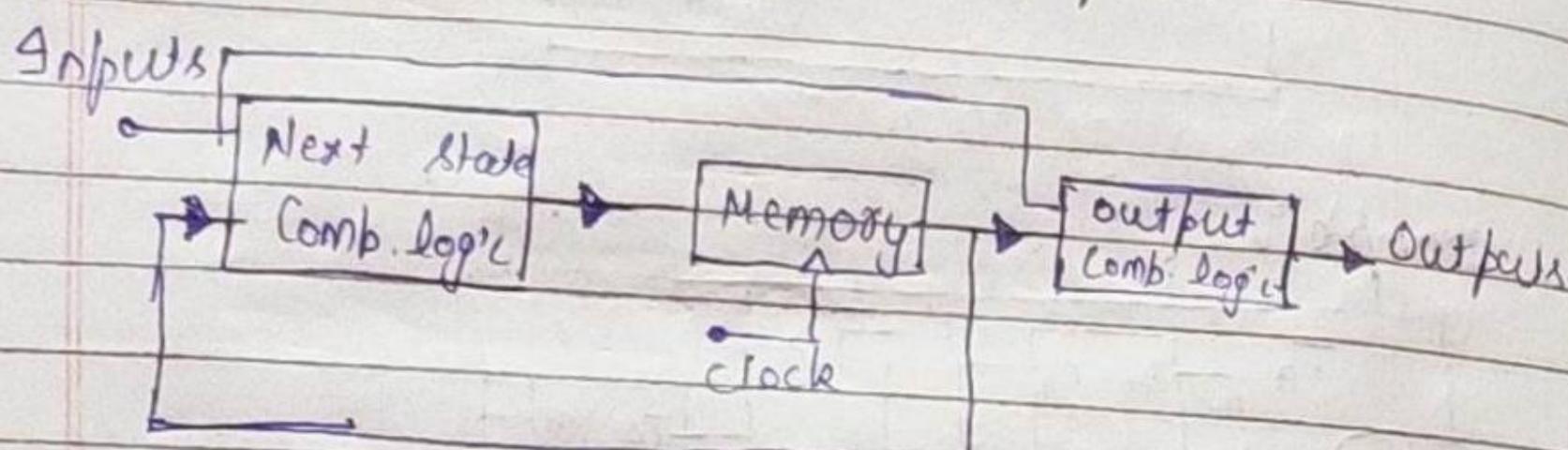


Mealy state machine :-

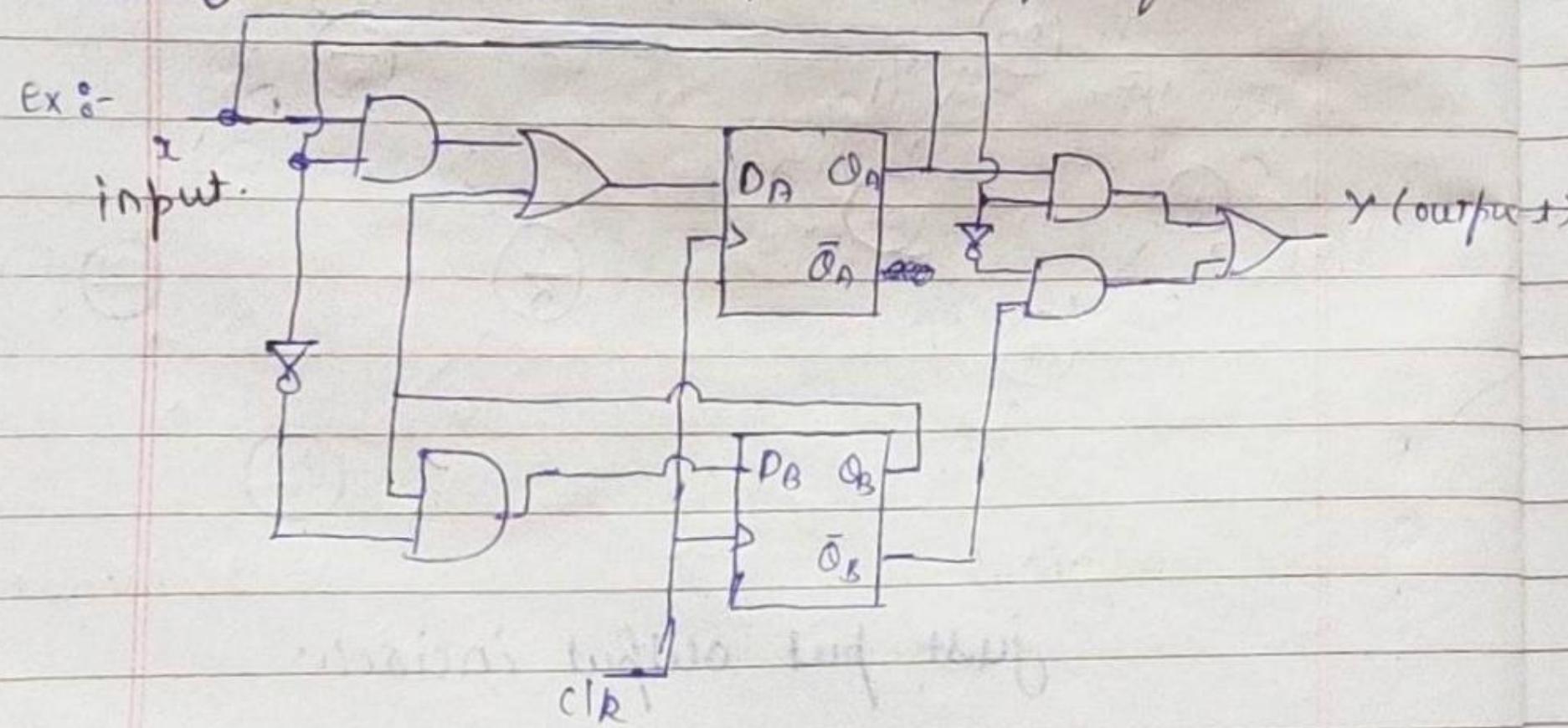
⇒ The output is the function of present state as well as the input.



In Mealy no. of states are less than Moore.

Analysis of clocked sequential circuits
(with flip-flops)

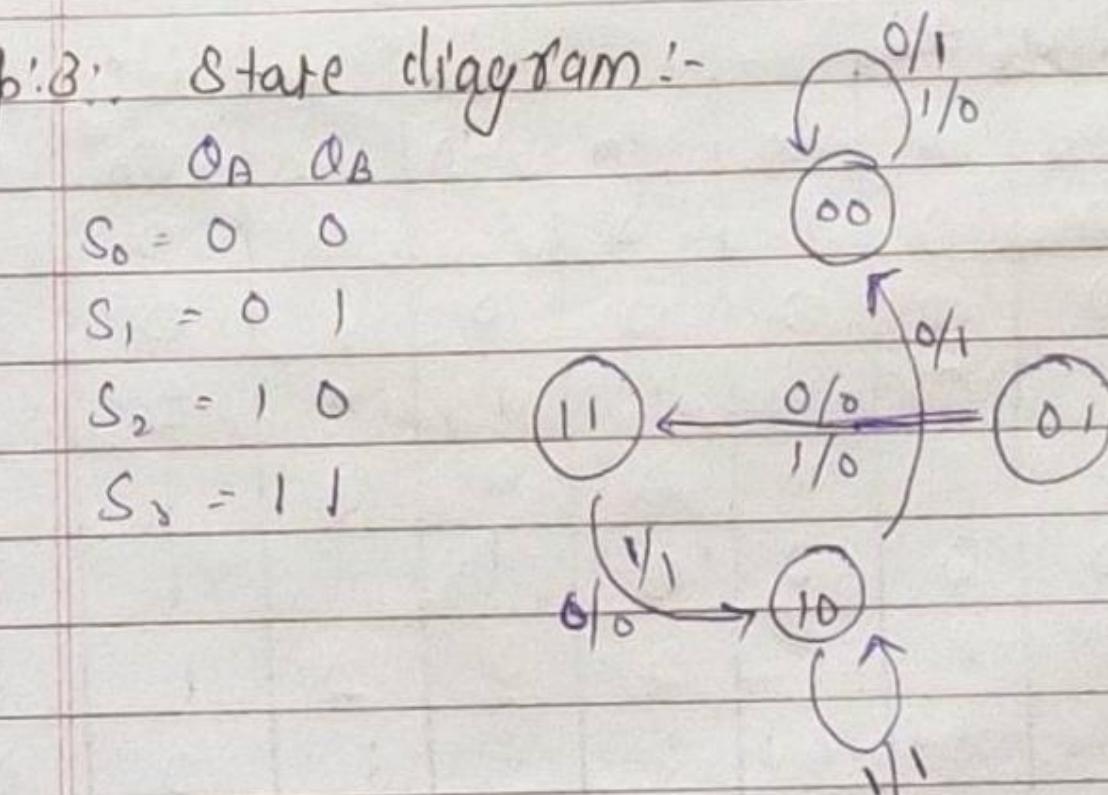
Step 1:- find out the s/p and o/p eqn.



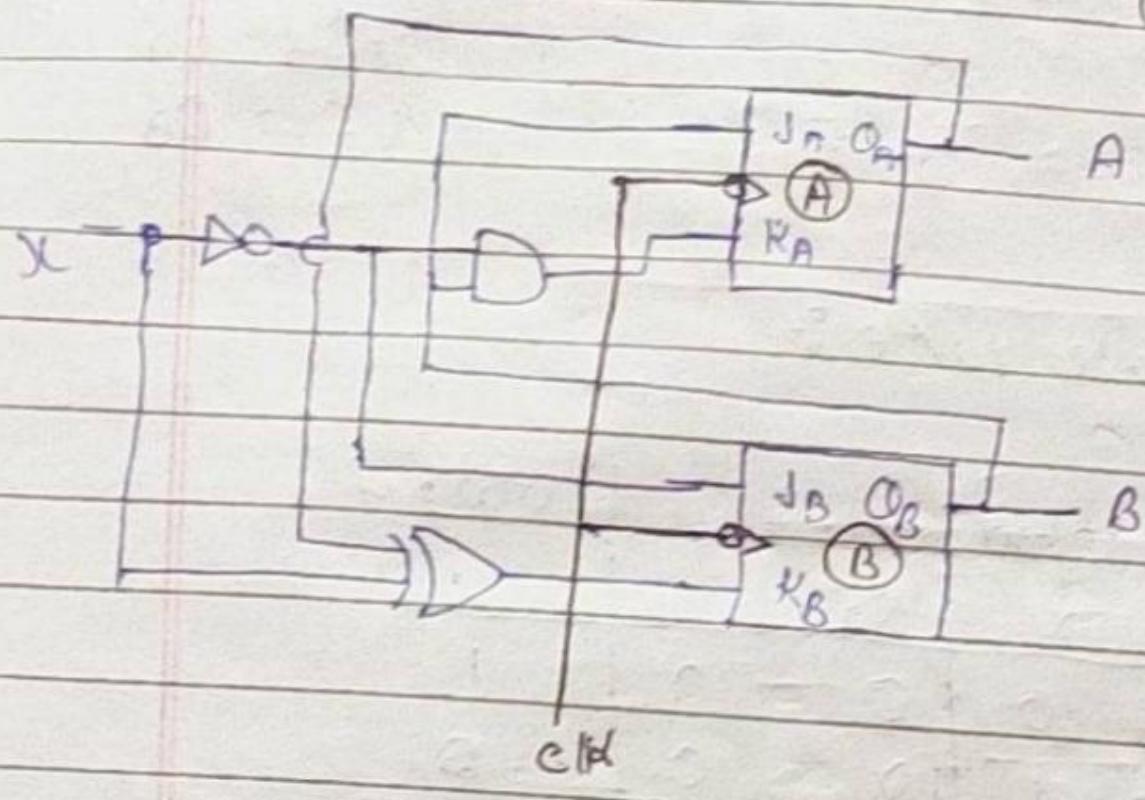
Step 2:- Make state table :-

P. S.		input	N. S.		output
Q_A	Q_B	x	Q_A'	Q_B'	y
0	0	0	0	0	1
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	0	1

Step 3:- State diagram :-



Analysis of clocked sequential Circuits
 with (JK flip-flop)



Step 1:- finding s/p egn

$$J_A = Q_B$$

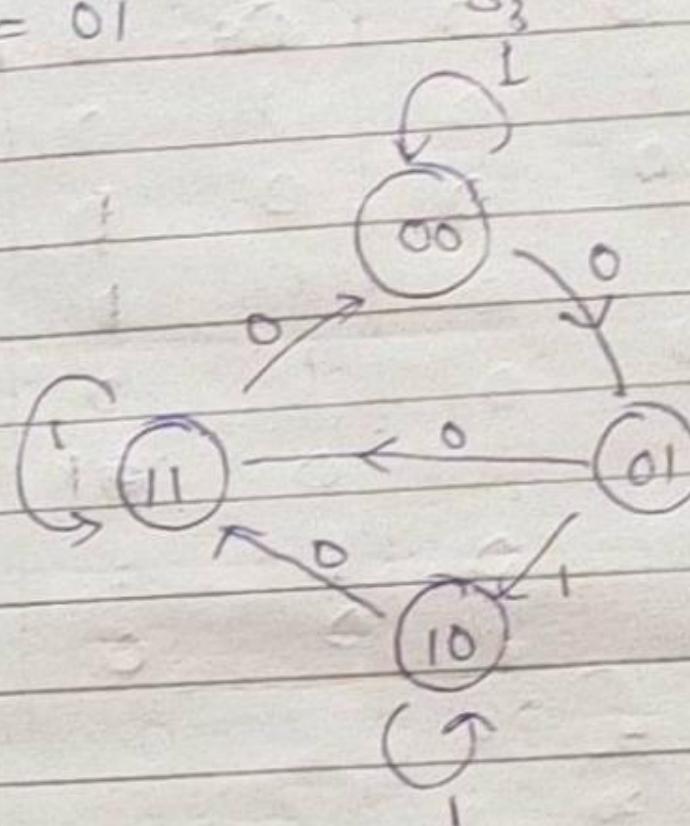
$$K_B = \bar{x} \cdot J_A$$

$$J_B = \bar{x}$$

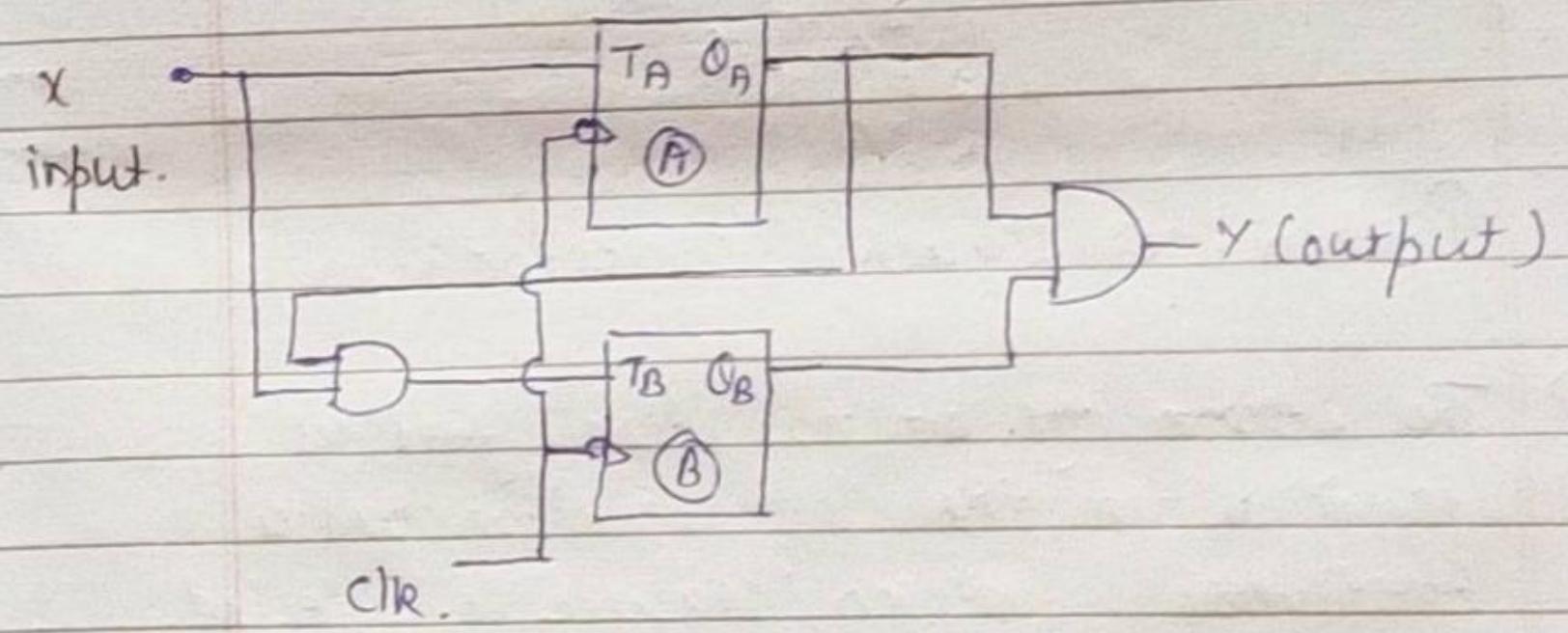
$$K_A = Q_A \oplus x$$

Present State		S/P	Next State					
Q_A	Q_B	x	J_A	K_A	J_B	K_B	Q_A*	Q_B*
0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	0
1	0	1	0	0	0	0	1	0
1	1	0	1	1	1	0	0	0
1	1	1	0	0	0	1	1	1

$$\begin{aligned} S_0 &= 00 \\ S_1 &= 01 \\ S_2 &= 10 \\ S_3 &= 11 \end{aligned}$$



Analysis of clocked sequential circuits
 with (T flip flop).

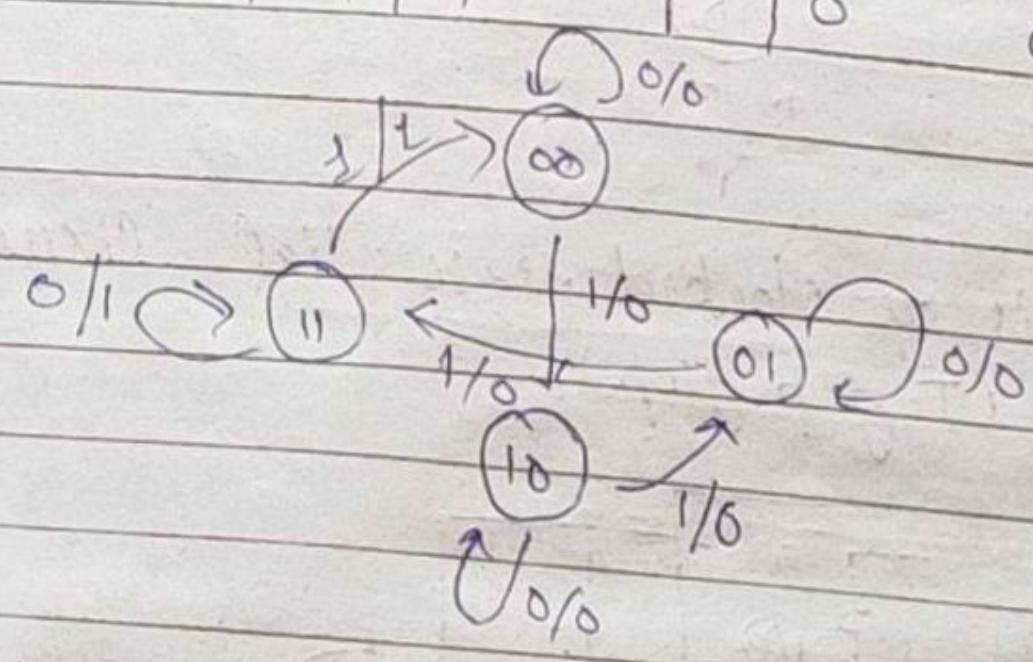


$$T_A = x$$

$$T_B = Q_A \cdot x$$

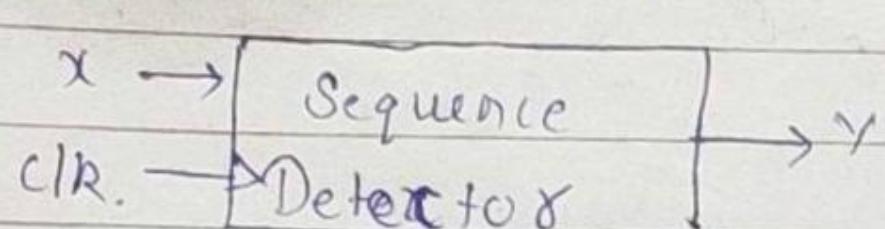
$$Y = Q_A \cdot Q_B$$

Present		%/P	x	T _A	T _B	(Q _A) ⁺	(Q _B) ⁺	O/P.
0	0	0	0	0	0	0	0	Y
0	0	1	1	1	0	1	0	0
0	1	0	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0
1	0	0	0	0	0	1	0	0
1	0	1	1	1	0	1	0	0
1	1	0	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1



Pattern or sequence Detector:-

- >> The stream of bit has been feed as input, when the clock is high and a particular pattern/sequence is detected.
- >> As soon as sequence is detected the output becomes high and then again becomes low.



Step 1:- State diagram (Mealy machine)

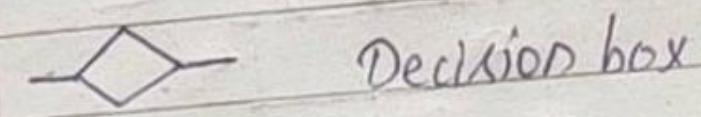
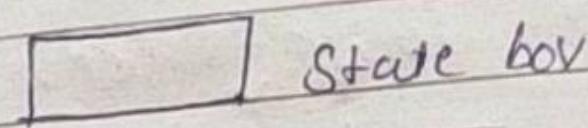
Set & Reset (Powerup)

Step 2:- State assignment.

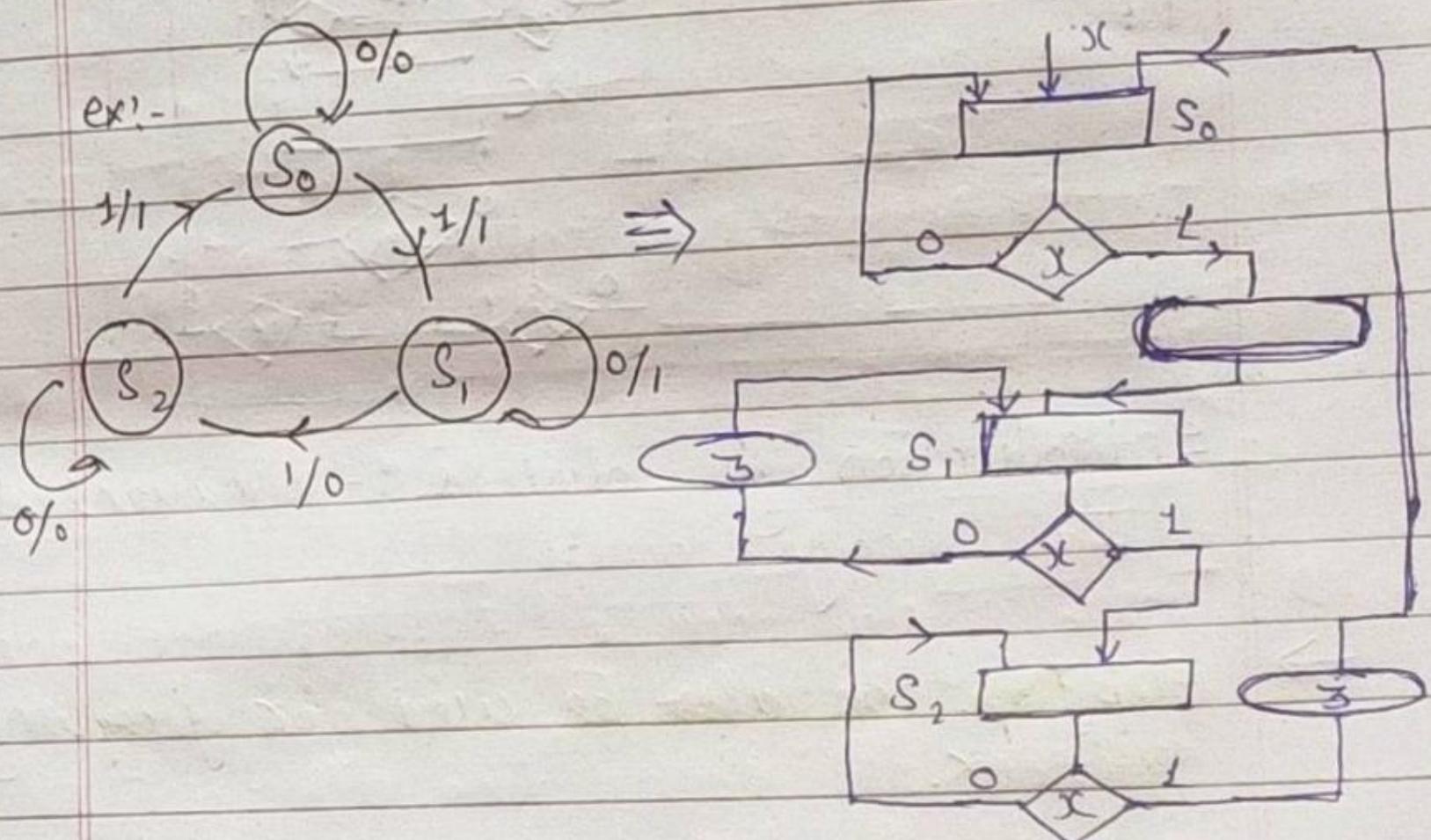
ASM chart

Algorithmic State Machine

enhanced version of state diagram:-)

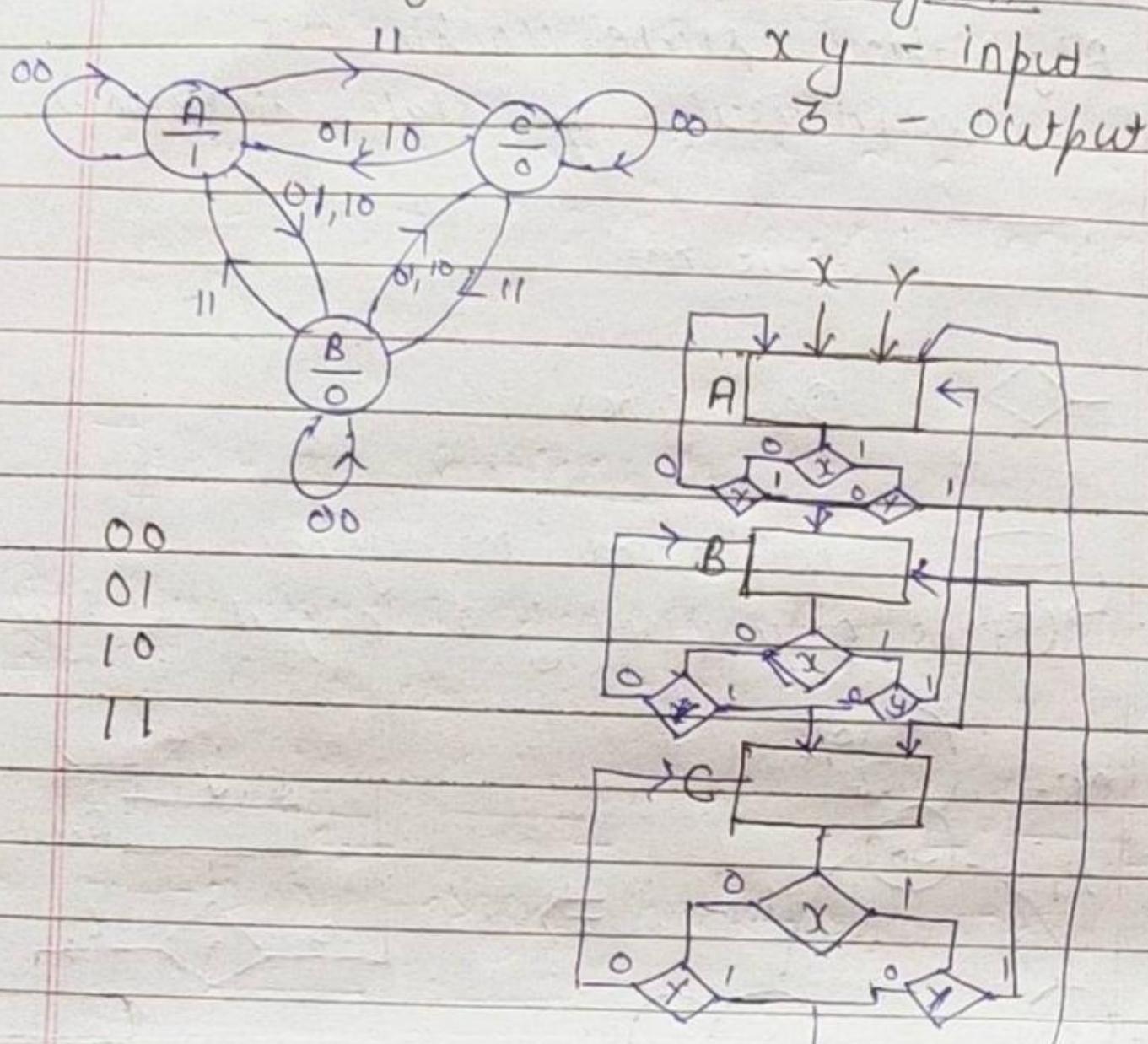


Condition box (mealy box).
(It is only used if the output is High)



Page No.:
Date: / /

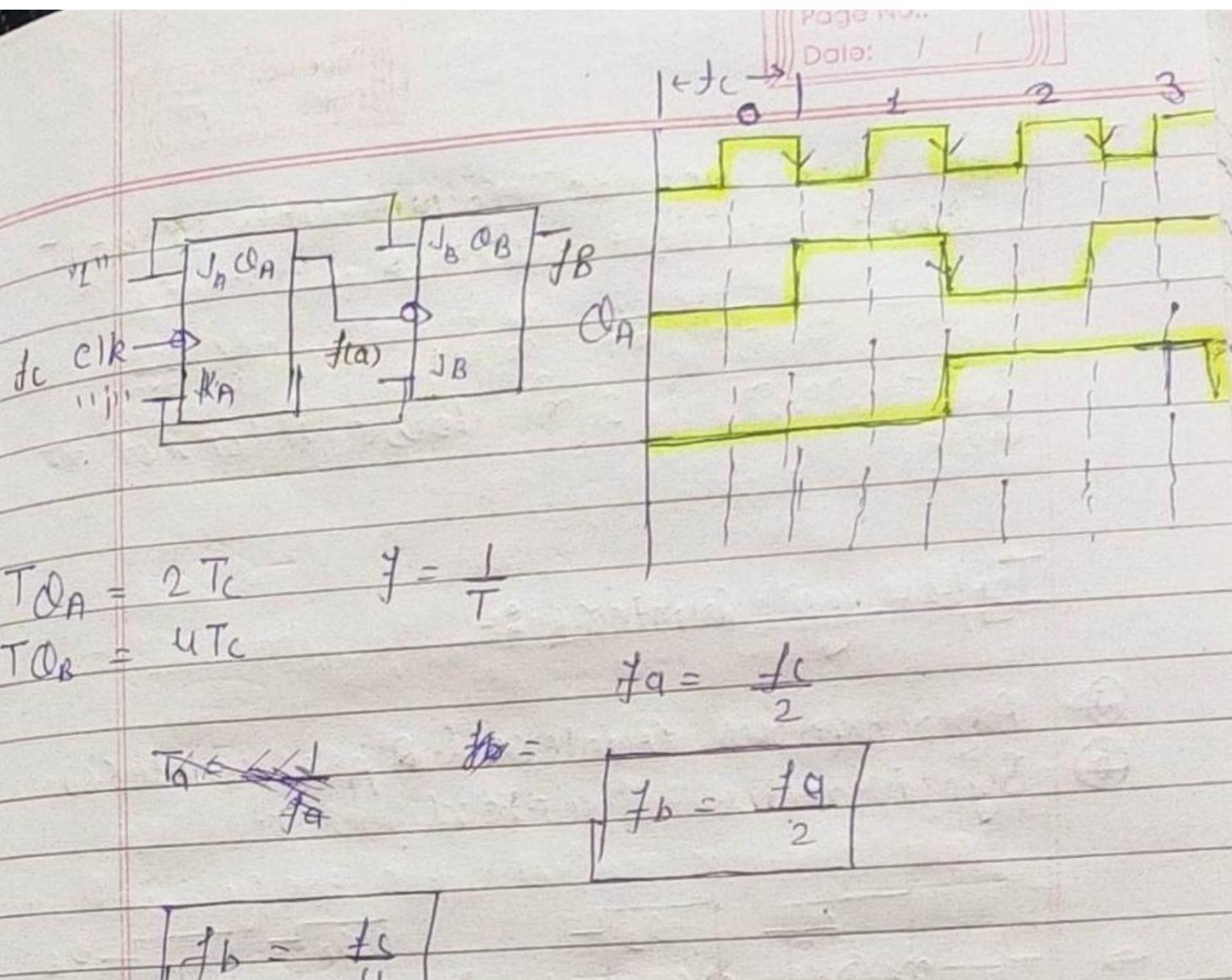
ASM chart for Moore state diagram :-



Introduction of counters :- (sequential)
It simply counts

⇒ Flip flop can also be used for frequency divider.

⇒ FF as divide by two ckt.



⇒ P no. of FF (where J, K = 1, and -ve edge triggered)
final frequency will be

$$f_{\text{final}} = \frac{f_{\text{initial}}}{2^P}$$

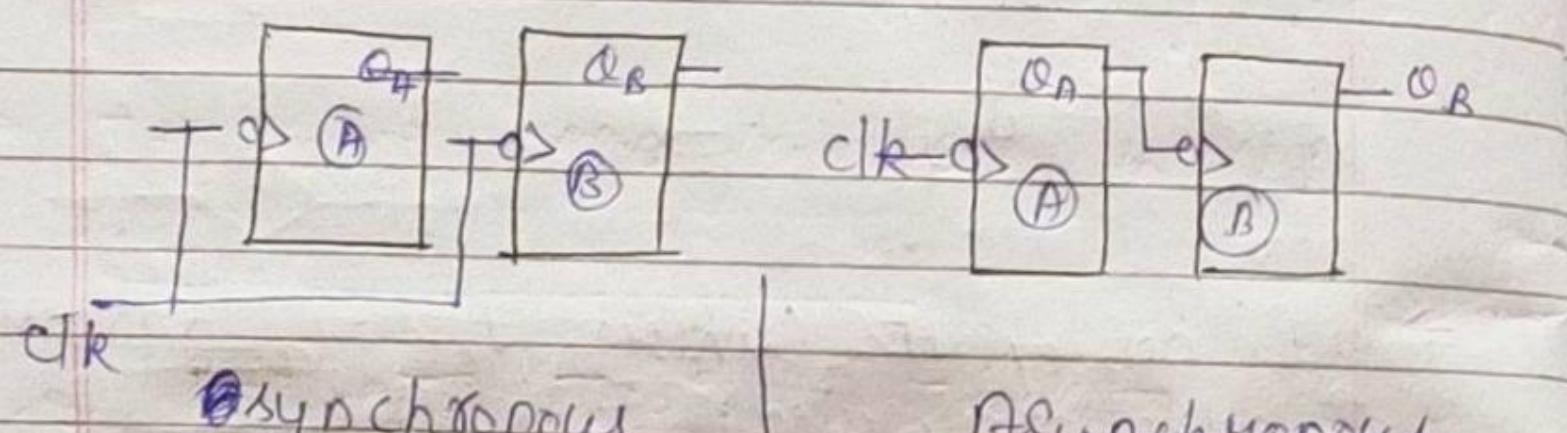
Clk	Q _A	Q _B
0	0	0 (0)
1	0	1 (1)
2	1	0 (2)
3	1	1 (3)

→ If you want to count upto 0-15 required is $2^4 = 16$

होड़ न परि तक कॉन्ट करना है उसे (उत्तर नं. को) 2 की पॉवर में कॉन्वर्ट कर लो।

Types of Counters :-

- ① Asynchronous counters (Ripple counters)
- ② Synchronous counters.



① There is no connection b/w o/p of present flip-flop and clock of next flip-flop

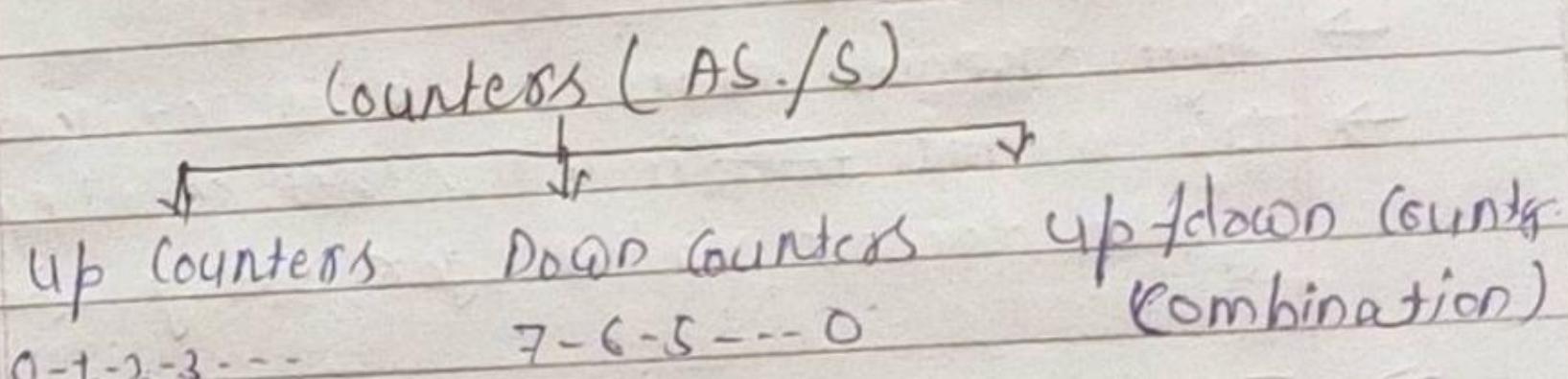
Flip-flops are connected in such a way that the o/p of present flip-flop drives the clock of next FF

② Flip-flops are not clocked simultaneously.

Flip-flops are clocked simultaneously.

③ circuit becomes complicated as no. of state increases

④ Speed is high as clock is given at the same time. speed is slow as clock is propagated through number of stages.



Difference between Synch. and Asynch. Sequential Circuits

Synchronous Asynchronous

① These are easy to design

② A clocked flip-flop acts as memory element

③ They are slower

① These circuits are difficult to design

② An Unclocked flip flop or time delay element is used as memory element

③ Faster as clock is not present

→ Latches are used in synchronous circuits
→ Where flip-flops used in synchronous circuits.

- a. The status of memory element is affected only at the active edge of clock, if input is changed.

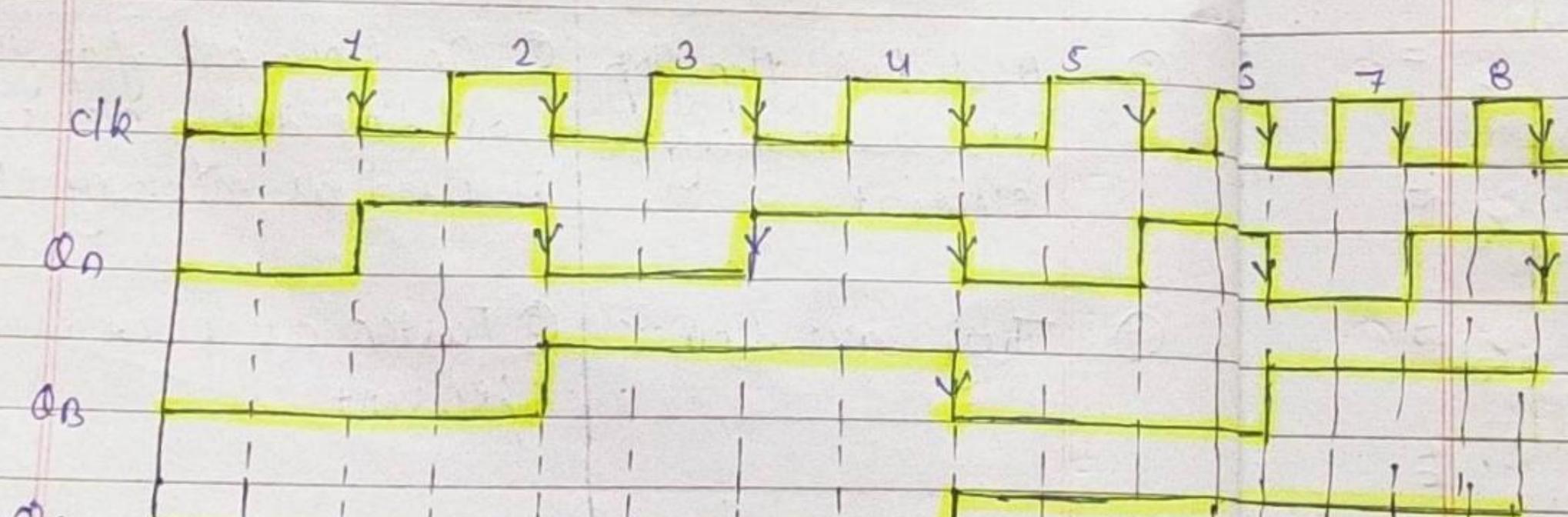
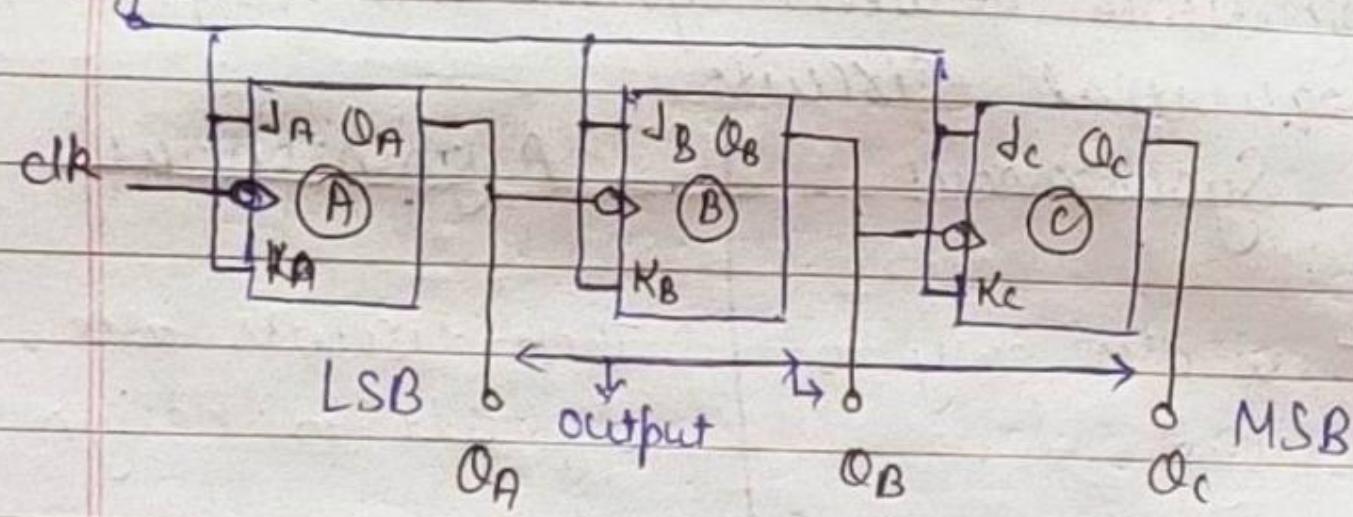
The status of memory element will change any time as soon as input is changed

3-Bit Asynchronous Up counter :-

- Asynchronous means, clock is not common
- Up means that it will start counting from lower value for every clock pulse.

- It counts 000 from 000 to 111

Logic



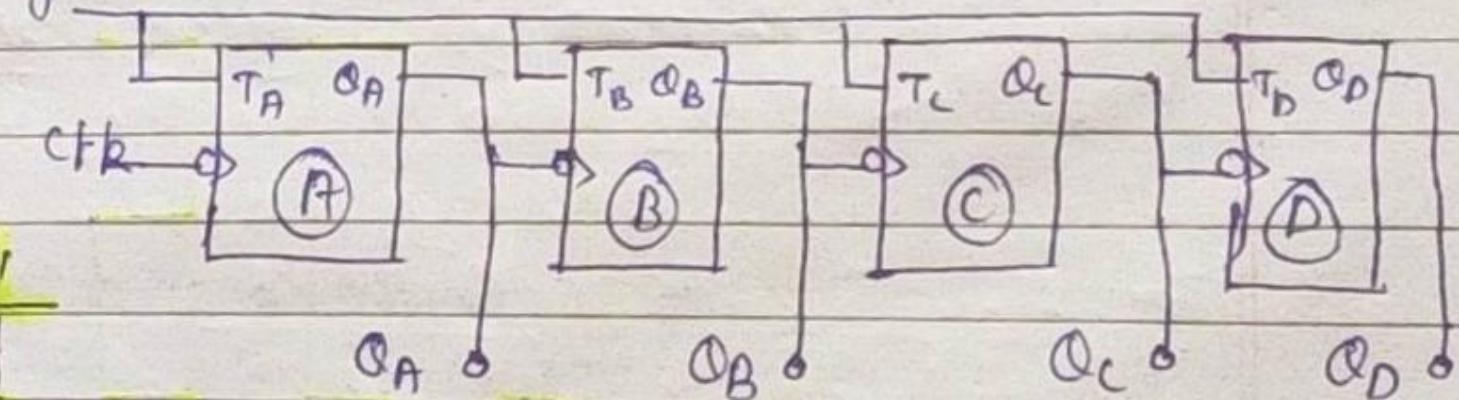
clock	d ₁	d ₂	d ₃	Decimal eq.
Initially	0	0	0	0
1 st (↓)	0	0	1	1
2 nd (↓)	0	1	0	2
3 rd (↓)	0	1	1	3
4 th (↓)	1	0	0	4
5 th (↓)	1	0	1	5
6 th (↓)	1	1	0	6
7 th (↓)	1	1	1	7
8 th (↓)	0	0	0	0

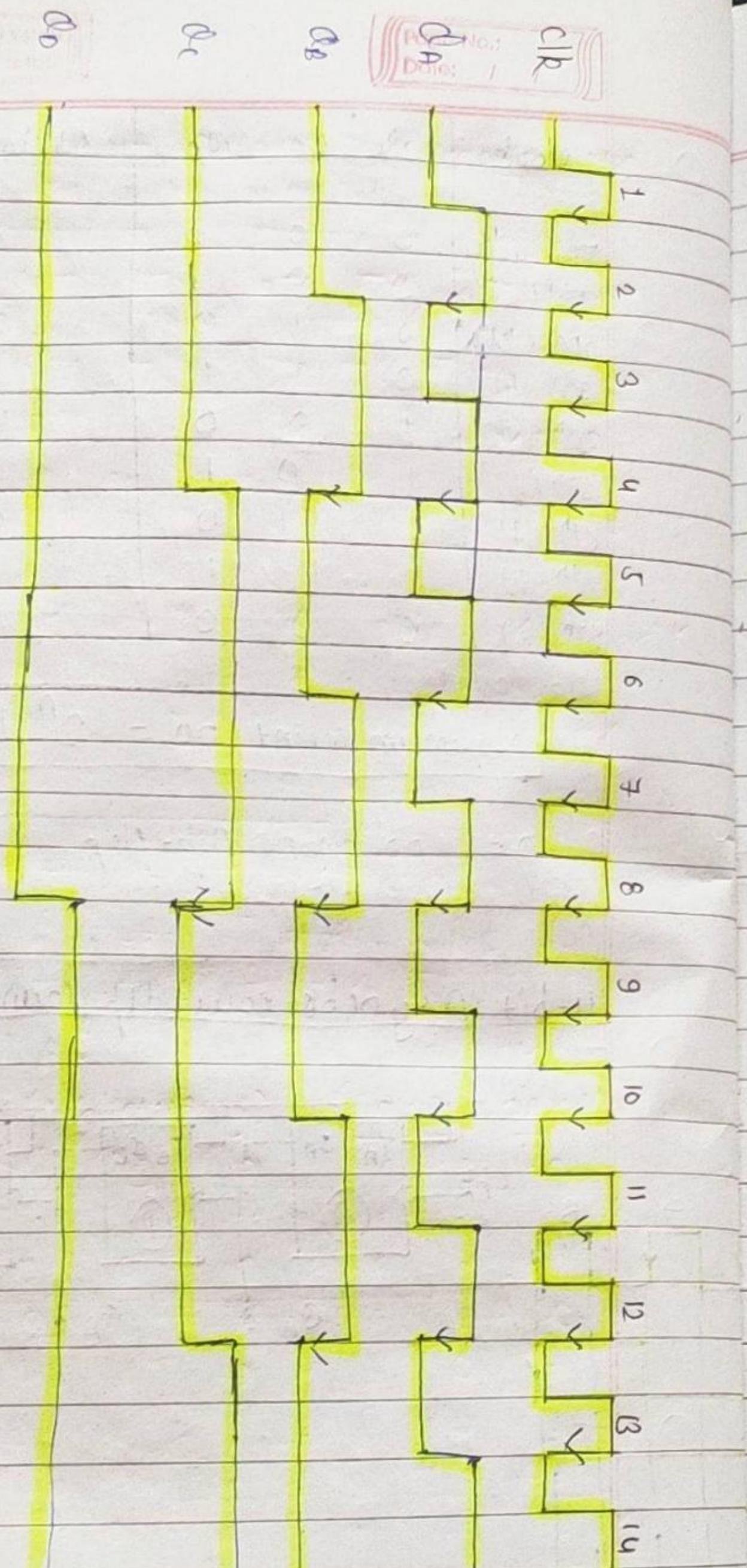
Maximum count no. = $2^n - 1$

n = no. of flip-flop

U-bit Asynchronous Up counter :-

Logic





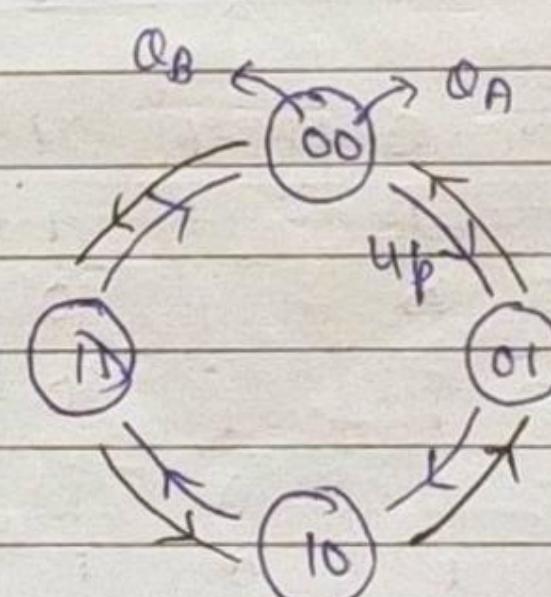
clk	Q ₀	Q ₁	Q ₂	Q ₃	Decimal
initial	0	0	0	0	0
1 st	0	0	0	1	1
2 nd	0	0	1	0	2
3 rd	0	0	1	1	3
4 th	0	1	0	0	4
5 th	0	1	0	1	5
6 th	0	1	1	0	6
7 th	0	1	1	1	7
8 th	1	0	0	0	8
9 th	1	0	0	1	9
10 th	1	0	1	0	10
11 th	1	0	1	1	11
12 th	1	1	1	0	12
13 th	1	1	1	1	13
14 th	1	1	1	1	14
15 th	1	1	1	1	15

State diagram of a counter :-

2 bit up counter :-

Q₀ Q₁
00
01
10
11

Maximum Count = 2^{D-1}
M.C. = $2^2 - 1 = 3$ (11)



Modulus of the counter & counting up to particular value:-

- » 2 bit ripple counter is called MOD-4 or modulus 4 counter
- » 3 bit ripple counter is called as MOD-8 counter.

n = no. of bits

$$\text{MOD number} = 2^n$$

Ex: MOD-6 counter using MOD-8 counter

States = 6

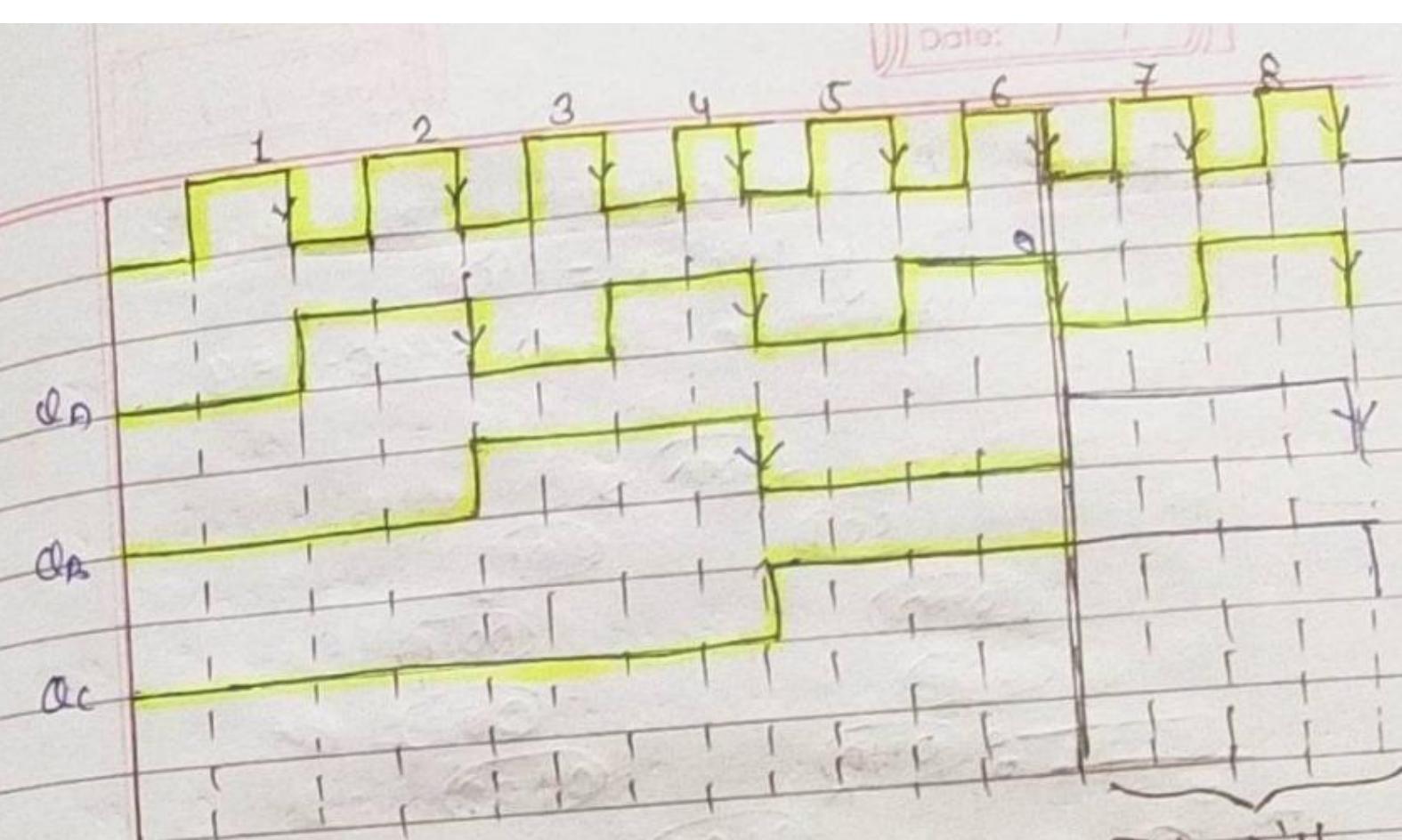
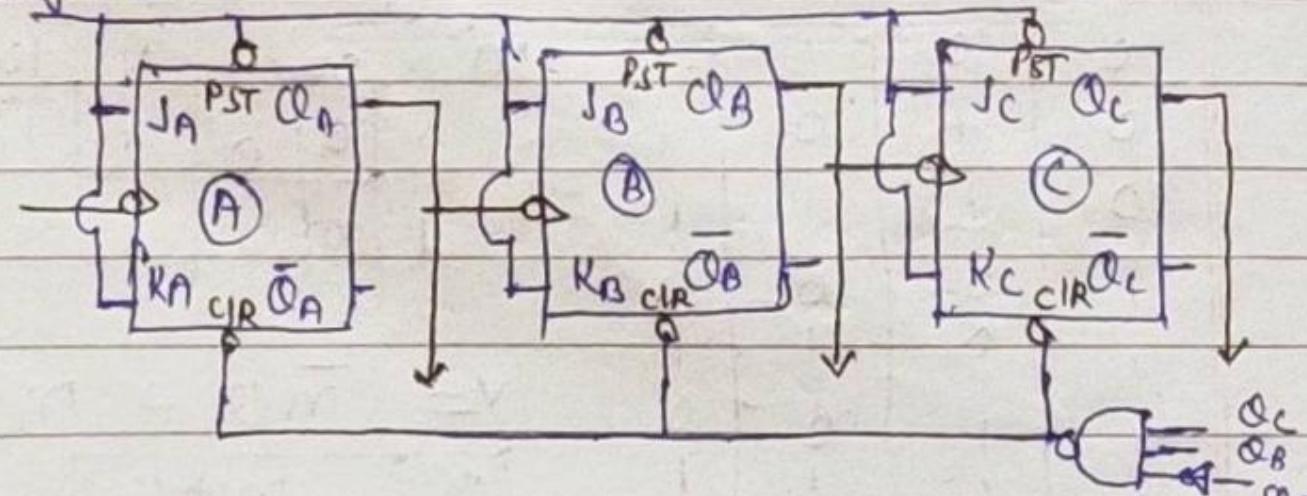
$$\begin{aligned} \text{Max Count} &= 6-1 \\ &= 5 \end{aligned}$$

① 000	MOD-6	$\left. \begin{array}{l} P_{87} = 0 \quad Q = 1 \\ CIR = 0 \quad \bar{Q} = 0 \end{array} \right\}$
② 001		
③ 010		
④ 011		
⑤ 100		
⑥ 101		

Active low (0):- they work when they are low

We don't want $\{110\}$

Logic 1.



Decade (BCD) Ripple counter :-

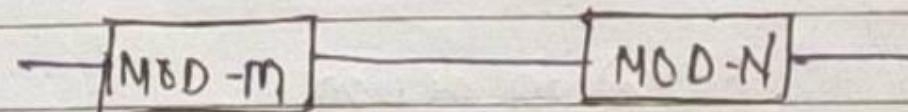
TWS will
back on
initial.

3 m p Import points:-

①

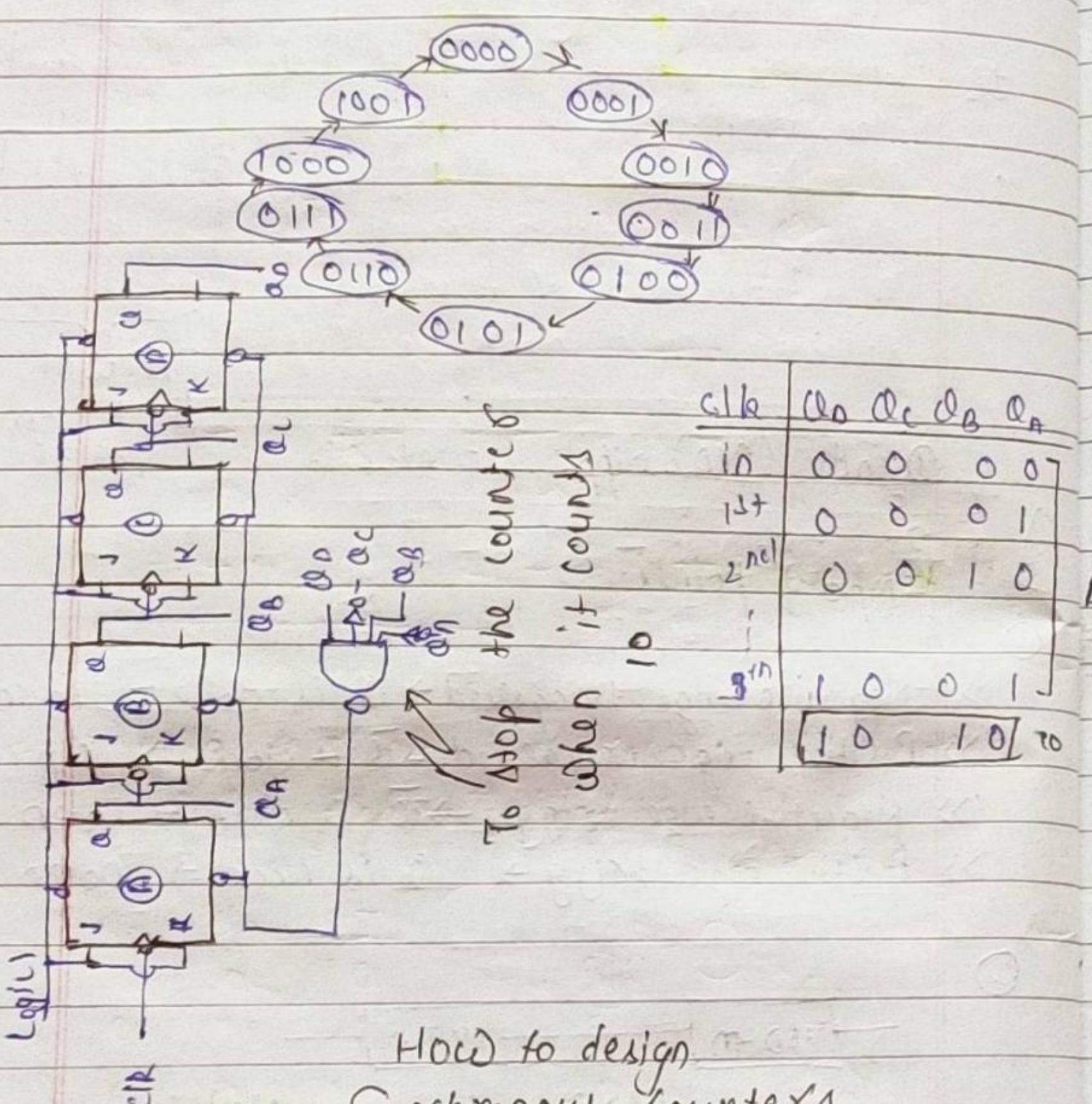
- » Negative edge triggered $\Rightarrow Q$ is clock \Rightarrow Up counter
- » Positive edge triggered $\Rightarrow \bar{Q}$ is clock \Rightarrow Up counter
- » Negative edge trigger $\rightarrow \bar{Q}$ is clock \Rightarrow Down counter
- » Positive edge trigger $\rightarrow Q$ is clock \Rightarrow Down counter

②



$$\text{MOD equivalent} = MN.$$

T + K BCD 10
no of states = 10
Maximum count = $10 - 1 = 9$.



How to design Synchronous counters

Step 1:- Decide the number of flip-flops
Step 2:- Excitation table of FF

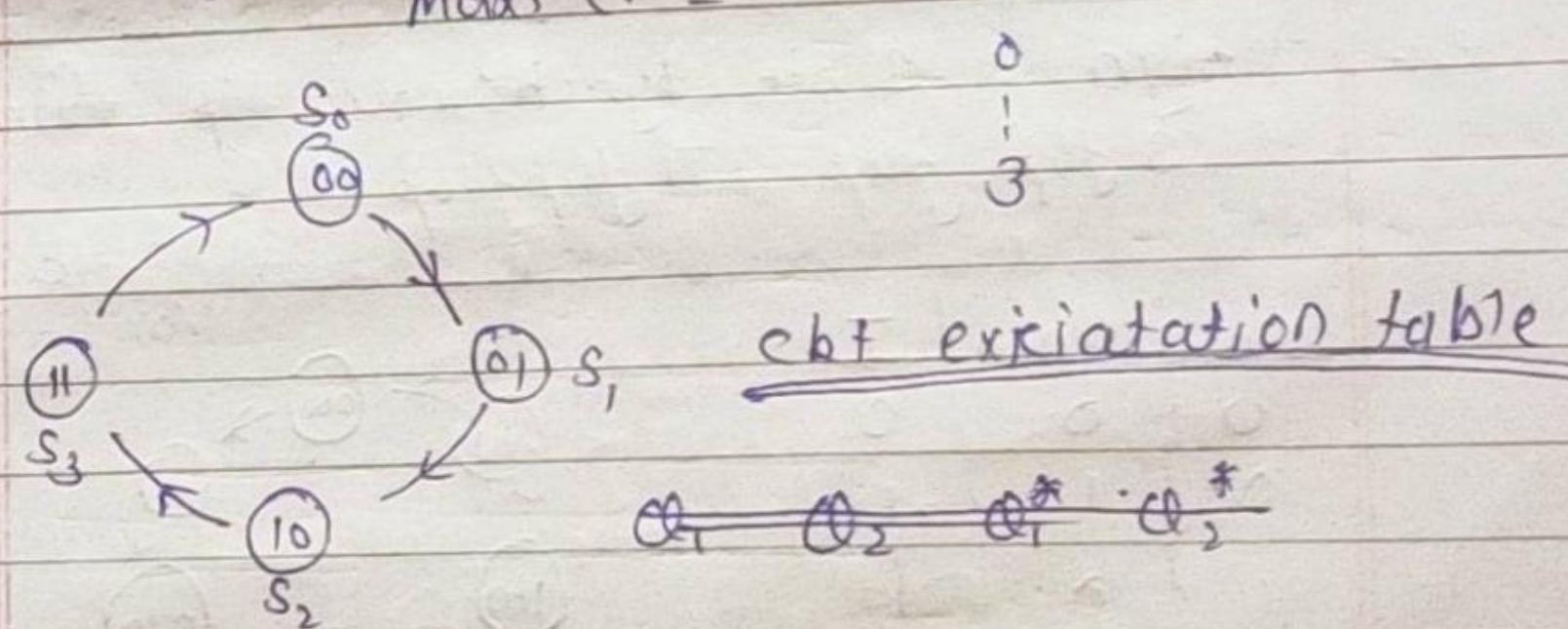
Step 3:- State diagram and circuit excitation table
Step 4:- Obtain simplified equations using K-map
Step 5:- Draw the logic diagram.

Ques:- Design the 2 bit synchronous up counter:-

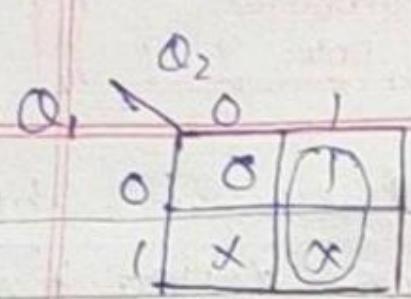
2 JK flip-flop

J	K	Q _n	Q _{n+1}
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

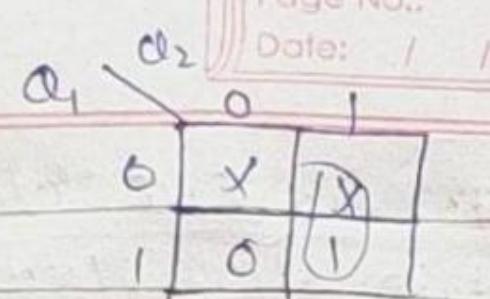
No. of states = $2^n = 2^2 = 4$
Maxi C. = 3



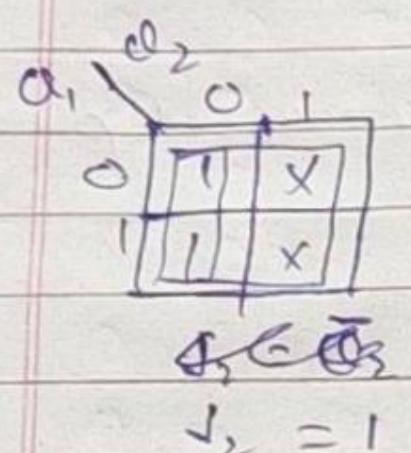
Q ₁ Q ₀	Q ₁ *	Q ₀ *	J, K ₁	K ₂
0 0	0	1	0 X	1 X
0 1	1	0	1 X	X 1
1 0	1	1	X 0	1 X
1 1	0	0	X 1	X 1



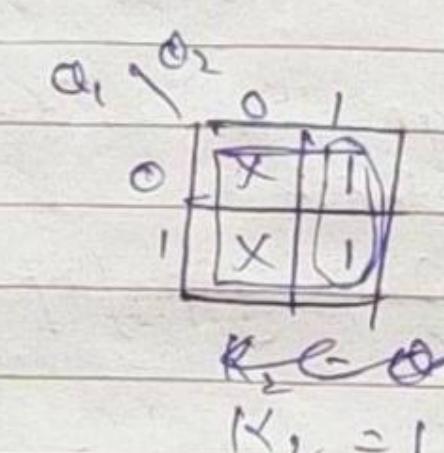
$$J_1 = Q_2$$



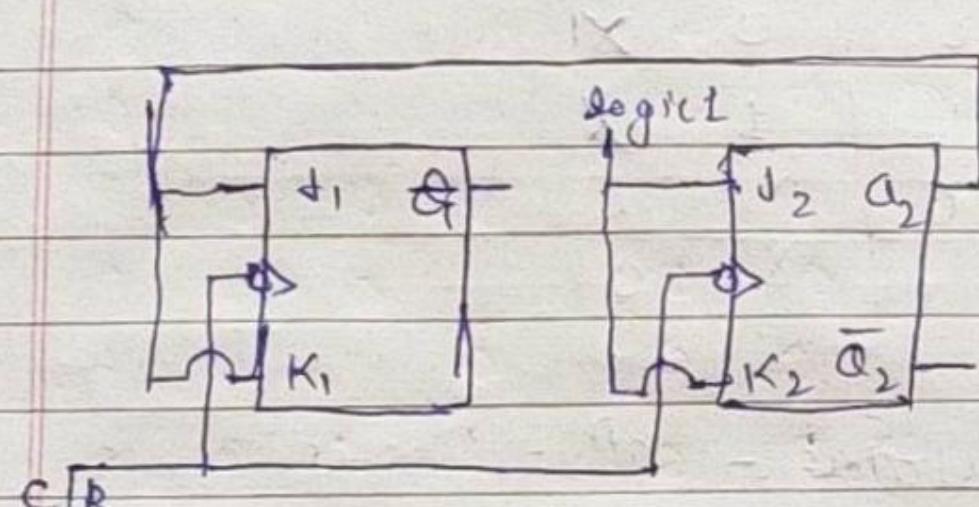
$$K_1 = Q_2$$



$$J_2 = 1$$



$$K_2 = 1$$

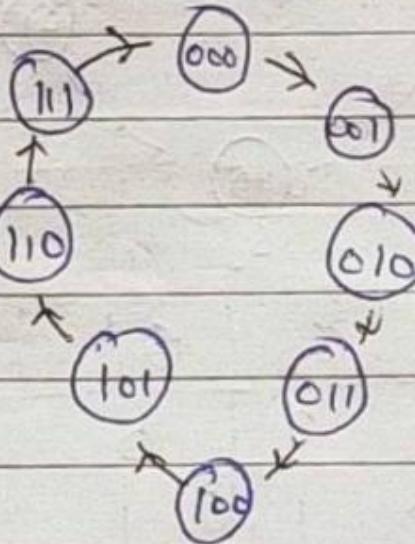


Ex:- design a 3-bit synchronous up counter.
 3 flip flop 1T flip flop

$$2^3 = 8$$

Max. L = 7

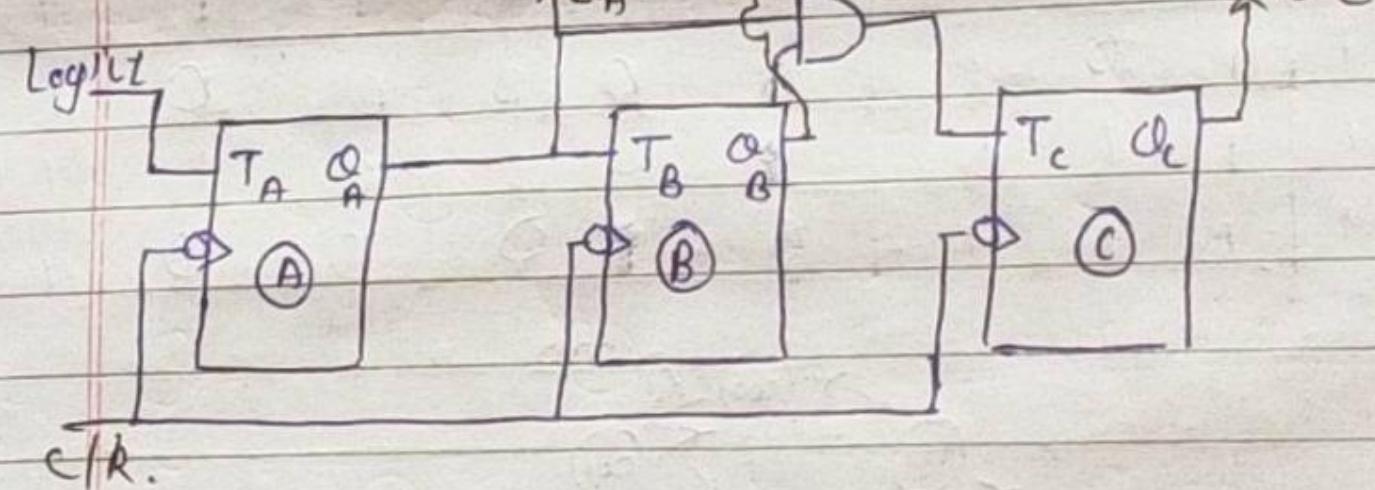
a_3	a_2	a_1	I
0	0	0	0
0	1	1	
1	0	1	
1	1	0	



P.S.			Q_C	Q_B	Q_A	Q_C^+	Q_B^+	Q_A^+	N.S.	T_C	T_B	T_A
0	0	0	0	0	1	0	1	0	0	0	1	1
0	0	1	0	1	0	1	1	1	0	0	0	1
0	1	0	0	1	1	0	0	1	1	1	1	1
0	1	1	1	0	0	1	0	1	0	0	0	1
1	0	0	1	0	1	0	1	0	0	0	1	1
1	0	1	1	1	0	1	1	0	0	0	0	1
1	1	0	1	1	1	1	1	1	0	0	1	1
1	1	1	0	0	0	0	0	0	1	1	1	1

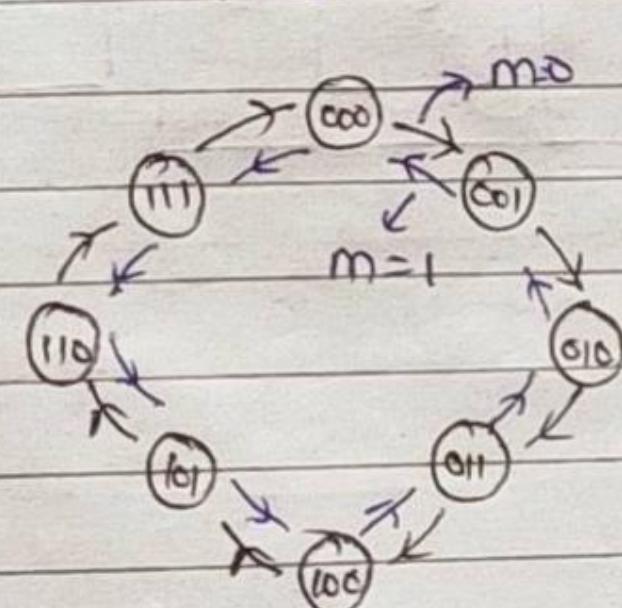
with K-map :-

$$T_A = 1 \quad T_B = Q_B \bar{Q}_A \quad T_C = Q_B Q_A$$



2-bit up/down synchronous counter

M	α_L	α_B	α_A	α_L^+	α_B^+	α_A^+	T _L	T _B	T _A
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	0	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	1	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	1	0	0	1	0	1	1	1
1	1	1	1	0	0	0	0	0	1

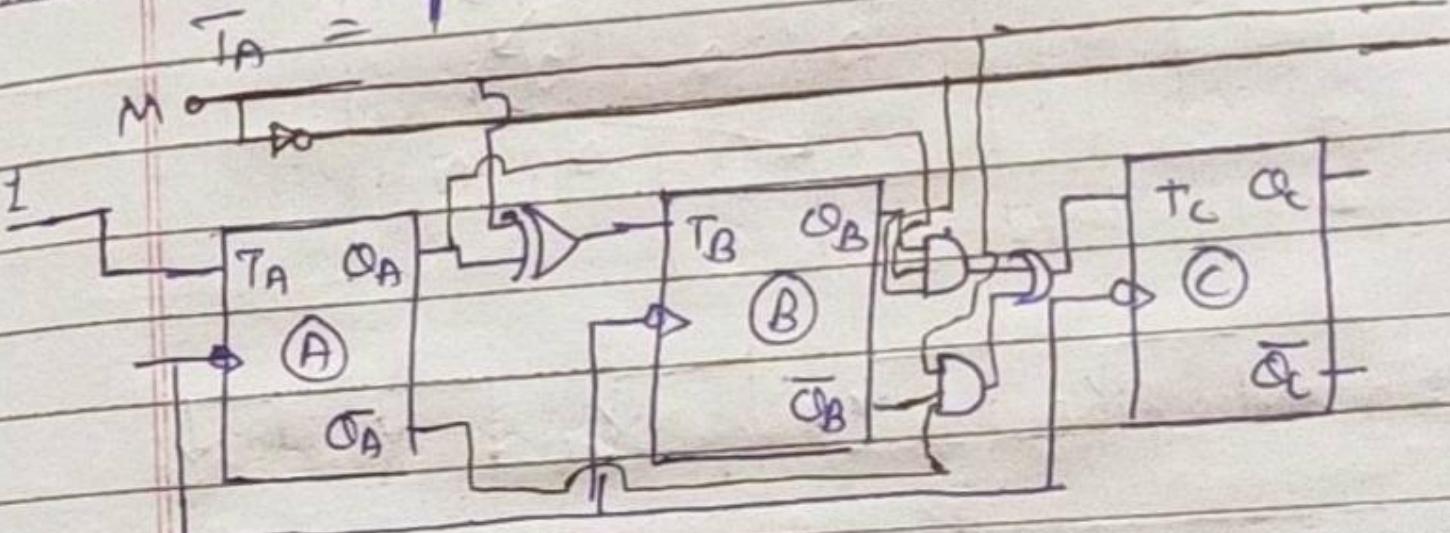


for T _C		T _C			
00	00	01	11	10	
00	01	02	12	03	
01	05	06	16	07	
11	13	14	04	15	
10	19	09	01	00	

$$T_C = \bar{M} \alpha_A \alpha_B + M \bar{\alpha}_B \bar{\alpha}_A$$

$$\text{Also } T_B = \bar{M} \alpha_A + M \bar{\alpha}_B = M \oplus \alpha_A$$

$$\bar{T}_A = 1$$

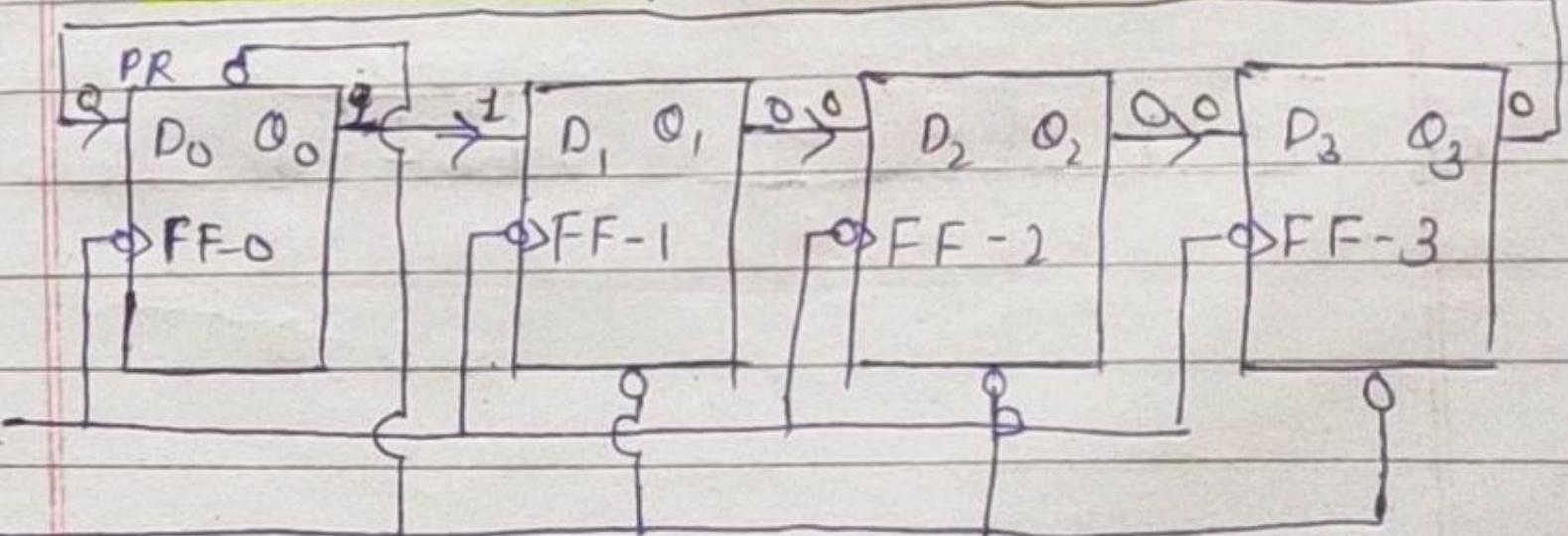


Ring Counter :-

» Ring counter is a typical application of shift register

» The only change is the output of last ff is connected to the input of first ff.

$$PR=0 \Rightarrow Q=1 ; CLR=0 \Rightarrow Q=0$$



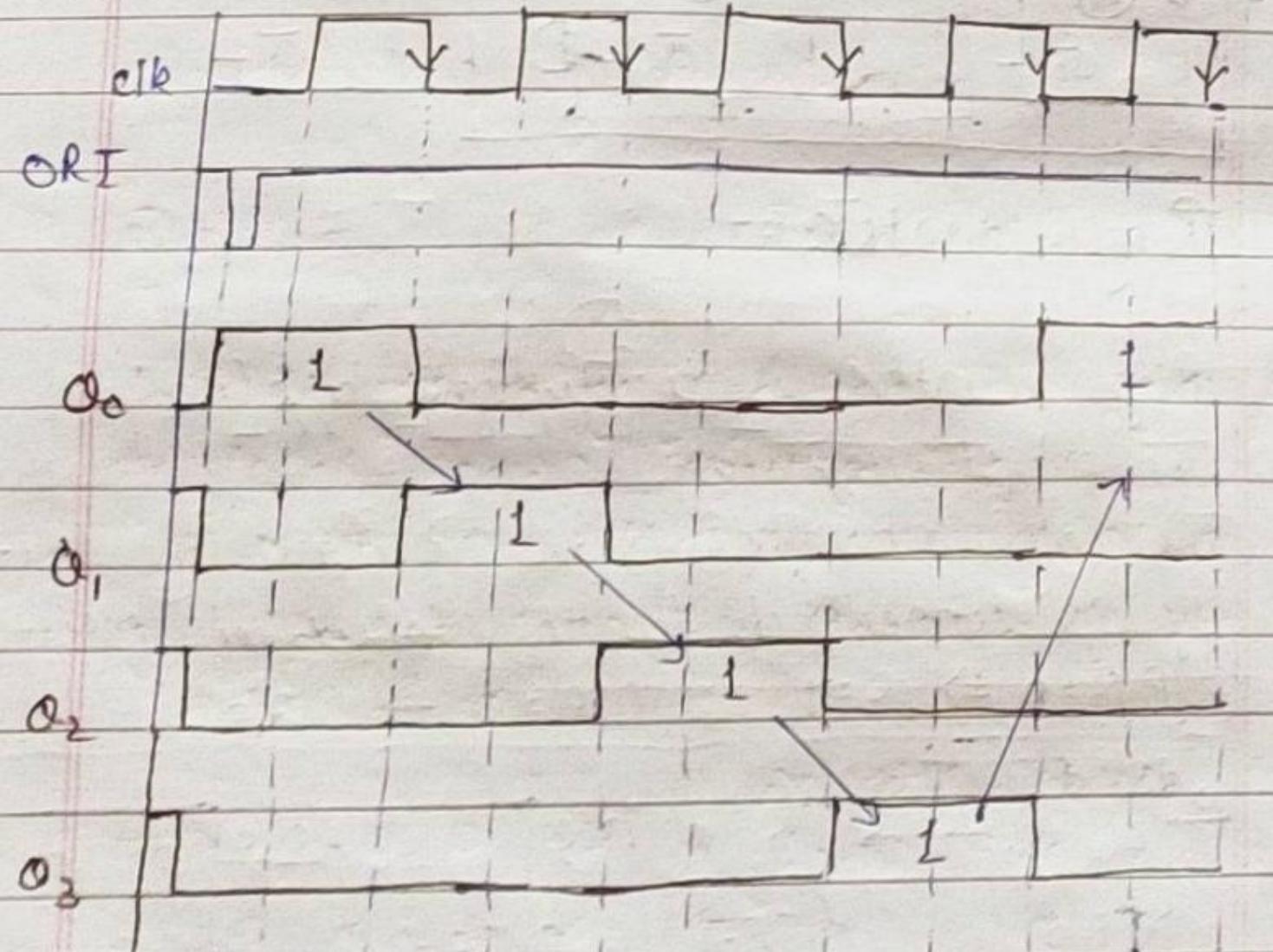
CRIT
(cover middling inputs)

imb for ring counter :-

No. of states = No. of ff used.

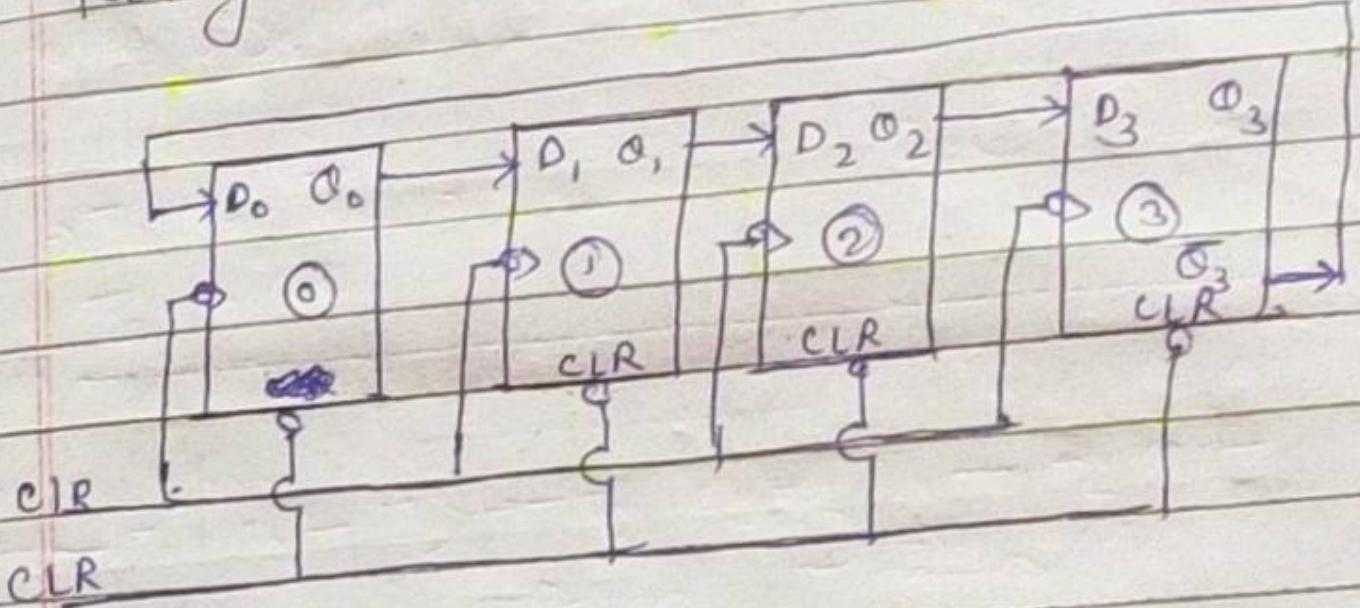
Preseted at

ORI	CLK	Q_0	Q_1	Q_2
✓ (low)	X	1	0	0
✓ (1)	↓	0	1	0
✓ (1)	↓	0	0	1
1	↓	0	0	0
1	↓	0	0	0

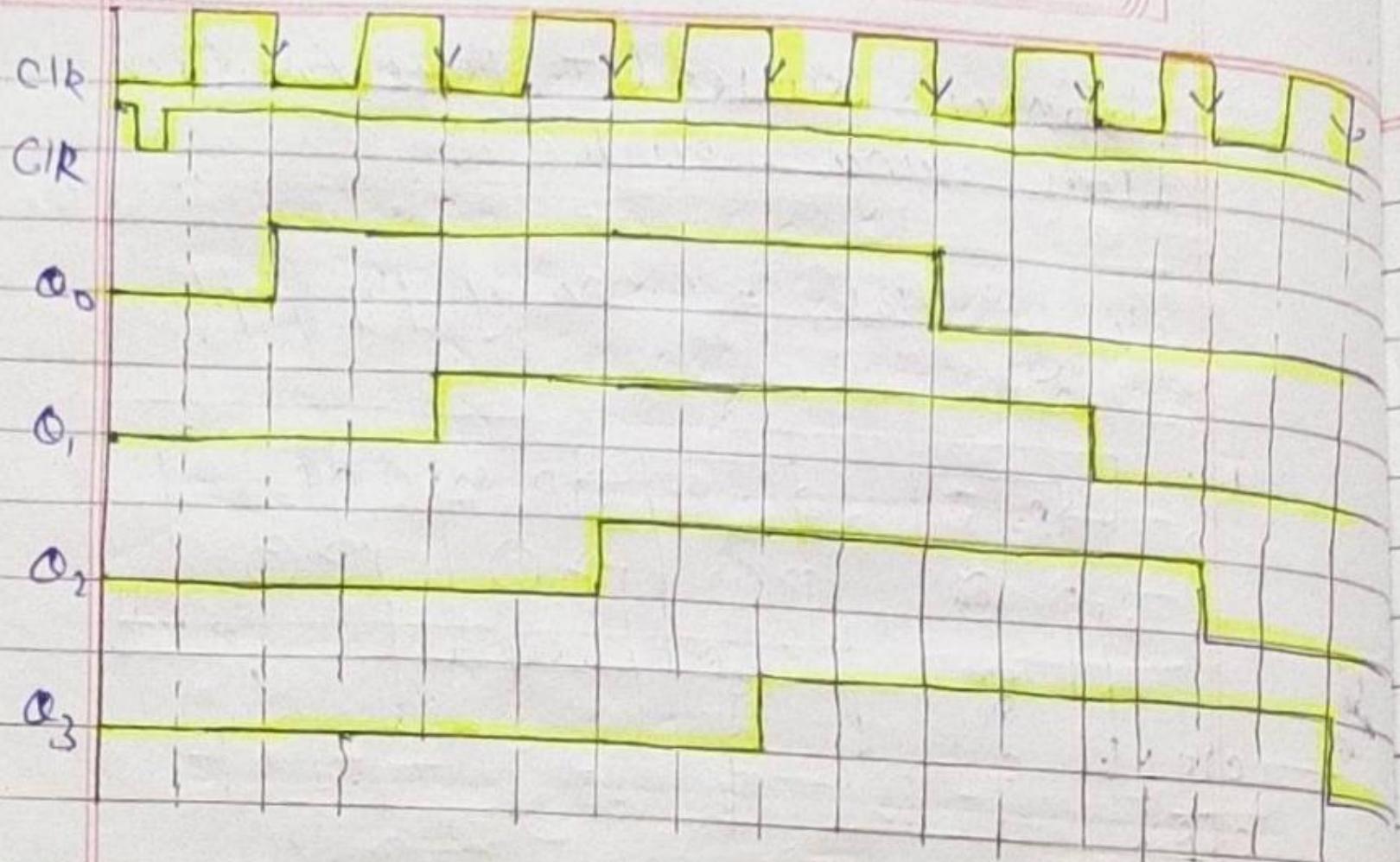


Johnson's counter (Twisted / switch tail ring counter) :-

No. of states = $2 \times$ no. of flip-flops.

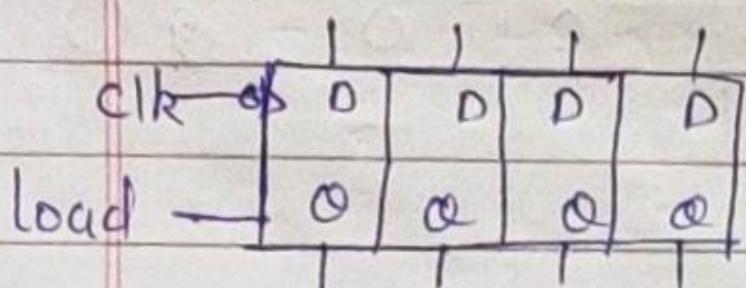


CER	CLK	Q_0	Q_1	Q_2	Q_3	
✓	X	0	0	0	0	①
1	↓	1	0	0	0	②
1	↓	1	1	0	0	③
1	↓	1	1	1	0	④
1	↓	1	1	1	1	⑤
1	↓	0	1	1	1	⑥
1	↓	0	0	1	1	⑦
1	↓	0	0	0	1	⑧
1	↓	0	0	0	0	⑨



Introduction to Registers:-

- » Flip-flop is 1-bit memory cell
- » To increase the storage capacity, we have to use group of flip-flop. This group of ff is known as Registers
- » The n-bit register consists of 'n' number of flip-flops and is capable of storing "n-bit" word.



we are bound to follow the clock

Load :- It allows us to enable any ff at a time
two types:-

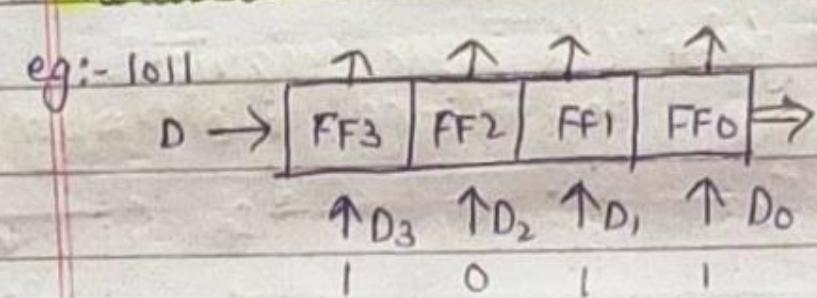
Synch :- clock ↑ is load ↑
Aynch :- only load ↑

Data formats & classification of registers.

» Data can be entered in serial or parallel form.

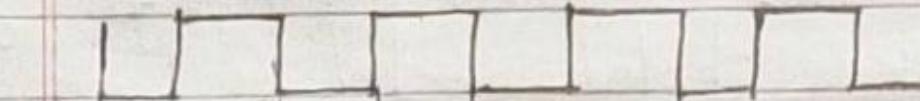
Serial → one bit at a time

Parallel → all bit at a time

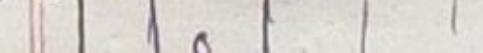


→ Serial form

= temporal code

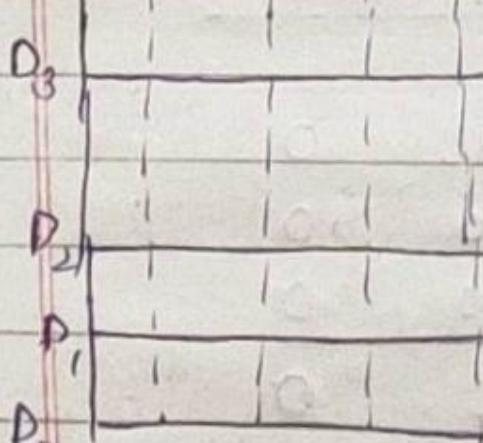


② Serial



→ Parallel form

= Special code



Classification of Registers :-

(i) Depending on S/p & op :-

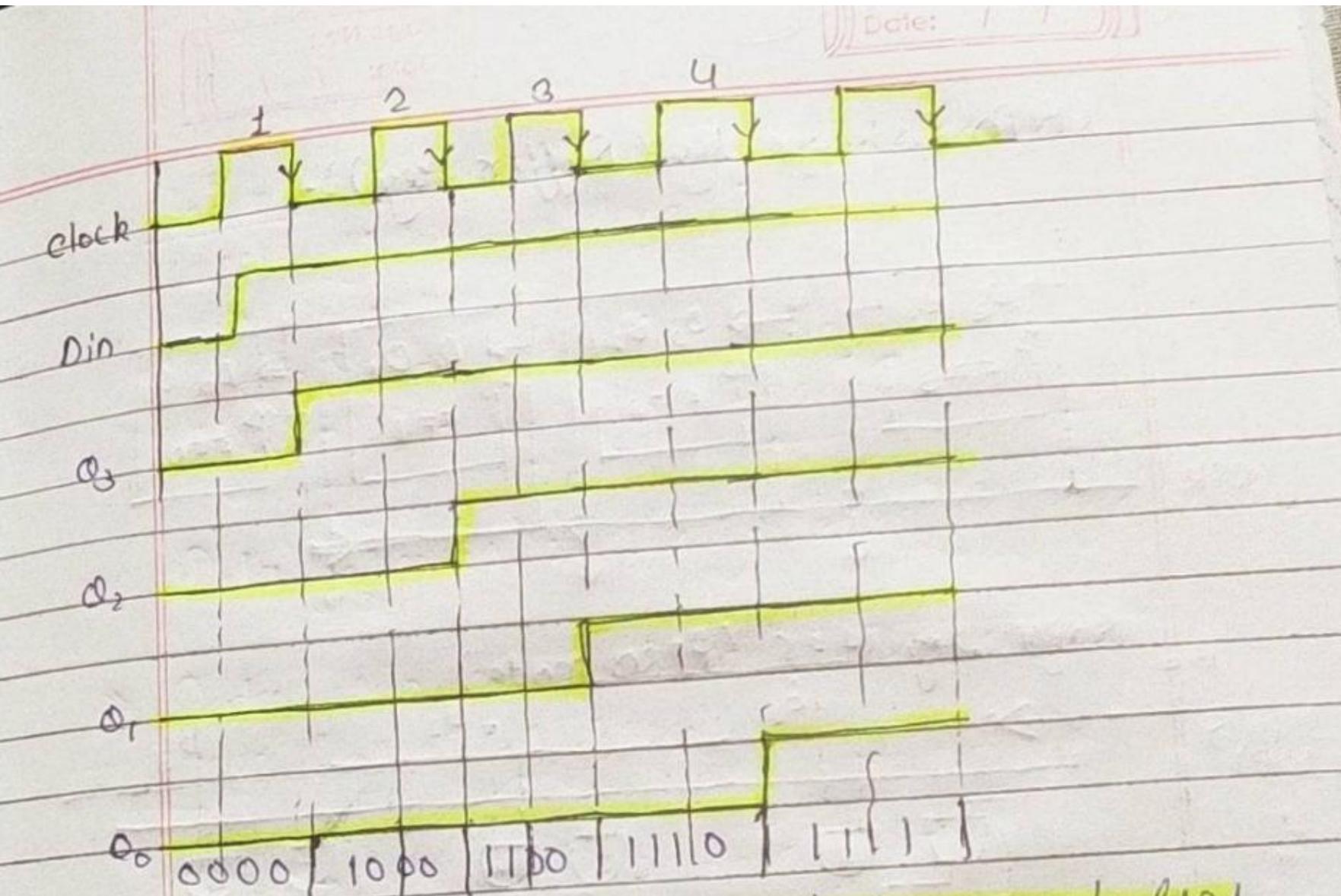
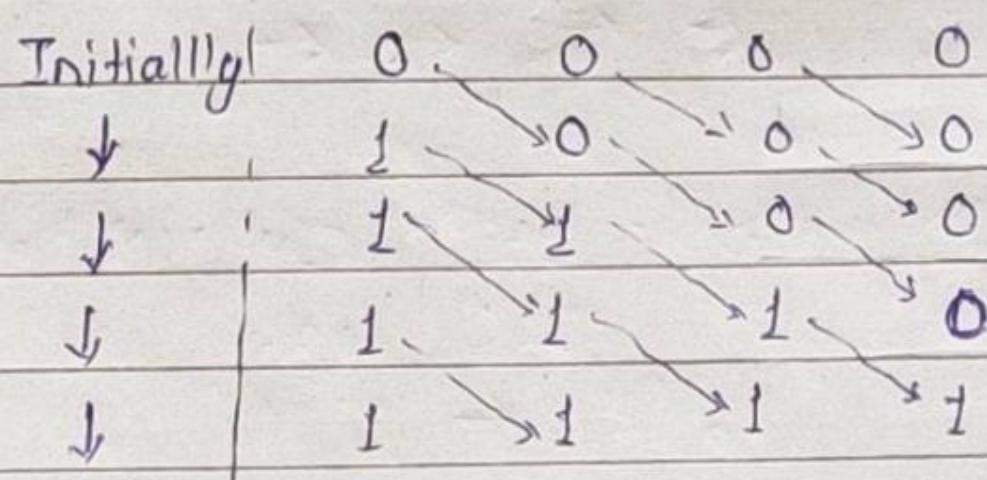
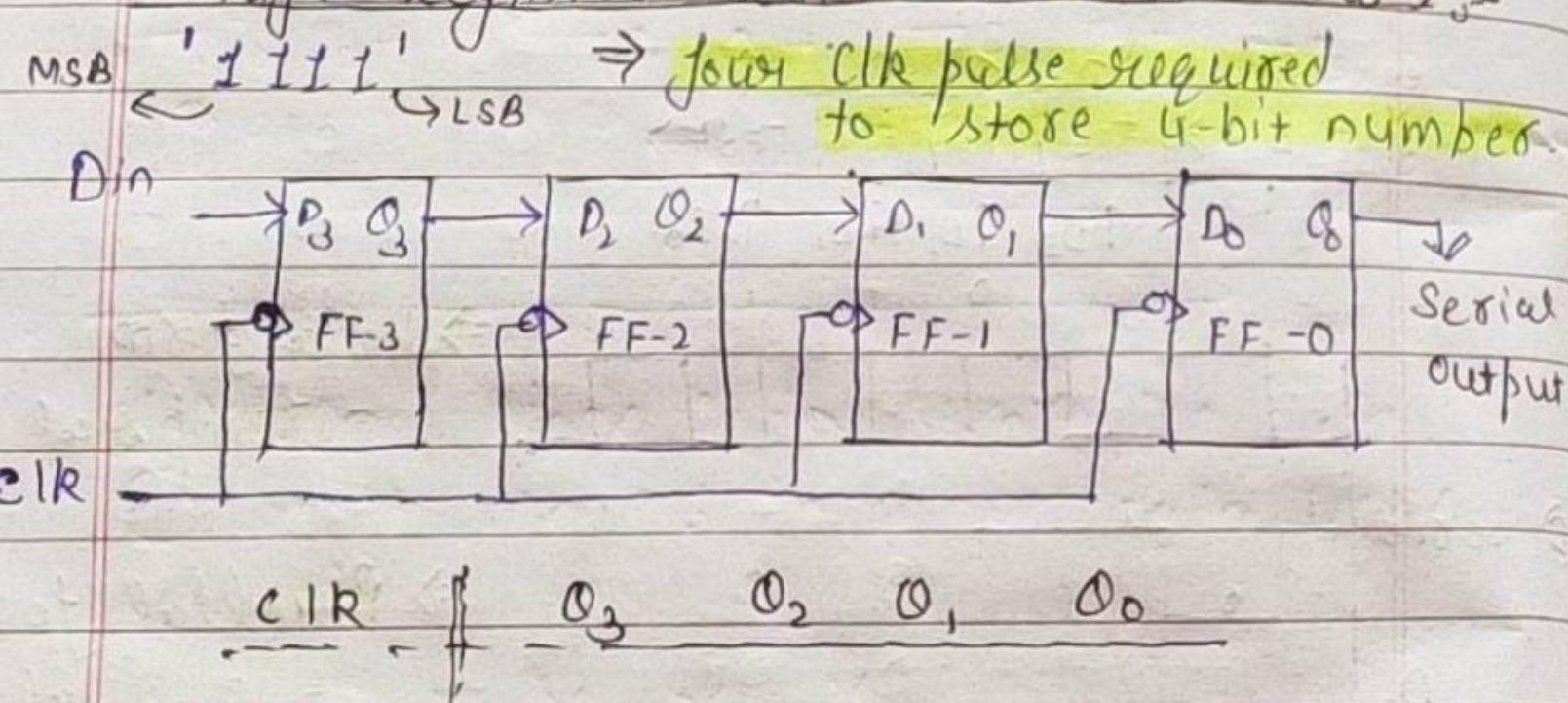
- (a) SISO (Serial S/p serial o/p)
- (b) SIPO (Serial S/p parallel o/p)
- (c) PIPO
- (d) PISO

(ii) Depending upon Application :-

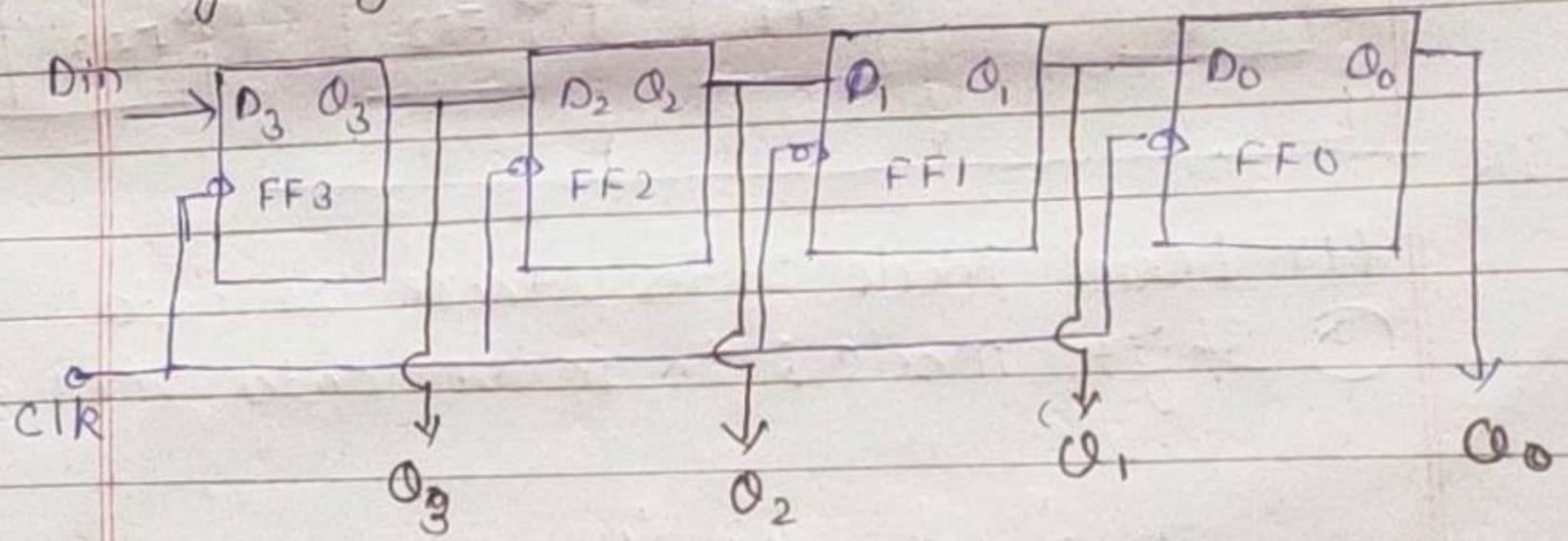
1. Shift register

2. Storage register

Shift Register (Serial in Serial out) :-



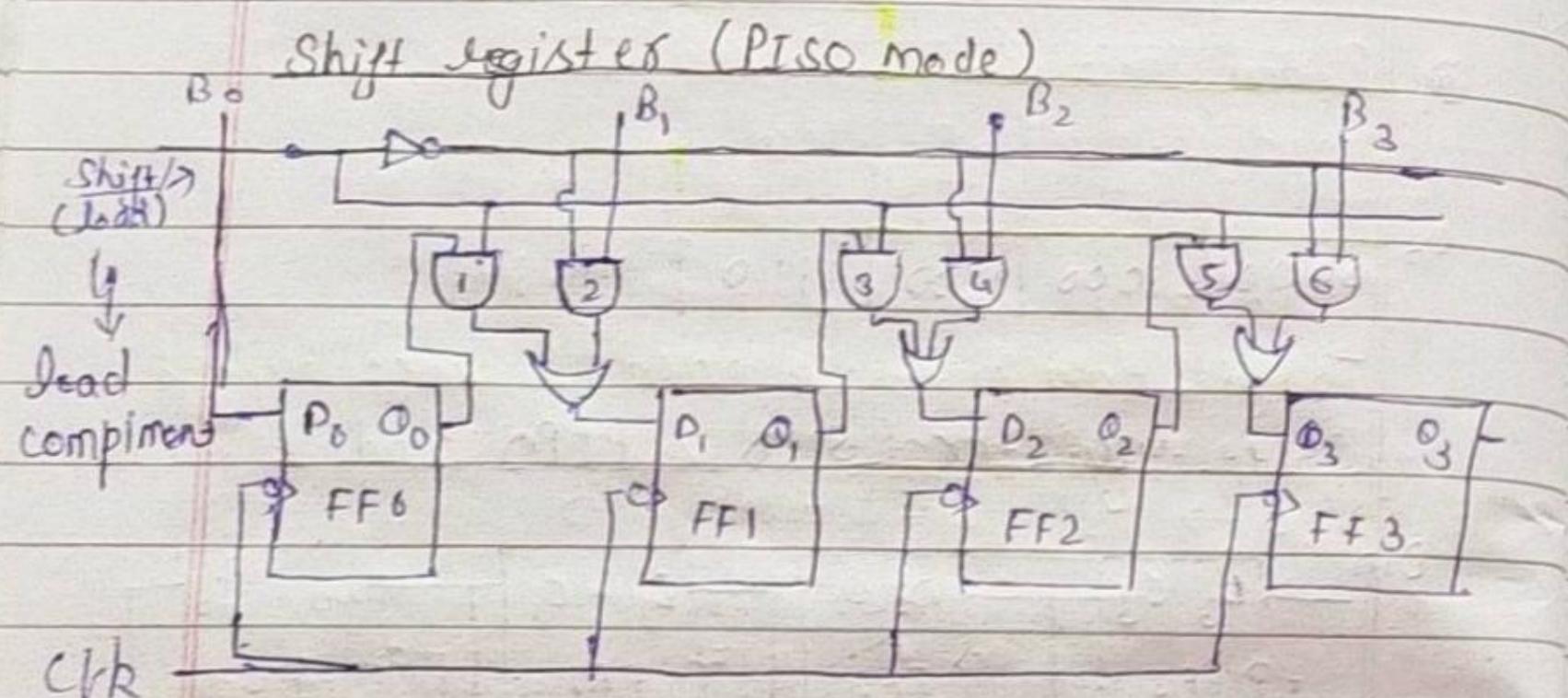
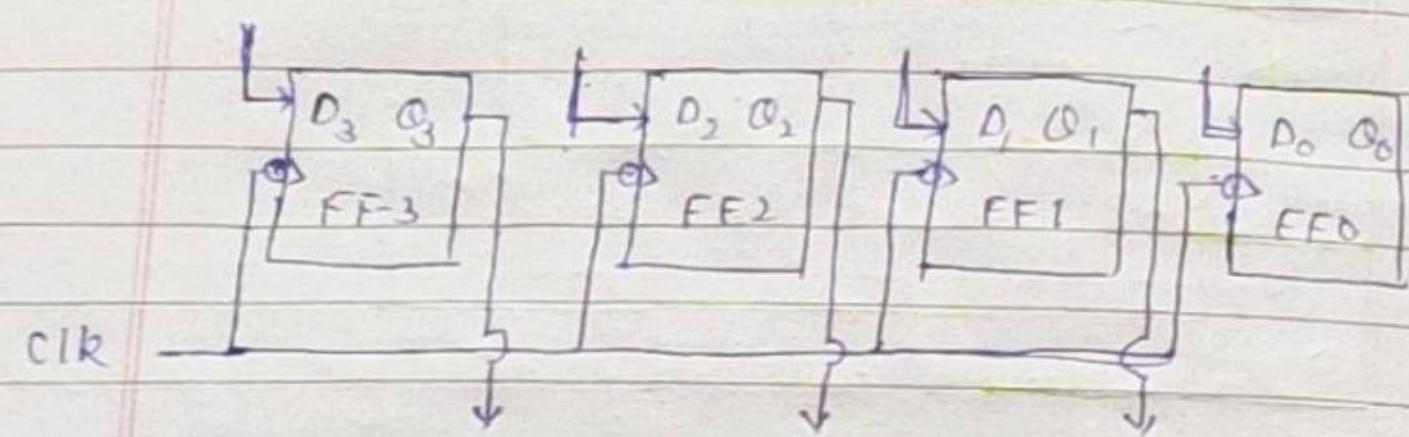
To get data back we need 4 more pulses
shift register (SIPO & PIPO mode) :-



As soon as pass the 4th clock pulse we will get the data.

(PIPO / storage reg / Buffer reg) :-

(PISO / A storage Reg / Buffer Reg): -



- (1) Load Mode :- to store the data.
- (2) Shift mode :- to move data vertically.

$\downarrow \downarrow \downarrow \downarrow$
 $\rightarrow \rightarrow \rightarrow \rightarrow$

Bidirectional shift Register :-

Data take 11

$$(11)_2 \rightarrow (3)_10$$

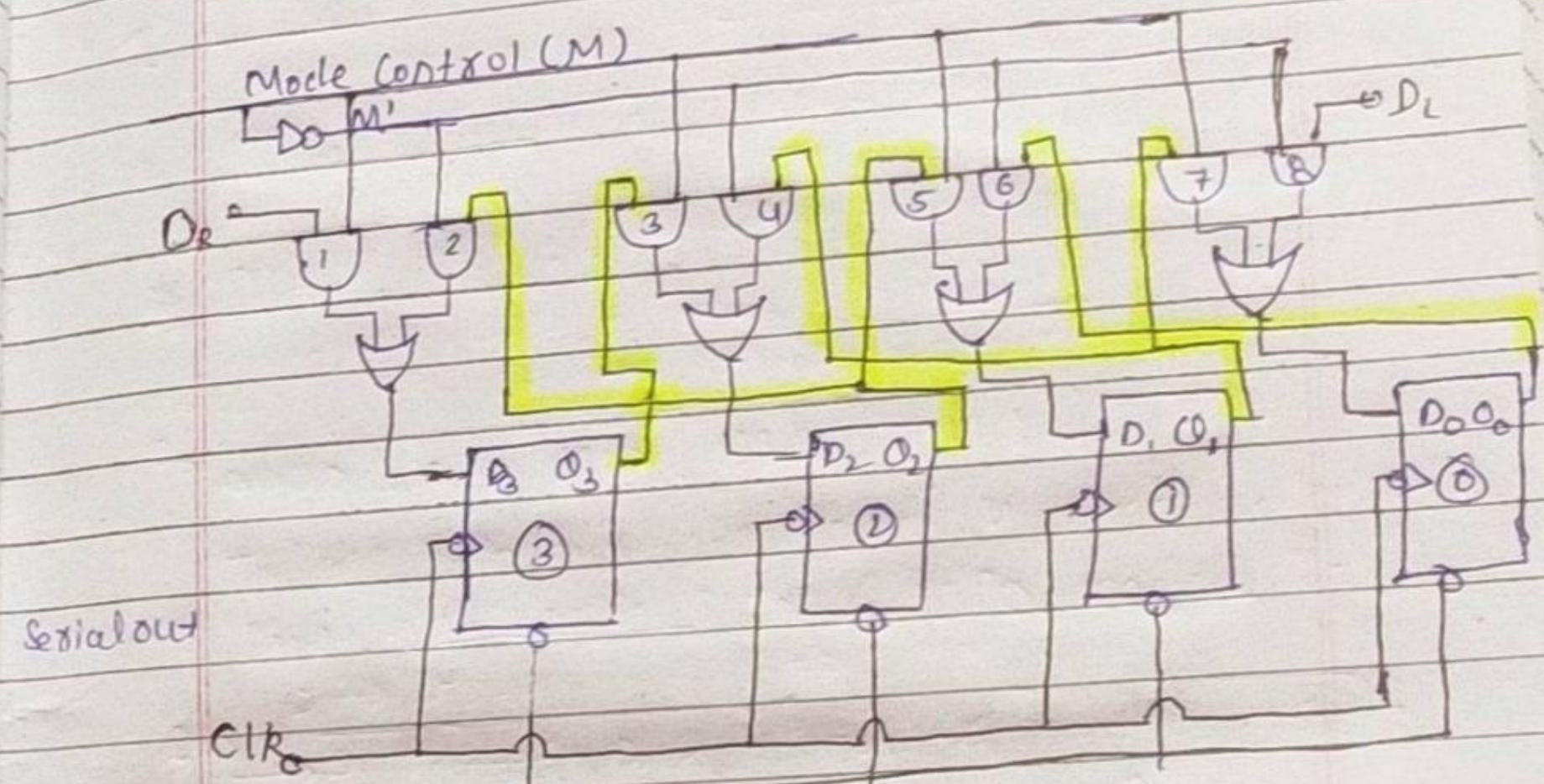
$$2^2 2^1 2^0$$

$$\leftarrow 1 \ 1 \rightarrow 3$$

$$1 \ 1 \ 0 \rightarrow 6$$

→ when we shifted it to left means we multiplied with 2

→ when we shift it to right means we divided by 2.



$M=1 \Rightarrow$ Shift right output operation
 $M=0 \Rightarrow$ Shift left output operation

Universal Shift Register :-

Bidirectional SR + Parallel Loading

Universal Shift Register.