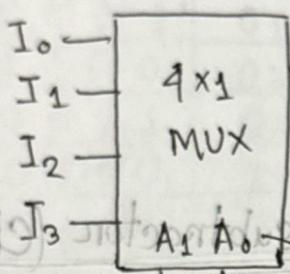


## Chapter 8

### Multiplexers (MUX)

$4 \times 1$  MUX



Address line (depends on input)

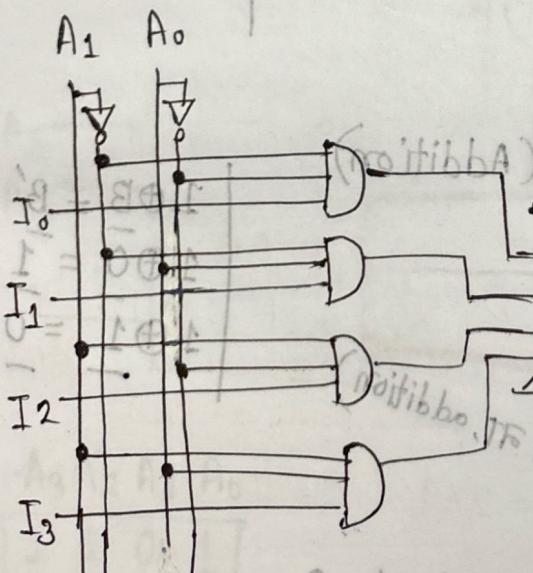
মনে করুন  $I_3 = 11$ , তাহলে এটির address নাইবে,

Truth Table:

$A_1 A_0$	$F$
0 0	$I_0$
0 1	$I_1$
1 0	$I_2$
1 1	$I_3$

$$F = I_0 A_1' A_0' + I_1 A_1' A_0 + I_2 A_1 A_0' + I_3 A_1 A_0$$

$$1 = 1 \oplus 0$$



$$B = 0 \oplus 0 = X$$

$$0 + B + A = 0 + X + A$$

$$B + A = A$$

$$B = 1 \oplus B = X$$

$$1 + B + A = 0 + X + A$$

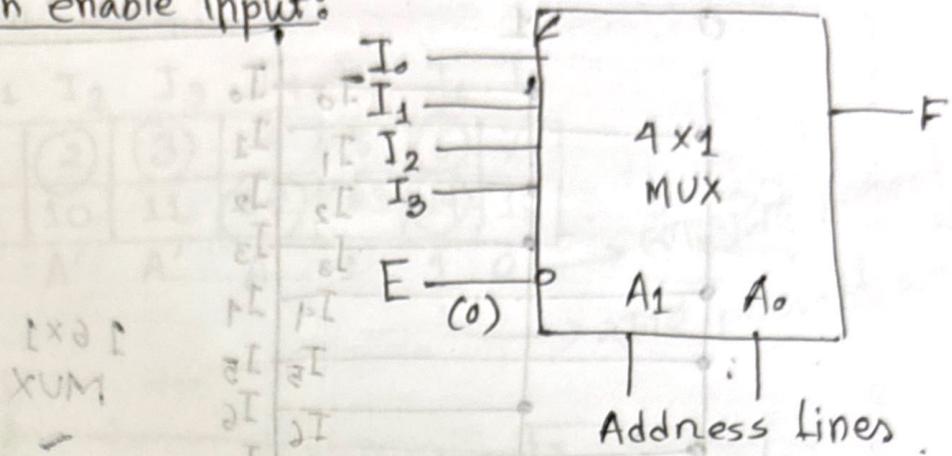
$$(1 + B) + A =$$

$$B \text{ if } B \neq 0 \\ (1 + B) + A = B - A = (B - 1) + A =$$

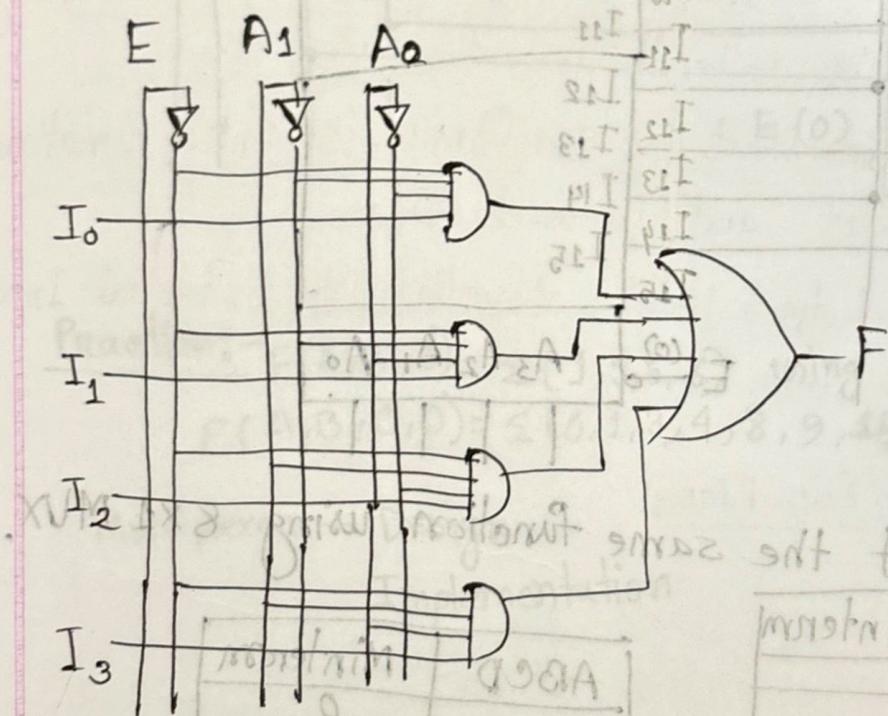
Multiplexer এর,  
 Most of the time enable  
 input active-low ২য়  
 Enable input = 1 হলে MUX disable হয়

4x1 MUX with enable input:

$E_1 A_1 A_0$	F
0 0 0	I <sub>0</sub>
0 0 1	I <sub>1</sub>
0 1 0	I <sub>2</sub>
0 1 1	I <sub>3</sub>
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	0



$$F = I_0 E' A_1 A_0 + I_1 E' A_1' A_0 + I_2 E' A_1 A_0' + I_3 E' A_1' A_0'$$



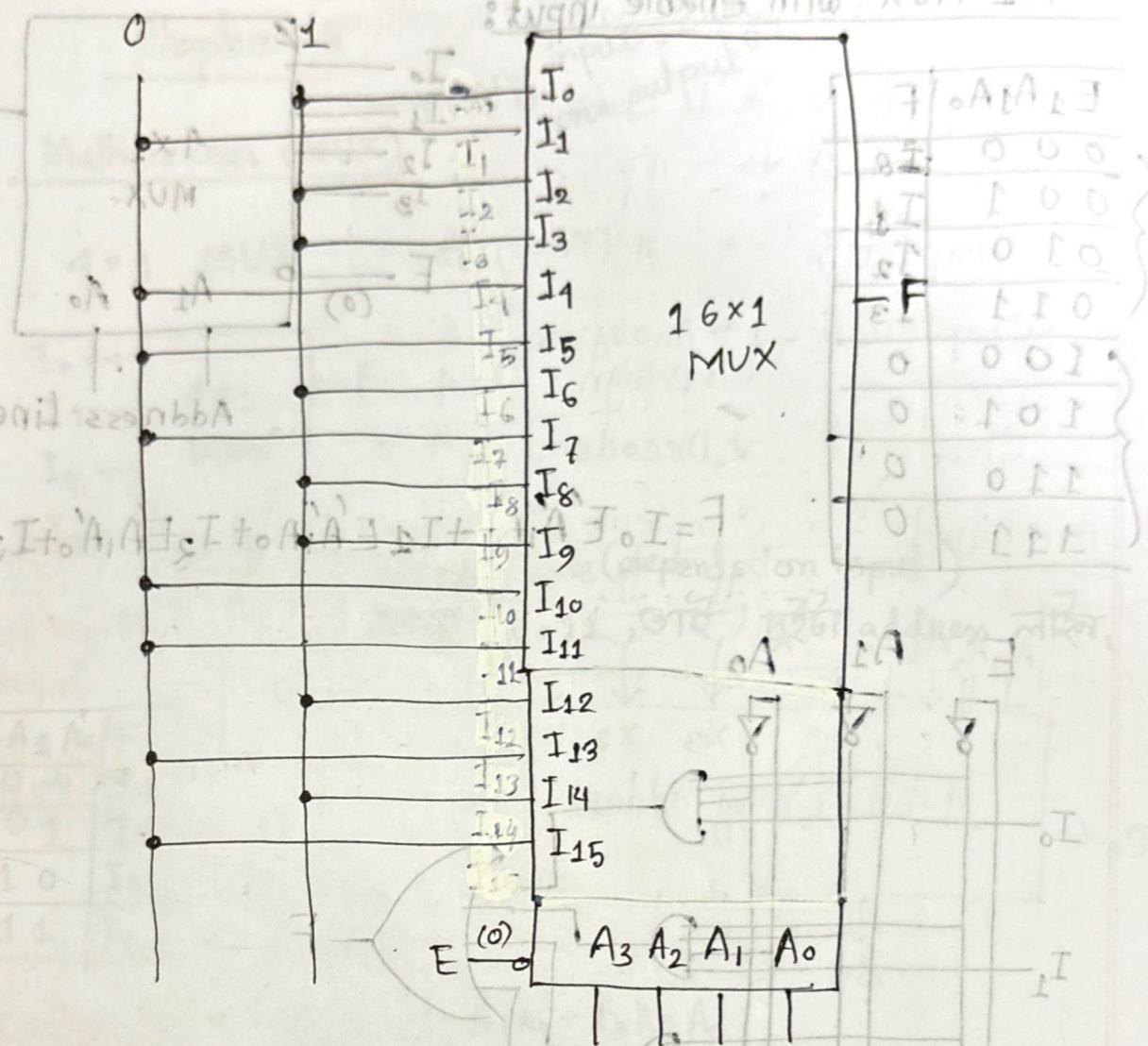
Function Implementation using MUX

$$F(A, B, C, D) = \sum(0, 2, 3, 6, 8, 9, 12, 14)$$

⇒ এটার জন্য 16 bit input আপনার। (0-15)

	ABC'D	ABC'D'	AB'C'D	AB'C'D'	AB'CD	AB'CD'	ABC'D'	ABC'D	AB'CD'	AB'CD	ABC'D'	ABC'D	ABC'D'	ABC'D	ABC'D'	ABC'D	ABC'D'	ABC'D	ABC'D'
0	0000	1000	0000	1000	0000	1000	0000	1000	0000	1000	0000	1000	0000	1000	0000	1000	0000	1000	0000
1	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
2	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
3	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
4	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
5	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
6	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
7	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
8	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
9	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
10	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
11	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
12	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
13	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
14	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
15	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
16	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
17	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
18	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
19	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
20	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
21	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
22	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
23	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
24	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
25	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
26	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
27	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
28	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
29	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
30	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111
31	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111	0111	1111

$$F(A_3, A_2, A_1, A_0) = \sum (0, 2, 3, 6, 8, 9, 12, 14)$$



Now, implement the same function using 8x1 MUX.

Truth Table

Binary	ABCD	Minterm
0000	0	0
0001	1	1
0010	2	
0011	3	XVM
0100	4	
0101	5	
0110	6	
0111	7	
1000	8	
1001	9	
1010	10	
1011	11	
1100	12	
1101	13	
1110	14	
1111	15	

ABCD	Minterm
0000	0
1000	8

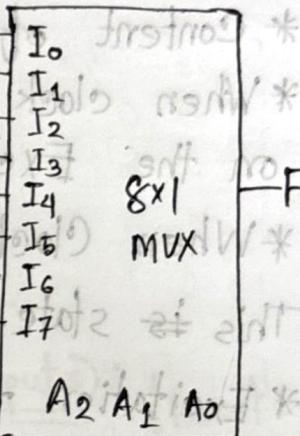
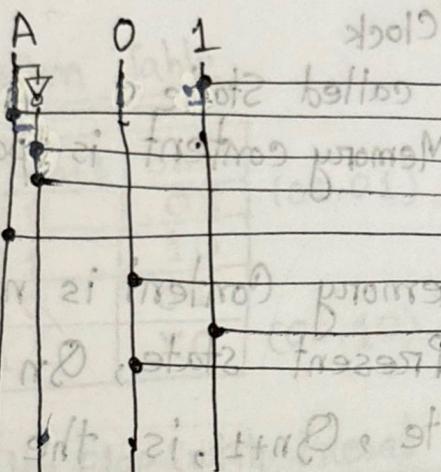
ABCD	Minterm
0001	1
1001	9

## Implementation table:

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
A'	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15

1 A A' A' A 0 1 0 → কোনটি present না আছে

→ ২টোই present আছে



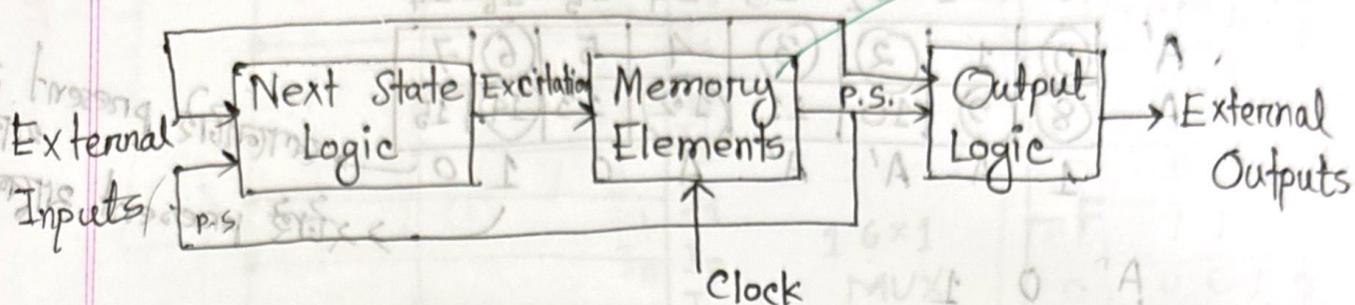
Practice:-  $F(A, B, C) = \sum(1, 3, 5, 6)$  using  $4 \times 1$  MUX

$F(A,B,C,D) = \Sigma(0,1,3,4,8,9,15)$  using 8x1 MUX.

# Multiplexer – Design Implementation

ch-8	input	output
Decoder	n	$2^n$
Encoder	$2^n$	n
MUX	many	1
DEMUX	1	many

$I + 15$	$\Theta$	$n \theta n \bar{n} n \bar{2}$
0	0	0 0 0
1	(E)	0 0 0
0	0	0 1 0
0	E	1 1 0
1	0	0 0 1
1	E	1 0 1
0	1	1 1 0
1	E	1 1 1

Chapter 9■ Flip-Flops (Sequential Circuit)

\* Content of Memory is called State

\* When clock is applied, Memory content is updated based on the Excitation.

\* When Clock is low, Memory Content is not changed.

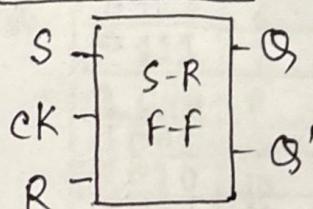
This state is called Present state,  $Q_n$

\* Excitation for Next state,  $Q_{n+1}$ , is the function of External Input and Present State.

\* External output is the function of External Input and Present State.

■ Four types of Flip-Flop:

- ① S-R F-F
- ② J-K F-F
- ③ D F-F
- ④ T F-F

■ ① S-R Flip-Flop:

$\{ CK=0, Q \text{ unchanged} \}$

$\{ CK=1, Q \text{ is determined by } S \text{ and } R \}$

Characteristic Table

$S$	$R$	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	?
1	1	1	?

Reduced Characteristic Table: (in General)

$S$	$R$	$Q_{n+1}$	$Q_n$
0	0	0	0
0	1	0	0
1	0	1	1
1	1	?	?

## Characteristic Equation / Next State Equation:

$S_n$	00	01	11	10
0	1			
1	1	X	X	

$$\therefore Q_{n+1} = S_n + R_n Q_n \text{ valid when } [S_n, R_n = 0]$$

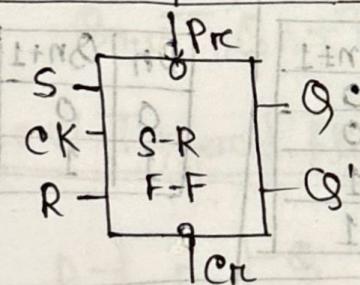
## Excitation Table

$Q_n Q_{n+1}$	$S_n R_n$
00	OX
01	10
10	01
11	X0

(00, 01)      (00, 10)

$Q_n Q_{n+1}$	$S_n R_n$
X0	00
X1	10
1X	01
OX	11

## S-R Flip Flop (with Preset and Clear Inputs):



Preset = active-low  
Clear = active-low

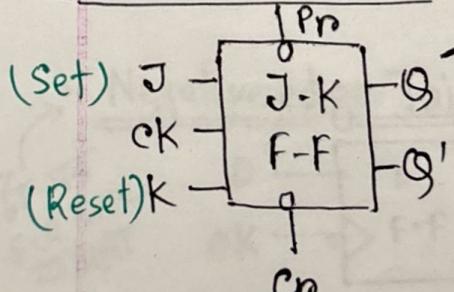
When  $CK=0$ ,  $Pr=Cr$  active

For synchronous operation,  $CK=1$  and  $Pr=Cr=$  inactive

$Pr Cr$	$Q_n$
00	Unused
01	1 (Preset)
10	0 (Clear)
11	Unchanged

$Pr Cr$	$Q_n$
00	0
01	1
10	0
11	1

## J-K Flip-Flop :-



## Characteristic Table

$J_n K_n Q_n$	$Q_{n+1}$
00 0	0
00 1	1
01 0	0
01 1	0
10 0	1
10 1	1
11 0	1
11 1	0

• Reduced Characteristic Table:

$J_n K_n$	$Q_{n+1}$
00	$Q_n$
01	0
10	1
11	$Q_n'$

## Characteristic Equation / Next State Equation:

Characteristic Table:

$J_n$	$K_n Q_n$	00	01	11	10
0	0	1			
1	1	1	1	1	1
		X	X	X	X
		$J_n Q_n'$	$K_n Q_n$		

$\therefore Q_{n+1} = K_n' Q_n + J_n Q_n'$

## Excitation Table:-

$Q_n$	$Q_{n+1}$	$J_n K_n'$
0 0	0 X	
0 1	1 X	
1 0	X 1	
1 1	X 0	

## Characteristic Table

$D_n$	$Q_n$	$Q_{n+1}$
0 0	0	0
0 1	0	1
1 0	1	0
1 1	1	1

## Reduced Characteristic Table

$D_n$	$Q_n$	$Q_{n+1}$
0	0	0
1	1	1

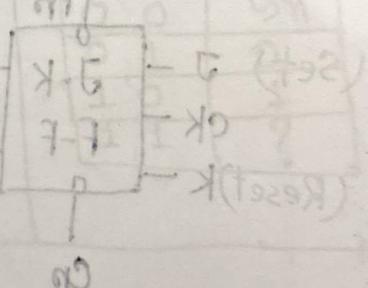
## Characteristic Equation / Next State Equations:

$D_n$	0	1
0	1	1
1	1	1

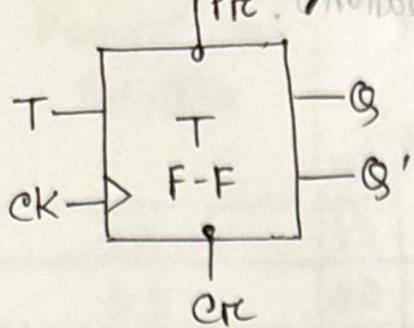
$$Q_{n+1} = D_n$$

## Excitation Table:

$Q_n$	$Q_{n+1}$	$P_n$
0 0	0	0 0 0
0 1	1	0 1 0
1 0	0	1 1 0
1 1	1	0 0 1



### ① T-F-F :



• Characteristic Table

$T_n Q_n$	$Q_{n+1}$
00	0
01	1
10	$Q_n$
11	$Q_n$

• Reduced Characteristic Table

$T_n$	$Q_{n+1}$
0	$Q_n$
1	$Q_n'$

• Characteristic Equation /  
Next State Equation

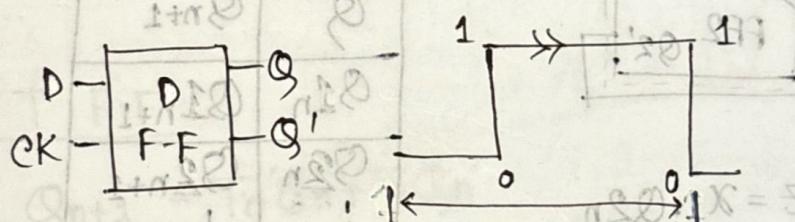
$T_n$	0	1
0	0	1
1	1	

$$Q_{n+1} = T_n Q_n + T_n Q_n' = T_n \oplus Q_n$$

• Excitation Table:

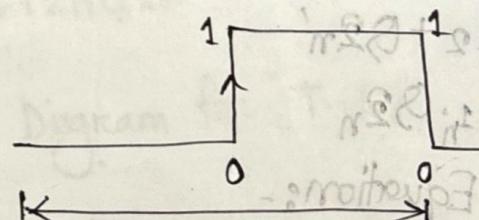
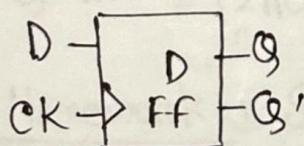
$Q_n Q_{n+1}$	$T_n$
00	0
01	1
10	1
11	0

■ Level-Trigged Flip Flop : (Clock) যেখানে F-F এ তিনটি clock ব্যবহৃত হয়



(পূর্ণ cycle utilize)  
ইন্স না

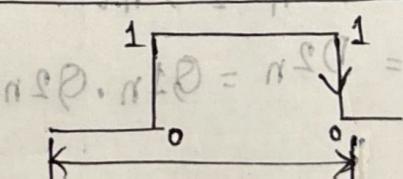
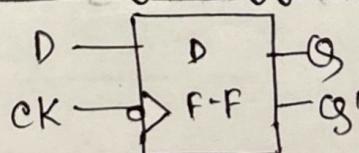
• Positive edge Triggered / Rising edge Triggered (0 to 1 হলে trigger):



(half cycle  
utilize ইন্স কারণ  
1st পর্যবেক্ষণ ইন্স)

• Negative edge Triggered / Falling-edge Triggered:-

এটি সব  
F-F এ কারণ



(পূর্ণ cycle utilize)  
ইন্স কারণ Last  
update ইন্স)

# Chapter - 10

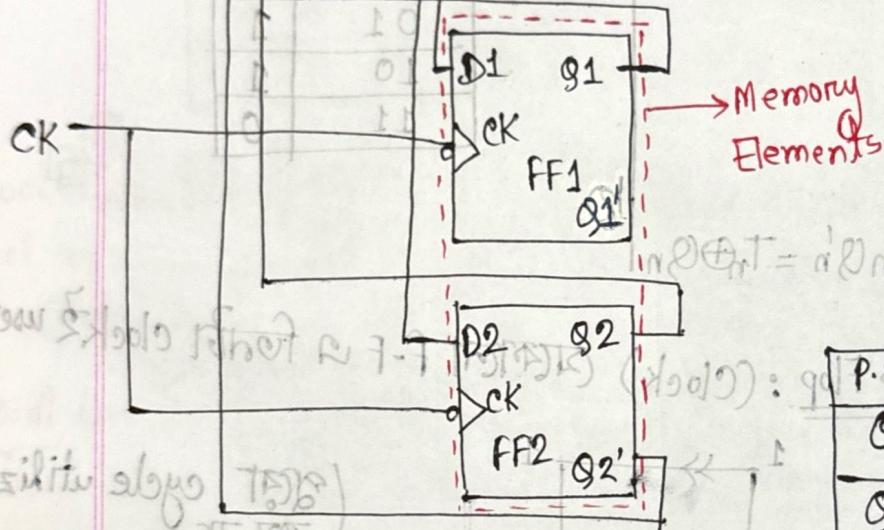
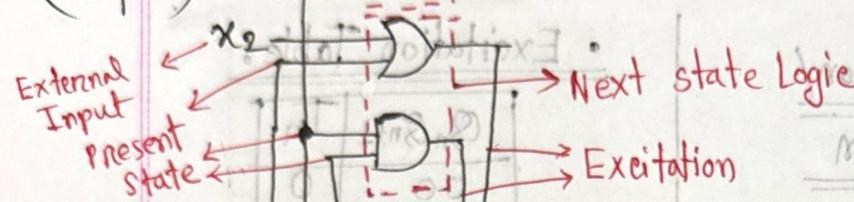
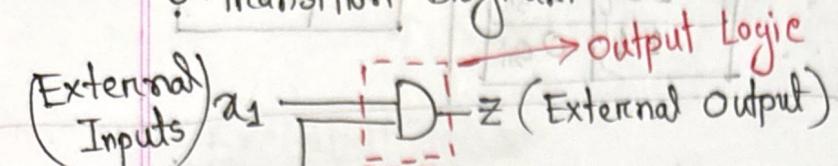
## Analysis of Sequential Circuits

: 7-7-T(A) 157

1. State Equations and Output Equations,

2. Transition Table

3. Transition Diagram



Output Equation :-  $z = x_1 \cdot Q_2 n$

Excitation Equation :-

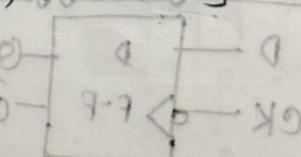
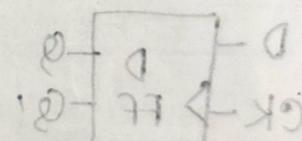
F.F input  $D_1 n = x_2 + Q_2 n'$

$D_2 n = Q_1 n \cdot Q_2 n$

Next State Equation :-

$$Q_1 n+1 = D_1 n = x_2 + Q_2 n' \quad [\text{We know, } Q_1 n+1 = D_1 n]$$

$$Q_2 n+1 = D_2 n = Q_1 n \cdot Q_2 n$$



Eqs:

$$Q_1^{n+1} = x_2 + Q_2^n$$

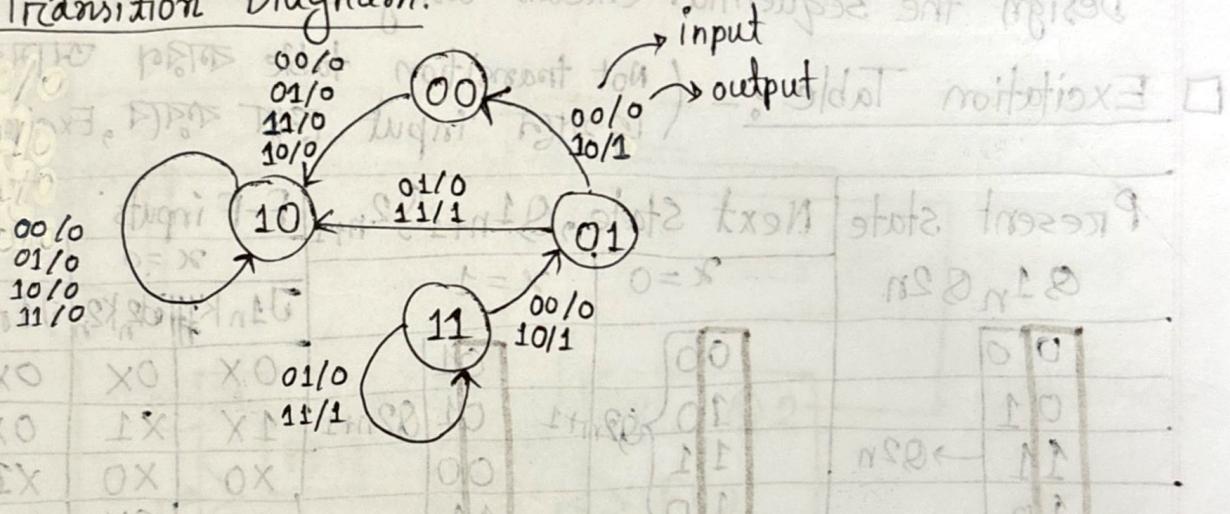
$$Q_2^{n+1} = Q_1^n \cdot Q_2^n$$

$$z = x_1 \cdot Q_2^n$$

### Transition Table:

Present State $Q_1^n Q_2^n$	Next State $Q_1^{n+1} Q_2^{n+1}$				Output, z Inputs $x_1, x_2$			
	Inputs $x_1, x_2$	00	01	11	10	00	01	11
00	10	10	10	10	0	0	0	0
01	00	10	10	00	0	0	1	1
11	01	11	11	01	0	0	1	1
10	10	10	10	10	0	0	0	0

### Transition Diagram:



T F-F:

$$Q_{n+1} = T' Q_n + T_n Q_n'$$

$$Q_1^{n+1} = T_1' Q_1^n + T_1^n Q_1^n'$$

$$Q_2^{n+1} = T_2' Q_2^n + T_2^n Q_2^n'$$

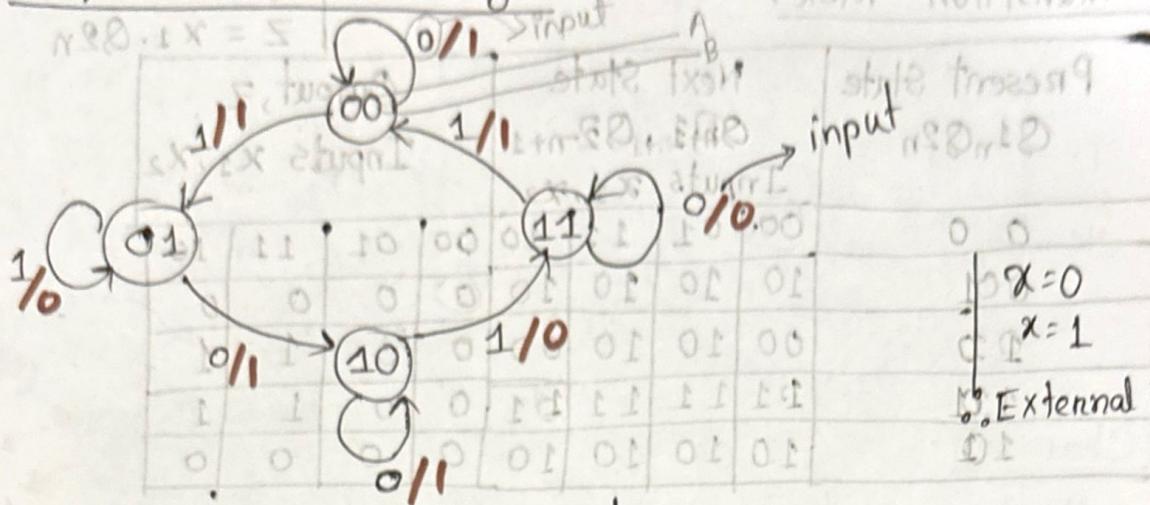
Homework : Same Diagram for T-F-F

0	F	10
X	X	01
X	X	11

Q <sub>n</sub>	Q <sub>n+1</sub>
X0	00
X1	10
1X	01
11	11

Sunday  
108 2025

## Sequence Circuit Design:



Design the sequential circuit using J-K F-F.

Excitation Table :- (Not transition table কারণ আমরা এখন input দেওয়ালে Excitation = input)

Present state $Q_{1n} Q_{2n}$	Next state, $Q_{1n+1} Q_{2n+1}$		F-F inputs	
	$x=0$	$x=1$	$J_1 n$	$K_1 n$
00	00	01	0X	0X
01	10	02n+1	1X	X1
11	11	00	X0	X0
10	10	11	X0	0X

We know, Excitation Table for JK-F-F:

$Q_n Q_{n+1}$	$J_k n$
00	0X
01	1X
10	X1
11	X0

K-MAP for  $J_1 n$ :

$Q_{1n} Q_{2n}$	0	1
00	0	0
01	1	0
11	X	X
10	X	X

$$J_1 n = Q_{2n} x'$$

K1n: K-map

	$Q_1' Q_2'$	$x$
$Q_1 Q_2$	00	X X
	01	X (X)
	11	0 1
	10	0 0

J2n K-map

	$Q_1' Q_2'$	$x$
$Q_1 Q_2$	00	0 1
	01	X X
	11	X X
	10	0 1

K2n K-map

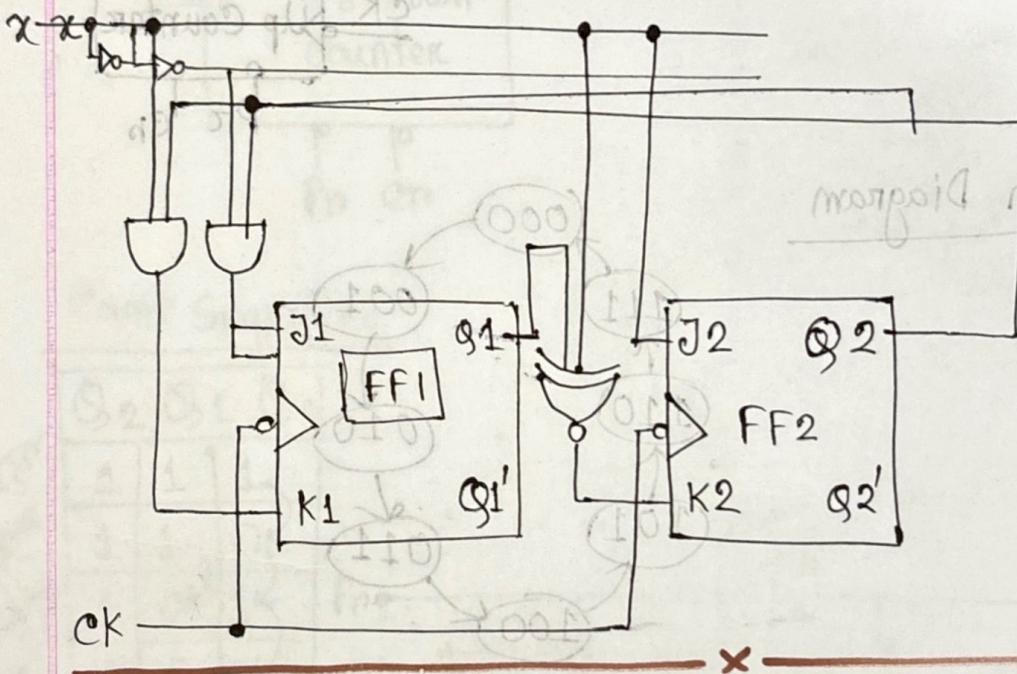
	$Q_1' Q_2'$	$x$
$Q_1 Q_2$	00	X X
	01	1 0
	11	0 1
	10	X X

$$K1n = Q2n x$$

$$J2n = x$$

$$\begin{aligned} K2n &= Q1n' x' + Q1n x \\ &= Q1n \otimes x \end{aligned}$$

Logic Diagram:



এখন output মনে করো,

Present state  $\oplus$  Output,  $z$

$Q_1n Q_2n$	$x=0$	$x=1$	$z$
00	1	1	
01	1	0	
10	0	1	
11	1	0	

(Practice: S-R, D, T for same Diagram)

$z = Q1' Q2' + Q1' n x' + Q2' n x + Q1 n Q2 n x$

$Q_1n Q_2n$	00	01	11	10
$x$	0	0	0	1
$Q_1' Q_2'$	1 1	1 0	0 1	0 0
$Q_1' n x'$	1 0	0 1	0 0	0 0
$Q_2' n x$	0 1	0 0	0 0	1 0
$Q_1 n Q_2 n x$	0 0	0 0	0 0	0 1
$z$	0	0	0	1

এই পর্যবেক্ষণ  
এ  $z$  add  
করতে হবে  
eqn অনুশীলন

Table  
সমান  
add  
করা

Tuesday  
12/08/2025

যৈকানা counter  
T-FF দিয়ে solve করবে

## □ Counters and their classification

- Based on Count Order

→ Up counter

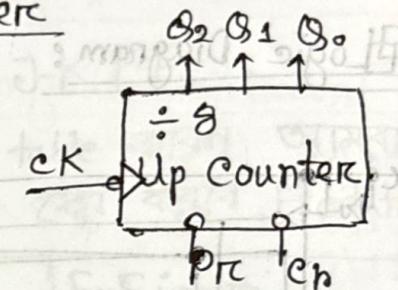
→ Down counter

- Based on Operating Mode

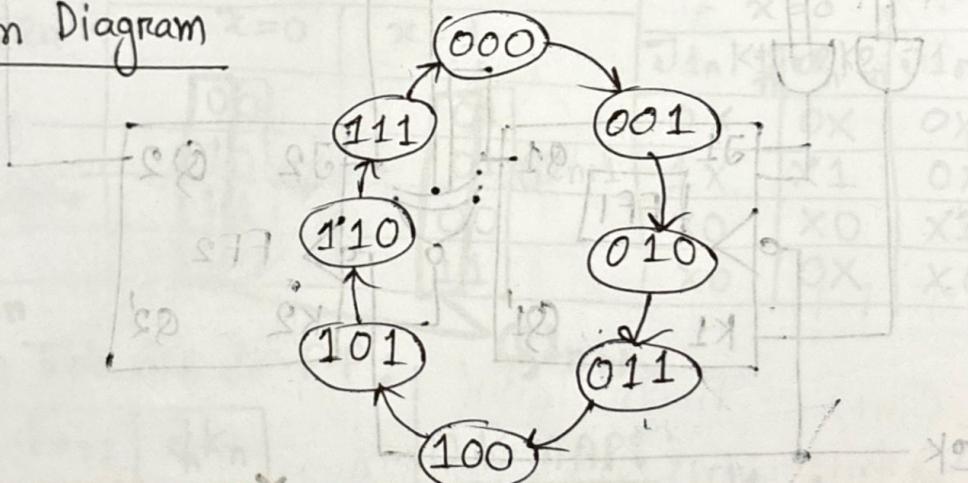
→ Asynchronous Counter (All FFs are not controlled by same clock)

→ Synchronous Counter (All FFs are controlled by same clock)

## □ Binary asynchronous $\div 8$ Up Counter



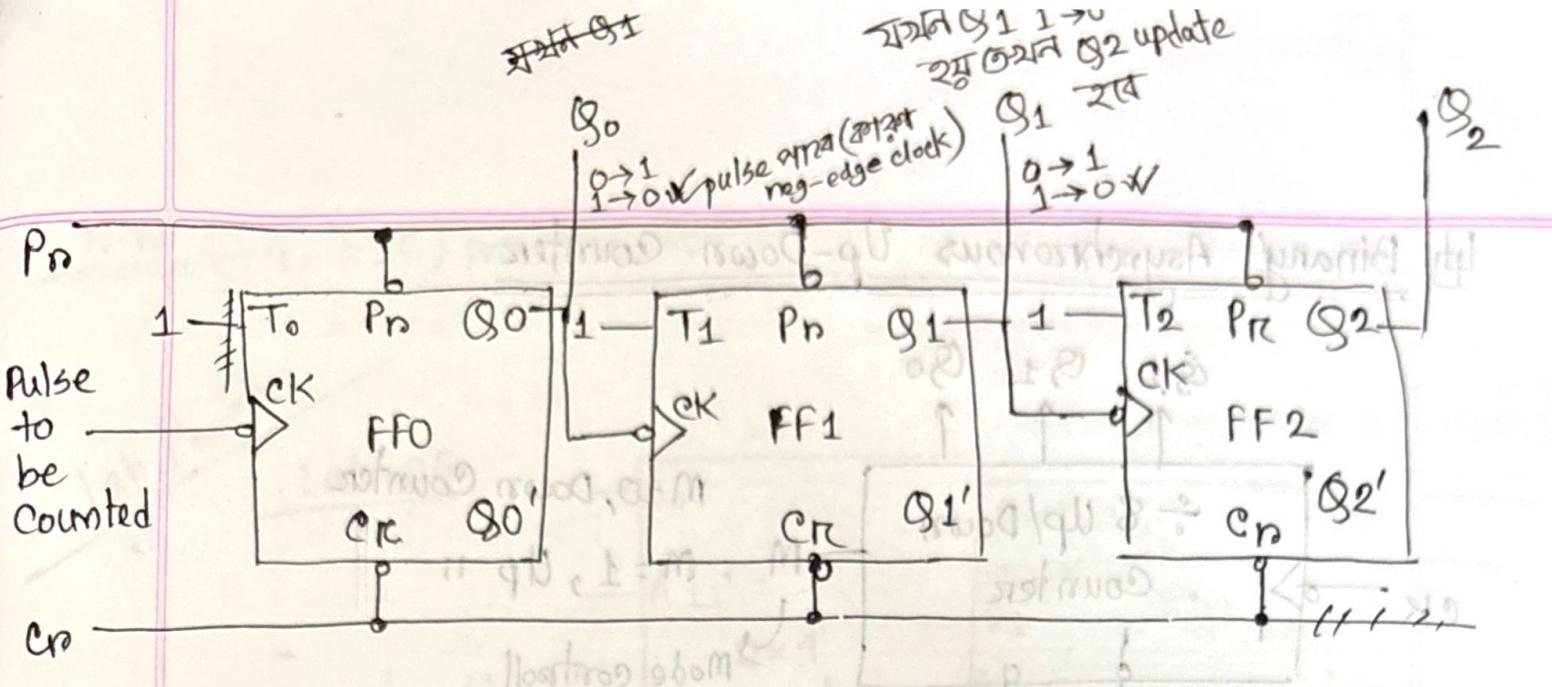
## Transition Diagram



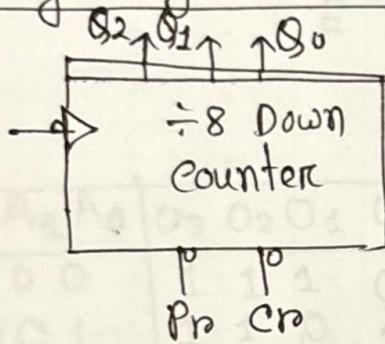
## Count Sequence

$Q_2\ Q_1\ Q_0$
0 0 0
0 0 1 ↘
0 1 0 ↘
0 1 1 ↘
1 0 0 ↘
1 0 1 ↘
1 1 0 ↘
1 1 1 ↘

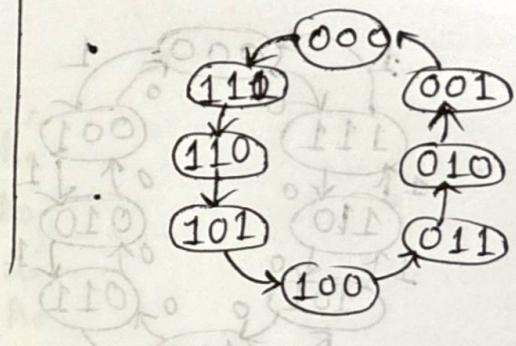
$1 \rightarrow 0$  ১ম  
change ২<sup>nd</sup>



Binary Asynchronous  $\div 8$  Down Counter:

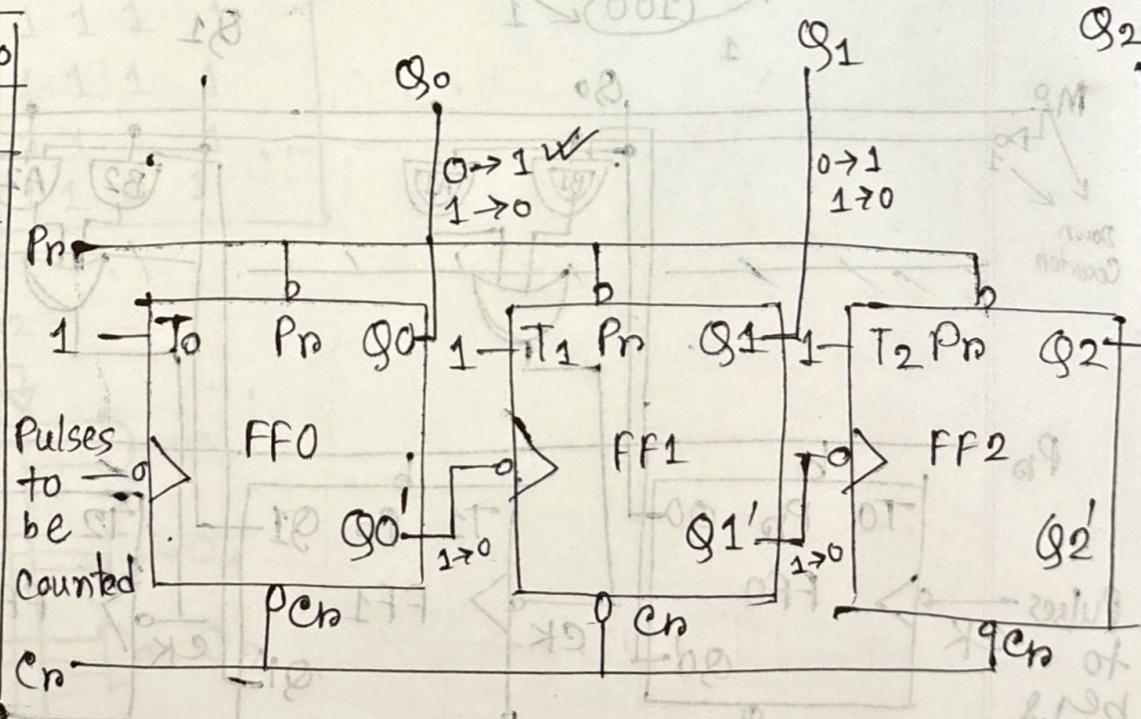


Transition Diagram

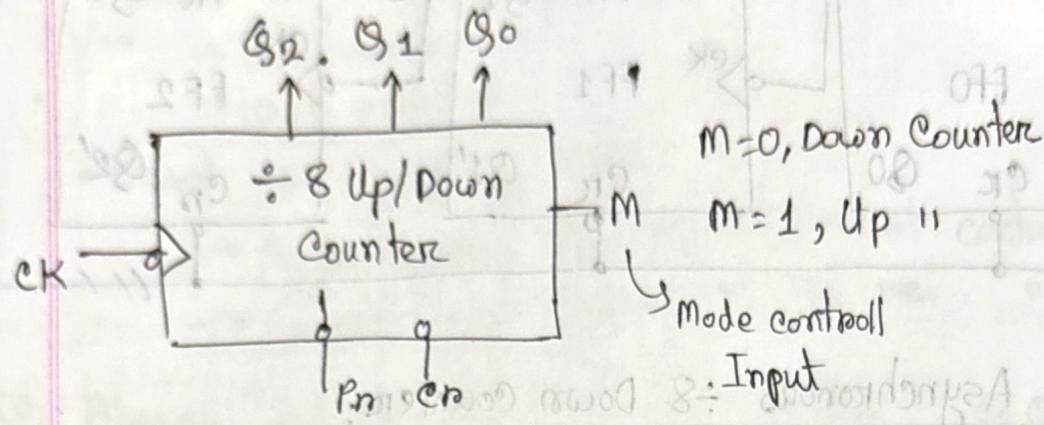


Count Sequence

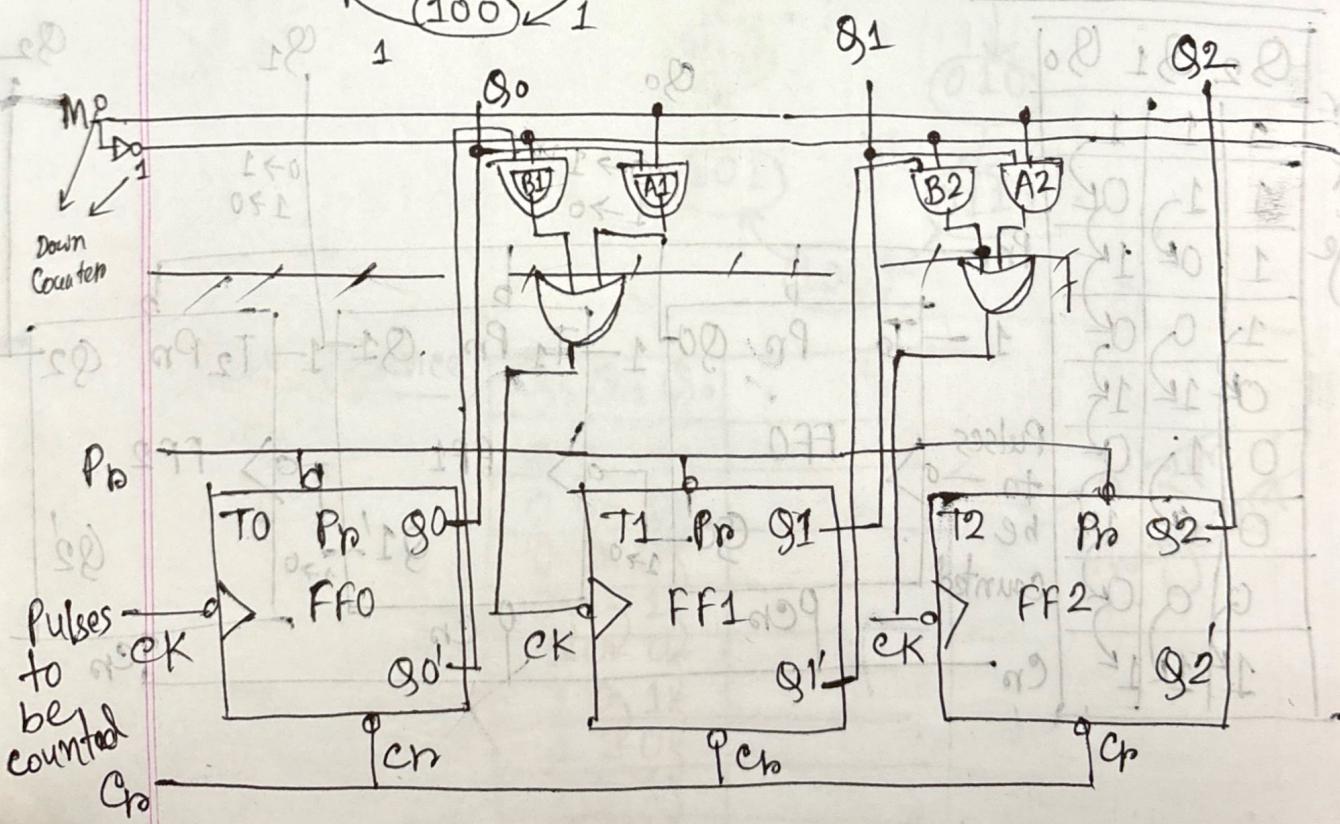
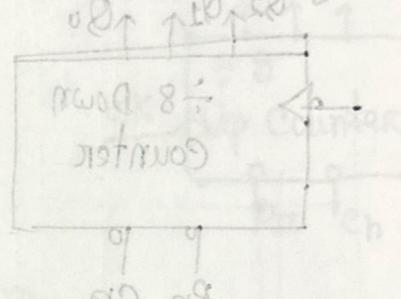
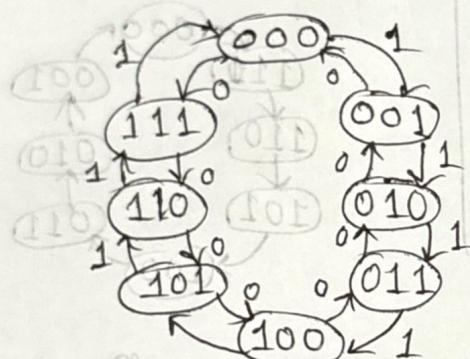
	$Q_2$	$Q_1$	$Q_0$
0 (initial)	1	1	1
1 (1st change)	1	1	0
2 (2nd change)	1	0	1
3 (3rd change)	1	0	0
4 (4th change)	0	1	0
5 (5th change)	0	0	1
6 (6th change)	0	0	0
7 (final)	1	1	1



## Binary Asynchronous Up-Down Counter



## Transition Diagram



17.8.2025  
Sunday

## Synchronous $\div 8$ Up Counter

Same as Asynchronous block and transitional diagram.

Excitation Table:

Present state	Next State	Flip-Flop Inputs
$Q_2^n Q_1^n Q_0^n$	$Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$	$T_2 \ T_1 \ T_0$
000	001	0 0 1
001	010	0 1 1
010	011	0 0 1
011	100	1 1 1
100	101	0 0 1
101	110	0 1 1
110	111	0 0 1
111	000	1 1 1

EXOR

$T_2$ :

$Q_1^n Q_0^n$			$Q_1^{n+1} Q_0^{n+1}$	
$Q_2^n$	00	01	11	10
0	0	0	1	1
1	1	1	0	0

$$T_2 = Q_1^n Q_0^n$$

$T_1$ :

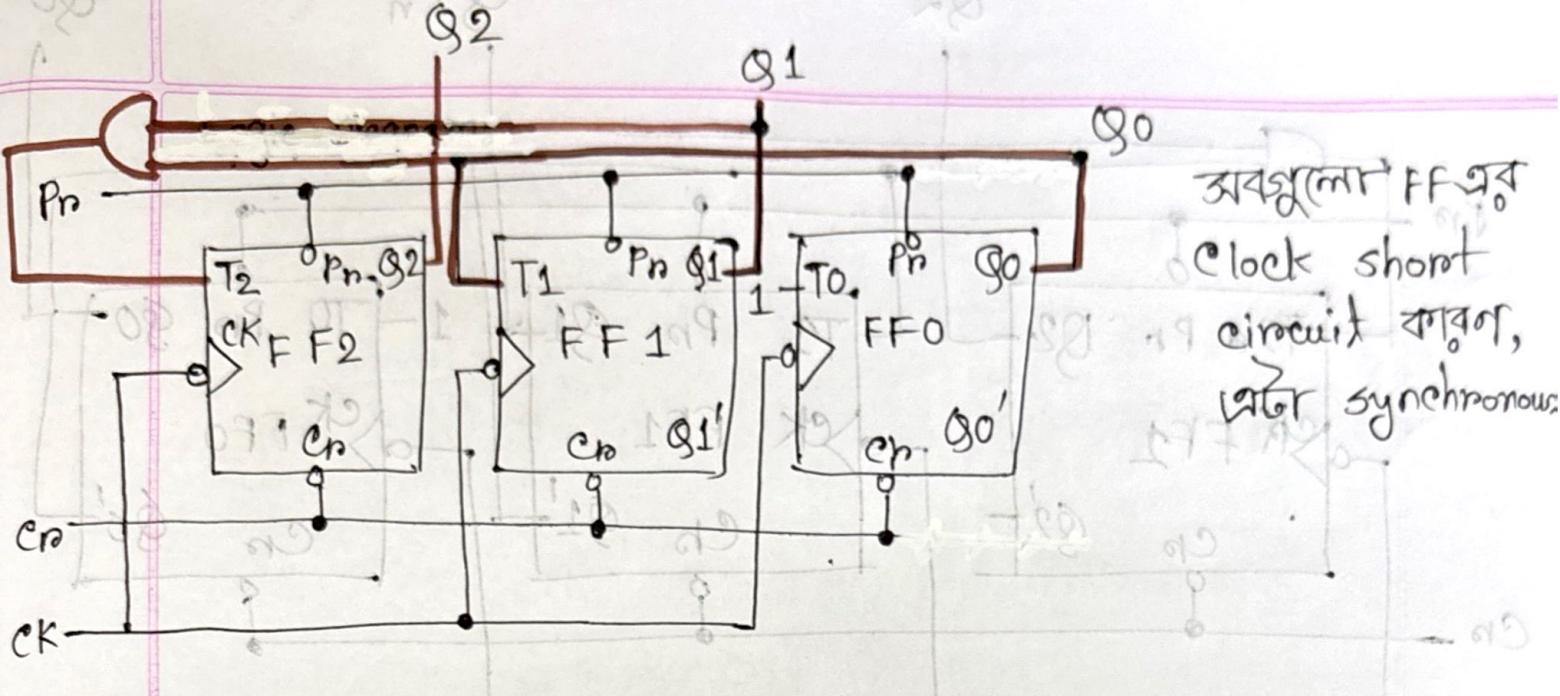
$Q_1^n Q_0^n$			$Q_1^{n+1} Q_0^{n+1}$	
$Q_2^n$	00	01	11	10
0	0	1	1	1
1	1	1	0	0

$$T_1 = Q_0^n$$

$Q_1^n Q_0^n$			$Q_1^{n+1} Q_0^{n+1}$	
$Q_2^n$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$$T_0 = 1$$

## Logic Diagram:



□ Synchronous  $\div 8$  Down Counter

Excitation Table:-

Present State	Next State	Flip Flop Inputs
$Q_2n \ Q_1n \ Q_0n$	$Q_{2n+1} \ Q_{1n+1} \ Q_{0n+1}$	$T_2 \ T_1 = T_0$
0 0 0	1 1 1	1 1 1
0 0 1	0 0 0	0 0 1
0 1 0	0 1 1	0 1 1
0 1 1	0 0 0	0 0 1
1 0 0	0 1 1	1 1 1
1 0 1	1 0 0	0 0 1
1 1 0	1 0 1	0 1 1
1 1 1	1 1 0	0 0 1

$Q_2n$	00	01	11	10
$Q_1n$	1	1		
$Q_0n$	1			

$$T_2 = Q_1n Q_0n$$

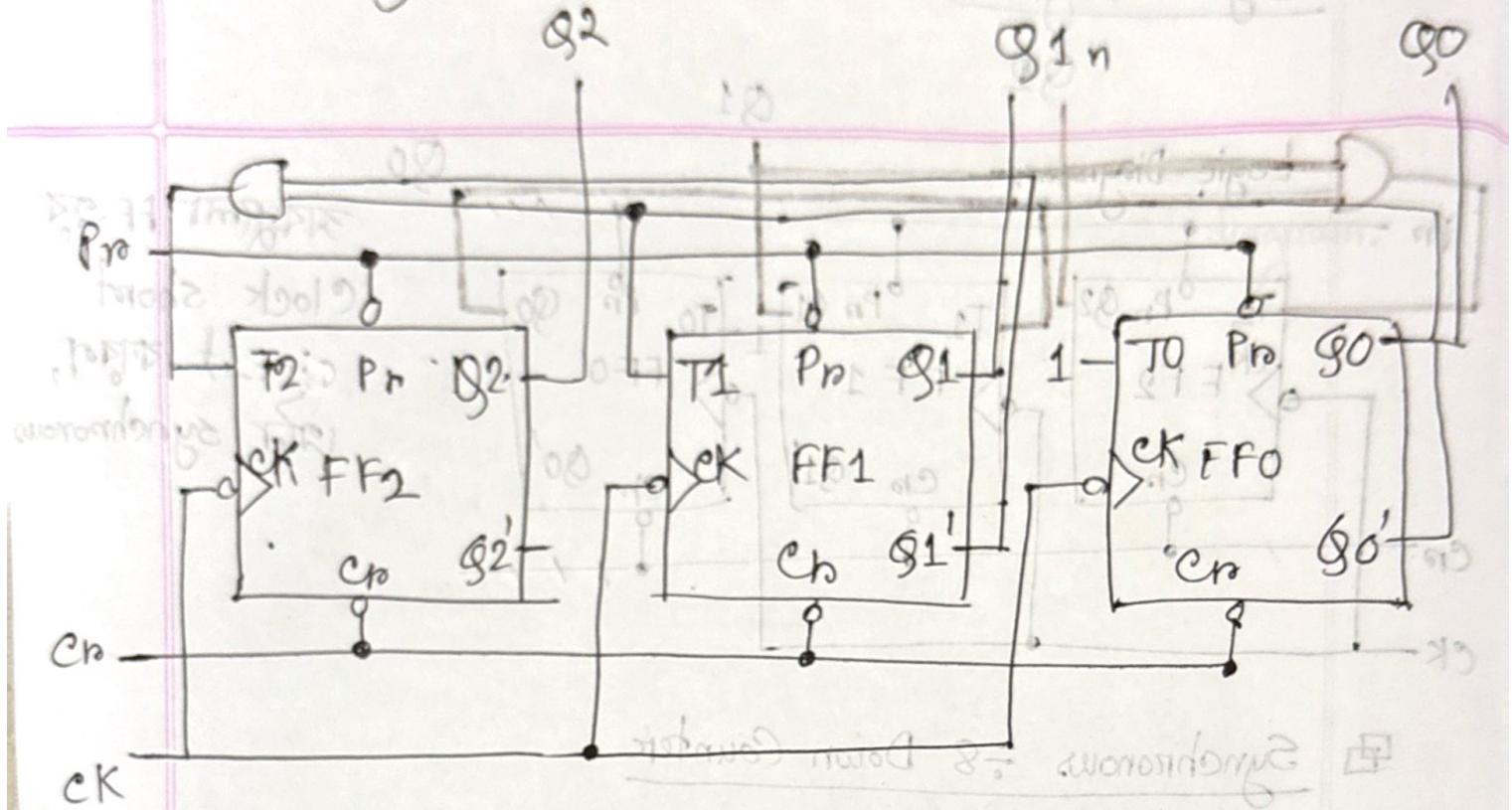
$Q_2n$	00	01	11	10
$Q_1n$	1	1	1	1
$Q_0n$	1	1	1	1

$$T_1 = Q_0n$$

$T_0$  :  
Same as previous

$$T_0 = 1$$

# Logic Diagram:



D)  $M=1$ , Up

$M=0$ , Down

□ Synchronous  $\div 8$  Up/Down Counter for K MAP

Excitation Table:

Present State $Q2n\ Q1n\ Q0n$	Next State $Q2_{n+1}\ Q1_{n+1}\ Q0_{n+1}$	Flip-Flop $M=0$	Input, T2 T1 T0 $M=1$
0 0 0	1 1 1    0 0 0 1	1 1 1	0 0 1
0 0 1	0 0 0    0 1 0	0 0 1	0 1 1
0 1 0	0 0 1    0 1 1	0 0 1	0 0 1
0 1 1	0 1 0    1 0 0	1 0 1	0 0 1
1 0 0	0 1 1    1 0 1	0 0 1	0 1 1
1 0 1	0 1 0 0    1 1 0	0 1 1	0 0 1
1 1 0	1 0 1    1 1 1	0 0 1	1 1 1
1 1 1	1 1 0    0 0 0	1 1 1	

T2:

		Q0nM			
		00	01	11	10
Q2nQ1n		1			
00					
01		1			
11		1			
10	1				

$Q1'Q0'M'$

T1:

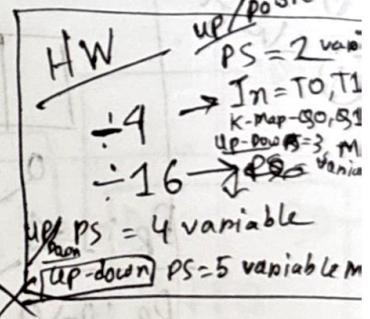
		Q0M			
		00	01	11	10
Q2n	1				
00		1			
01		1	1		
11		1	1	1	
10	1	1	1	1	1

$$\begin{aligned} T2 &= Q1'Q0'nM' + Q1nQ0nM \\ &= Q1_n \otimes Q0_n \otimes M \end{aligned}$$

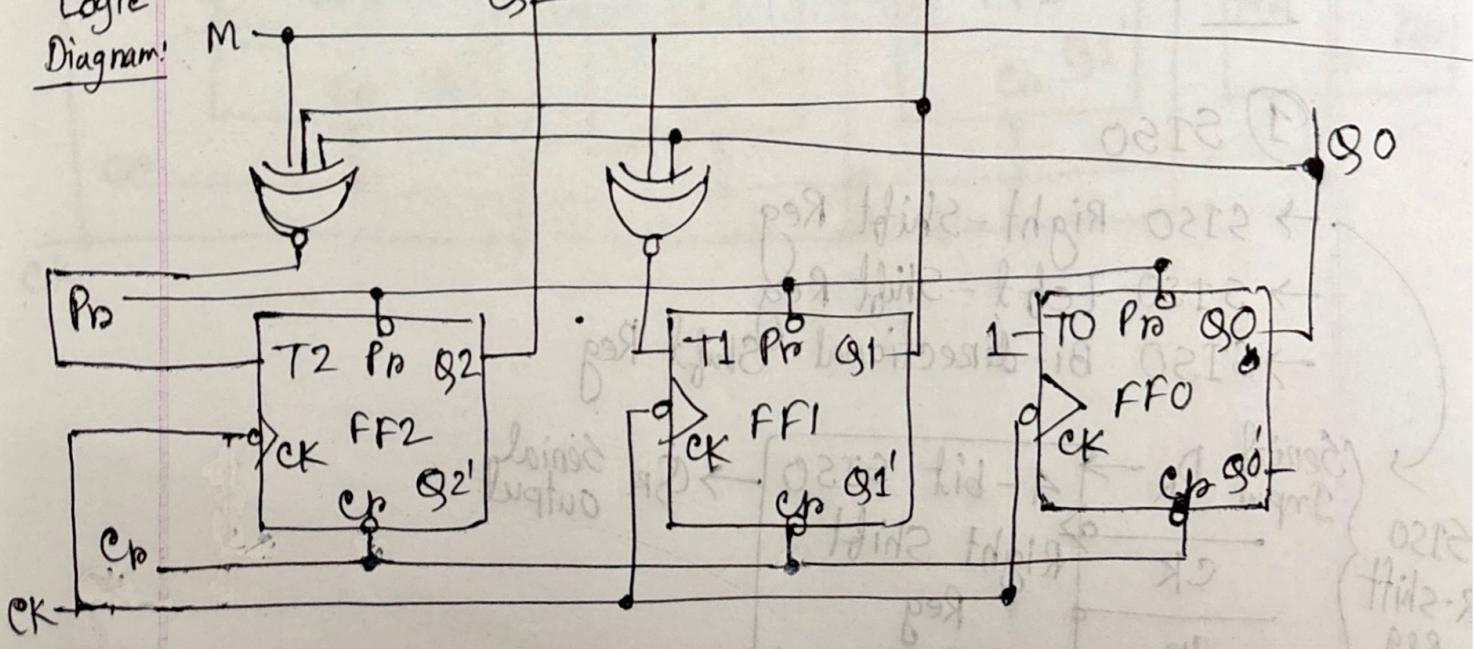
$$\begin{aligned} T1 &= Q0'M' + Q0_nM \\ &= Q0_n \otimes M \end{aligned}$$

		Q0M			
		00	01	11	10
Q2nQ1n	00	1	1	1	1
00		1	1	1	1
01		1	1	1	1
11		1	1	1	1
10	1	1	1	1	1

$$T0 = 1$$



Logic Diagram:



19.08.25  
Tuesday

Register 2Lm  
always design using  
D Flip Flop

## Registers and their classification:-

① Serial-In Serial-out (SISO)

② Serial-In Parallel-out (SIPO)

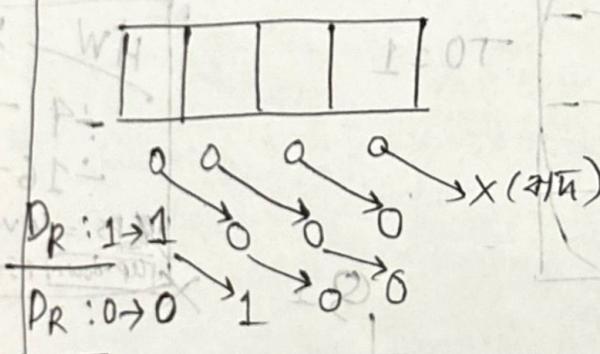
③ Parallel-In " " (PIPO)

④ Parallel-In Serial-out (PISO)

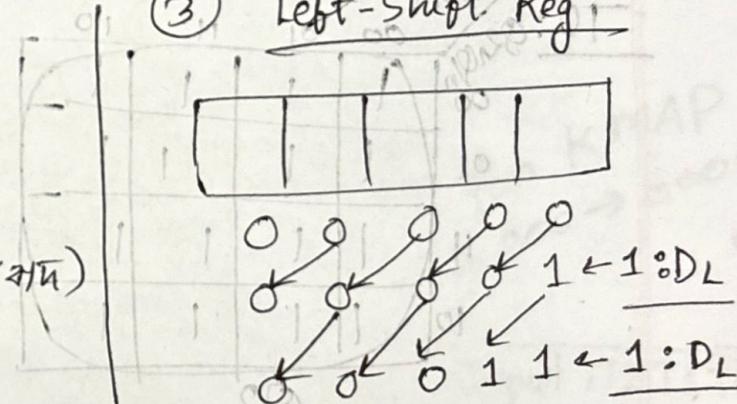
⑤ যখন Input অথবা Output এ Serial করাবে  
তখন দ্রুত প্রক্রিয়ার Reg হবে, (যমানঃ →)

Bidirectional  
Shift

② Right-Shift Reg



③ Left-Shift Reg

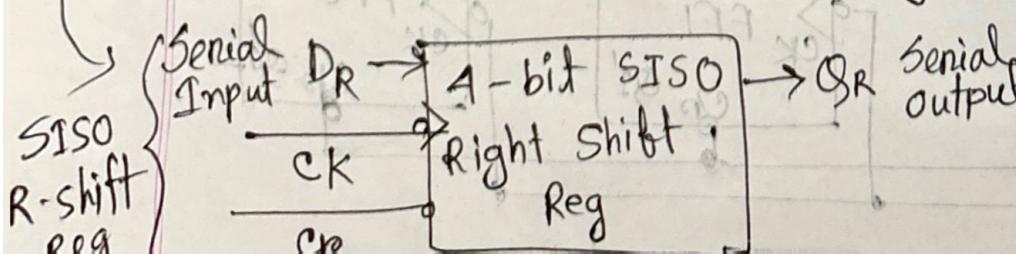


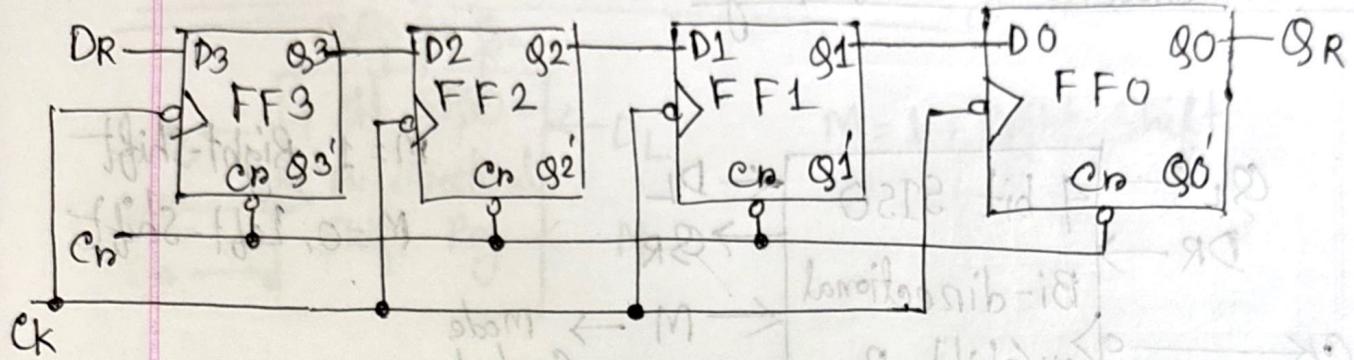
① SISO

→ SISO Right-Shift Reg

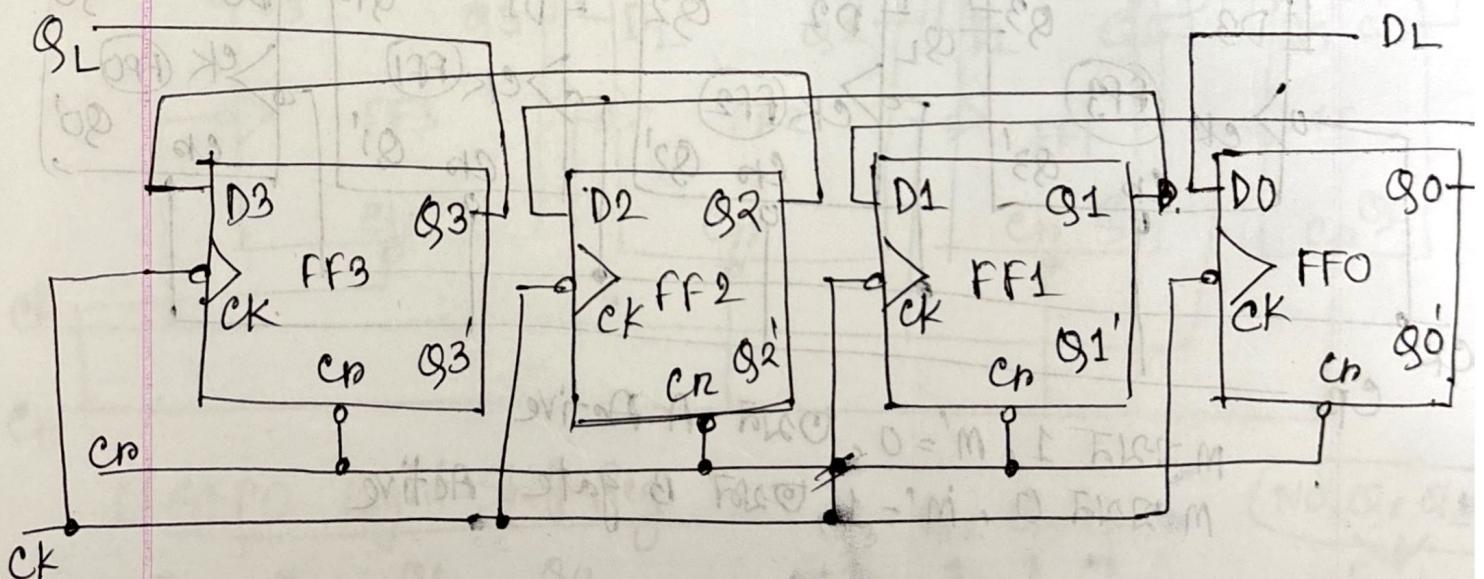
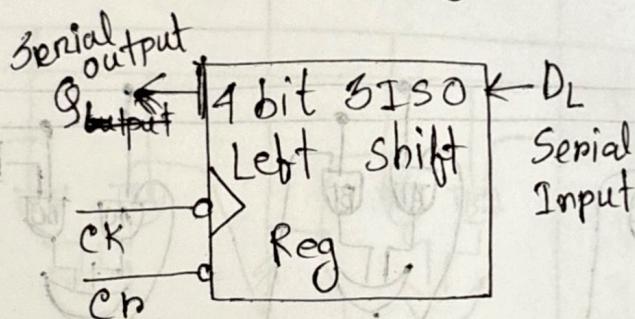
→ SISO Left-Shift Reg

→ SISO Bi-directional Shift Reg

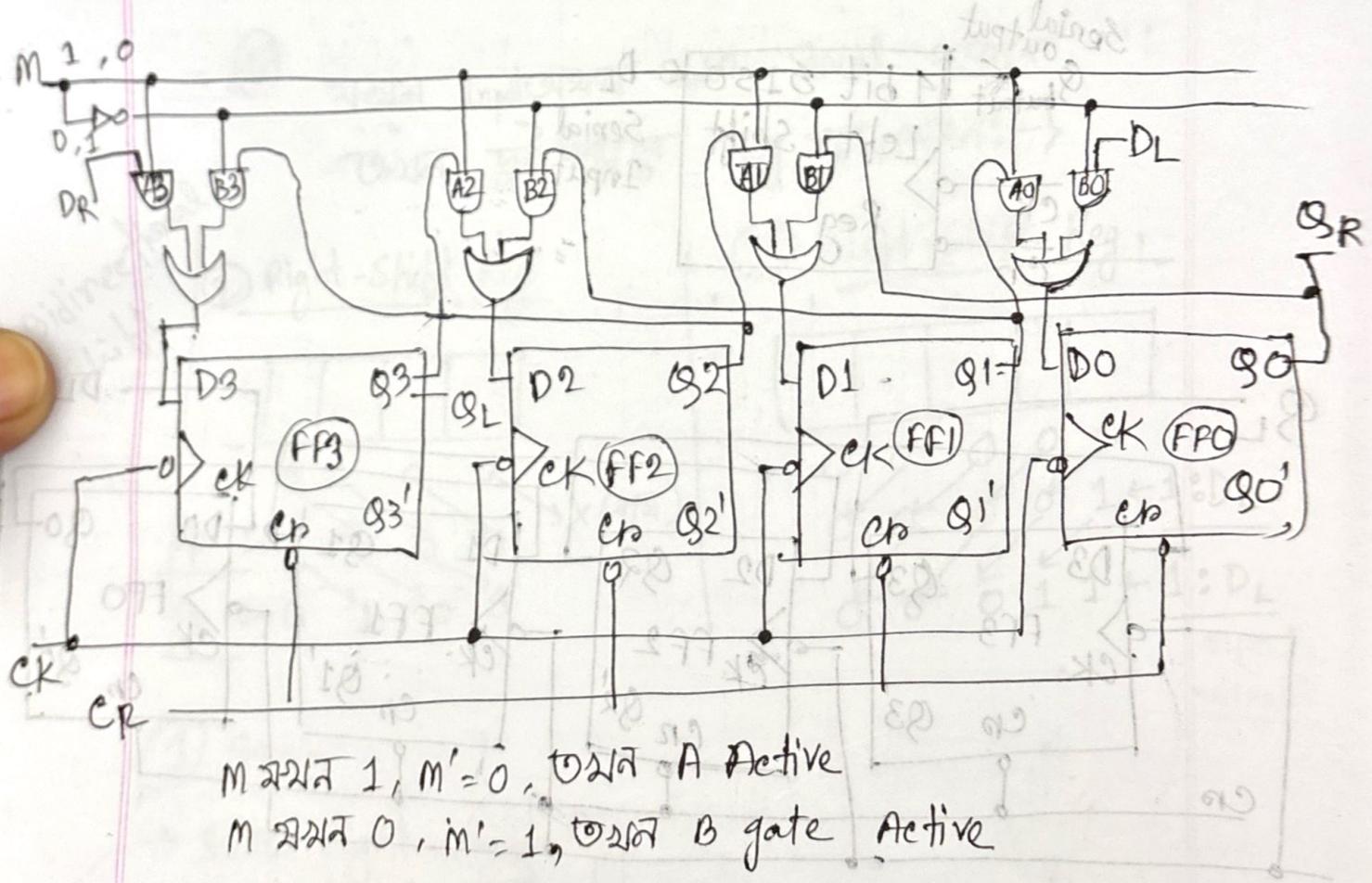
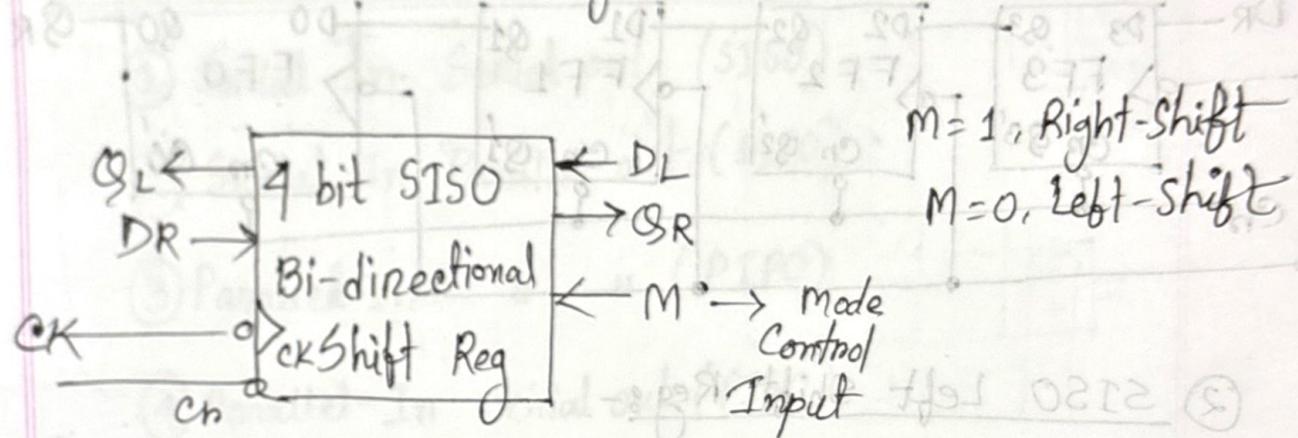




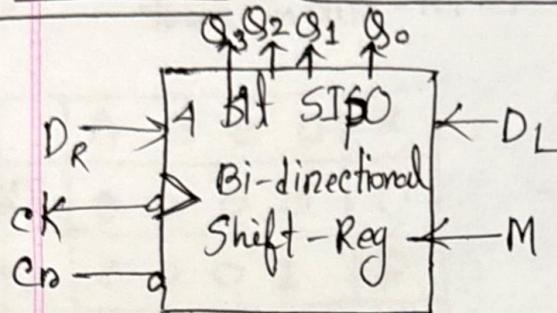
② SISO Left Shift Reg :-



## SISO Bi-directional shift Reg

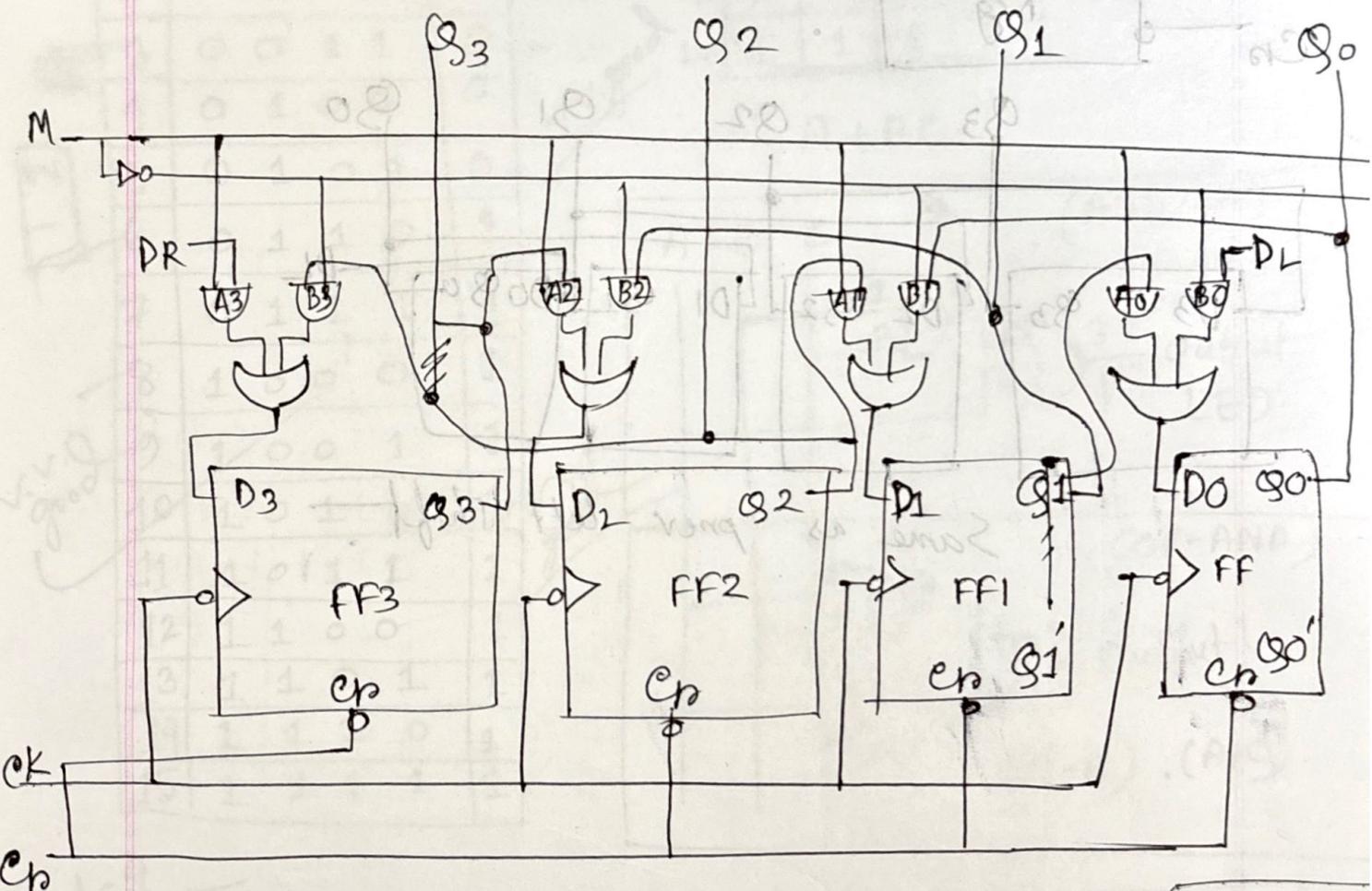


## SIPO : Bi-directional Shift Reg

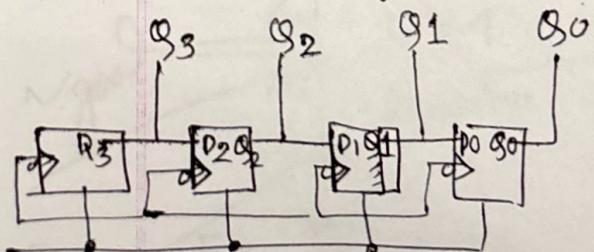


M = 1, Right Shift

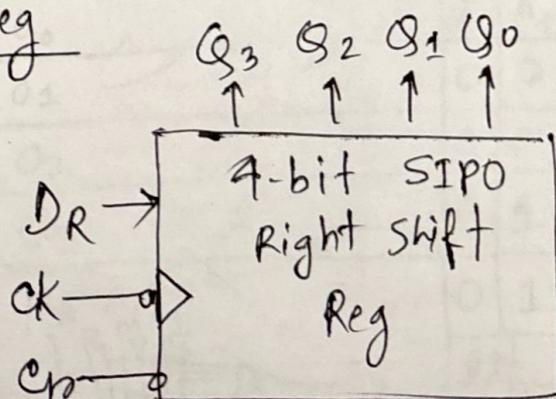
M = 0, Left " "



## SIPO Right-Shift Reg



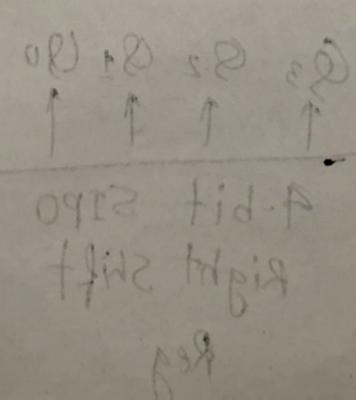
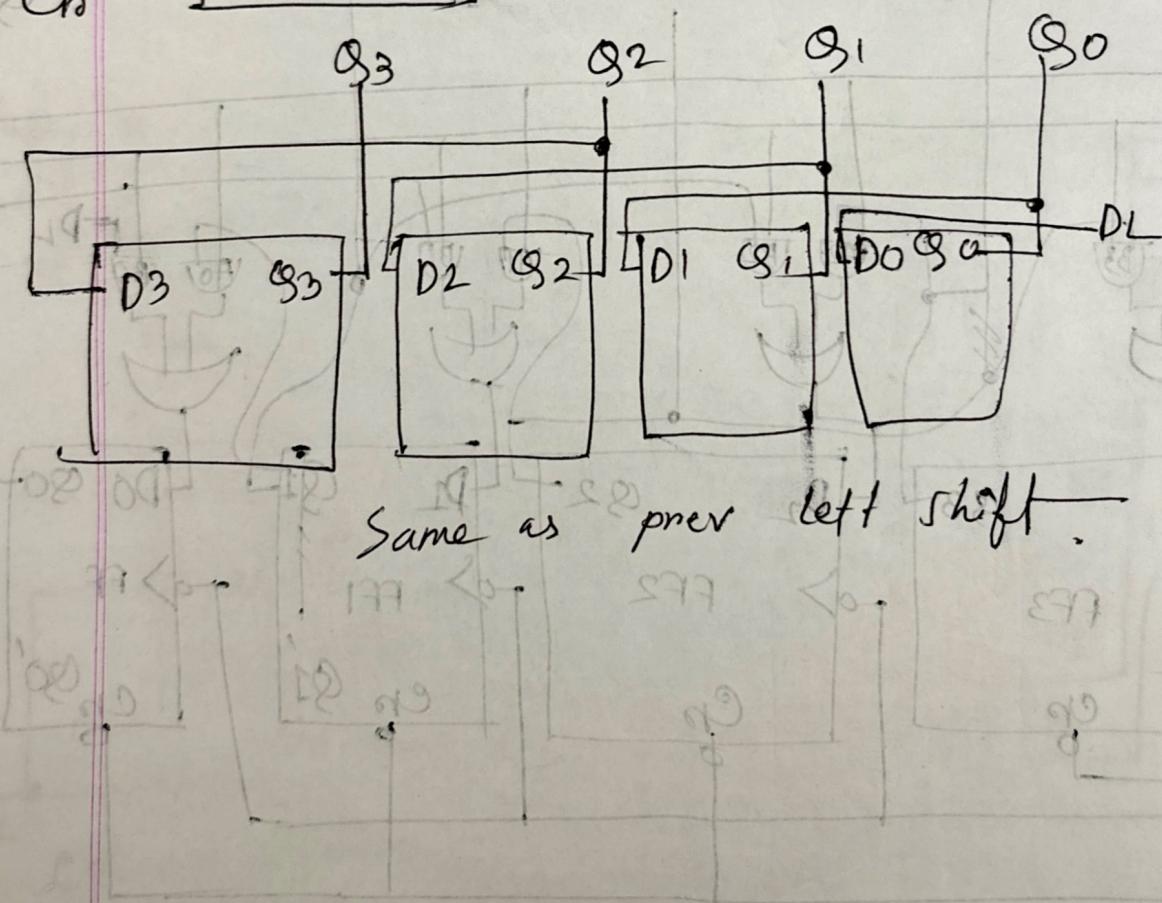
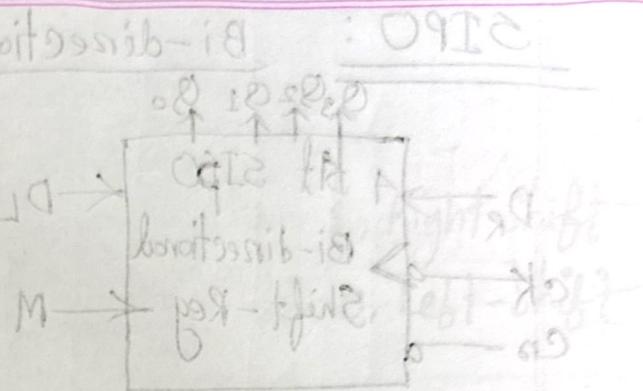
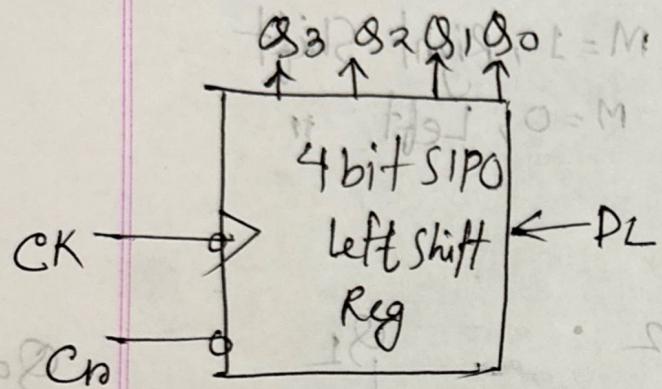
Same as pre R-shift



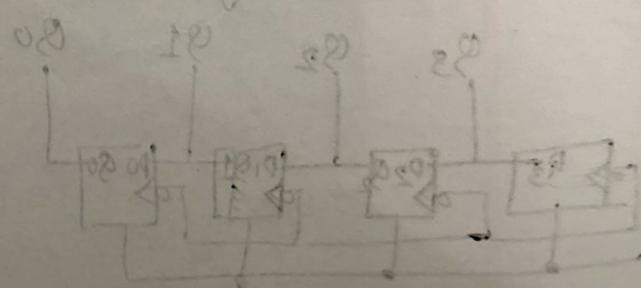
NO Q<sub>R</sub>, Q<sub>L</sub>

Serial Output

## SIPO left shift Reg

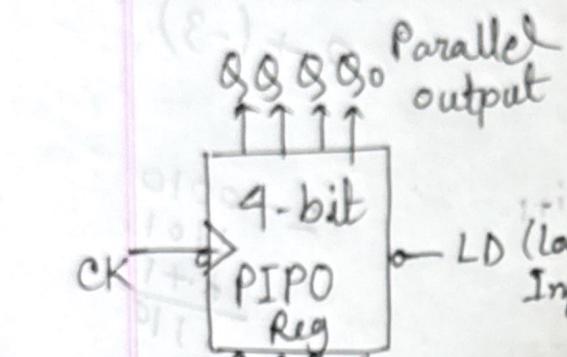


4 bit SIPO right shift Reg

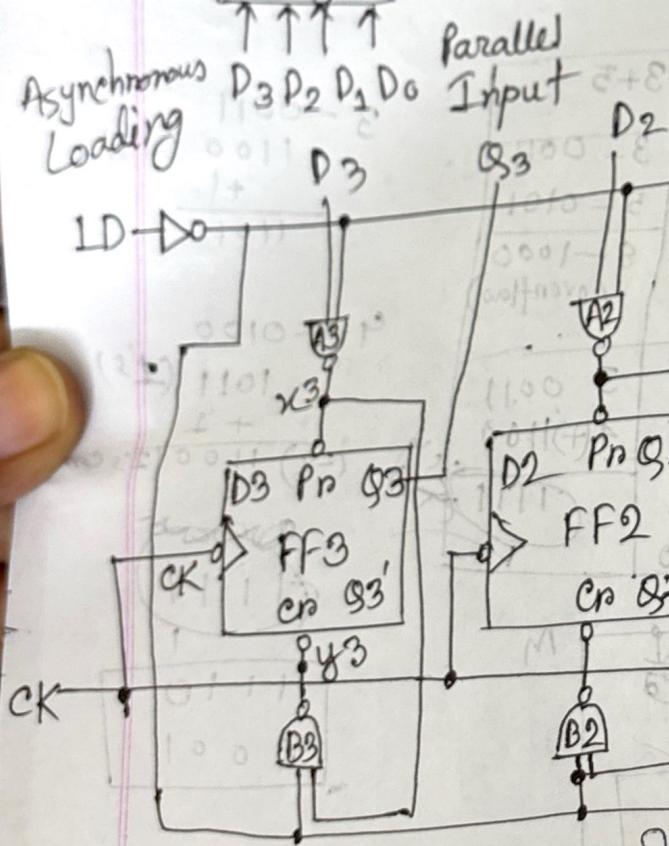


C.W  
08/2025

### Parallel-In, Parallel-out:



Load Input  $D_0$   
Parallel Input  $D_2, D_1, D_0$   
Input  $D_3$   
Input  $D_2$   
 $D_1$ ,  $D_0$   
CK active  $D_3, D_2, D_1, D_0$   
 $D_2, D_1$ ,  $D_0$   
CK active  $D_3, D_2, D_1, D_0$ ,  
LD = 1 then  $D_3, D_2, D_1, D_0$   
output  $D_3, D_2, D_1, D_0$



for FF3 , (FF এর মূল্যগতি same)

$$LD = 0$$

$$X_3 = (LD', D_3')' = (0', D_3)' = (1 \cdot D_3)' = D_3'$$

$$\therefore \text{Preset} = X_3 = D_3'$$

$$Y_3 = (LD', X_3)' = (0', X_3)' = (1 \cdot D_3)' = D_3$$

$$\therefore \text{Clear} = Y_3 = D_3.$$

তাত্ত্বিকভাবে  $D_3$  এর পরিষেবা ওপৰের অক্ষের উপরে, যদি  $D_3$  হয় তবে

Output:

D3	Pn	Cn	Q3
0	1	0	$C_n = 0$ (initial Cn active low)
1	0	1	$Q_3 = P_n$ (initial Pn active low)

PIPO ହୁଏ କାନ୍ତି right-shift / left-shift (57)  
(PISO କାନ୍ତି)

Project

Truth table, K-map, logic diagram

Verilog code (behavioural)

\* Structural

Verilog code

\* Behavioral

Verilog code → circuit truth table  
input-behaviour

$$S = A'B + AB'$$

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Procedural model  
Continuous assign Statement

Procedural Model code :-  
module expt 3-1(input A,B,  
output S);

always @ (A,B) begin

S=0;

if( $\sim A \& B$ ) S=1;

if( $A \& \sim B$ ) S=1;

end

end module

S কে unconditionally 1 ধরতে চাহিলে, code:-

code:

```
module expt3-1(input A,B,output reg S);
always@ (A,B) begin
S=1;
if (~A & ~B) S=0;
if (A & B) S=0;
end
endmodule
```

□ waveform

□ Continuous assign Statement:-

~~SOP~~ module expt3-2(input A,B, output S);
assign S = (~A & B) | (A & ~B);
end module

SOP		S = A'B + AB'
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

~~POS~~ module expt3-2(input A,B, output S);
assign S = (A | B) & (~A | ~B);
end module

$$\begin{aligned} S &= (A+B) \cdot \\ & (A'+B') \end{aligned}$$

□ waveform

	A	B	S
0	0	0	0
1	1	0	1
1	0	1	1
0	1	1	0

$$i_1 = 0 \quad (A \& A' \&)$$

$$i_2 = 1 \quad (B \& B' \&)$$

b10

elaborate

108 / 2023

## Chap - 14

### Procedural Modeling of Clocked Sequential Circuit:

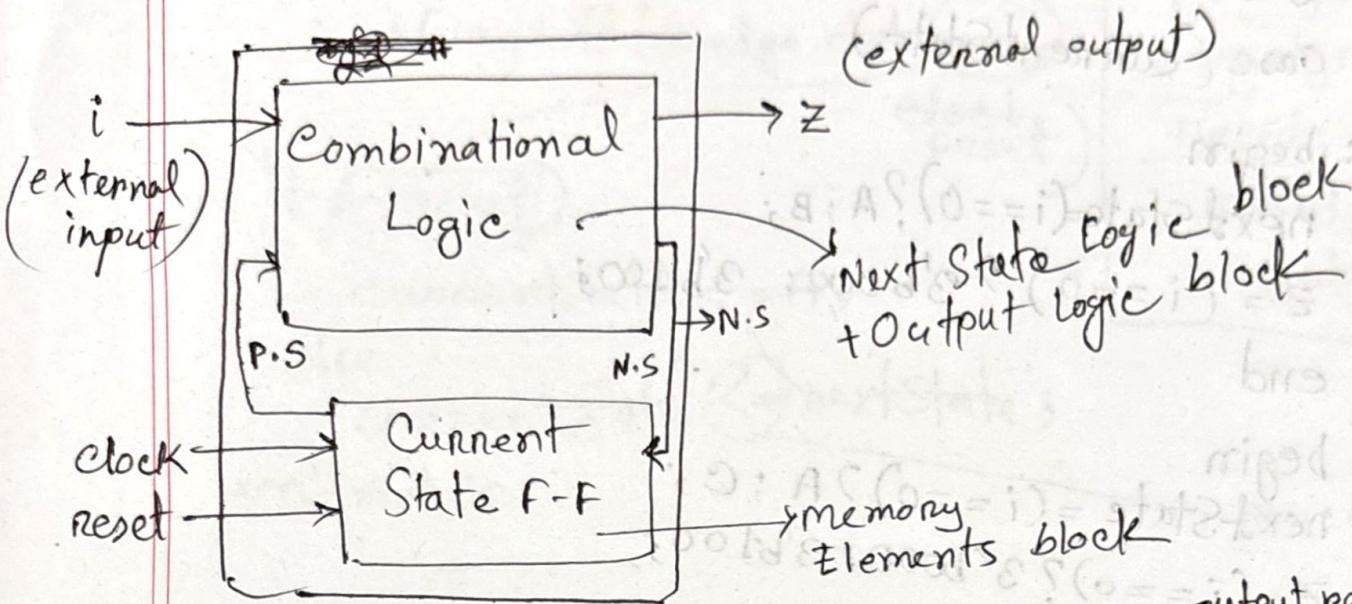
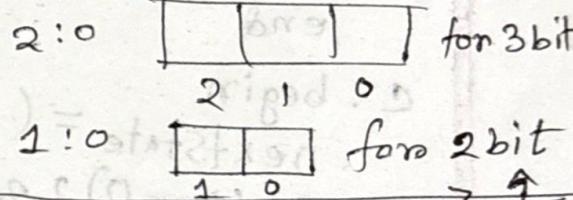
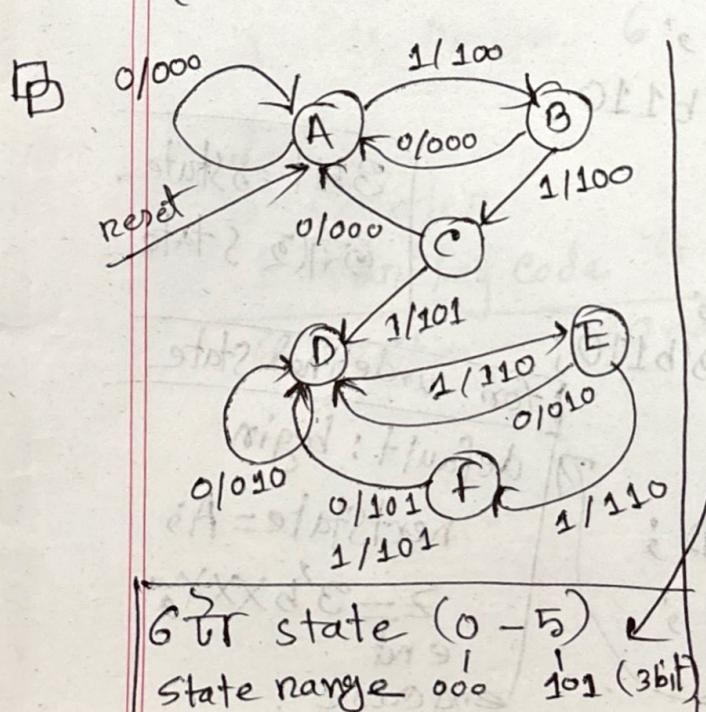


Fig: Model of Finite State Machine



Question 7 Transition diagram প্রয়োগ যাচাবে, যেটা

সূচিতে হবে।



```
module fsm (input i, clock, reset,
            output reg [2:0] z);
    reg [2:0] currentState, nextState;
    localparam [2:0] A = 3'b000,
                    B = 3'b001,
                    C = 3'b010,
                    D = 3'b011,
                    E = 3'b100,
                    F = 3'b101;
```

[PT.O]

for combinational logic block,

always @ (\*)

case (currentState)

A : begin

nextState(i==0)? A : B;

Z = (i==0)? 3'b000; 3'b100;

end

B : begin

nextState = (i==0)? A : C;

Z = (i==0)? 3'b000; 3'b100;

end

C : begin

nextState = (i==0)? A : D;

Z = (i==0)? 3'b000; 3'b101;

end

D : begin

nextState(i==0)? D : E;

Z = (i==0)? 3'b010; 3'b110;

end end;

E : begin

nextState(i==0)? D : F;

Z = (i==0)? 3'b010; 3'b110;

end;

F : begin

nextState = (i==0)? D ;

Z = (i==0)? 3'b101 ;

end;

3bit = 8 state

6 to 8 state

for undefined state

default : begin

nextState = A;

Z = 3'bXXX;

end

endcase

## Current state block (part)

Continue code:

```

    always @ (posedge clock, negedge
    clock
    reset)
        if (~reset)
            currentState <= A;
        else
            currentState <= nextState;
    endmodule

```

Ques 3  
Ch 202021

clock:

posedge clock

negedge clock

reset:

posedge reset

negedge reset

non-blocking  
assignment

Final marks distribution:

30 marks (6 Ques)

1 - MUX

5

1 - Seq. Analysis

6

1 - Seq. Design

6.

1 - Reg

1 - Counter

1 - Verilog Code

4

6 Question — 30 marks

