Sequential circuits (also contains feed back path)
↓
Combinational circuits + Memory elements.

Memory elements can be latches / flipflops.
(can store 1 bit either 1/0)
latches → do not contain clock.
Flip flops → contain a clock.

→ Differences.

latches

o/p ~~at any instance~~ depends on i/p ~~only~~.
(no clock required)

→ Combinational

o/p at any instance depends on only i/p.

Memory unit is not required

Faster

Easy to design

Parallel adder

flipflops.

o/p ~~at any instance~~ depends on clock as well as i/p.

Sequential -

o/p depends on i/p & also on past history of these inputs.

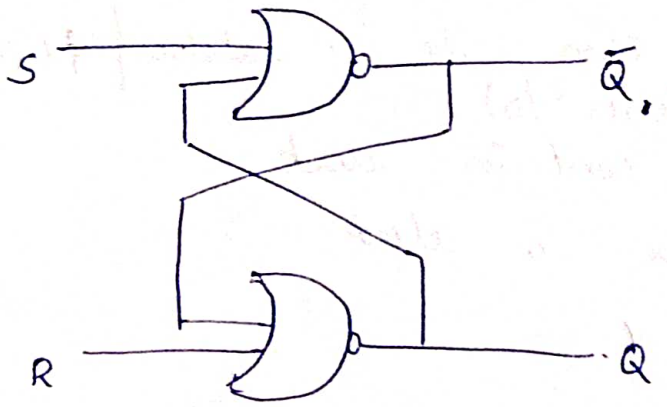
Memory unit is required

Slower.

Difficult:

Serial adder.

SR LATCH (nor gate)



i, S = 0 R = 0.

- Let us assume Q to be 1,

0 , 1 → $\bar{Q}$ will be 0

S → set R → reset

If we get Q = 0 → reset

Q = 1 → set.

S	R	Outputs Q	$\bar{Q}$
0	0	No change.	
0	1	0	1
1	0	1	0
1	1	X (invalid)	X

To get more clarity about no change state implement it after any state.

ii, S = 1 R = 0.

S = 1 → output at $\bar{Q}$ = 0.

0
0 → 1 ∴ Q = 1 $\bar{Q}$ = 0.

Set state.

Now check for no change (memory state)

S = 0 R = 0.

Now as Q = 1.

1 nor 0 will be 0.

$\therefore \bar{Q} = 0$.

Similarly, As. Take $\bar{Q} = 0$ initially.

0 nor 0 will give 1.

$Q = 1$.

$\therefore S = 0, R = 0$ is ~~~~ no change state

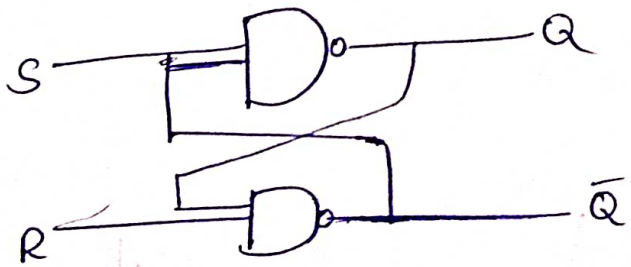
Similarly in other cases,

$S = 1, R = 1$ will be invalid case.

$S = 1$ gives $\bar{Q} = 1$ } → can't happen possibly.

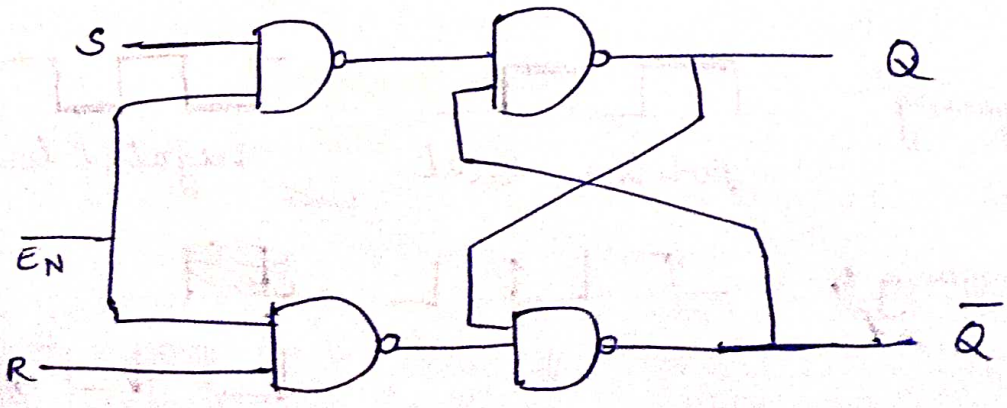
$R = 1$ gives $Q = 1$ }

Similarly using nand gates.



S	R	Q	$\bar{Q}$
0	0	No change	
0	1	0	1
1	0	1	0
1	1	Invalid.	

Gated SR Latch

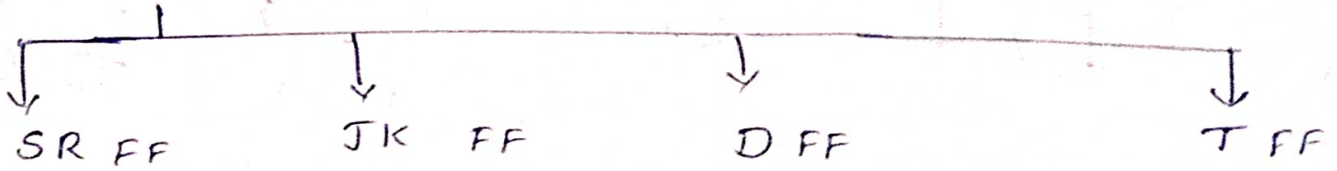


Latche works only if $E = 1$.

E	S	R	Q_n	$\bar{Q}_{n+1}$	State
0	✗	✗	0	0	No change.
0	✗	✗	1	1	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set.
1	1	0	1	1	
1	1	1	0	x	Indeterminate.
1	1	1	1	x	

Latch ⟶ Similar to ff but with 1 extra clock.

ff



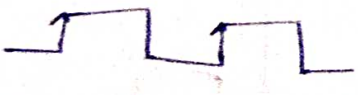
Triggering

Level triggering

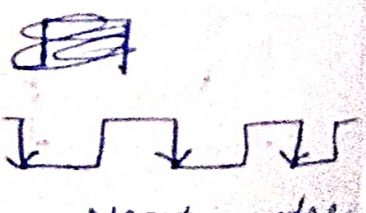
positive level

negative level

Edge triggering

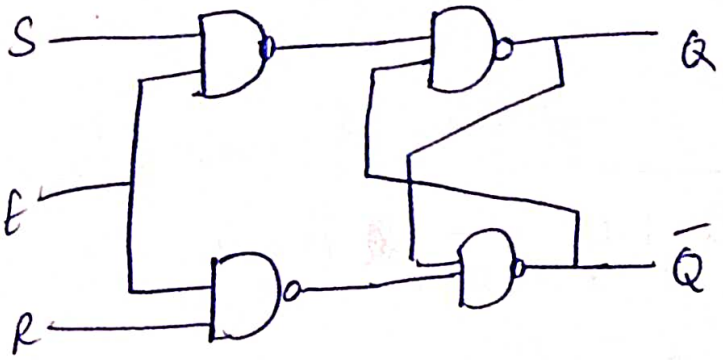


Positive edge



Negative edge

SR FF



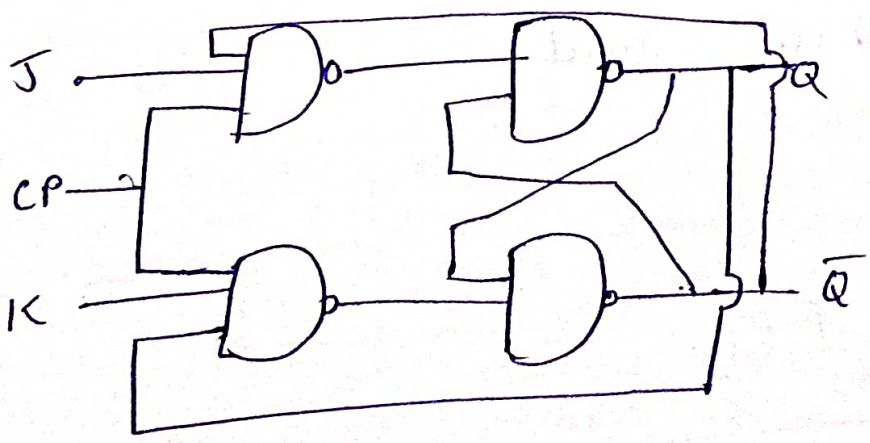
(Gated SR FF)

D FF

Not gate used for input

CP	D	Q
0	x	No change
1	0	0
1	1	1

Edge Triggered JK flip flop



C = 1.

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'

toggle.

$\bar{J} = 0$ $K = 0$.

$J = 0$ $C = 1$ $\bar{Q}$ $\longrightarrow$ $\overline{\bar{Q} \cdot 1 \cdot 0} = \bar{0} = 1$

$\overline{1 \cdot \bar{Q}} = \bar{\bar{Q}}$

$K = 0$ $C = 1$ Q $\overline{Q \cdot 1 \cdot 0}$ $= Q\bar{0} = 1$

$\overline{\bar{\bar{Q}} \cdot 1} = \bar{Q}$.

No change.

$\bar{J} = 0$ $K = 1$

$J = 0$ $C = 0$ $\longrightarrow$ Q.

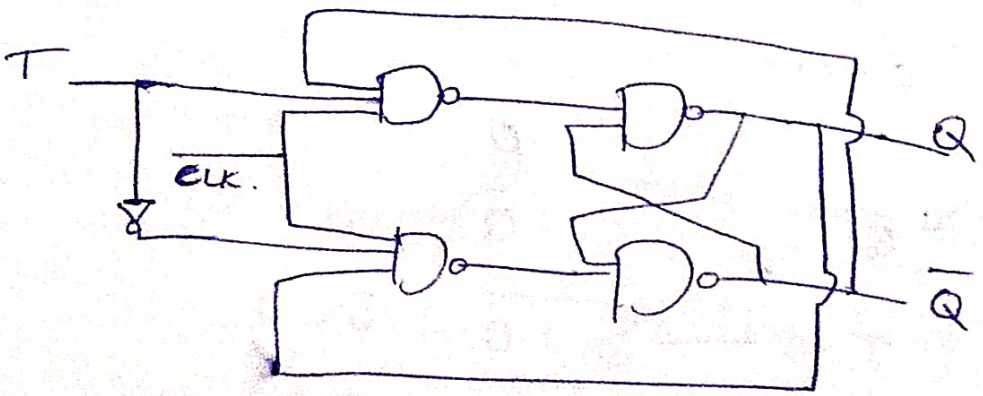
$K = 1$ $C = 1$ Q $= \bar{Q}$.

$\overline{\bar{Q} \cdot Q} = Q\bar{Q} = 0$

$\bar{Q} = 1 \rightarrow Q = 0$ reset.

Similarly others.

Edge Triggered T FF:



T	Q_{n+1}	
0	Q_n	No change
1	Q'_n	Toggle.

Race around condition

J=1 K=1 → JK.

Toggle.

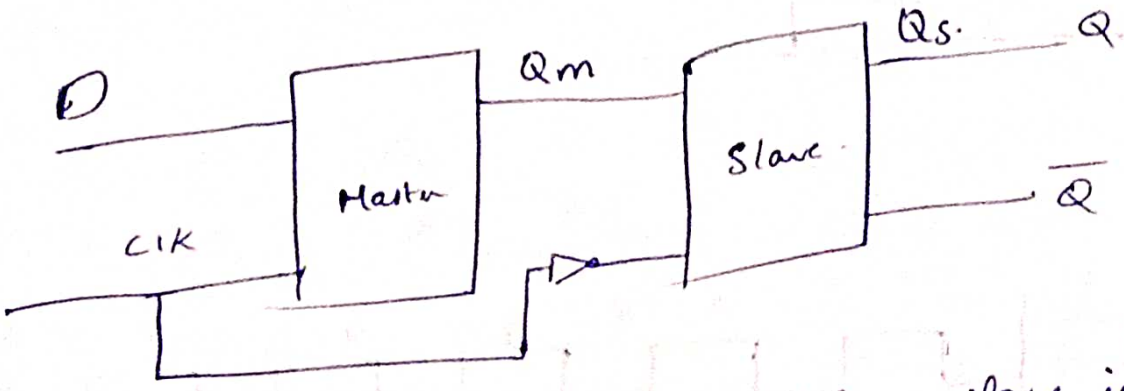
J=1 K=1

C=1 in JK

To avoid this we go for M-S FF

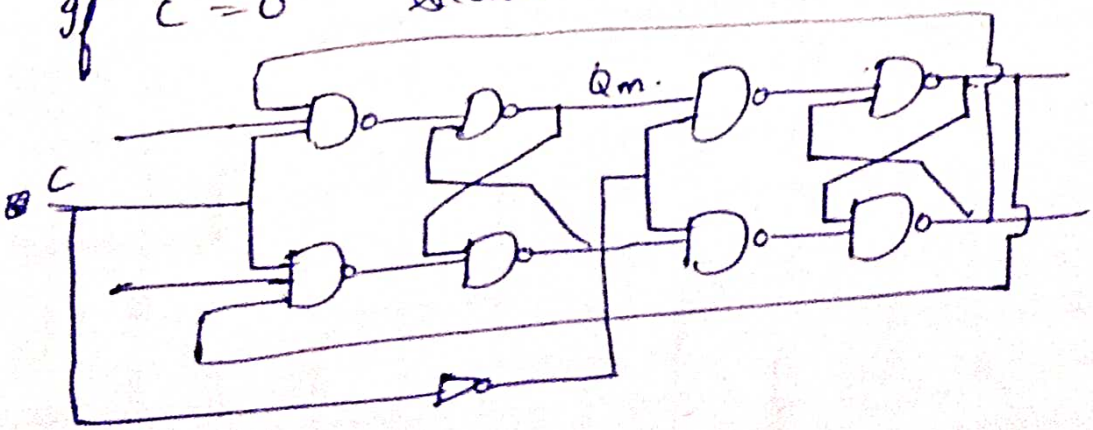
Master Slave FF

we can use D, T, JK, SR FF's



if C = 1 Master is active, slave is inactive

if C = 0 Slave " " , Master " "



Conversion of FF to other.

Convert SR to D.

→ Known SR unknown D.

Write Exitation table of SR.
" Characteristics " " D.

Exitation table SR

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

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Char of D

D	Q
0	0
1	1

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

Q_n	Q_{n+1}	S	R	D
0	0	0	x	0
0	1	1	0	1
1	0	0	1	0
1	1	x	0	1

S

Q_p \ D	0	1
0	0	1
1	0	x

$\bar{Q}_p D + Q_p D$

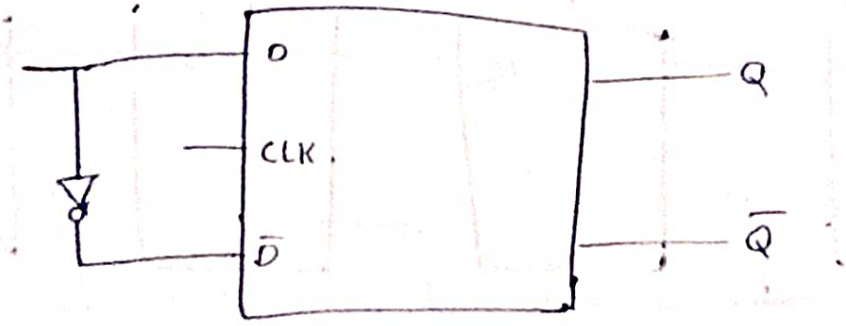
D

R

Q_p \ D	0	1
0	x	0
1	1	0

$Q_p \bar{D} + Q_p D$

$\bar{D}$



SHIFT REGISTERS

Reg → grp of ff's.

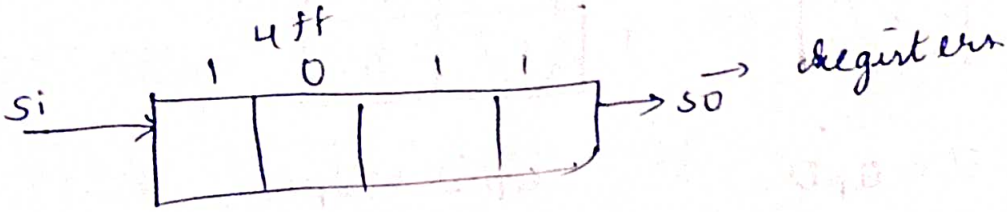
↓ to store more no. of bits.

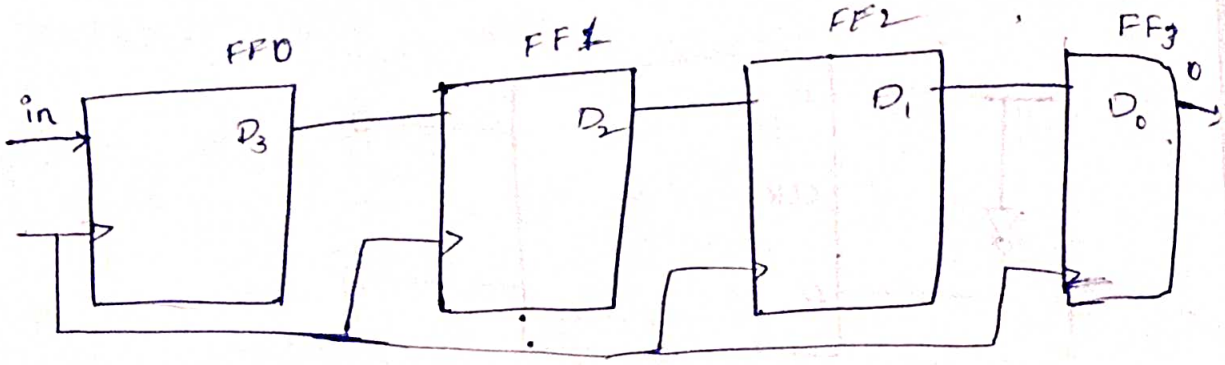
Types →
- SISO
- SIPO
- PISO
- PIPO
- BISR
- USR.

SISO

o/p → series.

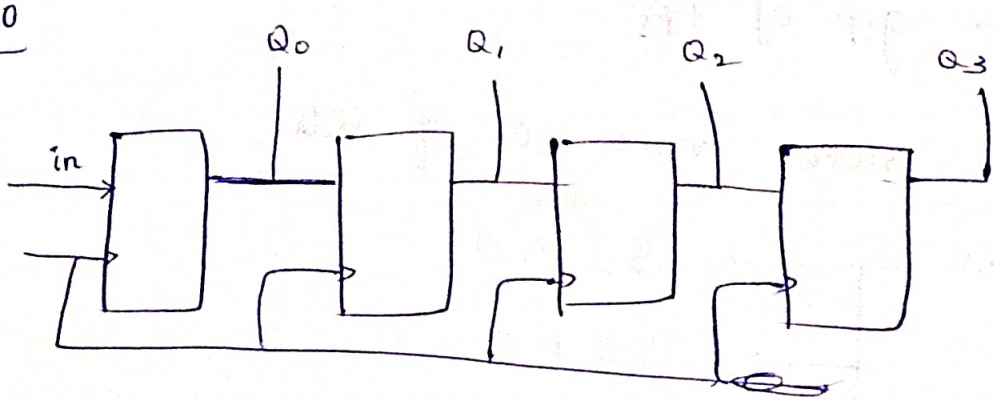
I/p → series





1011 → ip → after 4 clock pulses MSB reaches to FF3.

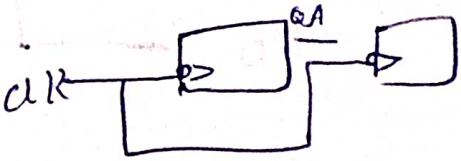
SIPO



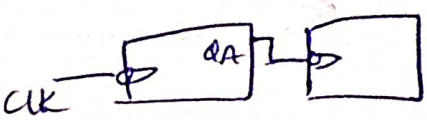
1011 → LSB → Q_0

CLK	I_n	Q_0	Q_1	Q_2	Q_3
0	x	0	0	0	0
1	1	1	0	0	0
2	1	1	1	0	0
3	0	0	1	1	0
4	1	1	0	1	1

Synchronous → same clock pulse.

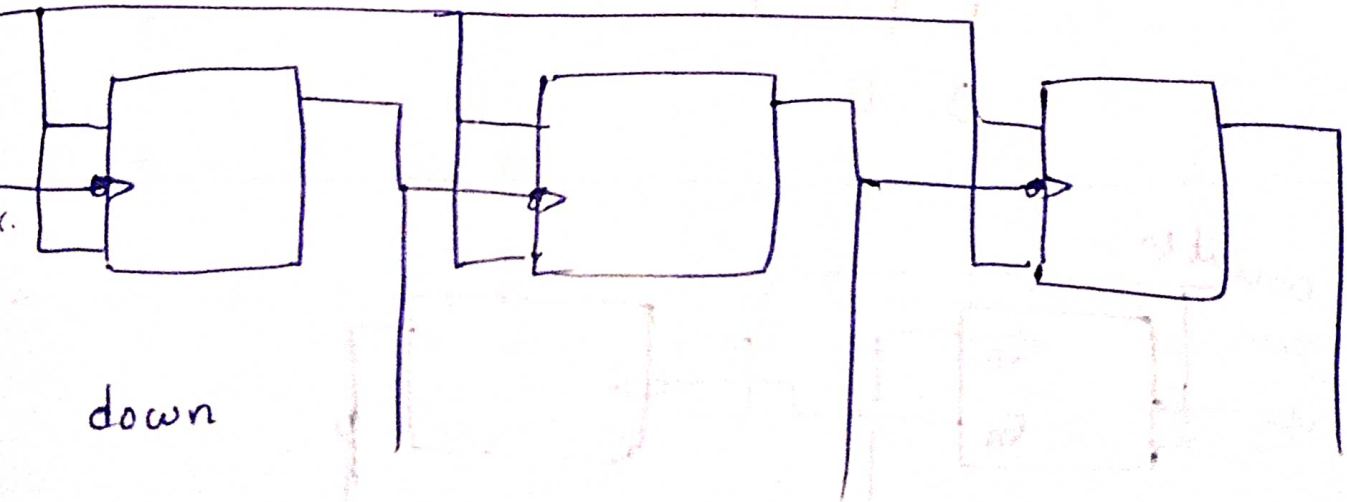


Asynchronous → diff clock.

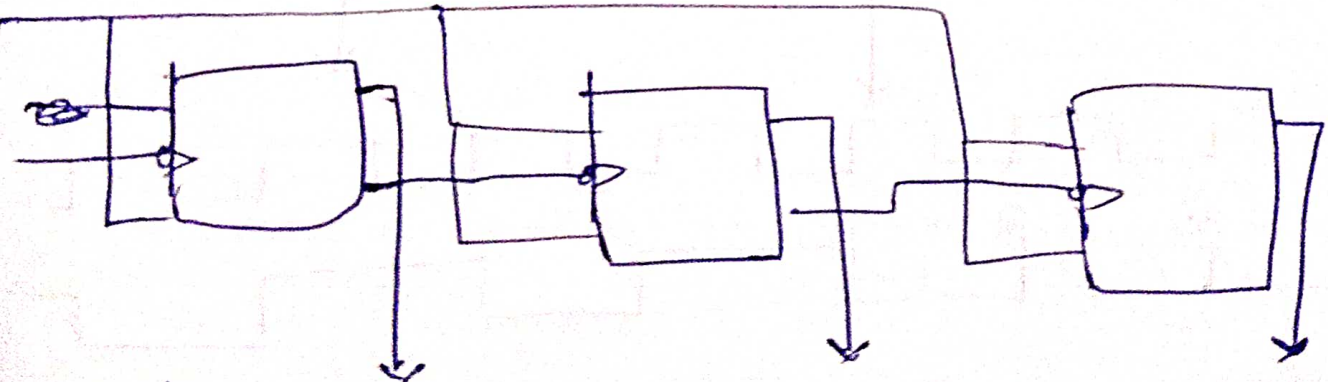


3bit Asynchronous.

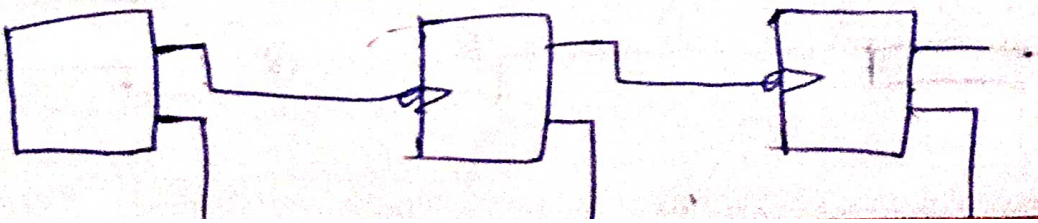
up



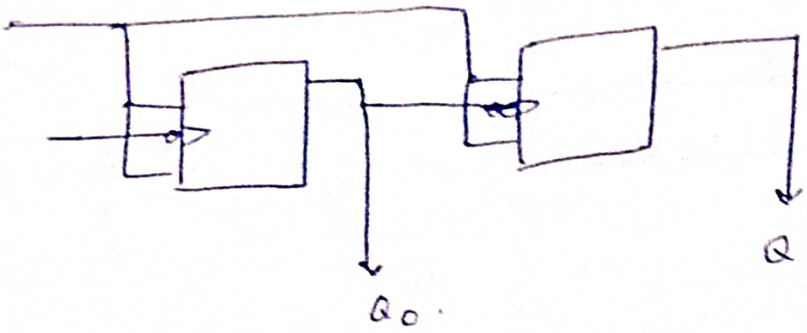
down

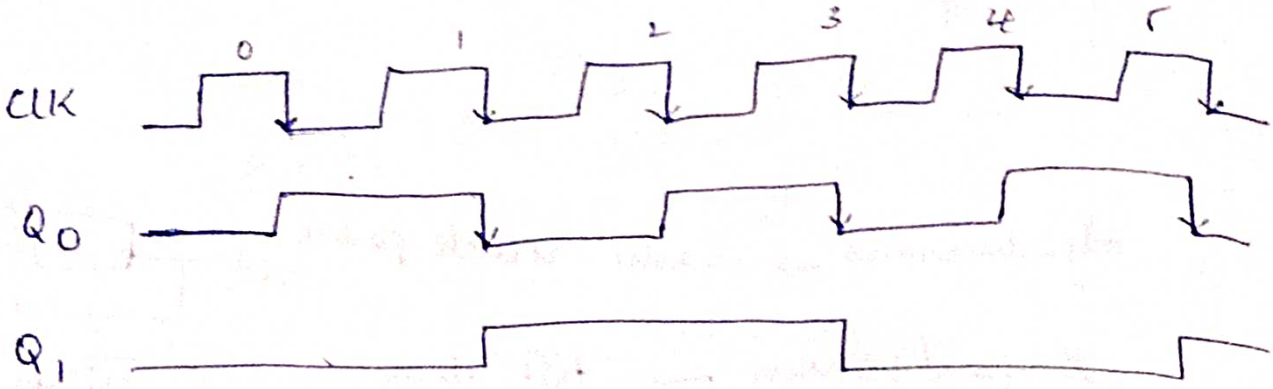


down



Mod 4 up counter (Asynchronous)



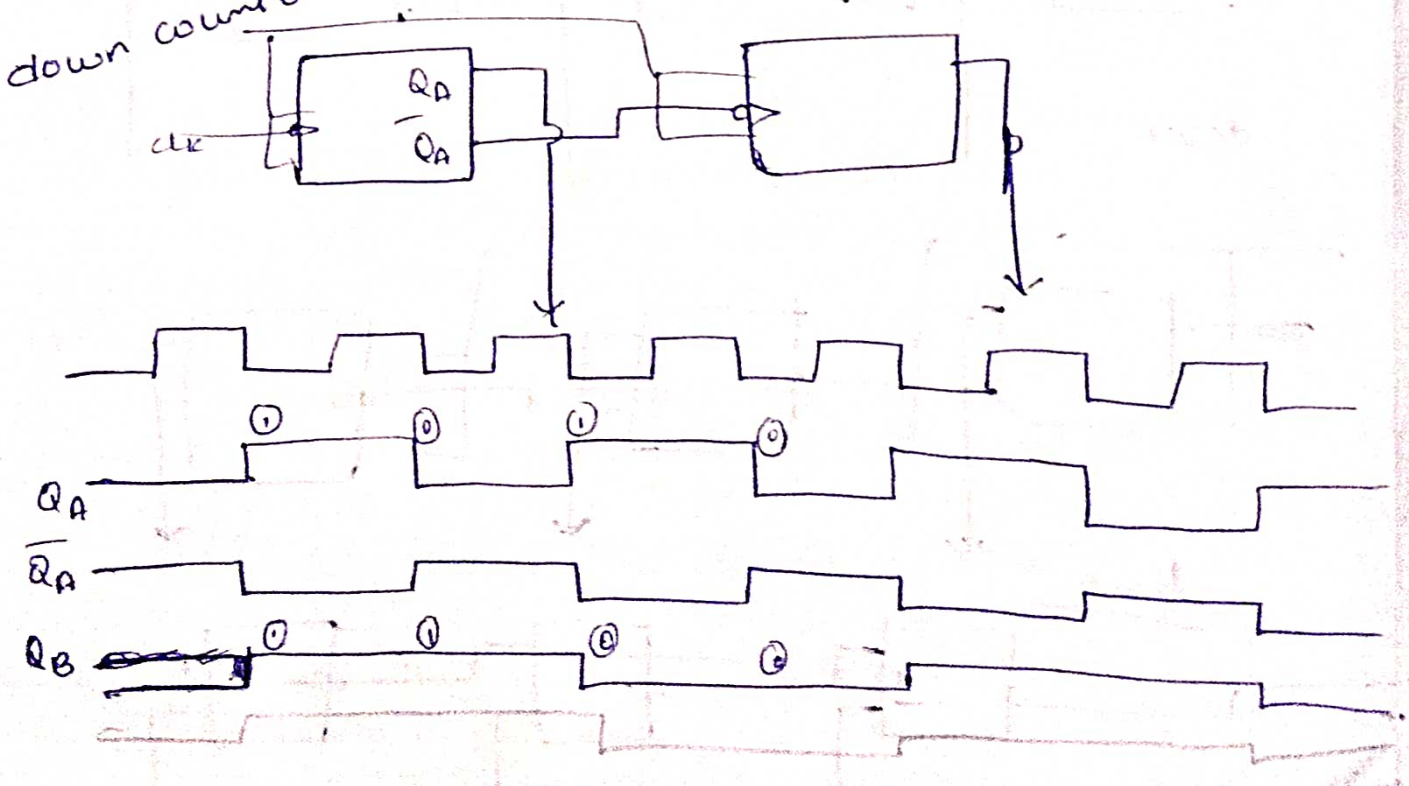


0 → 0 0
0 1
1 0
1 1
0 0

} 1 cycle

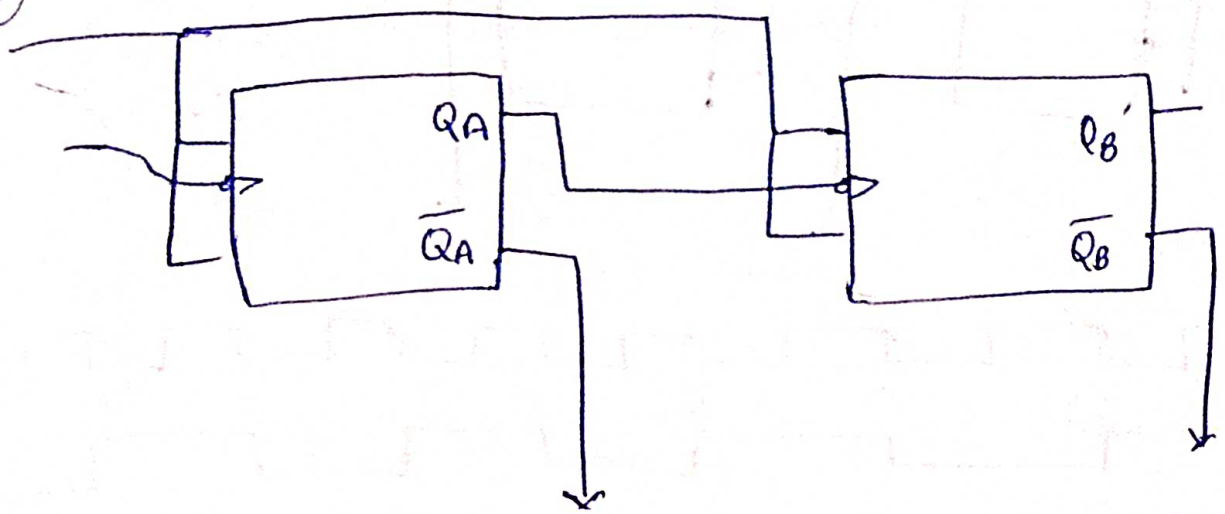
4 mod.

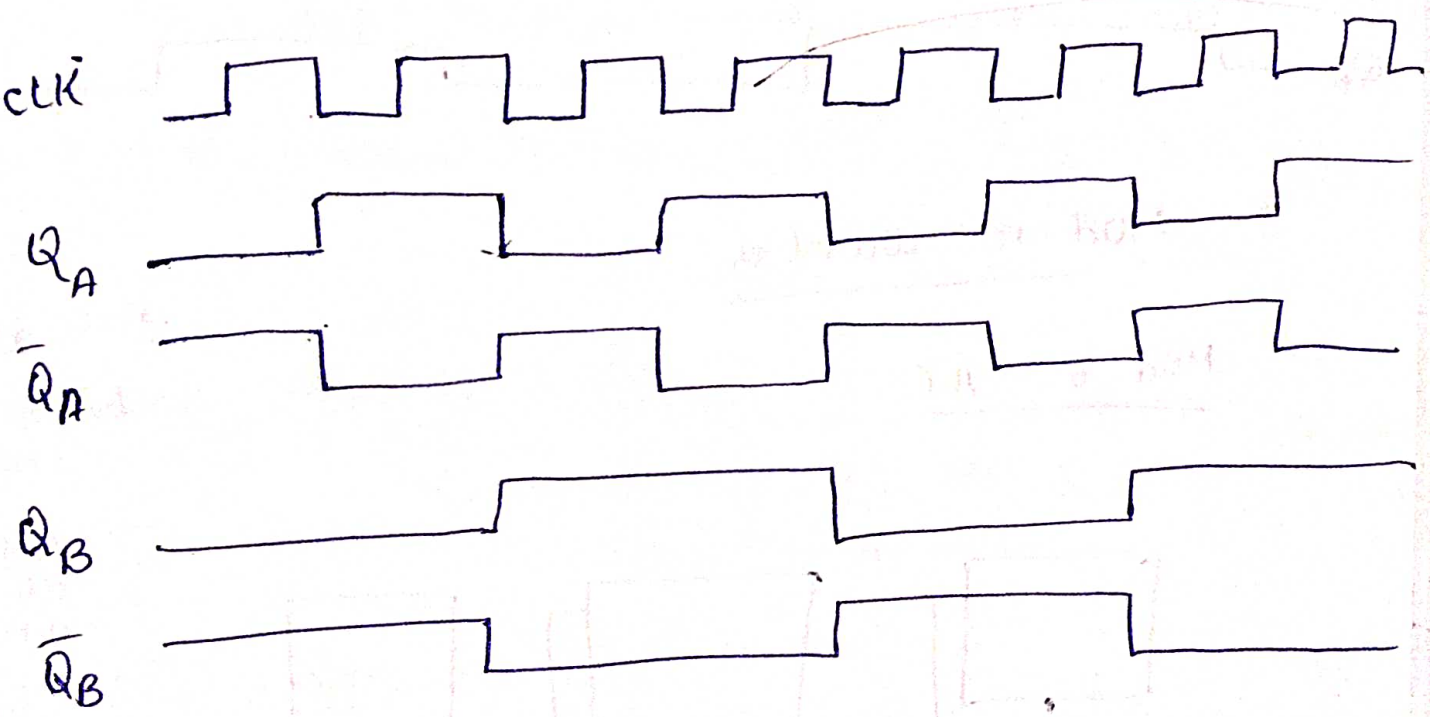
down counter.



Mod 4 down counter.

(1)





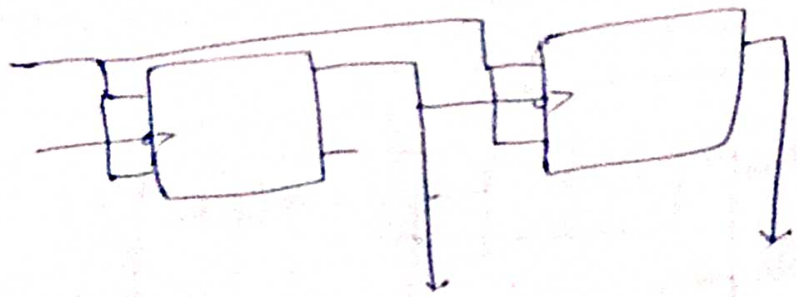
Output is $\bar{Q}_B$ Q_A.

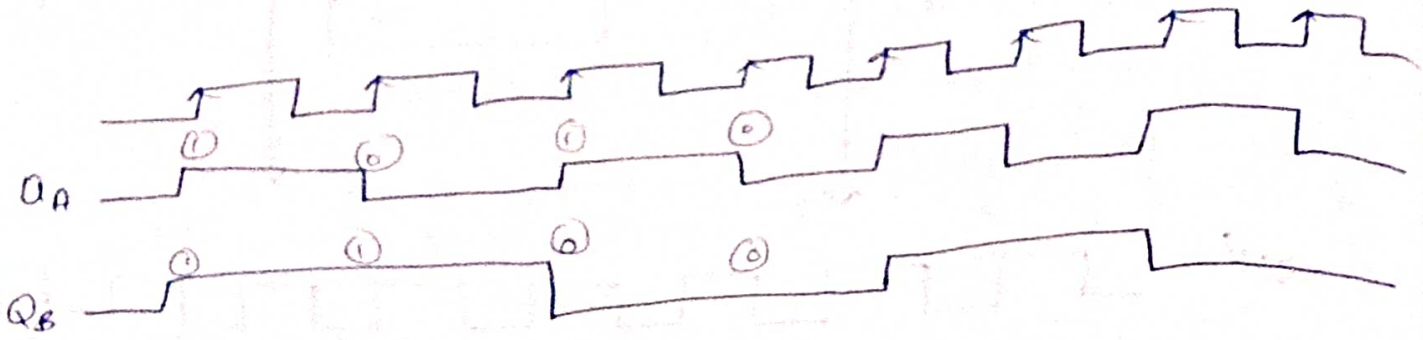
1	1
1	0
0	1
0	0
1	1

} mod 4 down counter

· DOWN

③





Mod-n counter

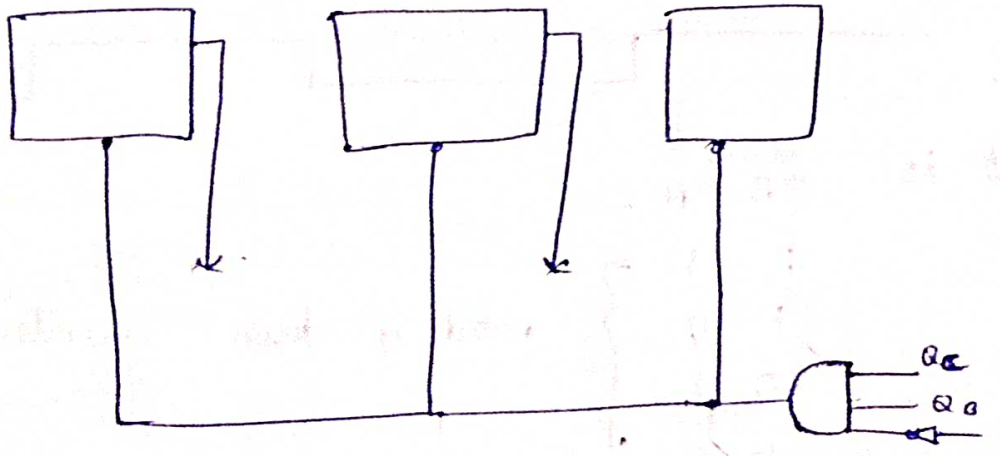
③ → 2^3 = 8 ff

but we want only 6

000 0
001 1
010 2
011 3
100 4
101 5
000 6

110 } X.
111 }

11X



Mod 6 A.C.

Clr → 0 then Q = 0

Preset = 0 then Q = 1.

upto

0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1 → as Clr = 0.

then 0 0 0. Q will be 0

10 10

Design . Synchronous counter

① Decide no of ffi
② Exitation table of ff
③ State diagram & circuit exitation table
④ Simplified exp using K Maps.
⑤ Draw logic diagram.

Mod 8 Synchronous counter using T ff

↓

3 ff

Exitation Table of T FF

T	Q
0	No change
1	Toggle.

T_n	T_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

T_n	T	T_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q_A	Q_B	Q_C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Q_A^*	Q_B^*	Q_C^*
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

T_{BC}	T_B	T_A
0	0	1
0	1	1
0	0	1
1	1	1
0	0	1
0	1	1
0	0	1
1	1	1

T_C

Q_C \ Q_AQ_B	00	01	11	10
0	0	0	1	0
1	0	0	1	0

T_B

Q_C \ Q_AQ_B	00	01	11	10
0	0	1	1	0
1	0	1	1	0

T_A

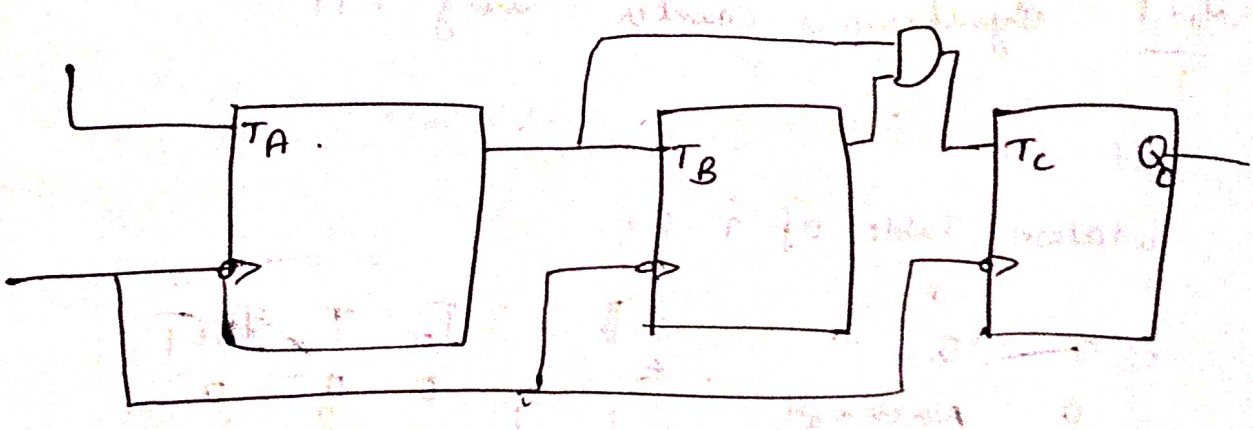
Q_C \ Q_AQ_B	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$T_C = \bar{Q}_C\, Q_A Q_B + Q_C Q_A Q_B$

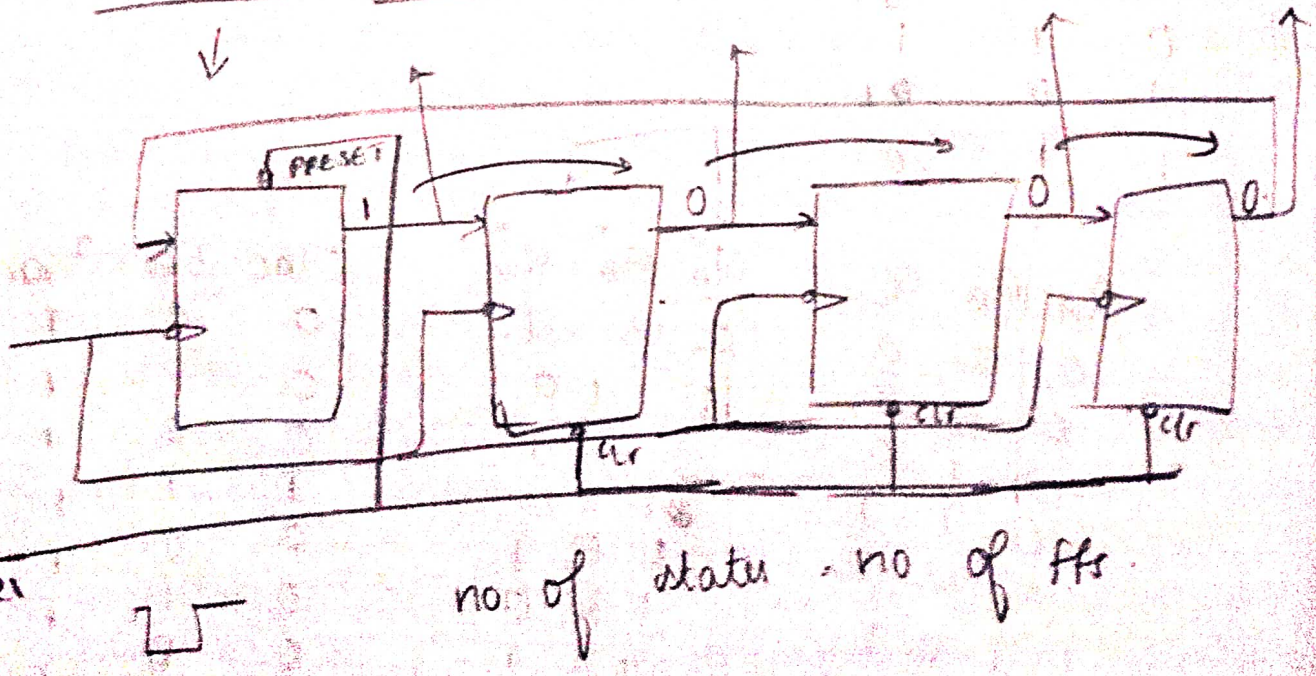
$= Q_A Q_B$

$T_B = Q_{B\_A}$.

$T_A = 0$.



RING COUNTERS



no. of states = no of ffs.

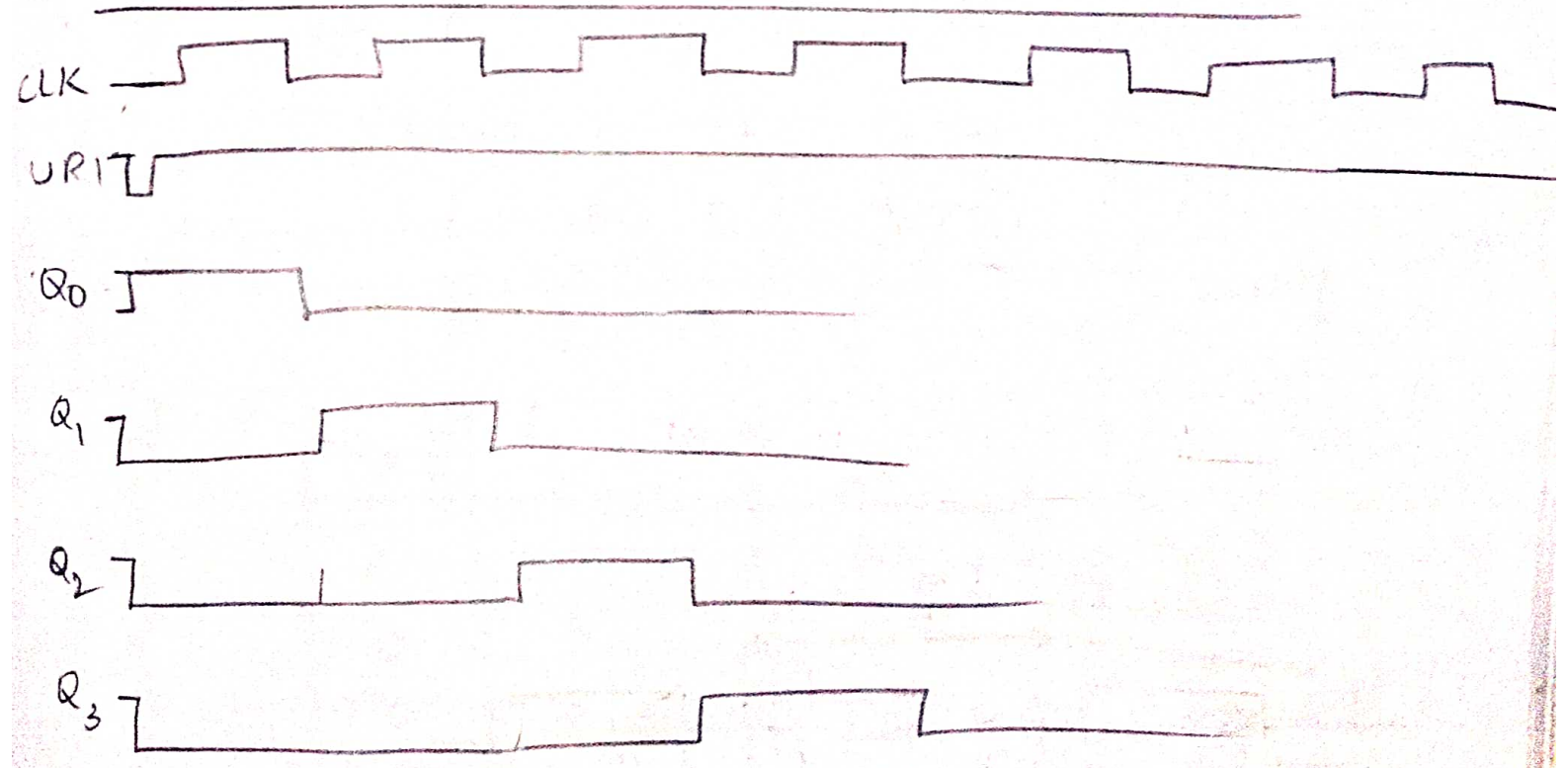
Pre 0 → Q = 1

Clr 0 → Q = 0

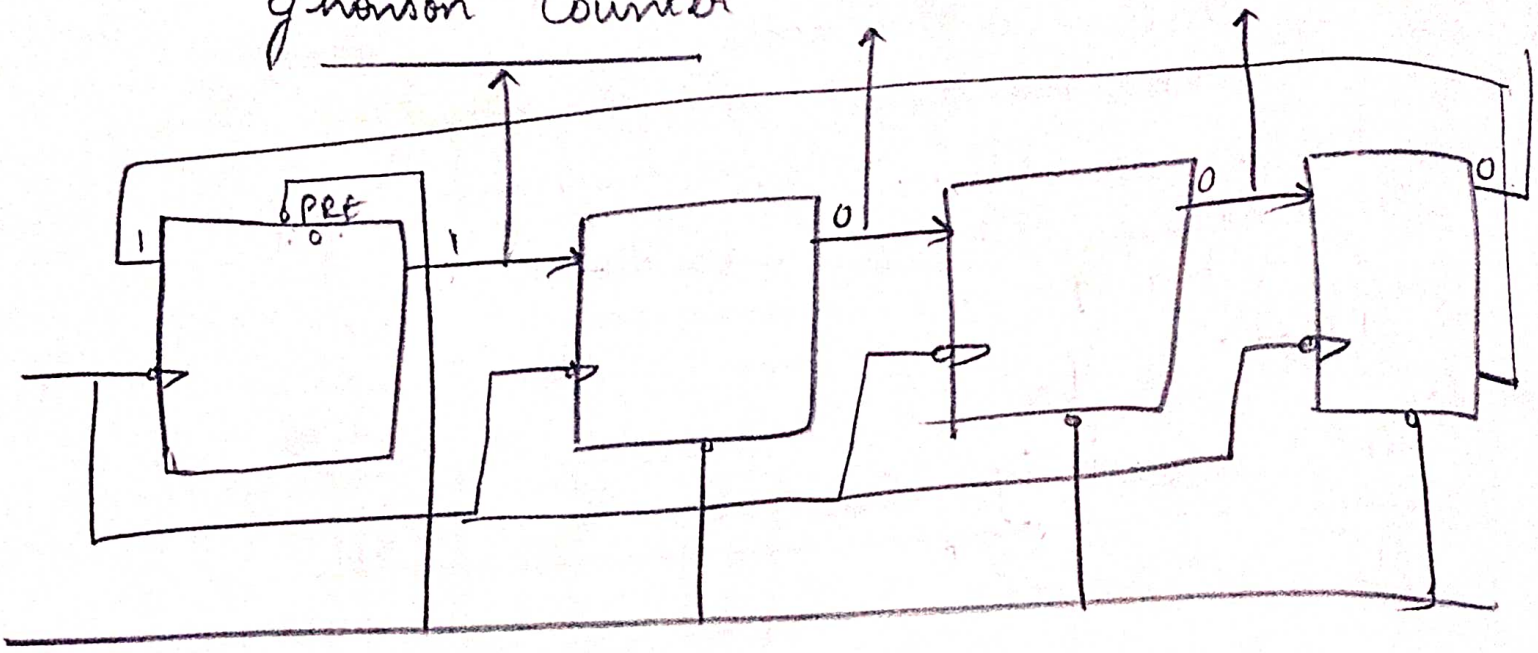
ORI low → Pr = 0 → Q = 1

shift

ORI	CLK	Q_0	Q_1	Q_2	Q_3
⊓	×	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0



Johnson Counter



ORI

1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0