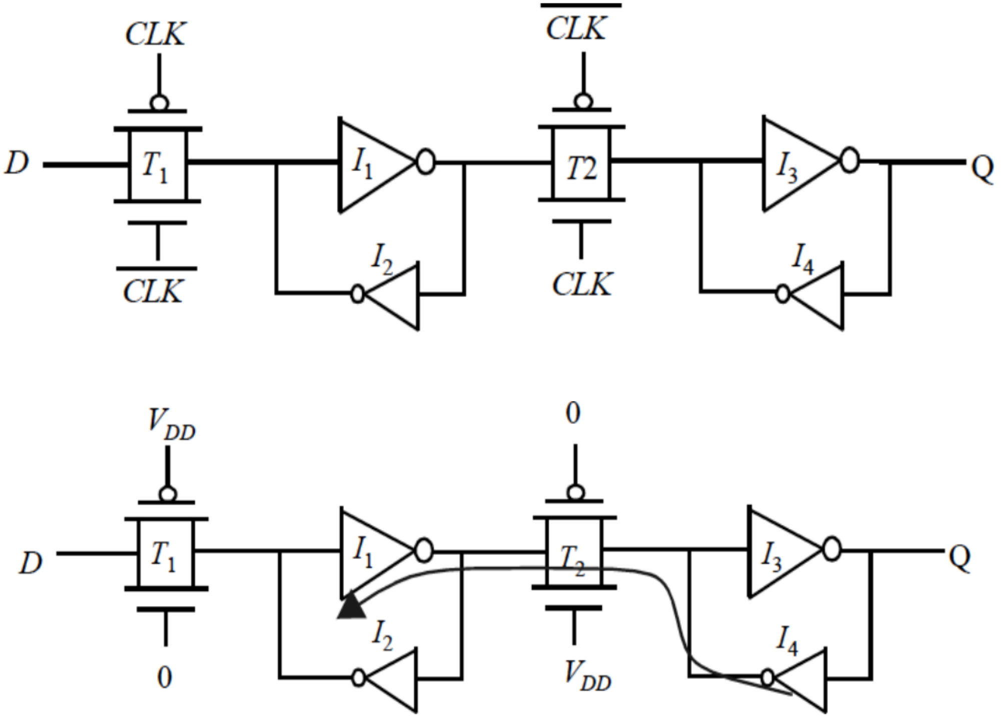




CLOCK LOAD REDUCTION: REVERSE CONDUCTION

- Another problem with this scheme is the reverse conduction. That is, the second stage can affect the state of the first latch. When the slave stage is on, it is possible for the combination of T_2 and I_4 to influence the data stored in I_1-I_2 latch.

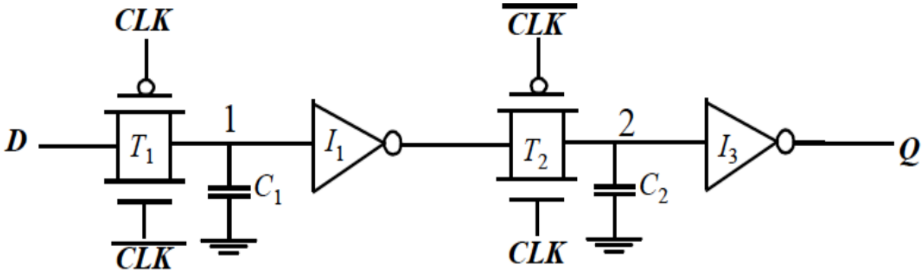
- As long as I_4 is a weak device, this is fortunately not a major problem.

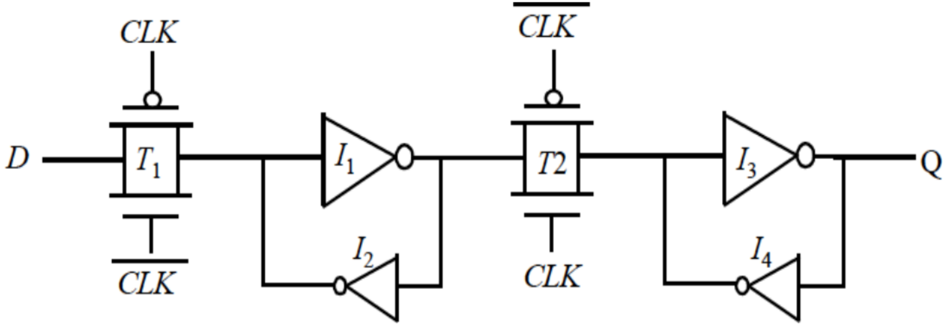


DYNAMIC TRANSMISSION GATE BASED DESIGN

- One important consideration for such a dynamic register is that the storage nodes (i.e., the state) has to be refreshed at periodic intervals to prevent a loss due to charge leakage, due to diode leakage as well as sub-threshold currents.

- Clock overlap is an important concern for this register.



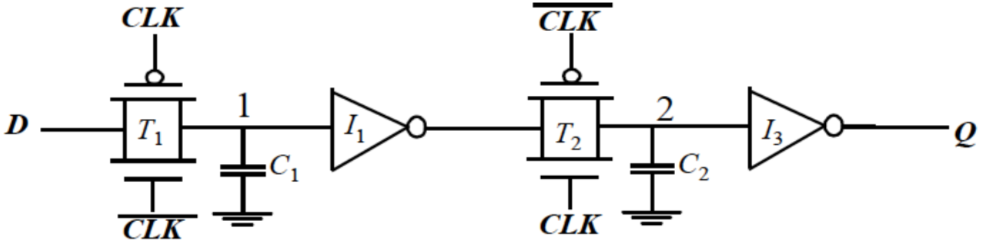


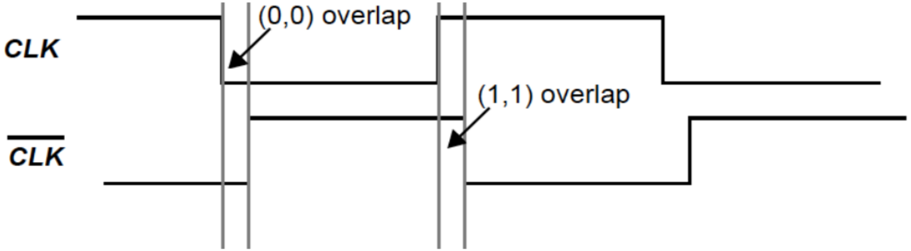
DYNAMIC TRANSMISSION GATE BASED DESIGN

- During the 0-0 overlap period, the NMOS of T_1 and the PMOS of T_2 are simultaneously on, creating a direct path for data to flow from the D input of the register to the Q output. This is known as a race condition. The output Q can change on the falling edge if the overlap period is large — obviously an undesirable effect for a positive edge-triggered register.

- The same is true for the 1-1 overlap region, where an input-output path exists through the PMOS of T_1 and the NMOS of T_2.

- We impose special delay and hold time constraint to avoid such a situation.



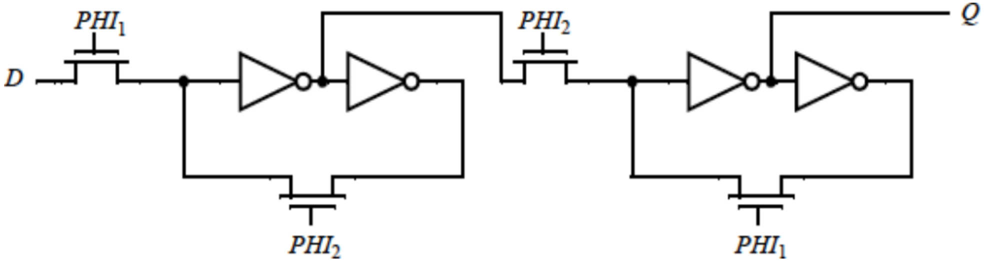


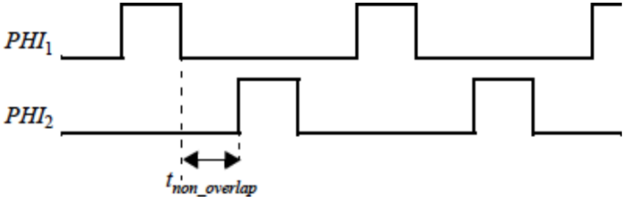
$$t_{overlap0-0} < t_{T1} + t_{I1} + t_{T2}$$

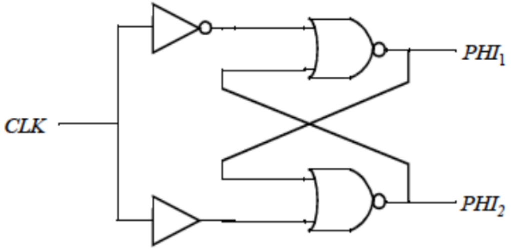
$$t_{hold} > t_{overlap1-1}$$

PSEUDO-STATIC TWO-PHASE D FLIP-FLOP/REGISTER

- Two non-overlapping clocks PHI_1 and PHI_2 can be used to avoid the CLK overlap problem.
- The non-overlap time $t_{non\text{-}overlap}$ between the clocks can be kept large enough, such that no overlap occurs even in the presence of clock-routing delays.
- During the nonoverlap time, the FF is in the high-impedance state—the feedback loop is open, and the input is disconnected. Leakage will destroy the state if this condition holds for too long a time. Hence the name pseudo-static: the register employs a combination of static and dynamic storage approaches depending upon the state of the clock.





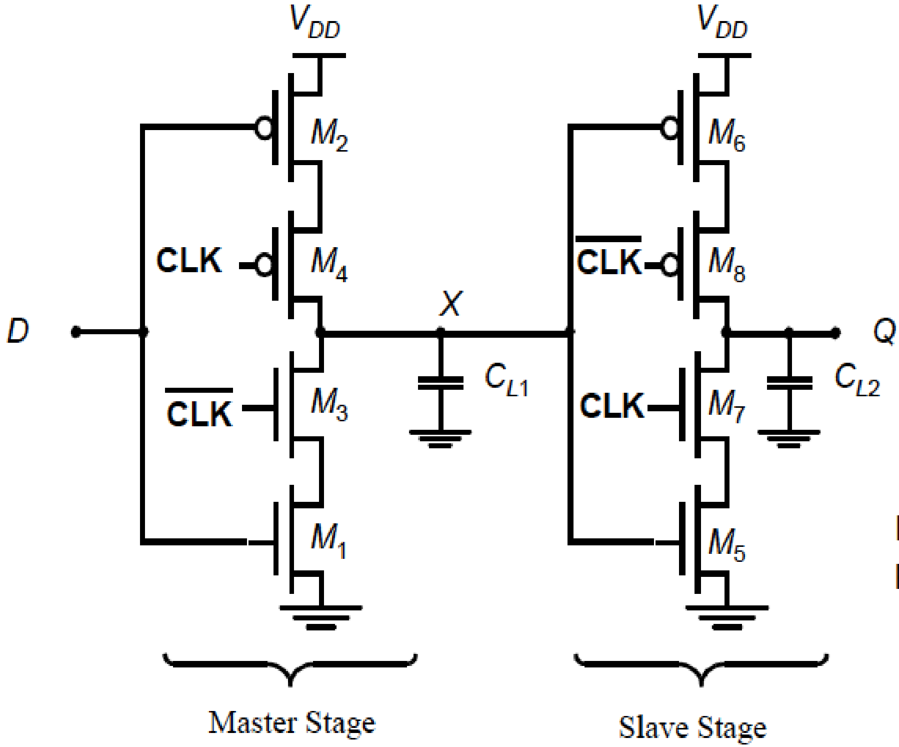


Circuitry for generating a two-phase non-overlapping clock.

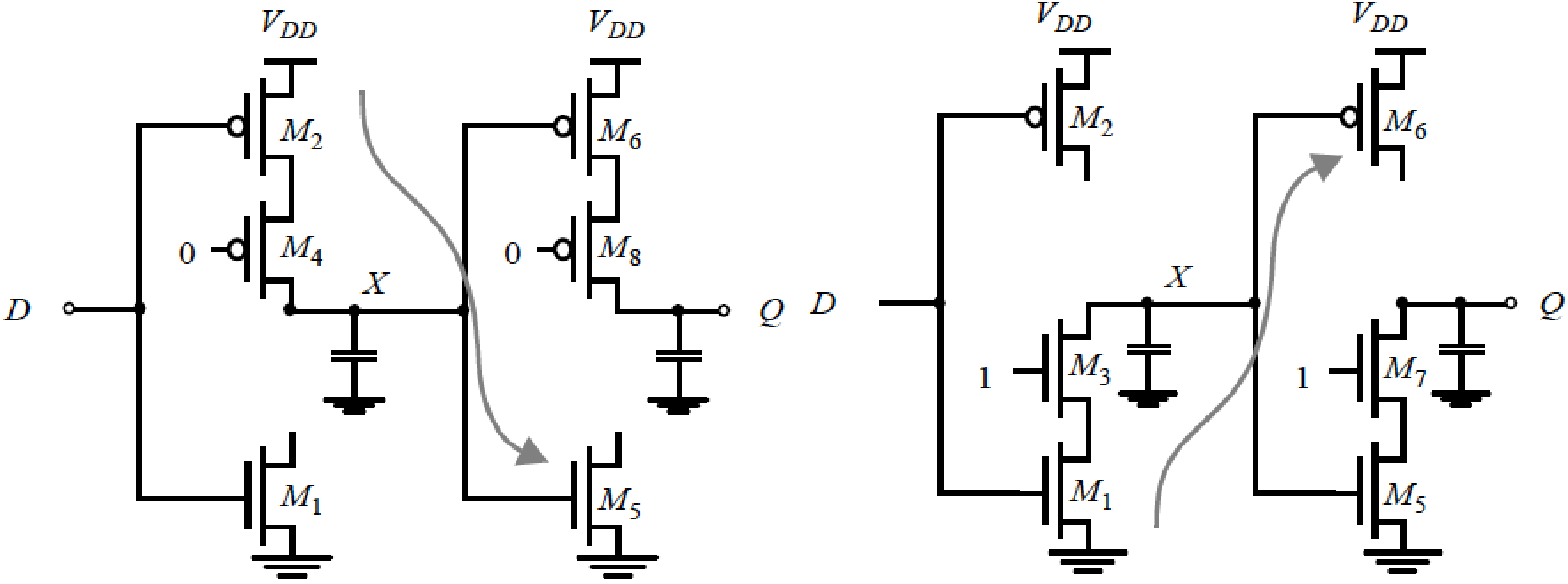
C^{2}MOS: A CLOCK SKEW INSENSITIVE APPROACH

- CLK = 0 : The first tri-state driver is turned on, and the master stage acts as an inverter sampling the inverted version of D on the internal node X. The master stage is in the evaluation mode. Meanwhile, the slave section is in a high-impedance mode, or in a hold mode. Both transistors M7 and M8 are off, decoupling the output from the input. The output Q retains its previous value stored on the output capacitor C_{L2}.

- The roles are reversed when CLK = 1: The master stage section is in hold mode (M3- M4 off), while the second section evaluates (M7-M8 on). The value stored on CL1 propagates to the output node through the slave stage which acts as an inverter.



C²MOS: RESPONSE TO CLK OVERLAP



(a) (0-0) overlap

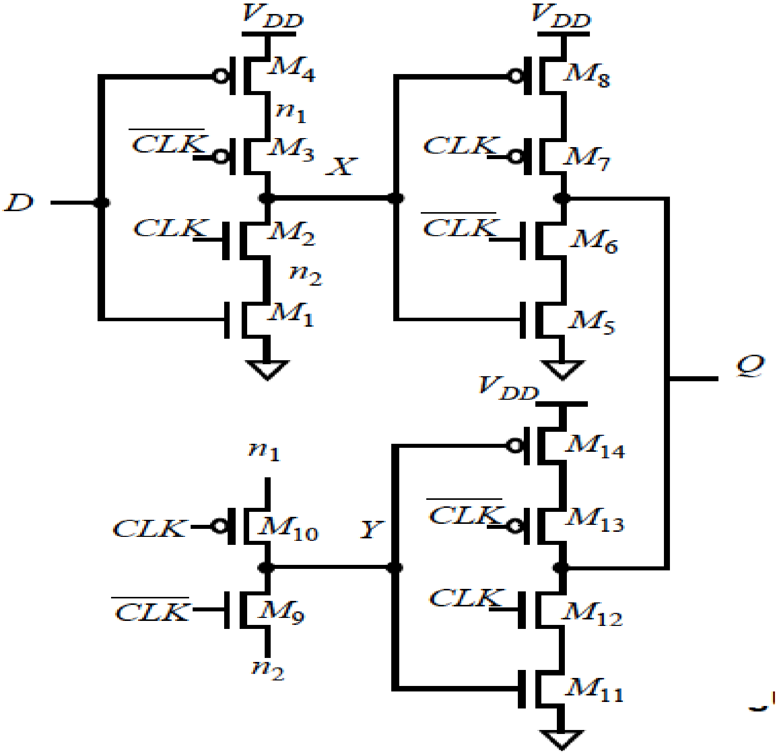
(b) (1-1) overlap

C²MOS: DUAL-EDGE TRIGGERED FLIP-FLOP/REGISTER

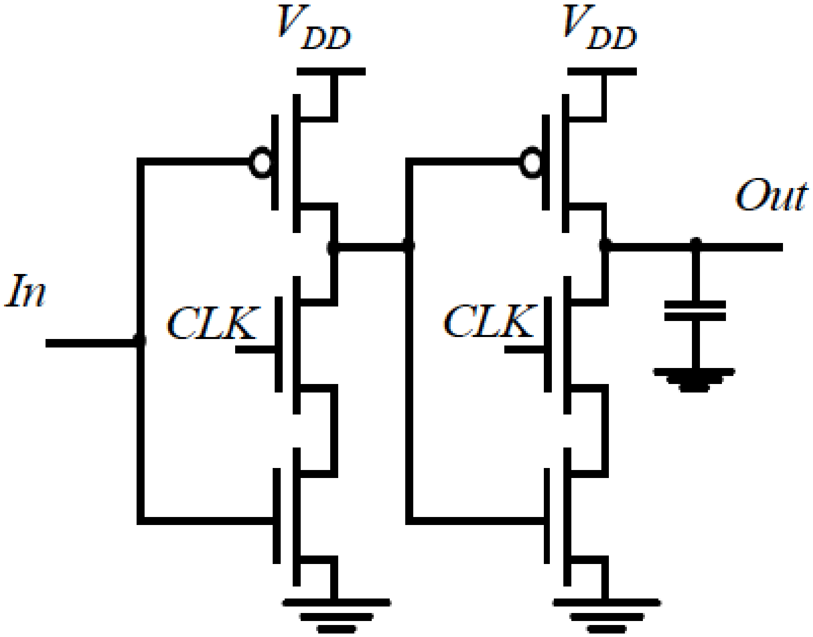
- So far, we have focused on edge-triggered registers that sample the input data on only one of the clock edges (rising or falling). It is also possible to design sequential circuits that sample the input on both edges. The advantage of this scheme is that a lower frequency clock (half of the original rate) is distributed for the same functional throughput, resulting in power savings in the clock distribution network.

- It consists of two parallel master-slave based edge-triggered registers.

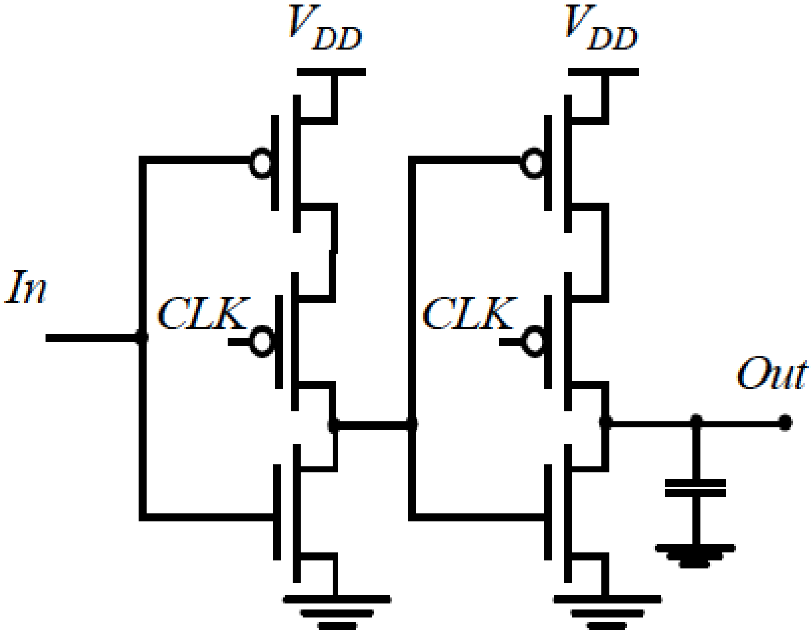
- Note that the slave latches operate in a complementary fashion — this is, only one of them is turned on during each phase of the clock.



TRUE SINGLE PHASE CLOCKED (TSPC) LATCH

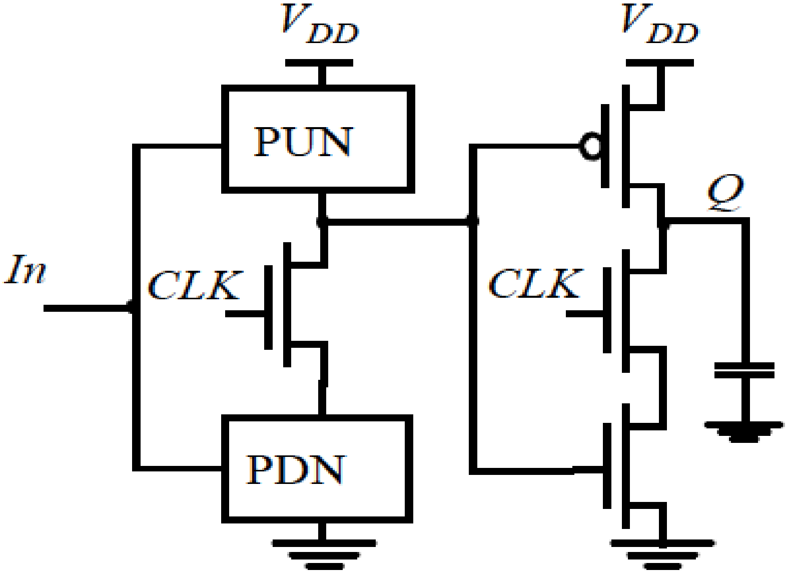


Positive Latch

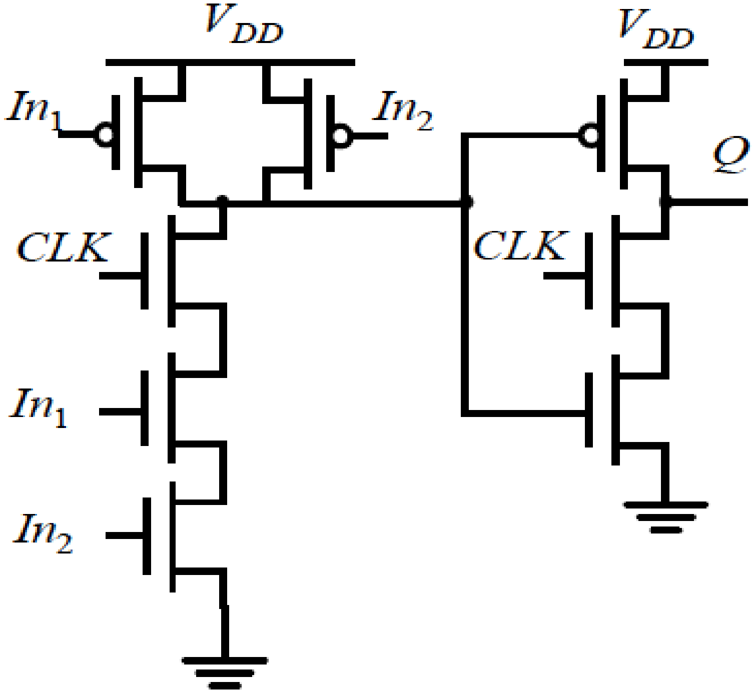


Negative Latch

EMBEDDING LOGIC INTO TSPC LATCHES



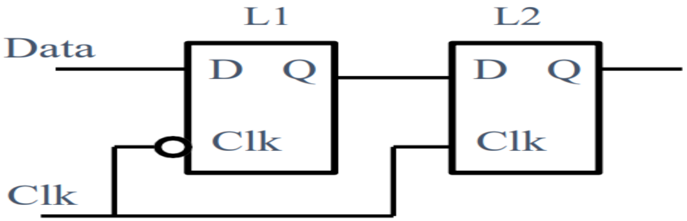
(a) Including logic into the latch



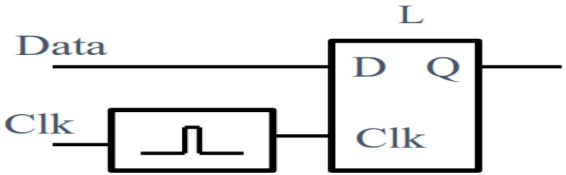
(b) AND latch

PULSED LATCHES

Edge-triggered Flip-flop based on Master-Slave Latches



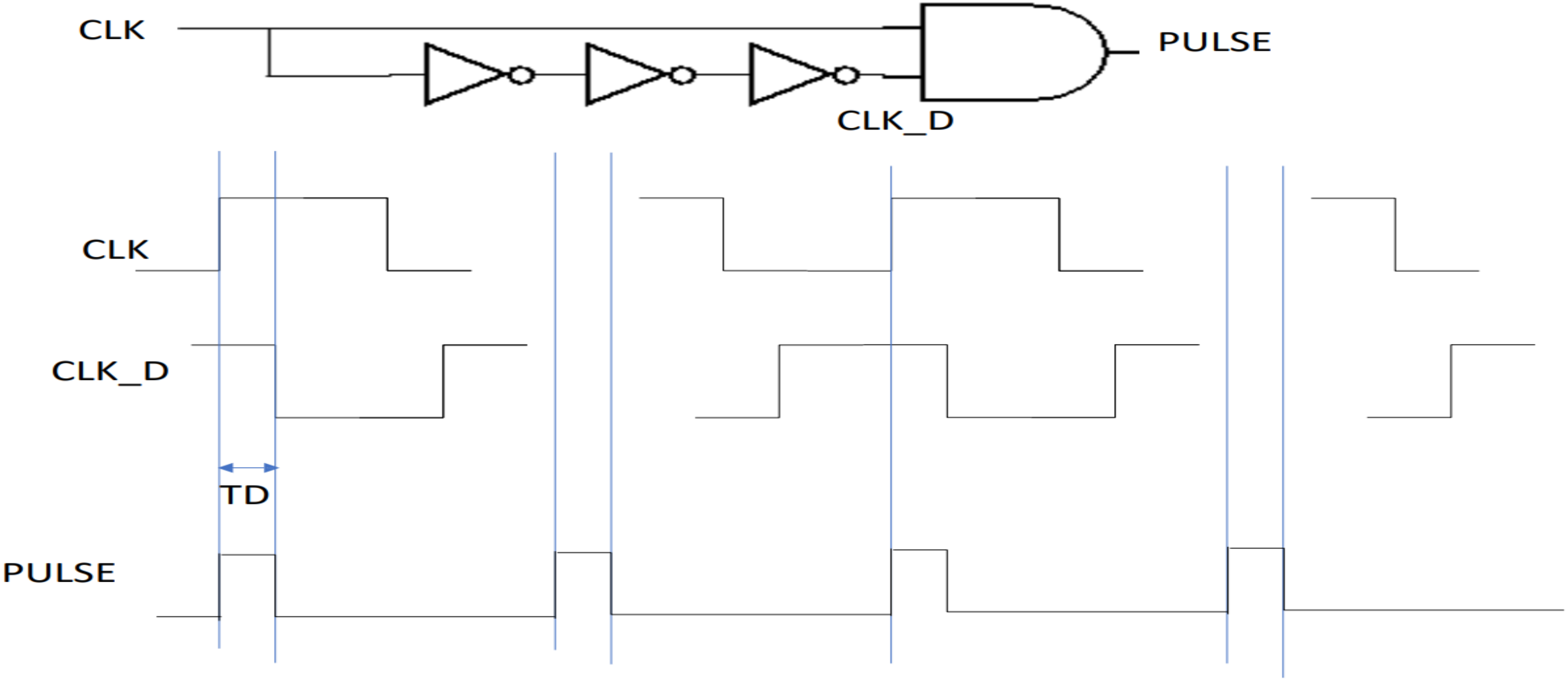
Pulsed Latch



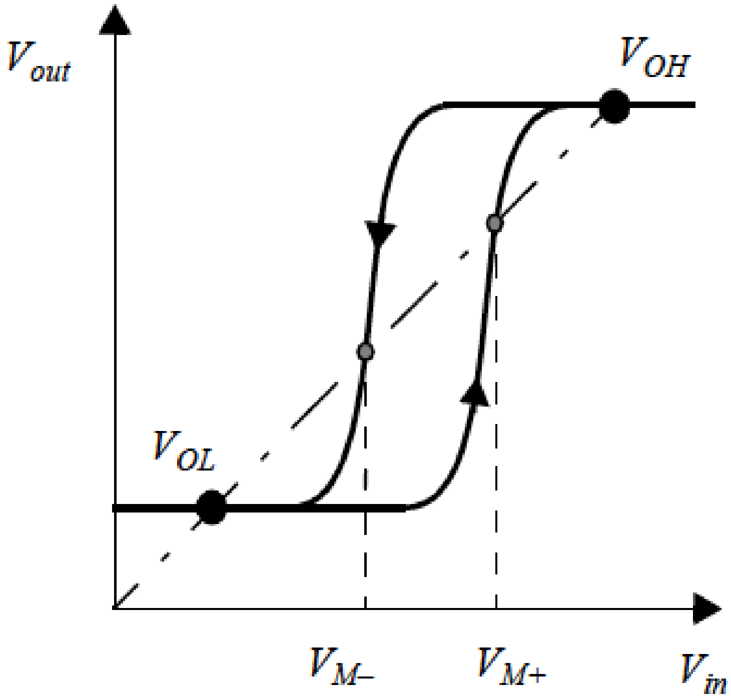
- Until now, we have used the master-slave configuration to create an edge-triggered register. A fundamentally different approach for constructing a register uses pulse signals.

- The idea is to construct a short pulse around the rising (or falling) edge of the clock. This pulse acts as the clock input to a latch, sampling the input only in a short window. Race conditions are thus avoided by keeping the opening time (i.e, the transparent period) of the latch very short.

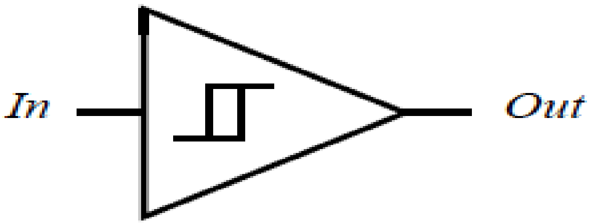
GLITCH/PULSE GENERATION



NON-BISTABLE SEQUENTIAL CIRCUIT: THE SCHMITT TRIGGER



(a) Voltage-transfer characteristic

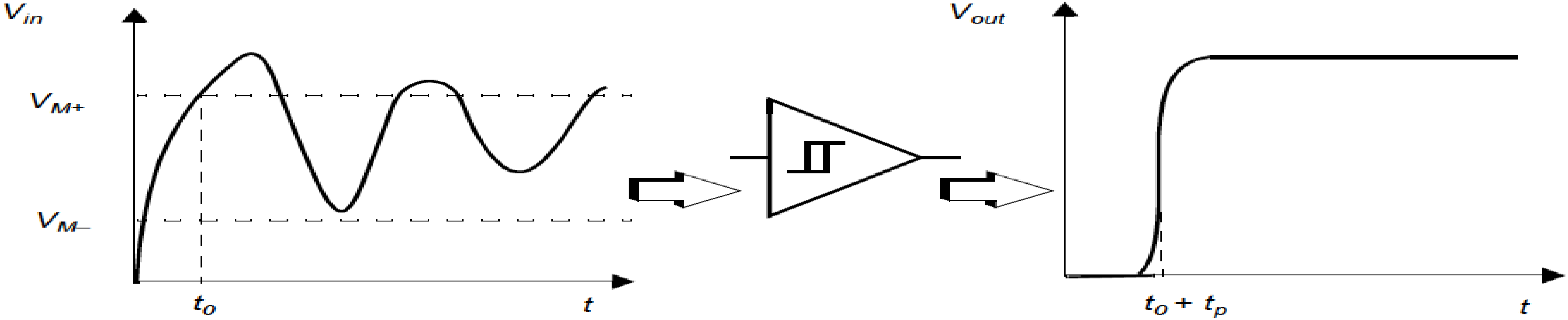


(b) Schematic symbol

- The voltage-transfer characteristic of the device displays different switching thresholds for positive- and negative-going input signals.

- One of the main uses of the Schmitt trigger is to turn a noisy or slowly varying input signal into a clean digital output signal.

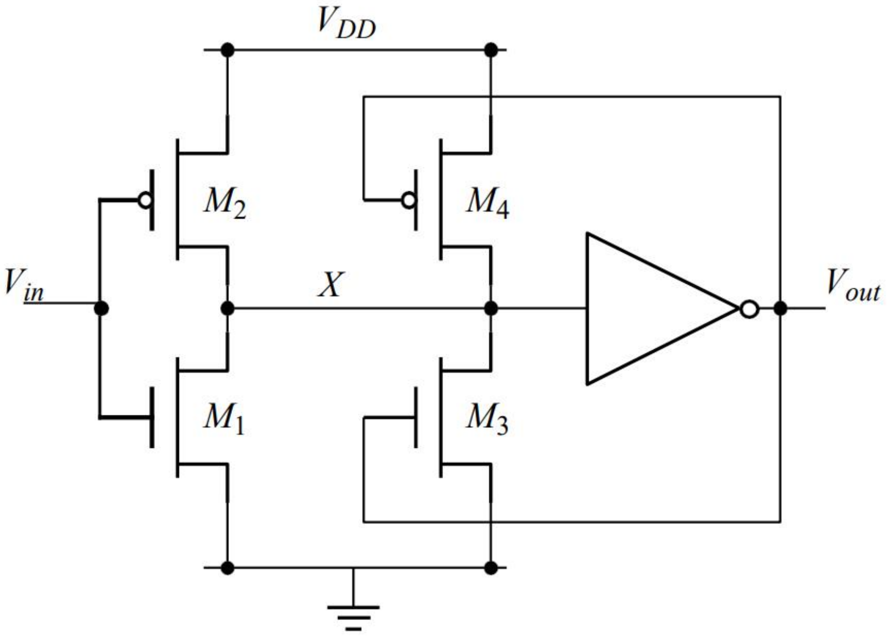
NOISE SUPPRESSION USING A SCHMITT TRIGGER



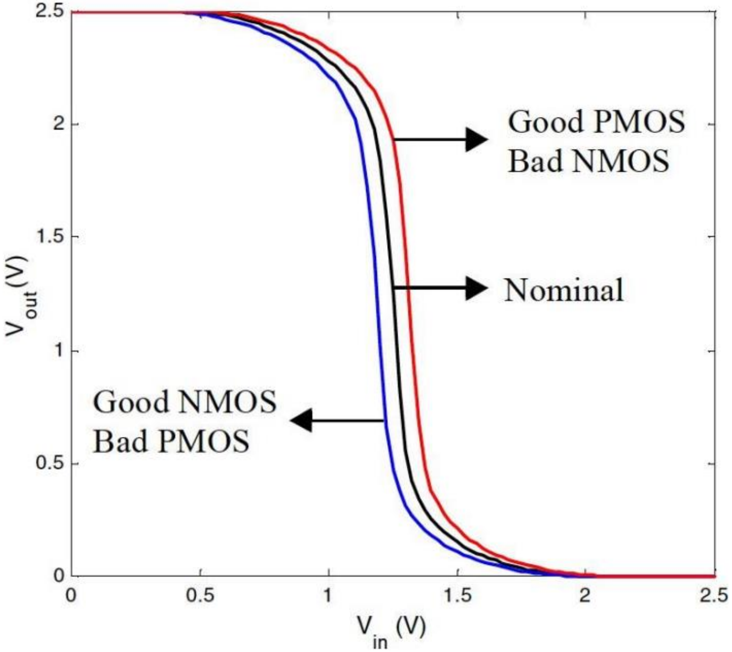
- Notice how the hysteresis suppresses the ringing on the signal. At the same time, the fast low-to-high (and high-to-low) transitions of the output signal should be observed. For instance, steep signal slopes are beneficial in reducing power consumption by suppressing direct-path currents.

- The “secret” behind the Schmitt trigger concept is the use of positive feedback.

CMOS SCHMITT TRIGGER: NON-INVERTING

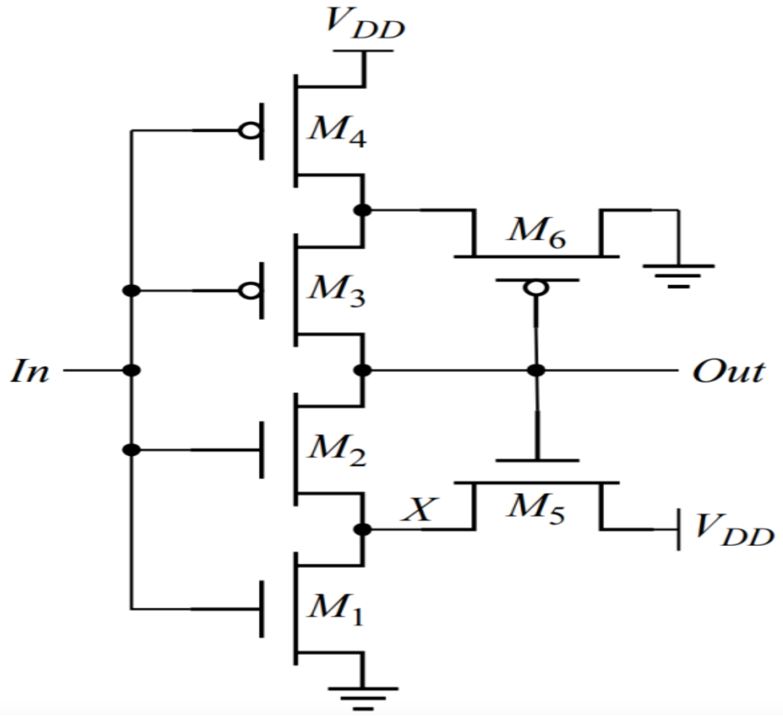


Recall



CMOS SCHMITT TRIGGER: INVERTING

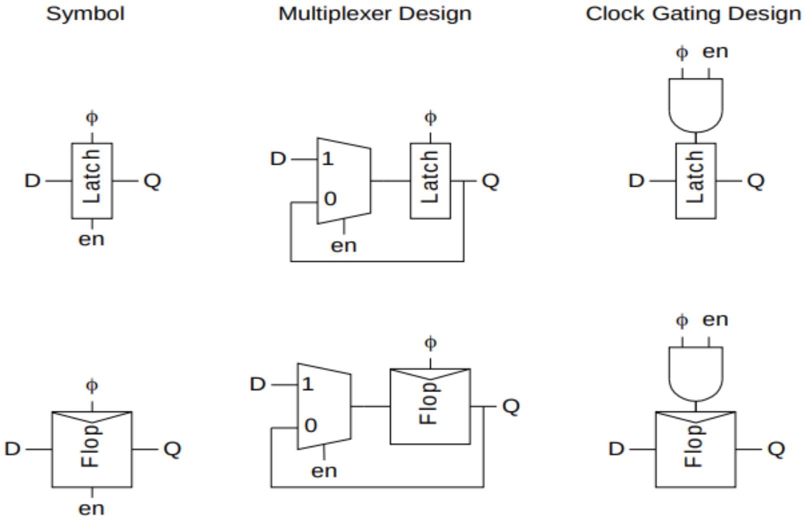
0 → VDD



VDD → 0

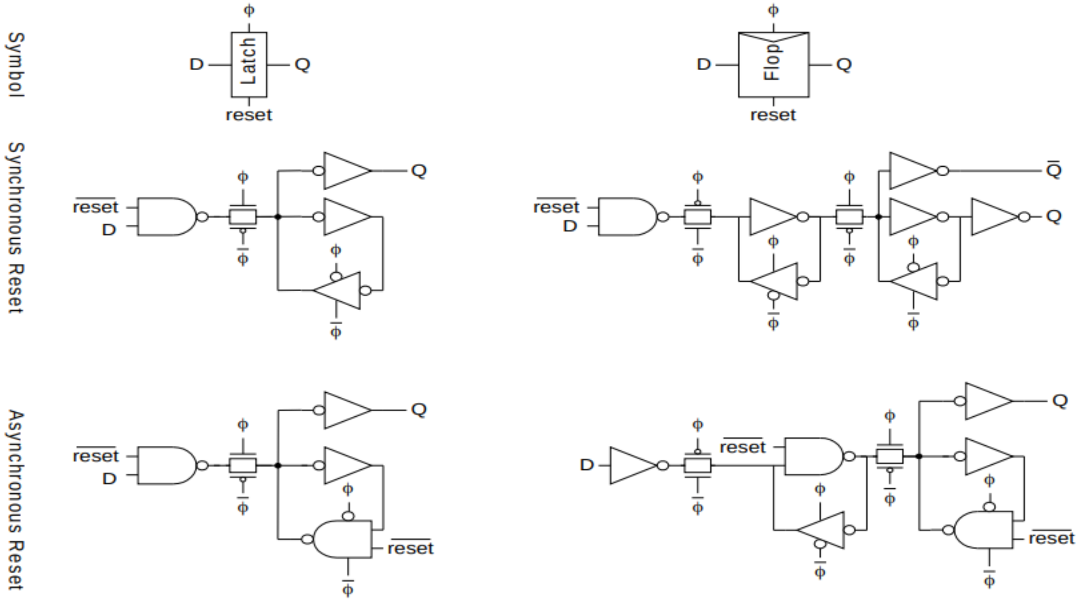
ENABLE

- Enable: ignore clock when en = 0
    - ➢ Mux: increase latch D-Q delay
    - ➢ Clock Gating: increase in setup time, skew



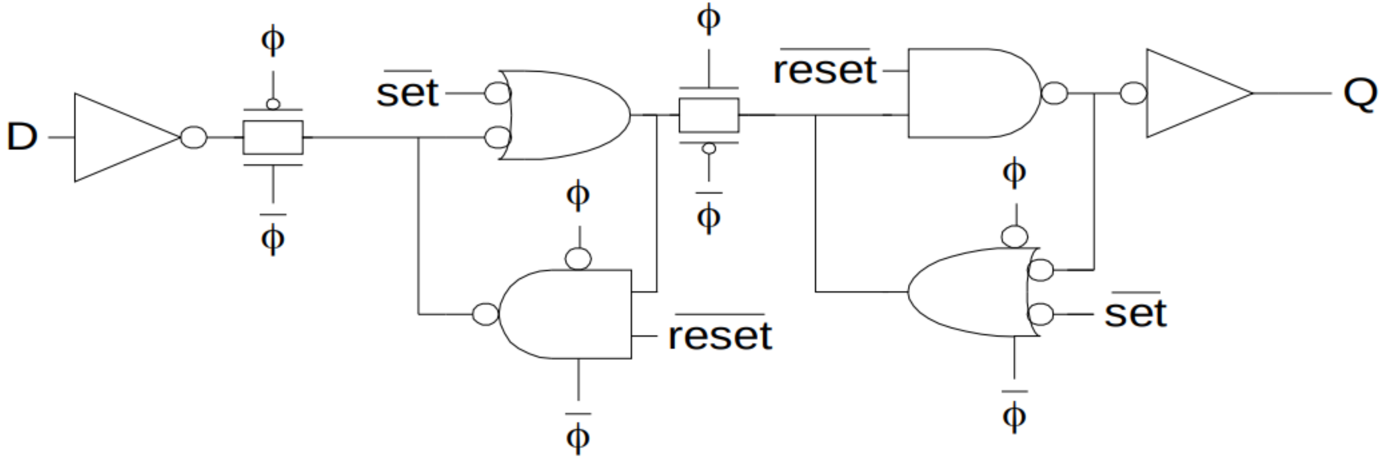
RESET

- Force output low when reset asserted
- Synchronous vs. asynchronous

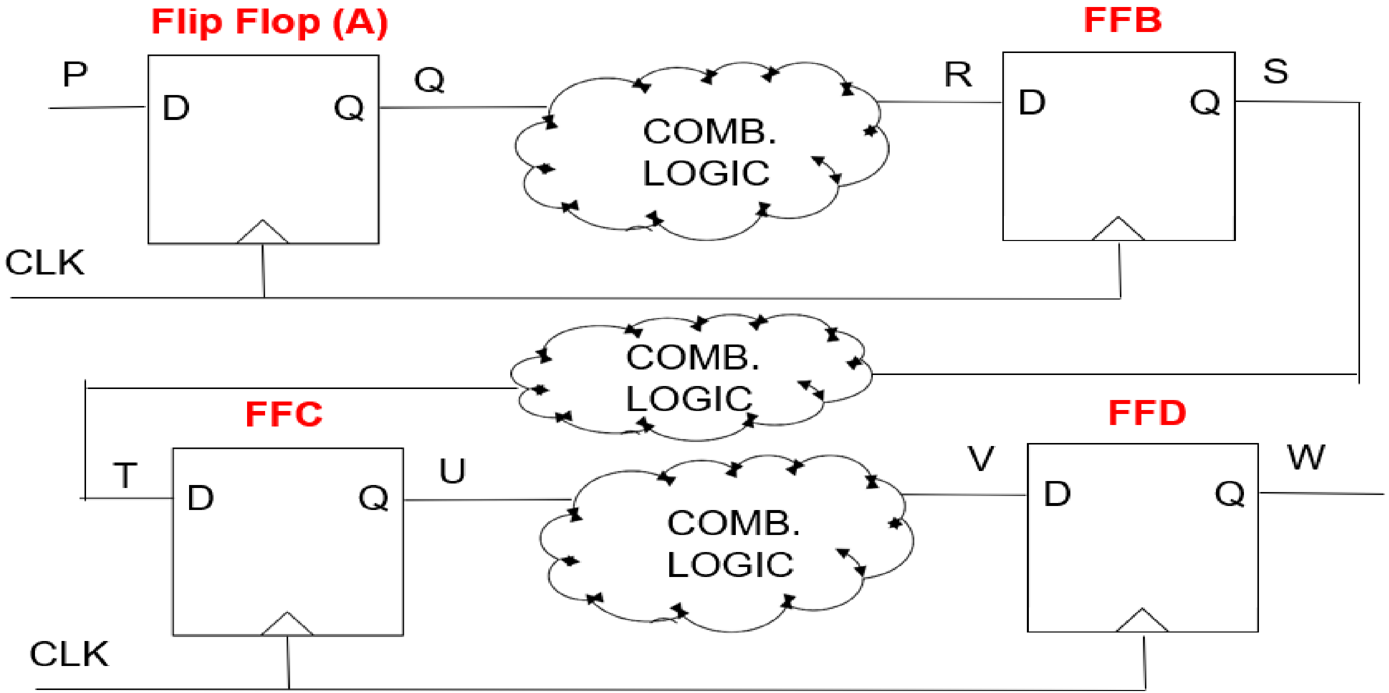


SET/RESET

- Set forces output high when enabled
- Flip-flop with asynchronous set and reset

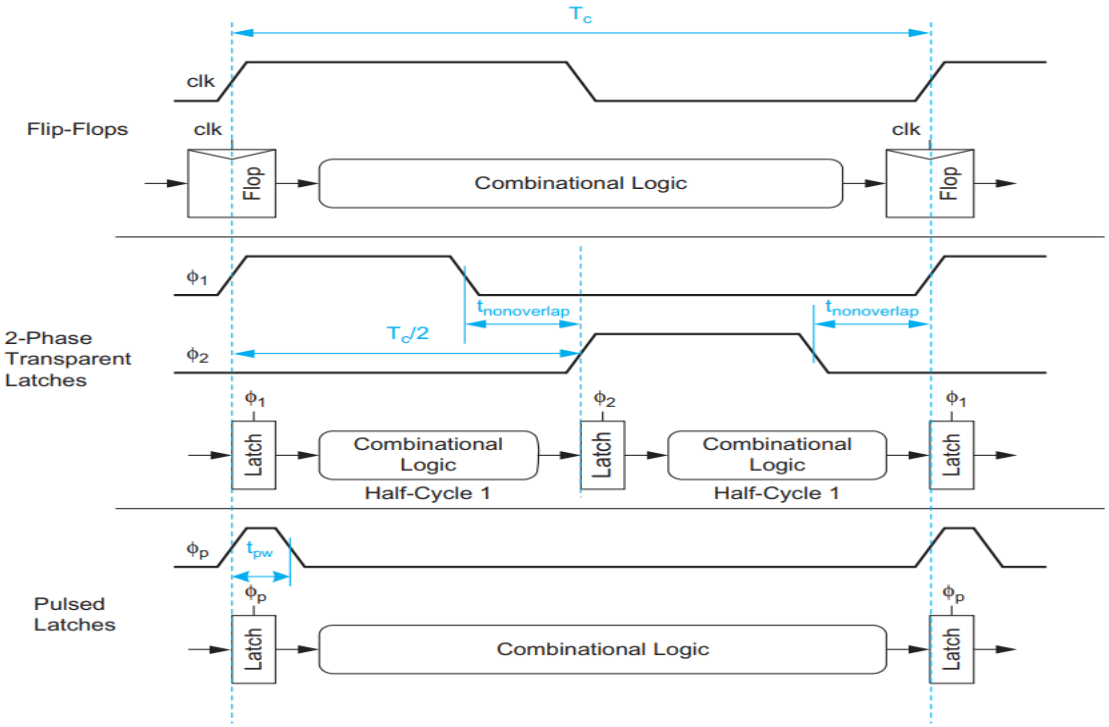


PIPE-LINED ARCHITECTURE



SEQUENCING METHODS

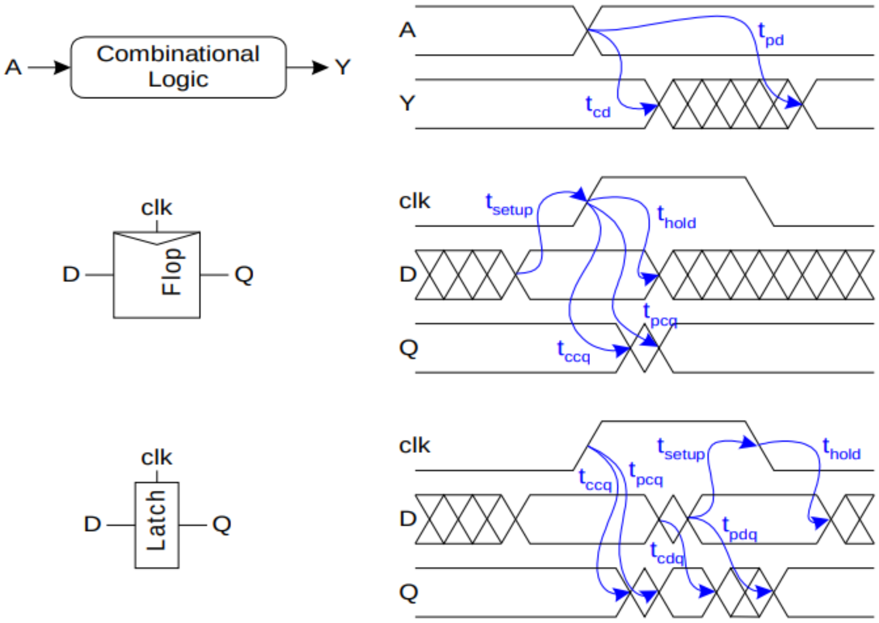
- Flip-flops
- 2-Phase Latches
- Pulsed Latches



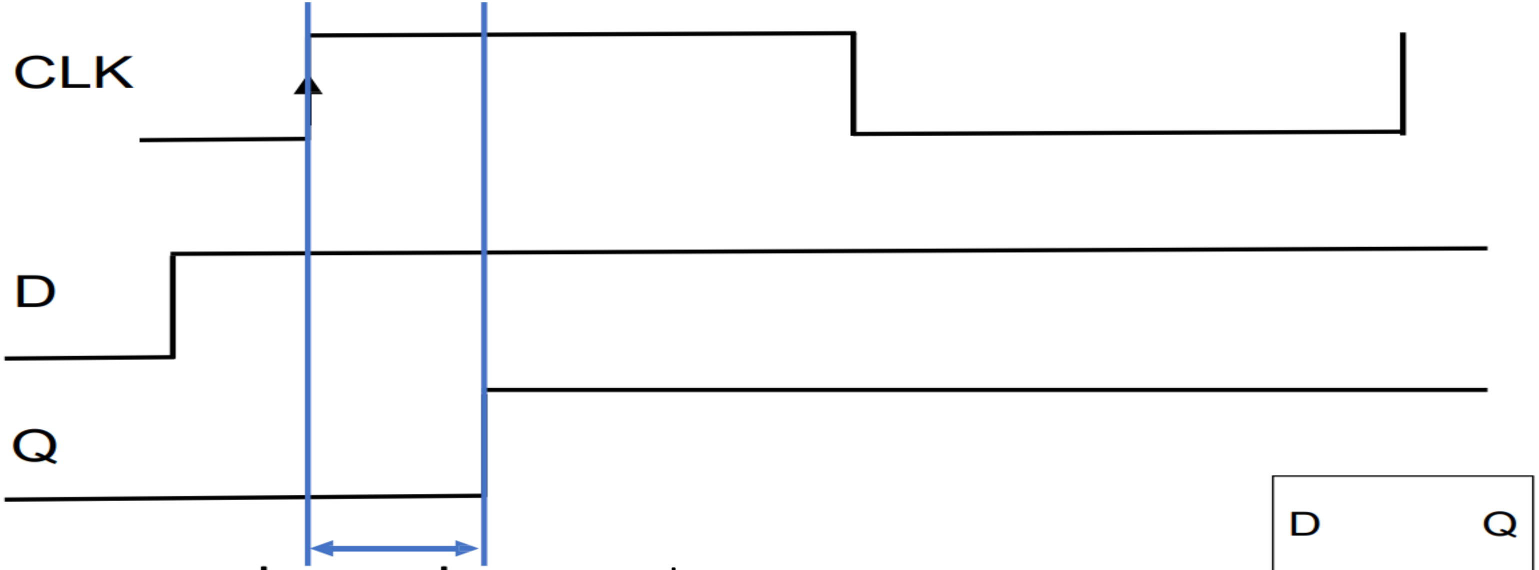
TIMING DIAGRAMS

- Contamination and Propagation Delays

t_{pd}	**Logic Prop. Delay**
t_{cd}	**Logic Cont. Delay**
t_{pcq}	**Latch/Flop Clk-Q Prop Delay**
t_{ccq}	**Latch/Flop Clk-Q Cont. Delay**
t_{pdq}	**Latch D-Q Prop Delay**
t_{cdq}	**Latch D-Q Cont. Delay**
t_{setup}	**Latch/Flop Setup Time**
t_{hold}	**Latch/Flop Hold Time**

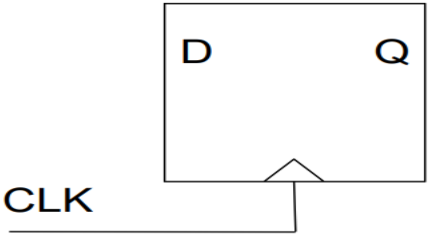


TIMING METRICS: RISING EDGE TRIGGERED FLIP-FLOP

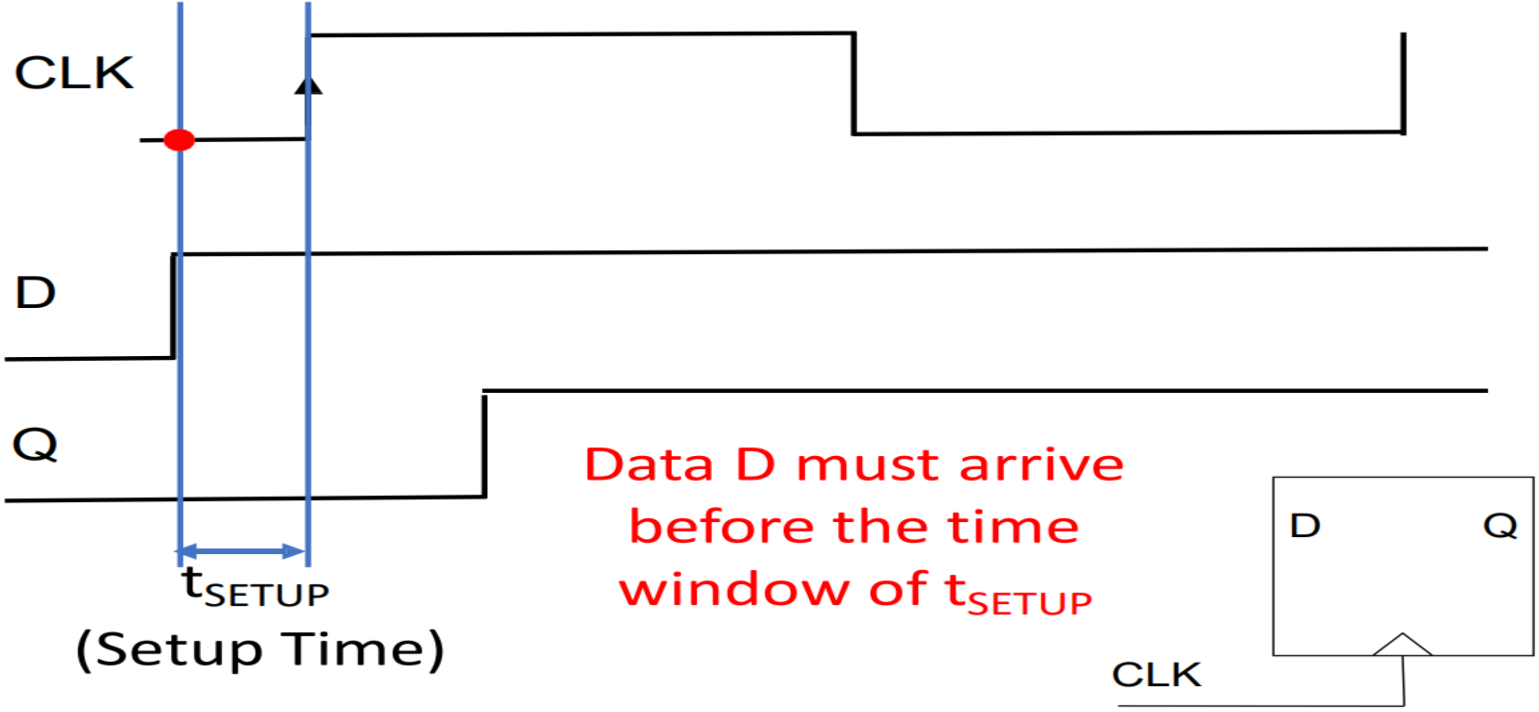


$t_{PFF} = t_{CLK \rightarrow Q} = t_{pcq}$

(Delay from CLK to Q)

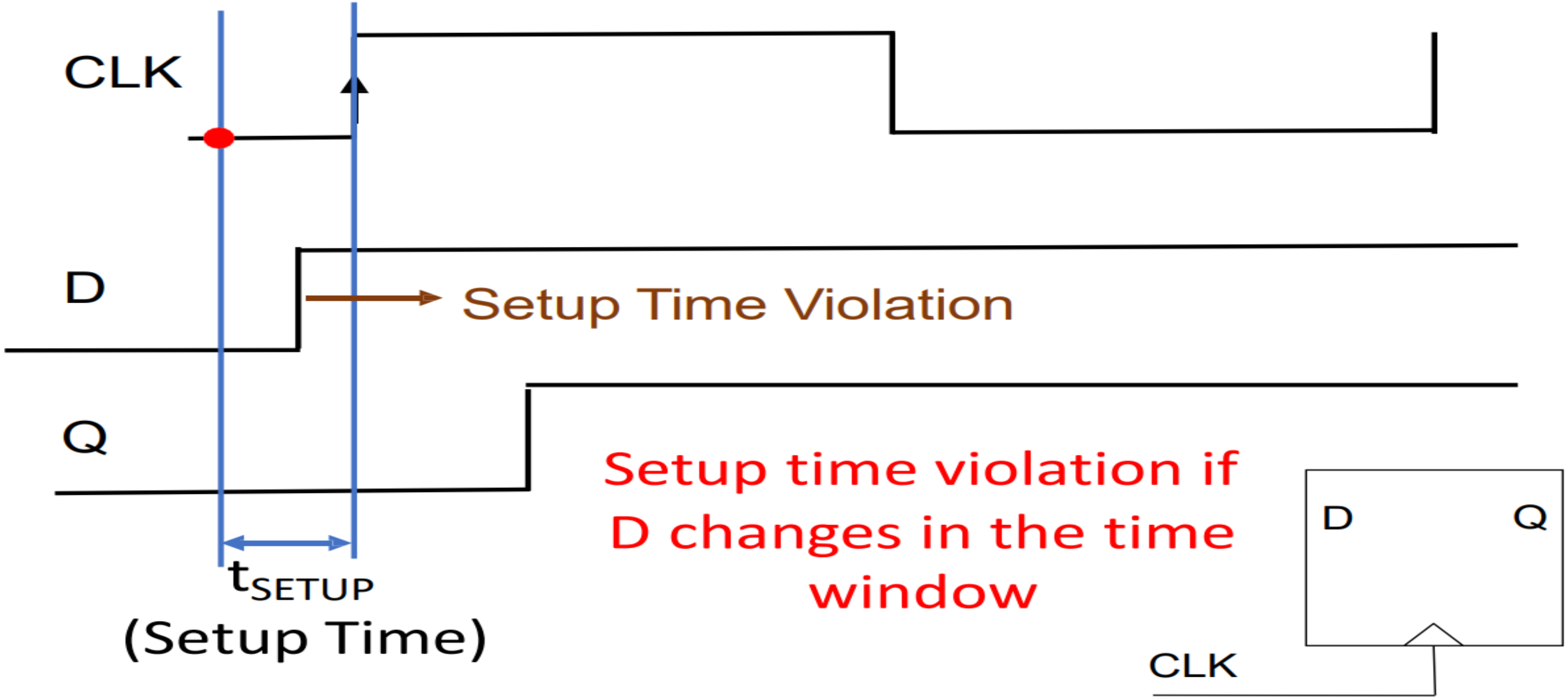


TIMING METRICS: RISING EDGE TRIGGERED FLIP-FLOP



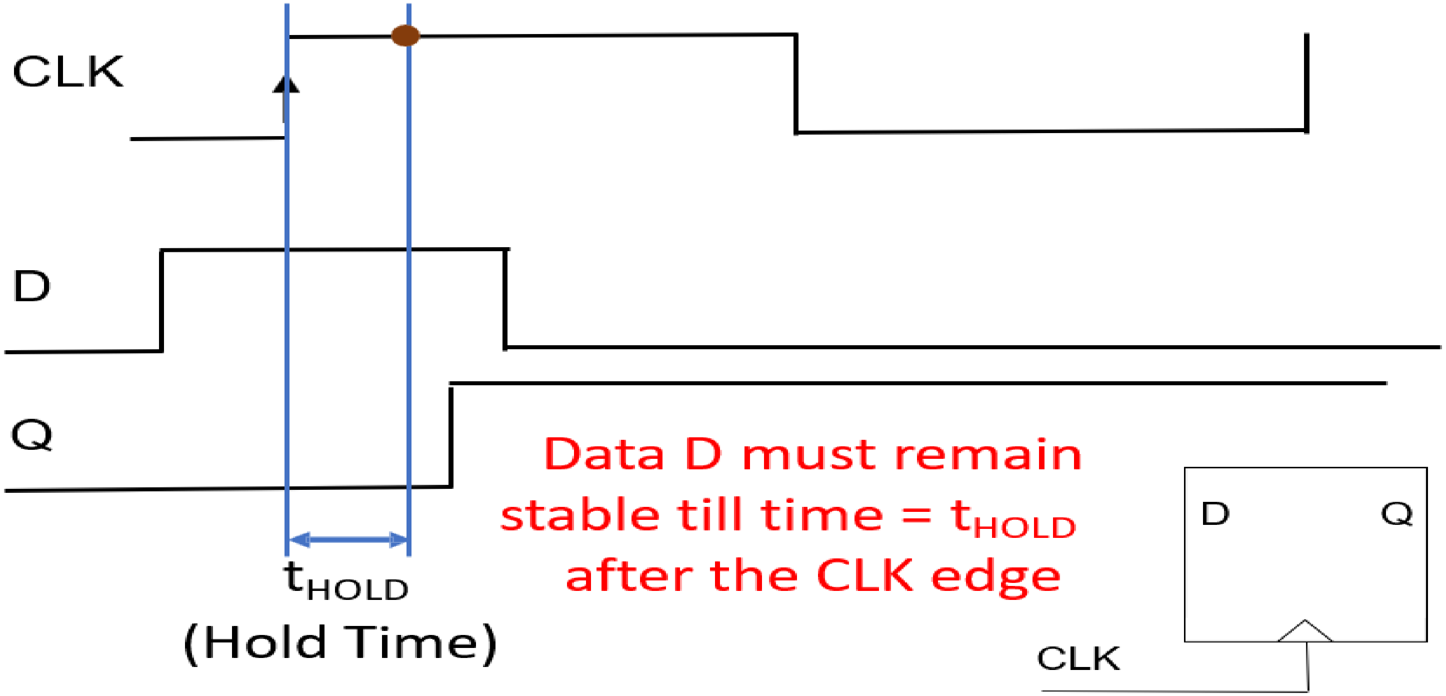
Data D must arrive before the time window of t_{SETUP}

TIMING METRICS: RISING EDGE TRIGGERED FLIP-FLOP



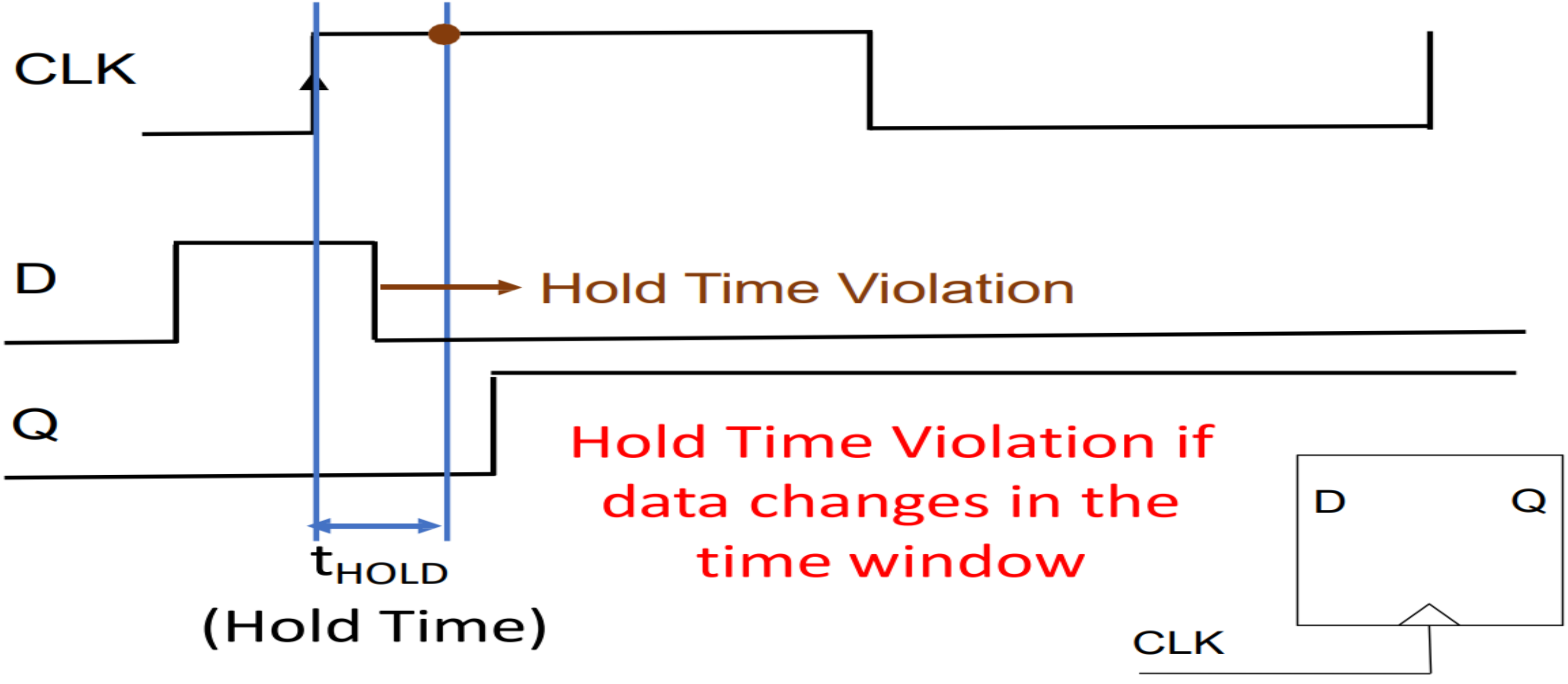
Setup time violation if D changes in the time window

TIMING METRICS: RISING EDGE TRIGGERED FLIP-FLOP



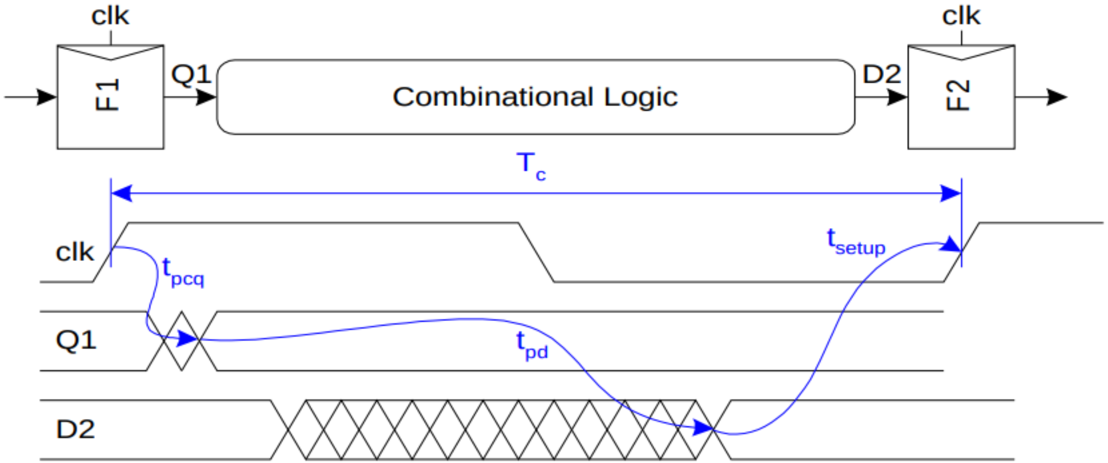
Data D must remain stable till time = t_{HOLD} after the CLK edge

TIMING METRICS: RISING EDGE TRIGGERED FLIP-FLOP



Hold Time Violation if data changes in the time window

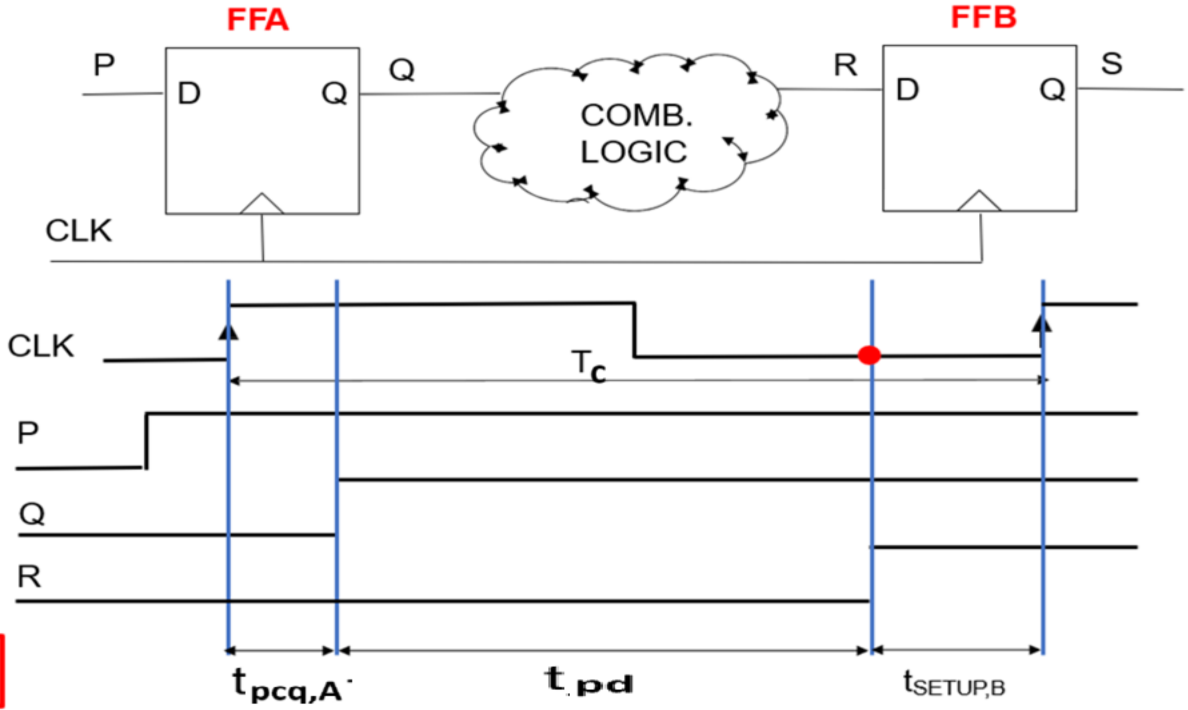
MAX-DELAY: FLIP-FLOPS



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

$$t_{pd} \leq T_c - \underbrace{\left(t_{\text{setup}} + t_{pcq}\right)}_{\text{sequencing overhead}}$$

SETUP TIME CONSTRAINT



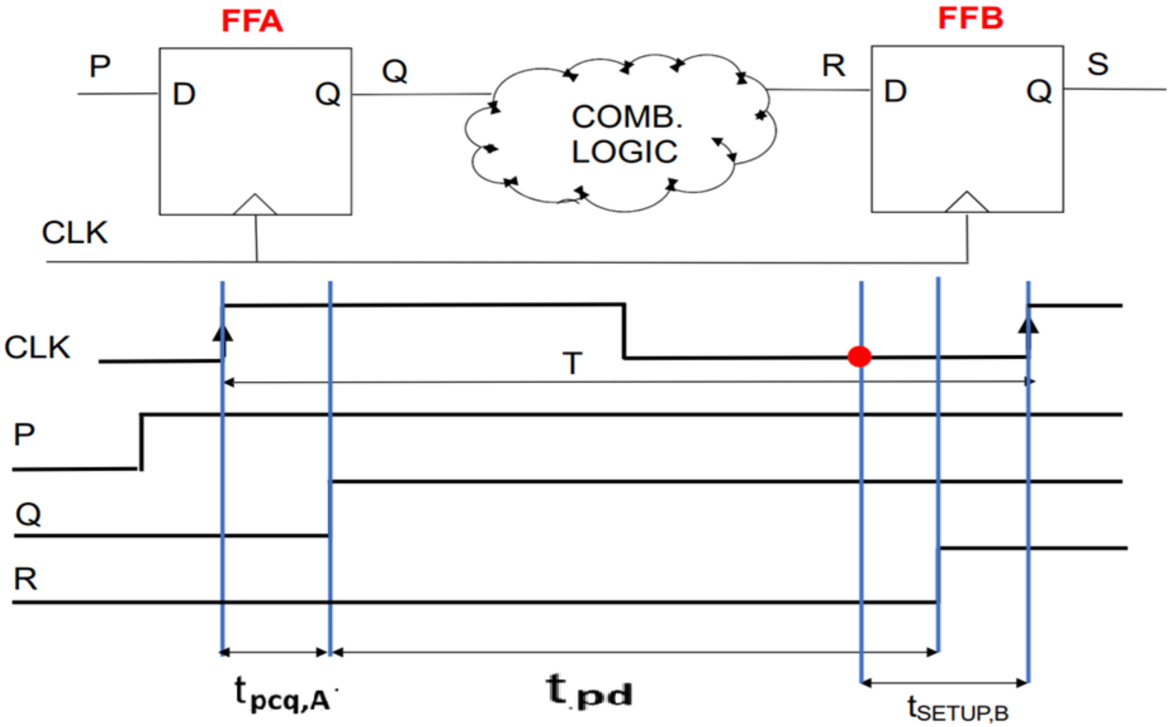
Must be ensured by design

$$\boxed{T_c >= t_{pcq,A} + t_{pd} + t_{SETUPB}}$$

SETUP TIME CONSTRAINT

$$\boxed{T_c < t_{pcq,A} + t_{pd} + t_{SETUPB}}$$

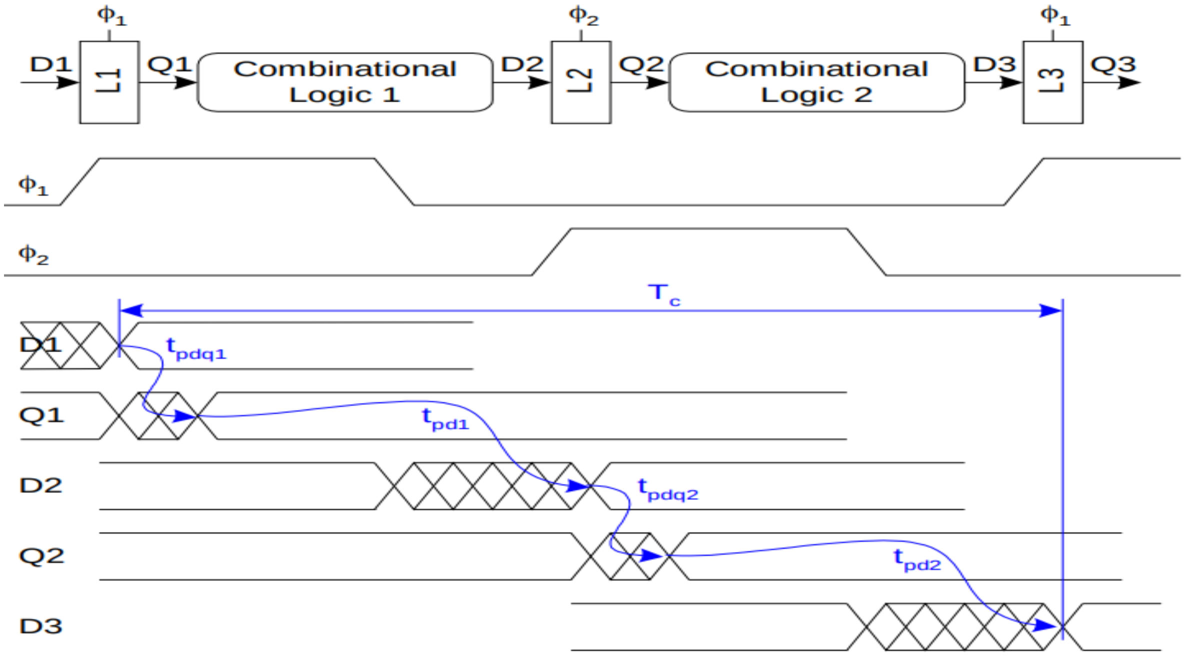
Setup time violation!
Must be avoided



MAX-DELAY: 2-PHASE LATCHES

$$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}$$

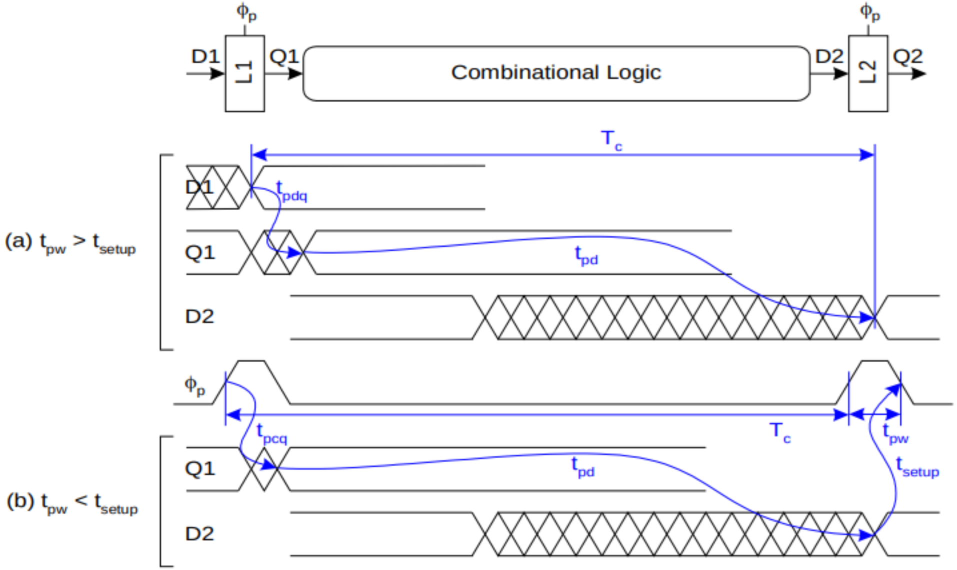
$$t_{pd} = t_{pd1} + t_{pd2} \leq T_c - \underbrace{\left(2t_{pdq}\right)}_{\text{sequencing overhead}}$$



MAX-DELAY: PULSED LATCHES

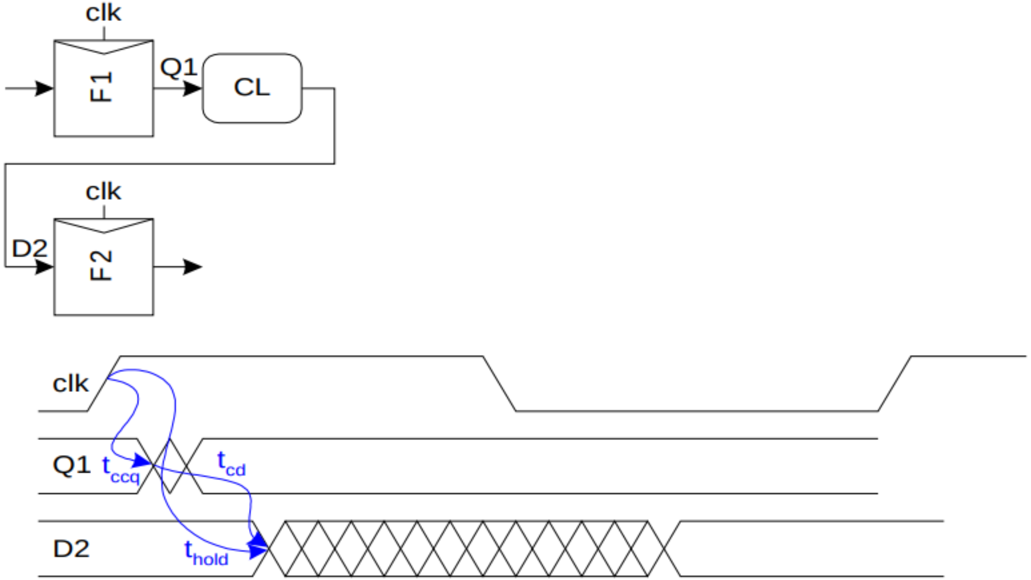
$$T_c \geq \max\left(t_{pdq} + t_{pd},\ t_{pcq} + t_{pd} + t_{\text{setup}} - t_{pw}\right)$$

$$t_{pd} \leq T_c - \underbrace{\max\left(t_{pdq},\ t_{pcq} + t_{\text{setup}} - t_{pw}\right)}_{\text{sequencing overhead}}$$



MIN-DELAY: FLIP-FLOPS

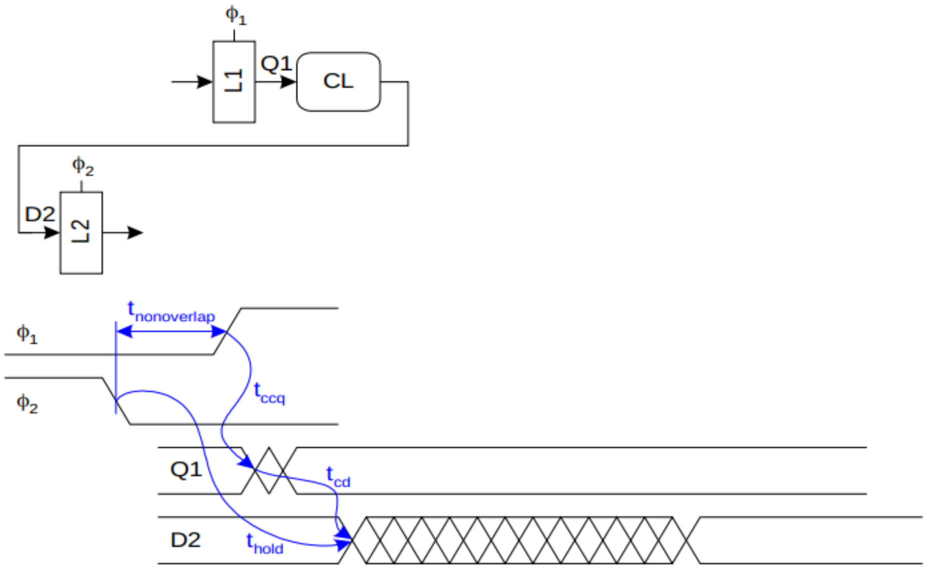
$$t_{cd} \geq t_{\text{hold}} - t_{ccq}$$



MIN-DELAY: 2-PHASE LATCHES

$$t_{cd1}, t_{cd2} \geq t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}}$$

- Hold time failure reduced by nonoverlap
- Paradox: hold applies twice each cycle, vs. only once for flops.
- But a flop is made of two latches!

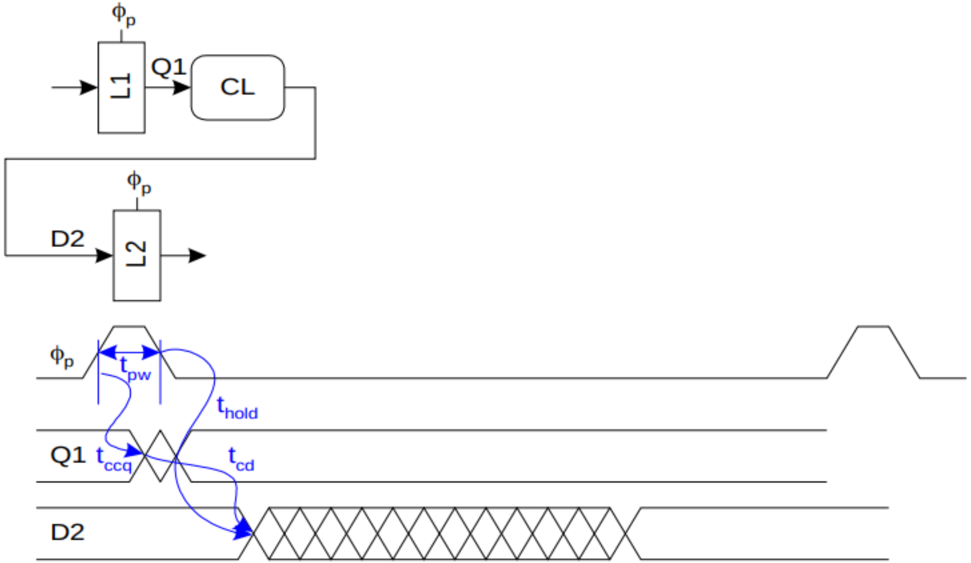


CAL POLY
SAN LUIS OBISPO

MIN-DELAY: PULSED LATCHES

$$t_{cd} \geq t_{\text{hold}} - t_{ccq} + t_{pw}$$

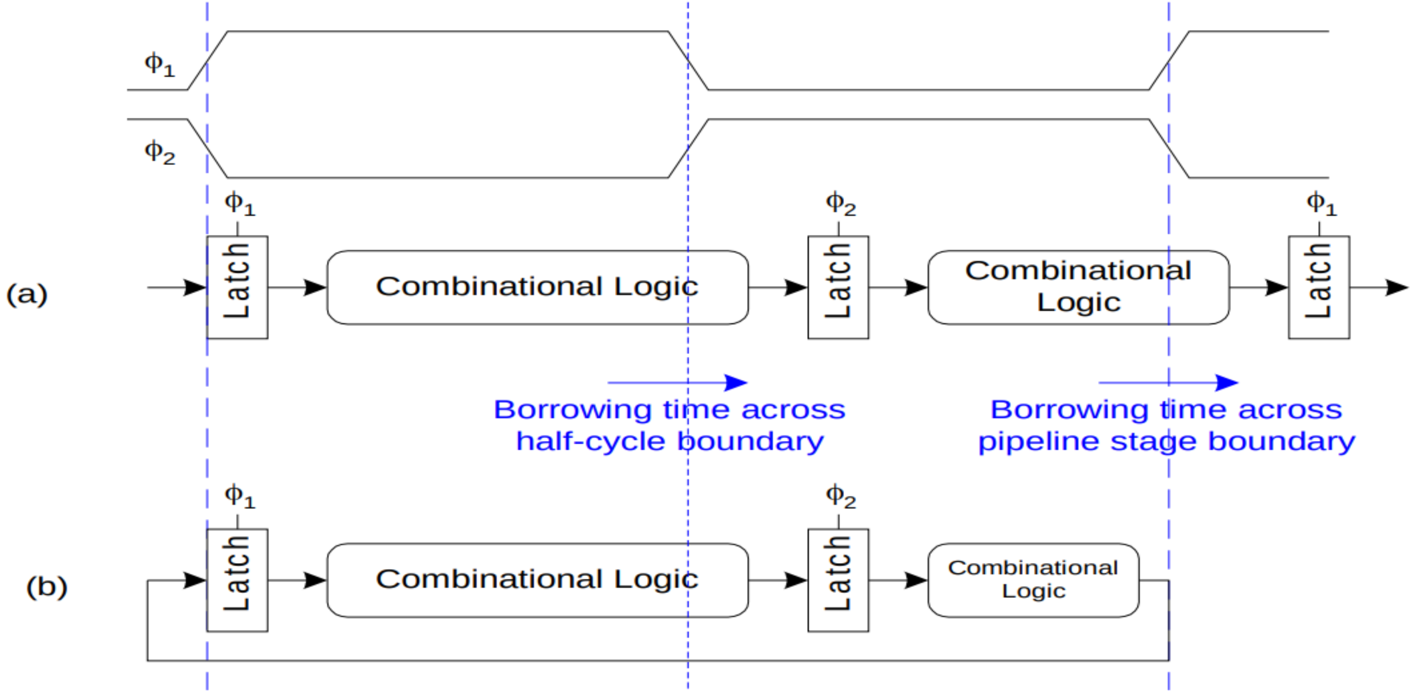
- Hold time increased by pulse width



TIME BORROWING

- In a flop-based system:
  - Data launches on one rising edge
  - Must setup before next rising edge
  - If it arrives late, system fails
  - If it arrives early, time is wasted
  - Flops have hard edges
- In a latch-based system:
  - Data can pass through latch while transparent
  - Long cycle of logic can borrow time into next
  - As long as each loop completes in one cycle

TIME BORROWING EXAMPLE



Loops may borrow time internally but must complete within the cycle

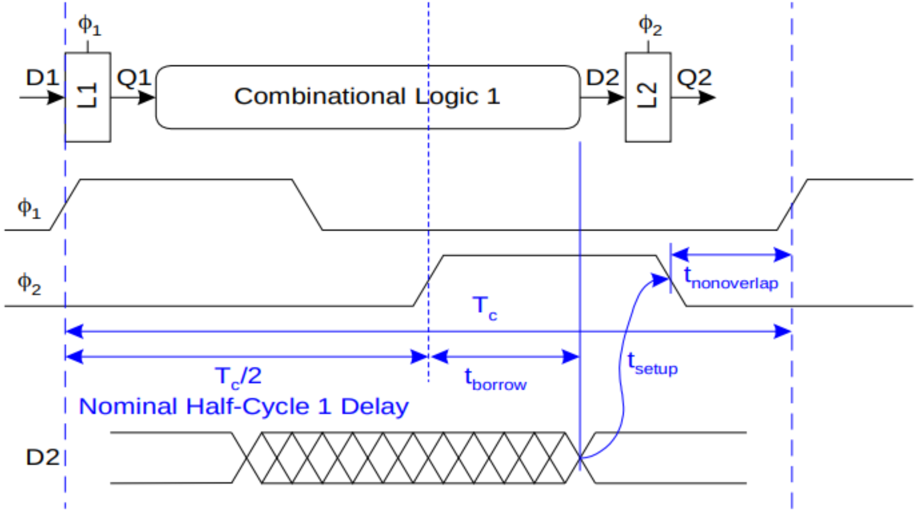
HOW MUCH BORROWING?

- 2-Phase latches

$$t_{\text{borrow}} \le \frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}}\right)$$

- Pulsed latches (only if $t_{pw} > t_{setup}$)

$$t_{\text{borrow}} \le t_{pw} - t_{\text{setup}}$$



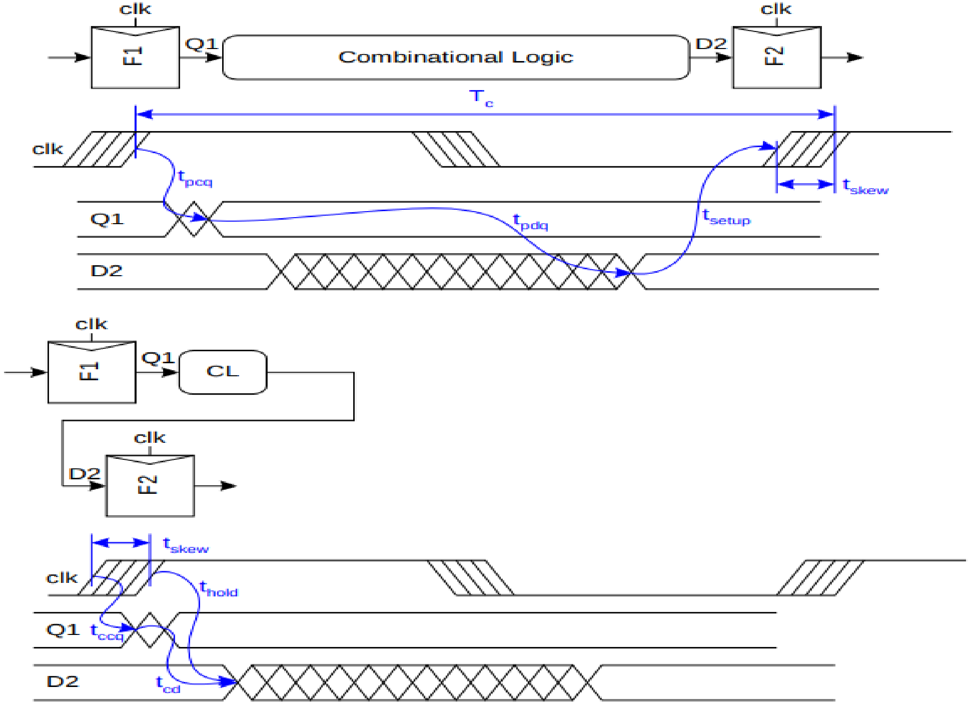
CLOCK SKEW

- We have assumed zero clock skew
- Clocks really have uncertainty in arrival time
  - ➢ Decreases maximum propagation delay
  - ➢ Increases minimum contamination delay
  - ➢ Decreases time borrowing

SKEW: FLIP FLOPS

$$t_{pd} \le T_c - \underbrace{\left(t_{pcq} + t_{\text{setup}} + t_{\text{skew}}\right)}_{\text{sequencing overhead}}$$

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



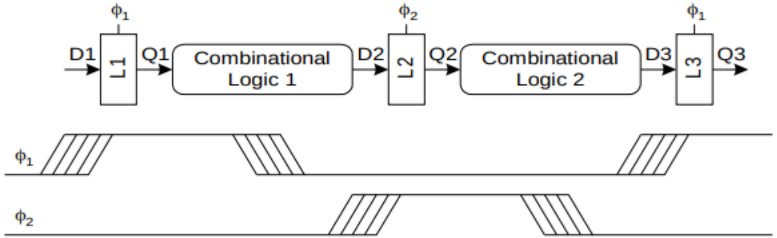
SKEW: LATCHES

- 2-phase latches

$$t_{pd} \le T_c - \underbrace{\left(2 t_{pdq}\right)}_{\text{sequencing overhead}}$$

$$t_{cd1}, t_{cd2} \ge t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}} + t_{\text{skew}}$$

$$t_{\text{borrow}} \le \frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}} + t_{\text{skew}}\right)$$



- Pulsed latches

$$t_{pd} \le T_c - \underbrace{\max\left(t_{pdq},\ t_{pcq} + t_{\text{setup}} - t_{pw} + t_{\text{skew}}\right)}_{\text{sequencing overhead}}$$

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

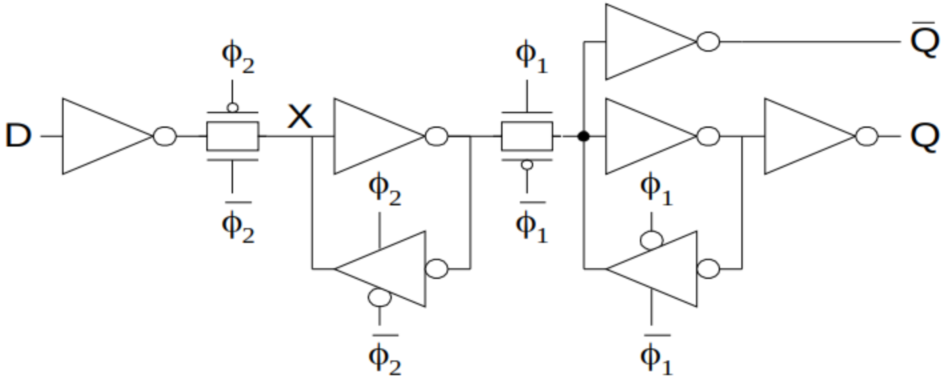
$$t_{\text{borrow}} \le t_{pw} - \left(t_{\text{setup}} + t_{\text{skew}}\right)$$

TWO PHASE CLOCKING

- If setup times are violated, reduce clock speed
- If hold times are violated, chip fails at any speed
- In this class, working chips are most important
    - ➢ No tools to analyze clock skew
- An easy way to guarantee hold times is to use 2-phase latches with big nonoverlap times
- Call these clocks Φ_1, Φ_2(ph1, ph2)

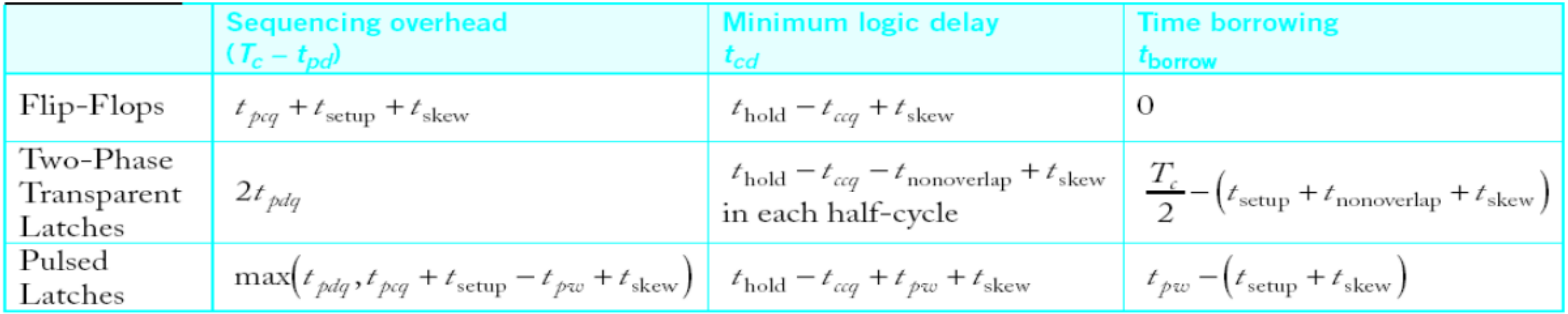
SAFE FLIP-FLOP

- In class, use flip-flop with nonoverlapping clocks
  - Very slow – nonoverlap adds to setup time
  - But no hold times
- In industry, use a better timing analyzer
  - Add buffers to slow signals if hold time is at risk



SUMMARY

- Flip-Flops:
  - Very easy to use, supported by all tools
- 2-Phase Transparent Latches:
  - Lots of skew tolerance and time borrowing
- Pulsed Latches:
  - Fast, some skew tolerance & borrow, hold time risk



	Sequencing overhead ($T_c - t_{pd}$)	Minimum logic delay t_{cd}	Time borrowing t_{borrow}
Flip-Flops	$t_{pcq} + t_{\text{setup}} + t_{\text{skew}}$	$t_{\text{hold}} - t_{ccq} + t_{\text{skew}}$	0
Two-Phase Transparent Latches	$2t_{pdq}$	$t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}} + t_{\text{skew}}$ in each half-cycle	$\frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}} + t_{\text{skew}}\right)$
Pulsed Latches	$\max\left(t_{pdq}, t_{pcq} + t_{\text{setup}} - t_{pw} + t_{\text{skew}}\right)$	$t_{\text{hold}} - t_{ccq} + t_{pw} + t_{\text{skew}}$	$t_{pw} - \left(t_{\text{setup}} + t_{\text{skew}}\right)$

Thank you!