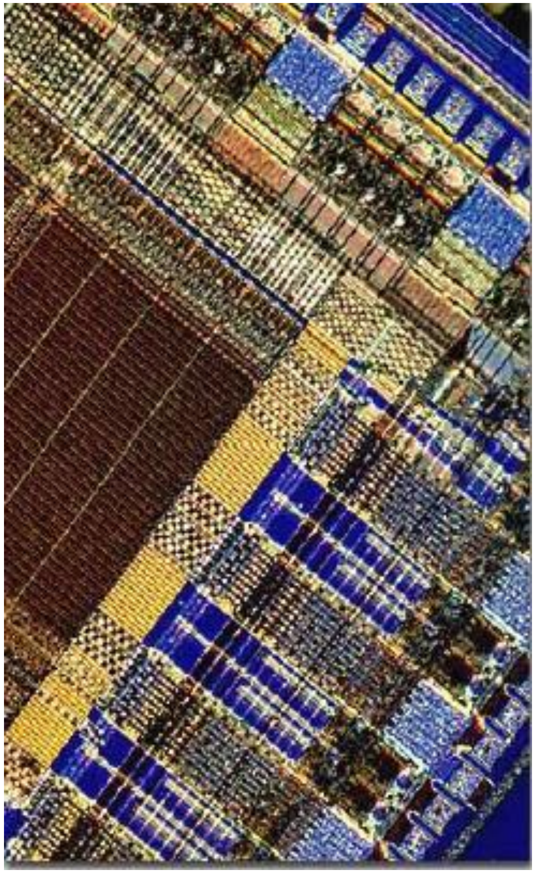
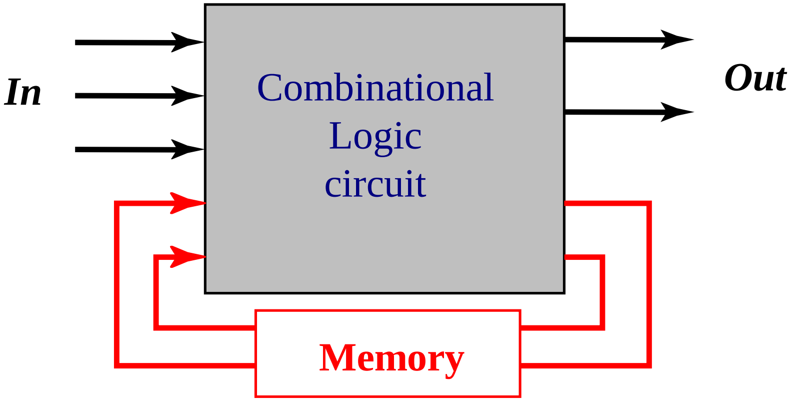
CMOS Digital Integrated Circuits



Lec 11

Sequential CMOS Logic Circuits

Sequential Logic



Sequential

The output is determined by

•Current inputs

•Previous inputs

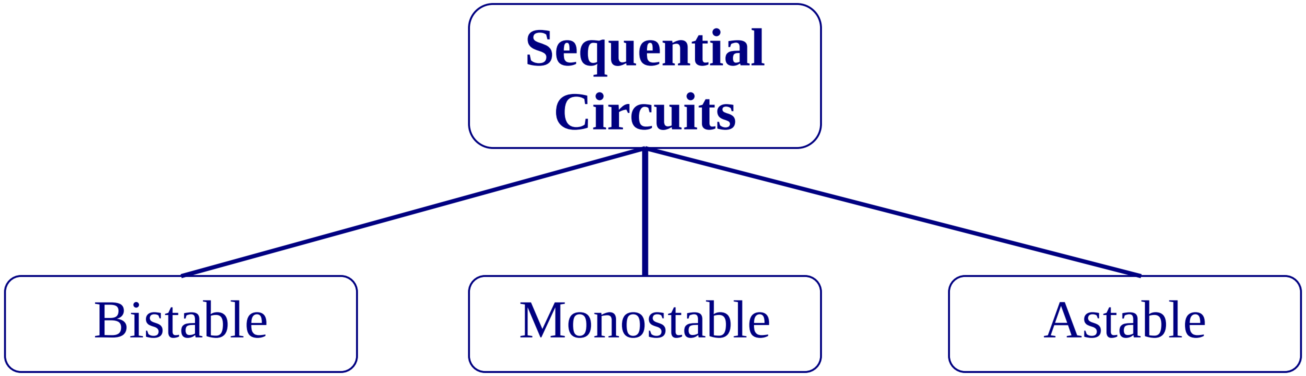
Output = *f(In, Previous In*)

- The regenerative behavior of sequential circuits is due to either a **direct** or an **indirect feedback** connection between the output and input

Critical Components of Sequential Circuits
Basic Regenerative Circuits

- **Categories of Basic Regenerative Circuits**

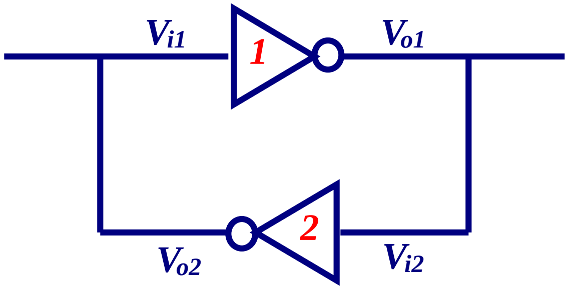
1. **Bistable Circuits**: Two stable states or operation modes, each of them can be attained under certain input and output conditions. **The most widely used** and the most important class which is used for the basic latch, flip-flop circuits, registers, and memory elements.
2. **Monostable Circuits:** One stable state or operation mode
3. **Astable Circuits:** No stable operating point or state which the circuit can preserve for a certain time period. The output oscillates without settling into a stable operating mode.

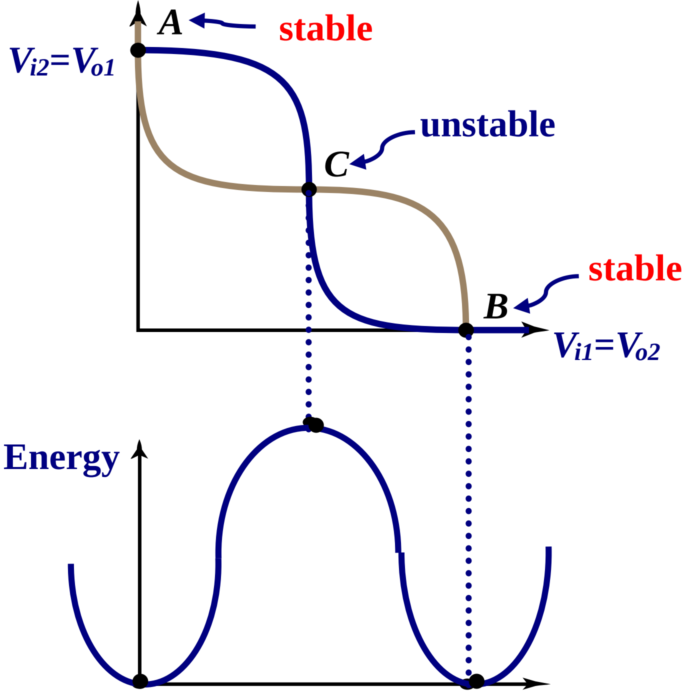


Behavior of Bistable Elements (*1/7*)

Two Identical Cross-Coupled Inverter Circuit

- **Voltage Transfer Curves**
  - The output voltage of inverter (1) is equal to the input voltage of inverter (2), and the output voltage of inverter (2) is equal to the input voltage of inverter (1).
  - ***A*** and ***B*** are stable points: If the circuit is initially operating at one of them, it will preserve this state. The gain is smaller than **unity**.

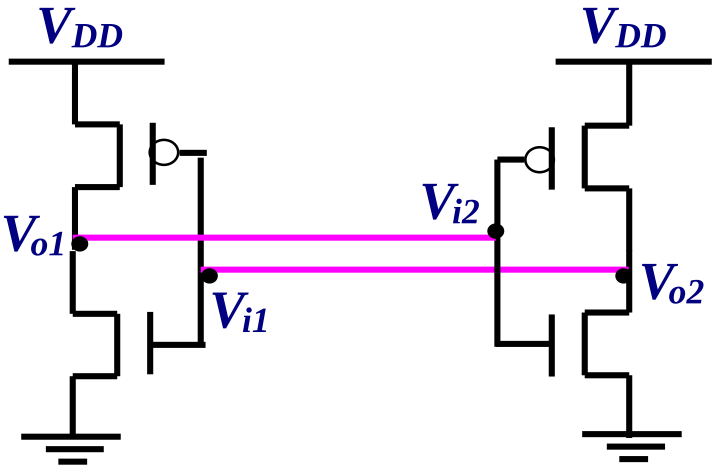


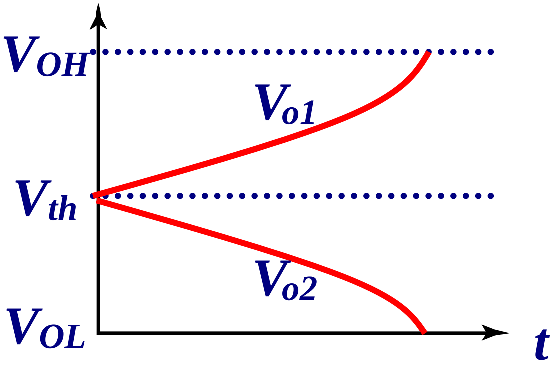


Behavior of Bistable Elements (2/7)

- » ***C*** is an unstable point: The voltage gains of both inverters are larger than unity. A small voltage perturbation at this operating point will be amplified ⇒ the operating point will move to one of the stable operating points, ***A*** or ***B***.

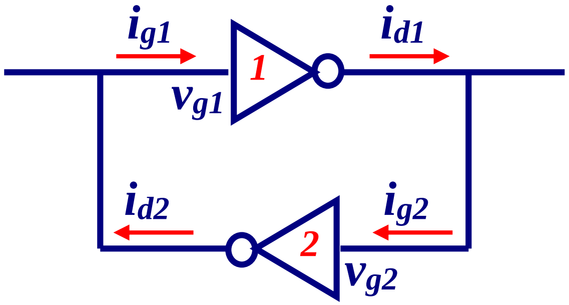
- **Energy Levels:**
  - » The potential energy is at its minimum at ***A*** and ***B***, since the voltage gains of both inverters are equal to zero.
  - » The potential energy is at its maximum at C, since the voltage gains of both inverters are maximum. (all four transistors are in saturation modes)





Behavior of Bistable Elements (3/7) Analysis of the Output Voltages

- Let the initially operating point is at $v_{o1}=v_{o2}=V_{th}$, and assume that the gate capacitance (C_g) of each inverter is much larger than the drain capacitance (C_d).
- The drain current of each inverter is equal to the gate current of the other inverter.



$$i_{g1} = i_{d2} = g_m v_{g2}$$
$$i_{g2} = i_{d1} = g_m v_{g1} \qquad \text{(Eq. A)}$$

g_m is the small-signal transconductance of the inverter.

- The gate voltages can be expressed by gate charges, q_1 and q_2

$$v_{g1} = q_1 / C_g$$
$$v_{g2} = q_2 / C_g \qquad \text{(Eq. B)}$$

- Also the small-signal gate currents can be expressed as

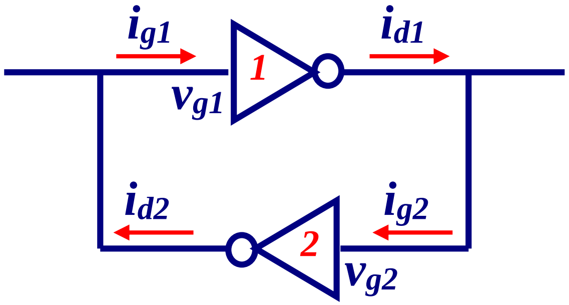
$$i_{g1} = C_g \cdot dv_{g1} /dt$$
$$i_{g2} = C_g \cdot dv_{g2} /dt \qquad \text{(Eq. C)}$$

Behavior of Bistable Elements (4/7)
Analysis of the Output Voltages

- Combine Eq. A and C, we have

$$g_m v_{g2} = C_g \frac{dv_{g1}}{dt}$$

$$g_m v_{g1} = C_g \frac{dv_{g2}}{dt}$$



- Replace the gate voltages by Eq. B, we obtain

$$\frac{g_m}{C_g} q_2 = \frac{dq_1}{dt}$$

$$\frac{g_m}{C_g} q_1 = \frac{dq_2}{dt}$$

- The above equations can be simplified to

$$\frac{g_m}{C_g} q_1 = \frac{C_g}{g_m}\frac{d^2 q_1}{dt^2} \Rightarrow \frac{d^2 q_1}{dt^2} = \left(\frac{g_m}{C_g}\right)^2 q_1 = \frac{1}{\tau_0^2} q_1;\ \tau_0 = \frac{C_g}{g_m}$$

Behavior of Bistable Elements (5/7)
Analysis of the Output Voltages

- Therefore,

$$q_1(t) = \frac{q_1(0) - \tau_0 q_1^{'}(0)}{2} e^{-\frac{t}{\tau_0}} + \frac{q_1(0) + \tau_0 q_1^{'}(0)}{2} e^{\frac{t}{\tau_0}}$$

where $\boldsymbol{q_1(0) = C_g \cdot v_{g1}(0)}$

- Replace the gate charge pf both inverters with the corresponding out-put voltages variables, we have

≈ 0 for $\boldsymbol{t} >> \tau_0$

$$v_{o2}(t) = \cancel{\frac{v_{o2}(0) - \tau_0 v_{o2}^{'}(0)}{2} e^{-\frac{t}{\tau_0}}} + \frac{v_{o2}(0) + \tau_0 v_{o2}^{'}(0)}{2} e^{\frac{t}{\tau_0}}$$

$$v_{o1}(t) = \cancel{\frac{v_{o1}(0) - \tau_0 v_{o1}^{'}(0)}{2} e^{-\frac{t}{\tau_0}}} + \frac{v_{o1}(0) + \tau_0 v_{o1}^{'}(0)}{2} e^{\frac{t}{\tau_0}}$$

≈ 0 for $\boldsymbol{t} >> \tau_0$

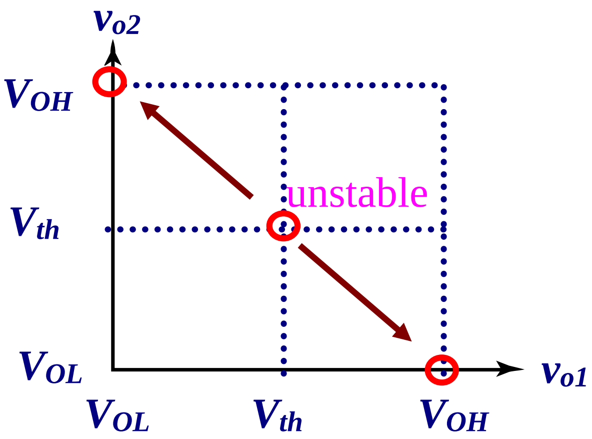
- For large values of ***t*,** the above equations can be approximated as

$$v_{o1}(t) \approx \frac{1}{2}\left(v_{o1}(0) + \tau_0 v_{o1}^{'}(0)\right) e^{\frac{t}{\tau_0}}$$

$$v_{o2}(t) \approx \frac{1}{2}\left(v_{o2}(0) + \tau_0 v_{o2}^{'}(0)\right) e^{\frac{t}{\tau_0}}$$

Behavior of Bistable Elements (6/7)
Analysis of the Output Voltages

- Depending on the polarity of the initial small perturbations $\boldsymbol{dv_{o1}}$**(0)** and $\boldsymbol{dv_{o1}}$(0), the $\boldsymbol{v_{o1}}$ and $\boldsymbol{v_{o2}}$ will diverge from their initial values of $\boldsymbol{V_{th}}$ to either $\boldsymbol{V_{OL}}$ and $\boldsymbol{V_{OH}}$.
- The polarity of $\boldsymbol{dv_{o1}}$ must always be **opposite** to that of $\boldsymbol{dv_{o2}}$, because of the **charge-conservation principle**. Therefore, $\boldsymbol{v_{o1}}$ and $\boldsymbol{v_{o2}}$ always diverge into opposite directions.



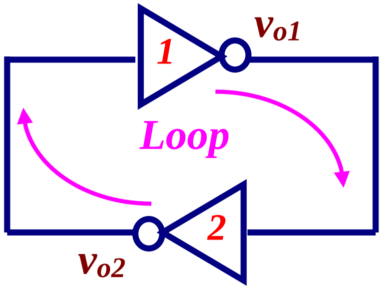
Phase-plane Representation

$$\boldsymbol{v_{o1}}: \boldsymbol{V_{th}} \rightarrow \boldsymbol{V_{OH}} \text{ or } \boldsymbol{V_{OL}}$$

$$\boldsymbol{v_{o2}}: \boldsymbol{V_{th}} \rightarrow \boldsymbol{V_{OL}} \text{ or } \boldsymbol{V_{OH}}$$

Behavior of Bistable Elements (7/7)
Analysis of the Output Voltages

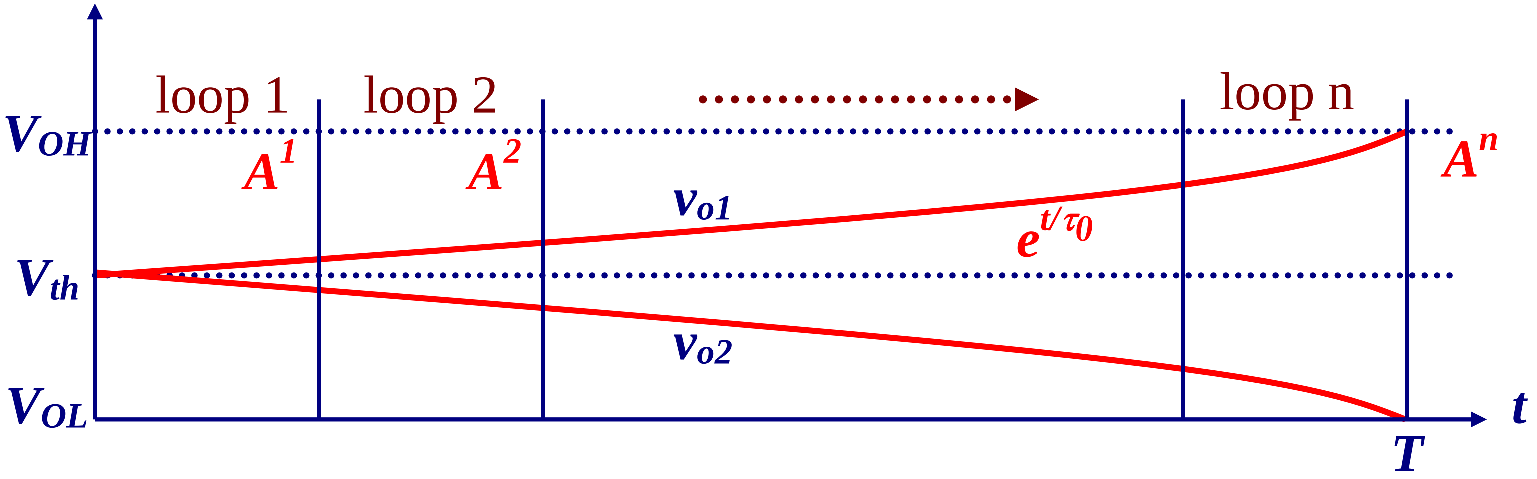
- As a bistable circuit settles from unstable operating point to its stable point, a signal travels around **2 INV loop n times.**



$$v_{o1}(t)/v_{o1}(0) \approx e^{t/\tau_0}$$

- If during interval $t = T$, the signal travels around the loop n times

$$A^n \approx e^{T/\tau_0}$$



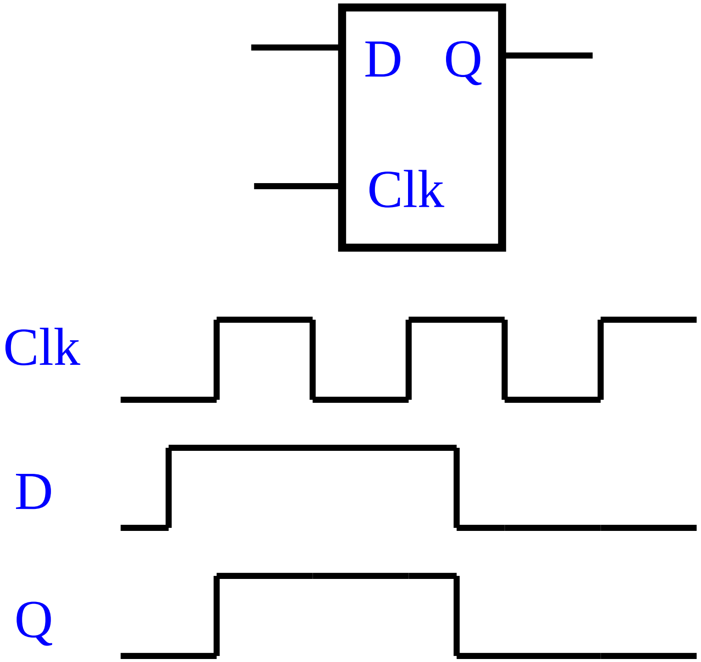
Naming Conventions

- A latch is level sensitive
- A register is edge-triggered
- There are many different naming conventions
  - For instance, many books call edge-triggered elements flip-flops

Latch versus Register

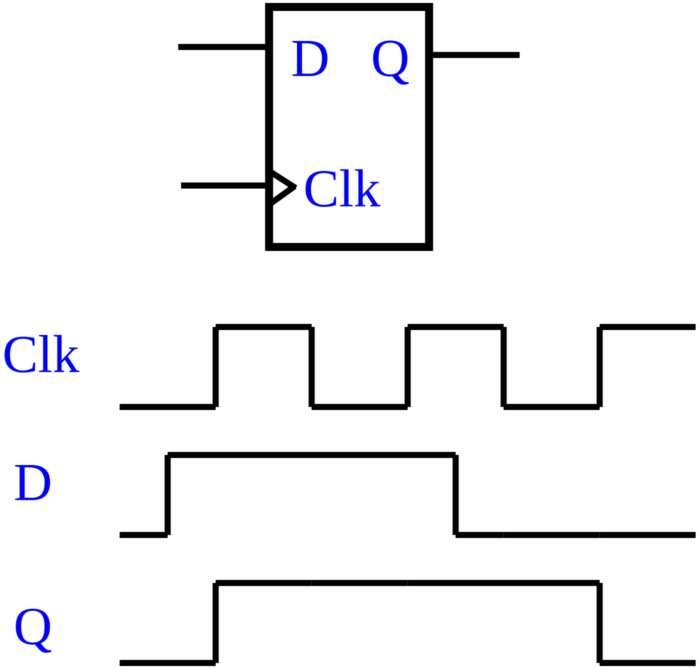
- Latch

  stores data when clock is low



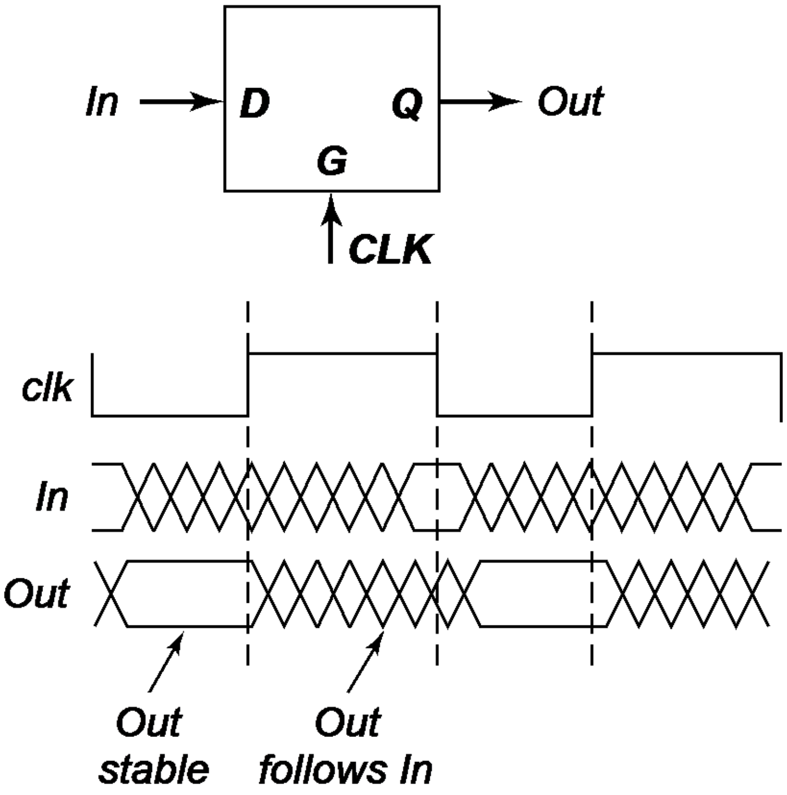
- Register

  stores data when clock rises

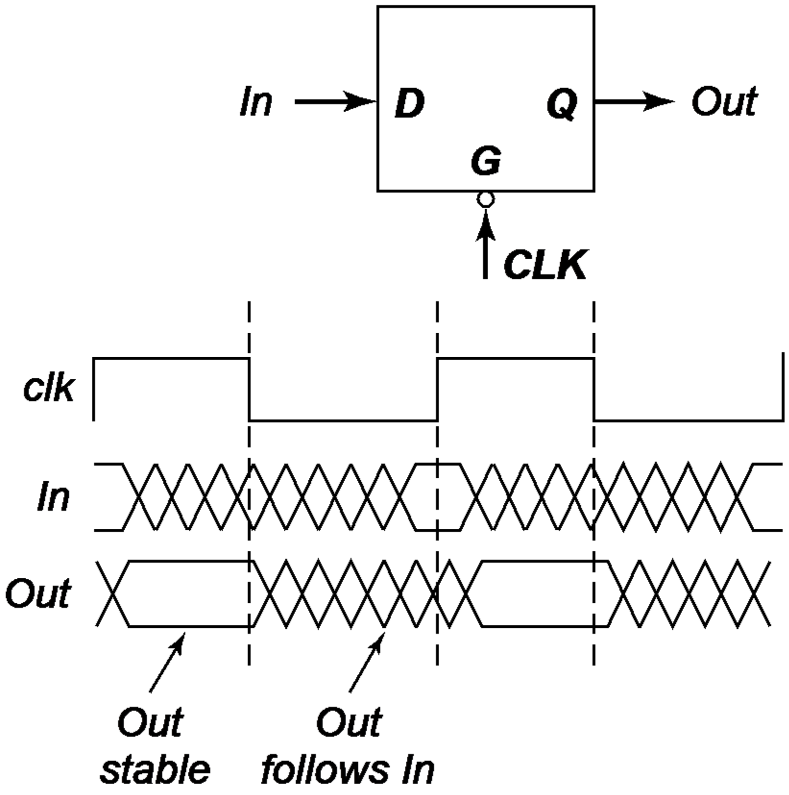


Latches

Positive Latch

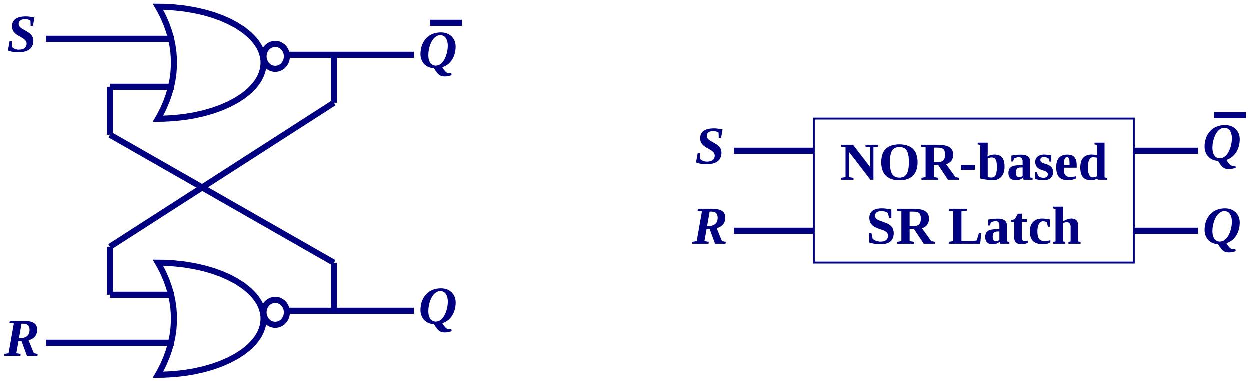


Negative Latch



SR Latch Circuit

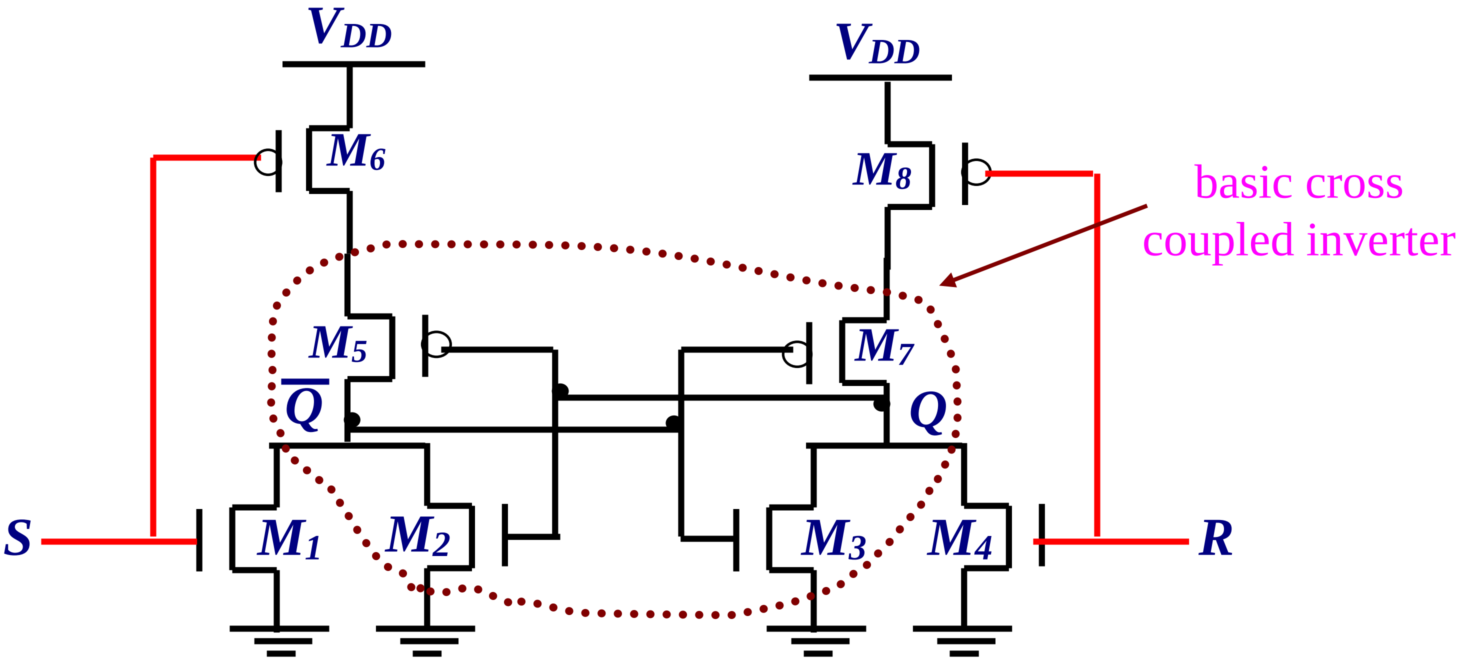
- The two cross-coupled inverters can perform a simple memory function of ***holding*** its state. However, the two-inverter circuit alone has no provision for allowing its state to be changed externally from one stable operating point to other.
- In order to allow such a change of state, we need to ***add simple switches*** which can be used to force or trigger the circuit from one operating point to the other.



Schematic Diagram of SR Latch

SR Latch Circuit (Cont.)

- The below circuit shows the simple CMOS SR latch which consists of two triggering inputs, ***S*** (***set***) and ***R*** (***reset***).
- The SR Latch consists of two CMOS NOR2 gates. One of the input terminals of each NOR gate is used to cross-couple to the output of the other NOR gate. The second input enables triggering of the circuit.



SR Latch Circuit
Truth Table

- **Set:** $S=1$, $R=0 \Rightarrow Q_{n+1}=1$, $\overline{Q_{n+1}}=0$. The SR latch will be ***set*** regardless of its previous state.
- **Reset:** $S=0$, $R=1 \Rightarrow Q_{n+1}=0$, $\overline{Q_{n+1}}=1$. The SR latch will be ***reset*** regardless of its previous state.
- **Hold:** $S=0$, $R=0 \Rightarrow Q_{n+1}=Q_n$, $\overline{Q_{n+1}}=\overline{Q_n}$. The previous states will be ***held***.
- **Not Allow:** $S=0$, $R=0 \Rightarrow Q_{n+1}=0$, $\overline{Q_{n+1}}=0$

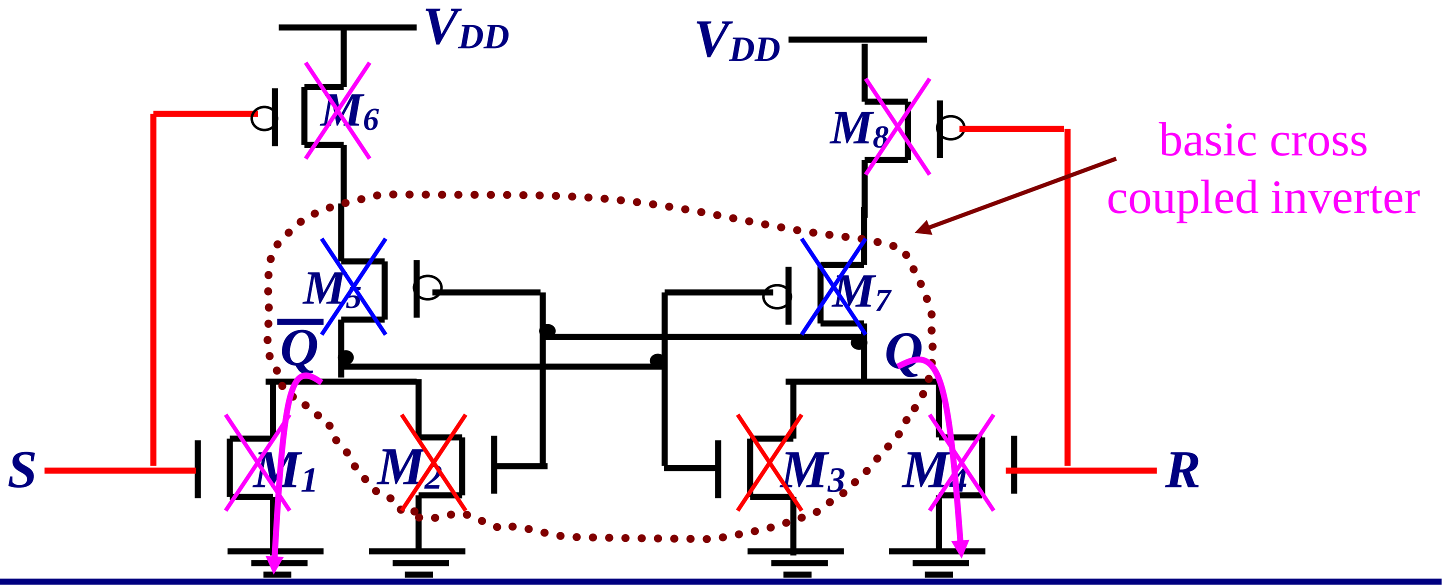
S	R	Q_{n+1}	$\overline{Q_{n+1}}$	**Operation**
0	0	Q_n	$\overline{Q_n}$	***Hold***
1	0	1	0	***Set***
0	1	0	1	***Reset***
1	1	0	0	***Not Allowed***

active high

Truth Table of NOR-based (*active high inputs*) SR latch

SR Latch Circuit
Operation Modes of the Transistors

S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation	
				NMOS	PMOS
V_{OH}	V_{OL}	V_{OH}②’	V_{OL}①’	$M_1$①, $M_2$③ on; $M_3$②, $M_4$① off	$M_7$②, $M_8$① on; $M_5$③, $M_6$① off
V_{OL}	V_{OH}	V_{OL}①’	V_{OH}②’	$M_1$①, $M_2$② off; $M_3$③, $M_4$① on	$M_7$③, $M_8$① off; $M_5$②, $M_6$① on
V_{OL}	V_{OL}	V_{OH}	V_{OL}	M_1, M_4 off; M_2, on	M_6, M_8 on; M_7, on
V_{OL}	V_{OL}	V_{OL}	V_{OH}	M_1, M_4 off; M_3, on	M_6, M_8 on; M_5, on



basic cross coupled inverter

SR Latch Circuit
Transient Analysis

- For transient analysis, we have to consider an event which results in a state change, **reset ⇒ set**, or **set ⇒ reset**
- In either case, we note that both of the output nodes undergo simultaneous voltage transitions. One is from logic-low to logic-high, and the other is from logic-high to logic-low.
- The exact transient analysis need to solve two coupled differential equations.
- For simplicity, we can assume that the ***two events take place in sequence rather than simultaneously***. **(overestimation)**

Switching Time Calculation

- The total lumped capacitance at each output node can be approximated as

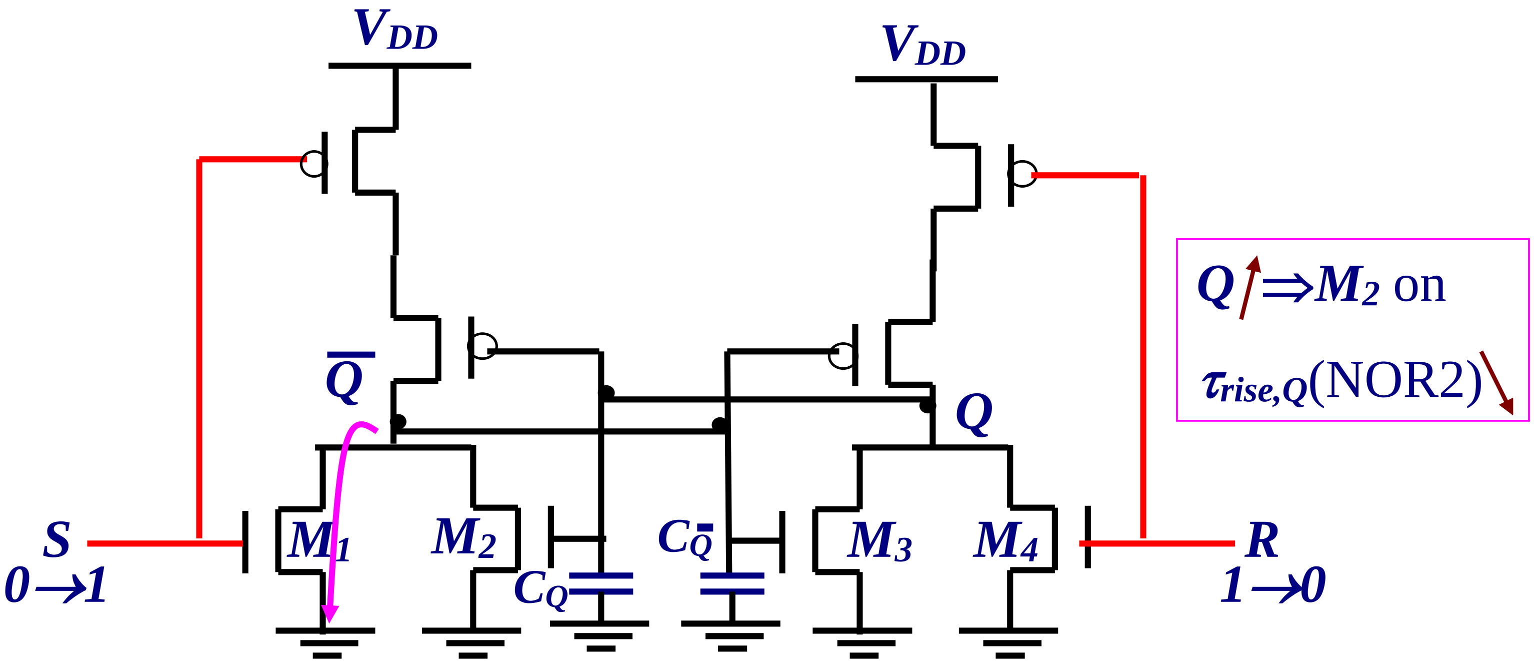
$$C_Q = C_{gb,2}+C_{gb,5}+C_{db,3}+C_{db,4}+C_{db,7}+C_{sb,7}+C_{db,8}$$

$$C_{\bar{Q}} = C_{gb,3}+C_{gb,7}+C_{db,1}+C_{db,2}+C_{db,5}+C_{sb,5}+C_{db,6}$$

SR Latch Circuit
Transient Analysis (Cont.)

- Assuming that the latch is initially reset and that a set operation is being performed, the rise time associated with node ***Q*** can be estimated as

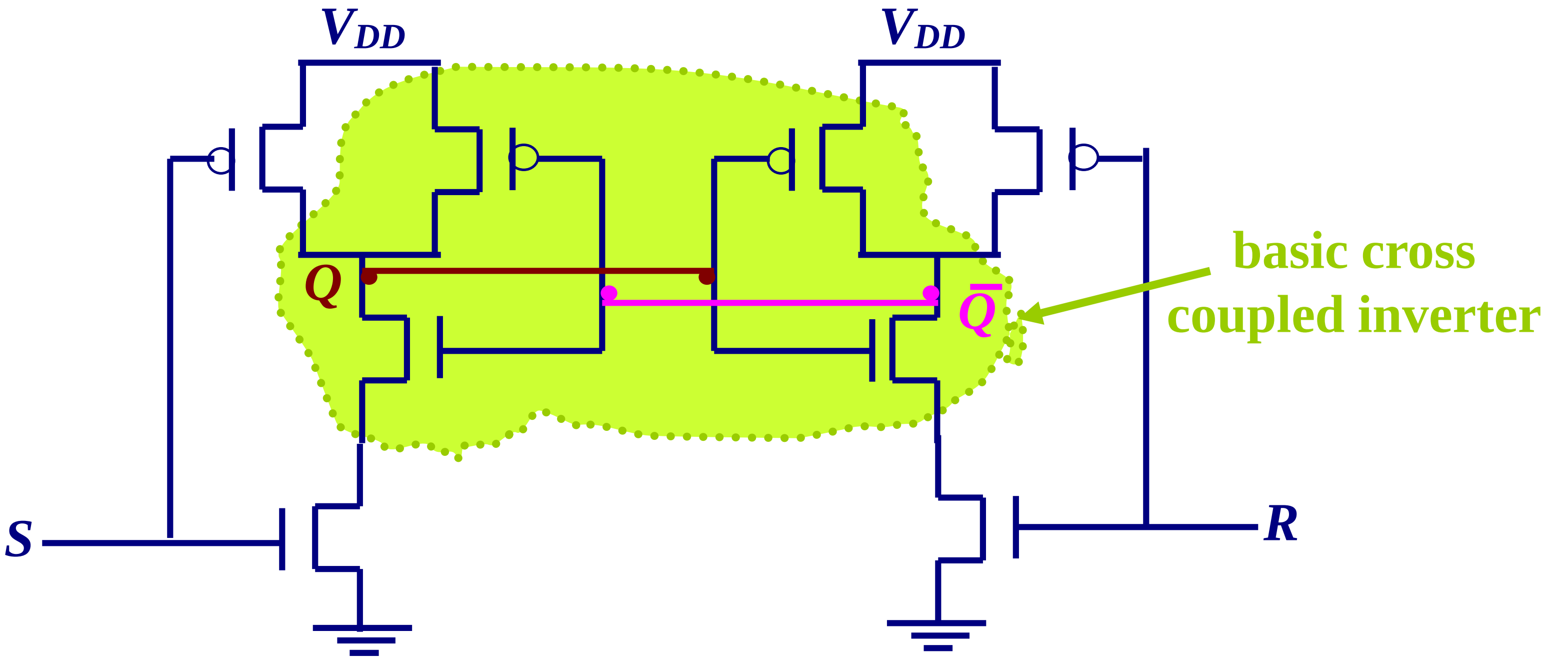
$$\tau_{rise,Q}(SR\text{-}latch) = \tau_{rise,Q}(\mathrm{NOR2}) + \tau_{fall,\bar{Q}}(\mathrm{NOR2})$$



$Q\uparrow \Rightarrow M_2$ on

$\tau_{rise,Q}(\mathrm{NOR2})\searrow$

SR Latch Circuit
NAND-based (active low signals)



basic cross coupled inverter

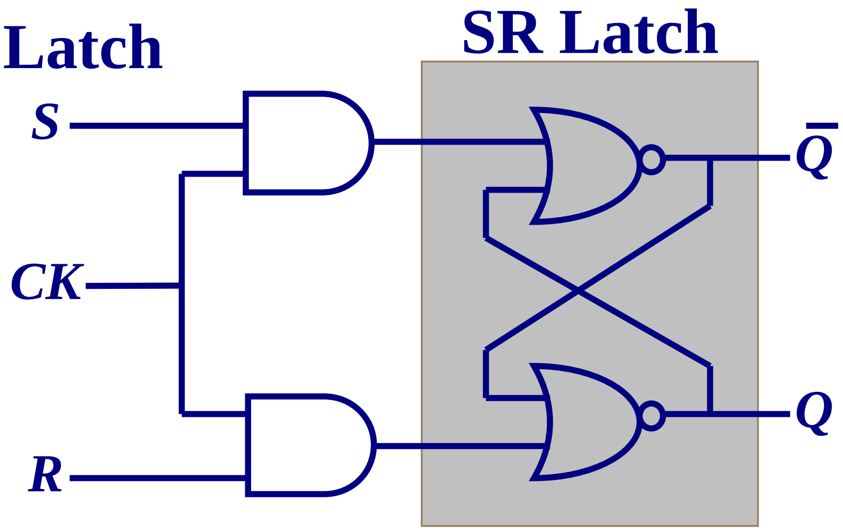
S	R	Q_{n+1}	$\overline{Q_{n+1}}$	**Operation**
0	0	1	1	***Not Allowed***
0	1	1	0	***Set***
1	0	0	1	***Reset***
1	1	Q_n	$\overline{Q_n}$	***Hold***

active low

Clocked Latch and Flip-Flop Circuits

The previous SR latch circuits are ***asynchronous*** sequential circuits. The ***synchronization*** can be introduced through **clock CK**, which the outputs will respond to the input levels only during the active period of a clock pulse.

Clocked SR Latch



- When ***$CK=0$, S, R*** have no influence of ***Q, $\overline{Q}$*** ⇒ ***Hold***

Set State: $CK=1,\ S=1,\ R=0 \Rightarrow Q_{n+1}=1,\ \overline{Q_{n+1}}=0$

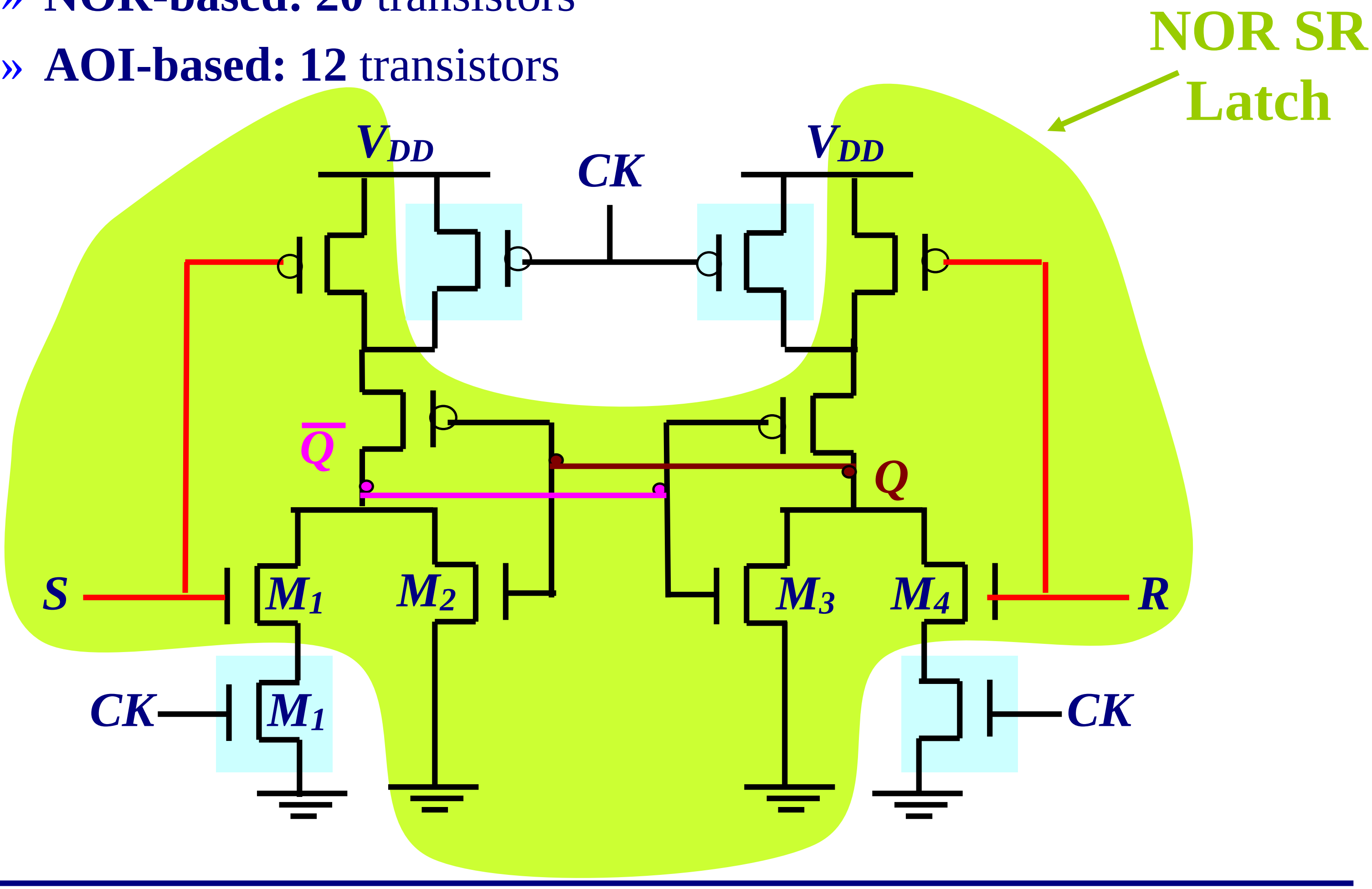
Reset State: $CK=1,\ S=0,\ R=1 \Rightarrow Q_{n+1}=0,\ \overline{Q_{n+1}}=1$

Not Allowed: $CK=1,\ S=1,\ R=1$

Active "High"

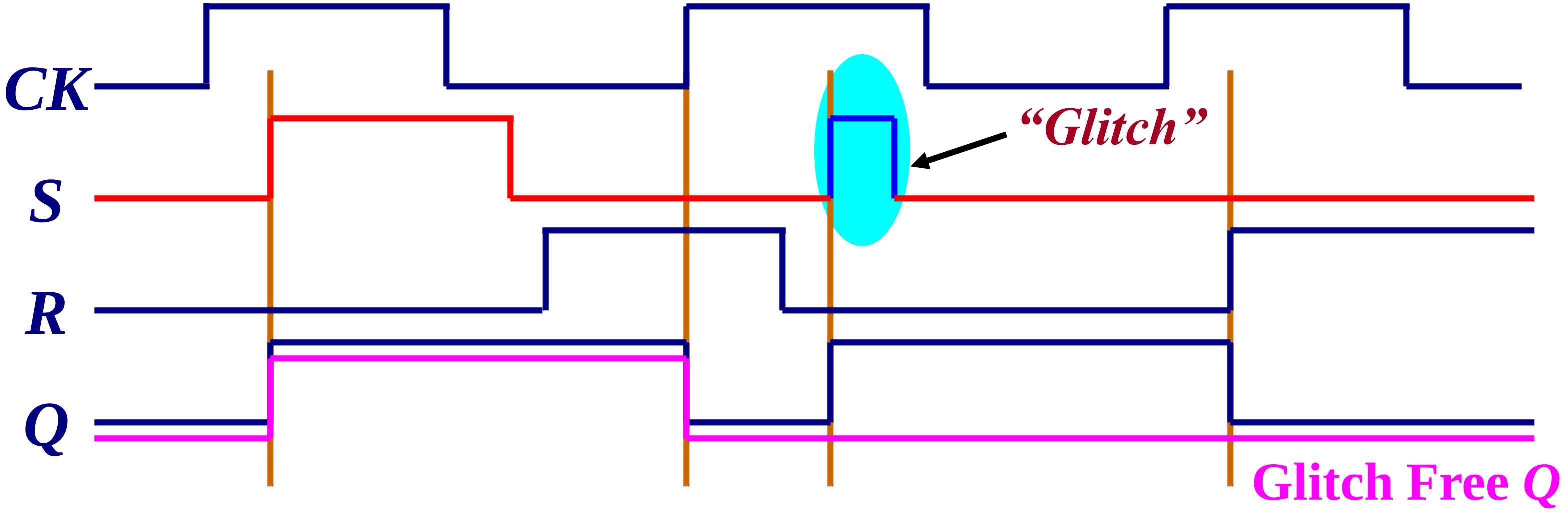
AOI-based Implementation of Clocked NOR-based SR Latch

- The AOI-based implementation need a very small transistor count, compared with the circuit consisting of two AND2 and two NOR2 gates
  - » **NOR-based: 20** transistors
  - » **AOI-based: 12** transistors



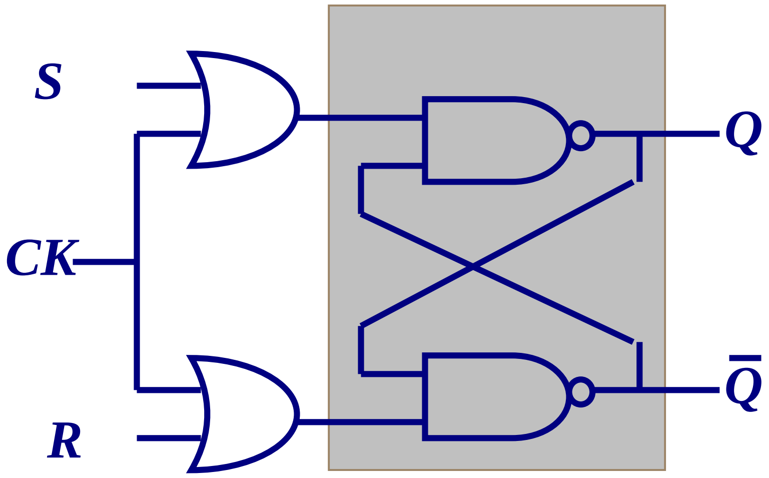
Operation of Clocked SR Latch

Operation	*CK*	*S*	*R*	Q_{n+1}	$\overline{Q_{n+1}}$
Hold	0	X	X	Q_n	$\overline{Q_n}$
Set	1	1	0	1	0
Reset	1	0	1	0	1
Not Allow	1	1	1	1	1



- When "Glitch" ON ***S*** (or ***R***) occurs during ***CK*** = ***1***, ***Q*** is set (or reset).
- **Level Sensitive:** When ***CK*** = ***1***, any changes in ***S***, ***R*** will effect ***Q***.

Clocked NAND-based SR Latch

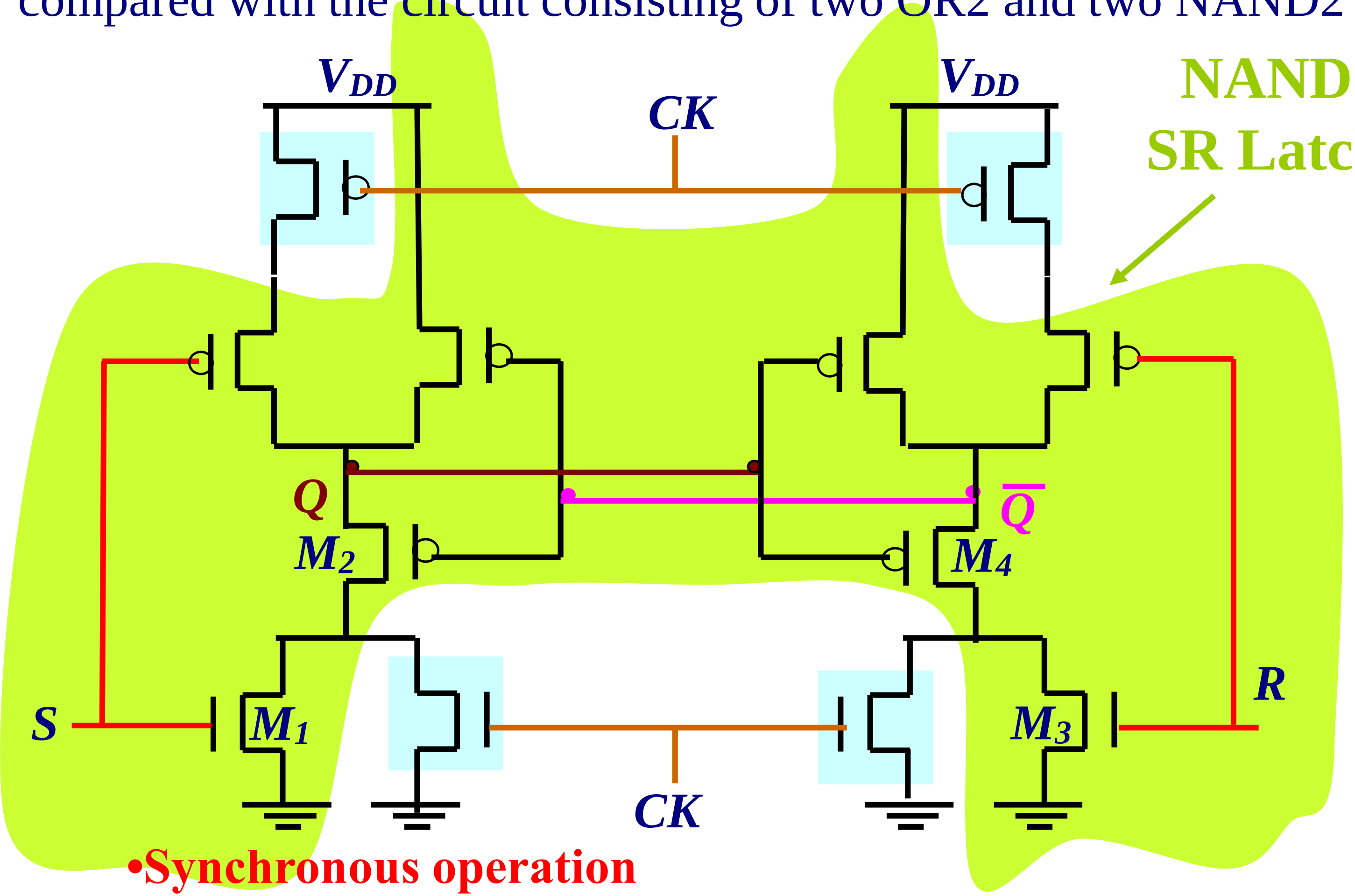


- When ***CK*** = ***1***, ***S*** and ***R*** have no influence of ***Q*** and $\overline{Q}$ ⇒ ***Hold***

Operation	*CK*	*S*	*R*	Q_{n+1}	$\overline{Q_{n+1}}$
Hold	1	X	X	Q_n	$\overline{Q_n}$
Set	0	0	1	1	0
Reset	0	1	0	0	1
Not Allow	0	0	0	0	0

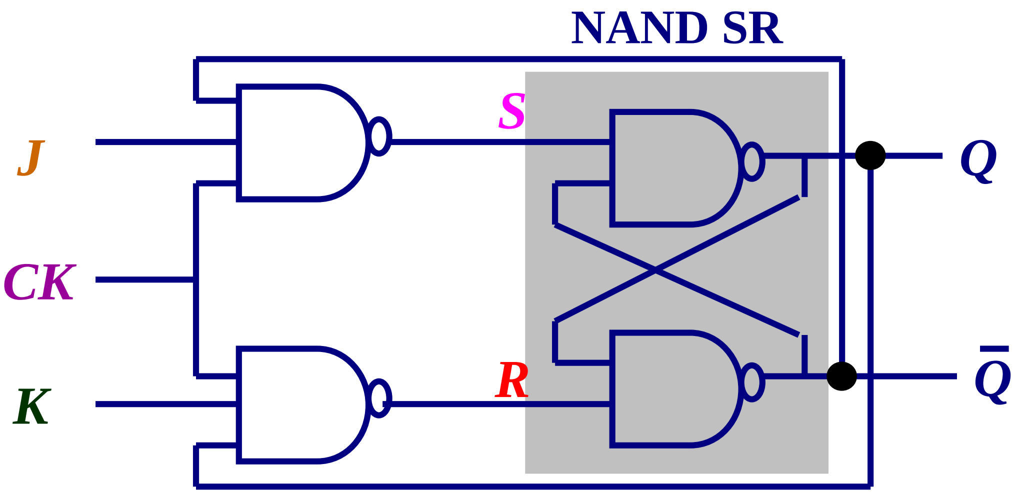
OAI-based Implementation of Clocked NAND-based SR Latch

- The OAI-based implementation need a very small transistor count, compared with the circuit consisting of two OR2 and two NAND2 gates



- **Synchronous operation**
- **Level sensitive** (any changes in ***S*** and ***R*** as ***CK***=1 will be reflected onto outputs)
- **Not allowed input sequence**

Clocked JK Latch



CK = 0 ⇒ hold

CK = 1 ⇒ active

No not allowed combination

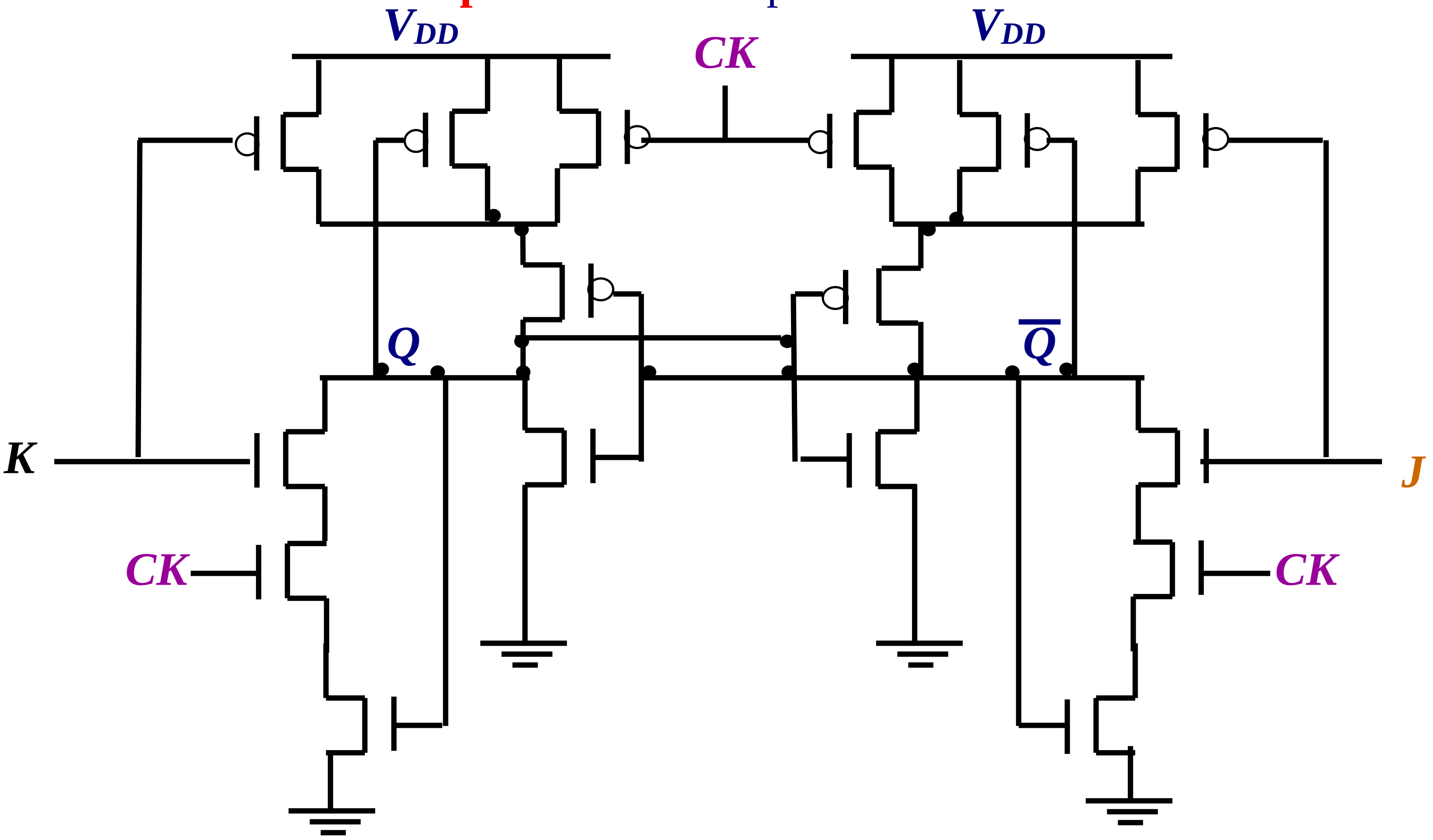
CK = 1

J	K	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation
0	0	0	1	1	1	0	1	*Hold*
0	0	1	0	1	1	1	0	*Hold*
0	1	0	1	1	1	0	1	*Reset*
0	1	1	0	1	0	0	1	*Reset*
1	0	0	1	0	1	1	0	*Set*
1	0	1	0	1	1	1	0	*Set*
1	1	0	1	0	1	1	0	*Toggle*
1	1	1	0	1	0	0	1	*Toggle*

OSC

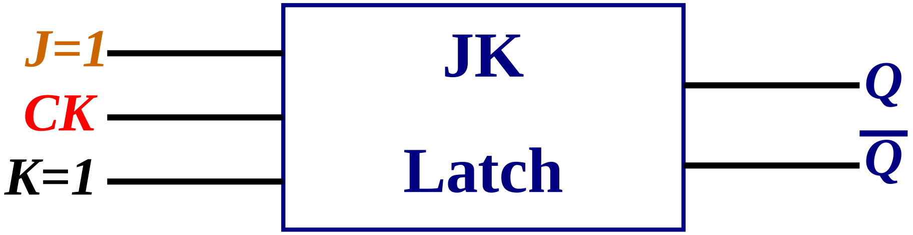
AOI-based Implementation of NOR-based Clocked JK Latch

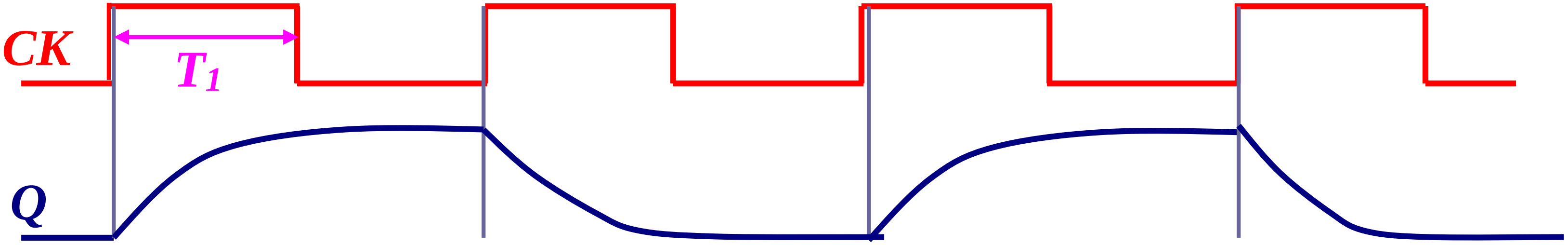
- The AOI-based implementation has a **very small transistor count**, and a **more compact** circuit compared to all-NAND realization.



JK Toggle Switch

$J = K = 1$





- Iff $\tau_{JKP} > T_1$ (awkward to implement)

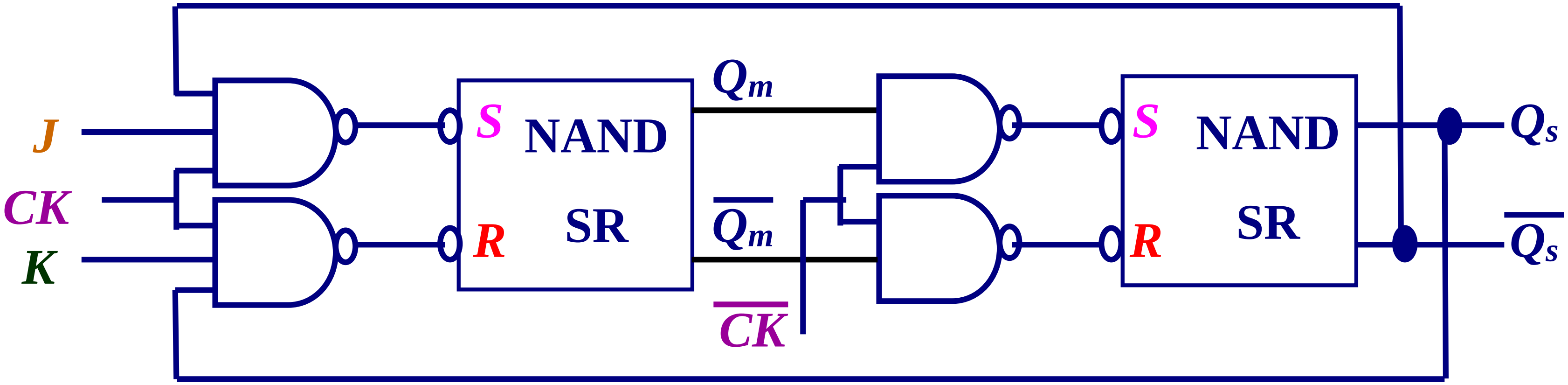
  Output Q changes only once per clock period

  » No not allowed input

  » Timing issues

  » Level sensitive

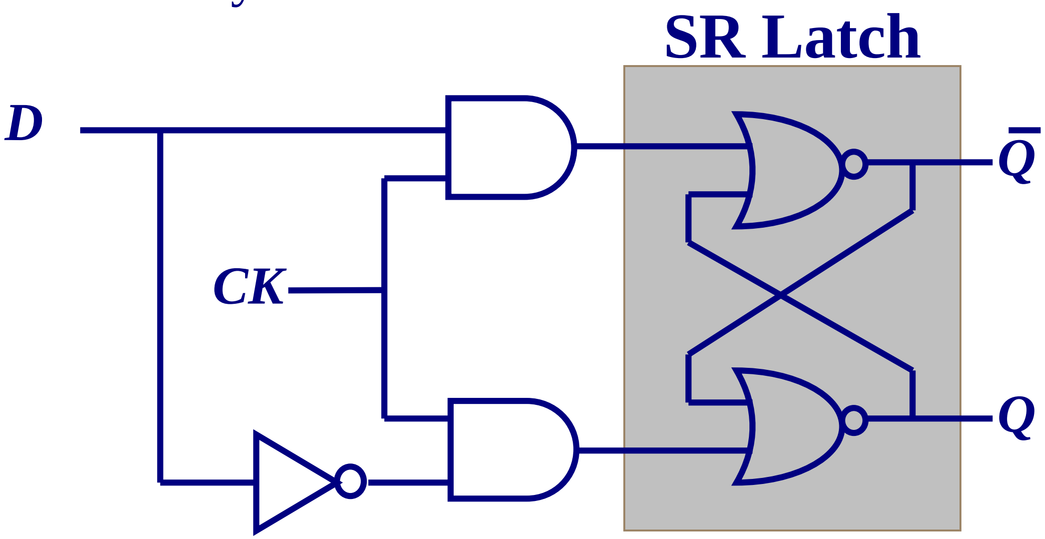
Master-Slave Flip-Flop



- Two cascaded latches operating on opposite clock phases insures that the flip-flop is ***never transparent***; i.e., a change occurring in the primary inputs is never reflected directly to the outputs.
- Eliminates oscillations when ***J* = *K* =*1*.**
- Still level sensitive.
- Number of transistors:
  - » **NAND-based: 36**
  - » **AOI-based: 28**

D-Latch

- D-latch is obtained by modifying the clocked NOR-based SR latch circuit. The circuit has a single input ***D*** which is connected to ***S*** input, and ***D*** is also inverted and connected to ***R*** input.
- The applications of D-latch are primarily for temporary storage of data or as a delay element.

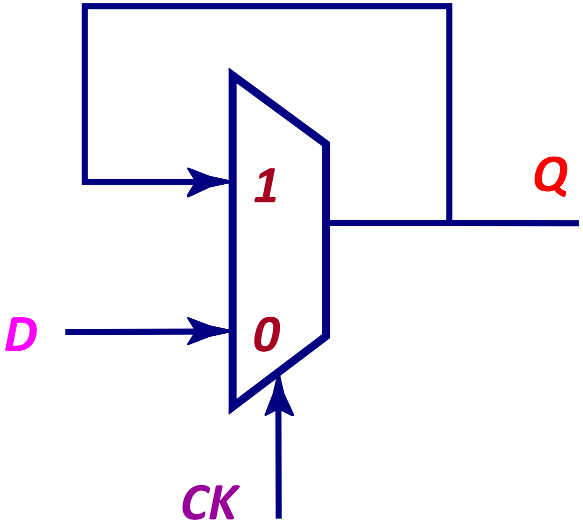


If $CK=1 \Rightarrow Q_{n+1} = D$

If $CK=0 \Rightarrow Q_{n+1} = Q_n$

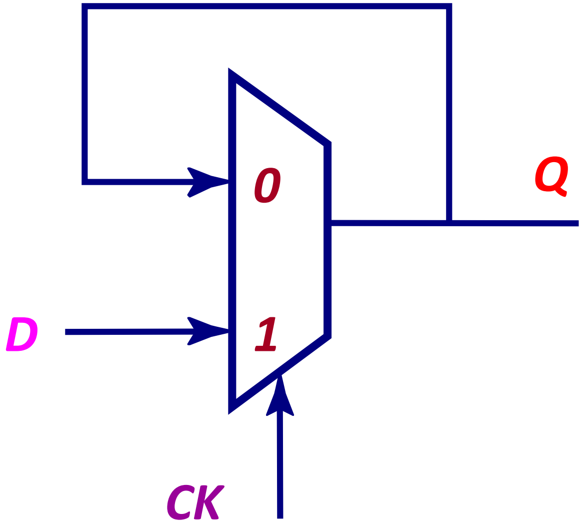
D-Latch (Cont.)

- D-latch is a mux-based latch which can be represented as



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

Negative latch
(transparent when *CK*= 0)

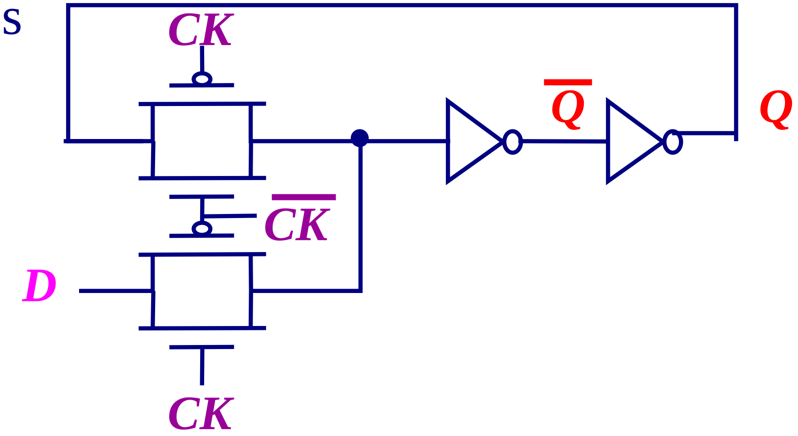


$$Q = \overline{CK} \cdot Q + CK \cdot In$$

Positive latch
(transparent when *CK*= 1)

D-Latch
Implementation with Transmission Gates

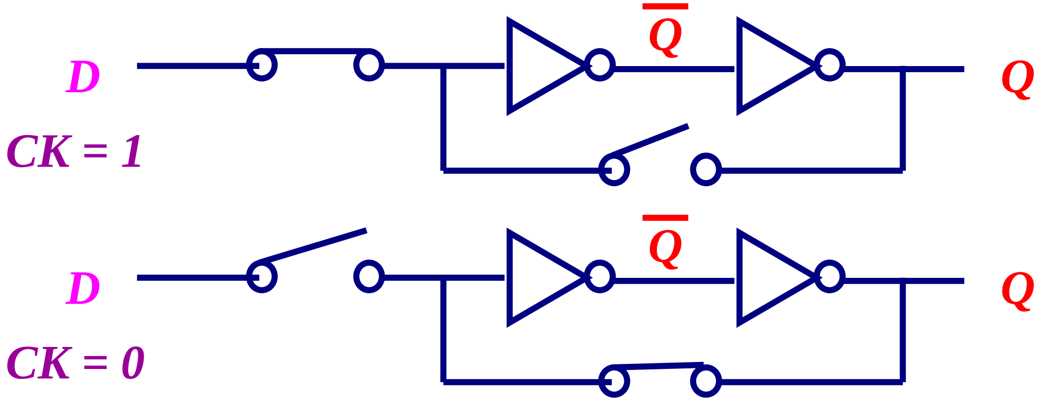
- Transmission gate **D-latch**: Use switch-like properties of transmission gates



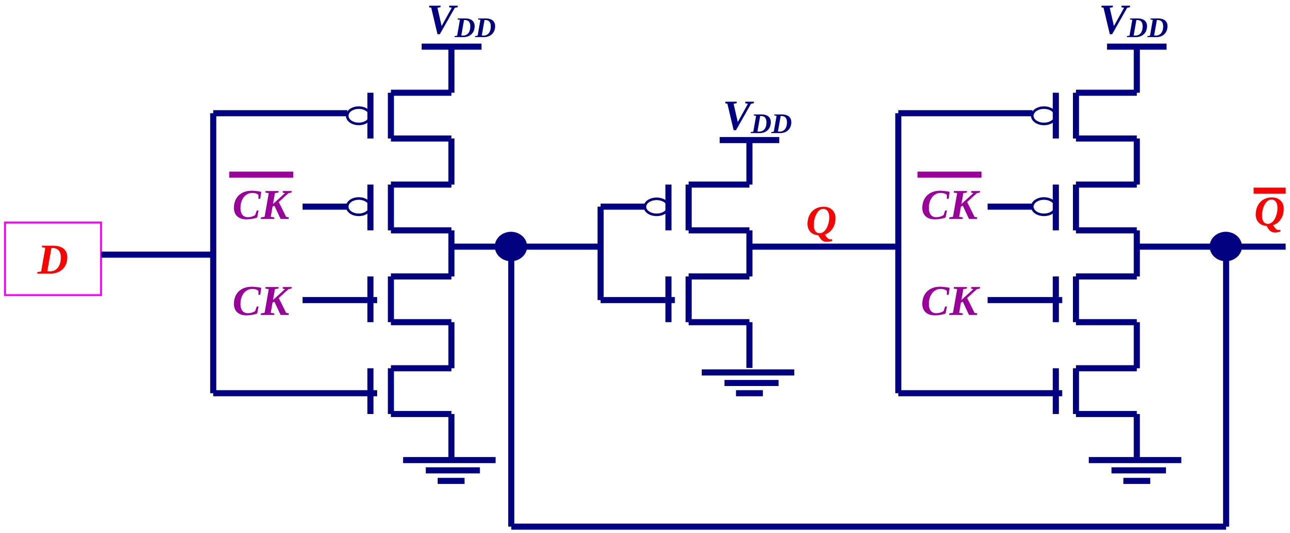
- Operation: For ***CK*** = 1, $\boldsymbol{\overline{Q_{n+1}}=\overline{D}}$ and $\boldsymbol{Q_{n+1}=D}$. A bit is loaded. For ***CK*** = 0, $\boldsymbol{\overline{Q_{n+1}}=\overline{Q_n}}$ and $\boldsymbol{Q_{n+1}=Q_n}$. Thus, a bit is stored. Note that Propagation delay to ***Q*** is less than delay to $\boldsymbol{\overline{Q}}$. What about changes in ***D*** relative to changes in ***CK***?

  Setup time and ***Hold time*** relative to ***CK***: 1→0
- Device counts for TG-based reduced from AOI/OAI
  - » **AOI-based: 14**
  - » **TG-based: 8 (plus 2 to invert clock)**

D-Latch
Implementation with Three-State

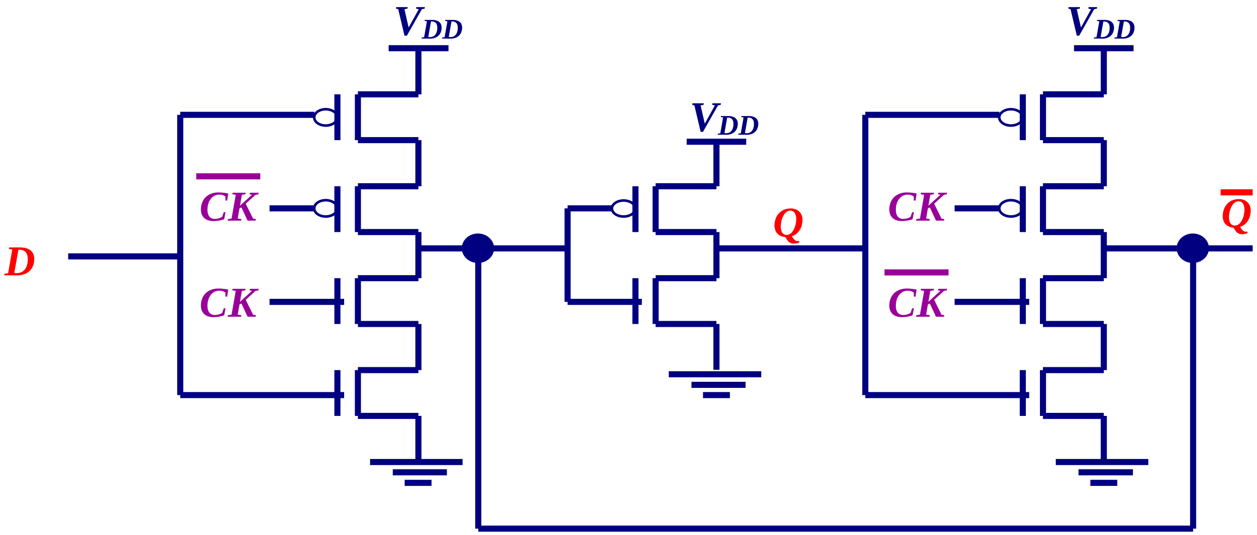


- Similar to the TG-based implementation, except as if connection between n and pFETs in a driving inverter and input side of a driven transmission gate is served. Require addition of inverter at input first.

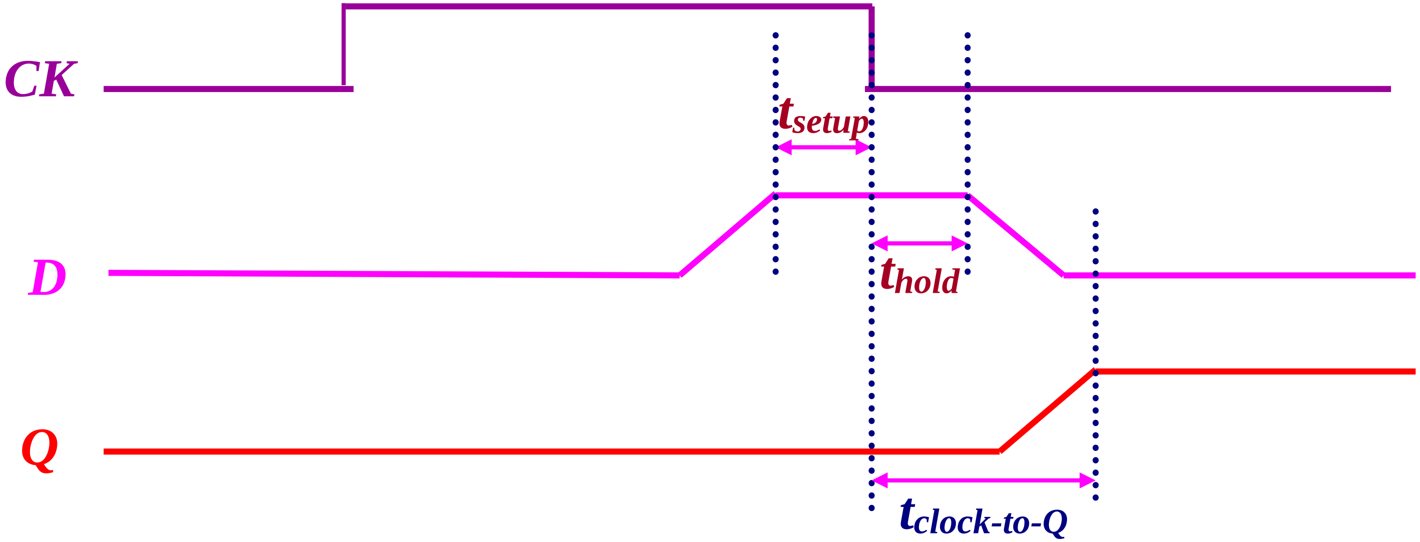


D-Latch
Implementation with Three-State (Cont.)

- The first three-state inverter acts as the **input switch**. Accept the input signal when ***CK*** is high, the second three-state inverter is at its high impedance state, and ***Q* = *D***.
- The first three-state inverter is **inactive** when the ***CK*** goes low, and the second three-state inverter completes the two-inverter loop, which preserves its state ($\boldsymbol{Q_{n+1} = Q_n}$)



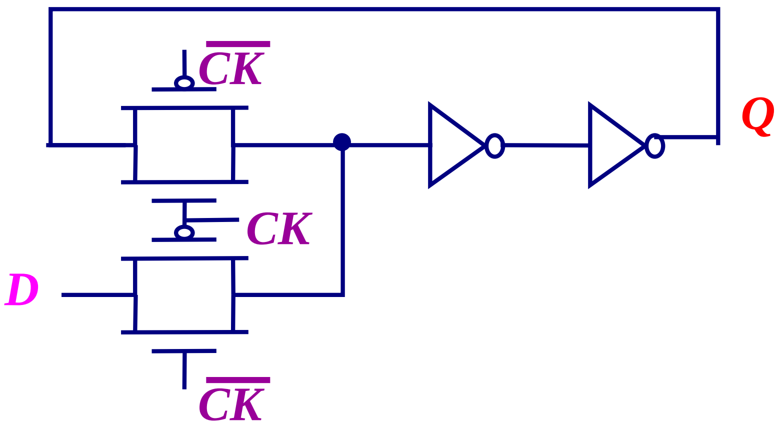
D-Latch
Setup Time and Hold Time

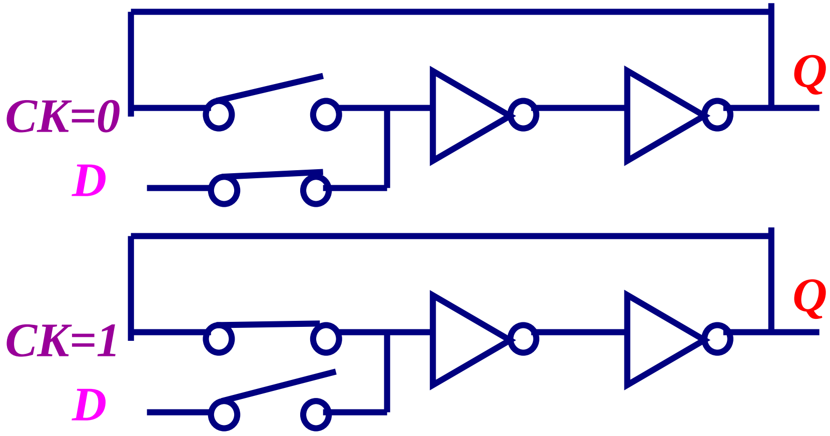


- ***T_{setup}***: time before the negative-CK edge the ***D-input*** has to be stable
  - » The setup time is the delay between the data input of the register and the storage element. As the data takes a finite time to travel to the storage point, the clock cannot be changed until the correct data value appears.
- ***T_{hold}***: time after the negative-CK edge ***D-input*** has to remain stable
  - » The hold time relates to the delay between the clock input to the register and the storage element. That is, the data has to be held for this period while the clock travels to the point of storage.
- ***$T_{clock\text{-}to\text{-}Q}$***: Delay from the negative-CK edge to new value of Q output

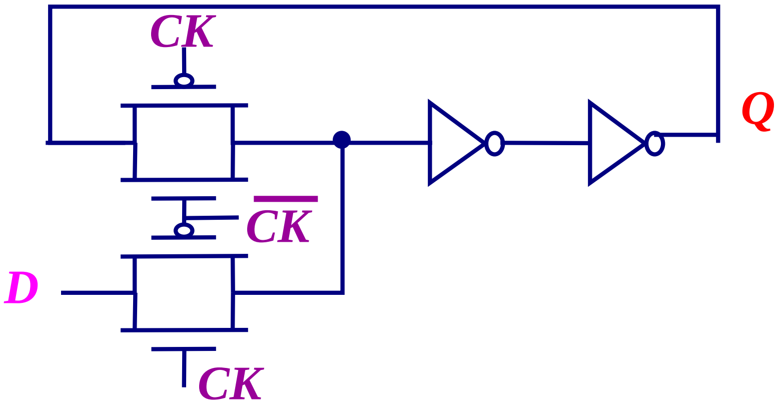
Edge Triggered Master-Slave Operation

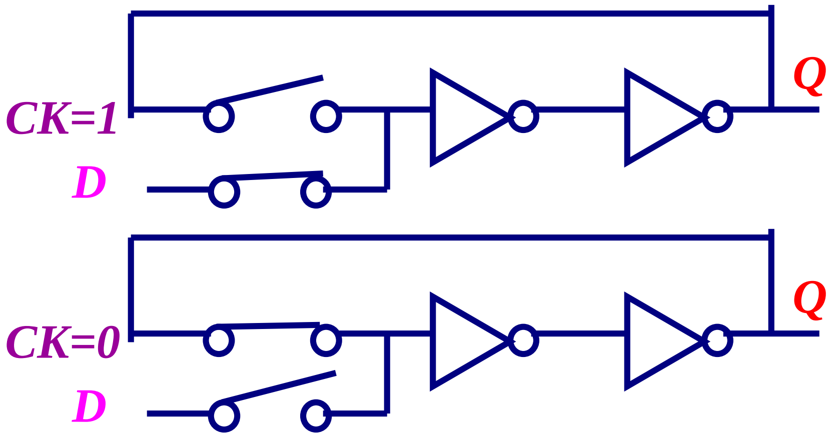
- Negative D-Latch



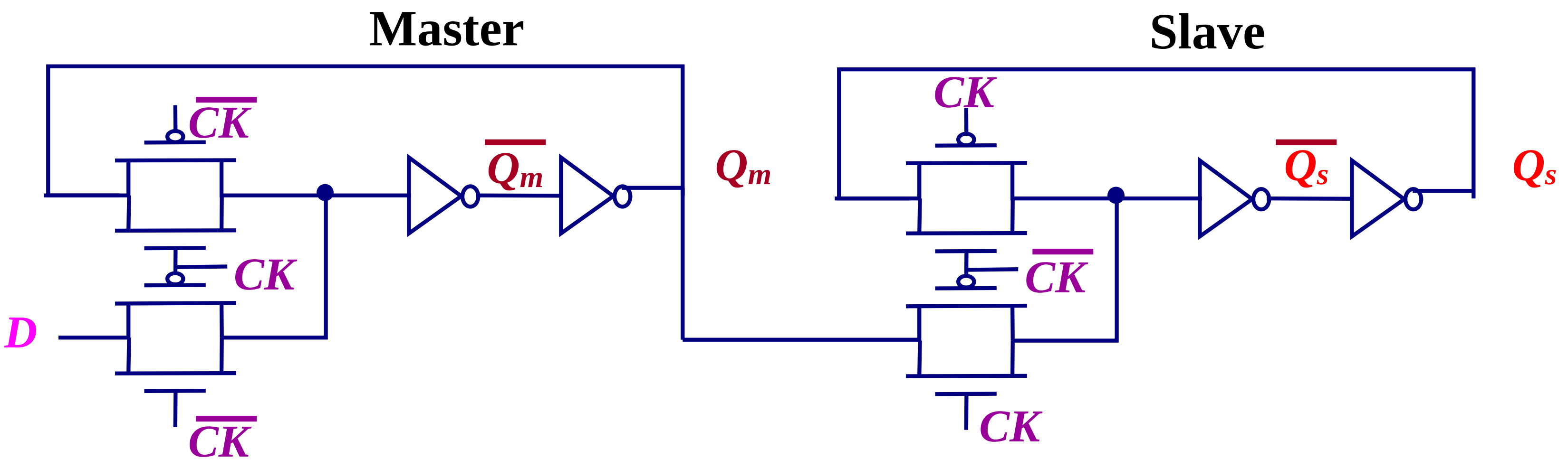


- Positive D-Latch

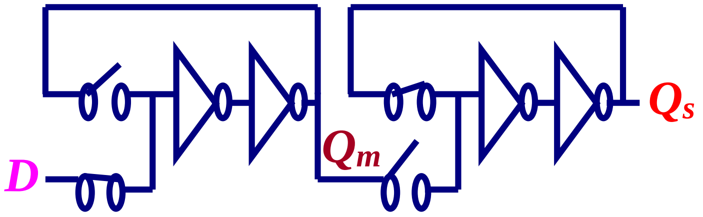




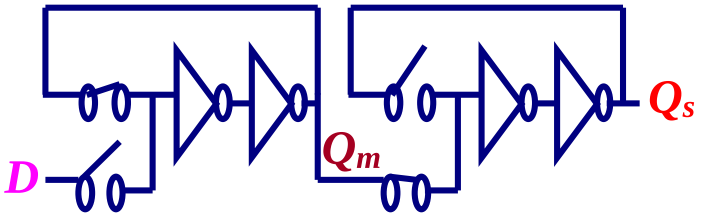
Positive Edge Triggered Master-Slave Flip-Flop



For CK=0



For CK=1



1. CK=0: Master Q_m tracks current D; Slave Q_s=previous D sample
2. CK=0→1: Master stores Q_m = D(new D sample).
3. CK=1:Master passes Q_m = D to Slave output Q_s
4. CK=1→0: Slave locks in new D, and Master Q_m begins tracking D.

DFF Transient Response

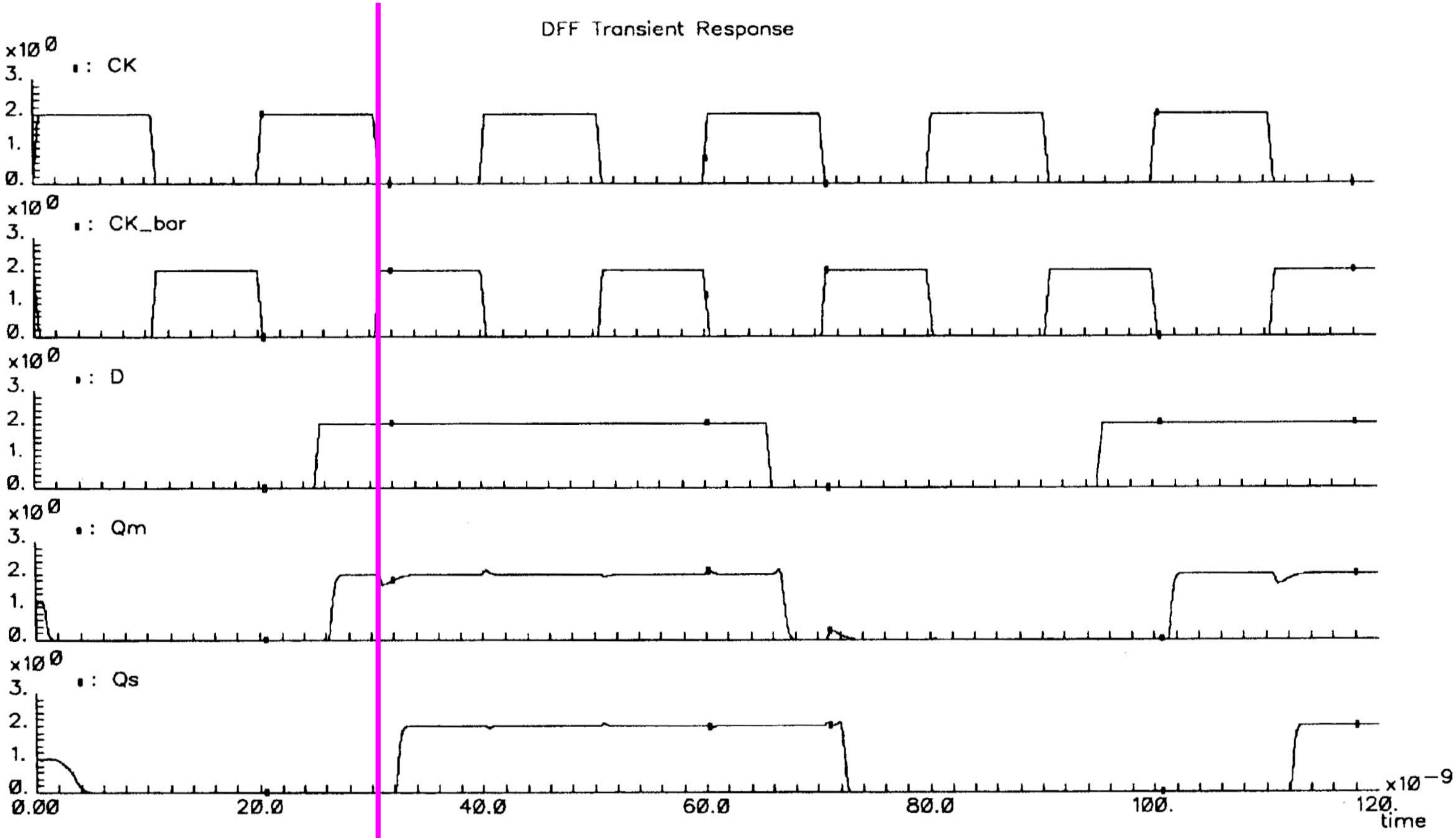
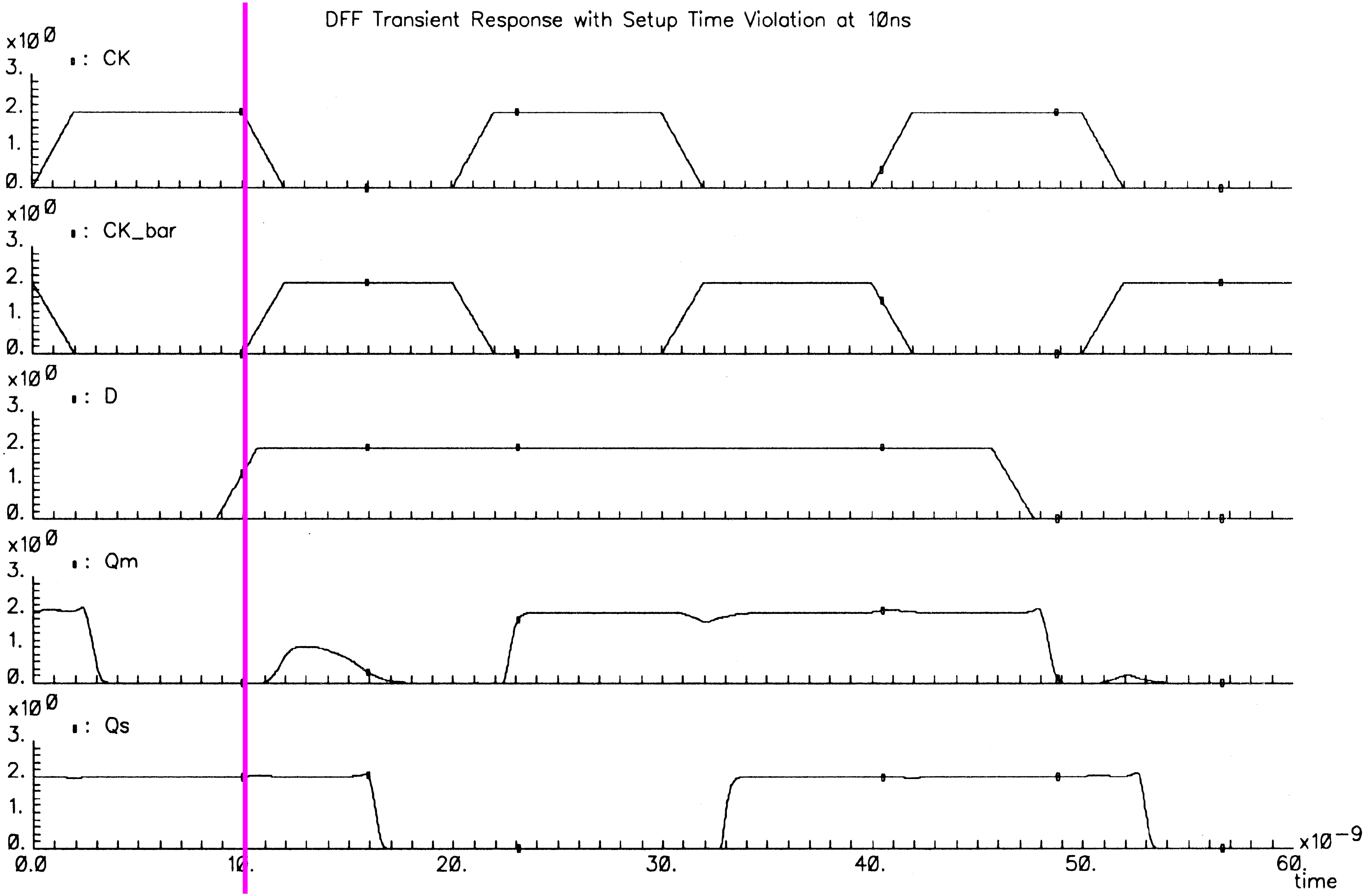


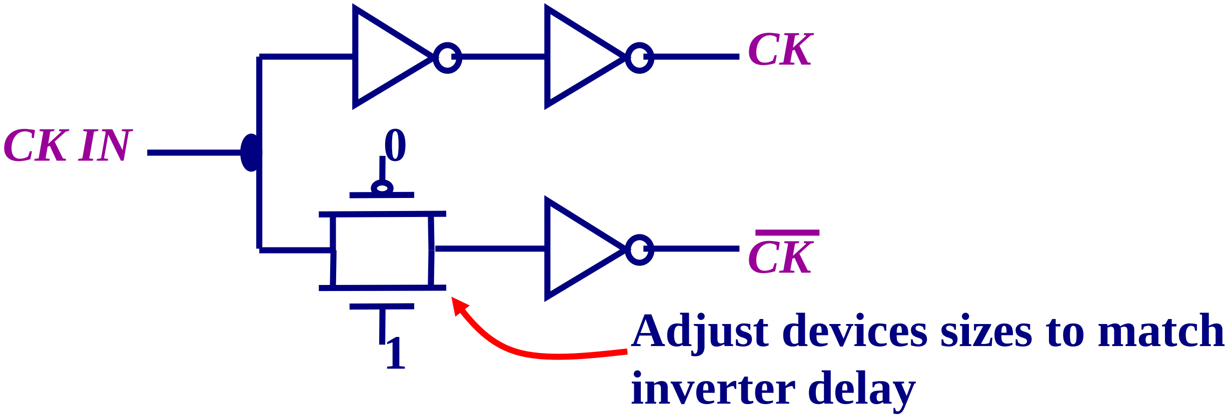
Figure 8.31 Simulated input and output waveforms of the CMOS DFF circuit in Fig. 8.30.

DFF Transient Response with Setup Time Violation



D Flip-Flop
Clock Skew Issues

- In a TG or three-state implemented flip-flop, if ***CK*** and ***$\overline{CK}$*** changes are *skewed (misaligned) enough*, then a change in Master can immediately propagate into Slave violating the master-slave (edge-triggered) concept.
- If global or shared drivers used, can use the following to reduce skew:

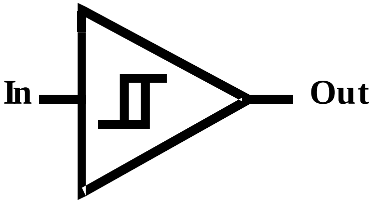


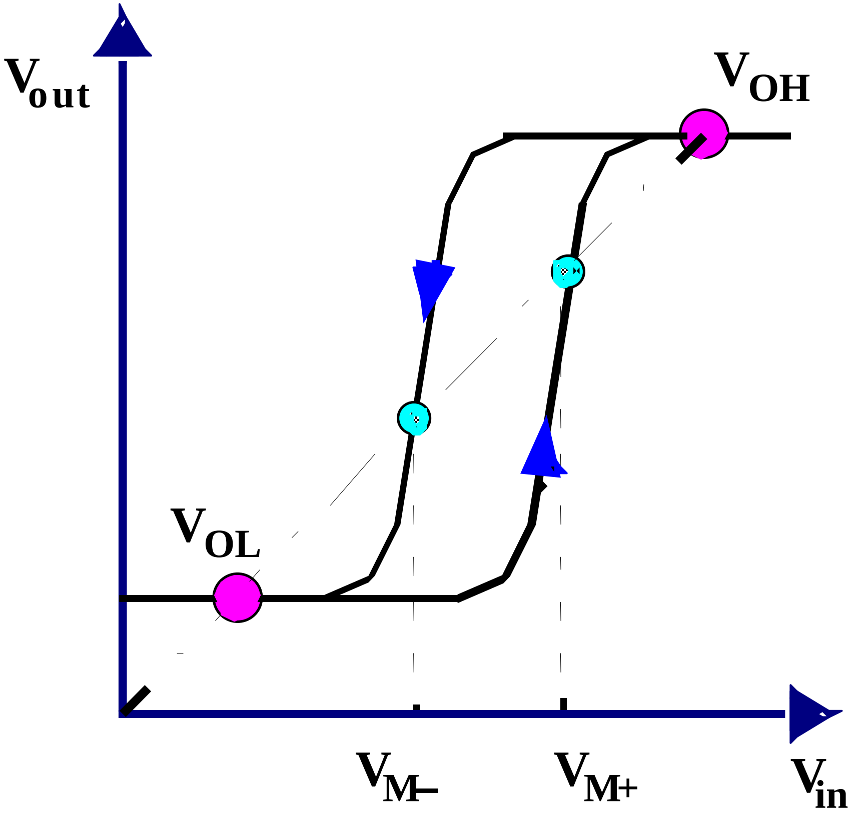
Adjust devices sizes to match inverter delay

- For the global case, skew can also arise due to interconnect delay.

Non-Bistable Sequential Schmitt Trigger

- The Schmitt trigger has an ***inverter-like*** voltage transfer characteristic, but with ***two different threshold voltages*** for increasing and decreasing input signals.

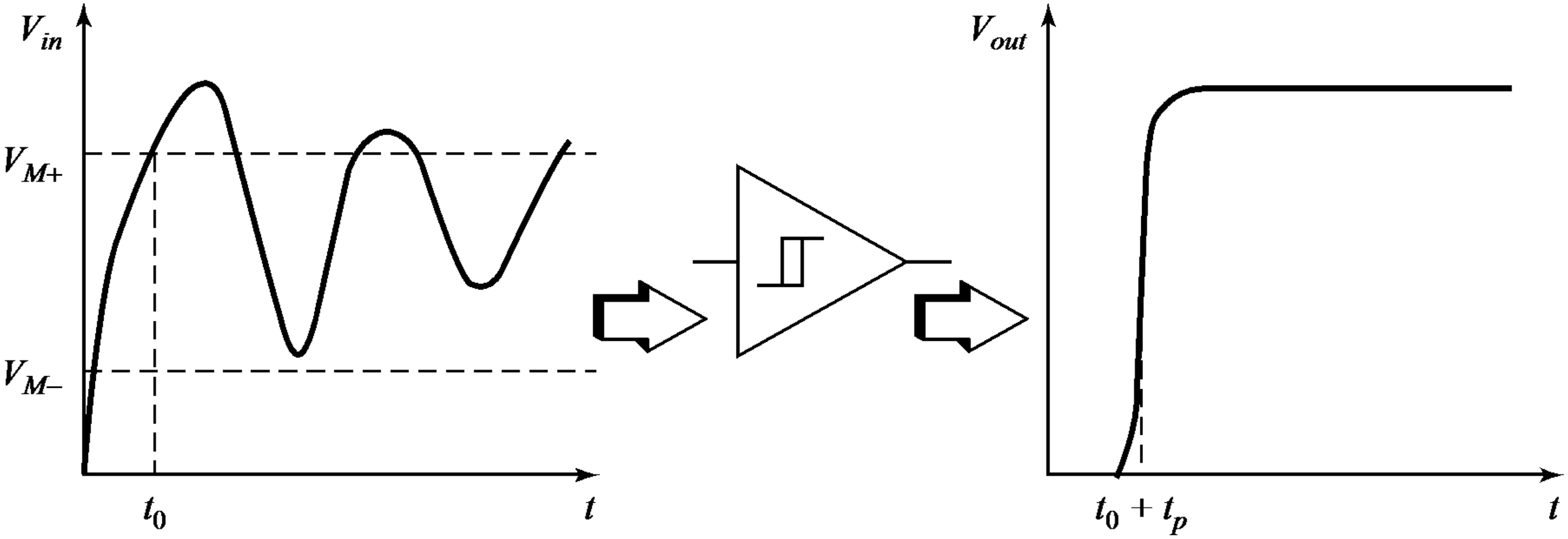




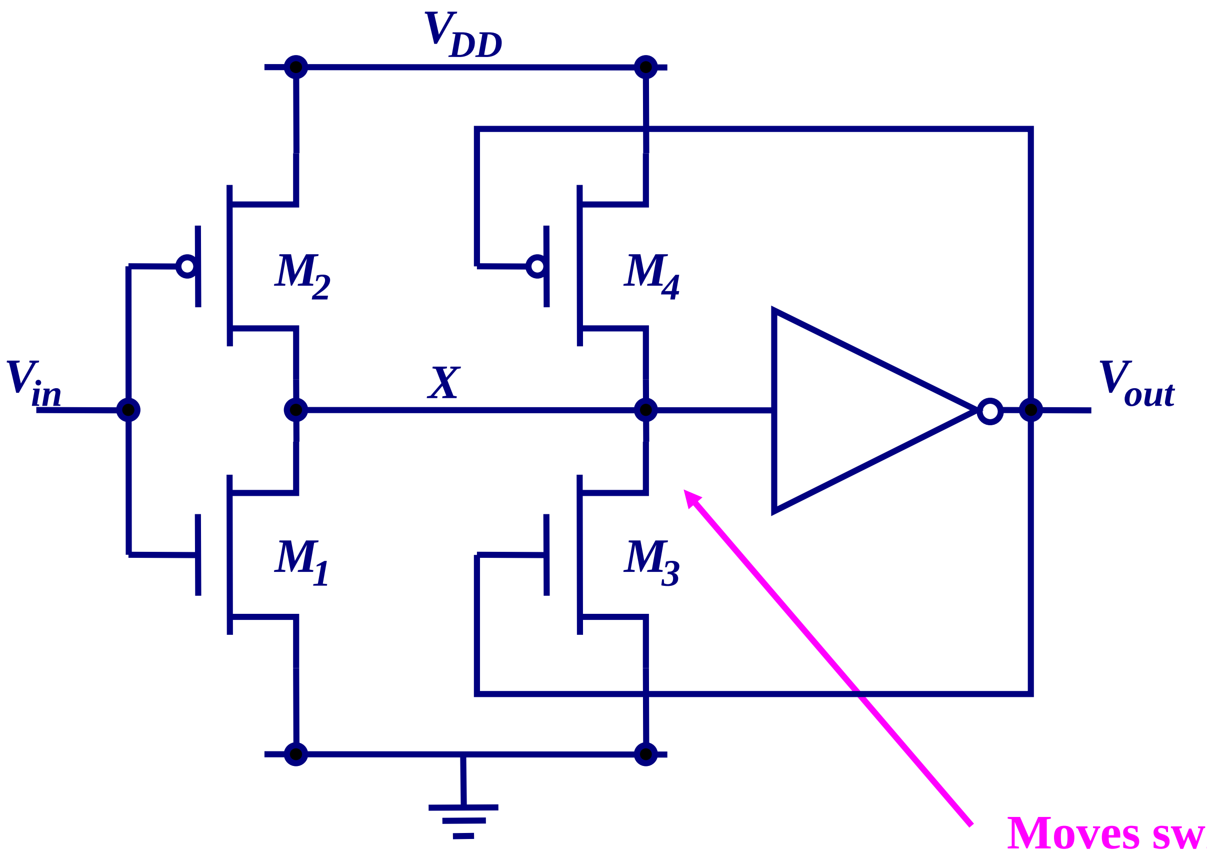
•VTC with hysteresis

•Restores signal slopes
(positive feedback)

Schmitt Trigger Application
Noise Suppression



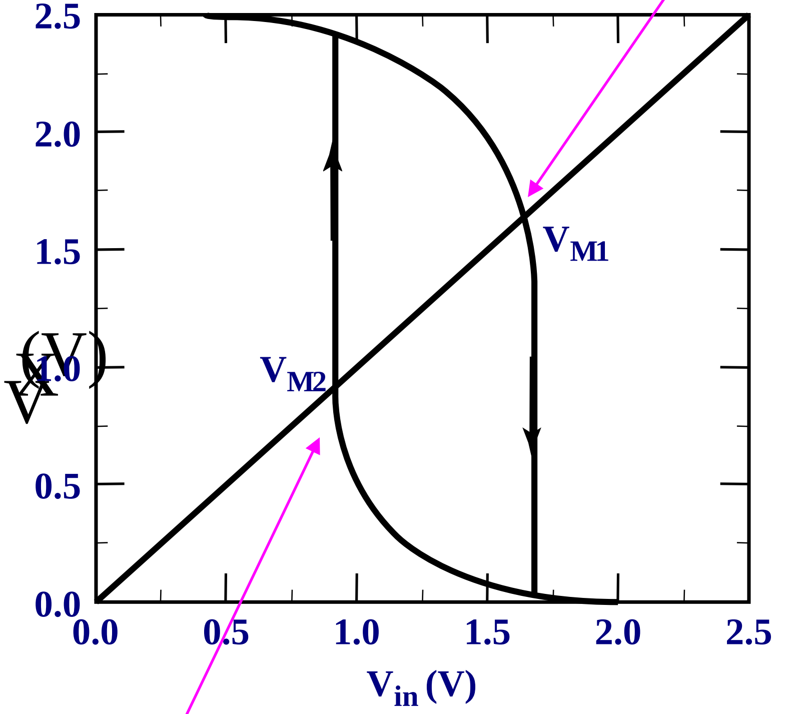
Schmitt Trigger
The Circuit(1)



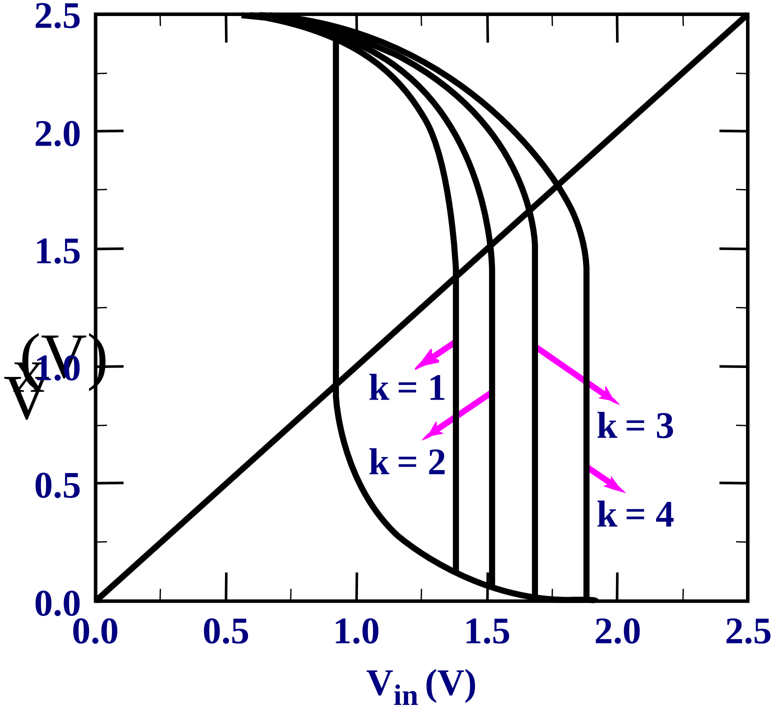
Moves switching threshold
of the first inverter

Schmitt Trigger Simulated VTC

$k_{M1}/(k_{M2}+k_{M4})$



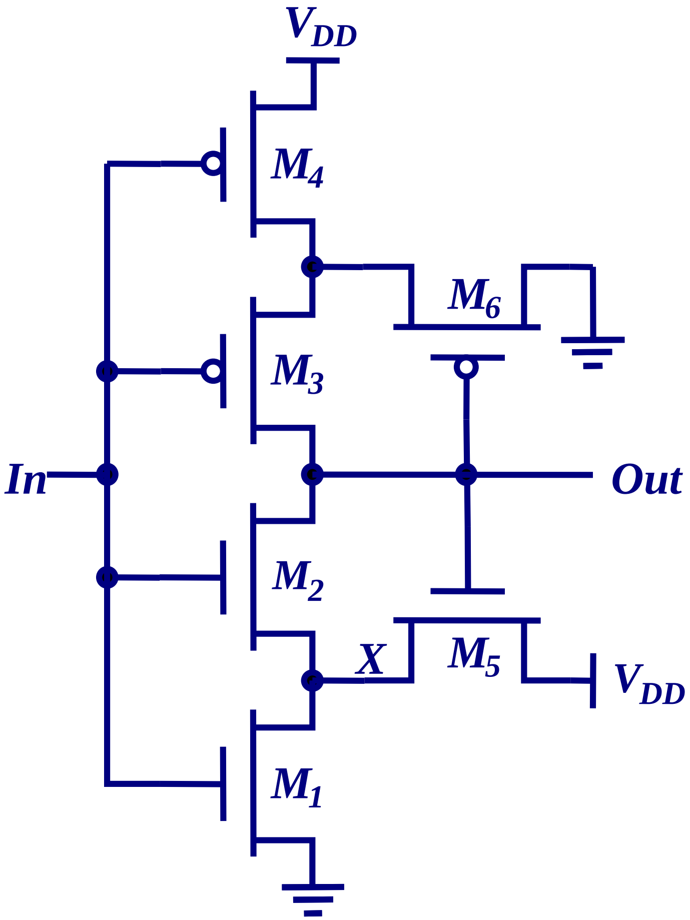
Voltage-transfer characteristics with hysteresis.



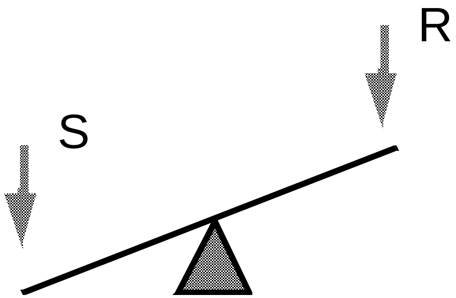
The effect of varying the ratio of the PMOS device M_4. The width is k* 0.5μm.

$(k_{M1}+k_{M3})/k_{M2}$

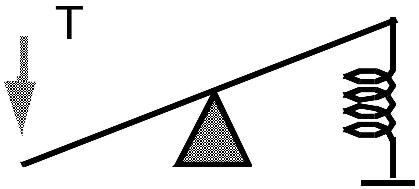
Schmitt Trigger
The Circuit(2)



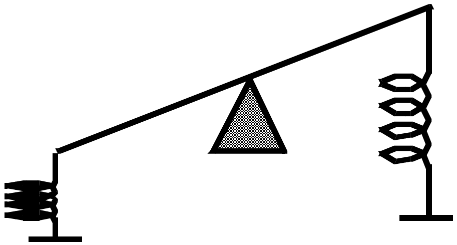
Multivibrator Circuits



Bistable Multivibrator
flip-flop, Schmitt Trigger

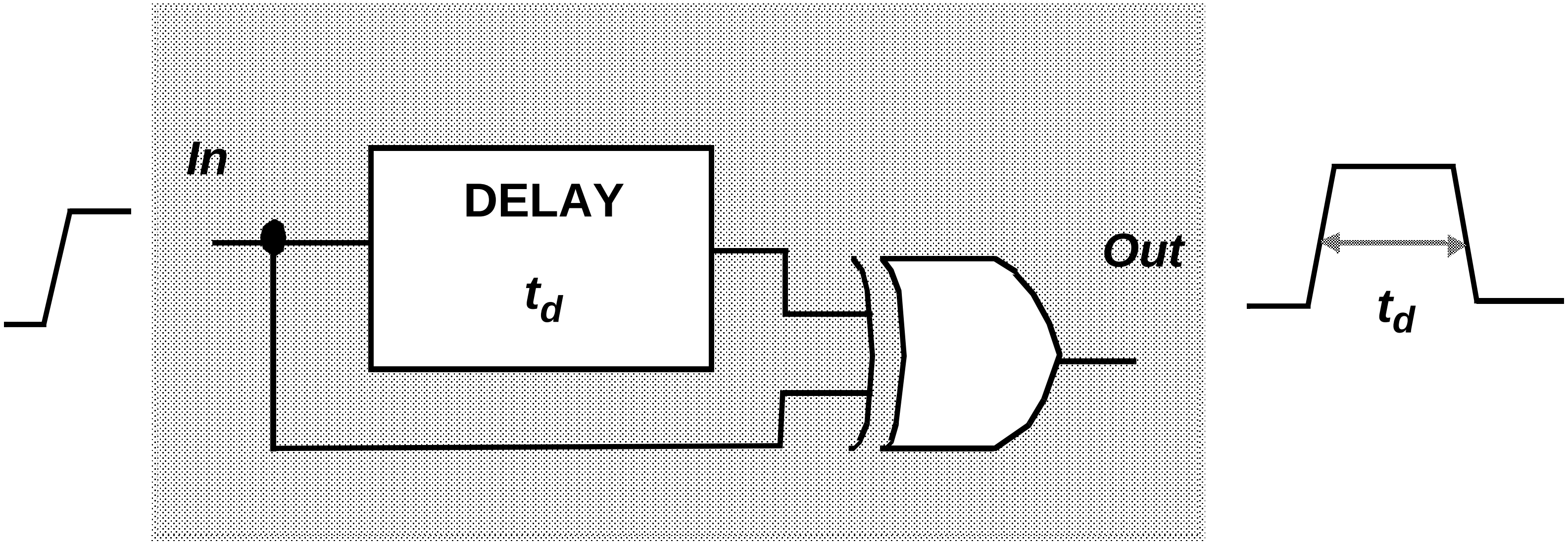


Monostable Multivibrator
one-shot

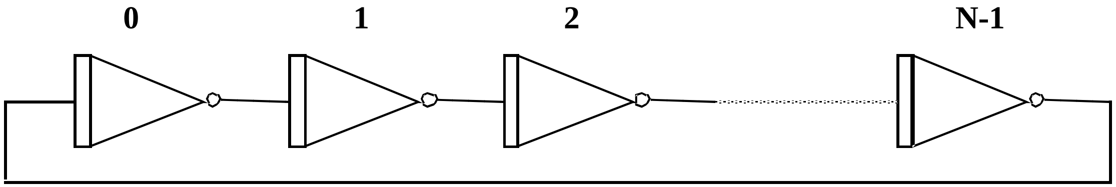


Astable Multivibrator
oscillator

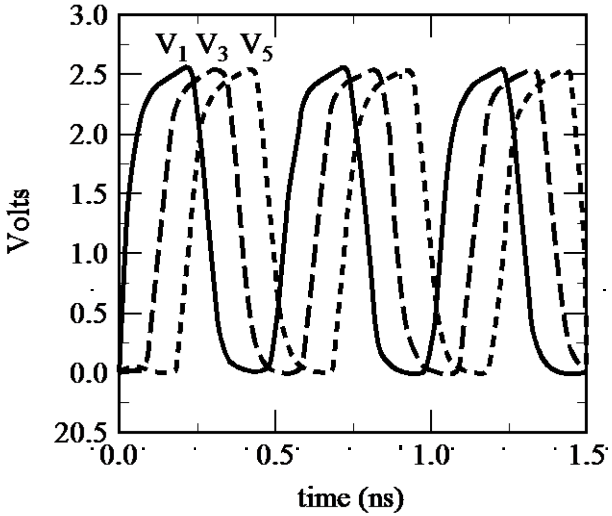
Transition-Triggered Monostable



Astable Multivibrators (Oscillators)



Ring Oscillator



simulated response of 5-stage oscillator