Ring Counters and Johnson's Ring Counter

by

Prof. Sujata Wakchaure

Ring Counter

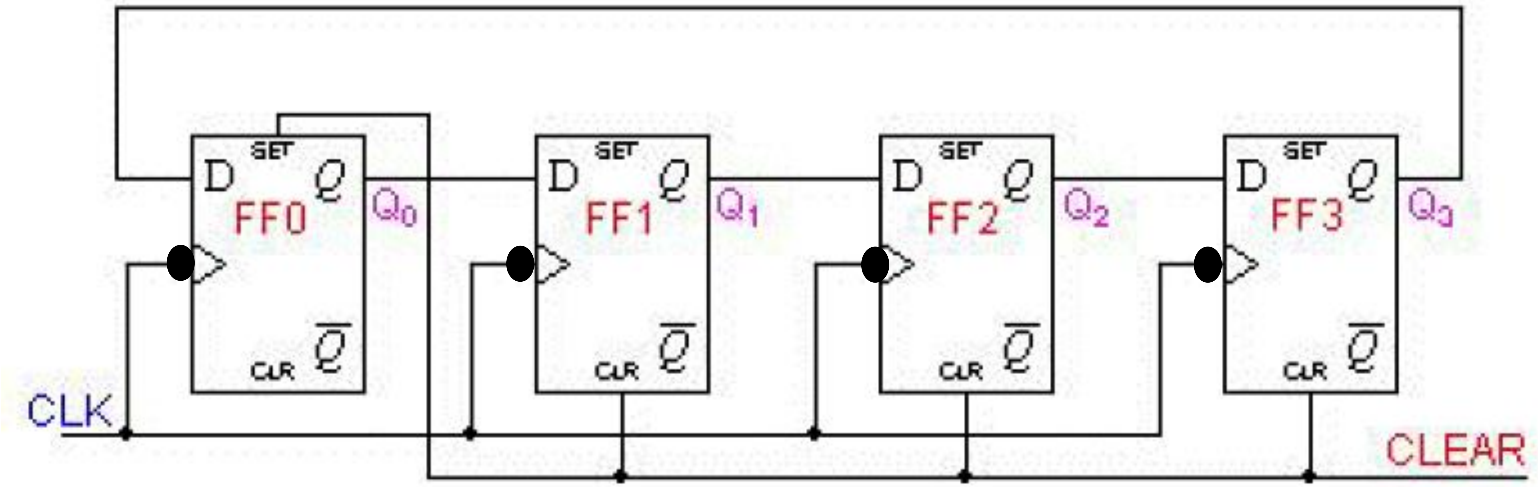
- It is a sequential logic ckt that is constructed using shift register

- Same data recirculates in the counter depending on the clock pulse

- It is self decoding ckt

- The O/P of last shift register is fed to the input of the 1st register

Ring Counter (Contd..)

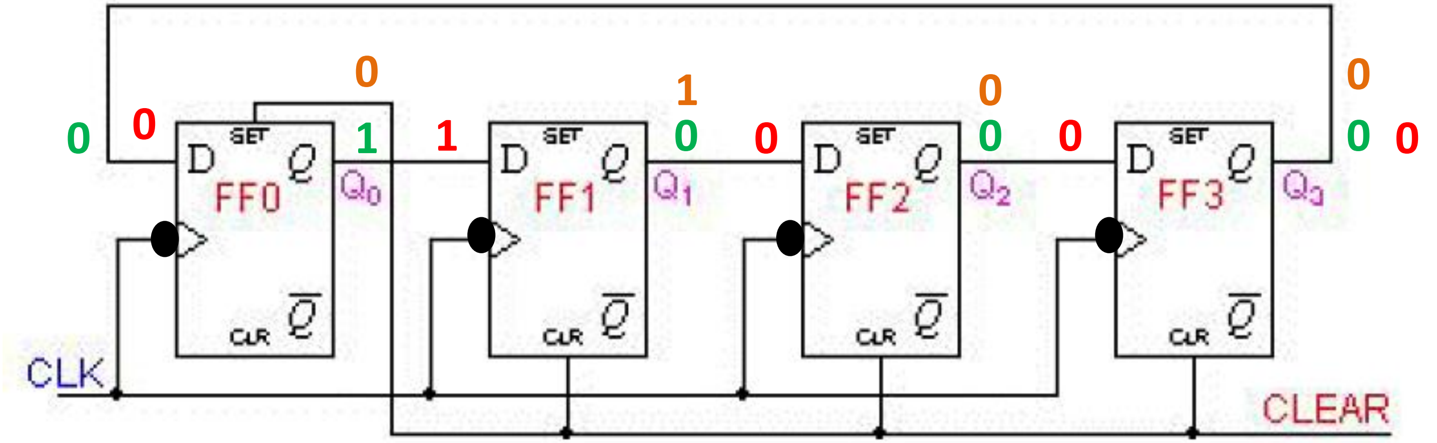


- For Normal Counter
  - » No. of states = $2^n = 2^4 = 16$
  - » Max Count = 2^n-1 = 16 -1
    = 15
- For Ring Counter
  - » No. of States = No. of FF used

Ring Counter (Contd..)

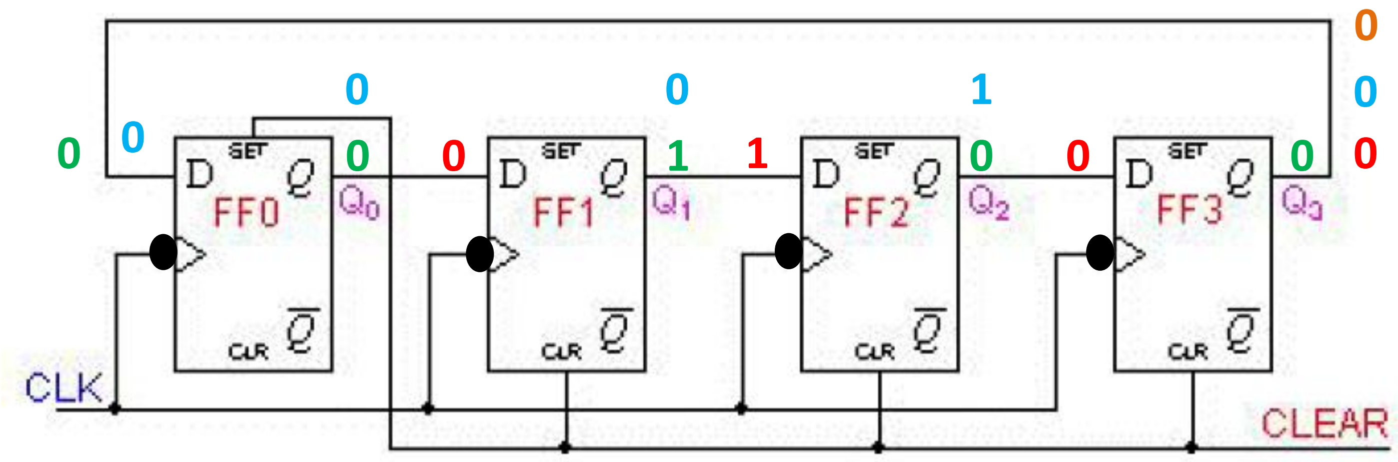
- Ring counter is Synchronous counter and we have used Overriding I/Ps (ORI)
- ORI => Preset(PR) and Clear(CLR)
- If PR = 0 then Q = 1 ← **It will set FF**
- If CLR = 0 then Q = 0
- Why PR & CLR are called as ORI?
- For FF-1, 2, & 3 ORI is connected to CLR & for FF-0 it is connected to PR
- Therefore, Q1=0
  Q2=0
  Q3=0
  Because CLR=0
  Q0=1 => Because PR=0

Ring Counter (Contd..)



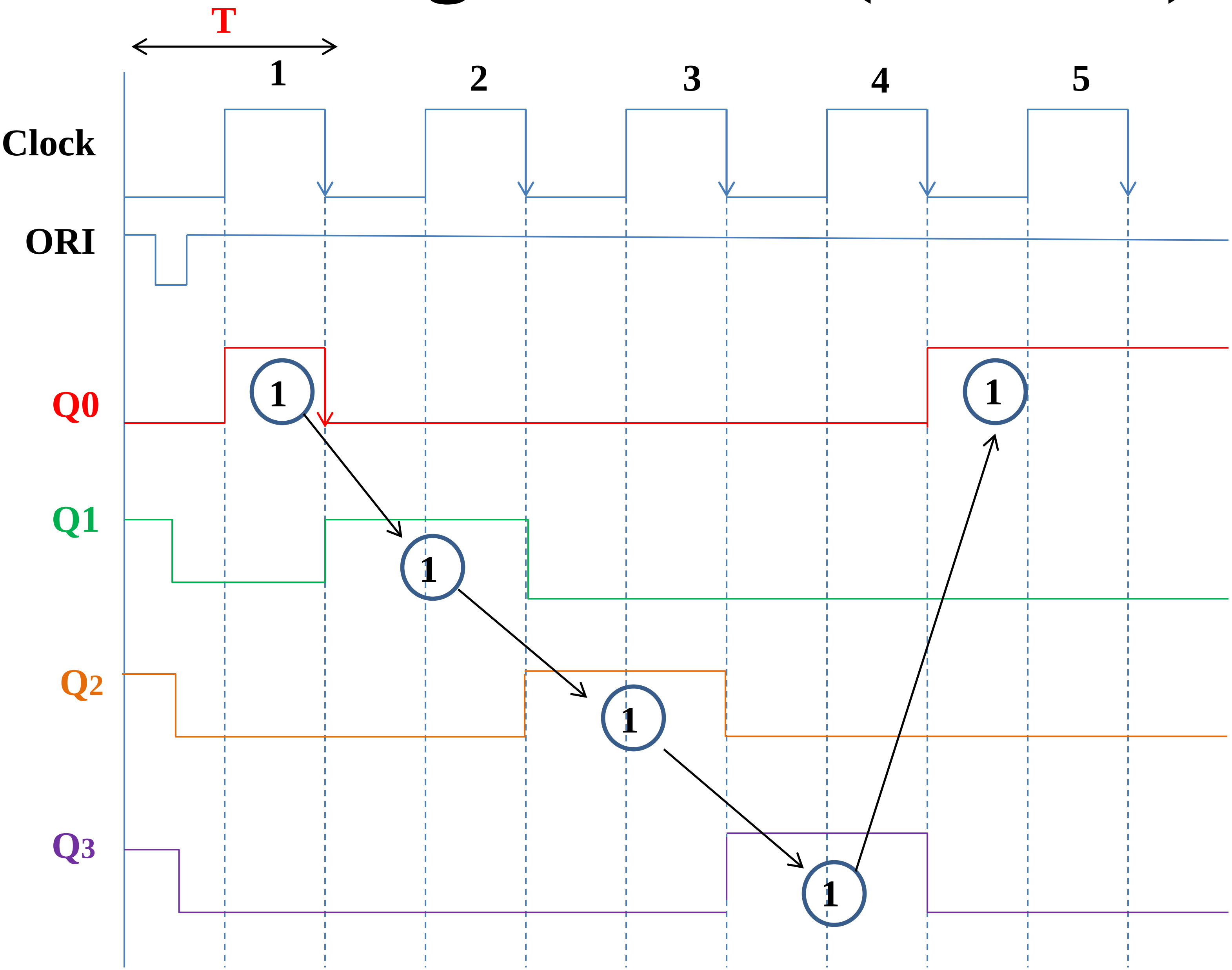
ORI	CLK	Q0	Q1	Q2	Q3
	X	1	0	0	0
1	↓	0	1	0	0
1	↓				
1	↓				
1	↓				

Ring Counter (Contd..)



ORI	CLK	Q0	Q1	Q2	Q3
‾|_|‾	X	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0

Ring Counter (Contd..)



Johnson’s Ring Counter

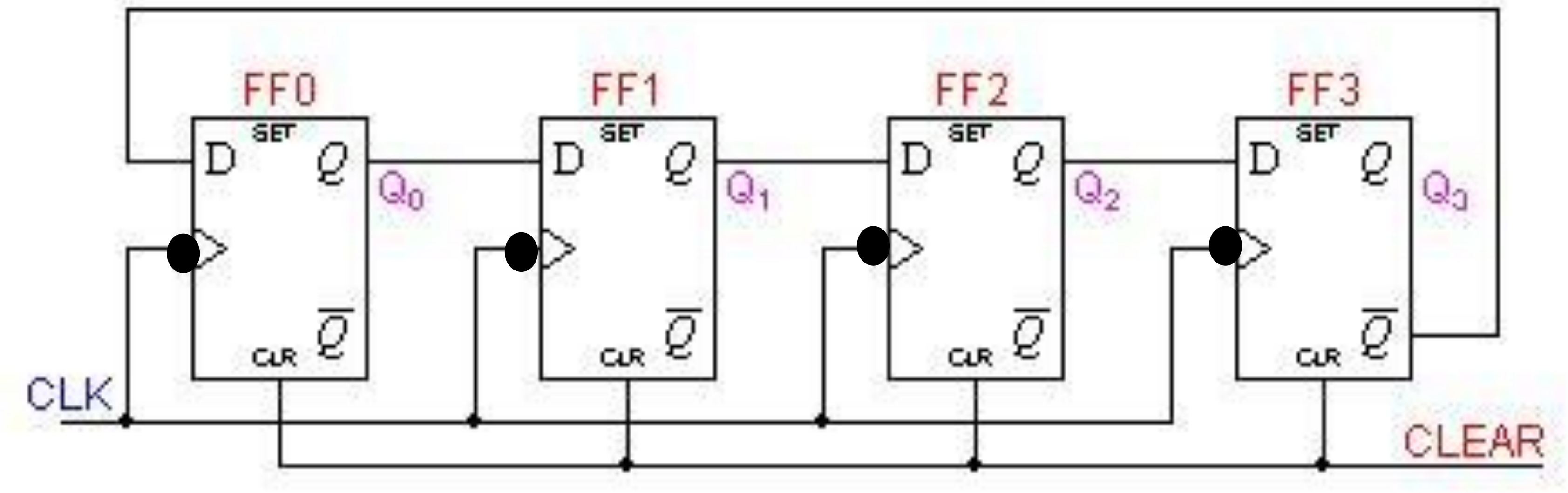
- Also known as **Twisted / Switch Tail Ring Counter**

- It is modified Ring Counter, where the inverted O/P from the last FF is connected to the I/P to the first FF

- If we compare Ring Counter and Johnson’s Ring Counter then Johnson’s Ring Counter is much more better than Ring Counter.

- There are 2 changes in Johnson’s Ring Counter:
  1. The shifting to the CLR I/P than the Preset I/P
  2. $\overline{Q3}$ is connected to the D0

Johnson's Ring Counter (Contd..)



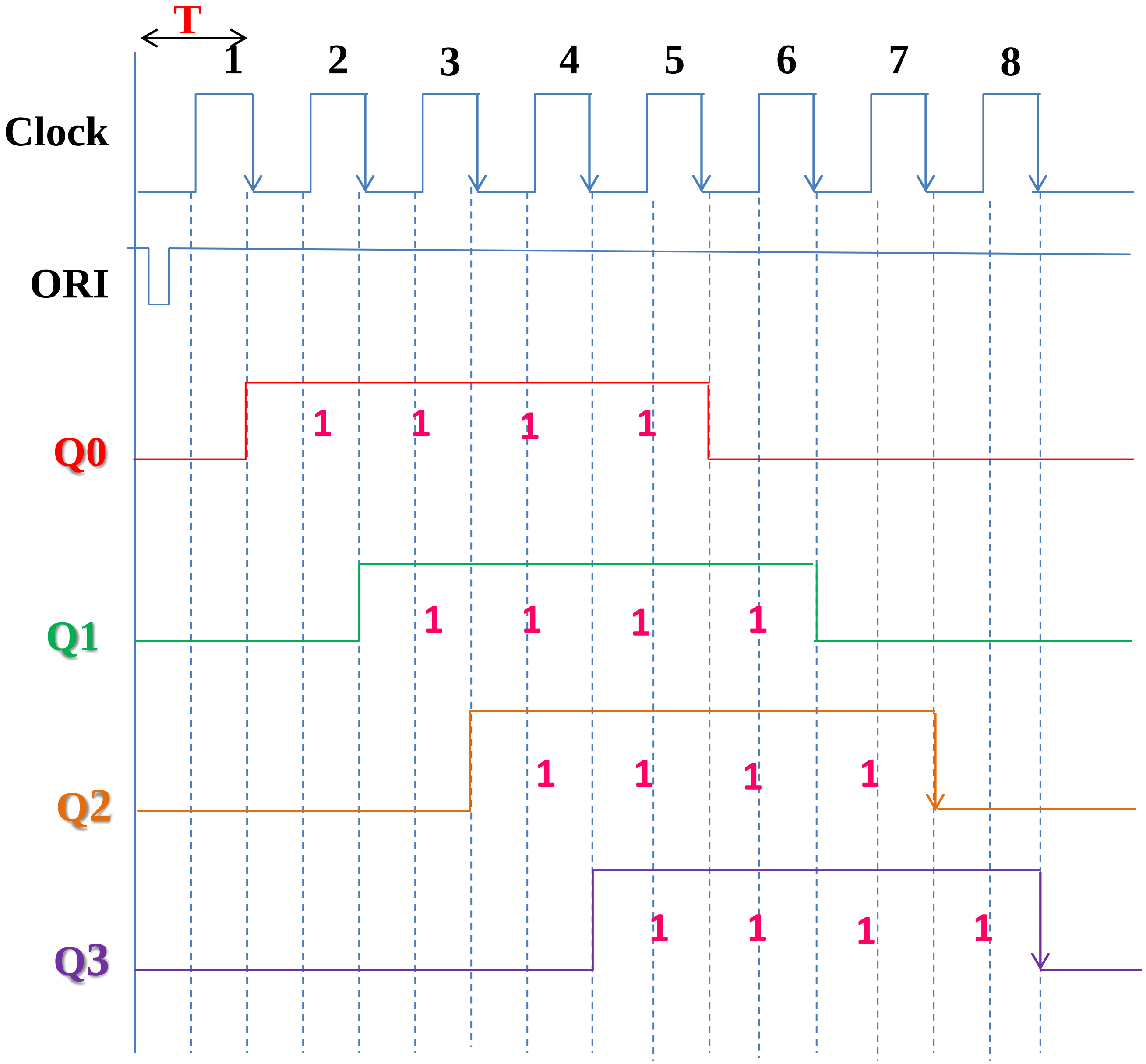
- Initially when ORI I/P is low, the O/P Q0=Q1=Q2=Q3= 0

Johnson’s Ring Counter (Contd..)

CLR	CLK	Q0	Q1	Q2	Q3
⊔	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1
1	↓	0	1	1	1
1	↓	0	0	1	1
1	↓	0	0	0	1
1	↓	0	0	0	0

No. of States = 2 x No. of FF

Johnson’s Ring Counter (Contd..)



BCD Counter

- It is a special type of counter which can count to ten on the applications of a clock signal

- **Negative Edge Triggered**
  - Q is Clock => Up Counter

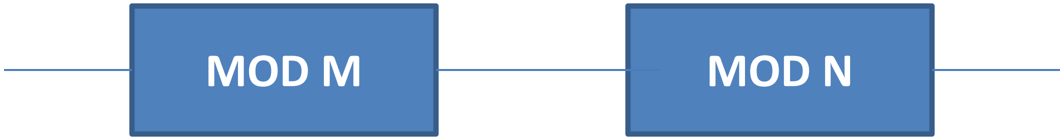
- **Negative Edge Triggered**
  - $\overline{Q}$ is Clock => Down Counter

- **Positive Edge Triggered**
  - $\overline{Q}$ is Clock => Up Counter

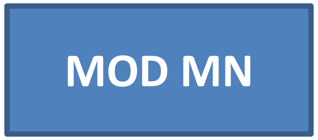
- **Positive Edge Triggered**
  - Q is Clock => Down Counter

Cascading of Counters

- What if I have 2 counters?

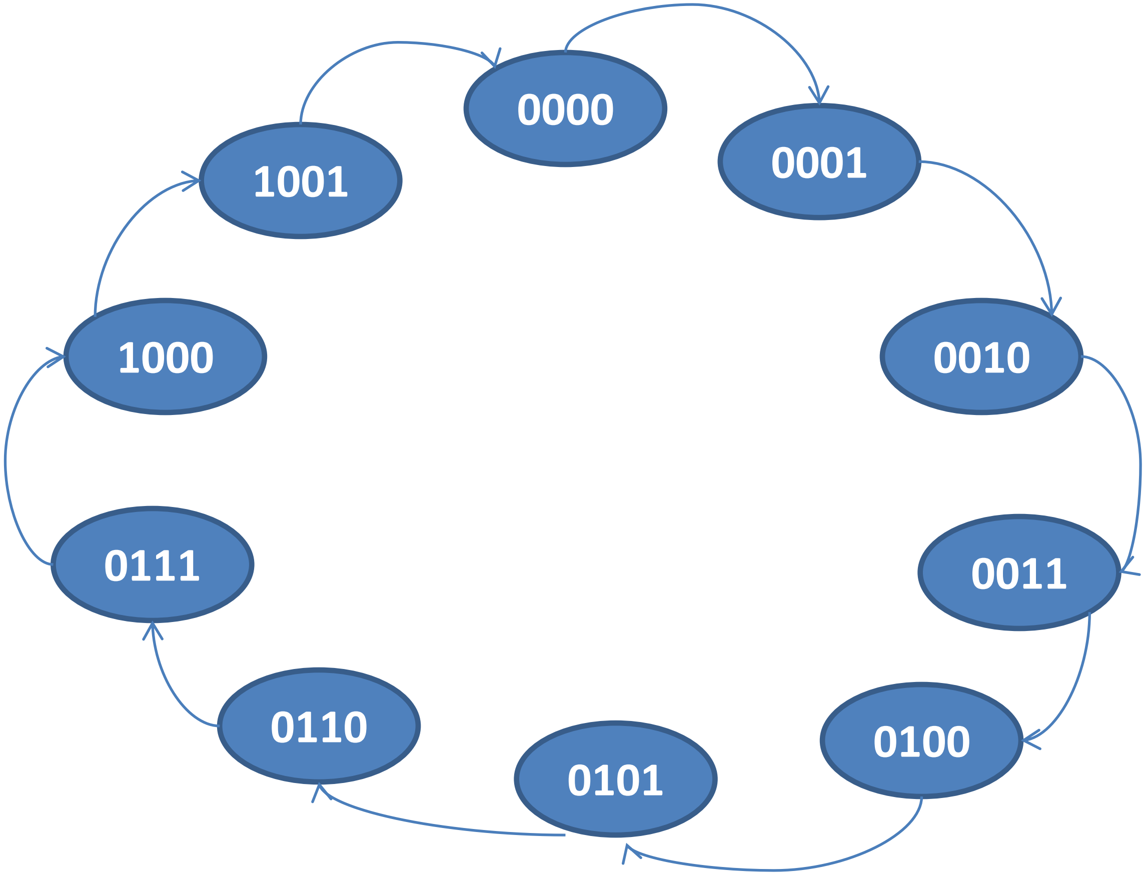


- After cascading the resulting FF:



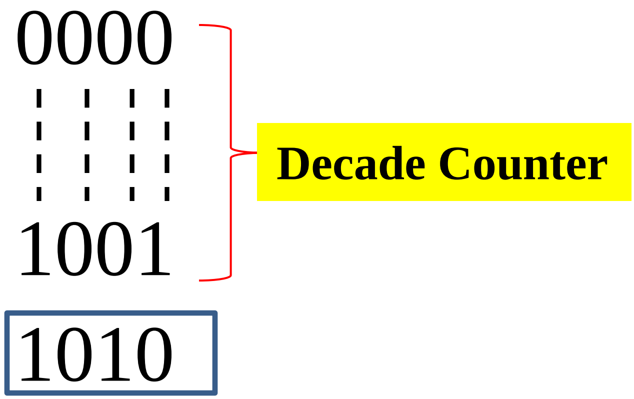
BCD Counter (Contd..)

- No. of States = 10



BCD Counter (Contd..)

- As we have 4 bit our counting will start from 0000 to 1111
- But we want to stop it at 1001 because this is BCD Counter
- So what we can do?



0000

1001

Decade Counter

1010

- We have to Convert 4 bit asynchronous Up Counter to Decade Counter

BCD Counter (Contd..)

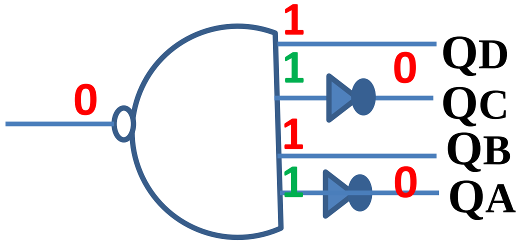
- To achieve this conversion we need to focus on 1010 state

- As soon as 1010 has arrived, we have to reset our FF by using 2 asynchronous I/Ps i.e. PR and CLR

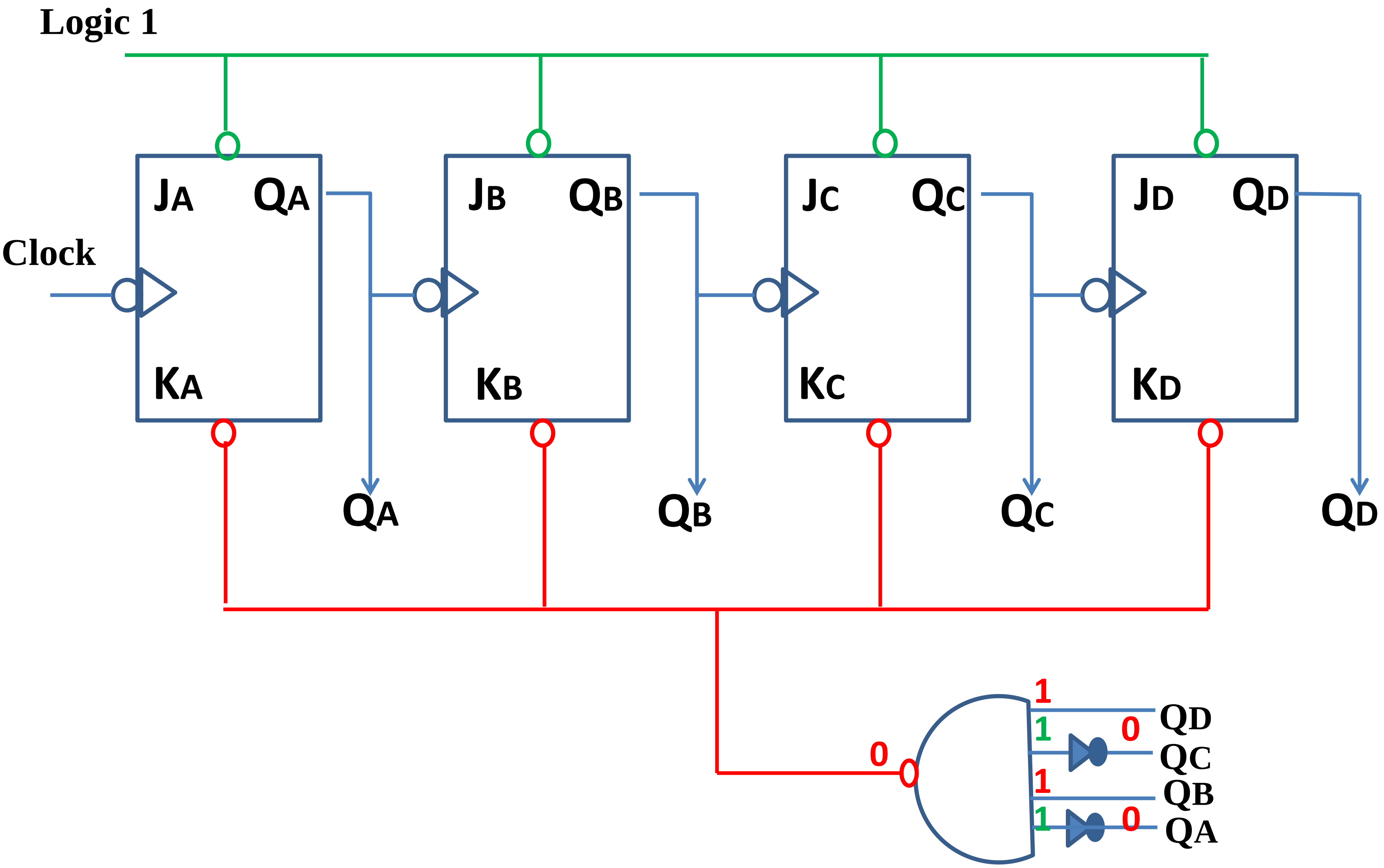
PR = 0 O/P = 1 ← SET

CLR = 0 O/P = 0 ← RESET

- We have to use NAND gate to achieve this



BCD Counter (Contd..)



Modulus of Counter (IC 7490)

- **2 – Bit** Ripple Counter is called as **MOD - 4** Counter or **Modulus 4** Counter

- **3 – Bit** Ripple Counter is called as **MOD – 8** Counter or **Modulus 8** Counter

n = No. of bits

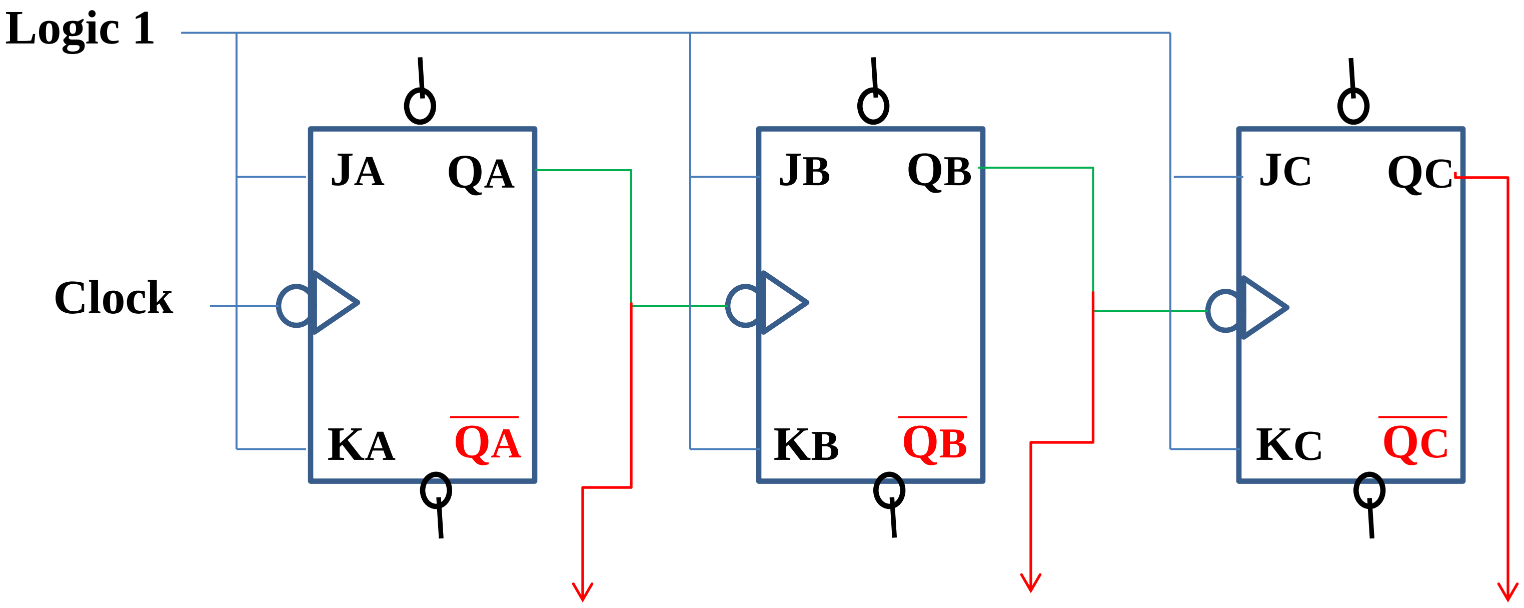
State / MOD No. = 2^n

Modulus of Counter= No. of States

MOD -6 Counter Using MOD 8 Counter

- MOD – 6 Counter
- MOD – 8 Counter

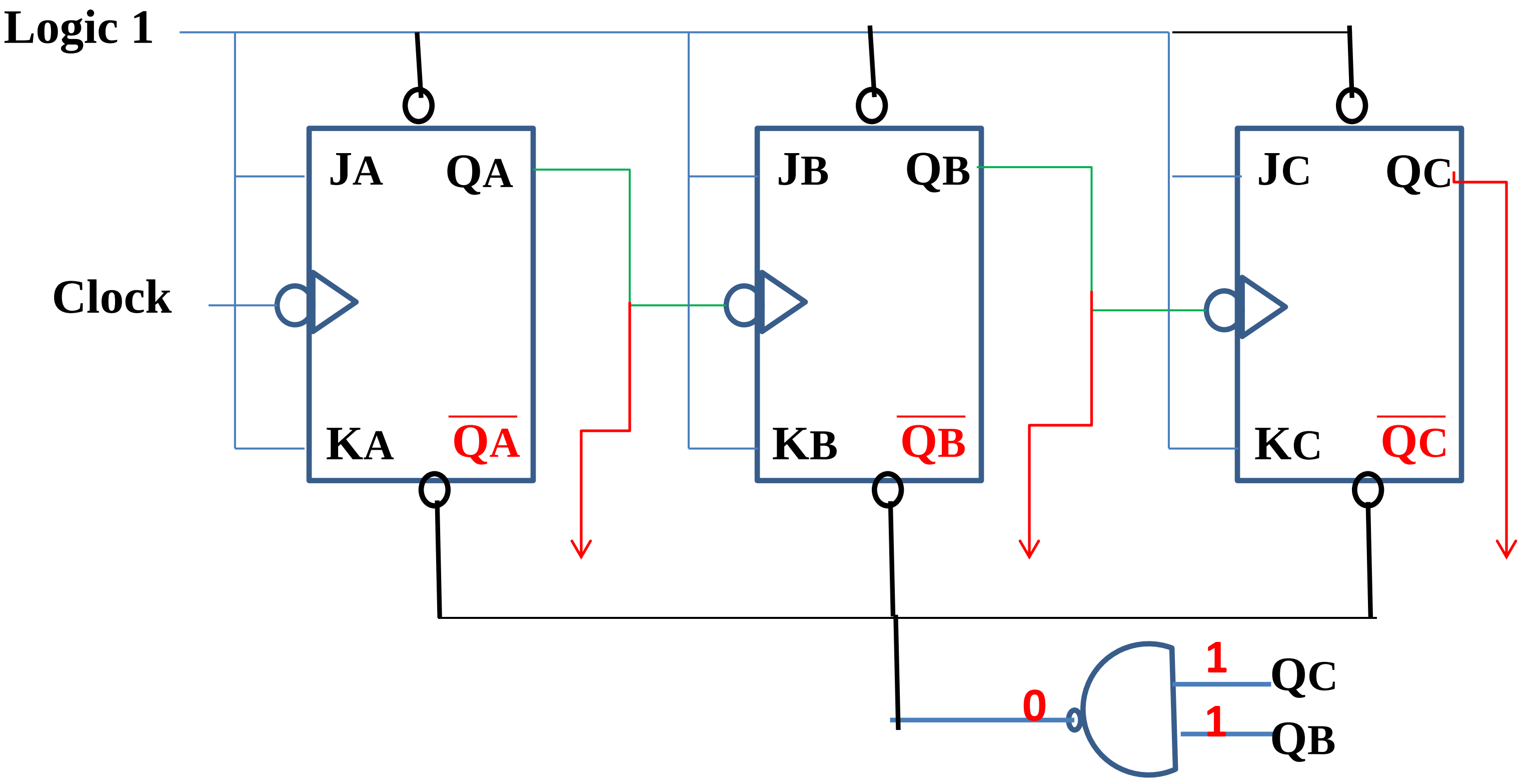
MOD -6 Counter :
000
001
010
011
100
101



MOD -6 Counter Using MOD 8 Counter (Contd..)

- If PR = 0 O/P = 1
- If CLR = 0 O/P = 0

- Now in what way we can use this PR & CLR I/Ps to count up to 5 instead of 7?


Logic 1
Clock
JA
QA
KA
$\overline{QA}$
JB
QB
KB
$\overline{QB}$
JC
QC
KC
$\overline{QC}$
0
1
QC
1
QB