Introduction to Electrical Engineering

Course Code: EE 103

Department: Electrical Engineering

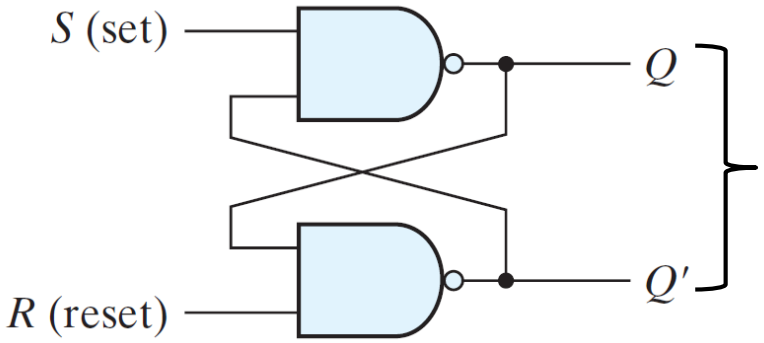
Instructor Name: B. G. Fernandes

E-mail id: bgf@ee.iitb.ac.in

Review

Sequential circuits: Output depends on the present set of inputs and also on the past output (Eg: Latches, Flip-flops, Digital Clocks)

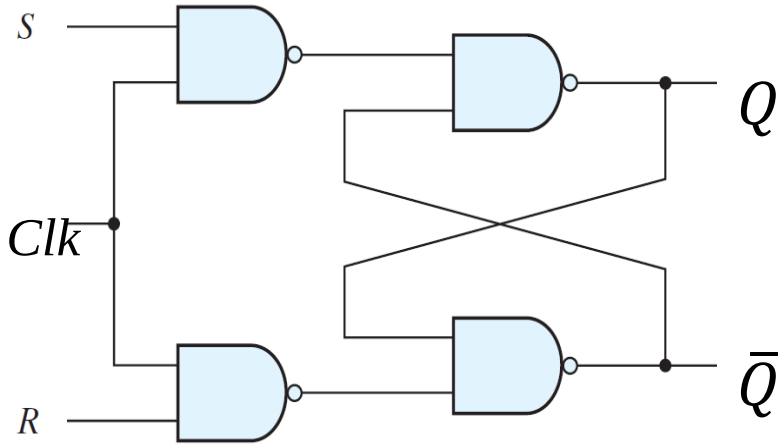
S-R Latch



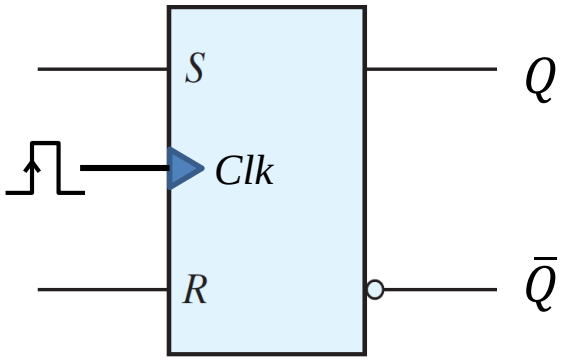
Distinct

S	R	output
1	1	No change (Hold)
0	1	Q =1
1	0	Q =0
0	0	invalid

Clocked S-R Flip-Flop:

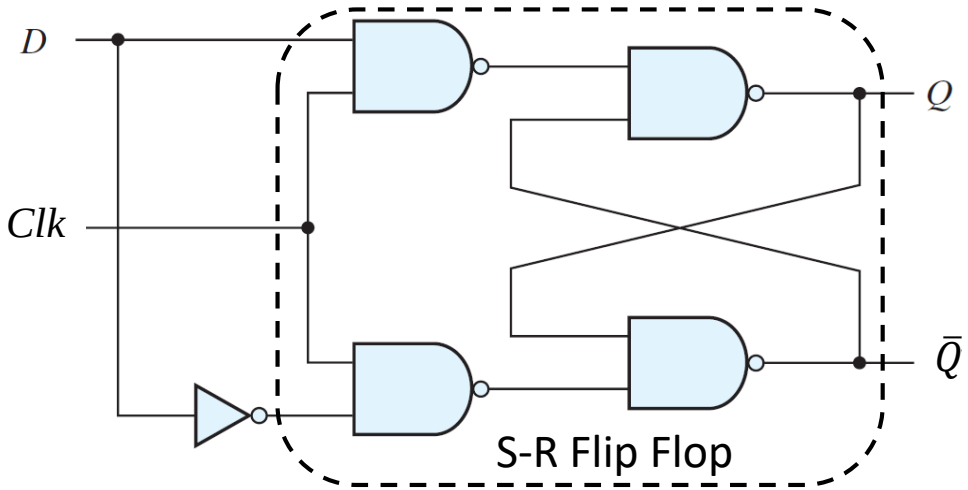


S	R	Clk	Q
0	0	↑	Q(Hold)
1	0	↑	1
0	1	↑	0
1	1	↑	invalid

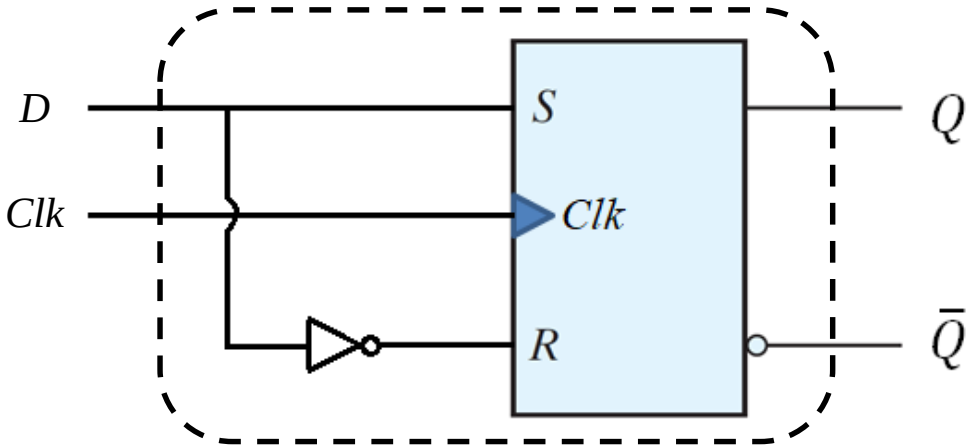


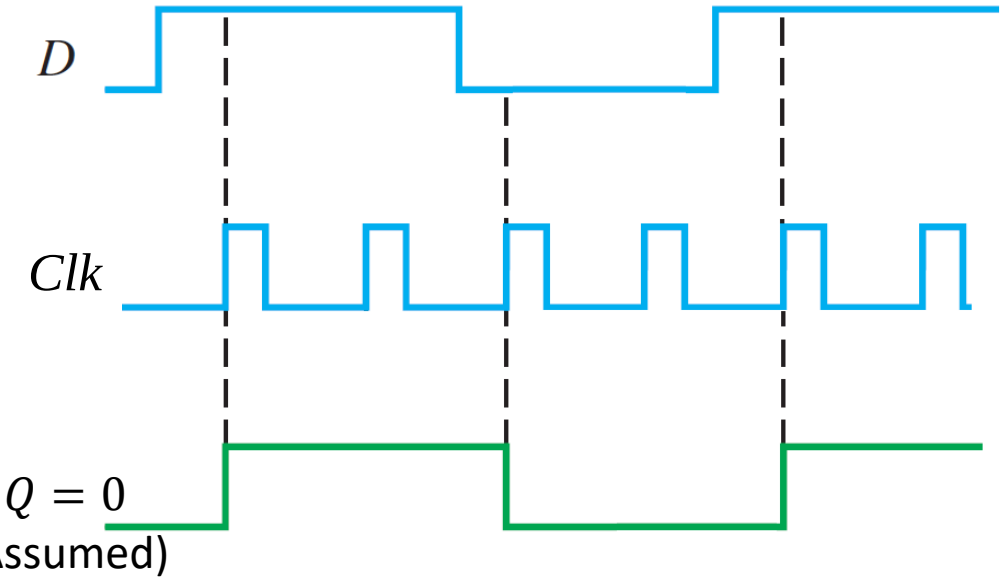
D Flip-Flop: (Data Flip-Flop)

One input split into two and used as S and R with a clock.

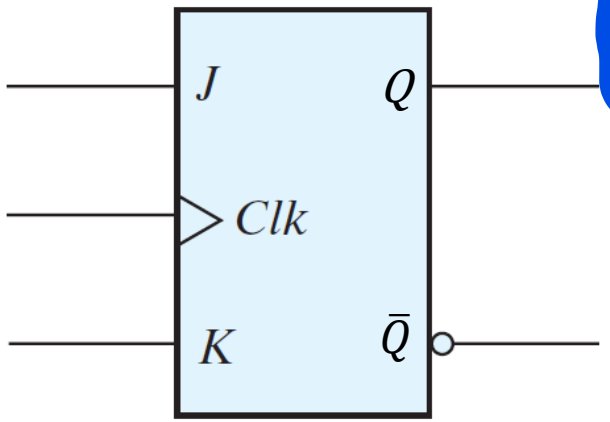


D	*Clk*	*Q*
0	↑	0
1	↑	1

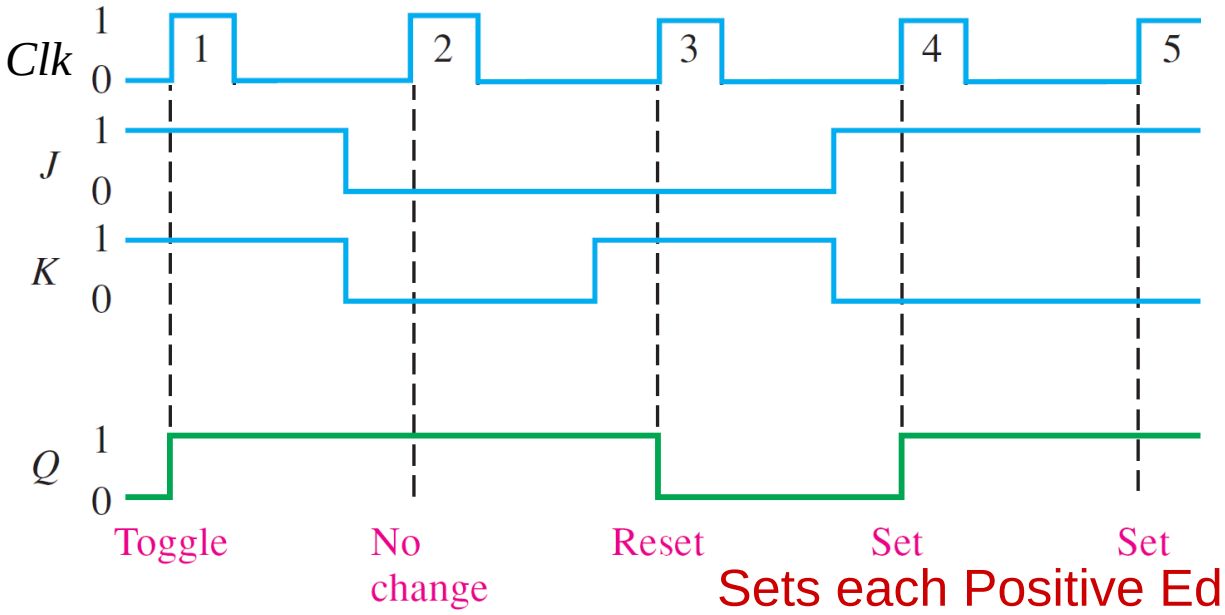




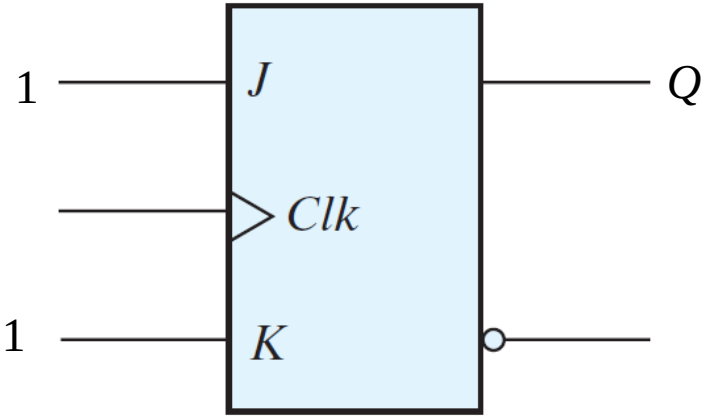
J-K Flip-Flop

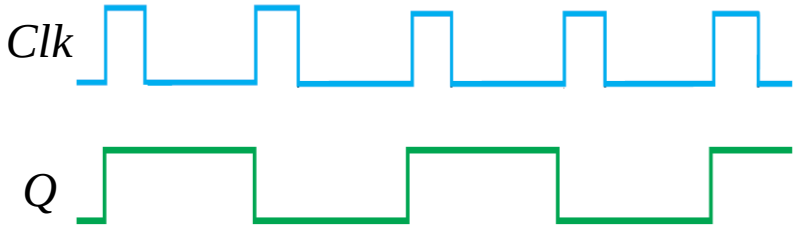


J	K	*Clk*	*Q*
0	0	↑	Q_o(Hold)
1	0	↑	1
0	1	↑	0
1	1	↑	toggle



Sets each Positive Edge Trigger.



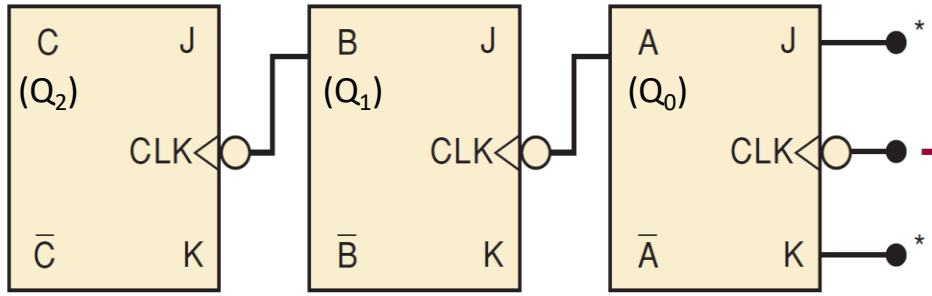


- For every second rising edge output will change .
- Frequency of signal at 'Q' is $^1/_2$ of *Clk* frequency
  ⇒÷ 2 Counter

3 bit Asynchronous (Ripple) Counter

(Asynchronous or Ripple)

Why ripple/ Asynchronous?



*All J and K inputs assumed to be 1.

Q_0 changes at every clock (+ve or –ve rising).

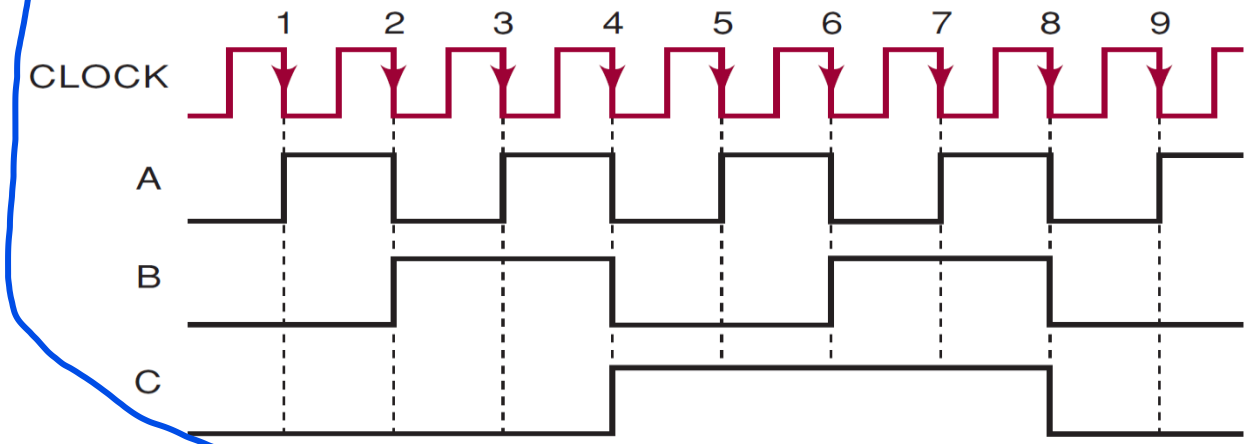
⇒ J-K in toggle mode.

Q_1 changes when Q_0 ↓

⇒ Use Q_0 as clock to 2nd Flip-Flop.

Q_2 changes when Q_1 ↓

⇒ Use Q_1 as clock to 3rd Flip-Flop.



$2^3 =$ 8 distinct states ⇒mod-8 counter

$f = \frac{f_{clock}}{2^3}$ ⇒ $f = \frac{f_0}{2^n}$ (General)

Q_2 C	Q_1 B	Q_0 A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

3 bit Asynchronous (Ripple) Counter

⇒ Only 1st Flip-Flop changes with external clock.

⇒ 2nd Flip-Flop waits for 1st Flip-Flop to ↓.

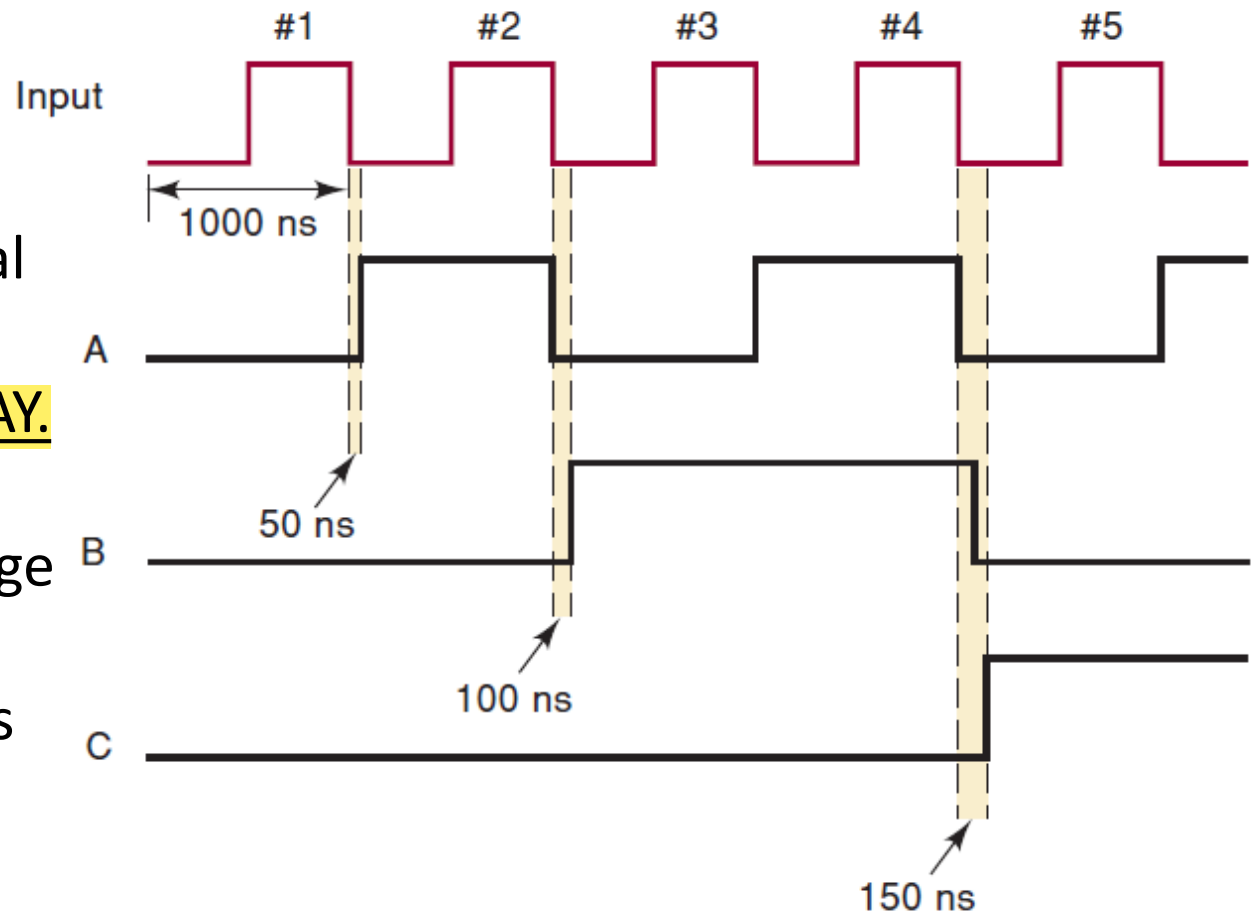
⇒ This takes a finite delay between external clock ↓ and output of 1st Flip-Flops to ↓ (a few ns) known as ⇒ PROPOGATION DELAY.

⇒ Similarly for 3rd Flip-Flop.

⇒ Outputs of all the Flip-Flops do not change simultaneously.

⇒ Hence Asynchronous or ripple (Flip-Flops respond one after another).

⇒ If clock frequency is high and there are many Flip-Flops (n-bit counter) this delay is not acceptable.



$\therefore$ Mod-4 $\Rightarrow$ uses 2 Flip-Flops (00 - 11).

Mod-16 $\Rightarrow$ uses 4 Flip-Flops (0000 - 1111).

How to Design a counter whose number $\neq 2^N$ (N=1,2,3,$\cdots$) ?

$\Rightarrow$ J-K, D, S-R are synchronous Inputs.

$\Rightarrow$ Their effect on the output is synchronized with its clock input.

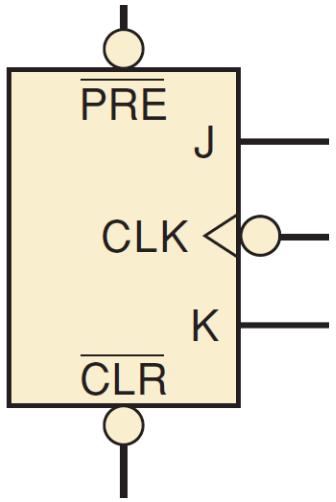
$\Rightarrow$ There are asynchronous inputs also.

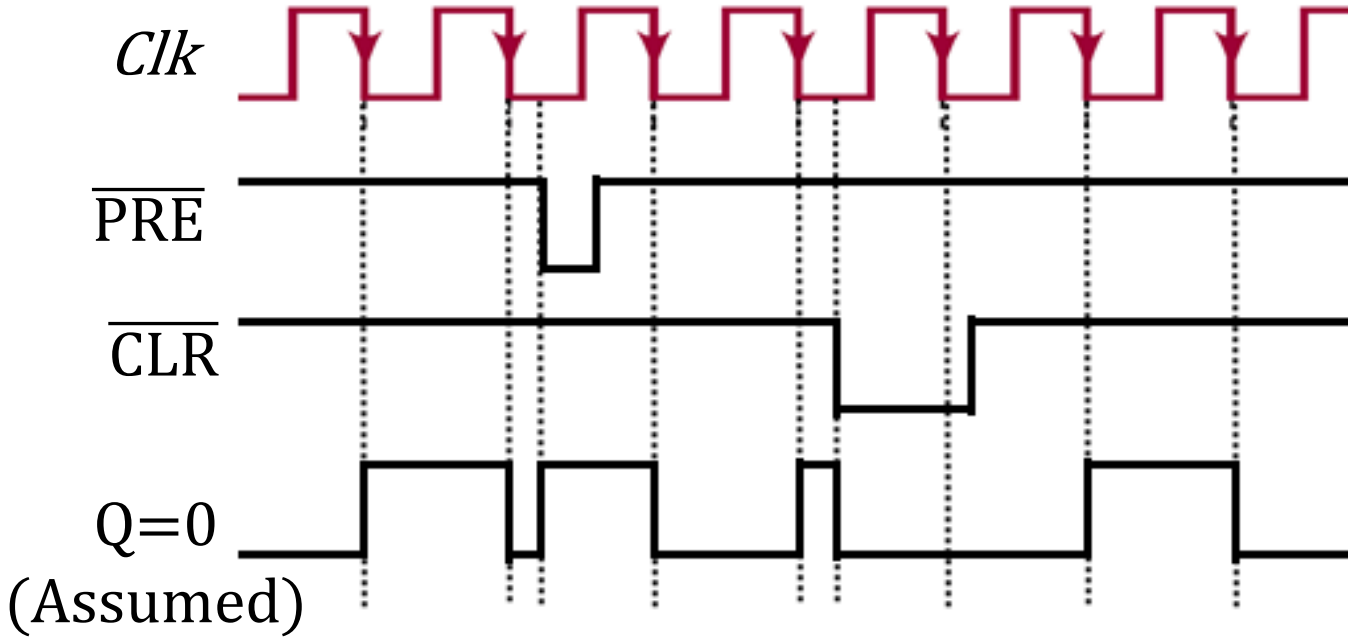
PRESET (PRE or $\overline{\mathrm{PRE}}$) $\Rightarrow$ Output= 1

when PRE pin $= 1$ ($\overline{\mathrm{PRE}} = 0$); NO MATTER what conditions are present at J, K and Clk.

$\overline{\mathrm{PRE}}$ $\Rightarrow$ is ACTIVE when LOW.

Similarly CLEAR (CLR or $\overline{\mathrm{CLR}}$) $\Rightarrow$ Output= 0 when CLR pin $= 1$ ($\overline{\mathrm{CLR}} = 0$)



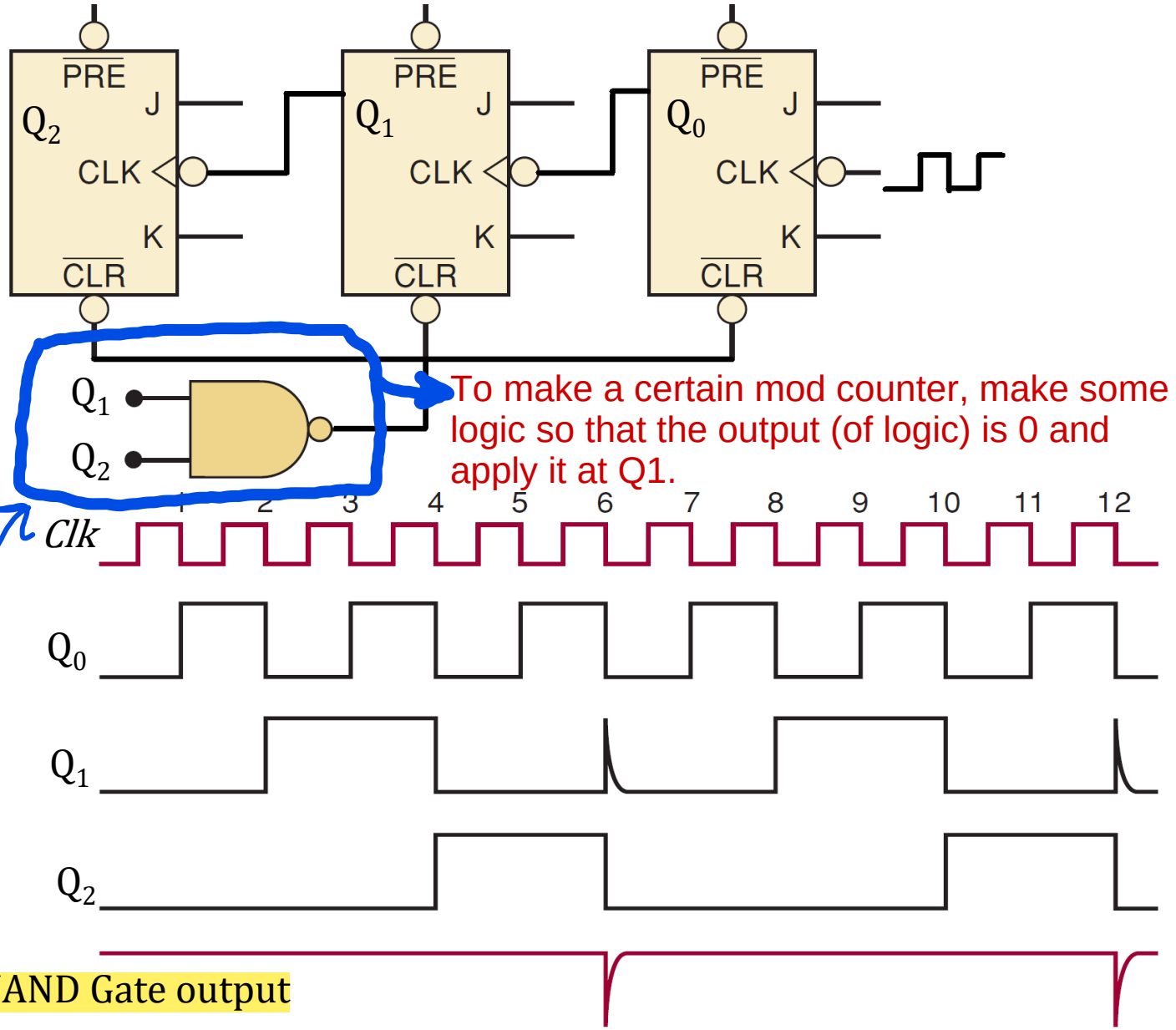


PRE ($\overline{PRE}$)	CLR ($\overline{CLR}$)	
0(1)	0(1)	Clocked operation
1(0)	0(1)	Q=1
0(1)	1(0)	Q=0

Mod-6 Counter:

Q_2 C	Q_1 B	Q_0 A	
0	0	0	6 Stable States
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	(Temporary)

Reset all the counts to 000 when this happens.



Asynchronous Down Counter:

They change at alternate edge trigger.

Q_2	Q_1	Q_0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0
1	1	1

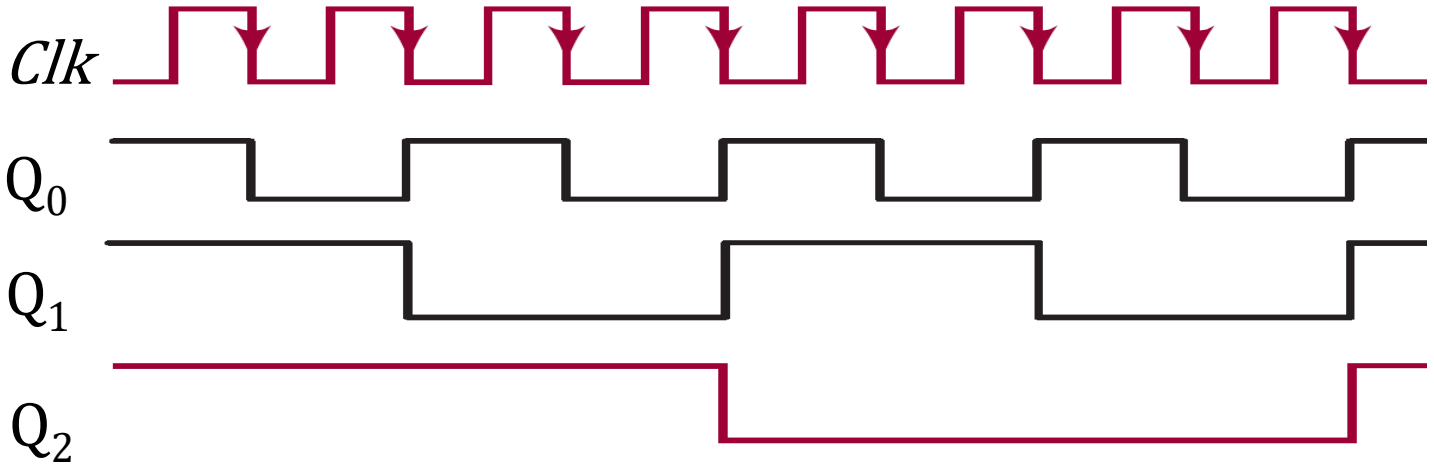
Q_0 →changes at every clock pulse (either ⌐↑ or ↓¬)

J=K=1

Q_1 → changes when Q_0 ⌐↑ or $\overline{Q_0}$ ↓¬

∴ Use $\overline{Q_0}$ as clock to 2nd Flip-Flop if it is NGT

Q_2 → changes when Q_1 ⌐↑ or $\overline{Q_1}$ ↓¬



Synchronous Counter (Parallel Counter):

- All Flip-Flops receive clock simultaneously.
- Making J=K=1 and apply same clock pulses to all Flip-Flops will not give us the required ‘Mod’ number or sequence.
- Some means must be used to control when an Flip-Flop should toggle and when it remains unaffected by a clock pulse.

Characteristic Table

J	K	Q_{t+1}
0	0	Q(Hold)
1	0	1
0	1	0
1	1	toggle

Learn this to derive this

Transition Table

N^{th} state	$(N+1)^{th}$ state	Inputs Required at J	Inputs Required at K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Design Procedure: Mod-8 Synchronous Counter

Steps to make mod 8 synchronous counter :

1. Determine the desired number of bits (Flip-Flops) and the possible states .

Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

Example with 3 bits and corresponding 8 states.

Design Procedure: Mod-8 Synchronous Counter

2. Prepare a table that lists all the present states and their next state.

	Q_2	Q_1	Q_0	Q_{2+}	Q_{1+}	Q_{0+}	
(0)	0	0	0	0	0	1	(1)
(1)	0	0	1	0	1	0	(2)
(2)	0	1	0	0	1	1	(3)
(3)	0	1	1	1	0	0	(4)
(4)	1	0	0	1	0	1	(5)
(5)	1	0	1	1	1	0	(6)
(6)	1	1	0	1	1	1	(7)
(7)	1	1	1	0	0	0	(0)
	0	0	0	0	0	1	

N^{th} state	$(N+1)^{th}$ state	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Design Procedure: Mod-8 Synchronous Counter

3. Add two column per Flip-Flop (one for each J and K). There will be six columns for each 'PRESENT' state, indicate the levels requires at each J and K input in order to produce the transition to the 'NEXT' state

Q_2	Q_1	Q_0	Q_{2+}	Q_{1+}	Q_{0+}	J_0	K_0	J_1	K_1	J_2	K_2
0	0	0	0	0	1	1	X	0	X	0	X
0	0	1	0	1	0	X	1	1	X	0	X
0	1	0	0	1	1	1	X	X	0	0	X
0	1	1	1	0	0	X	1	X	1	1	X
1	0	0	1	0	1	1	X	0	X	X	0
1	0	1	1	1	0	X	1	1	X	X	0
1	1	0	1	1	1	1	X	X	0	X	0
1	1	1	0	0	0	X	1	X	1	X	1
0	0	0	0	0	1						

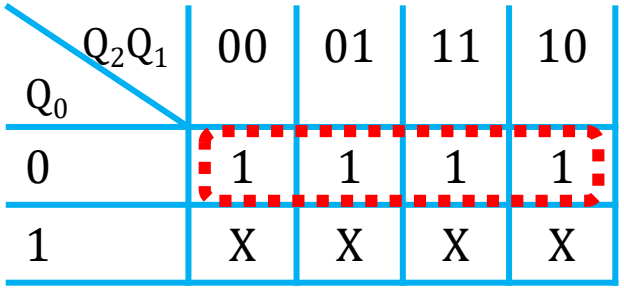
N^{th} state	$(N+1)^{th}$ state	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Now learn this to derive this :(

Design Procedure: Mod-8 Synchronous Counter

4. Design the logic circuit needed to generate the levels required at each J&K input

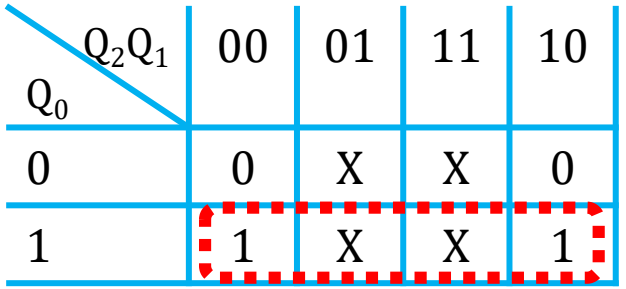
J_0



Q_0 \ Q_2Q_1	00	01	11	10
0	1	1	1	1
1	X	X	X	X

$J_0=1 (\text{or } J_0=\overline{Q_0})$

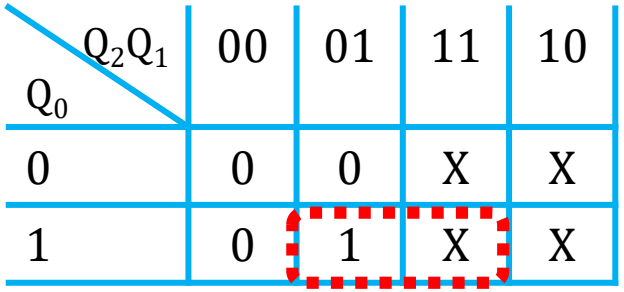
J_1



Q_0 \ Q_2Q_1	00	01	11	10
0	0	X	X	0
1	1	X	X	1

$J_1=Q_0$

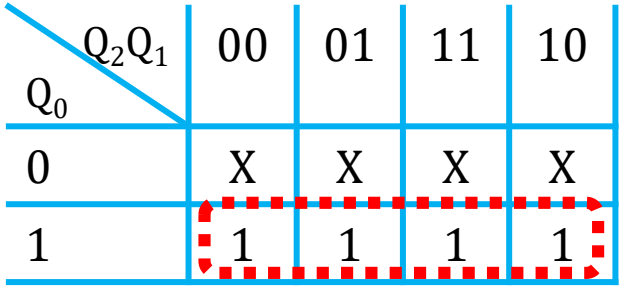
J_2



Q_0 \ Q_2Q_1	00	01	11	10
0	0	0	X	X
1	0	1	X	X

$J_2=Q_0Q_1$

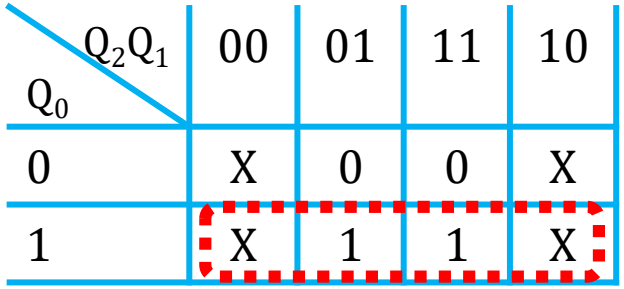
K_0



Q_0 \ Q_2Q_1	00	01	11	10
0	X	X	X	X
1	1	1	1	1

$K_0=1 (\text{or } K_0=Q_0)$

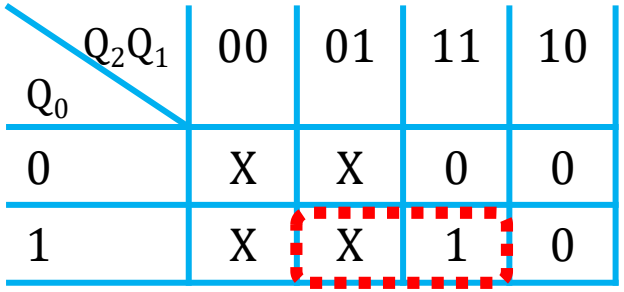
K_1



Q_0 \ Q_2Q_1	00	01	11	10
0	X	0	0	X
1	X	1	1	X

$K_1=Q_0$

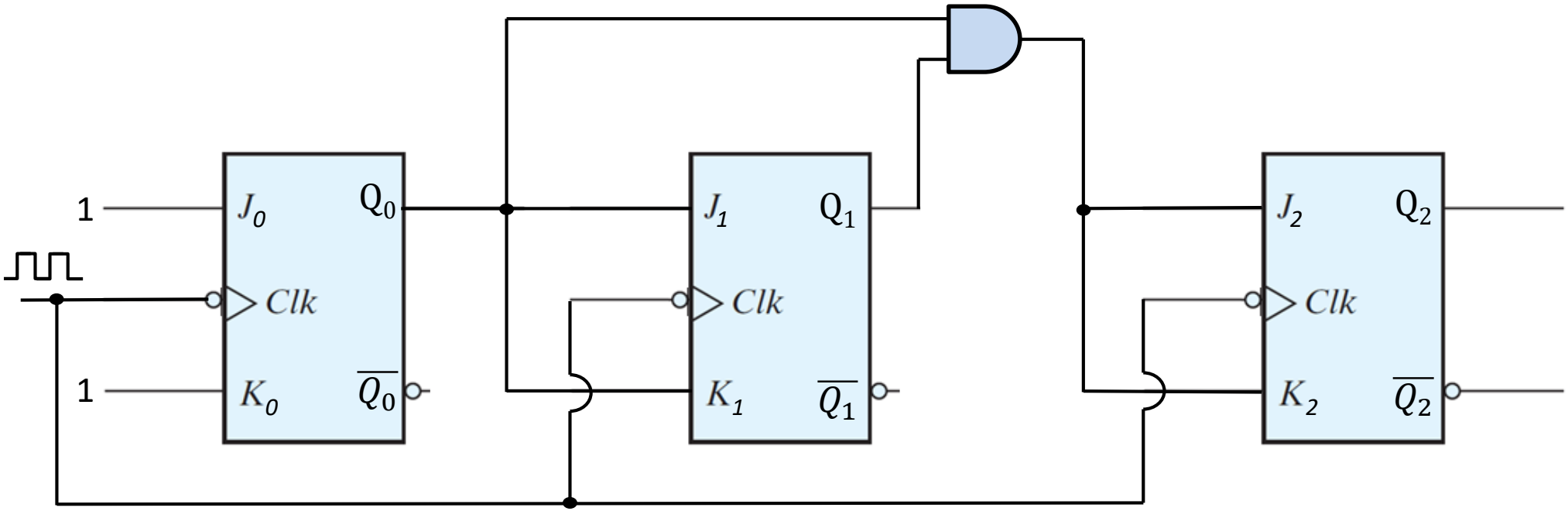
K_2



Q_0 \ Q_2Q_1	00	01	11	10
0	X	X	0	0
1	X	X	1	0

$K_2=Q_0Q_1$

Logic Realisation: Mod-8 Synchronous Counter



$J_0=1$ (or $J_0=\overline{Q_0}$)

$J_1=Q_0$

$J_2=Q_0Q_1$

$K_0=1$ (or $K_0=Q_0$)

$K_1=Q_0$

$K_2=Q_0Q_1$