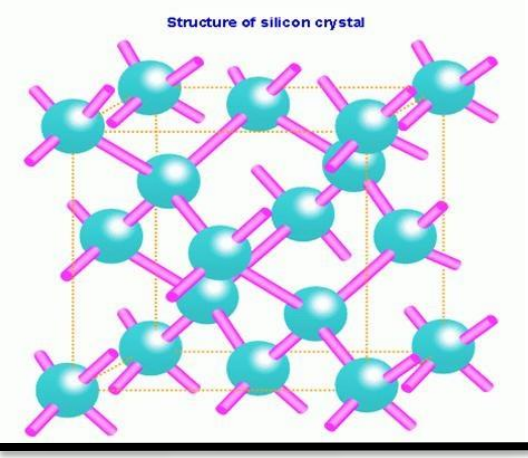
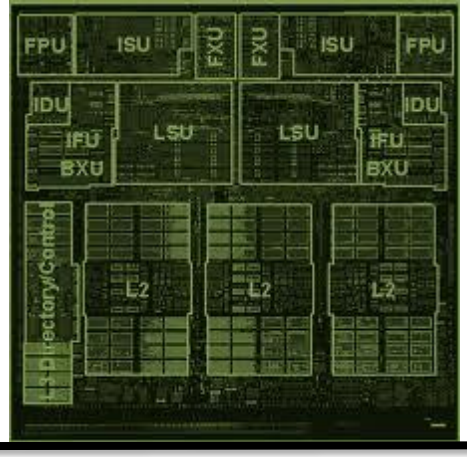
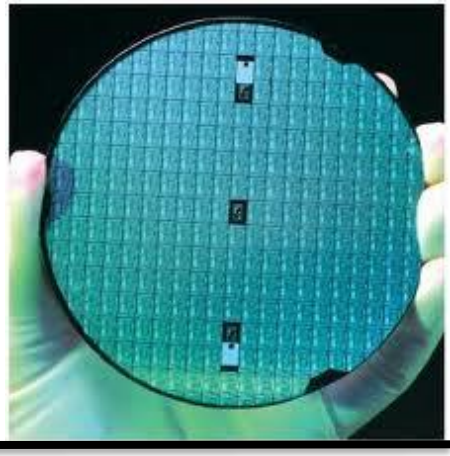






ECE 122A

VLSI Principles

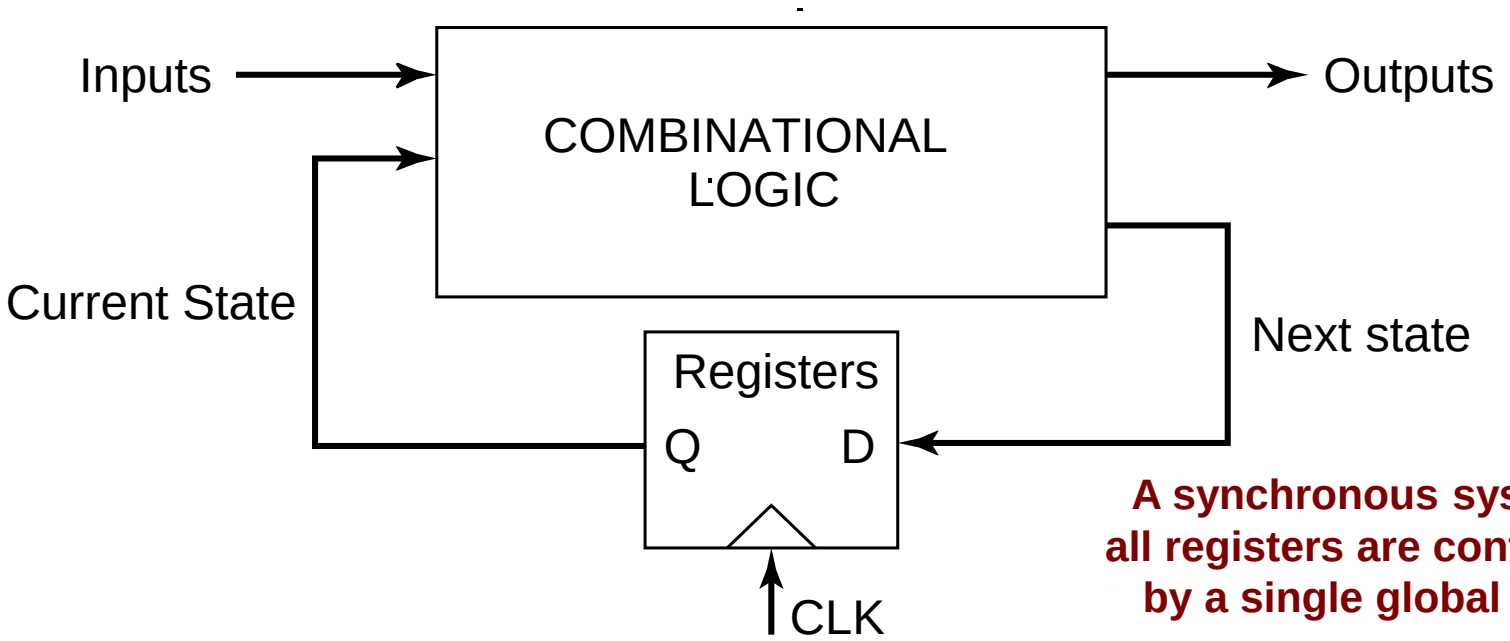
Lectures 16/17

Prof. Kaustav Banerjee
Electrical and Computer Engineering
University of California, Santa Barbara
E-mail: kaustav@ece.ucsb.edu

Designing Sequential Logic Circuits

Sequential Logic

All useful systems require storage of state information….



A synchronous system: all registers are controlled by a single global CLK

A generic Finite State Machine (FSM) consisting of combinational logic and registers.

Output of the FSM = *F* (*current inputs*, *current state*)

Next State is determined based on current state and current inputs—fed to the input (D) of the registers

At the rising edge of the CLK, D copied to Q (with some delay)

Note: There are 2 storage mechanisms: 1) positive feedback and 2) charge storage

Classification of Memory Elements

- Background Memory: large centralized memory core (high density array structures)---SRAMs and DRAMs
- Foreground Memory: embedded in a logic (individual registers or register banks)—focus of this section

Classification of Memory Elements

Static Memory:

- preserves state as long as power is ON
- built by using positive feedback or regeneration where the circuit consists of intentional connections between the output and input of a combinational circuit
- most useful when register will not be updated for extended periods of time (eg., configuration data loaded at power-up time).
- Condition also holds for most processors that use conditional clocking, (gated CLK) where the CLK is turned off for unused modules----**no guarantee on how frequently the registers will be clocked and static memories are needed to store information.**
- bistable element is the most popular form

Classification of Memory Elements

Dynamic Memory:

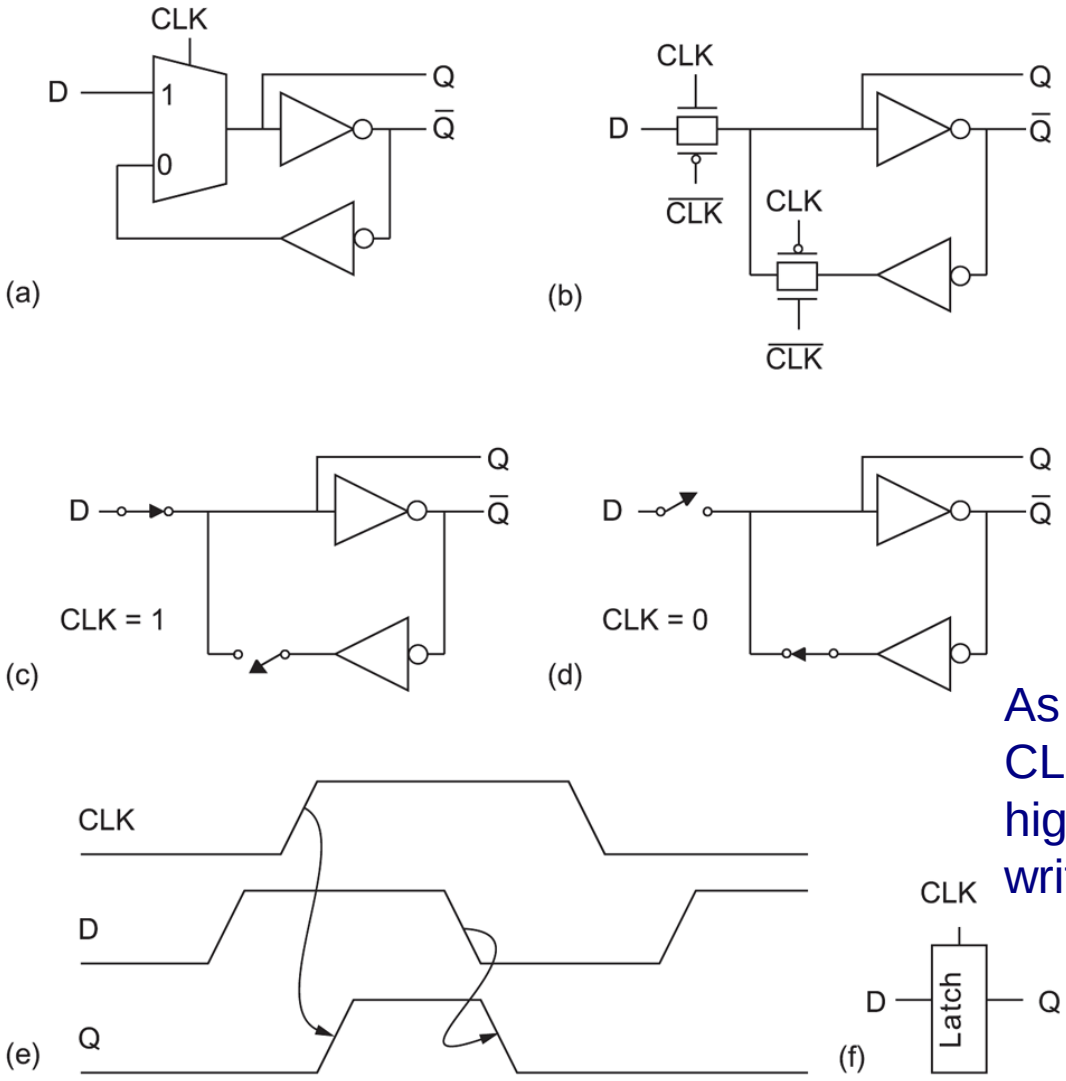
- store data for short (ms) period of time
- based on the principle of temporary charge storage on parasitic capacitors in MOS devices
- similar to dynamic logic….. capacitors need to be refreshed periodically to compensate for charge leakage
- significantly simpler----hence, provide higher performance and lower power dissipation
- most useful in datapath circuits that require higher performance levels and are **periodically clocked**

Naming Conventions

- Definitions:
  - a latch is a level sensitive device
  - a register is an edge-triggered storage element
- There are many different naming conventions
  - For instance, many books call edge-triggered elements flip-flops
  - This may lead to confusion however…
  - Any bistable component formed by the cross coupling of gates is a flip-flop (FF)

Latches

Multiplexer based



CLK=1: D to Q

CLK=0:Holds state of Q

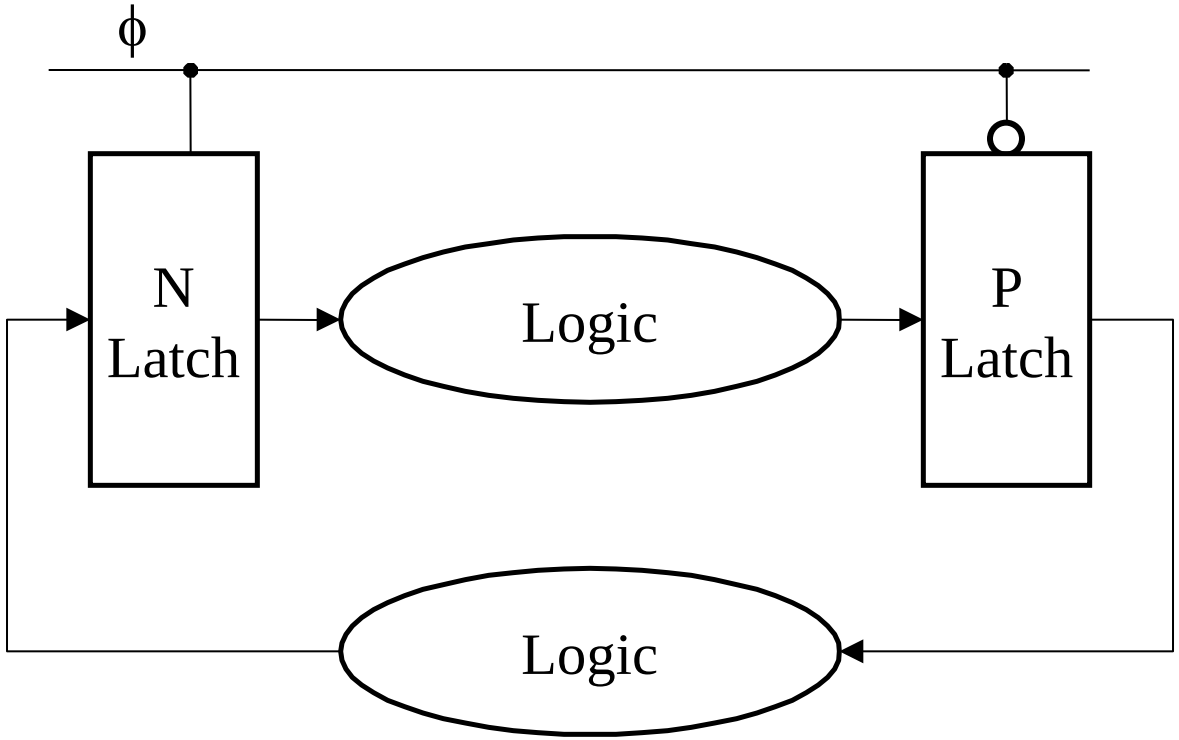
As long as CLK remains high, D will be written on Q

FIG 1.30 CMOS positive-level-sensitive D latch

Latch-Based Design

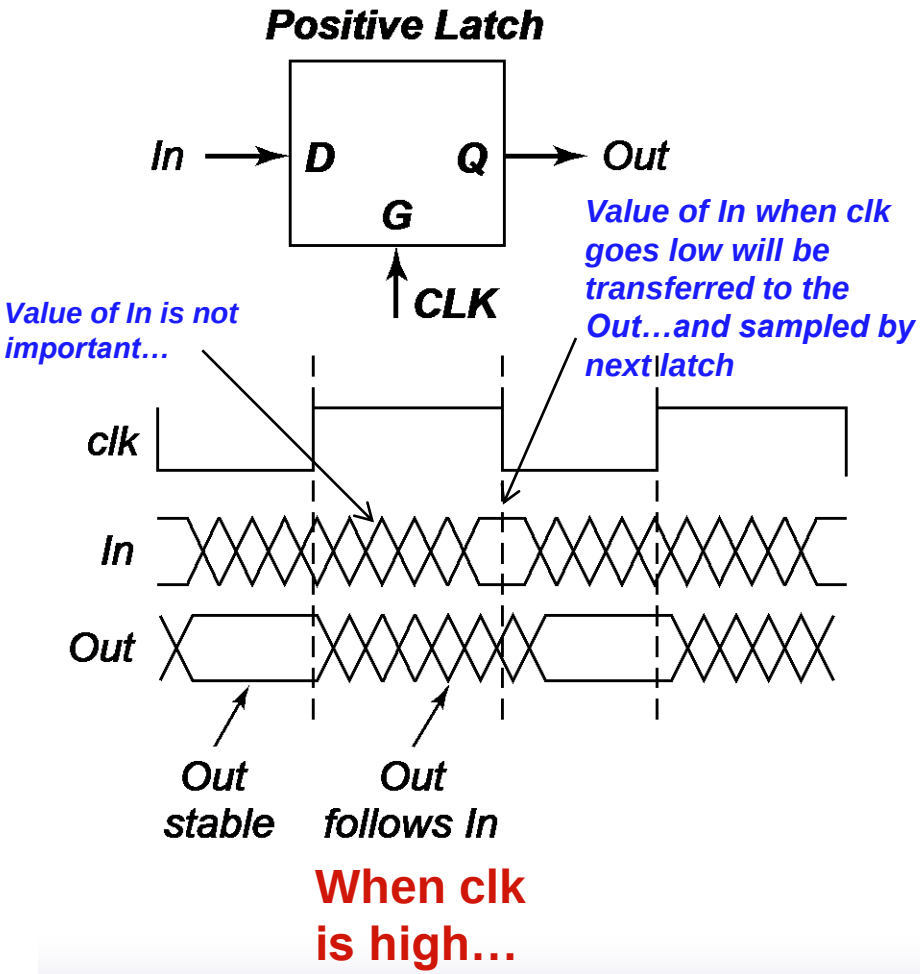
• N (negative) latch is transparent when $\phi = 0$

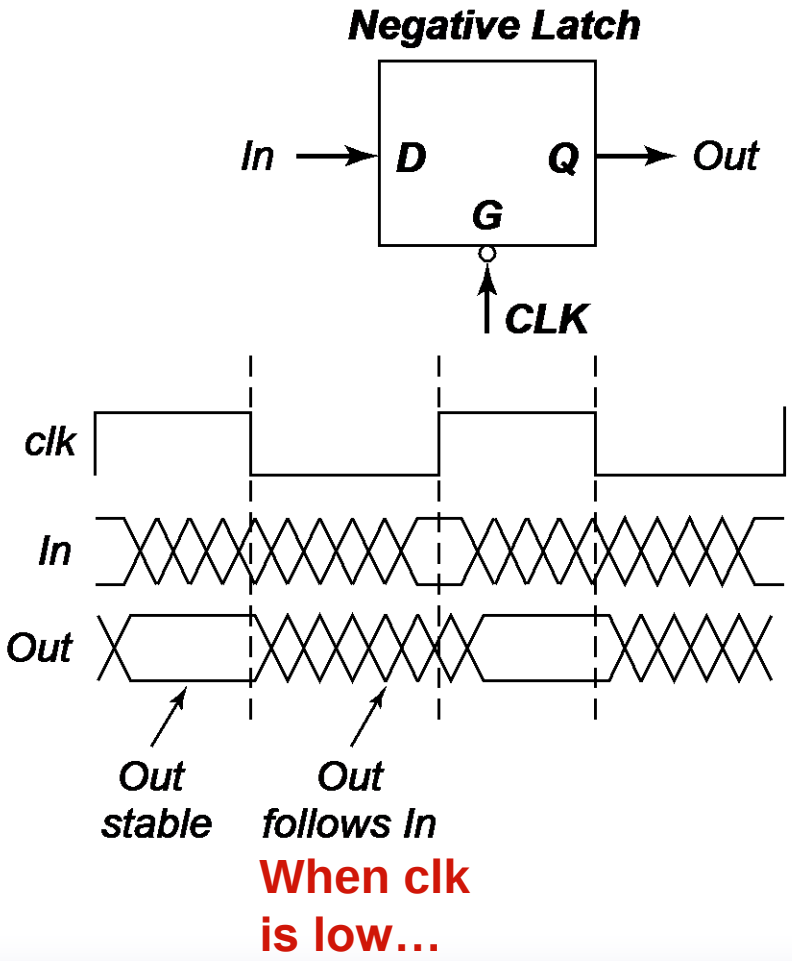
• P (positive) latch is transparent when $\phi = 1$



Difficult to eliminate Race conditions…..under CLK overlap

Timing of P/N Latches





Latch versus Register

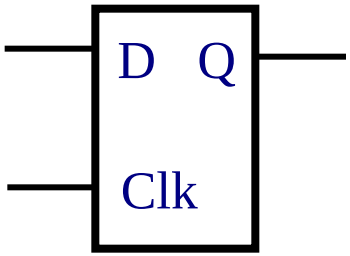
- Latch

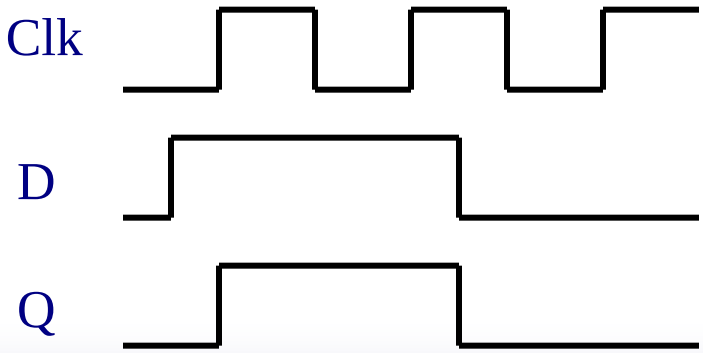
  stores data when clock is **high** (or low)

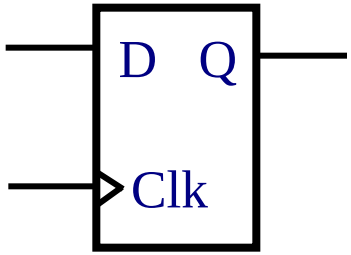
- Register

  stores data when clock **rises** (or falls)

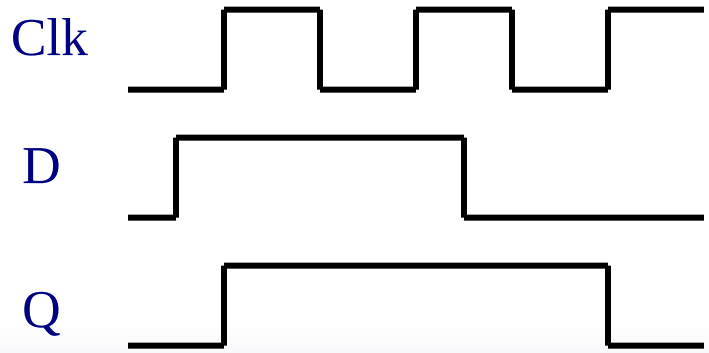
What kind of latch is this?



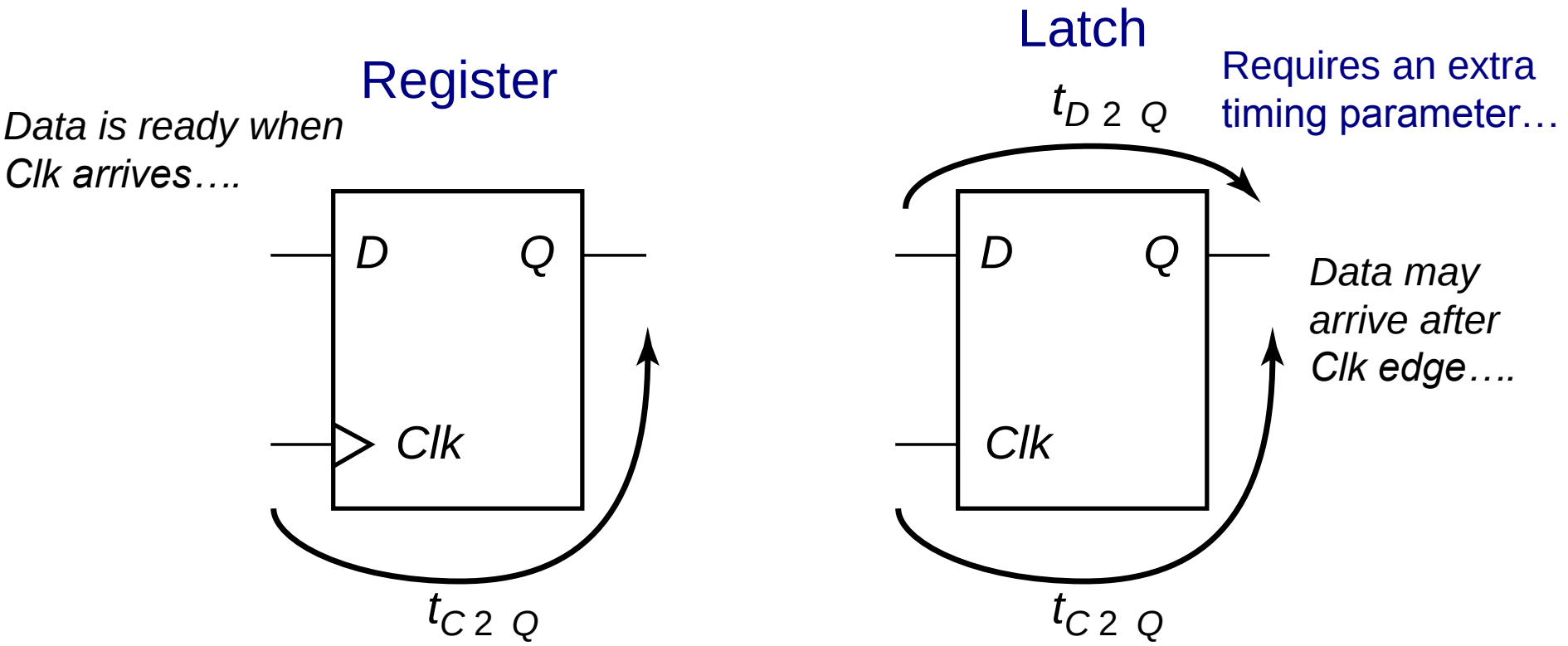




What kind of Register is this?



Characterizing Timing



Note: In a FF, D is valid before CLK edge arrives……hence only c2q is relevant.

In a Latch, the relevant timing parameter (c2q or d2q) depends on the relative position of the arrival of D w.r.t the clk edge……if D arrives after clk edge, then d2q is important, while if D arrives before clk edge, c2q is important.

Registers or Flip-Flops

Combines two latches:

One +ve sensitive (slave) and one –ve sensitive latch (master)

Edge Triggered FF or Master-Slave FF

CLK=0: D to QM

$\overline{QM} = \overline{D}$

Slave holds previous value of Q

CLK=1: master can't sample input and holds value of D

Slave opens and QM=(D) =Q

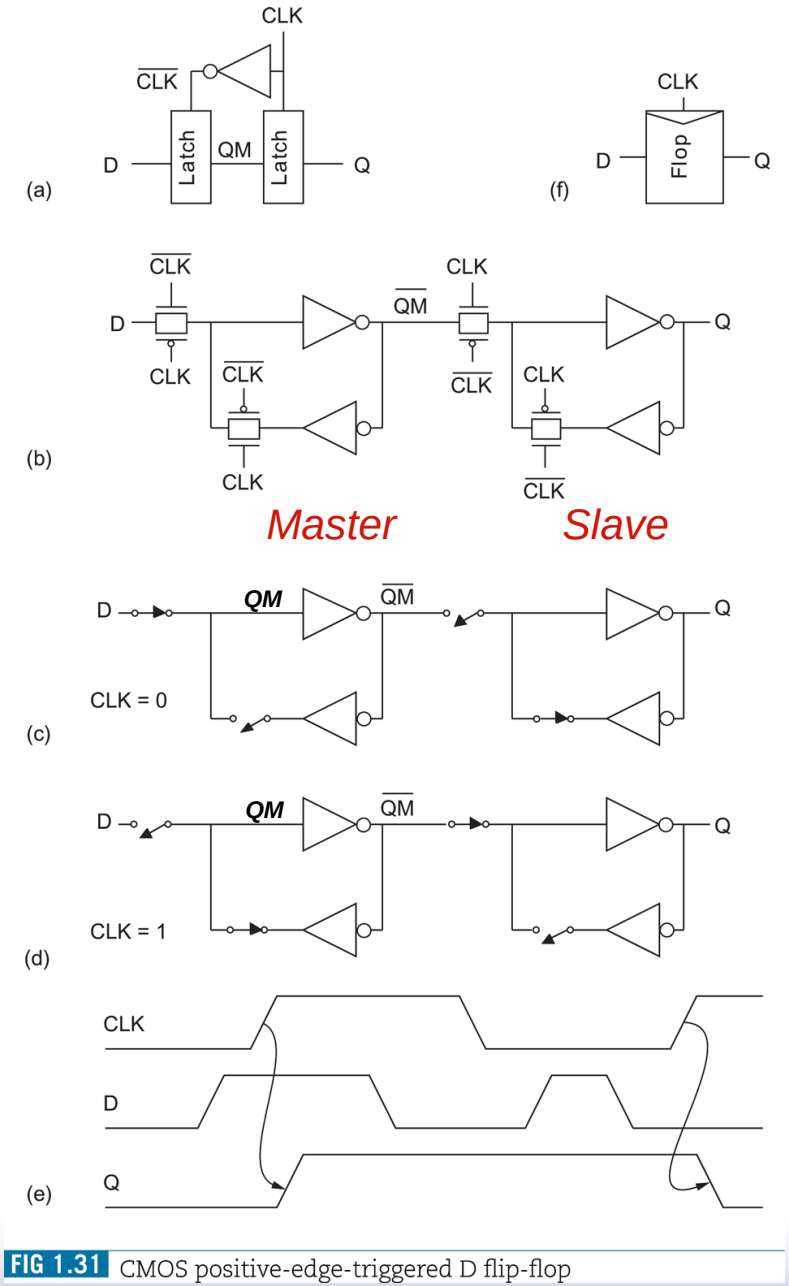
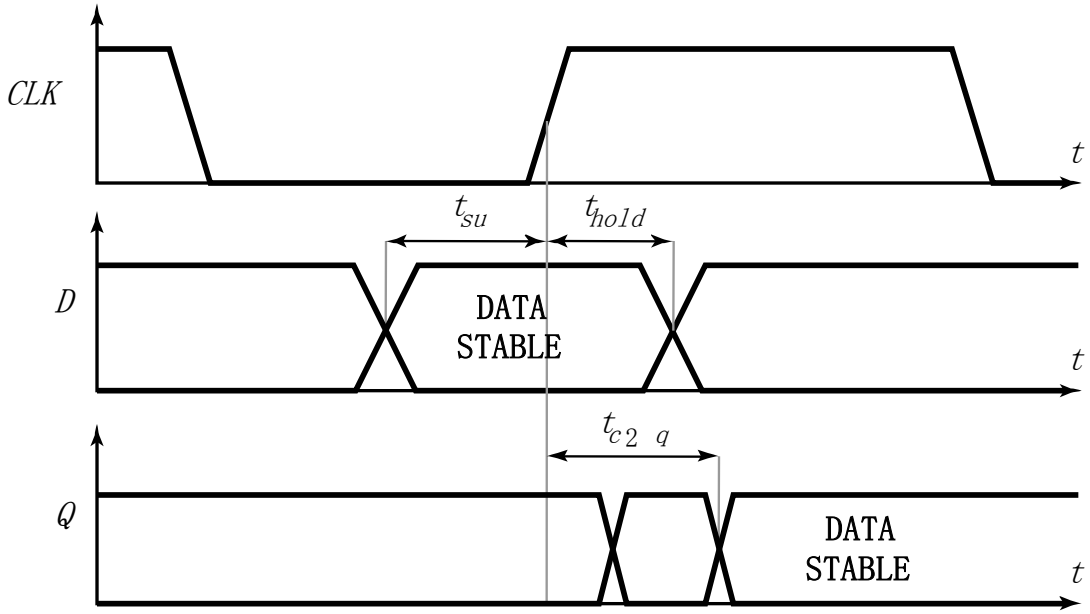
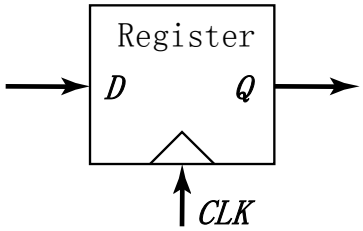


FIG 1.31 CMOS positive-edge-triggered D flip-flop

Timing Definitions



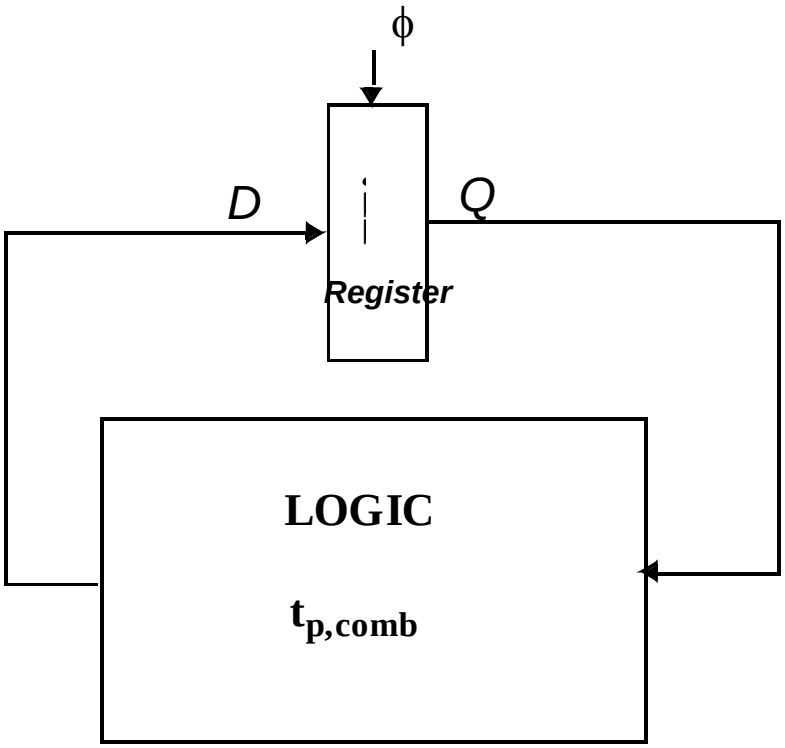


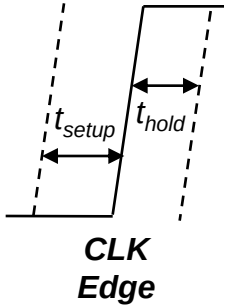
t_{su}= setup time =time for which the data inputs (D) must be valid before the CLK edge

t_{hold}= hold time =time for which data input must remain valid after the CLK edge

t_{c2q}= worst case propagation time through the Register (w.r.t the CLK edge)

Maximum Clock Frequency





1) $T_{min} = t_{clk\text{-}Q} + t_{p,comb} + t_{setup}$

Clk period must accommodate the longest delay of any stage in the network

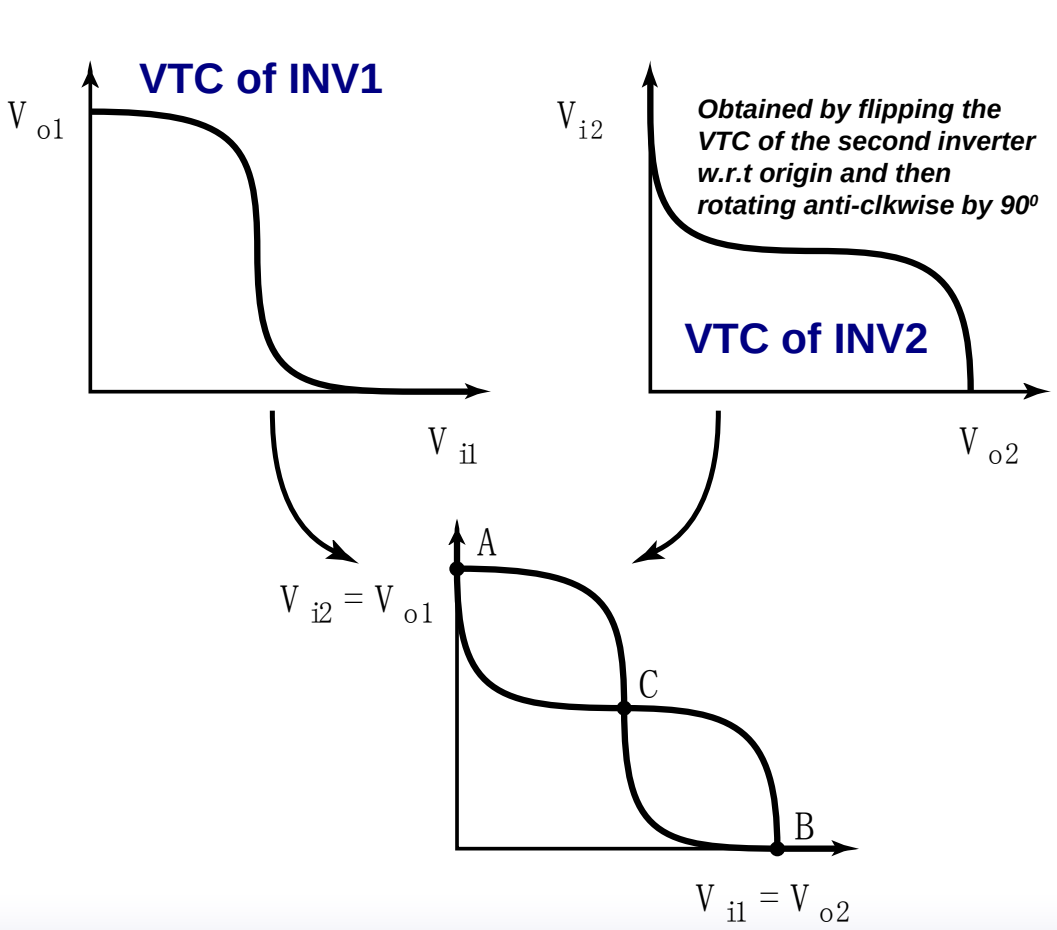
2) $t_{cdreg} + t_{cdlogic} > t_{hold}$

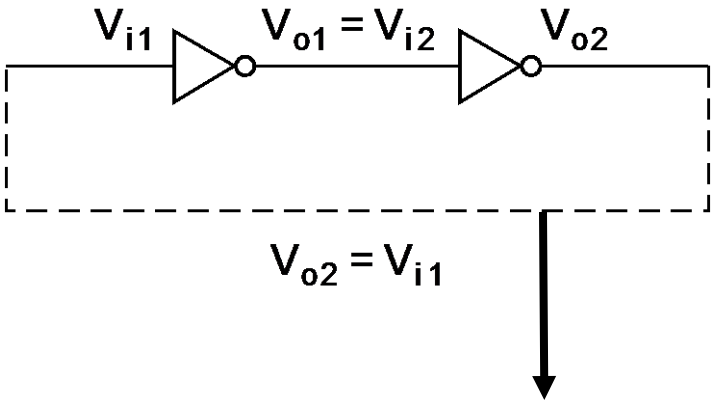
t_{cd}: contamination delay = minimum delay

To ensure that the input data of the sequential elements is held long enough after the CLK edge and is not modified too soon by the new wave of data coming in:

Static Latches and Registers

Static Memories use Positive Feedback: Bi-Stability





A, B, and C are the only three possible operating points

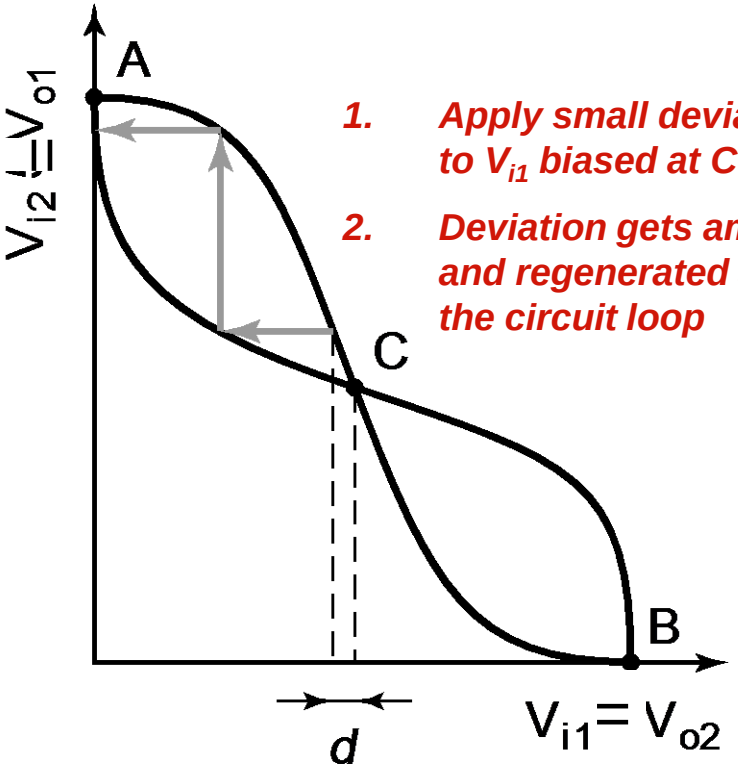
If gain>1 in the transient region: A and B are the only stable operating points, C is a metastable point

Point C is the V_M of the Inverters…

Meta-Stability in Bi-Stable Circuits

Point C is the V_M of the Inverters…

A: $V_{i1}=0$, $V_{i2}=1$

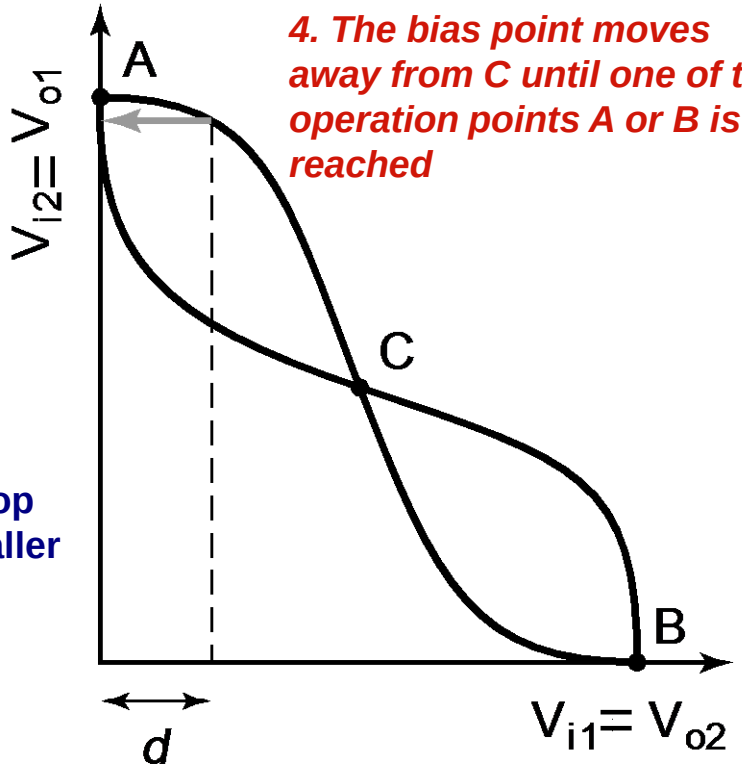


1. ***Apply small deviation d to V_{i1} biased at C***
2. ***Deviation gets amplified and regenerated around the circuit loop***

At A and B the loop gain is much smaller than 1…hence, stable points

3. Deviation is amplified by the gain of the first inverter and then further amplified by the second inverter

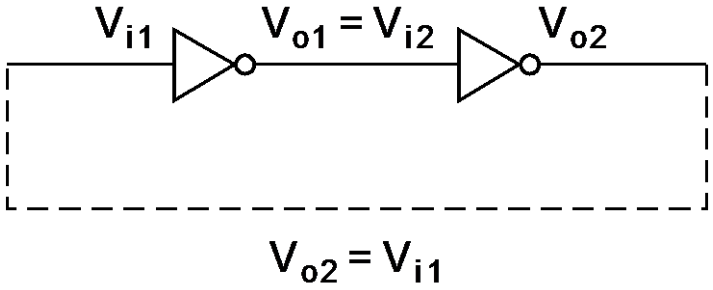
4. The bias point moves away from C until one of the operation points A or B is reached



- **Gain is larger than 1 in the transition region**
- **Every small deviation causes the operation point to move away from its original bias point, C ---therefore metastable**

Flip-Flop

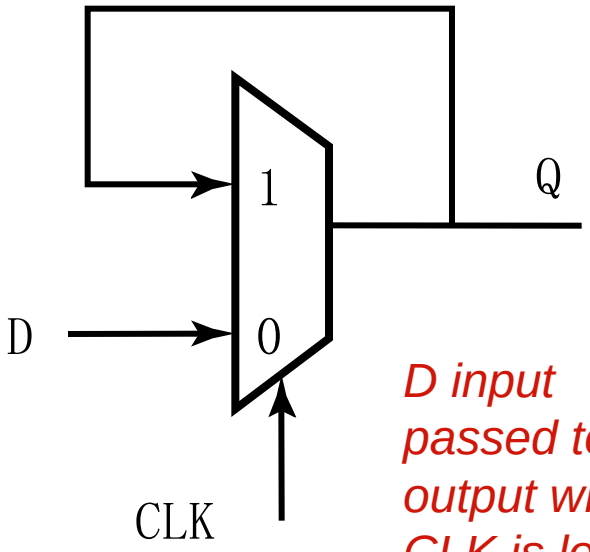
A cross coupled pair of inverters results in a bistable circuit…



- A FF is a bistable circuit, which has 2 stable states
- In the absence of triggering the circuit remains in a single state
- The state can be changed by applying an external trigger
- Two ways to achieve a change of state:
  - Cut the feedback loop: so that a new value can be written into **out** or **Q**
    - **This is MUX based: Q=CLK . Q + CLK . In (most common)**
  - Overpower the feedback loop
    - Apply a trigger signal at the input of the FF and force the new value into the cell by overpowering the stored value
    - Needs careful sizing of transistors in the feedback loop and the trigger circuit
    - Used mostly in static background memories

Mux-Based Latches

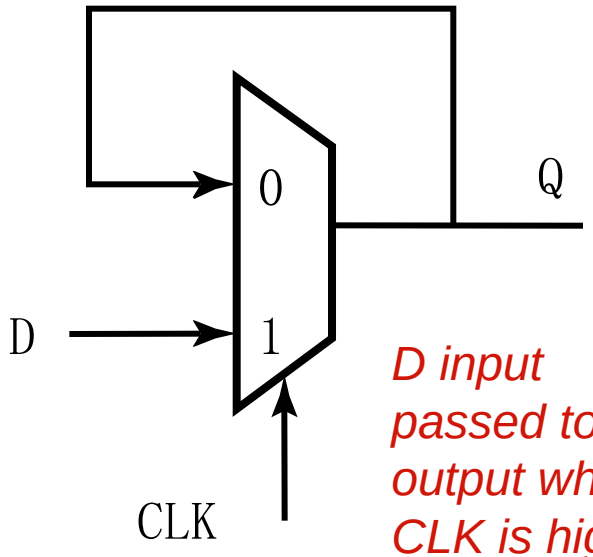
Negative latch (transparent when CLK= 0)



D input passed to output when CLK is low

$$Q = Clk \cdot Q + \overline{Clk} \cdot In$$

Positive latch (transparent when CLK= 1)

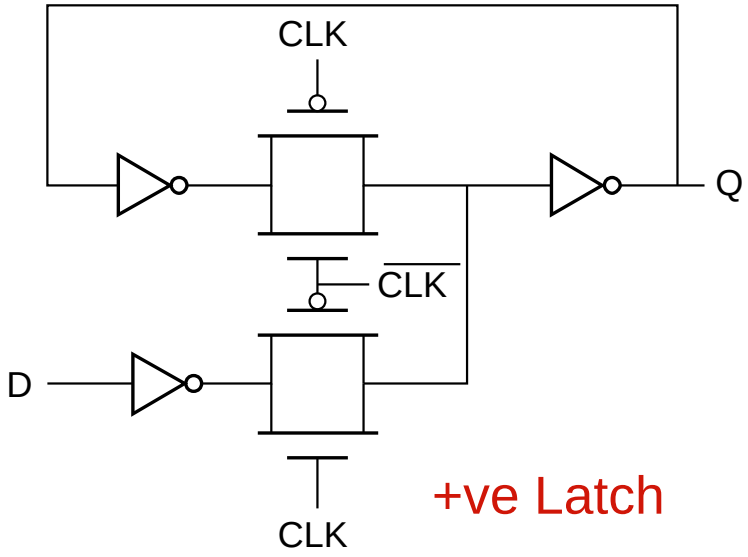


D input passed to output when CLK is high

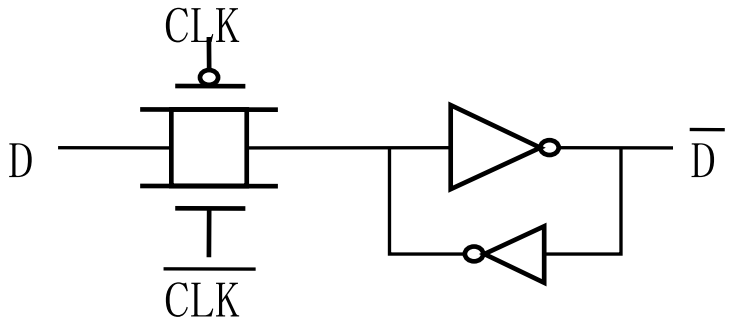
$$Q = \overline{Clk} \cdot Q + Clk \cdot In$$

Writing into a Static Latch

Use the clock as a decoupling signal,
that distinguishes between the transparent and opaque states



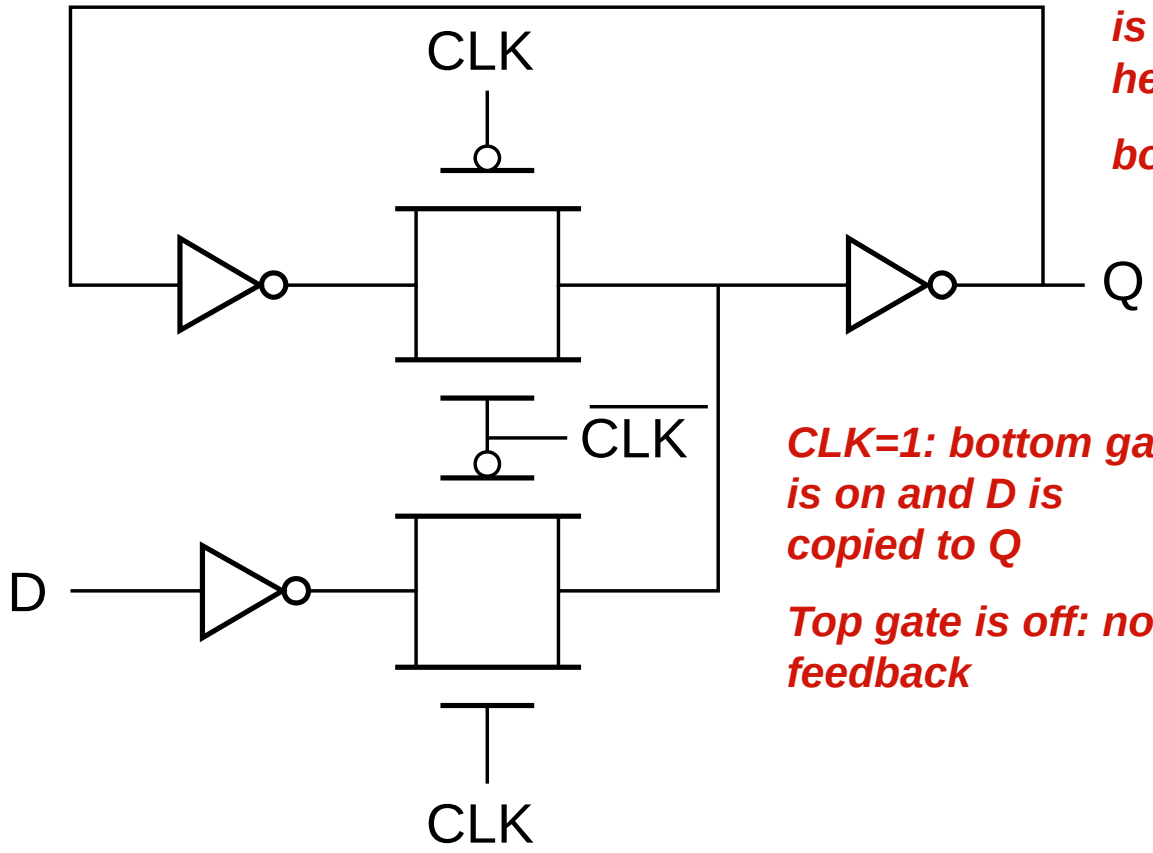
+ve Latch



MUX based (not so efficient….# of transistors driven by CLK is high= 4)

Forcing the state
(can implement as NMOS-only)

Mux-Based Latch



CLK=0: top gate is on and Q is held stable

bottom gate is off

CLK=1: bottom gate is on and D is copied to Q

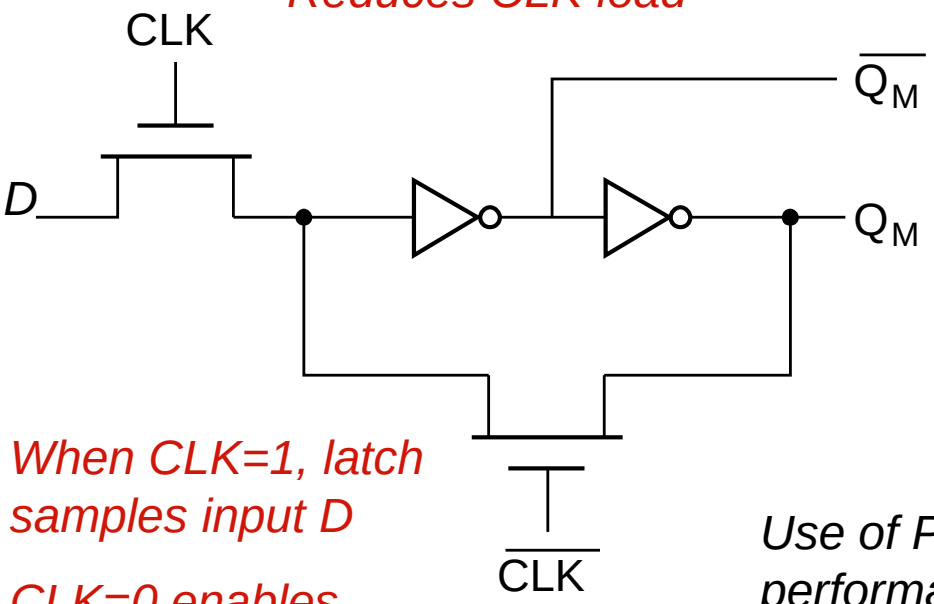
Top gate is off: no feedback

CLK is driving several transistors with activity=1 (not good from power perspective!)

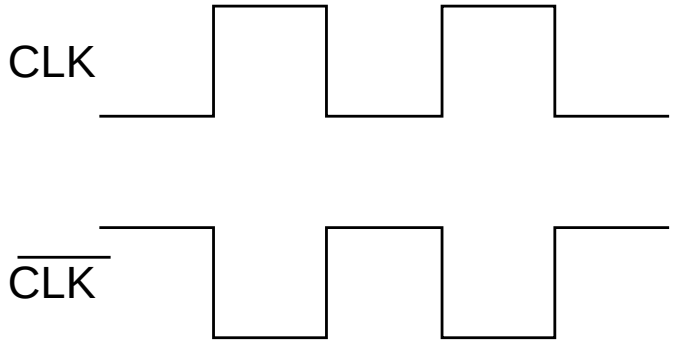
Mux-Based Latch with Reduced Load

NMOS only Pass Transistor

Reduces CLK load



Need Non-overlapping clocks

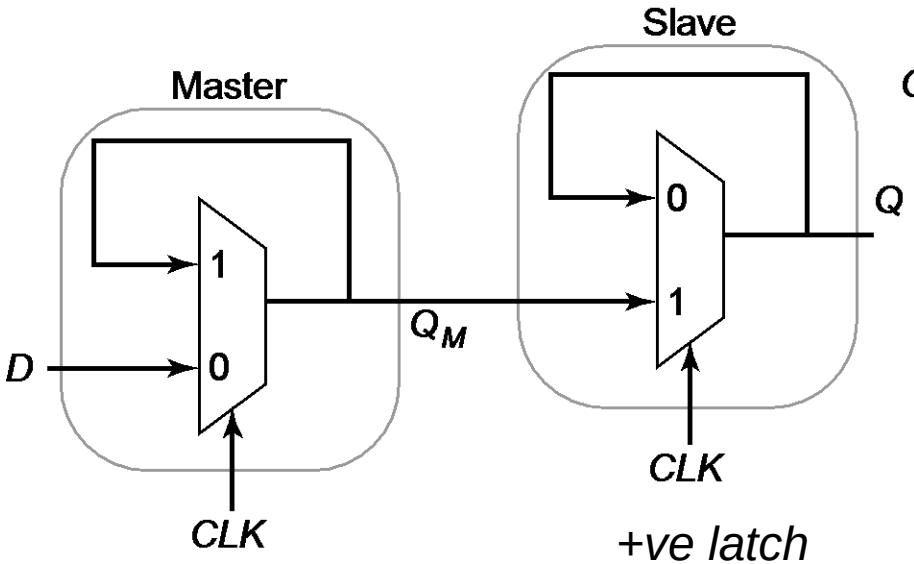


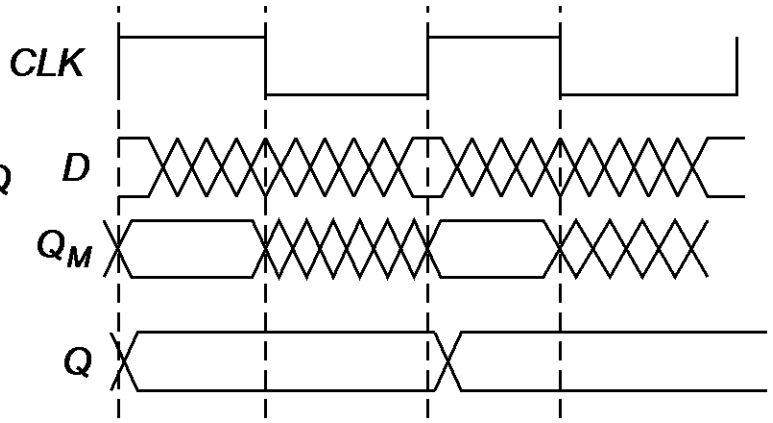
When CLK=1, latch samples input D

CLK=0 enables feedback loop, puts latch in hold mode

Use of PT degrades NM and switching performance by passing V_{dd}-V_{Tn} to the input of first inverter + increases static power (PMOS of inverter is never fully turned off)

Master-Slave (Edge-Triggered) Register





Two opposite latches trigger on edge
Also called master-slave latch pair

CLK=0: master is transparent and D is copied to QM

During this time slave is in hold mode

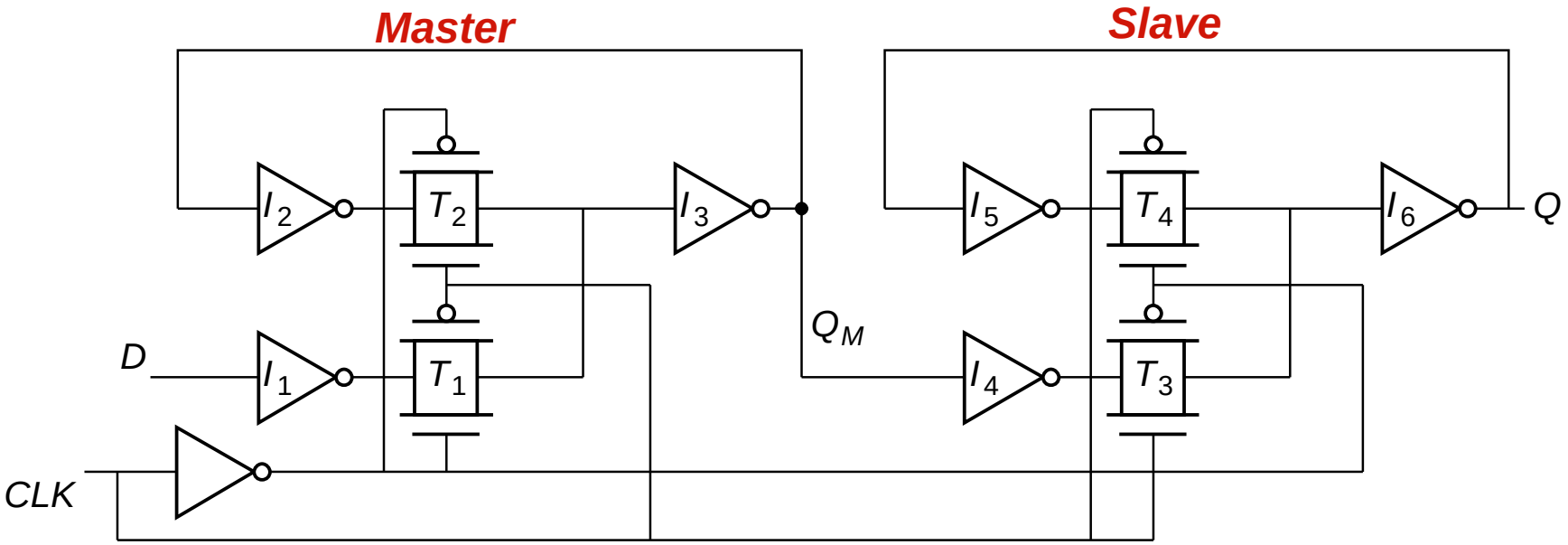
As CLK=1: slave starts sampling, master in hold mode

Value of Q=Value of D right before the rising edge of the CLK: +ve edge triggered effect

Master-Slave +ve Edge Triggered Register

Transistor Level Implementation

X-gate Multiplexer-based latch pair



CLK=0: T1 is on, T2 is off, D input sampled onto QM

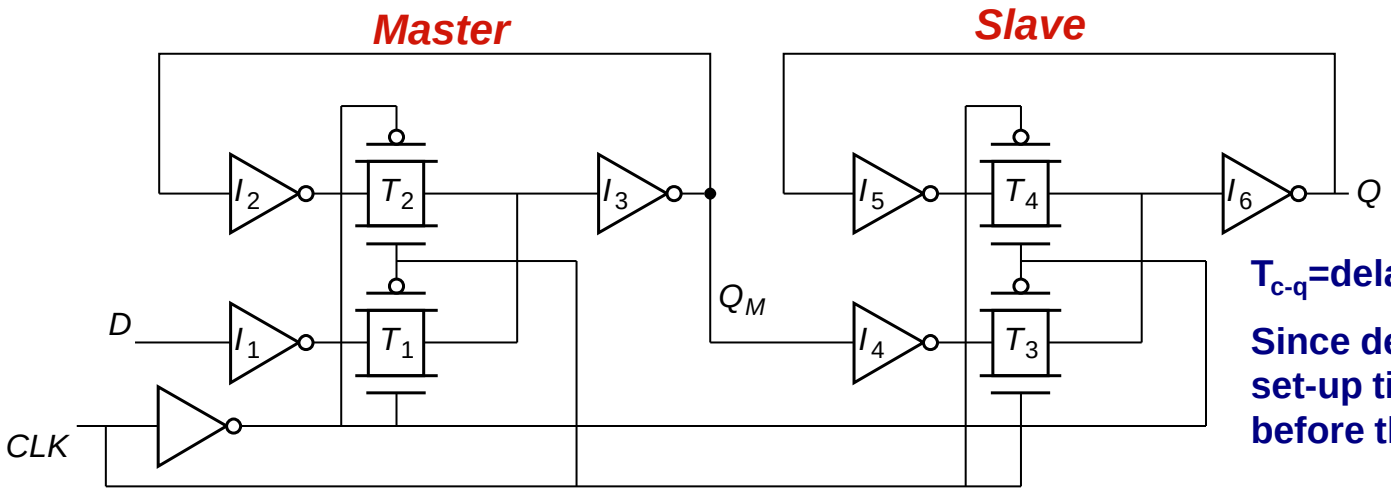
T3 off and T4 on: I5 and I6 hold the state of the Slave

CLK=1: T3 is on, T4 is off, QM sampled onto Q

T2 on and T1 off: I2 and I3 hold the state of QM

Master-Slave +ve Edge Triggered Register

Transistor Level Implementation



$T_{c\text{-}q}$=delay through T3 and I6

Since delay of I2 is included in set-up time output of I4 is valid before the rising edge of CLK

$T_{c\text{-}q} = t_{pd\_inv} + t_{pd\_tx}$

t_{su} = set-up time = time before the rising edge of the CLK during which the D input should remain stable so that QM samples the value reliably

Since D must propagate through I1, T1, I3, and I2 before the rising edge

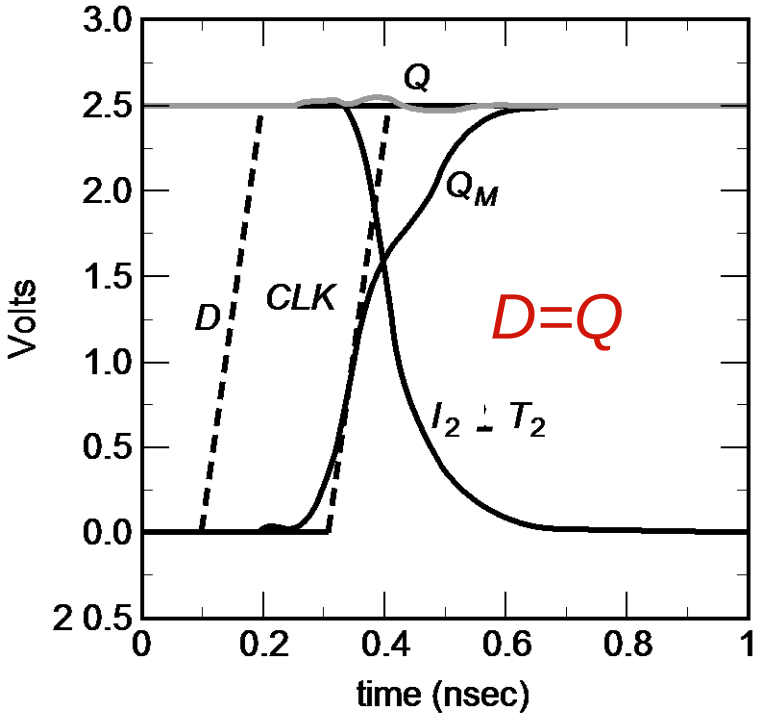
$t_{su} = 3t_{pd\_inv} + t_{pd\_tx}$

To ensure equal node voltages on both sides of the Xgate

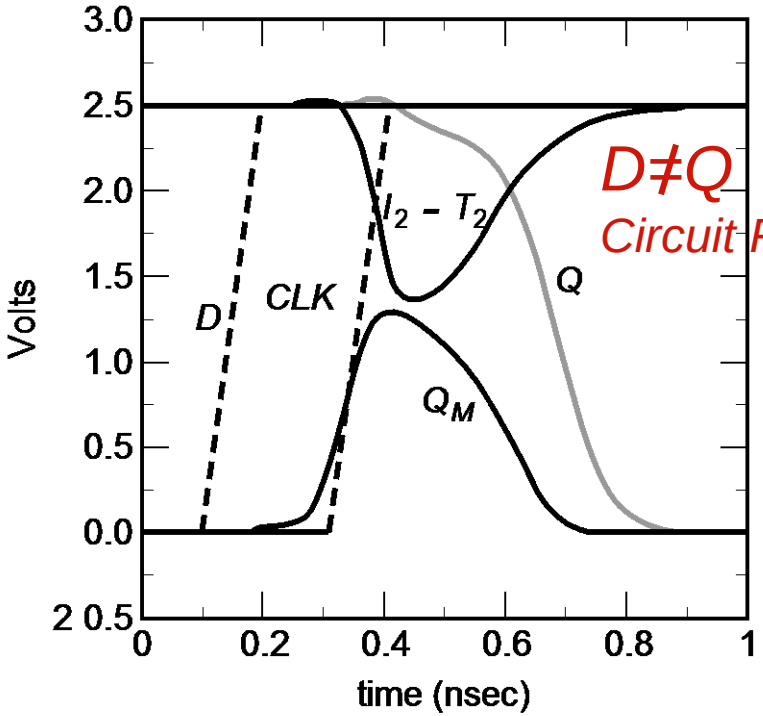
$t_{hold} = 0$ (since T1 is cut off after CLK edge)

Timing Analysis: Setup Time

SPICE Simulations: progressively skew the input w.r.t CLK edge until the circuit fails



(a) T_{setup} = 0.21 nsec



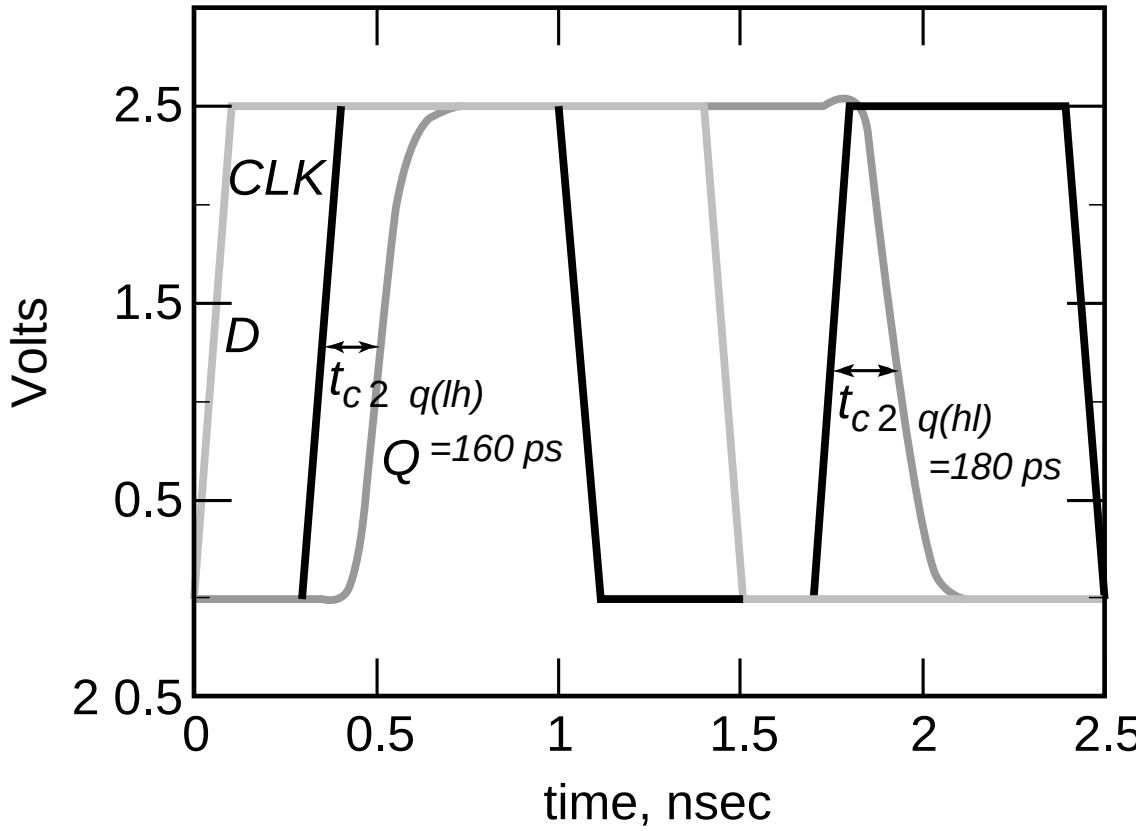
(b) T_{setup} = 0.20 nsec

CLK is enabled before the voltage across T2 settles to the same value

Set-up time for this register = 210 ps and hold time =0

Clk-Q Delay

$t_{c\text{-}q}$ = 50% point of CLK to 50% point of Q

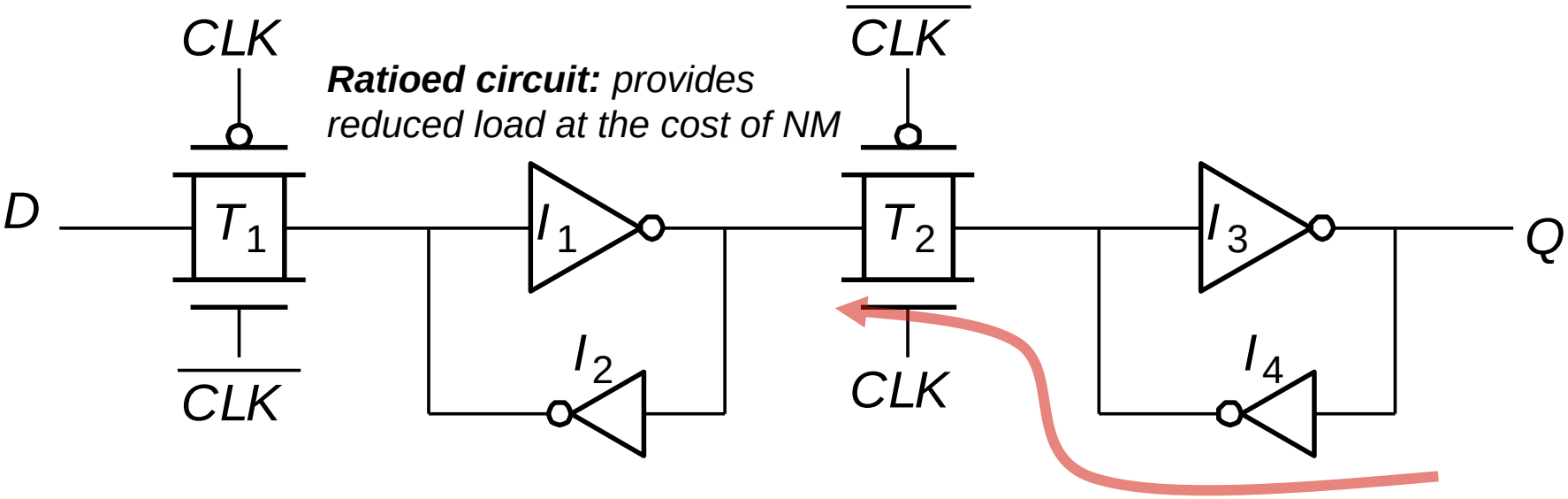


Reduced Clock Load Master-Slave Register

Note: *X-gate register presents high capacitive load to the CLK signal*

Minimum sized devices are desired for X-gates….why? (CLk power)

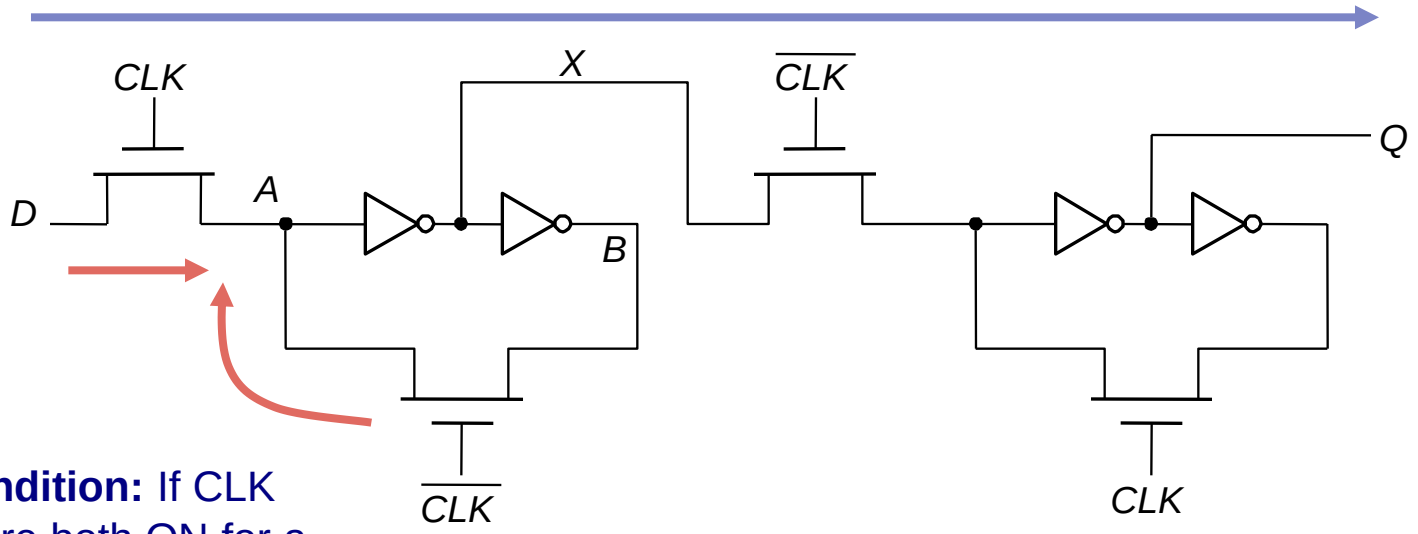
However, input to I1 must be brought below its switching threshold….to make a transition. Hence, for minimum sized X-gate, I2 should be made even weaker…..by increasing L_{ch}.



Cons: 1) T1 and its source driver must overpower I2 to switch the state of the cross-coupled inverter

2) Reverse conduction---second stage (T2 and I4) can affect the state of the first latch (I1-I2) when slave stage is ON…..not a major problem if I4 is weak.

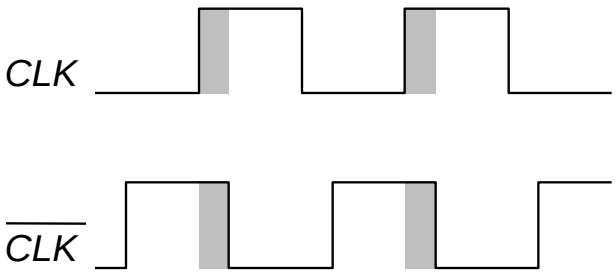
Avoiding Clock Overlap



(a) Schematic diagram of an NMOS only –ve M-S register

RACE Condition: If CLK and $\overline{CLK}$ are both ON for a short time, both sampling pass transistors are ON providing a direct path from D to Q. Hence, data at the output can change at the rising CLK edge

Also node A can be driven by both D and B: undefined state



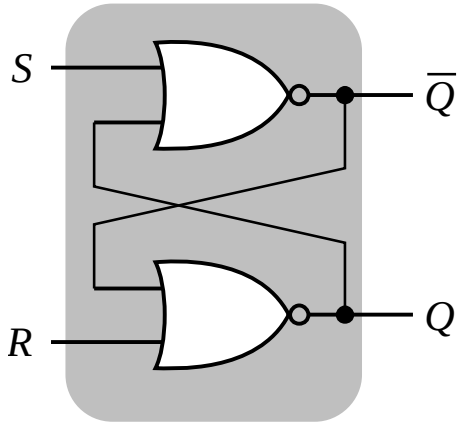
(b) Overlapping clock pairs

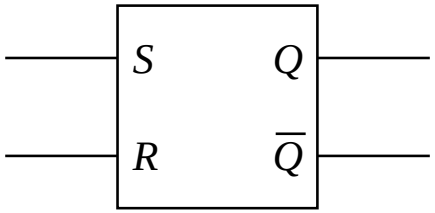
Clock skew is a problem!

One Solution: use non-overlapping CLKs

Overpowering the Feedback Loop — Cross-Coupled Pairs

NOR-based set-reset (SR)-FF





S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
1	0	1	0
0	1	0	1
1	1	0	0

Forbidden State

Use external triggers, S and R to change the output states (Q and $\overline{Q}$):

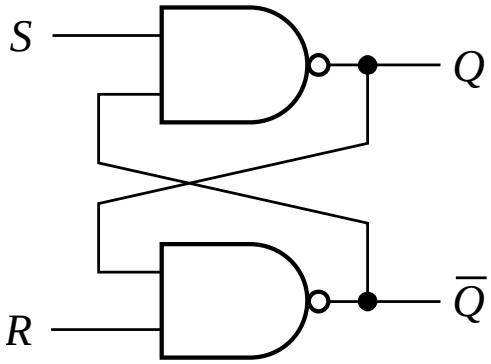
S=1 forces Q=1,

R=1 forces Q=0

Note: A NOR gate with one of its inputs=0, looks like an inverter and the above structure looks like a cross-coupled inverter

SR-FF using Cross-Coupled NAND

Cross-coupled NANDs

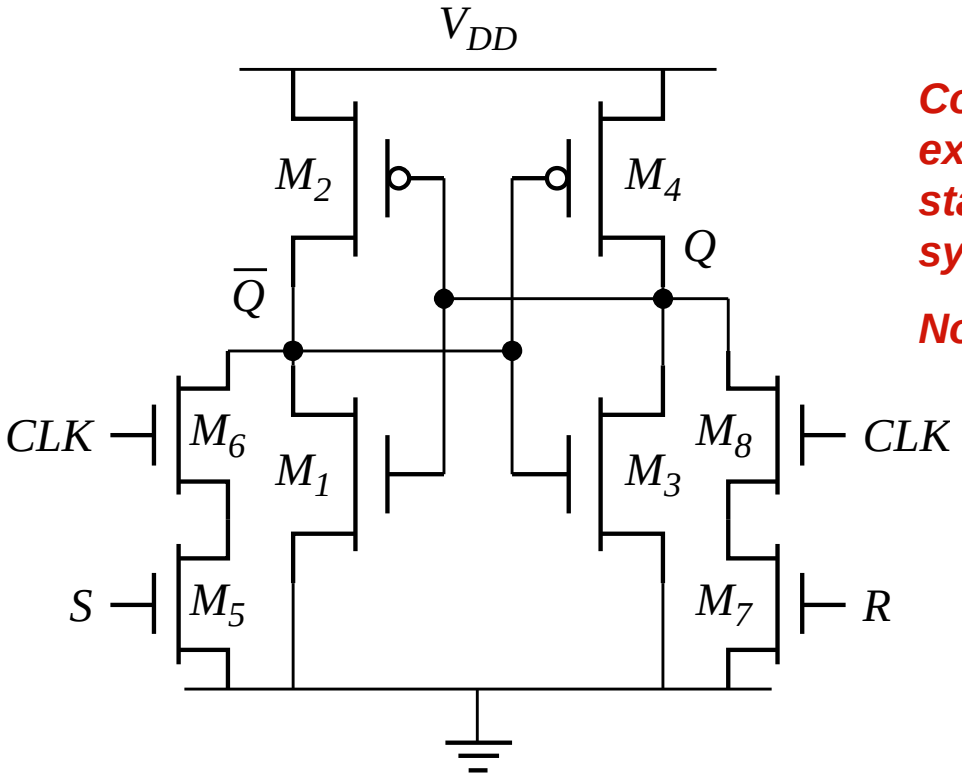


This is not used in datapaths any more, but is a basic building memory cell

Note: These FFs are purely asynchronous….doesn't match with synchronous design method

Need a Clocked Latch!

Ratioed CMOS Clocked SR Latch



Consists of two cross-coupled inverters + 4 extra transistors to drive the FF from one state to another and to provide synchronization

No static path between Vdd and Gnd

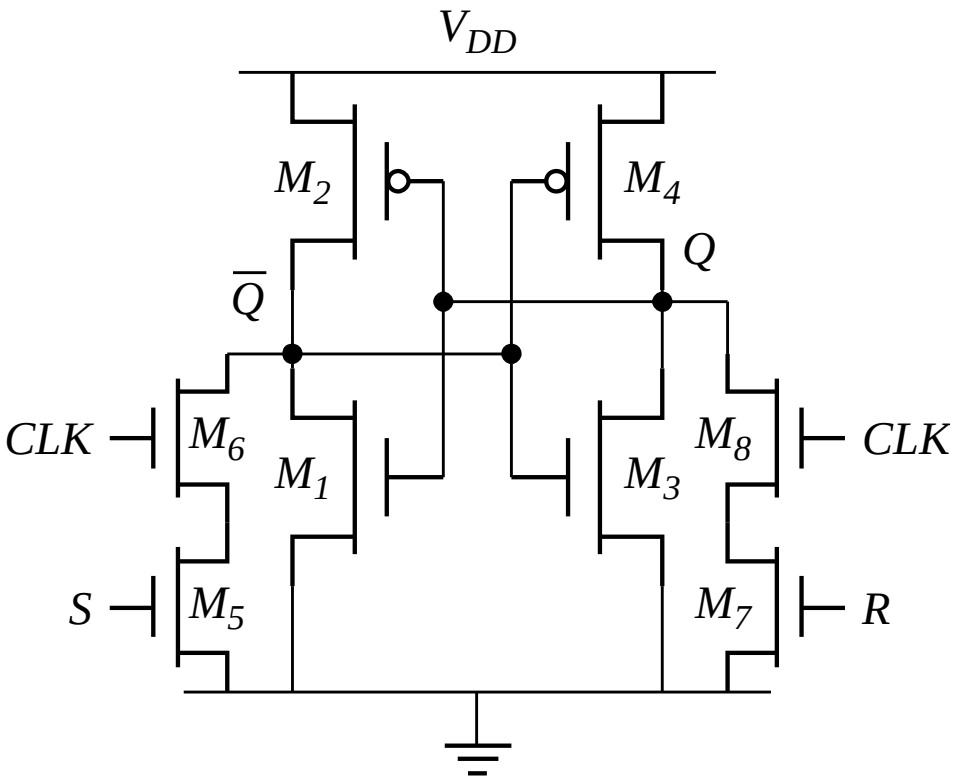
Transistor sizing is essential to ensure FF transition:

If Q is high and R=1, then V_Q must be < V_M of INV M1-M2, to make the latch switch.

Similar condition is needed to switch INV M3-M4 for S=1.

This means we must increase the sizes of M5, M6, M7 and M8. M4-M7-M8 (and M5-M6-M2) form ratioed inverters.

CMOS Clocked SR Latch (Cont'd)



For a 0.25 um technology, select the following sizes:

$(W/L)_{M1}=(W/L)_{M3}$ = (0.5 um /0.25 um)

$(W/L)_{M2}$ = $(W/L)_{M4}$ = (1.5 um/0.25 um).

Assuming Q =0, how do we determine the minimum sizes of M5, M6, M7 and M8 to make the device switchable?

To switch from Q=0 to Q=1, the low-level of the pseudo-NMOS inverter (M5-M6)-M2 should be below the V_M of the inverter M3-M4 (= $V_{DD}/2$).

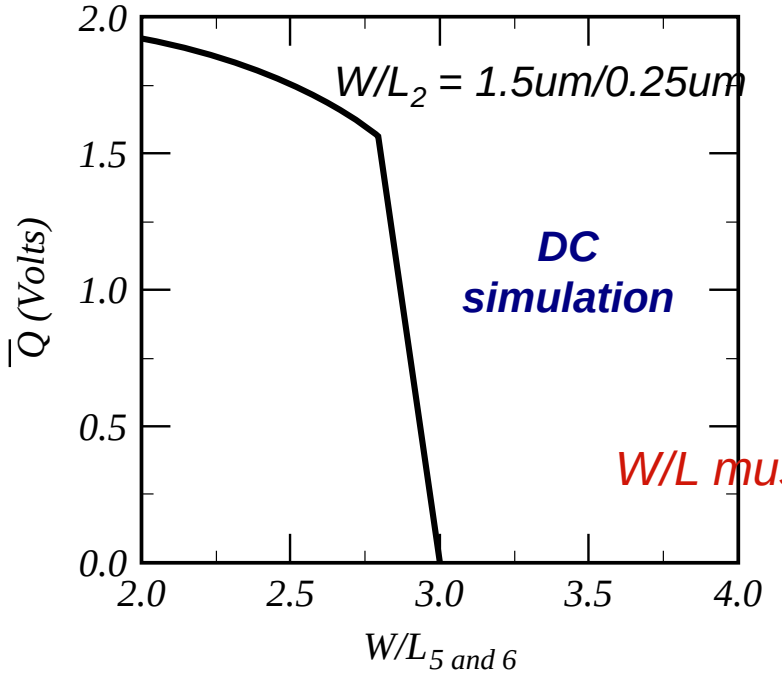
As long as $V_{\overline{Q}} > V_M$, V_Q=0 and gate of M2 is grounded.

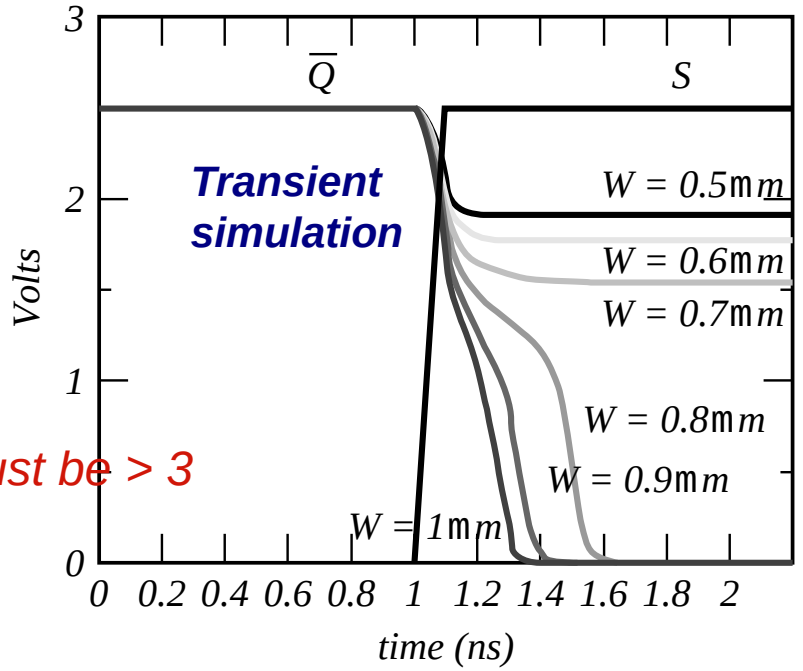
The boundary condition on the transistor sizes can be derived by equating the currents in the inverter for $V_{\overline{Q}}$ = Vdd/2. The currents are derived by the saturation current since $V_S = V_{DD}$ =2.5 V and V_M = 1.25 V.

Sizing Issues for Clocked SR FF

Output voltage dependence on transistor width

0.25 um process





W/L must be > 3

Assume that M5 and M6 have identical sizes and that $(W/L)_{5\text{-}6}$ is the effective ratio of the series connected devices. Under this condition, the PD network can be modeled by a single transistor M5-6, whose length is twice the length of the individual devices:

$$k_n'\left(\frac{W}{L}\right)_{M5-6}\left[(V_{DD}-V_{Tn})V_{DSATn}-\frac{V_{DSATn}^2}{2}\right]=-k_p'\left(\frac{W}{L}\right)_{M2}\left[(-V_{DD}-V_{Tp})V_{DSATp}-\frac{V_{DSATp}^2}{2}\right]$$

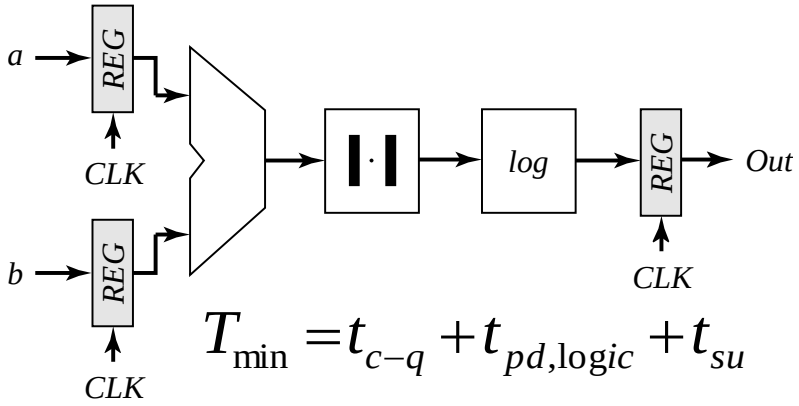
This results in $(W/L)_{M5\text{-}6}$ = 2.26.

Note: This would imply that the individual device sizes of M5 and M6 be 4.5…..somewhat higher than that predicted by simulation (=3) ….due to second order effects like Channel-length modulation and DIBL.

Pipelining: Optimizing Sequential Circuits

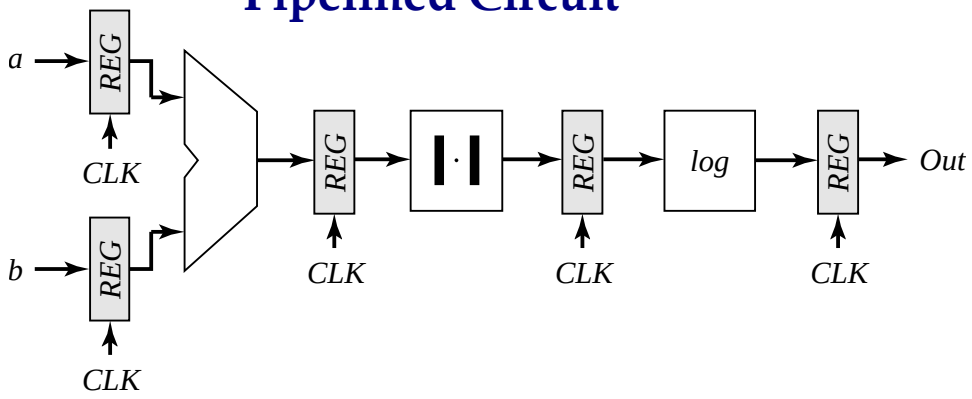
Widely used to accelerate the operation of datapaths in digital microprocessors…

Reference Circuit: computes log (|a+b|)



$$T_{\min} = t_{c-q} + t_{pd,\mathrm{logic}} + t_{su}$$

Pipelined Circuit



$$T_{\min,pipe} = t_{c-q} + \max\left(t_{pd,adder} + t_{pd,abs} + t_{pd,\log}\right) + t_{su}$$

Clock Period	Adder	Absolute Value	Logarithm
1	$a_1 + b_1$		
2	$a_2 + b_2$	$\lvert a_1 + b_1 \rvert$	
3	$a_3 + b_3$	$\lvert a_2 + b_2 \rvert$	$\log(\lvert a_1 + b_1 \rvert)$
4	$a_4 + b_4$	$\lvert a_3 + b_3 \rvert$	$\log(\lvert a_2 + b_2 \rvert)$
5	$a_5 + b_5$	$\lvert a_4 + b_4 \rvert$	$\log(\lvert a_3 + b_3 \rvert)$

Computation of one set of input data spreads over several clock cycles.

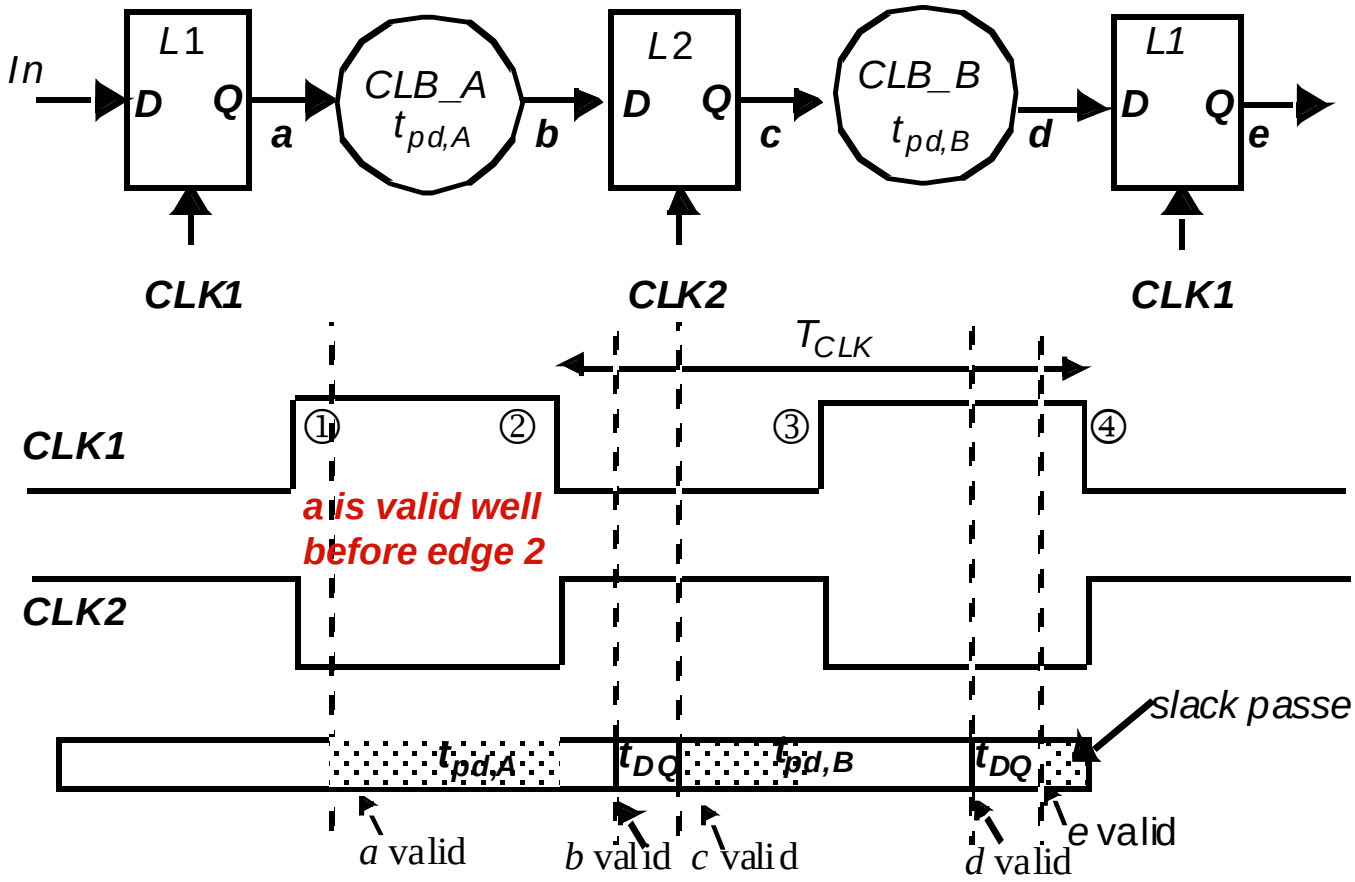
Pipelining improves resource utilization and increases functional throughput.

Register vs Latch Based Clocking…

- In an edge-triggered system, the worst case logic path between two registers determines the minimum CLK period for the entire system….
- If the logic block finishes before the end of the CLK period, it has to sit idle until the next CLK edge…
- Latch based design offers more flexibility…. one stage can *pass slack* or *borrow time* from other stages…

Slack-borrowing

In a latch based system, it is possible for a logic block to utilize time that is left from a previous logic block…



CLB_A starts computing right away and uses the slack to finish well before its allocated time (edge 3)….and so on

If $T_{clk} < t_{pd,A} + t_{pd,B}$ -----slack-passing has taken place

T_{clk} /2 = maximum time that can be borrowed from previous stage