



**EC/EE/CS/IN**

# Digital Electronics

**MAHA  
REVISION**

**PART 2**



**Chandan Jha Sir (CJ Sir)**

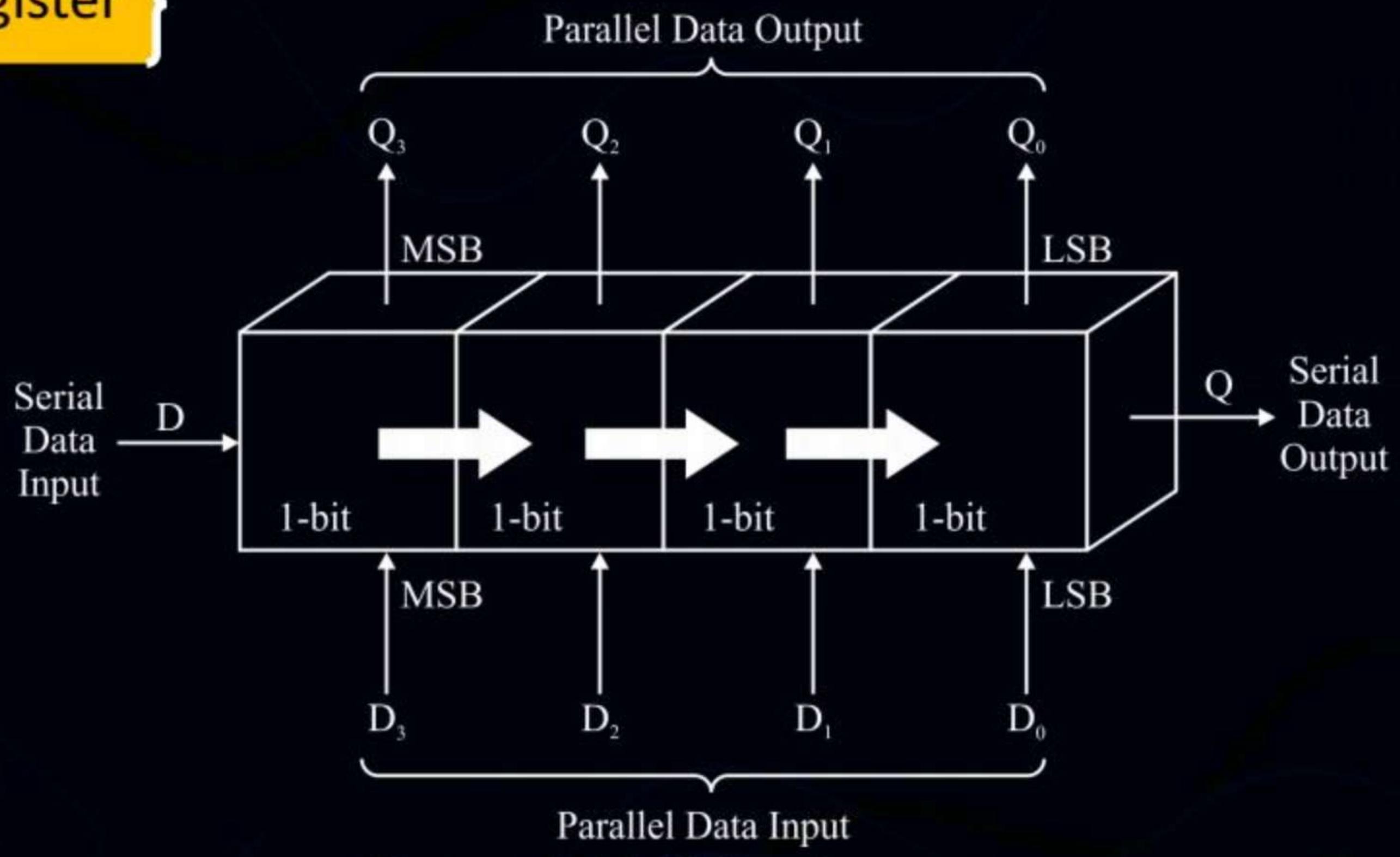
# Today's Targets



- ✓ Registers.
- ✓ COUNTER
- ✓ NUMBER SYSTEM
- ✓ FSM
- ✓ ADC/DAC ✓



# Shift Register



**Figure 1: Shift Registers**

## Shift Register

1. Registers are used to store group of bits ✓
2. To store 'n' bits minimum 'n' Flip Flips are required
3. Generally D Flip Flops are used to Design Register

## Shift Register }

1. Serial input serial output shift register (  $\text{SISO}$  )
2. Serial input parallel output shift register (  $\text{SIPo}$  )
3. Parallel input serial output shift register (  $\text{PISo}$  )
4. Parallel input parallel output shift register (  $\text{PIPo}$  )

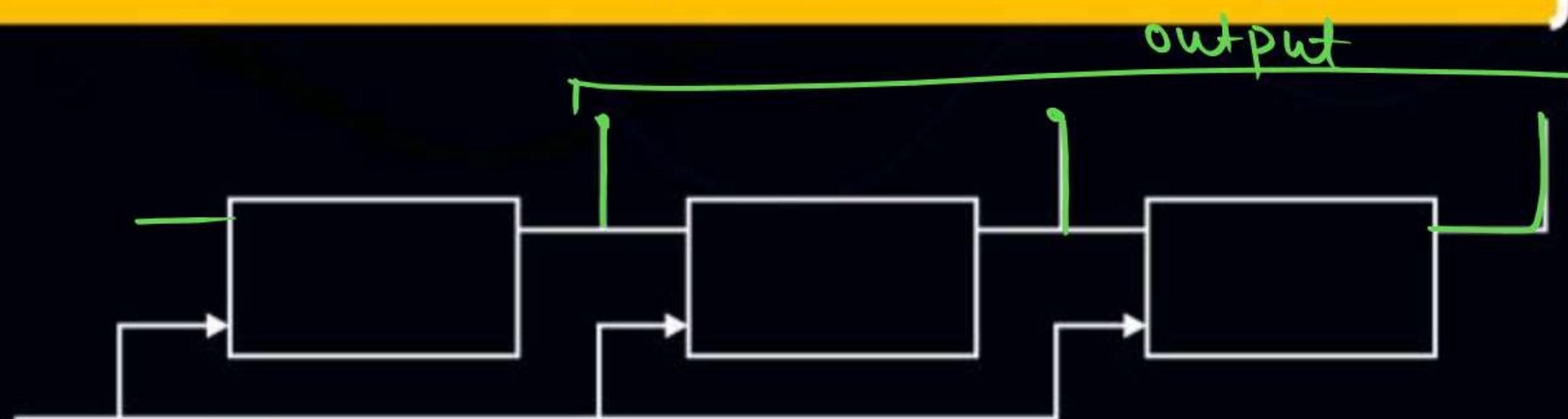
# Serial Input Serial Output (SISO) Shift Register



Store  $\rightarrow n \cdot T_{\text{CLK}}$

Retrieve  $\rightarrow (n-1)T_{\text{CLK}}$

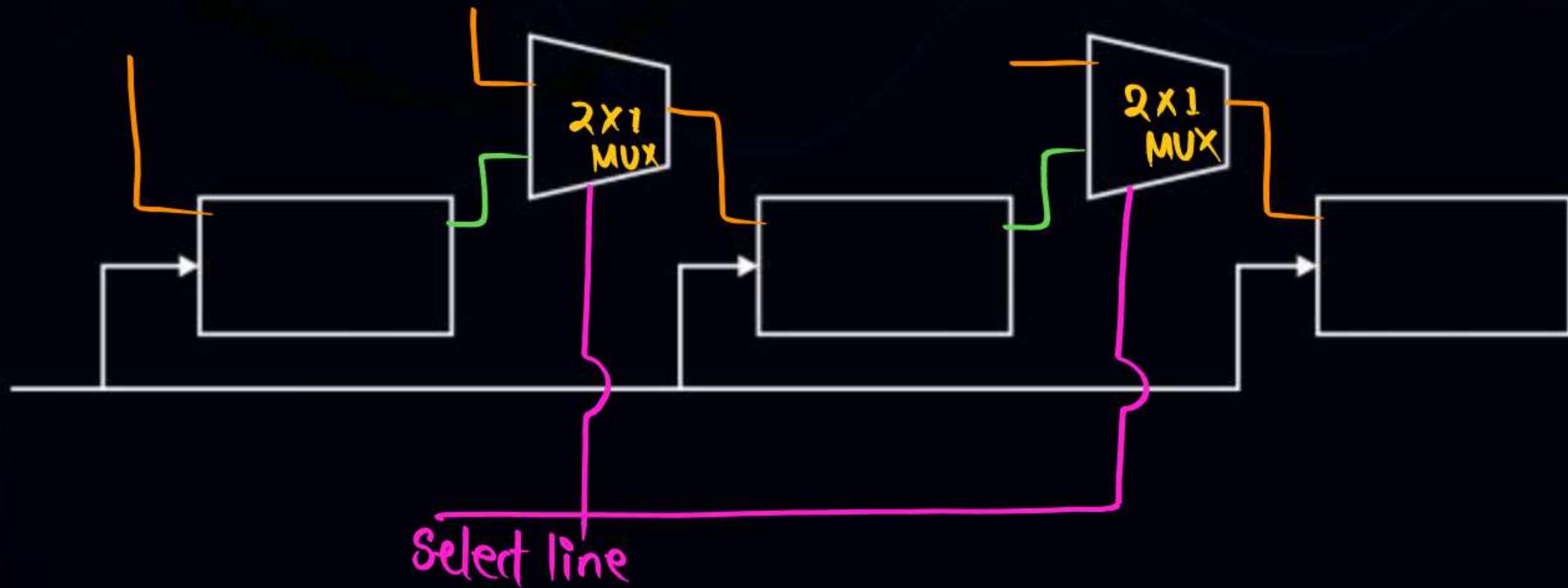
# Serial Input Parallel Output (SIPO) Shift Register



$$\text{Store} = n \cdot T_{\text{clk}}$$

$$\text{Retribe} = 0 \cdot T_{\text{clk}}$$

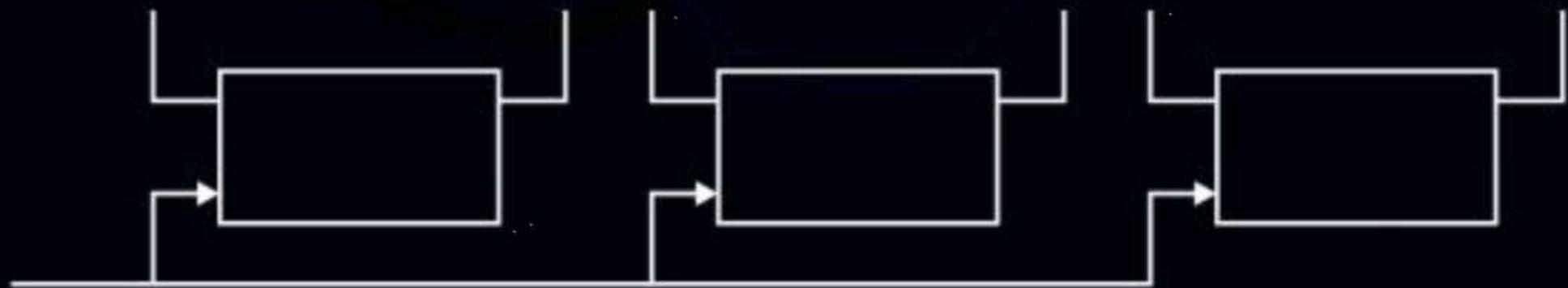
# Parallel Input Serial Output (PISO) Shift Register



$$S_{\text{store}} = t \cdot T_{\text{clk}}$$

$$R_{\text{receive}} = (n-1)T_{\text{clk}}$$

# Parallel Input ~~Serial~~ Output (PISO) Shift Register



Store = 1· $\tau_{clk}$

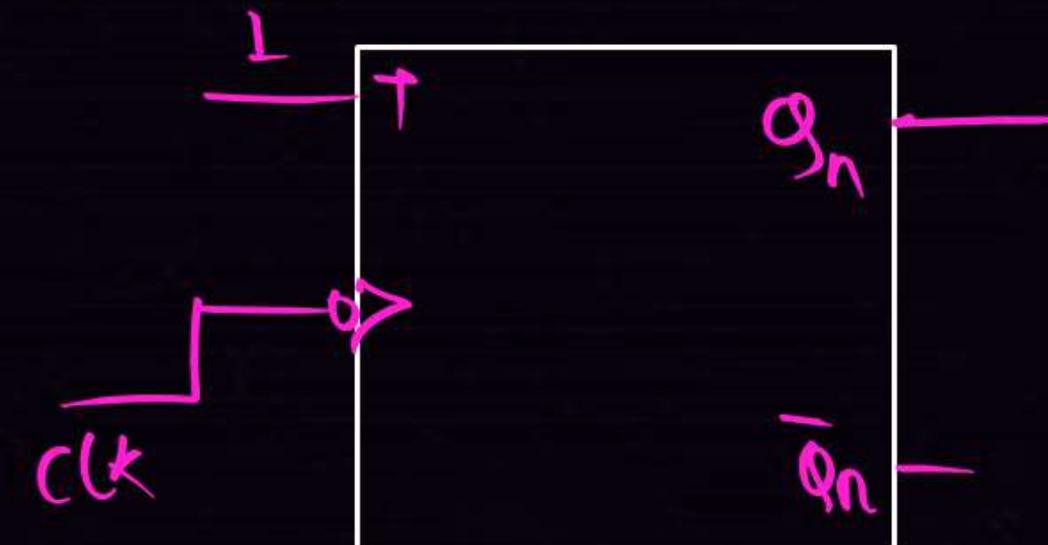
Retrige = 0· $\tau_{clk}$

# Parallel Input Serial Output (SISO) Shift Register

	Store	Retrive	Total	
SISO	$n$	$n-1$	$2n-1$	→ slowest
SIPO	$n$	0	$n$	
PISO	1	$n-1$	$n$	
PIPO	1	0	1	→ fastest

## Toggle Mode.

① By T-FF

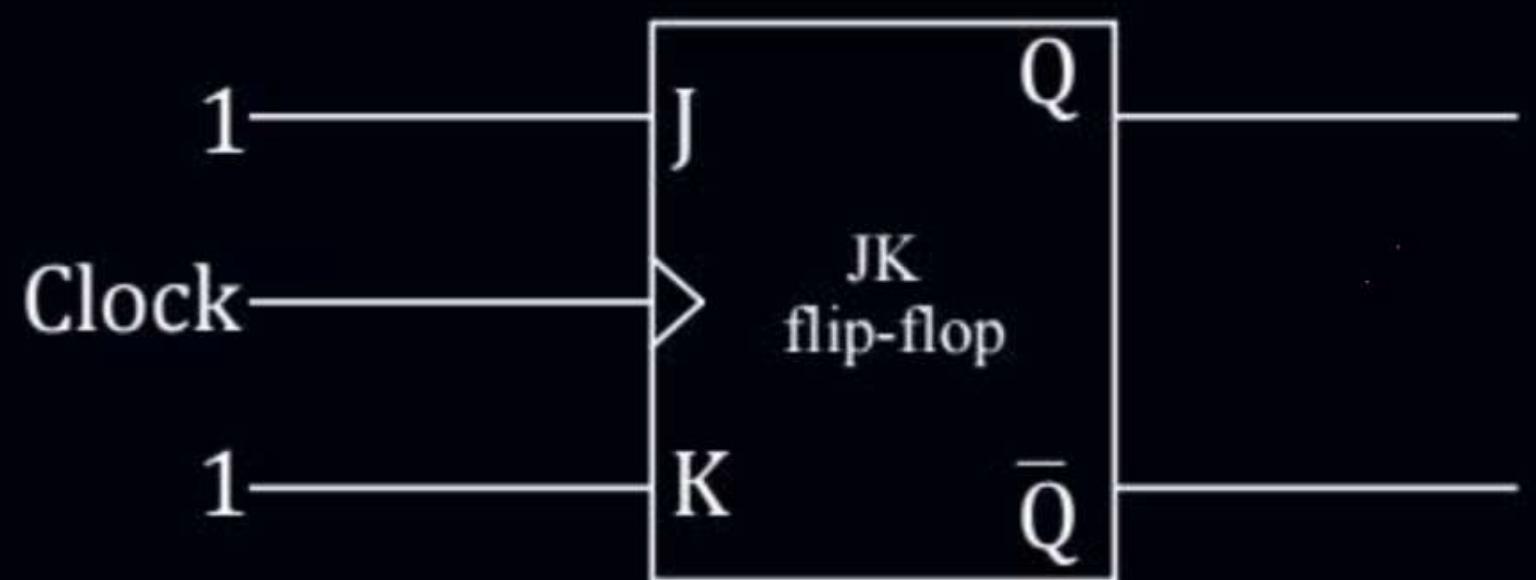


↙

## TOGGLE MODE OF FLIP FLOPS

2. Toggle mode of J K Flip Flop.

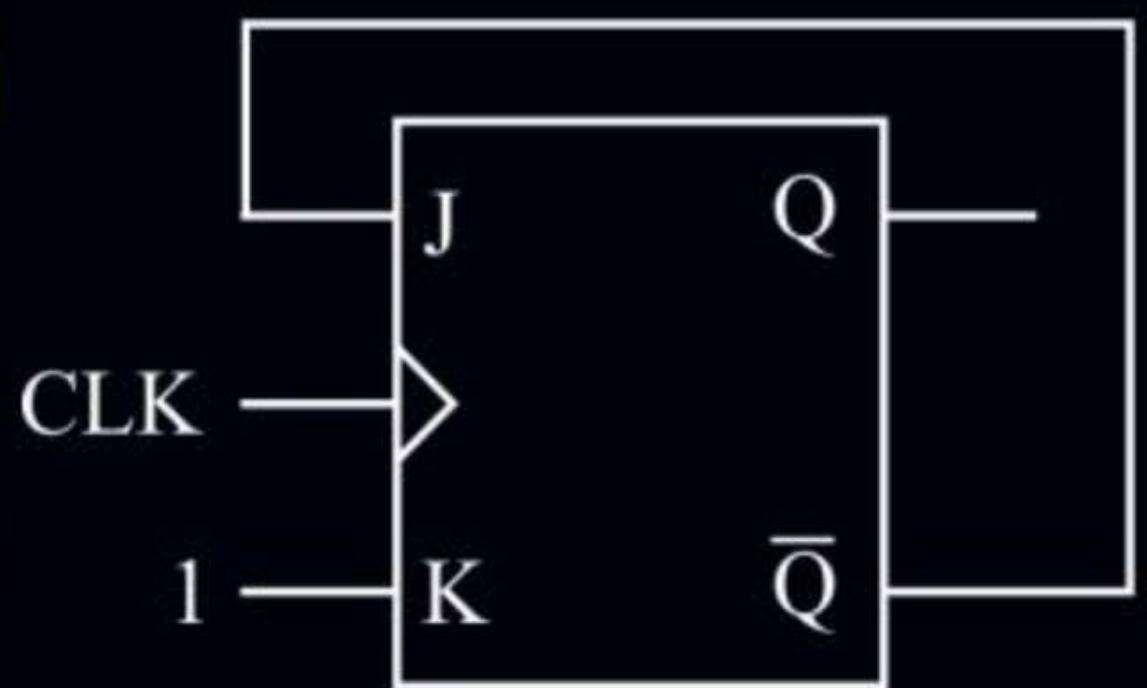
(i)



## TOGGLE MODE OF FLIP FLOPS

2. Toggle mode of J K Flip Flop.

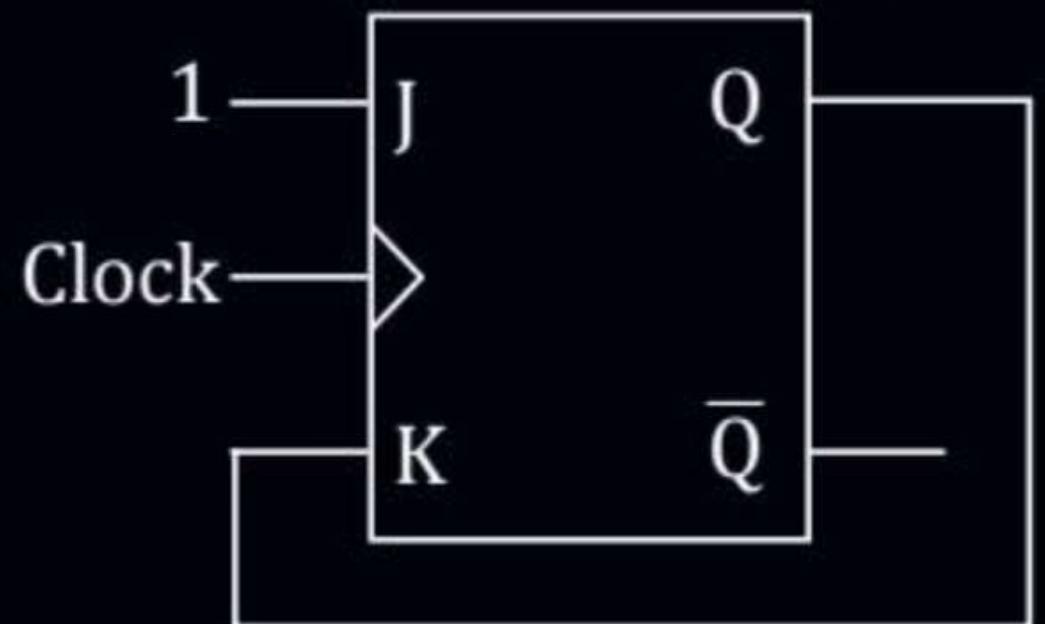
(ii)



## TOGGLE MODE OF FLIP FLOPS

2. Toggle mode of J K Flip Flop.

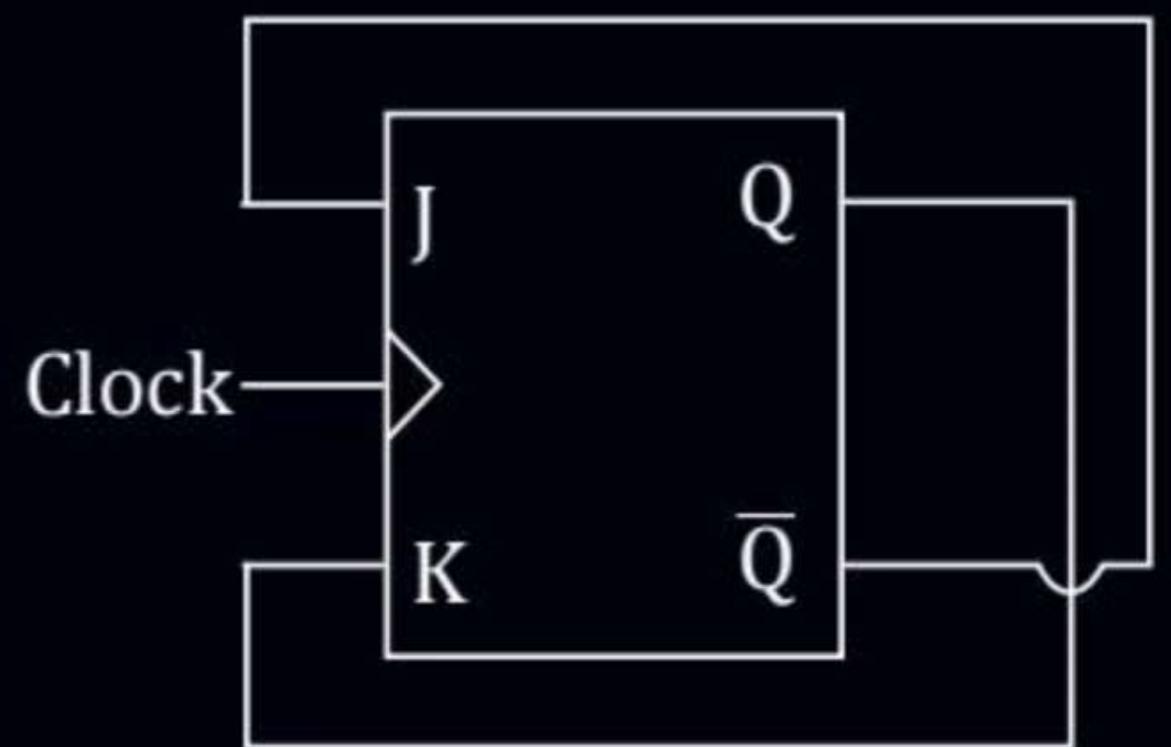
(iii)



## TOGGLE MODE OF FLIP FLOPS

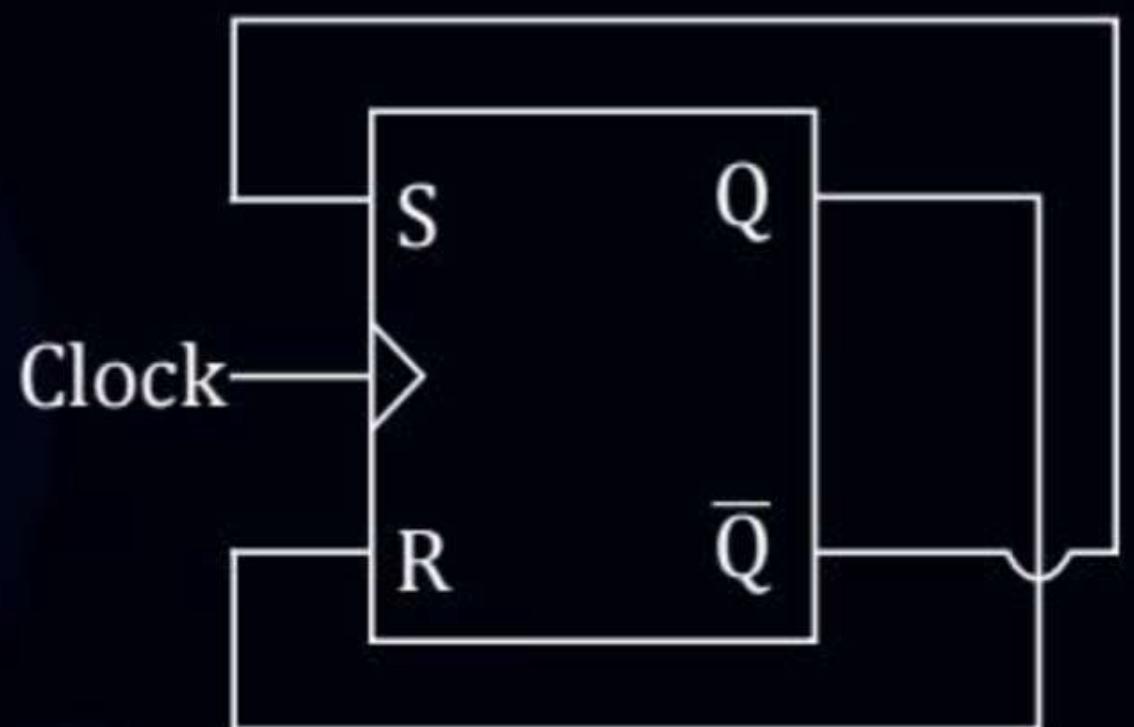
2. Toggle mode of J K Flip Flop.

(iv)



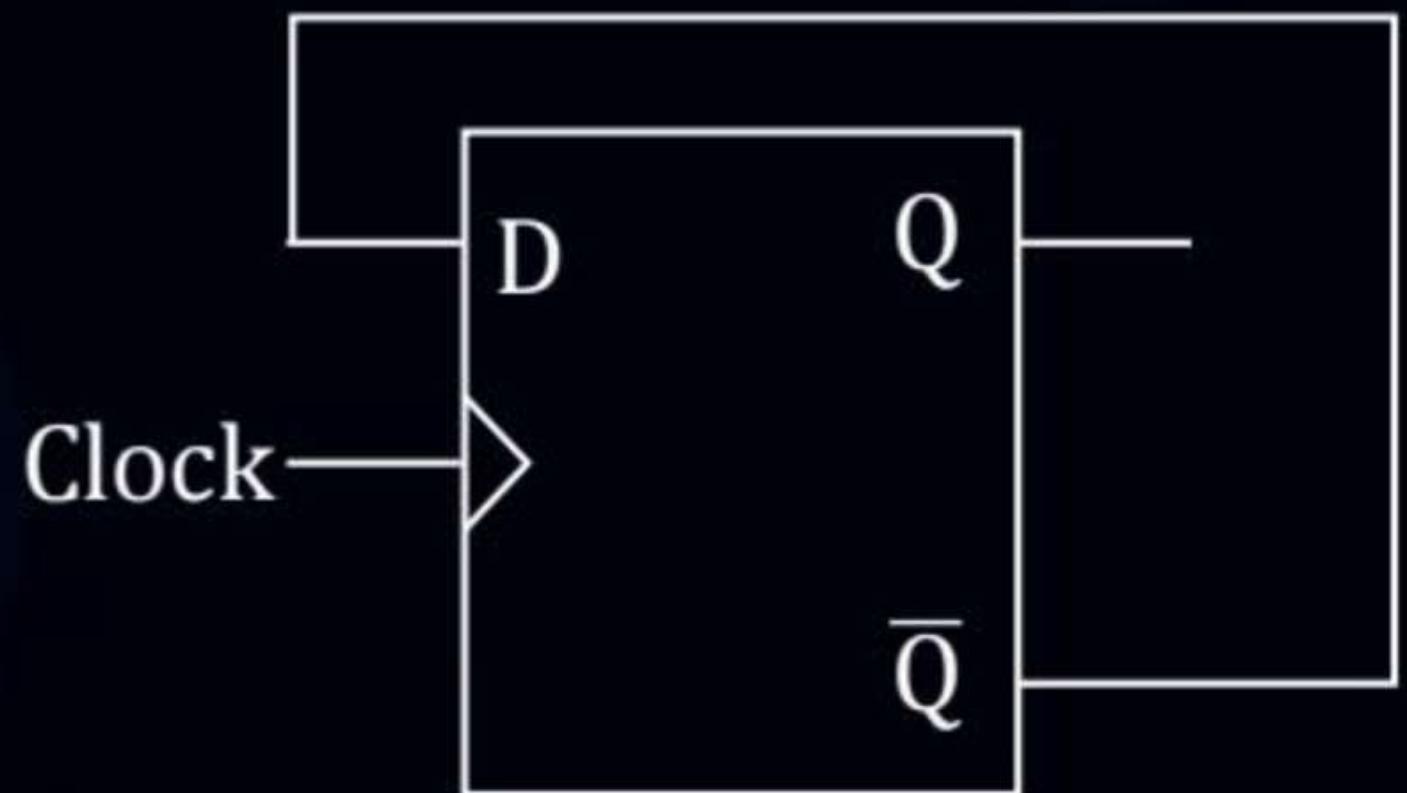
## TOGGLE MODE OF FLIP FLOPS

3. Toggle mode of S R Flip Flop.



## TOGGLE MODE OF FLIP FLOPS

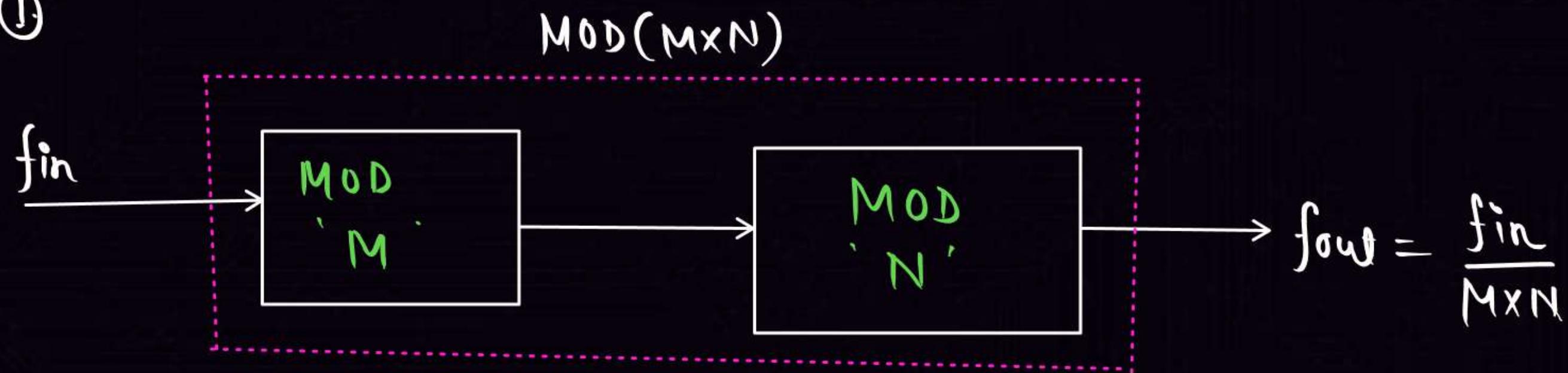
4. Toggle mode of D Flip Flop.



$$\text{Q}_{n+1} = \bar{\text{Q}}_n$$

COUNTERS.

①

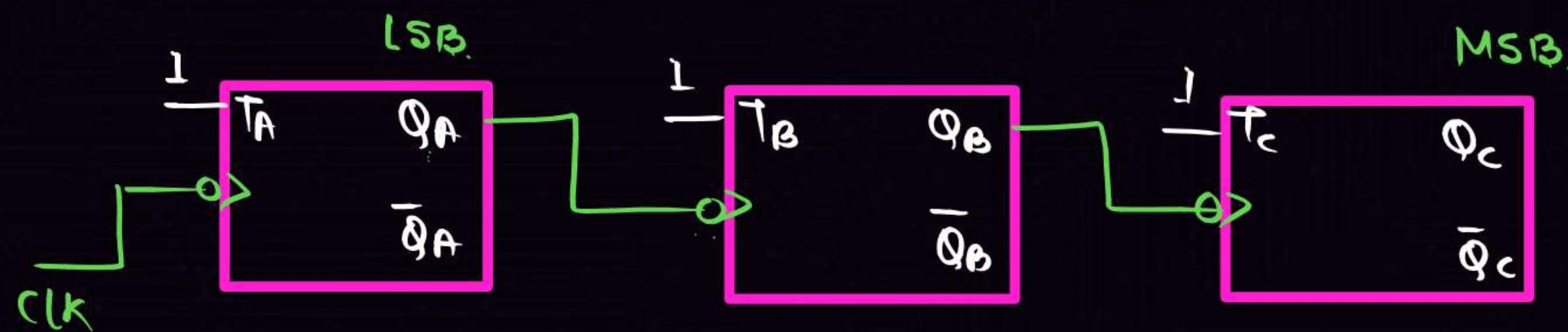


②  $(M) \text{ MOD} \leq 2^n$        $\lceil \text{no. of FF} \rceil$

$$\boxed{n \geq \log_2 M}$$

## Asynchronous counter :-

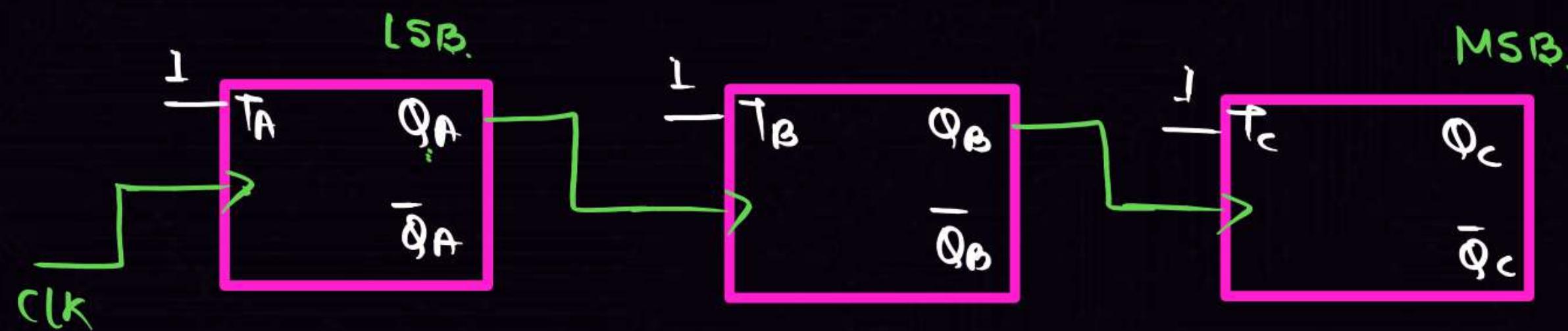
-ve  $\rightarrow$  UP  
+ve  $\rightarrow$  DOWN



MOD = 8  
UP

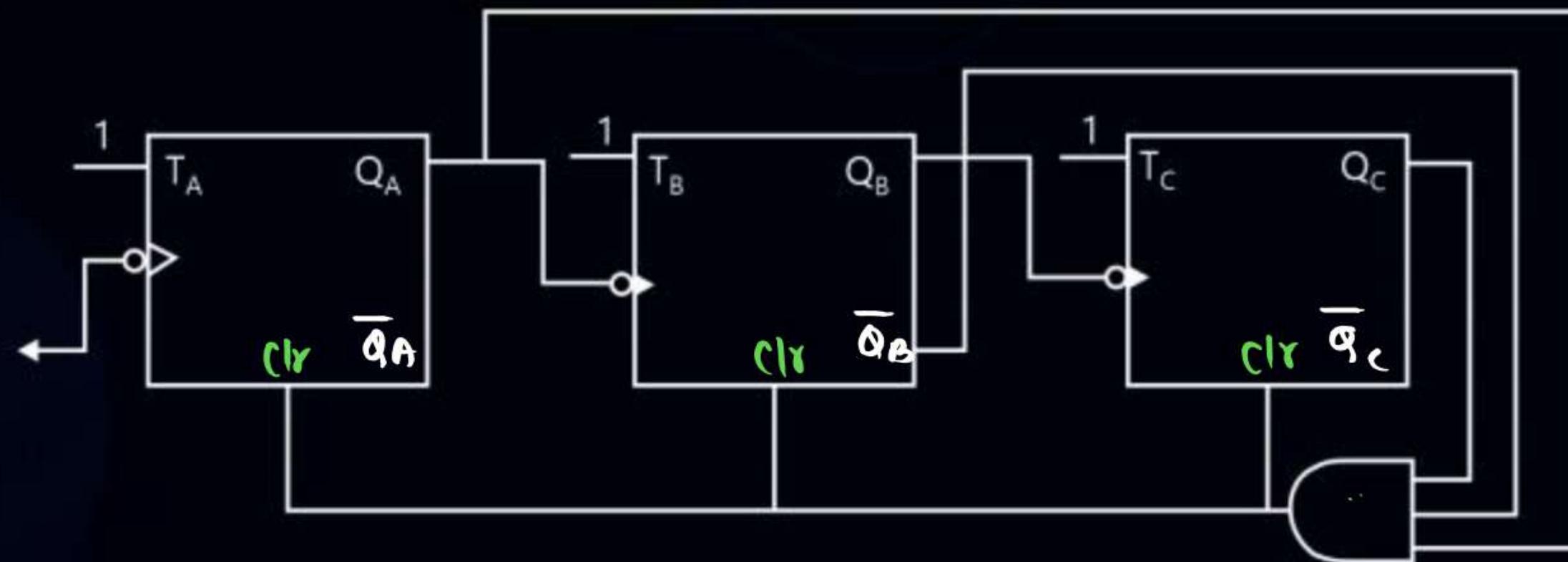
## Asynchronous Counter :→

-ve → UP  
+ve → DOWN



MOD-8 Down

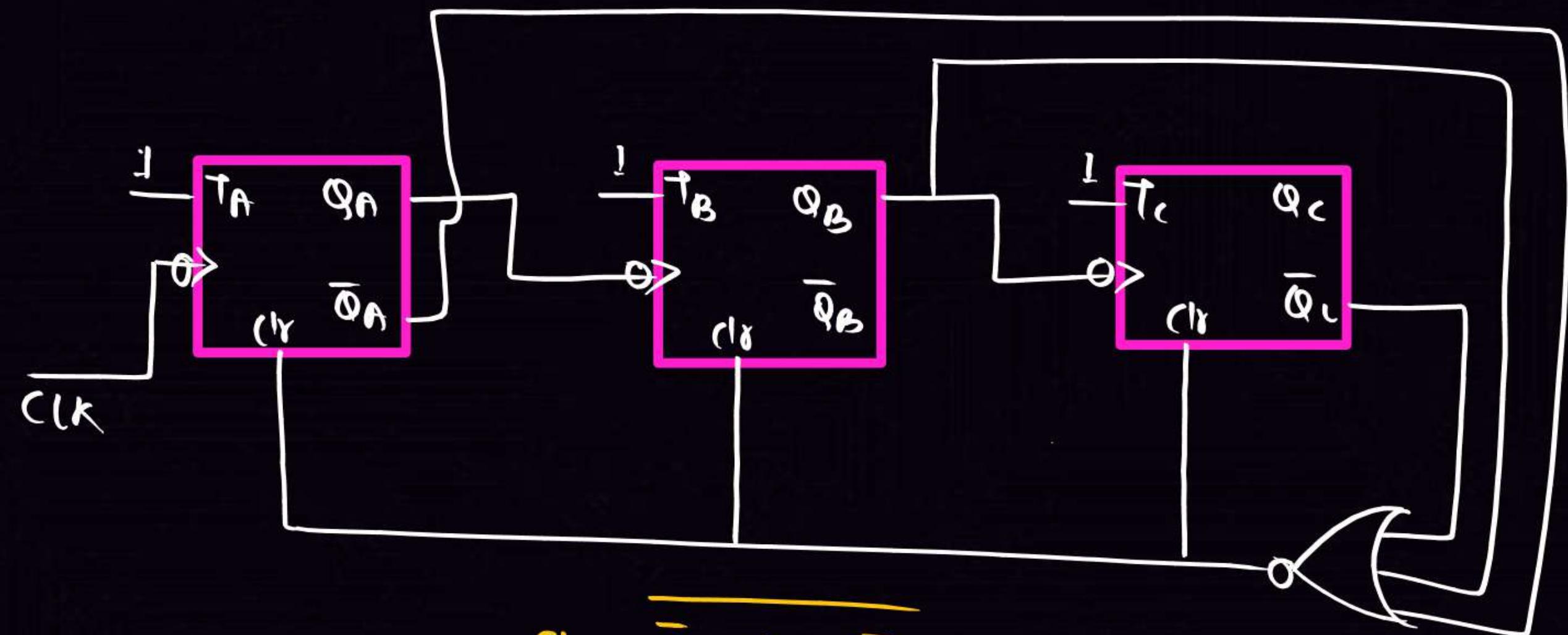
# Feedback Reduces the Number of States



$\text{Clr} = \bar{Q}_C \bar{Q}_B Q_A$

$1 \ 0 \ 1 = 5$

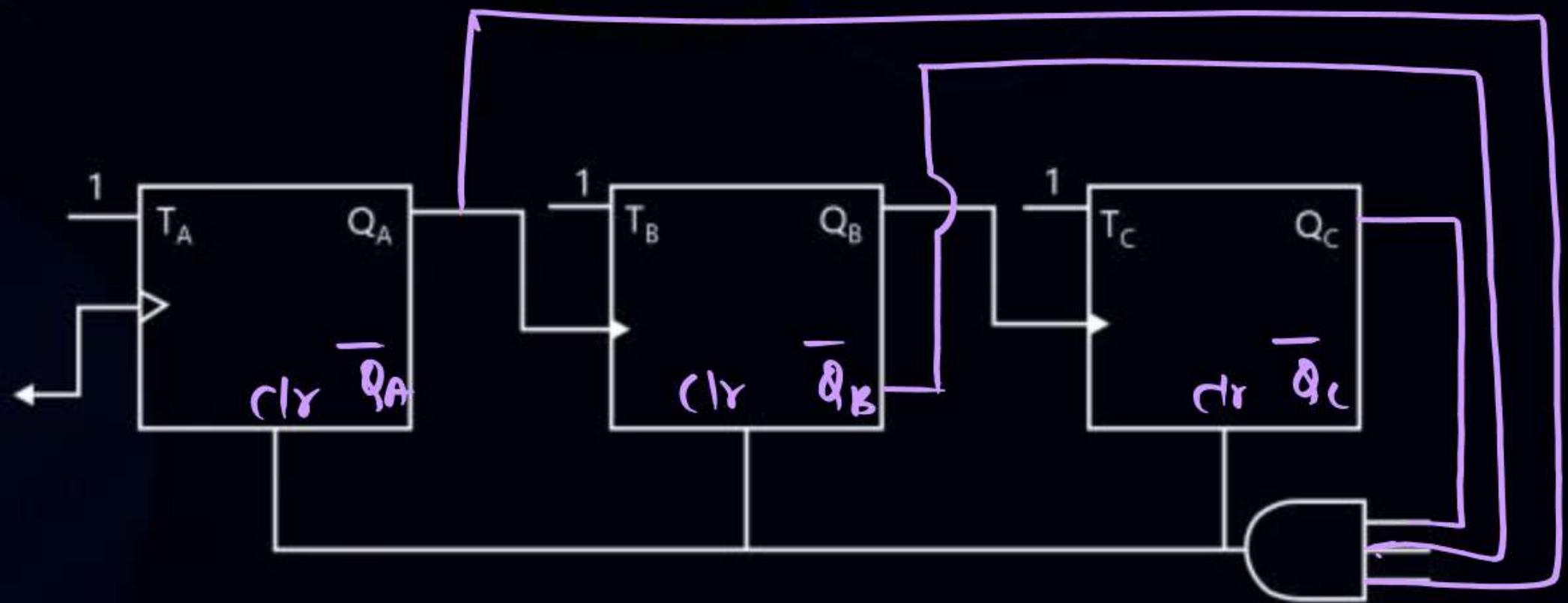
Mod-5 UP Ripple Counter



$$\text{Clk} = \overline{\bar{Q}_C + Q_B + \bar{Q}_A}$$

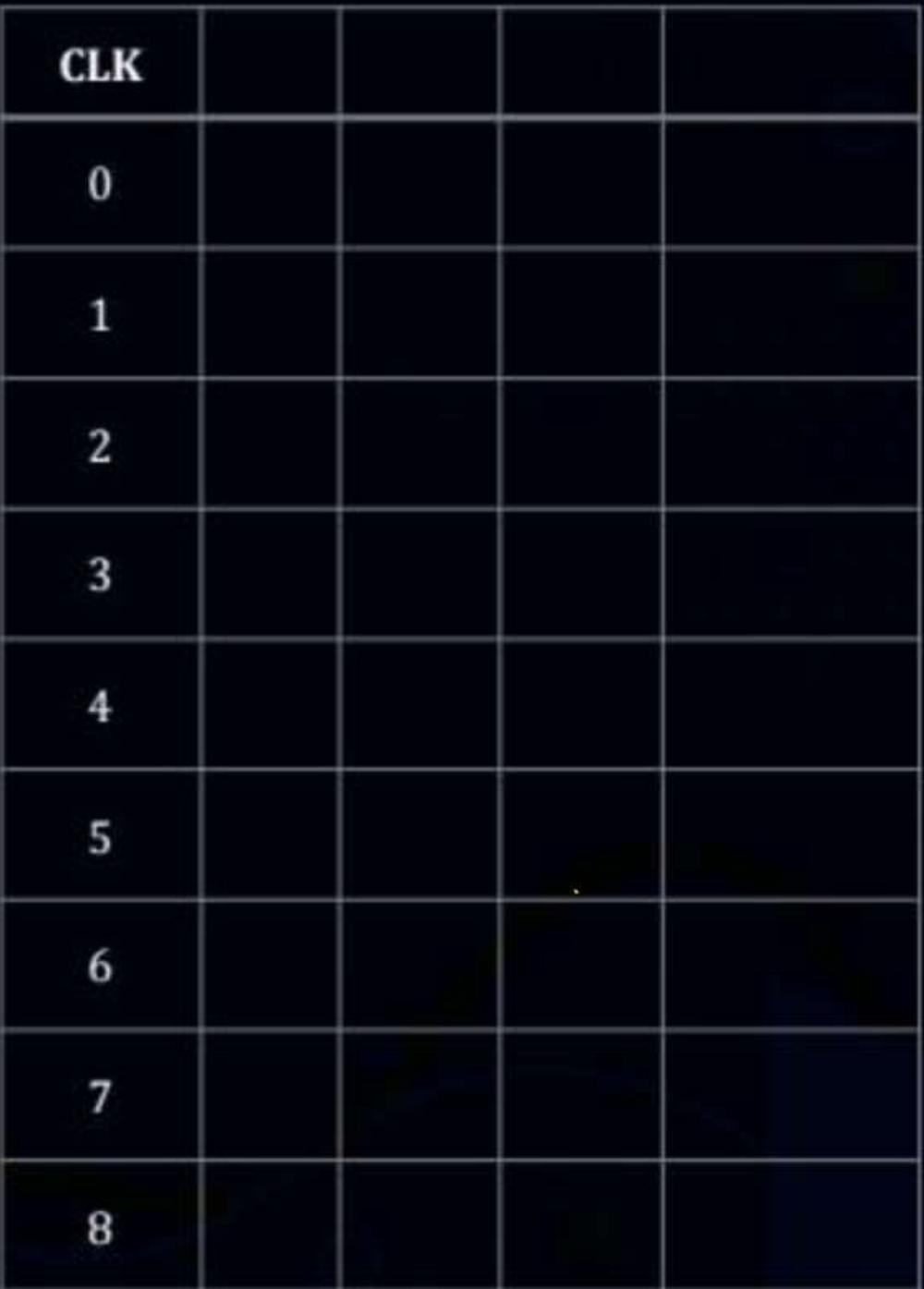
$\text{Clk} = Q_C \cdot \bar{Q}_B \cdot Q_A = \text{(101)} \rightarrow \text{MOD-5 UP counter}$

# Asynchronous Counter

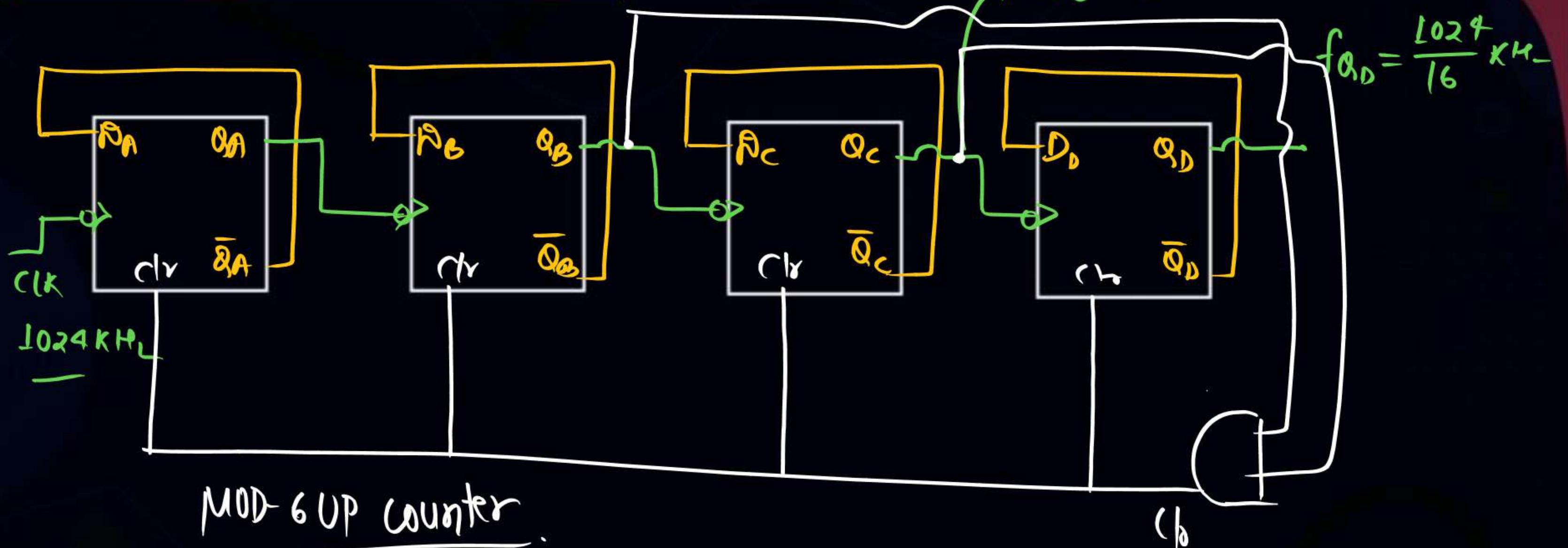


Full- UP = DOWN  
 $8 - 5 = 3$

MOD 3 Down  
Ripple counter



# ASYNCHRONOUS COUNTER



$$\text{Clr} = \overline{Q_D} \overline{Q_C} \overline{Q_B} \overline{Q_A}$$

0 1 1 0 → 6

## Asynchronous Counter

$$f_{clk} \leq \frac{1}{n \cdot \tau_{pd_{ff}}}$$

$$(f_{clk})_{\max} = \frac{1}{n \cdot \tau_{pd_{ff}}}$$

$n \rightarrow$  no. of FF's.

$\tau_{pd_{ff}}$   $\rightarrow$  Propagation Delay of FF's

Q.

Consider the following counter

If counter starts at 000, what will be the count after 13 clock pulses?

A.

100

B.

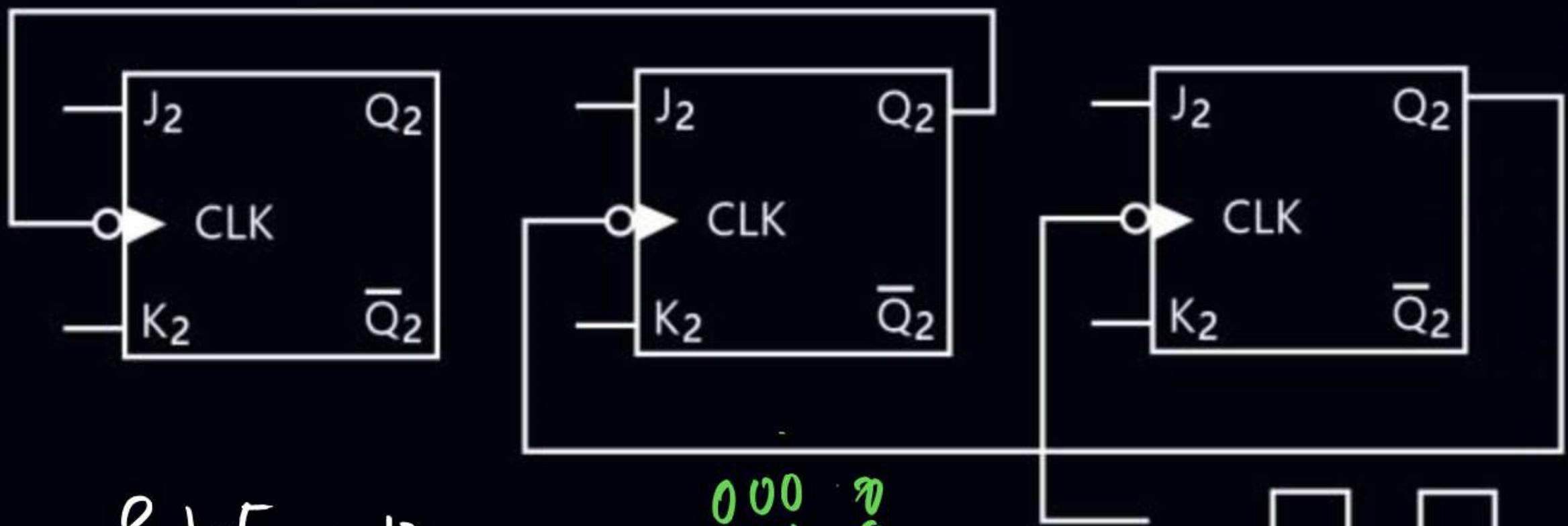
101

C.

110

D.

111



$$8 + 5 = \underline{13}$$

101

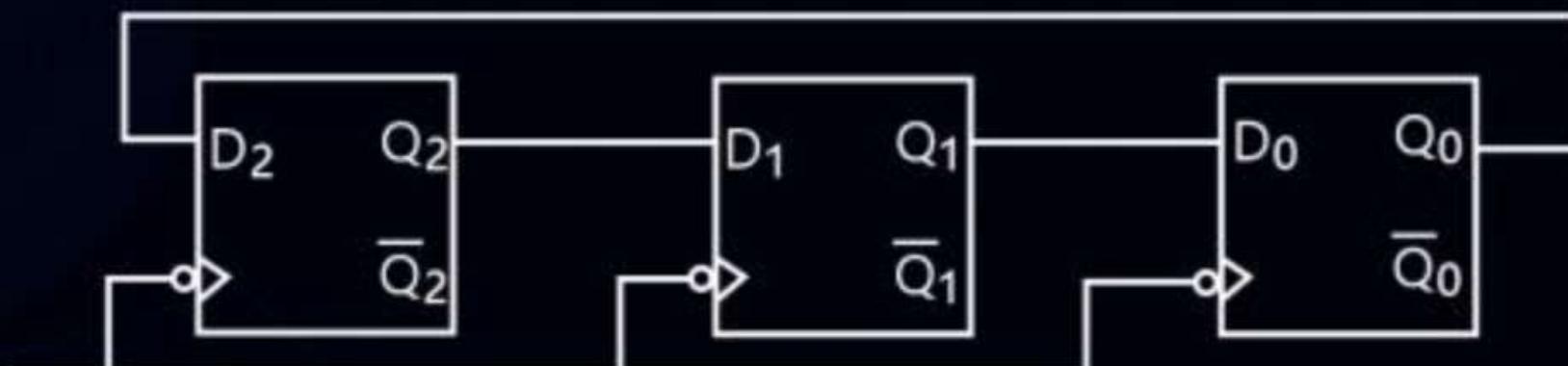
000  
001  
010  
011  
100  
101  
110  
111

Input clock pulse

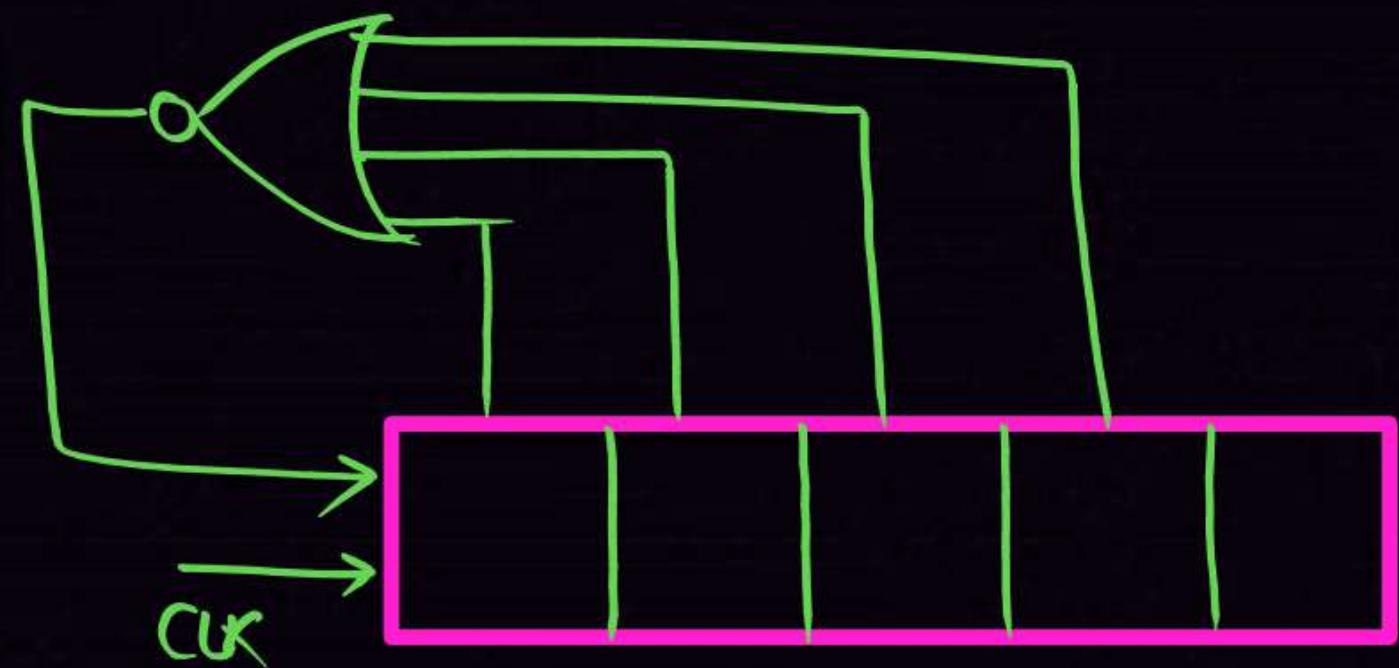
# Ring Counter

It is a SISO Shift Register in the Form of Ring.

- It is not a self Starting Counter. To start the Counter we have to place “1” at MSB after that “1” rotates among all the Flip-Flop.
- N Bit Ring Counter's Used state =  $N \text{ (MOD)}$



Clock	$Q_2$	$Q_1$	$Q_0$
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			



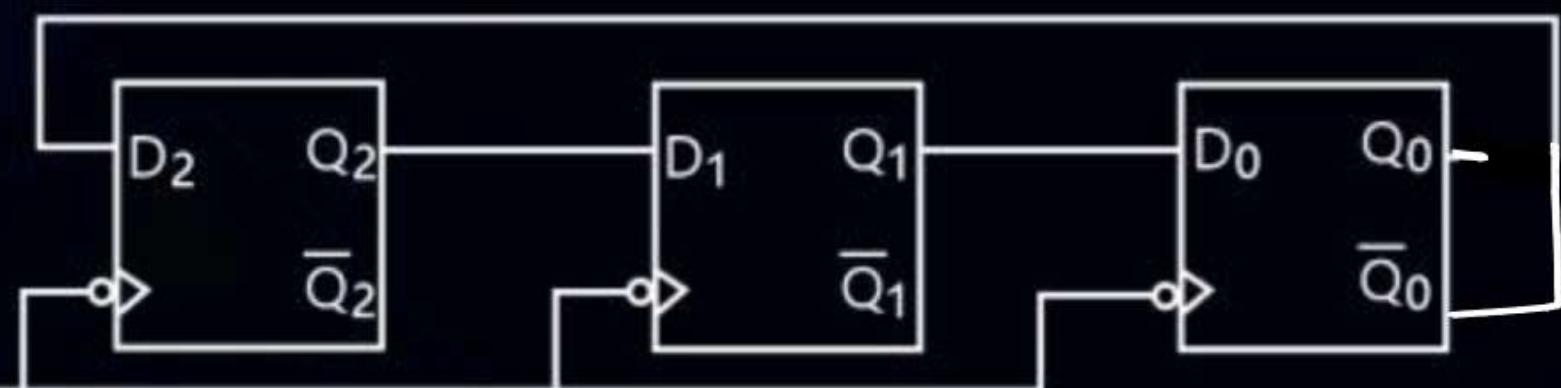
Self starting Ring Counter

$$\left\{ \begin{array}{l} 10000 \\ 01000 \\ 00100 \\ 00010 \\ 00001 \end{array} \right.$$

# Johnson Counter

- Twisted Ring Counter ✓
- Creeping Counter ✓
- Mobies Counter ✓
- Walking Counter ✓

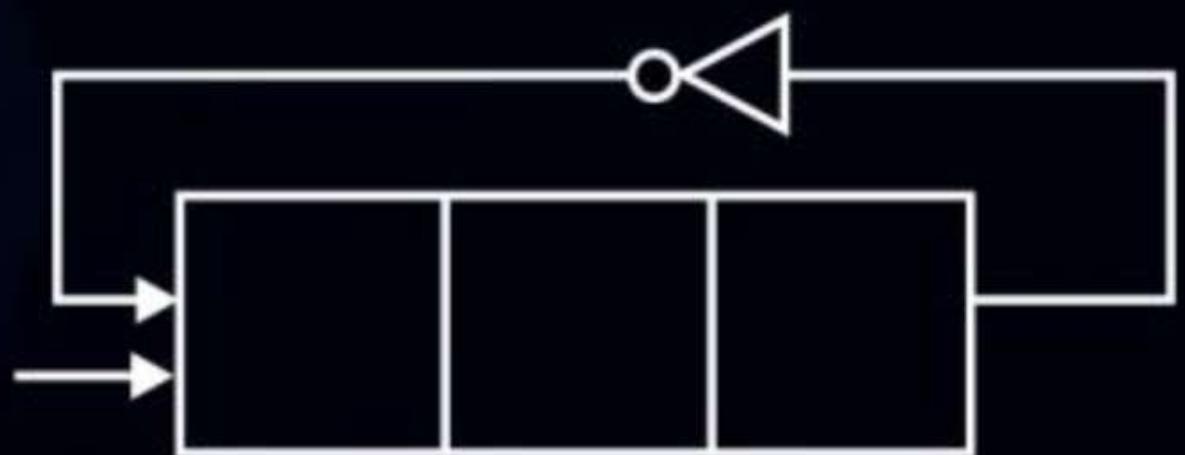
$\text{MOD} = 2^N$



Clock	$Q_2$	$Q_1$	$Q_0$
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0
7			
8			
9			

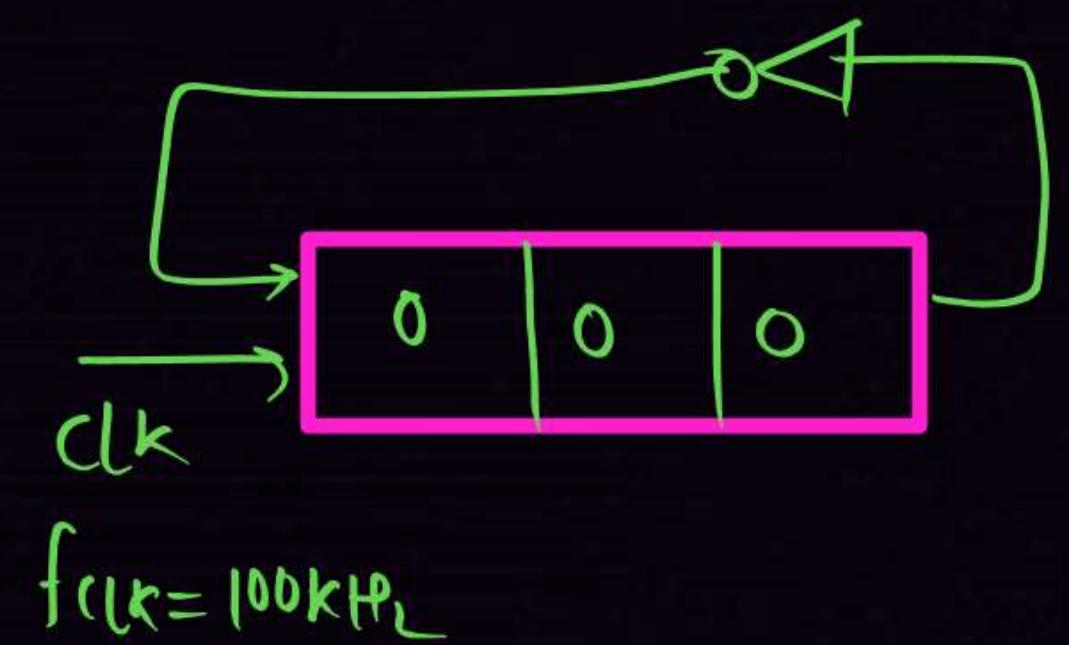
## Lock Out Problem

- Whenever Johnson Counter enters into its unused state then it will lock into its unused state is called **lock out problem**.

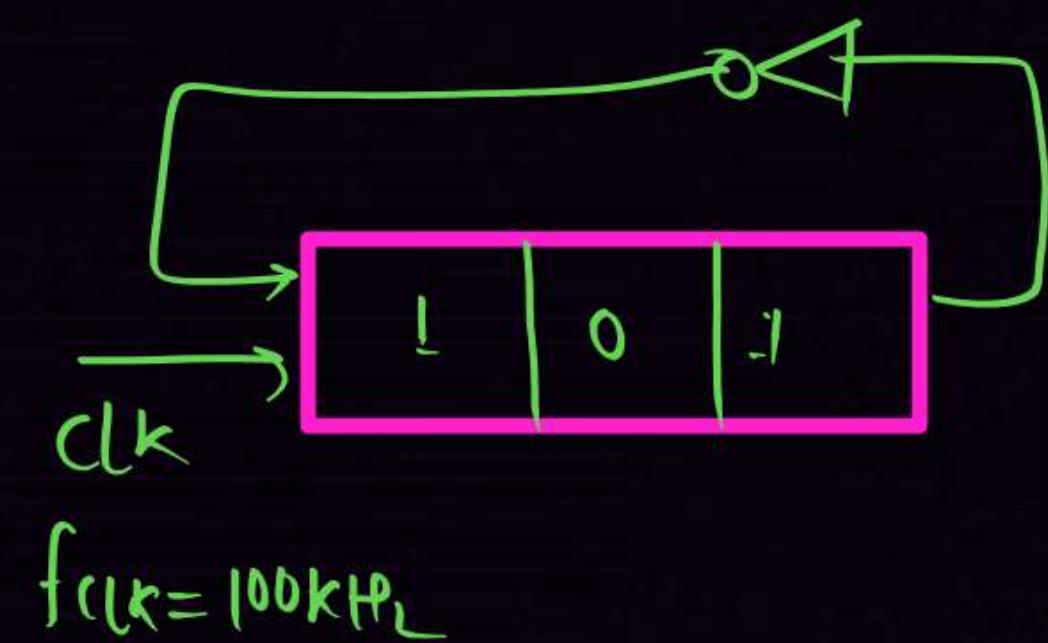


Clock	$Q_2$	$Q_1$	$Q_0$	
0	0	1	0	
1	1	0	1	
2	0	1	0	
3	1	0	1	
4				
5				
6				
7				
8				
9				

$\text{Mod} = 2$

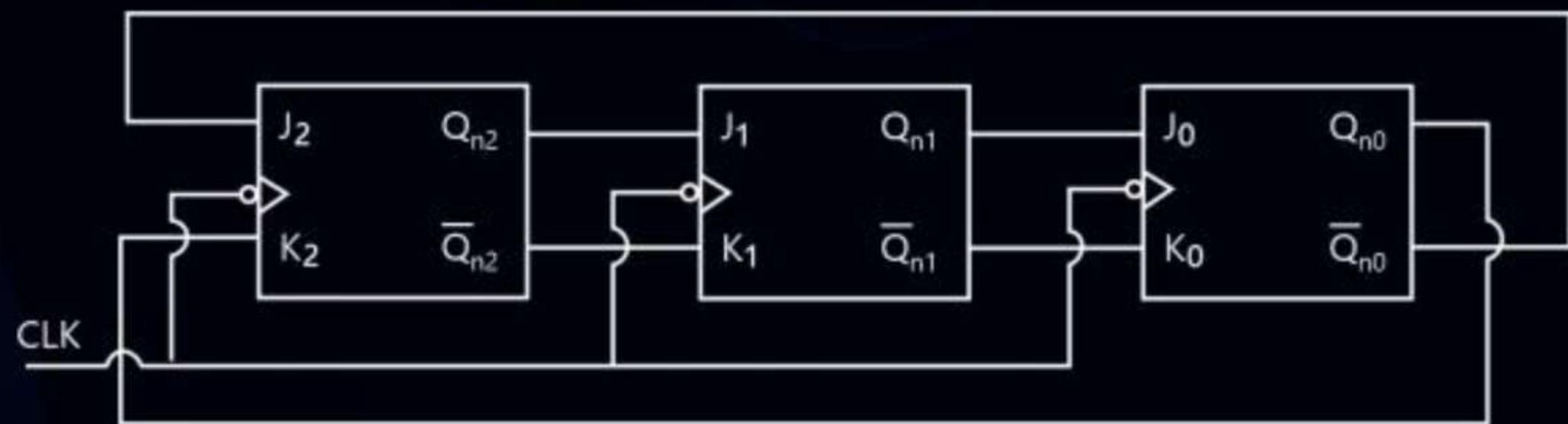


$$f_{out} = \frac{100}{6} \text{ KHz}$$



$$f_{out} = \frac{100}{2} = \underline{\underline{50 \text{ KHz}}}$$

# Johnson Counter By Using JK flip flop



## Synchronous counter.

$$f_{CLK} \leq \frac{1}{\tau_{Pdff}}$$

## Designing of Synchronous Counter

STEP 1. Write the Previous and Present State.

STEP 2. Write the Excitation Table of FF.

STEP 3. Write the Logical Expression.

STEP 4. Minimize the Logical expression.

STEP 5. Hardware Implementation.

		0 → 1 → 3 → 0 →					
		$Q_1 Q_0$	$Q_1 + Q_0$	$Q_0 + Q_1$	$\beta_1 \beta_0$		
$Q_1 Q_0$	$Q_1 + Q_0$	00	01	11	10	1	1
		10	X X	X X	X	X	X
$Q_1 Q_0$	$Q_1 + Q_0$	11	00	00	0	0	0
		11	11	11	11	11	11

Q Design a Synchronous counter by T-FF which count

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0 \rightarrow \dots$

Step 1 & step 2.

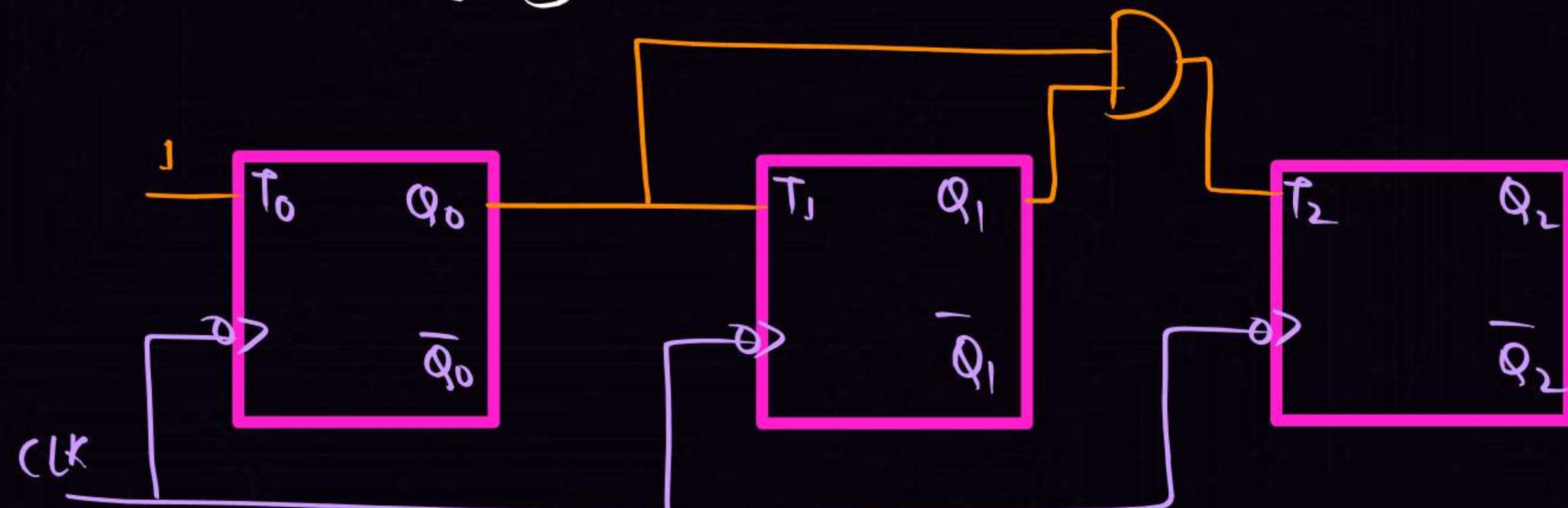
$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

$$T_2 = Q_1 Q_0$$

$$T_1 = Q_0$$

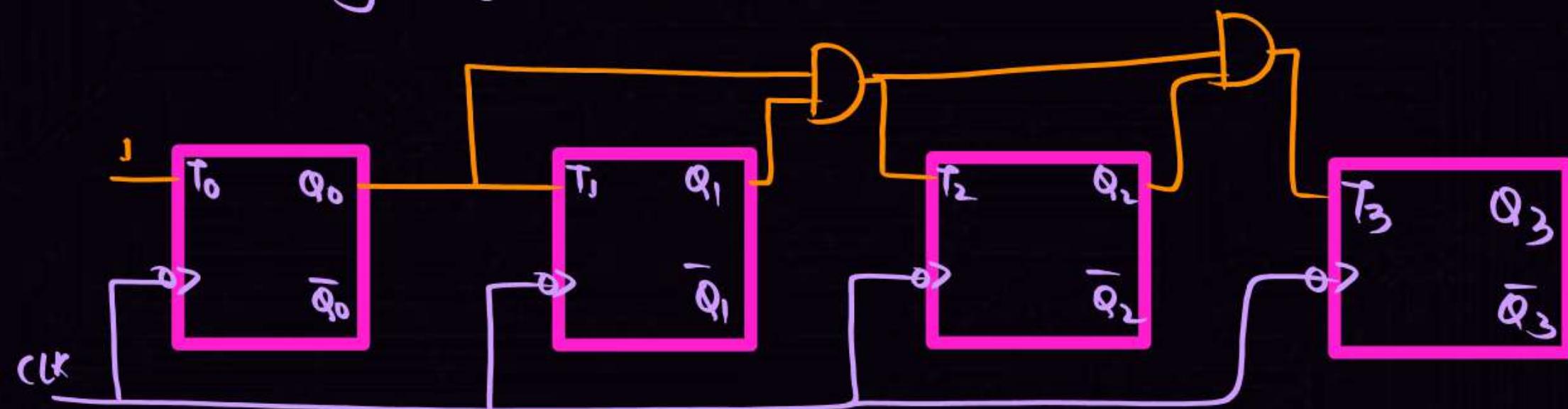
$$T_0 = 1$$

# Series carry synchronous counter



$$f_{CLK} \leq \frac{1}{\tau_{pd_{ff}} + \tau_{pd_{AND}}}$$

# Series carry Synchronous counter



$$f_{CLK} \leq \frac{1}{\tau_{Pd_{ff}} + 2 \cdot \tau_{Pd_{AND}}}$$

"n" bit Series carry synchronous counter.

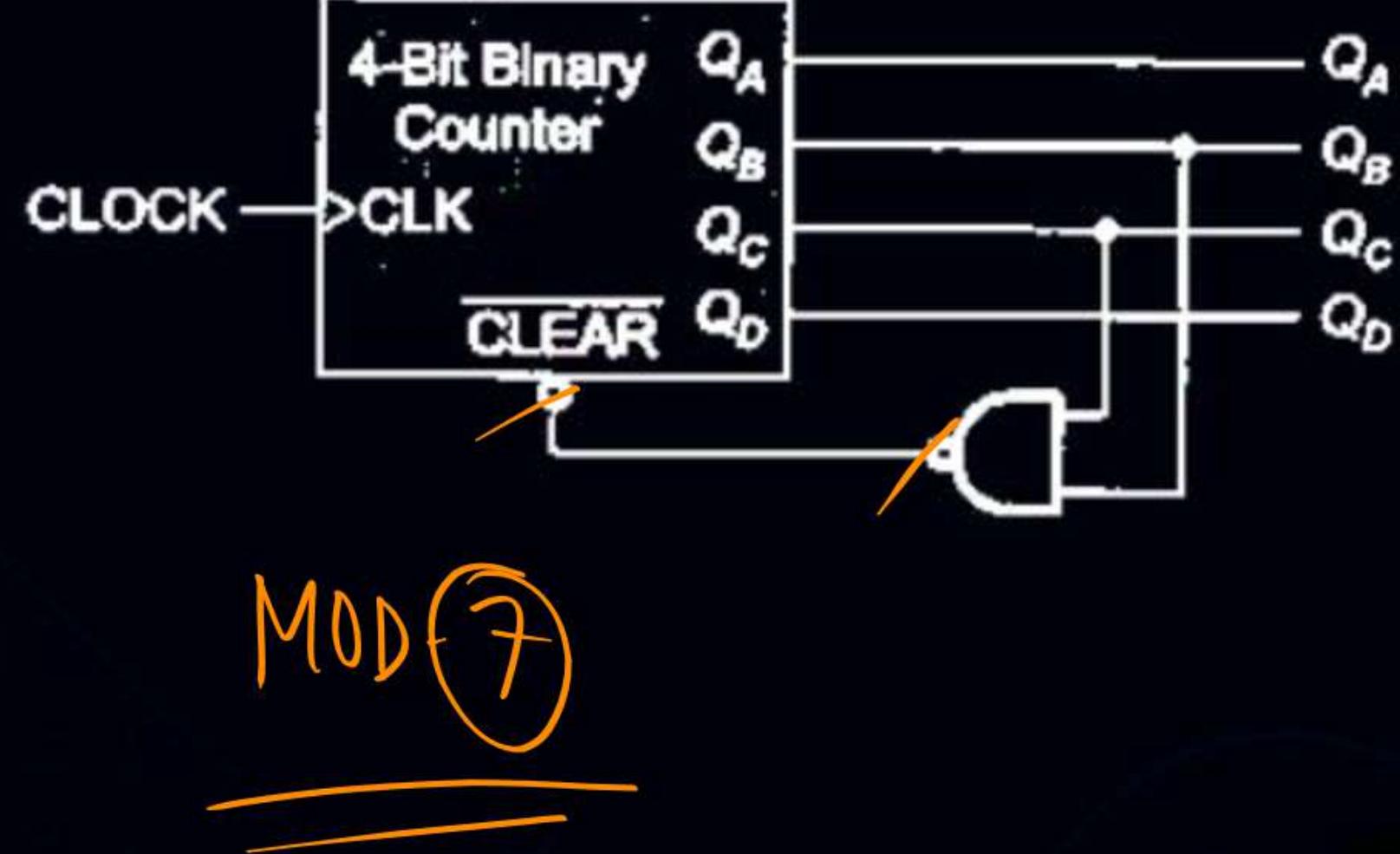
$$f_{CLK} \leq \frac{1}{\tau_{PdFF} + (n-2)\tau_{PdAND}}$$

"n" bits Parallel carry synchronous counter

$$f_{CLK} \leq \frac{1}{\tau_{PdFF} + \tau_{PdAND}}$$

NAT

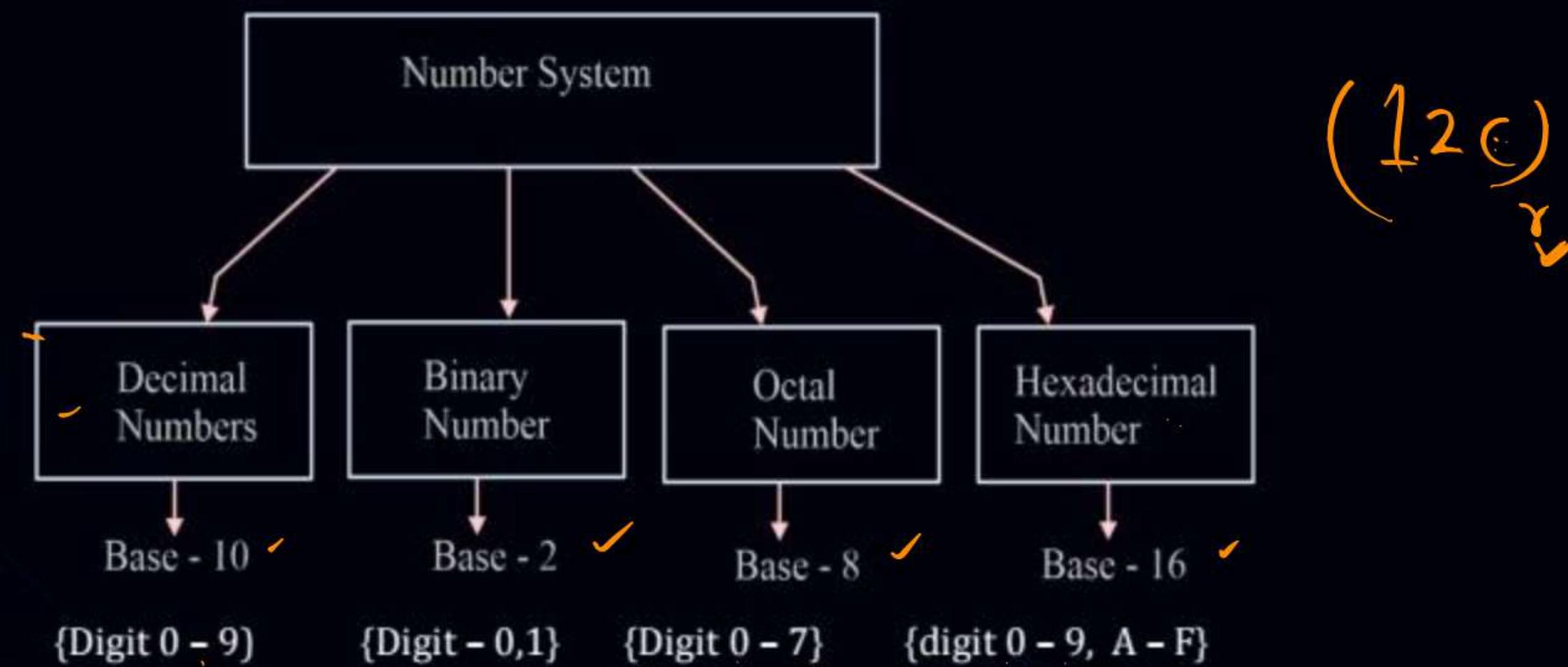
A mod-n counter using a **synchronous** binary up-counter with synchronous clear input is shown in the figure. The value of n is



$Q_D$	$Q_C$	$Q_B$	$Q_A$	$Clk \cdot \bar{Q}_B \bar{Q}_C$
✓ 0	0	0	0	0
✓ 0	0	0	1	0
✓ 0	0	1	0	0
✓ 0	0	1	1	0
✓ 0	1	0	0	0
✓ 0	1	0	1	0
✓ 0	1	1	0	1

## Base (Radix)

Total number of digit used in the system



## Decimal Number System

...	$10^4$	$10^3$	$10^2$	$10^1$	$10^0$	$10^{-1}$	$10^{-2}$	$10^{-3} \dots$
...	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$	$a_{-1}$	$a_{-2}$	$a_{-3} \dots$

$a^i \rightarrow$  Coefficient of decimal number system

$10_i \rightarrow$  Weight of decimal number system

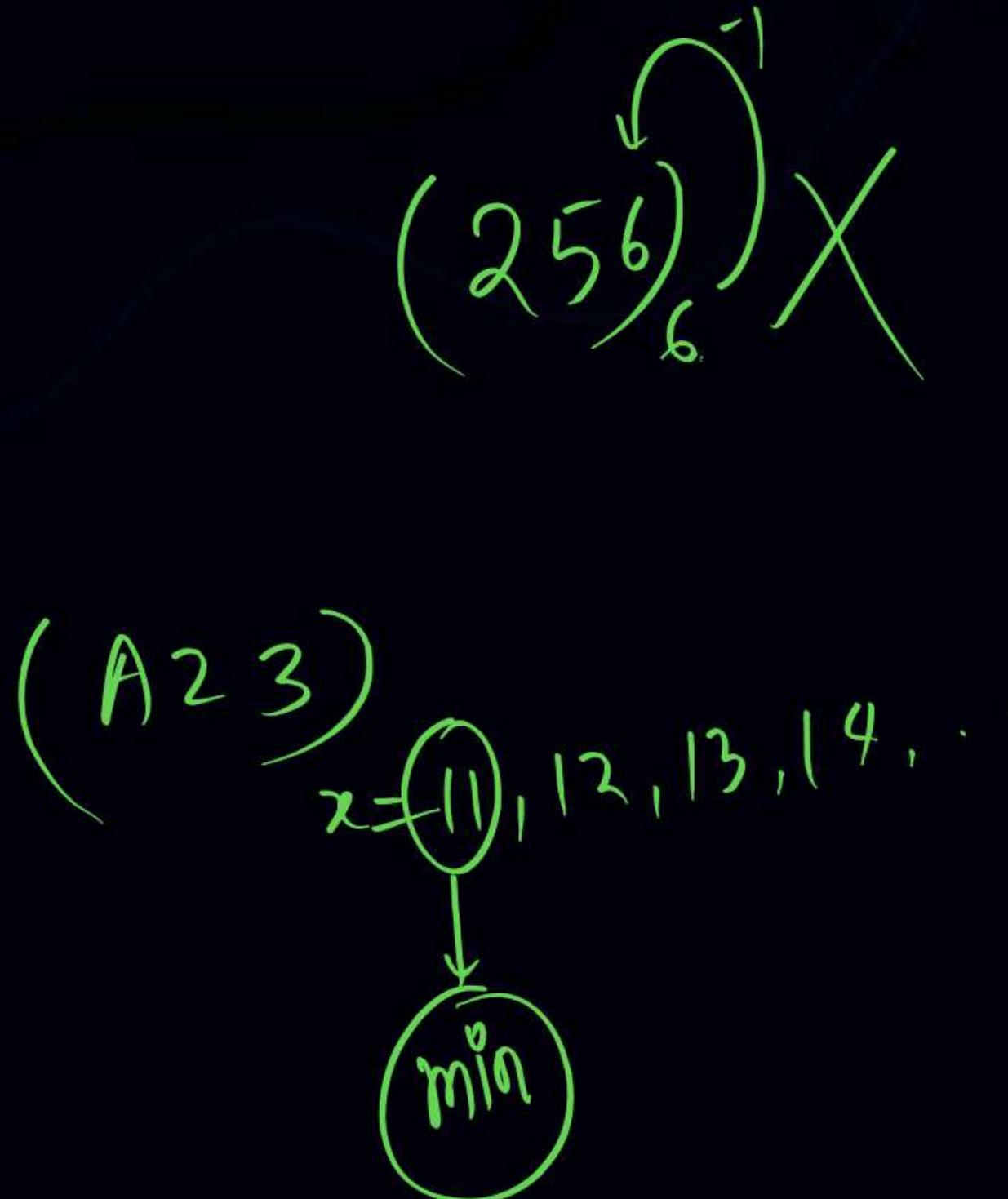
Example :

$(501.23)_{10}$

$10^2$	$10^1$	$10^0$	$10^{-1}$	$10^{-2}$
5	0	1	2	3

$$\left( \begin{matrix} 10^2 & 10^1 & 10^0 & 10^{-1} & 10^{-2} \\ 5 & 0 & 1 & 2 & 3 \end{matrix} \right)_{10}$$

Base	Digit
2	0, 1
3	0, 1, 2
4	0, 1, 2, 3
5	0, 1, 2, 3, 4
6	0, 1, 2, 3, 4, 5
7	0, 1, 2, 3, 4, 5, 6
8	0, 1, 2, 3, 4, 5, 6, 7
9	0, 1, 2, 3, 4, 5, 6, 7, 8
10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
11	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A
12	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
13	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C
14	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D
15	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E
16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F



## Binary Number System (Base (Radix) = 2)

...	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$ ...
...	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$	$a_{-1}$	$a_{-2}$	$a_{-3}$ ...

$2^i \rightarrow$  Weight of Binary number system

$a_i \rightarrow$  Coefficient of Binary number system {0, 1}

$$2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$$

Example:-

$$(101.11)_2$$

$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$
1	0	1	1	1

## Octal Number System (Base (Radix) = 8)

...	$8^3$	$8^2$	$8^1$	$8^0$	$8^{-1}$	$8^{-2}$	$8^{-3} \dots$
...	$a_3$	$a_2$	$a_1$	$a_0$	$a_{-1}$	$a_{-2}$	$a_{-3} \dots$

$8^i \rightarrow$  Weight of Octal number system

$a_i \rightarrow$  Coefficient of Octal number system {0 - 7}

$8^2 \ 8^1 \ 8^0 \ 8^{-1} \ 8^{-2}$

Example:-

$(723.64)_8$

$8^2$	$8^1$	$8^0$	$8^{-1}$	$8^{-2}$
7	2	8	6	4

## Hexadecimal Number System (Base (Radix) = 16)

$$\dots \quad 16^3 \quad 16^2 \quad 16^1 \quad 16^0 \quad 16^{-1} \quad 16^{-2} \quad 16^{-3} \dots$$
$$\dots \quad a_3 \quad a_2 \quad a_1 \quad a_0 \quad a_{-1} \quad a_{-2} \quad a_{-3} \dots$$

$16^i \rightarrow$  Weight of Hexadecimal number system

$a_i \rightarrow$  Coefficient of Hexadecimal number system {0 - 9, A-F}

|<sup>2</sup> |<sup>1</sup> |<sup>6</sup> |<sup>6</sup> |<sup>6</sup> |<sup>-1</sup>

Example: (A2C.F)<sub>16</sub>

$$\begin{array}{cccc} 16^2 & 16^1 & 16^0 & 16^{-1} \\ A & 2 & C & F \end{array}$$

## Number system

- ① Base conversion
- ② Magnitude Representation

## In Base Conversion 2 Key Points are There:

- (A) Any base to Decimal conversion
- (B) Decimal to any other base conversion

**(A) Any base to Decimal conversion :**

$$\gamma^3 \gamma^2 \gamma^1 \gamma^0 \gamma^{-1} \gamma^{-2}$$
$$(a_3 \ a_2 \ a_1 \ a_0 \cdot a_{-1} \ a_{-2})_{\gamma} = ( \quad )_{10}$$

$$(a_3 \times r^3 + a_2 \times r^2 + a_1 \times r^1 + a_0 \times r^0 + a_{-1} \times r^{-1} + a_{-2} \times r^{-2})_{10}$$

**Case (1) : Binary to Decimal conversion**

Ex. 
$$(1011.11)_2 = (\quad)_{10}$$

$$\Rightarrow [(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2})]$$

$$\Rightarrow [8 + 0 + 2 + 1 + 0.5 + 0.25]_{10}$$

$$\Rightarrow (11.75)_{10}$$

### Case (2) : Octal to Decimal conversion

$8^2 \ 8^1 \ 8^0 \ 8^{-1}$

Ex.  $(721.4)_8 = (?)_{10}$

$$\Rightarrow [(7 \times 8^2) + (2 \times 8^1) + (1 \times 8^0) + (4 \times 8^{-1})]_{10}$$

$$\Rightarrow [448 + 16 + 1 + 0.5]_{10}$$

$$\Rightarrow (465.5)_{10}$$

### Case (3) : Hexadecimal to Decimal conversion

$$\text{Ex. } (\text{A}2\text{B.C})_{16} = (?)_{16}$$

$$\Rightarrow [(A \times 16^2) + (2 \times 16^1) + (B \times 16^0) + (C \times 16^{-1})]_{10}$$

$$\Rightarrow [(10 \times 256) + (2 \times 16) + (11 \times 1) + (12 \times 16^{-1})]_{10}$$

$$\Rightarrow [2560 + 32 + 11 + 0.75]_{10}$$

$$\Rightarrow (2603.75)_{10}$$

### Case (4) : Base 5 to Decimal conversion

Ex.  $(432.22)_5 = (?)_{10}$

$$\Rightarrow [(4 \times 5^2) + (3 \times 5^1) + (2 \times 5^0) + (2 \times 5^{-1}) + (2 \times 5^{-2})]_{10}$$

$$\Rightarrow [100 + 15 + 2 + 0.4 + 0.08]_{10}$$

$$\Rightarrow (117.48)_{10}$$

$$\frac{(312)_b}{(20)_b} = (13 \cdot 1)_b$$

$$\frac{3b^2 + 1b + 2b^0}{2b} = b + 3 + \frac{1}{b}$$

$$\frac{3b^2 + b + 2}{2b} = \frac{b^2 + 3b + 1}{b}$$

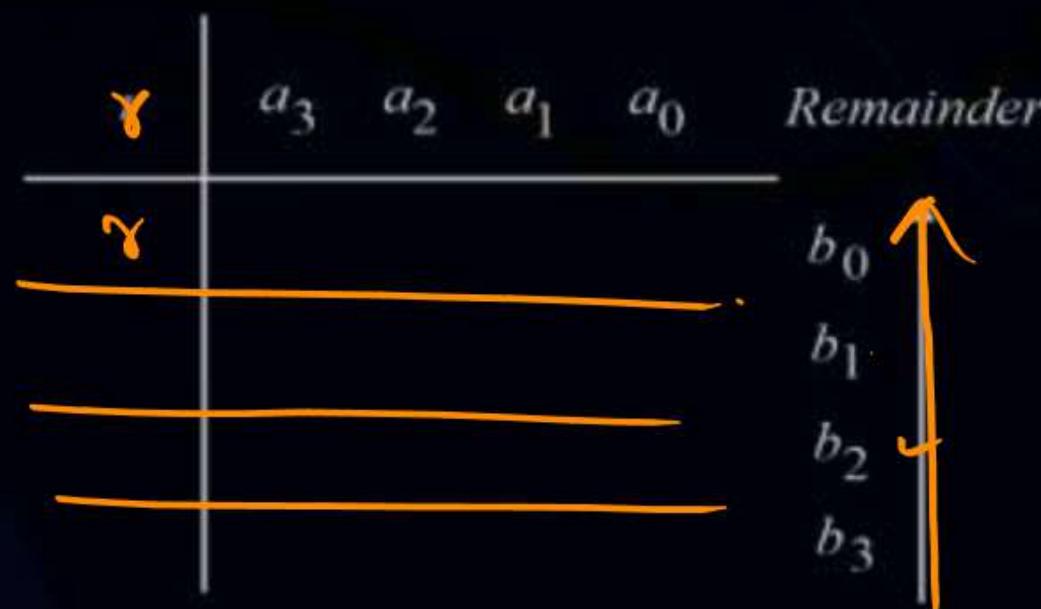
$$3b^2 + b + 2 = 2b^2 + 6b + 2$$

$$b^2 - 5b = 0$$

$$b(b-5) = 0$$

$$b=0 \quad b=5$$

## (B) Decimal to any other Base conversion



$$(a_3 \ a_2 \ a_1 \ a_0 \cdot a_{-1} \ a_{-2} \ a_{-3})_{10} = ( )_r$$

Before Decimal      After Decimal

$$0 \cdot a_{-1} a_{-2} a_{-3} \times r = x_0 \cdot x_{-1} x_{-2}$$

$$0 \cdot x_{-1} x_{-2} \times r = x_1 \cdot x_{-3} x_{-4}$$

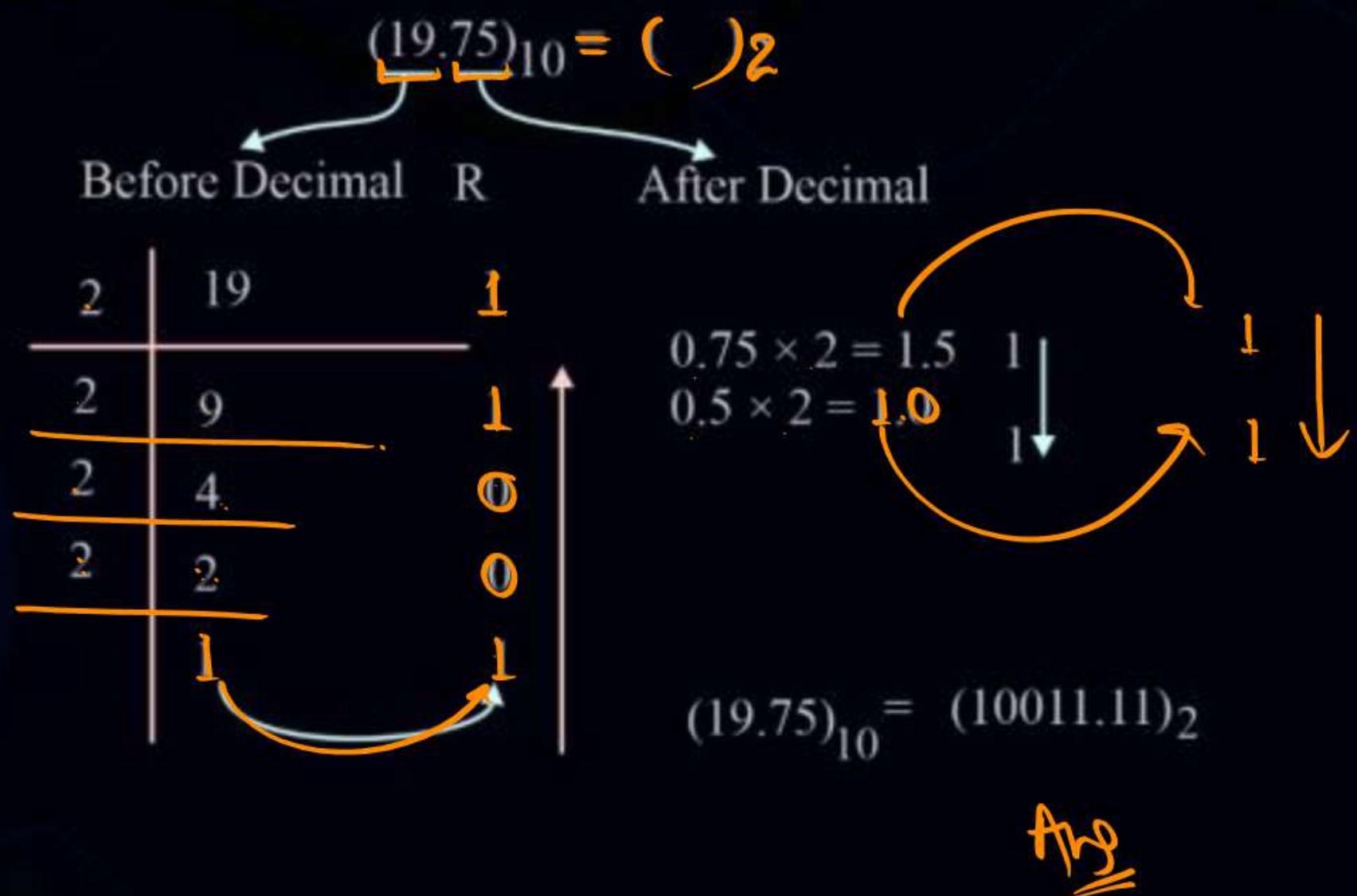
$$0 \cdot x_{-3} x_{-4} \times r = x_2 \cdot x_{-5} x_{-6}$$

$x_0$   
 $x_1$   
 $x_2$

$$(a_3 a_2 a_1 a_0 \cdot a_{-1} a_{-2} a_{-3})_{10} = (b_3 b_2 b_1 b_0 \cdot x_0 x_1 x_2)_r$$

**Case (1) : Decimal to Binary Base conversion.**

Ex.



**Case (2) : Decimal to Octal Base conversion.**

Ex.

$$(210.23)_{10} = (?)_8$$

Before Decimal      Remainder      After Decimal

$$\begin{array}{r}
 & 210.23 \\
 \times & 8 \\
 \hline
 16 & 26 \\
 + & 3 \\
 \hline
 1680 & 23 \\
 + & 3 \\
 \hline
 1744 & 16
 \end{array}$$

0.23 × 8 = 1.84      1  
 0.84 × 8 = 6.72      6  
 0.72 × 8 = 5.76      5

$(210.23)_{10} = (322.165)_8$

18

### Case (3): Decimal to Hexadecimal Base conversion.

Ex.

$$(1228.55)_{10} = (?)_{16}$$

Before Decimal Remainder      After Decimal

	$1228$	$12(C)$	↑
$16$	$\overline{16}$	$12(C)$	
$16$	$76$	$4$	

$0.55 \times 16 = 8.8$        $8$   
 $0.8 \times 16 = 12.8$        $12(C)$

$(1228.56)_{10} = (4CC.8C)_{16}$

$$( )_x = ( )_y$$
$$( )_{10}$$

$$\underline{x \neq y \neq 2^n}.$$

$$( )_x = ( )_y$$


 $x \neq y$ 

$$x, y = 2^n$$

## Some Special Case }

### Case (1): Binary to Octal base conversion

Ex.  $(10110111)_2 = (?)_8$

Octal  $\rightarrow$  means base 8

$8 = 2^3$   $\curvearrowright$  3 bit group

Every three digits of binary represent one digit of octal

010      110      111

2      6      7

Hence  $(10110111)_2 = (267)_8$



## Some Special Case }

### Case (2) : Binary to Hexadecimal base conversion

Ex.  $(\overbrace{1011011}^{\text{4 bits}})_2 = (\quad )_{16}$

Hexadecimal  $\rightarrow$  means base 16

$16 = 2^4$  4 bits group

Every four digits of binary represent one digit of Hexadecimal.

0101 1011

5      11(B)

Hence  $(1011011)_2 = (5B)_{16}$



ANS

$\stackrel{Q}{=}$ 

$$(3 \ 2 \ 1)_4 = (\quad)_{16}$$

$\swarrow 2^2$        $\searrow 2^1$

$$\left( \begin{array}{cc} \overline{00\ 11} & \overline{10\ 01} \\ \hline 3 & 9 \end{array} \right)_2 = (3\ 9)_{16}$$

Ale -

## BCD (Binary Coded Decimal)

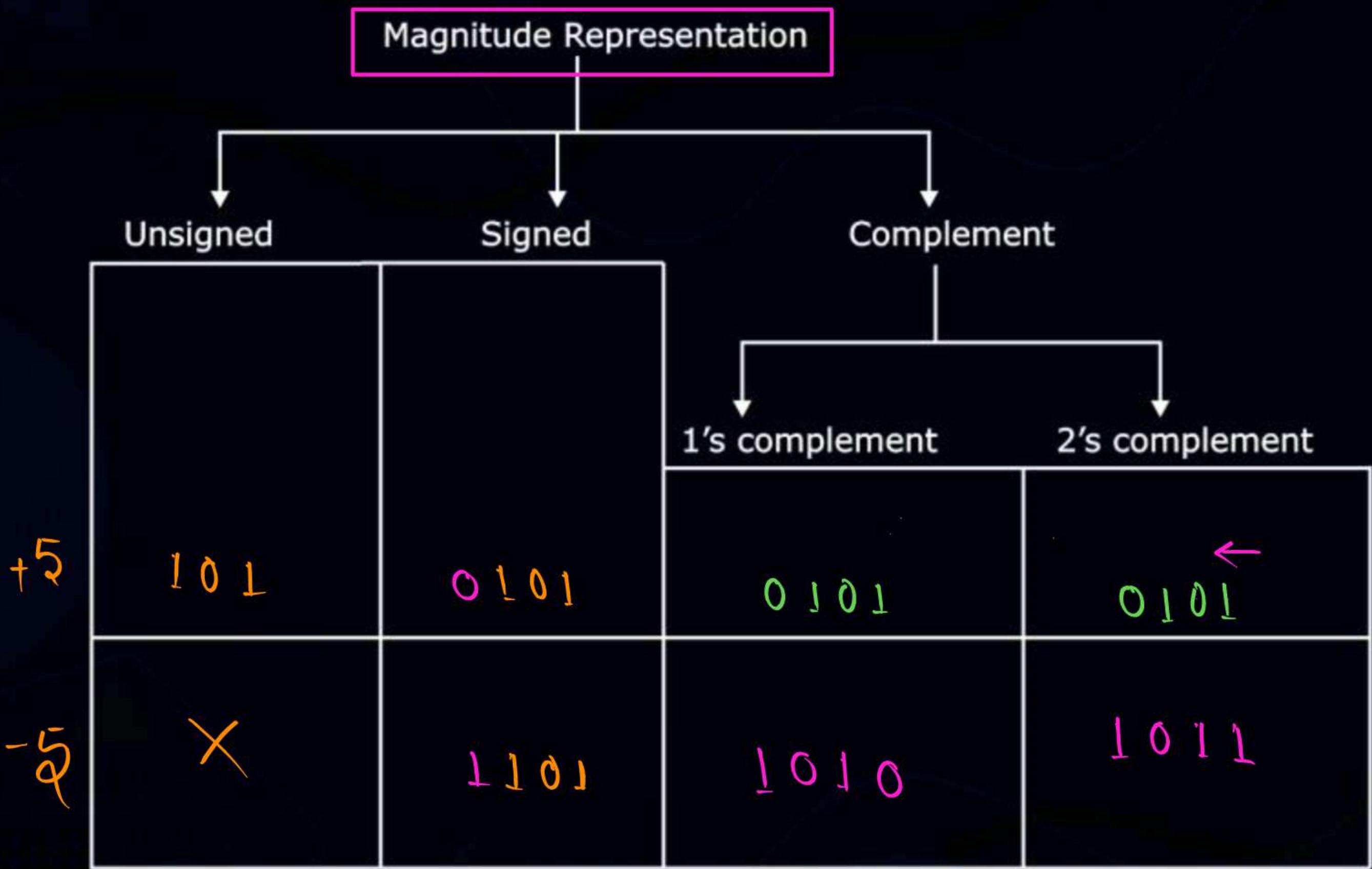
In this each digit of the decimal number is represented by its four-bit binary equivalent. It is also called natural BCD or 8421 code. It is weighted code.

**Excess - 3 Code:** This is an non weighted binary code used for decimal digits. Its code assignment is obtained from the corresponding value of BCD after the addition of 3.

**BCO (Binary Coded Octal):** In this each digit of the Octal number is represented by its three-bit binary equivalent.

**BCH (Binary Coded Hexadecimal):** In this each digit of the hexadecimal number is represented by its four bit binary equivalent.

Decimal Digits	BCD 8421	Excess - 3	Octal digits	BCO	Hexadecimal Digits	BCH
0	0000	0011	0	000	0	0000
1	0001	0100	1	001	1	0001
2	0010	0101	2	010	2	0010
3	0011	0110	3	011	3	0011
4	0100	0111	4	100	4	0100
5	0101	1000	5	101	5	0101
6	0110	1001	6	110	6	0110
7	0111	1010	7	111	7	0111
8	1000	1011			8	1000
9	1001	1100			9	1001
					A	1010
					B	1011
					C	1100
					D	1101
					E	1110
					F	1111



written in 1's complement. convert it into equivalent

Q① 001011 →  $+11$  decimal

Q② 1110010 →  $-13$

00 011011

written in 2's complement. Convert it into equivalent decimal.

Q① 001011 →  $+11$

Q② 1110010 →  $-14$

0001110

→  $+14$

Q Let  $X$  be the number of distinct 16 bit integers in 2's complement representation, Let  $Y$  be the number of distinct 16 bit integer in signed Representation

Then  $X - Y$  is \_\_\_\_\_ ?

GATE-2016

$$n=16$$

$$Y = -(2^{n-1}) \text{ to } (2^{n-1}-1)$$

$$-32768 \text{ to } 32767$$

$$Y \boxed{65535}$$

$$X \rightarrow -(2^{n-1}) \text{ to } (2^{n-1}-1)$$

$$X \rightarrow -(2^{15}) \text{ to } (2^{15}-1) \quad \times$$

$$X \rightarrow -32768 \text{ to } 32767 \rightarrow \boxed{65536}$$

$$X - Y = 65536 - 65535 = 1$$

$$\therefore x \rightarrow -(2^{n-1}) \text{ to } +(2^{n-1}-1)$$

$$y \rightarrow -(2^{n-1}-1) \text{ to } (2^{n-1}-1)$$

$$x \rightarrow z$$

$$y \rightarrow \cancel{z}-1$$

$$x-y = z - (z-1)$$

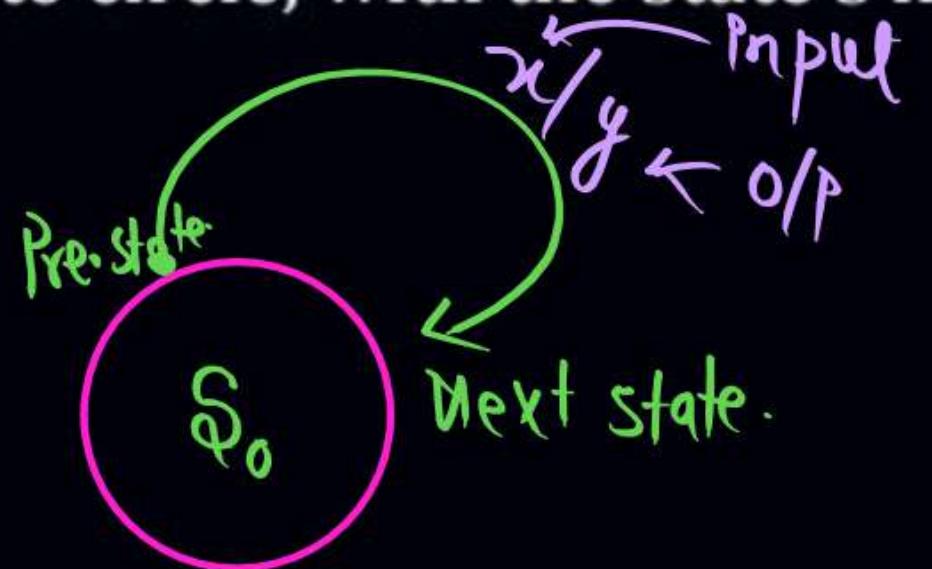
$$= \cancel{z} - \cancel{z} + 1$$

$$= 1$$

## FSM, Sequence Detector

### States Diagrams

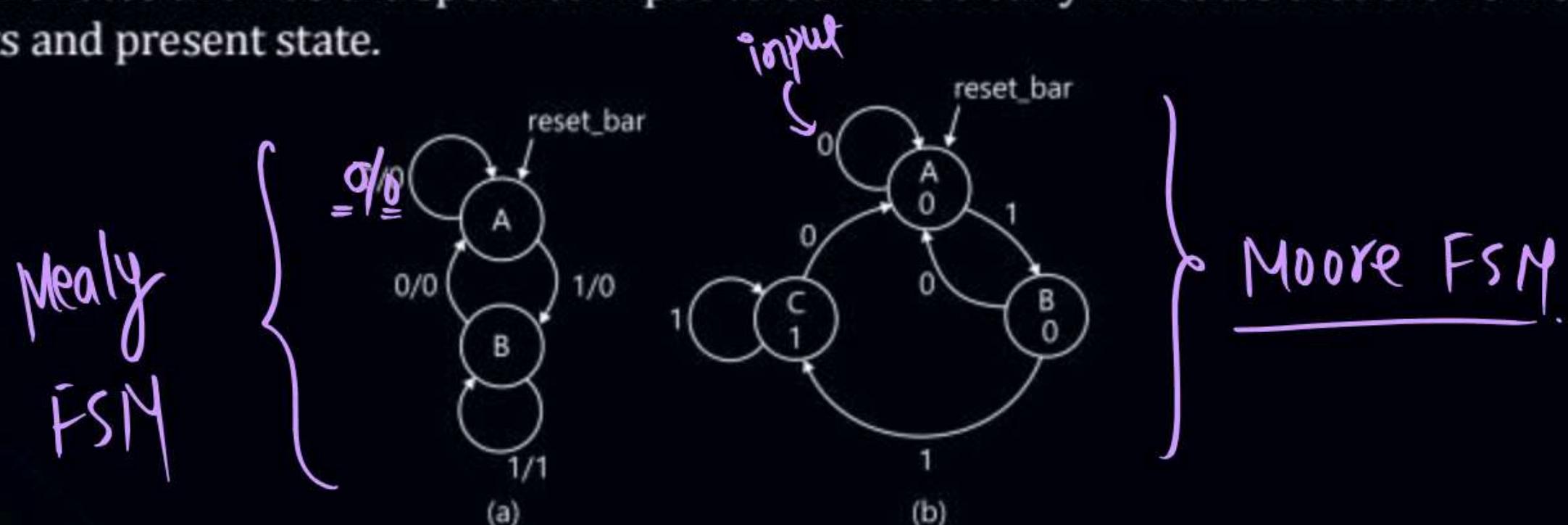
A state diagram provides an abstract graphical representation of the operation of an FSM. It allows the conceptualization of the FSM's operation to be separated from its implementation. Each individual state of the FSM is represented by a state circle, with the state's name located inside.



# FSM, Sequence Detector

## States Diagrams

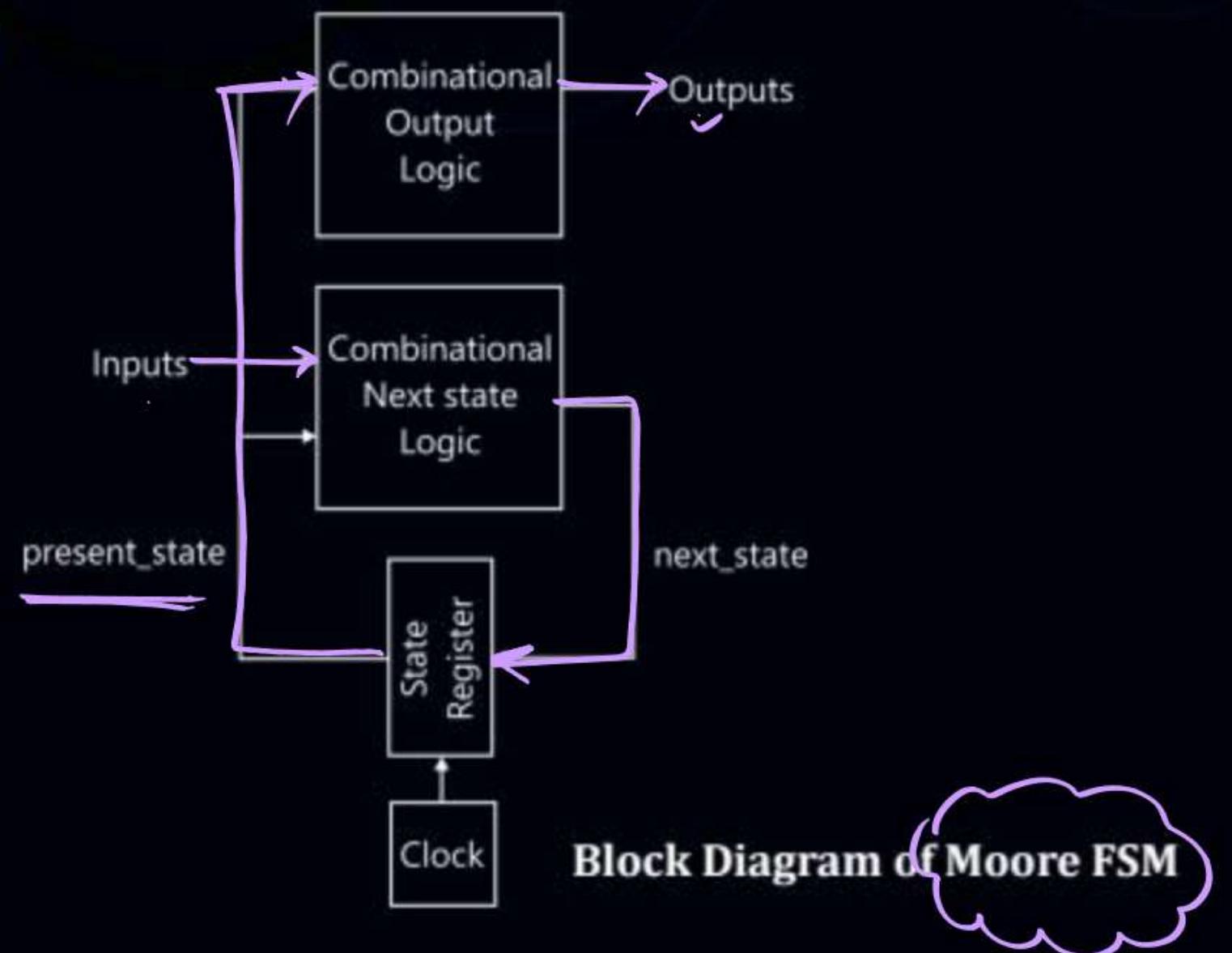
**Mealy FSM state diagram** has two states, A and B. A state diagram for a Mealy FSM has each directed arc labelled with an input/output value pair. This value pair indicates the FSM's output when it is in the state from which the arc emanates and has the specified input value. This clearly indicates that the FSM's outputs are a function of its inputs and present state.



State diagrams of an (a) Mealy machine and (b) Moore machine

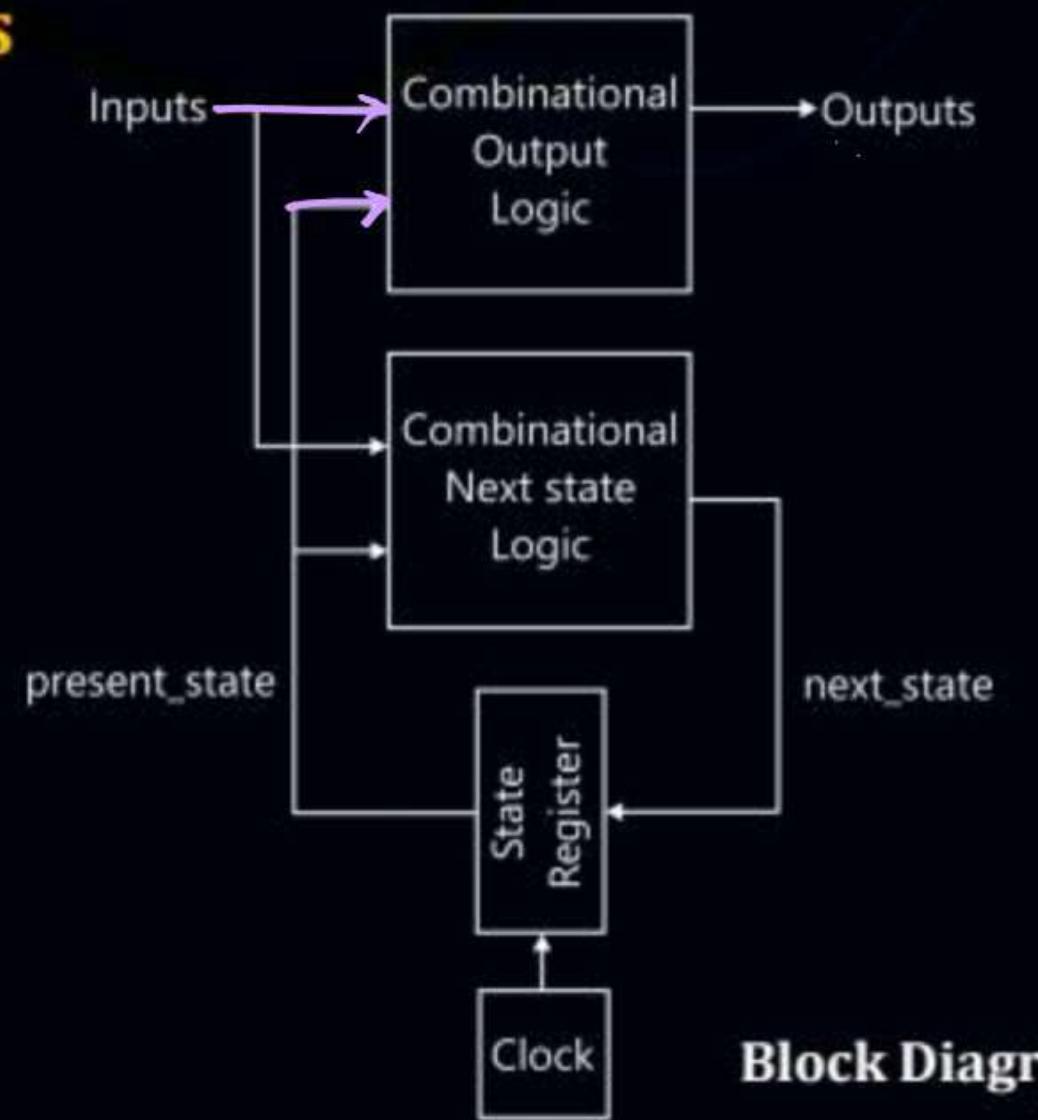
# FSM, Sequence Detector

## Block Diagrams



# FSM, Sequence Detector

## Block Diagrams

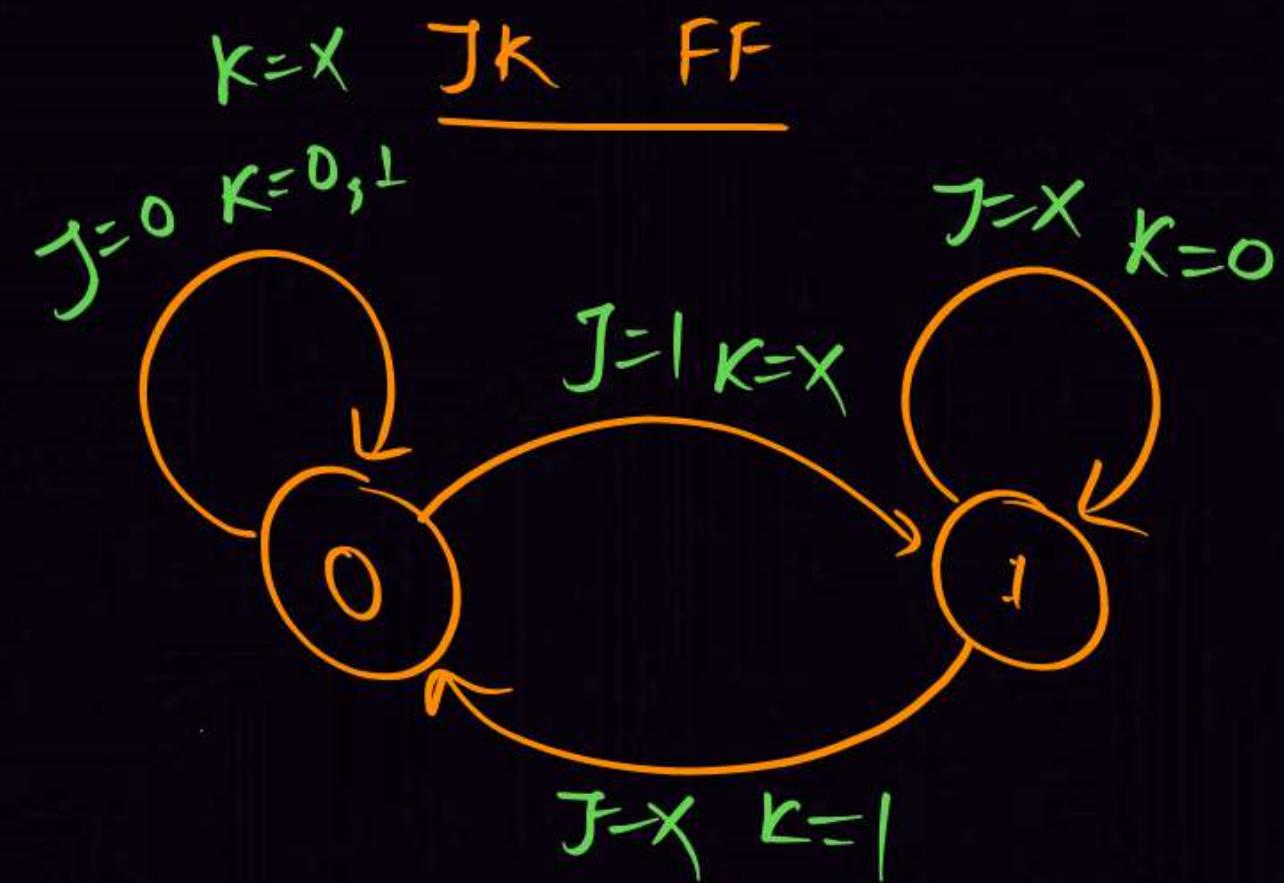
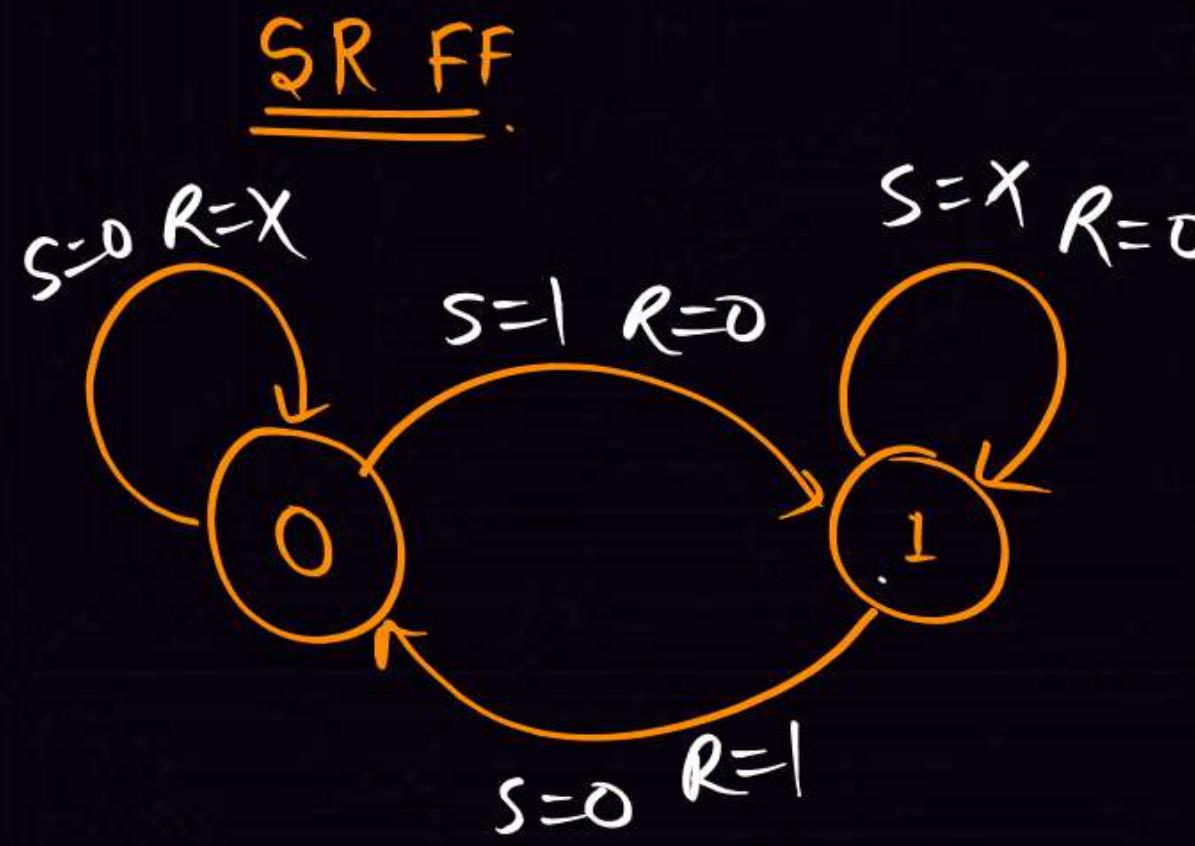


Block Diagram of Mealy FSM

# FSM, Sequence Detector

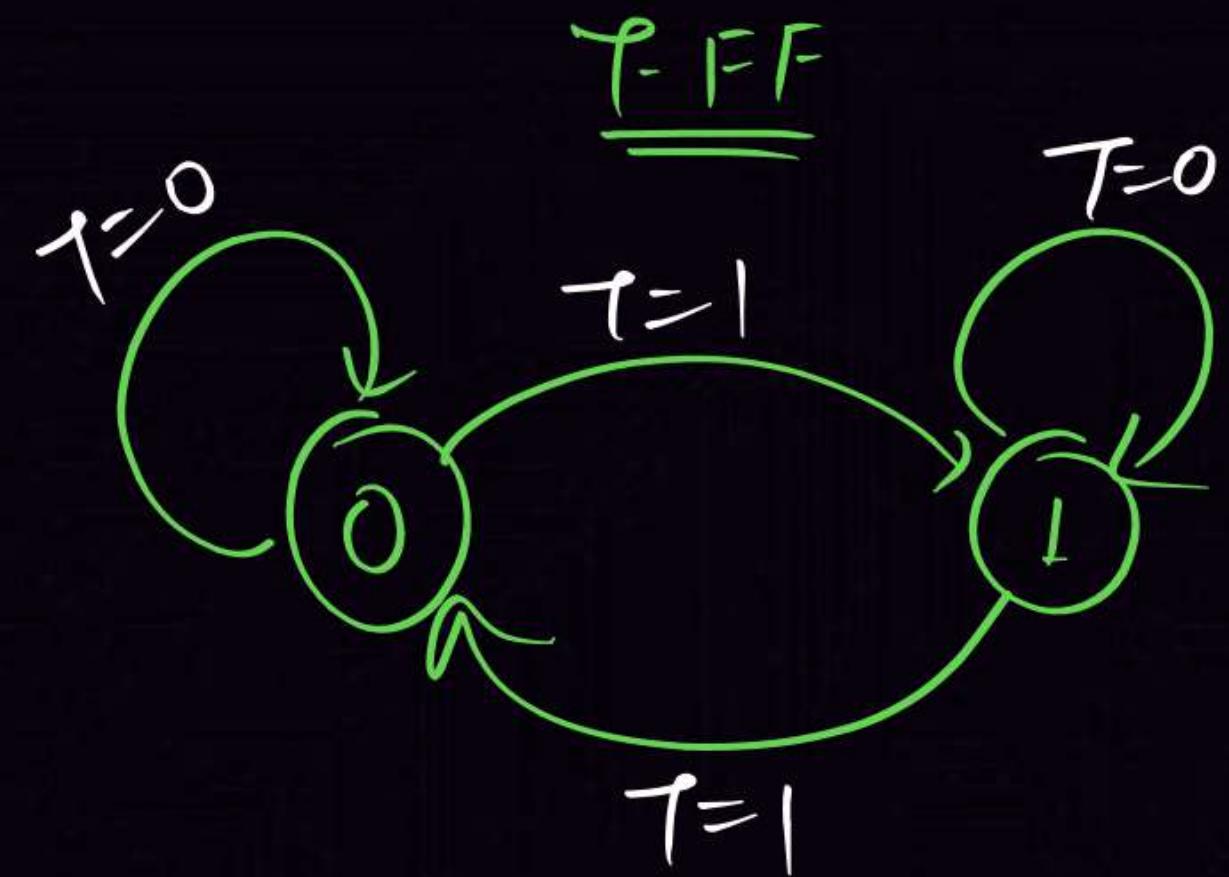
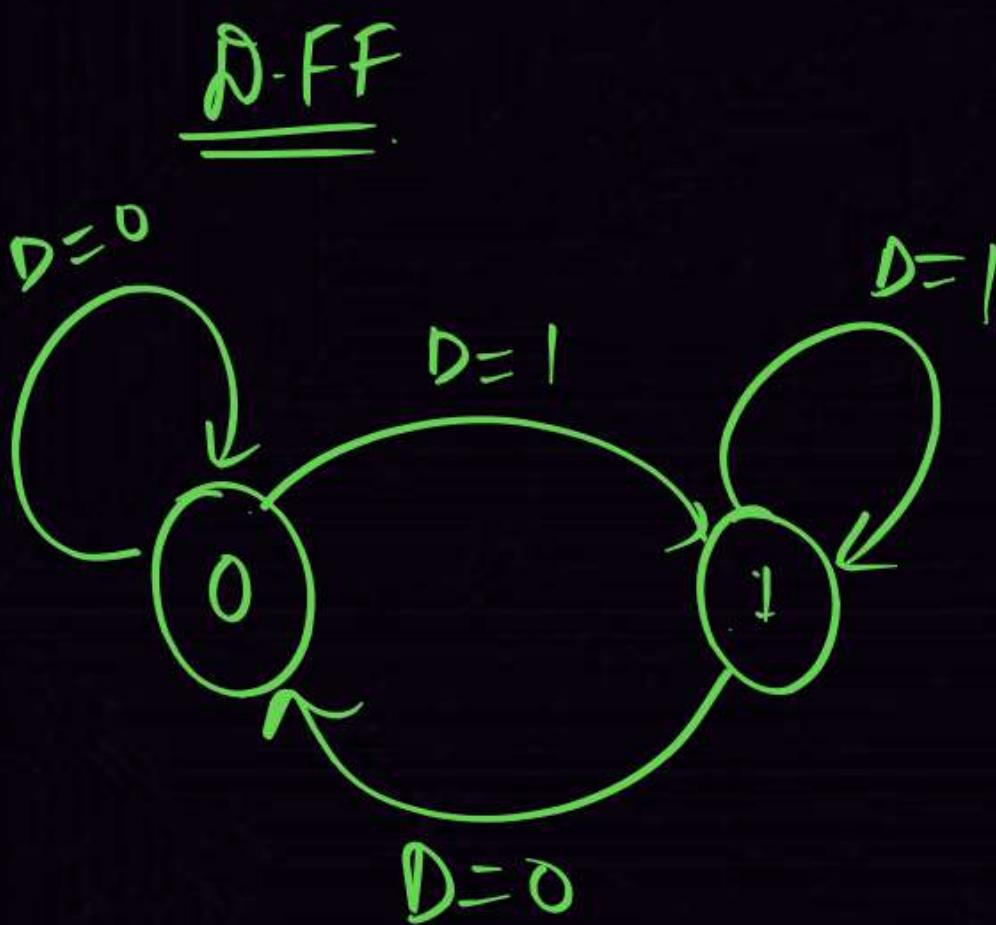
## Difference between Moore and Mealy FSM

S. No.	MOORE FSM	MEALY FSM
1.	Outputs of Moore FSM are a function of its present state.	Outputs of Mealy FSM are a function of its present input and present state.
2.	Outputs change only at triggering clock edge.	Outputs change as soon as any of its input changes. However, states cannot change until triggering clock edge.
3.	Conceptually simpler but usually required more states.	Require fewer states than Moore FSM a output can have different values in a single state.
4.	Since, it has more circuit delays, more logic is needed to decode the output(s) and therefore it reacts slower to input(s) in comparison with Mealy FSM.	React faster to inputs.



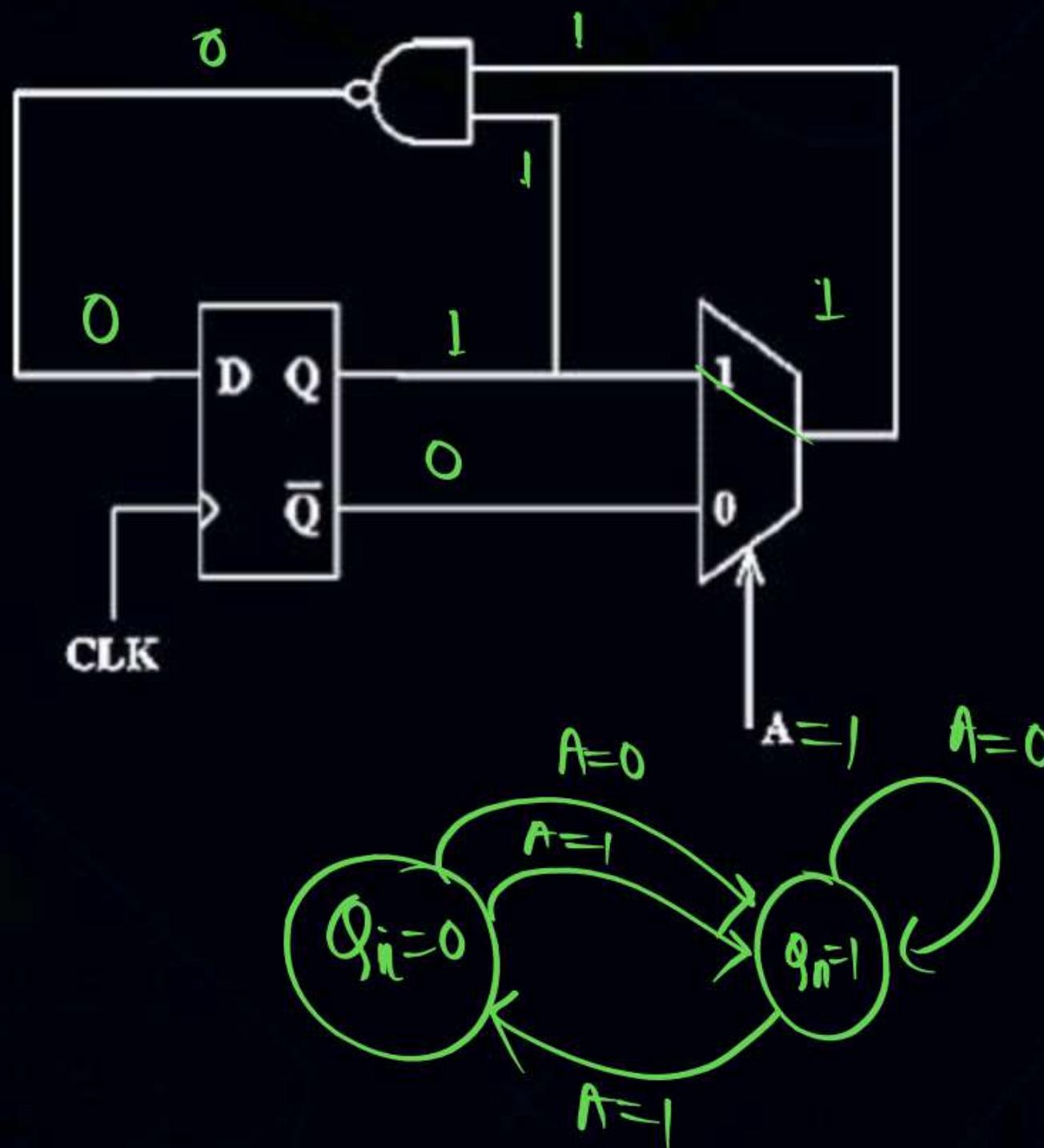
$Q_n$	$Q_{n+1}$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	X		

$Q_n$	$Q_{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	X	X	0

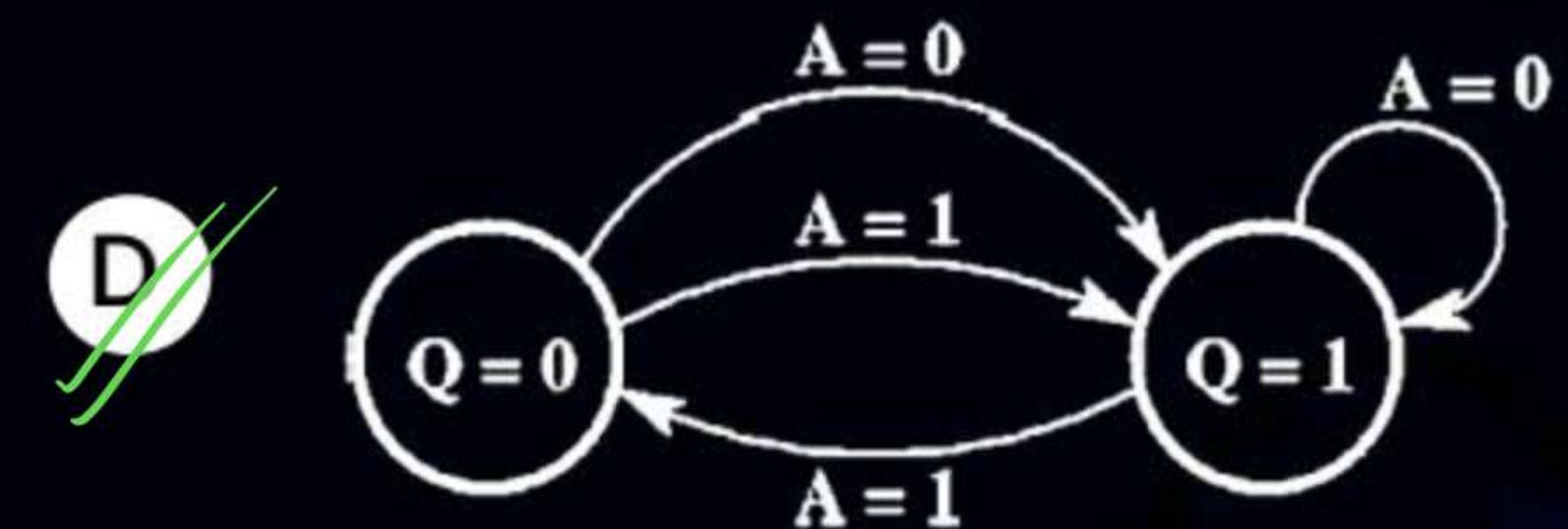
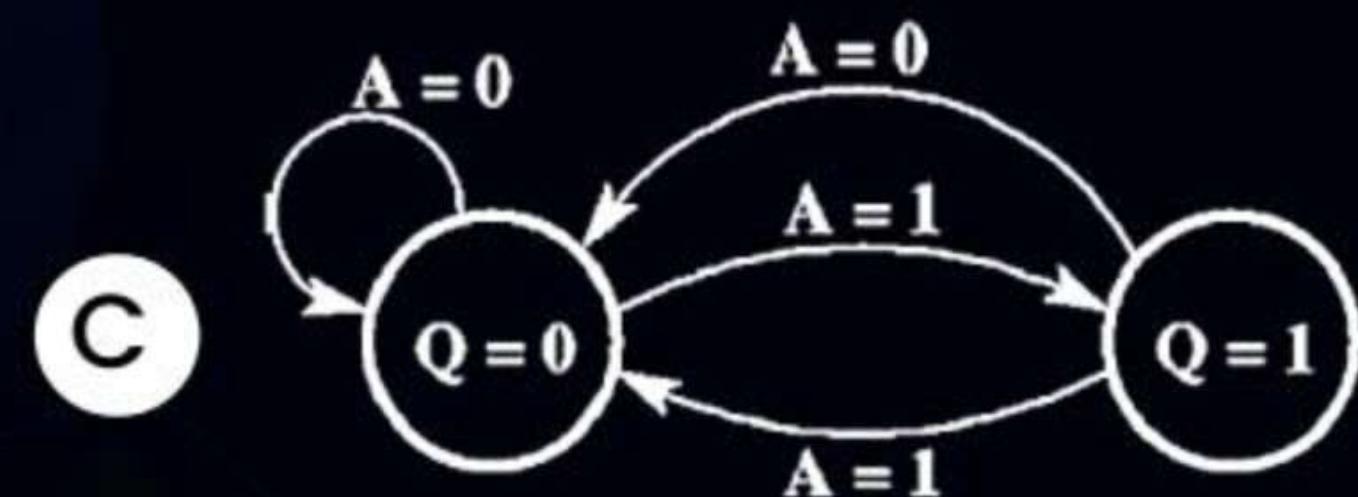
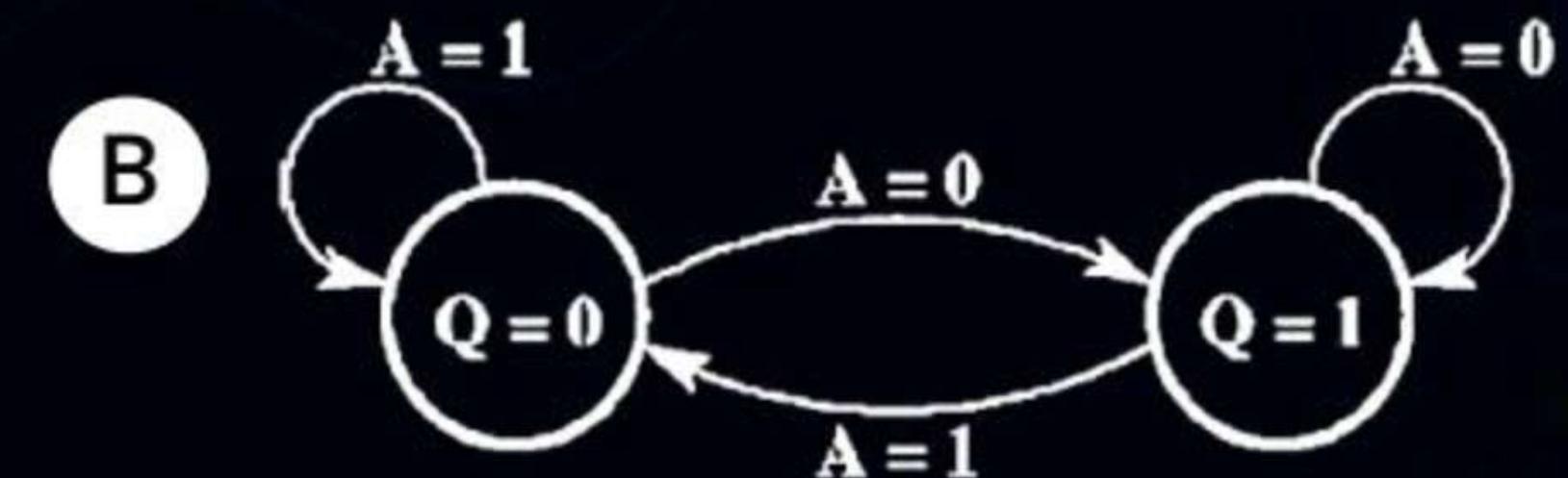
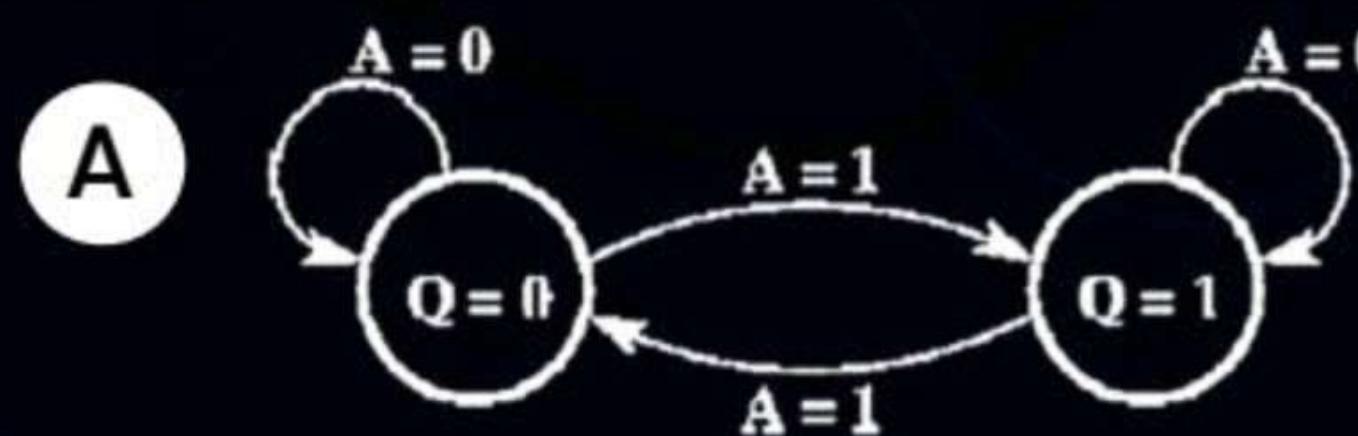


**MCQ**

The state transition diagram for the circuit shown is

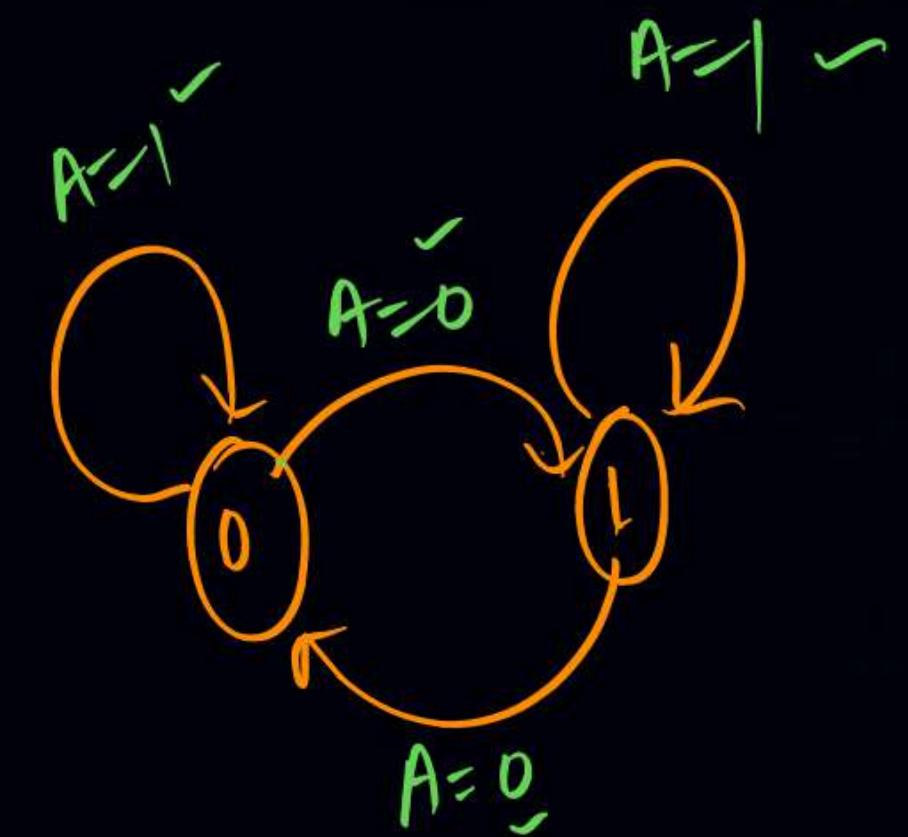
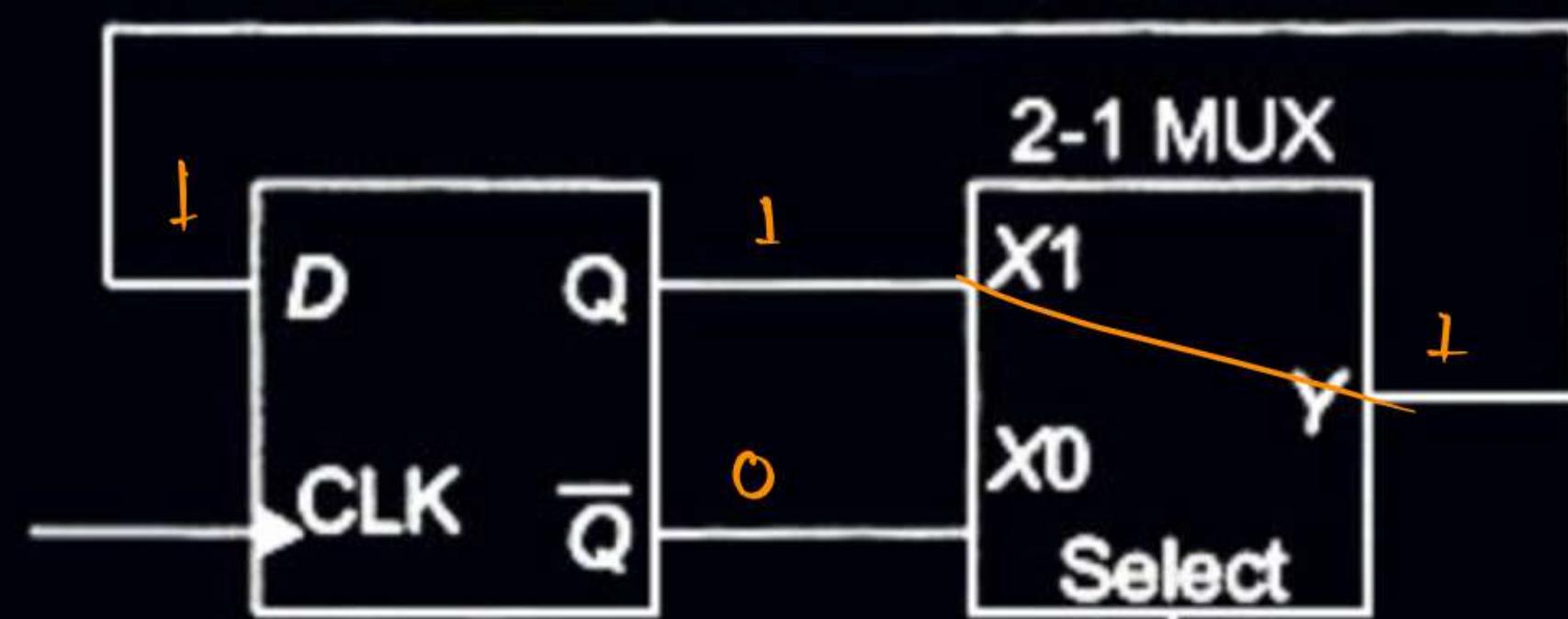


$A$	$q_n$	$q_{n+1}$
0	0	1
0	1	1
1	0	1
1	1	0

**MCQ**

**MCQ**

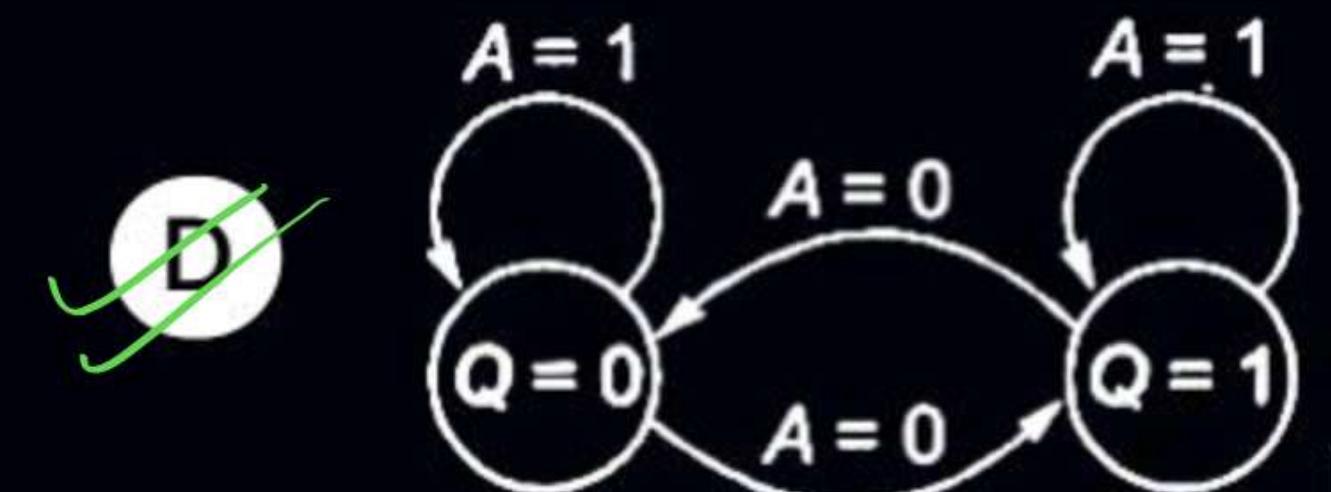
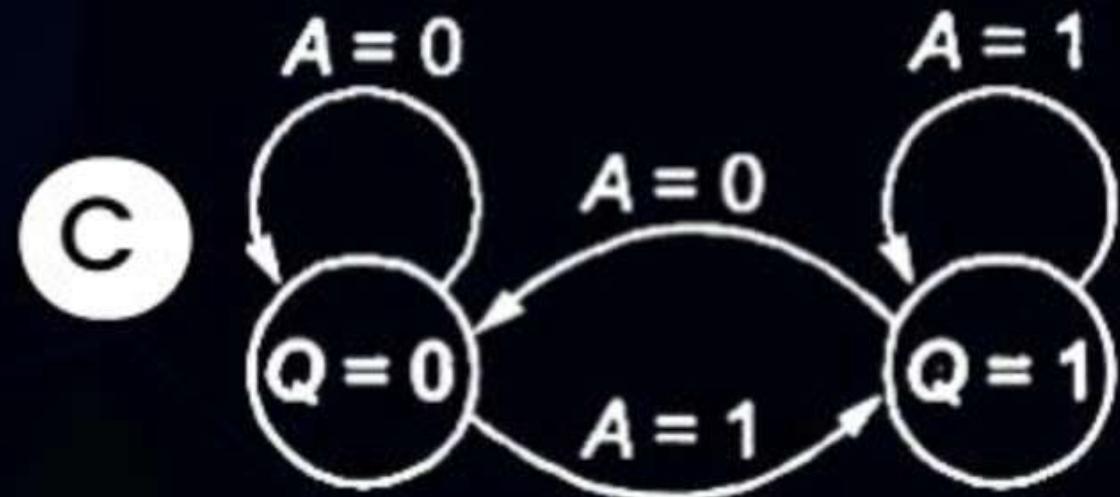
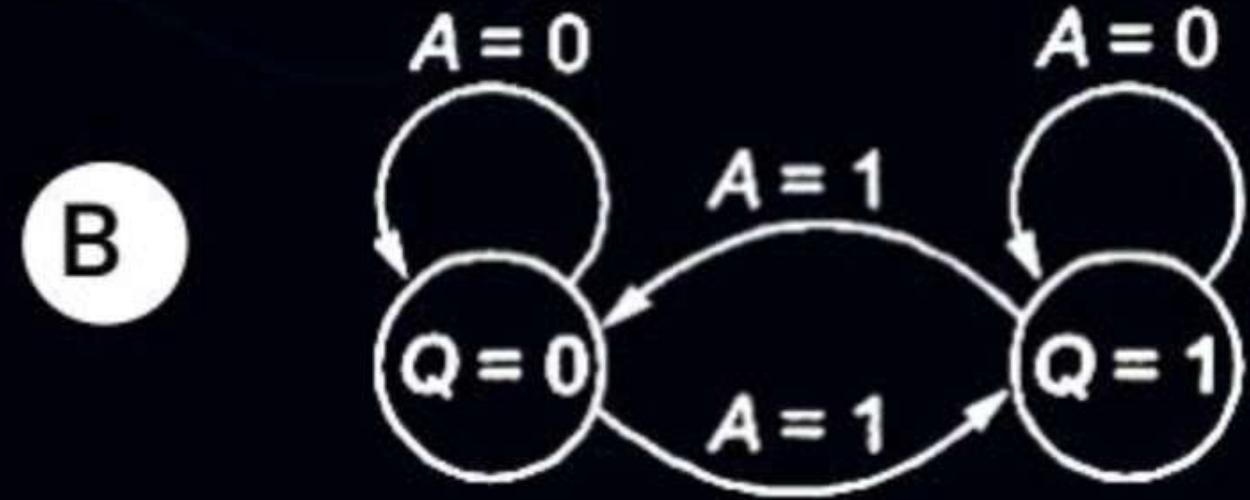
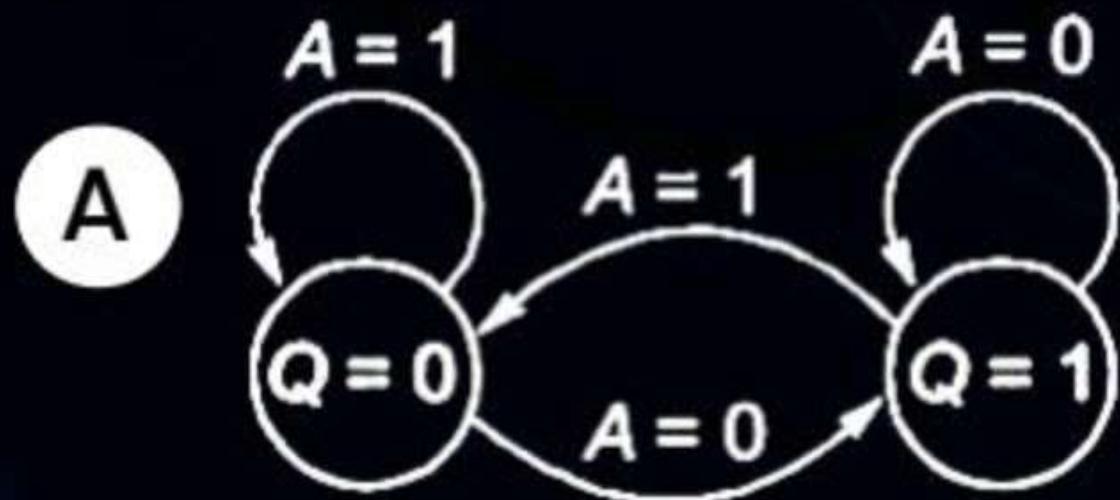
The state transition diagram for the logic circuit shown in



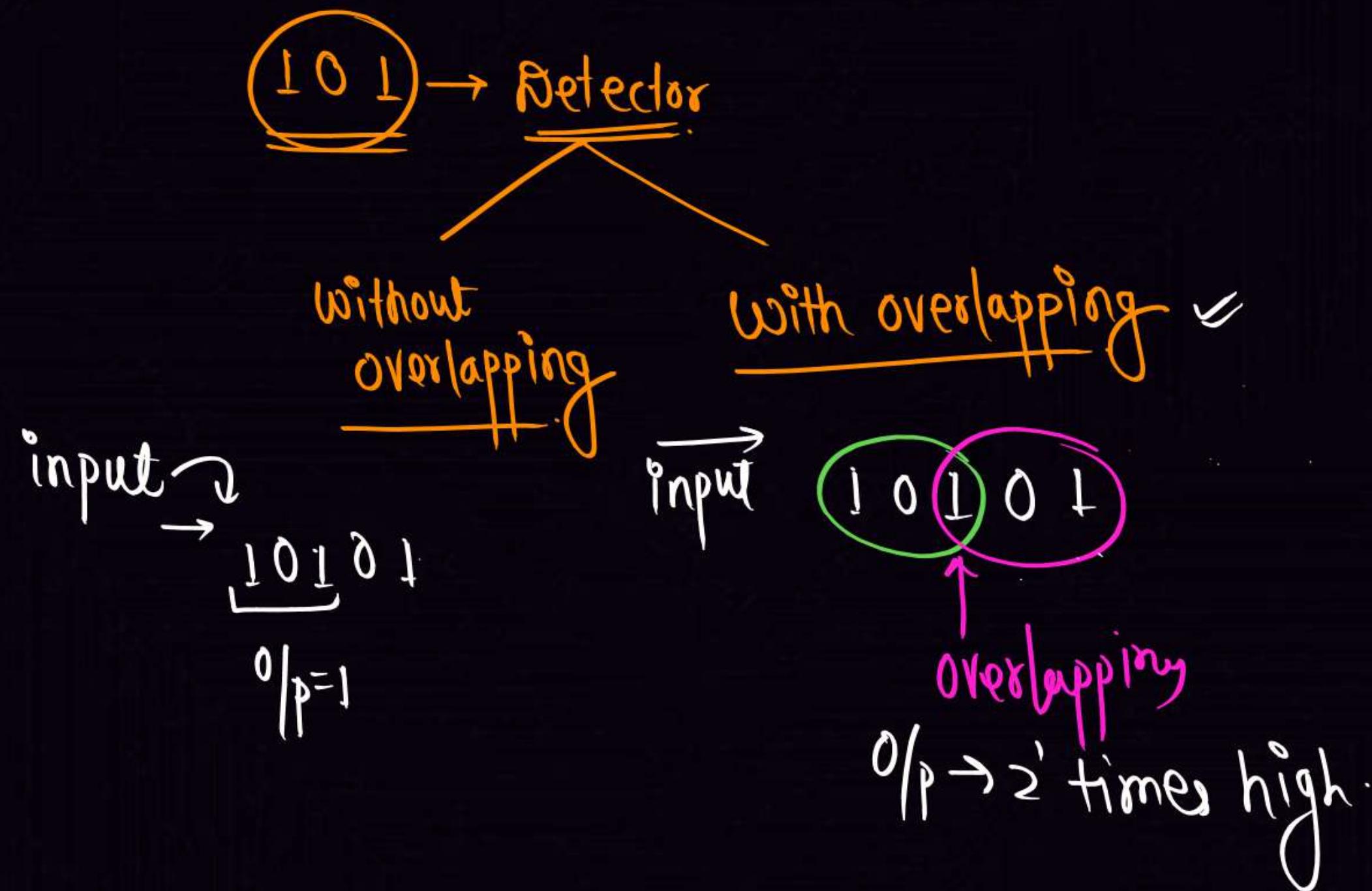
A	$Q_{in}$	$Q_{out}$
0	0	1
0	0	0
1	0	0
1	1	1

# MCQ

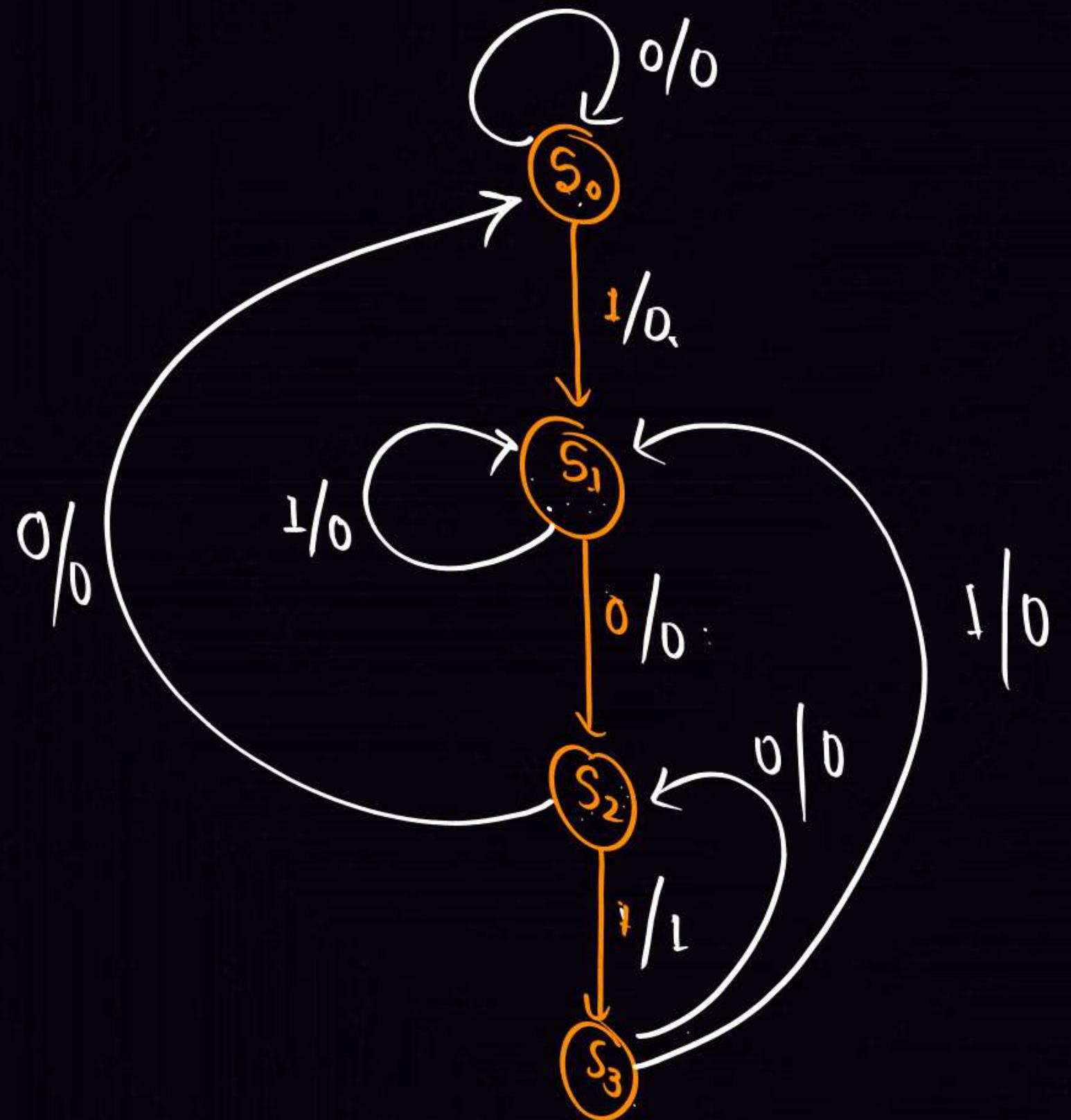
P  
W



# Sequence Detector



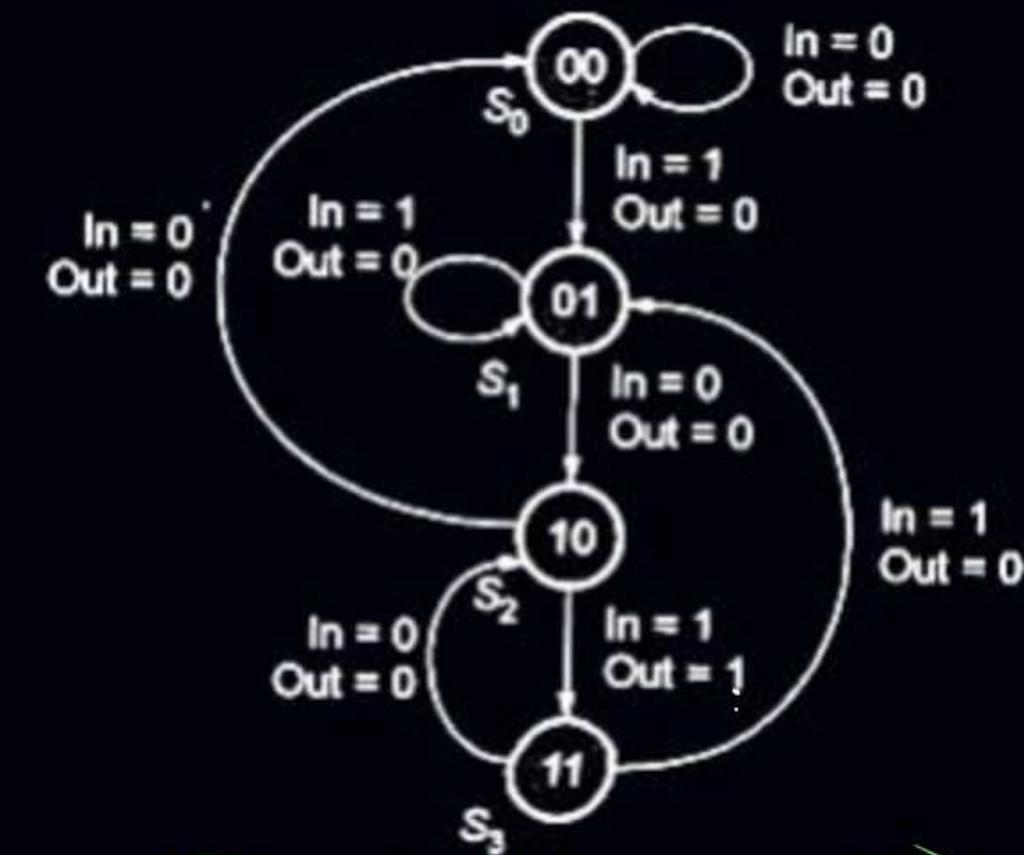
# 101 Detector with overlapping



# NAT

The state diagram of a finite state machine (FSM) designed to detect an overlapping sequence of three bits is shown in the figure. The FSM has an input 'In' and an output 'Out'. The initial state of the FSM is  $S_0$ .

101 Detector  
with  
overlapping



If the input sequence is 10101101001101, starting with the left-most bit, then the number of times 'Out' will be 1 is 4

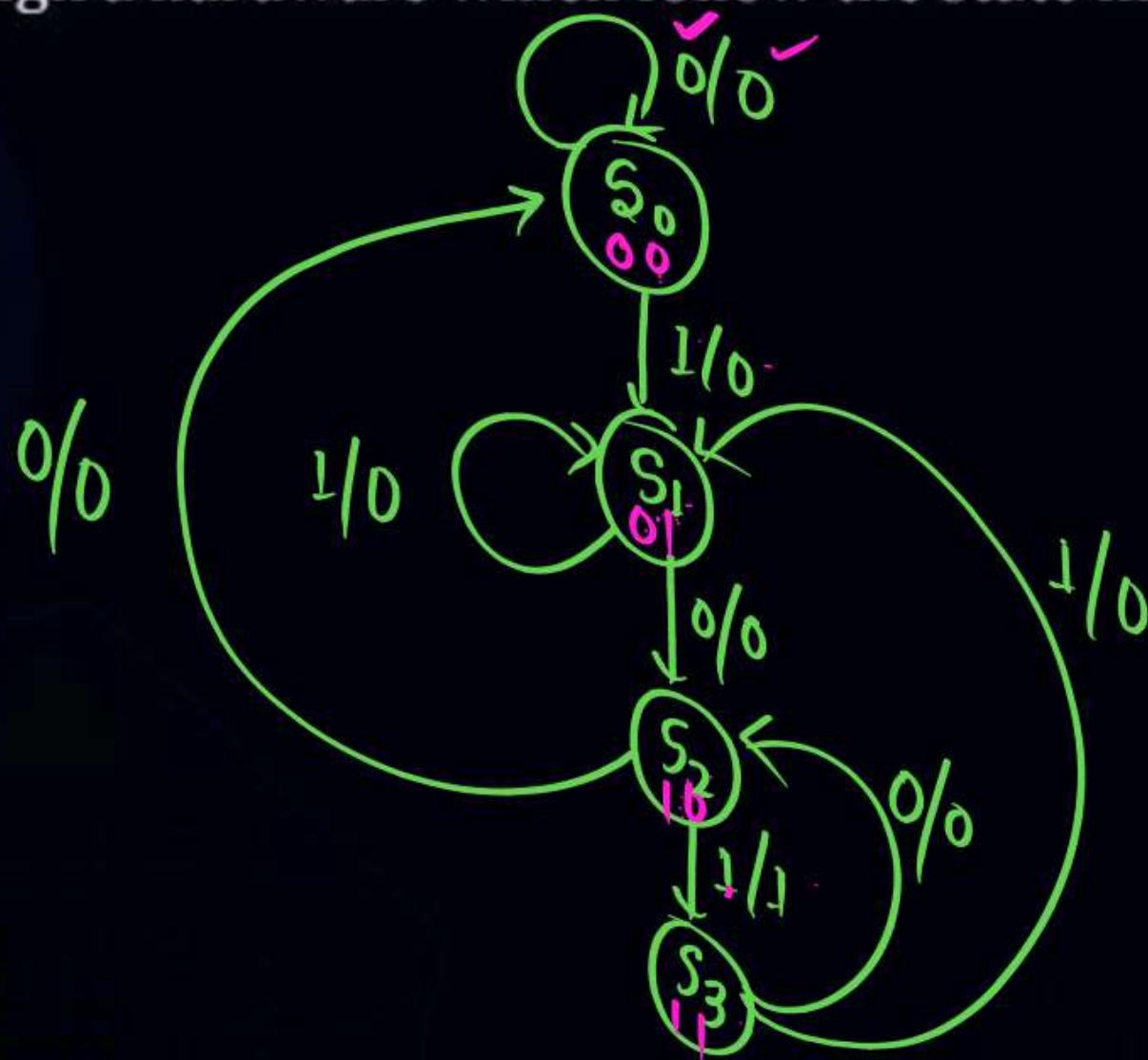
# FSM, Sequence Detector

**Hardware Design from state machine**

Example 1

By using D-FF.

Design a hardware which follow the state machine given below...

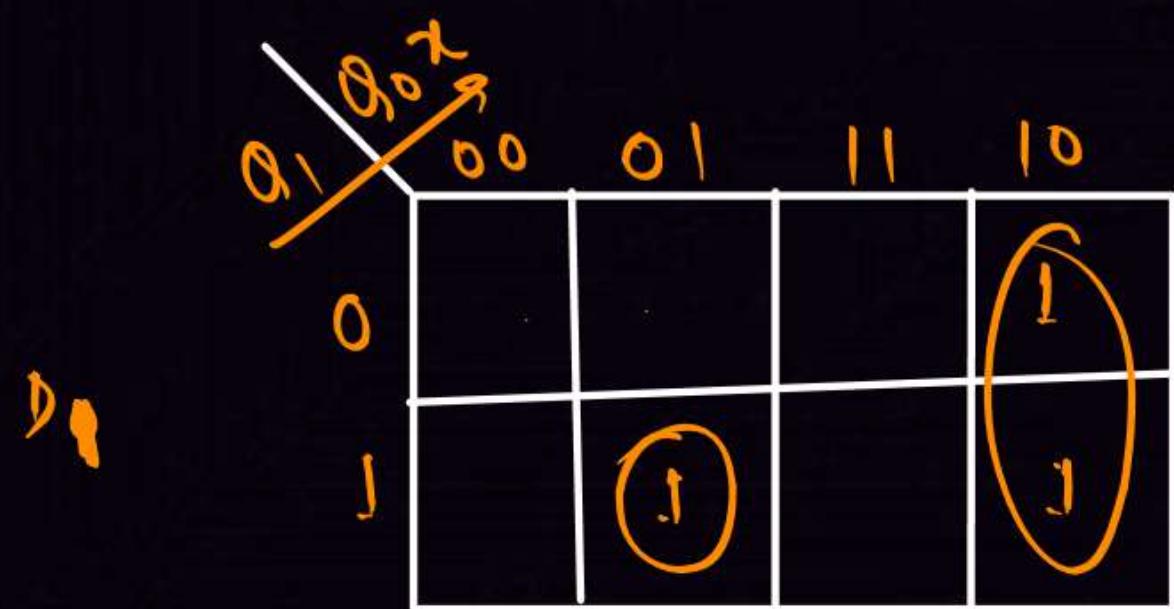


$$y = Q_1 \bar{Q}_0 x$$

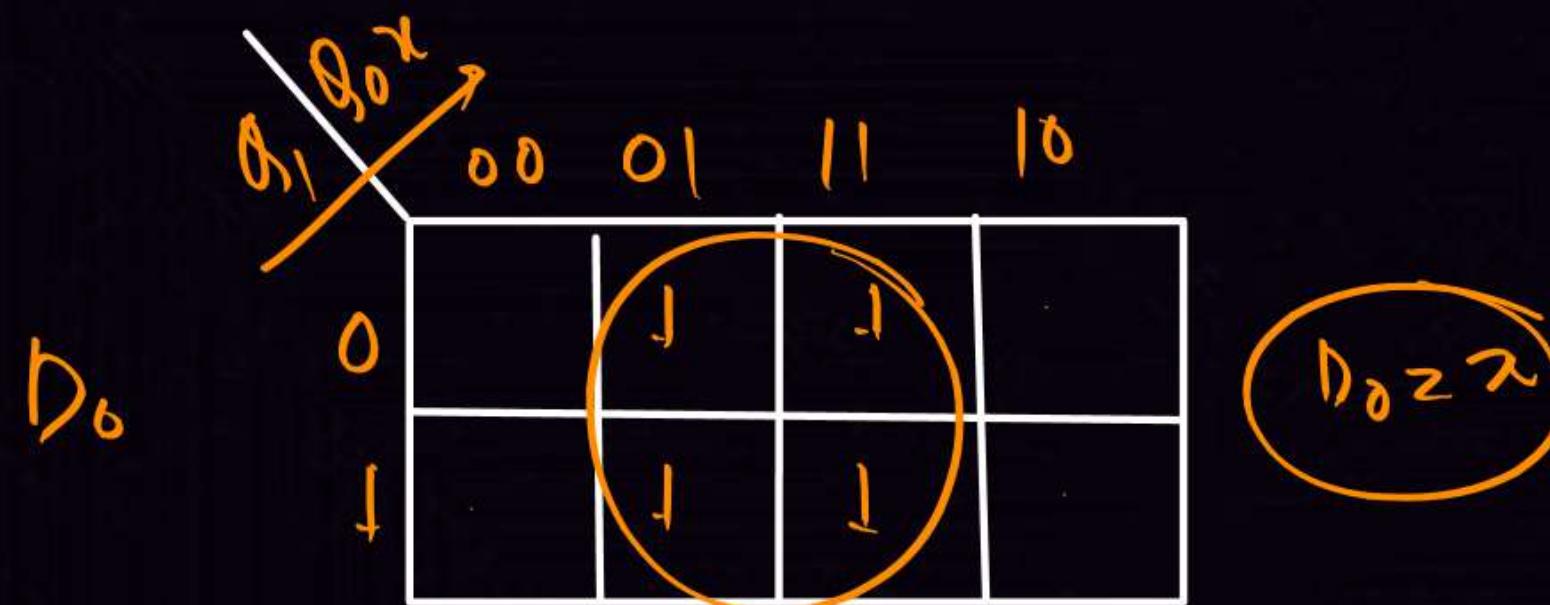
$$\begin{aligned} S_0 &= 00 \\ S_1 &= 01 \\ S_2 &= 10 \\ S_3 &= 11 \end{aligned}$$

$$\left\{ \begin{array}{l} S_0 \\ S_1 \\ S_2 \\ S_3 \end{array} \right\}$$

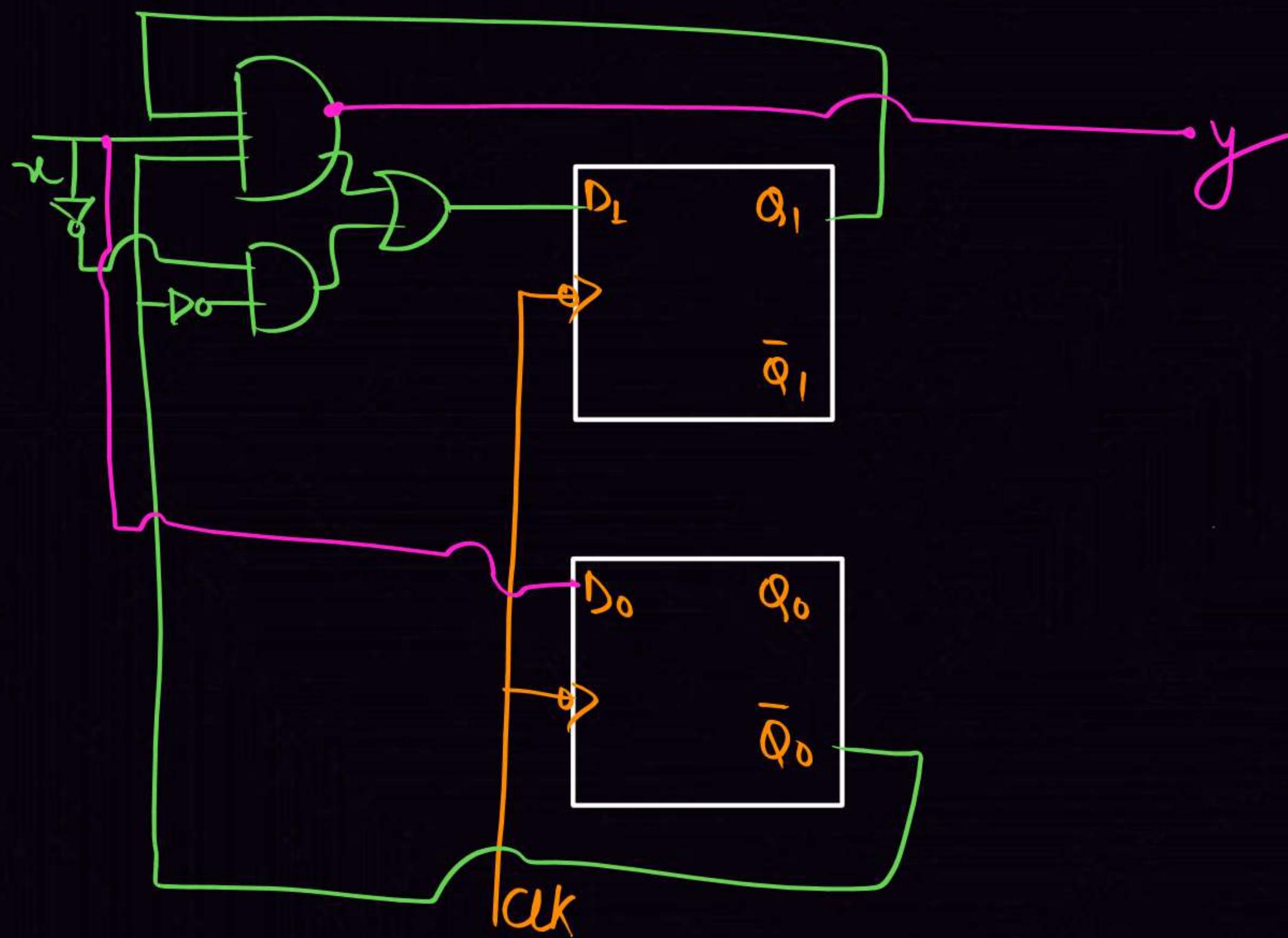
Q <sub>t</sub>	Q <sub>0</sub>	x	Q <sub>t</sub> <sup>+</sup>	Q <sub>0</sub> <sup>+</sup>	D <sub>1</sub>	D <sub>0</sub>	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1
1	1	0	1	0	1	0	0
1	1	1	0	1	0	1	0



$$D_1 = Q_1 \bar{Q}_0 x + Q_0 \bar{x}$$



$$D_0 = x$$



## Resolution

The resolution of ADC converter refers to the smallest change in the analog output voltage correspond to change in one-bit LSB of digital. For a N-bit D/A converter, the maximum number of steps is  $2^n - 1$ . When the reference voltage is V, the Least significant Bit (LSB) value is

$$\text{Resolution} = \frac{\text{Reference Voltage}}{\text{Number of steps}} = \frac{V}{2^n - 1}$$

$$\% \text{ Resolution} = \frac{\text{Step size}}{\text{Full scale}} \times 100\%$$

## Types of ADC }

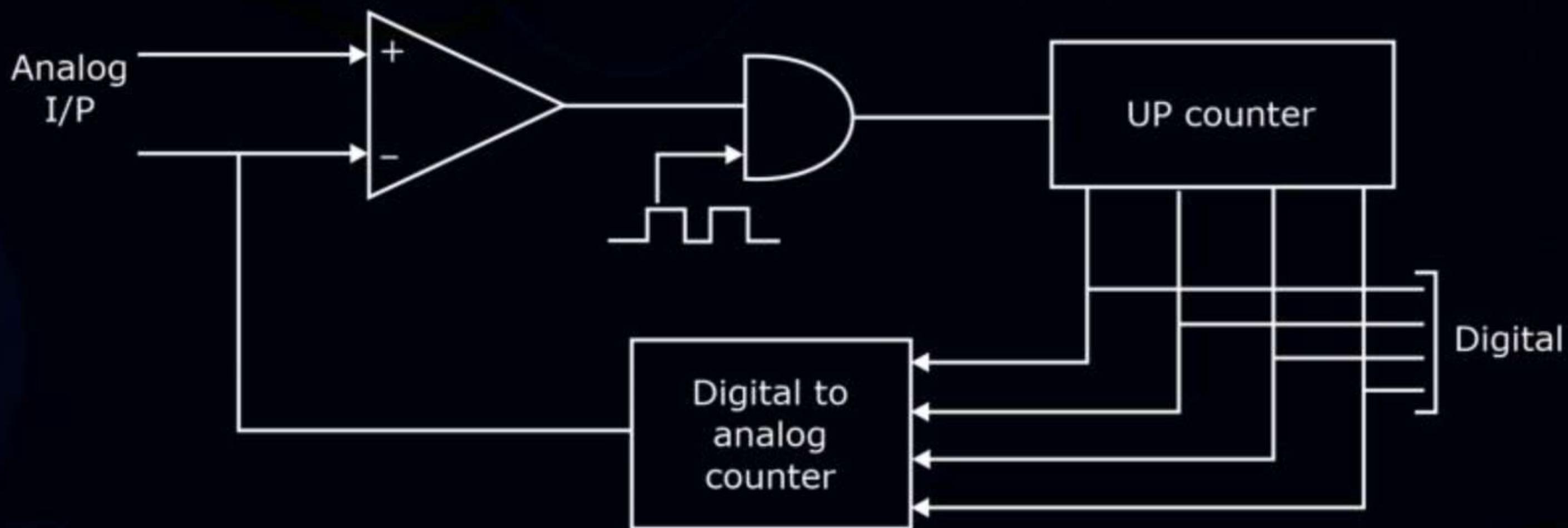
1. Counter Type ADC
2. Successive approximation Register Type ADC (SAR)
3. Flash Type ADC
4. Dual Slope ADC

$$\text{Resolution} = \frac{V_R}{2^n - 1}$$

$$\% \text{ Resolution} = \frac{1}{2^n - 1} \times 100$$

$$\% \text{ Accuracy} = \frac{\text{Error}}{V_{FS}} \times 100$$

## Counter Type ADC



## Analog to Digital Converter

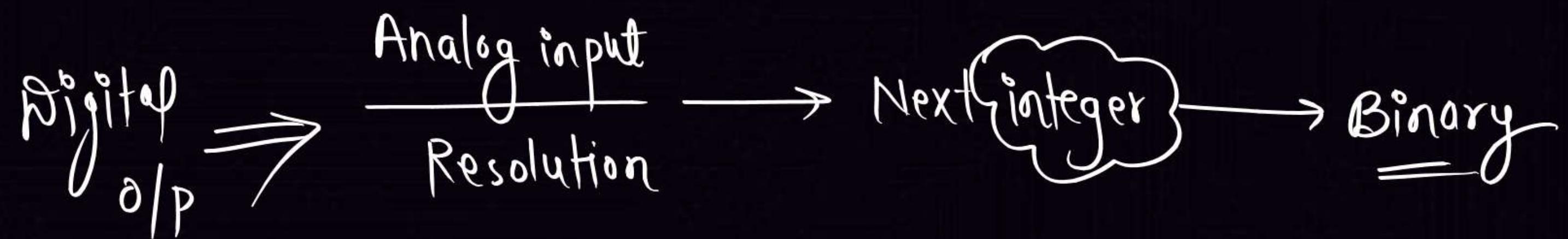
Conversion time :

$$\text{Minimum Conversion Time} = 1 \cdot T_{\text{CCLK}}$$

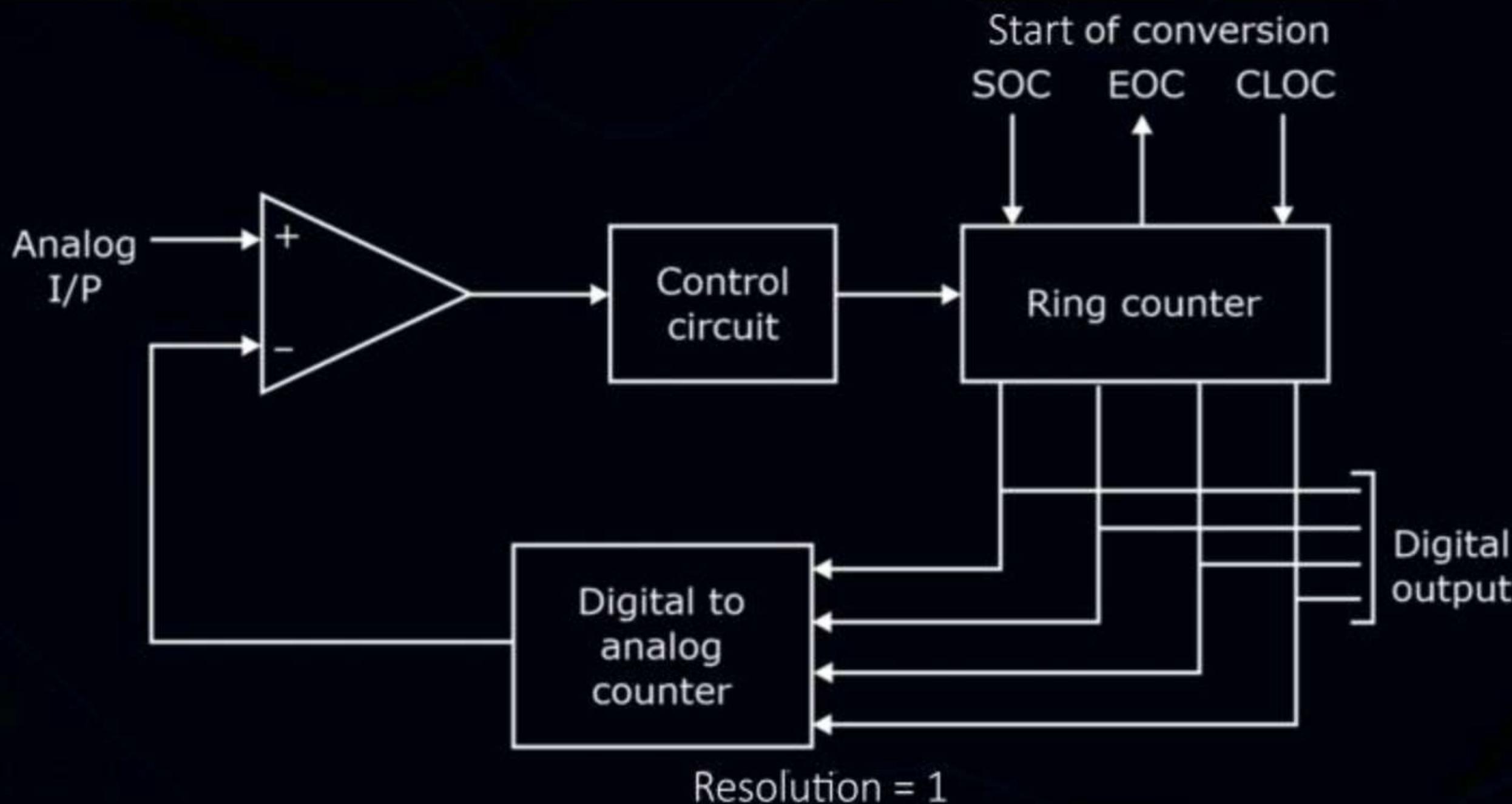
$$\text{Maximum Conversion Time} = (2^n - 1) T_{\text{CCLK}}$$

$$\text{Average Conversion Time} = (2^{n-1}) T_{\text{CCLK}}$$

Conversion time is a function of analog input, As analog input increases the conversion time will also increases hence it is known as **ramp type ADC** or **stair type ADC**.



# Successive Approximation Register Type ADC(SAR)



(soc) (EOC) X

$$\text{Min} = \text{Max} = \text{Average} = n \cdot T_{CLK}$$

(soc) (EOC) ✓

$$\text{Min} = \text{Max} = \text{Average} = (n+2)T_{CLK}$$

## 3 Bit Flash Type ADC

**3bit flash type ADC**

Resistance = 8  $\hookrightarrow$  fastest ADC.

Comparator = 7

Encoder =  $8 \times 3$

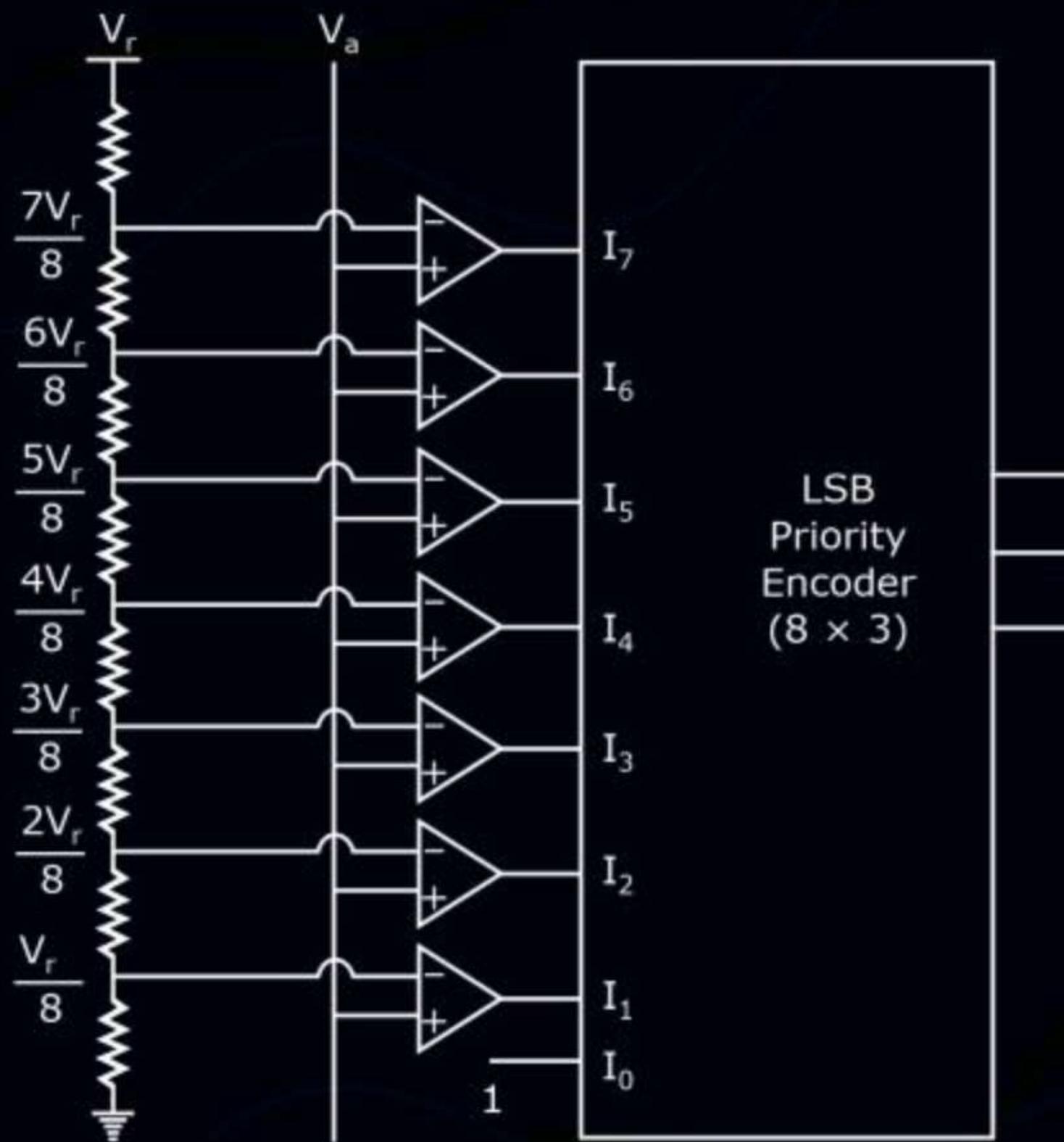
**n-bit flash type**

✓ Resistance =  $2^n$

✓ Comparator =  $2^n - 1$

Encoder =  $2^n \times n$

$\hookrightarrow$  Priority Encoder



## Flash Type ADC

- There is no clock requirement for flash type ADC but when PIPO shift resistor is used for synchronization then one clock is required for PIPO.

$V_a$	$Y_0 Y_1 Y_2$
$V_a \leq \frac{V_r}{8}$	0 0 0
$\frac{V_r}{8} < V_a \leq \frac{2V_r}{8}$	0 0 1
$\frac{2V_r}{8} < V_a \leq \frac{3V_r}{8}$	0 1 0
$\frac{3V_r}{8} < V_a \leq \frac{4V_r}{8}$	0 1 1
$\frac{4V_r}{8} < V_a \leq \frac{5V_r}{8}$	1 0 0
$\frac{5V_r}{8} < V_a \leq \frac{6V_r}{8}$	1 0 1
$\frac{6V_r}{8} < V_a \leq \frac{7V_r}{8}$	1 1 0
$\frac{7V_r}{8} < V_a \leq V_r$	1 1 1

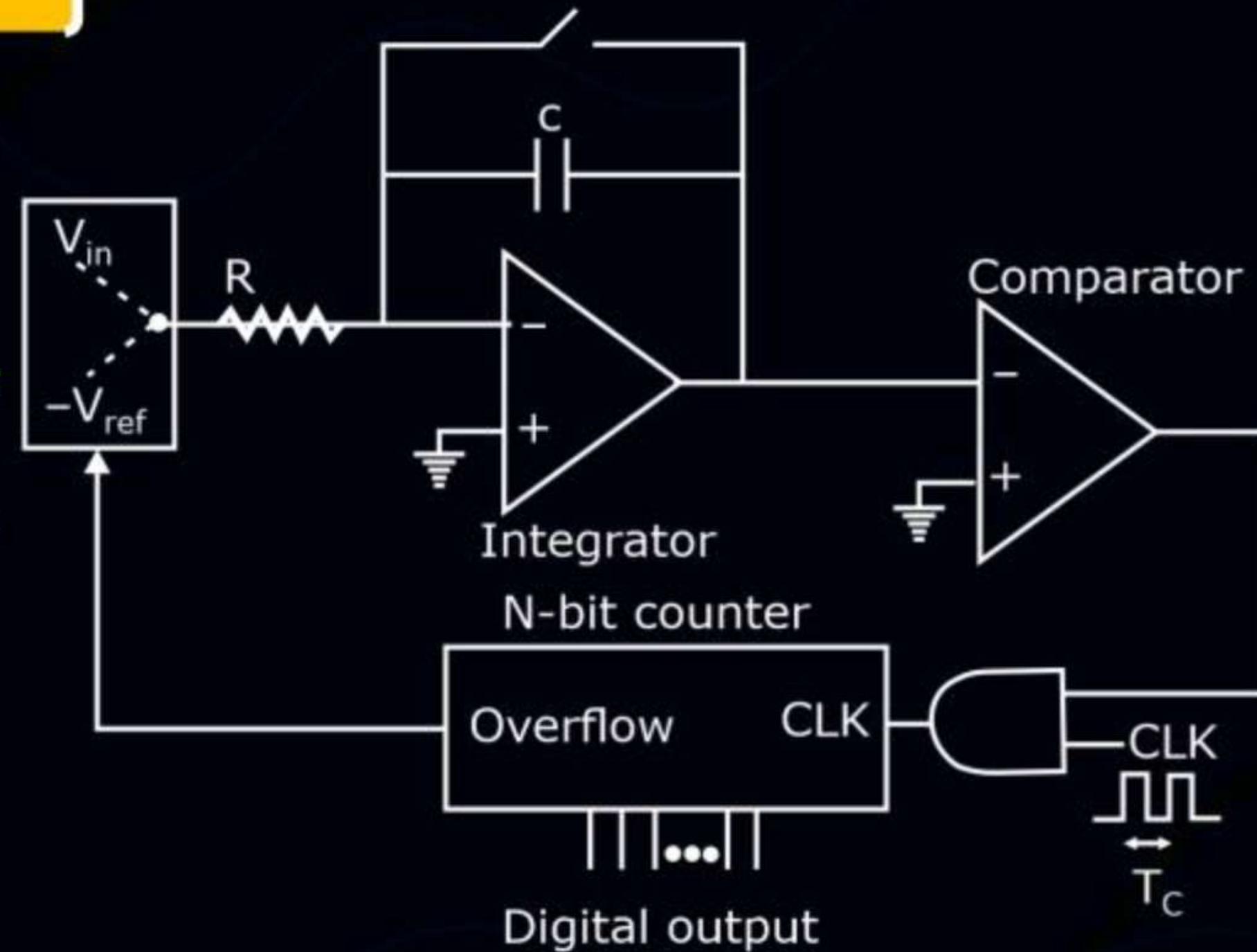
## Dual Slope ADC

↪ Slowest but

most accurate

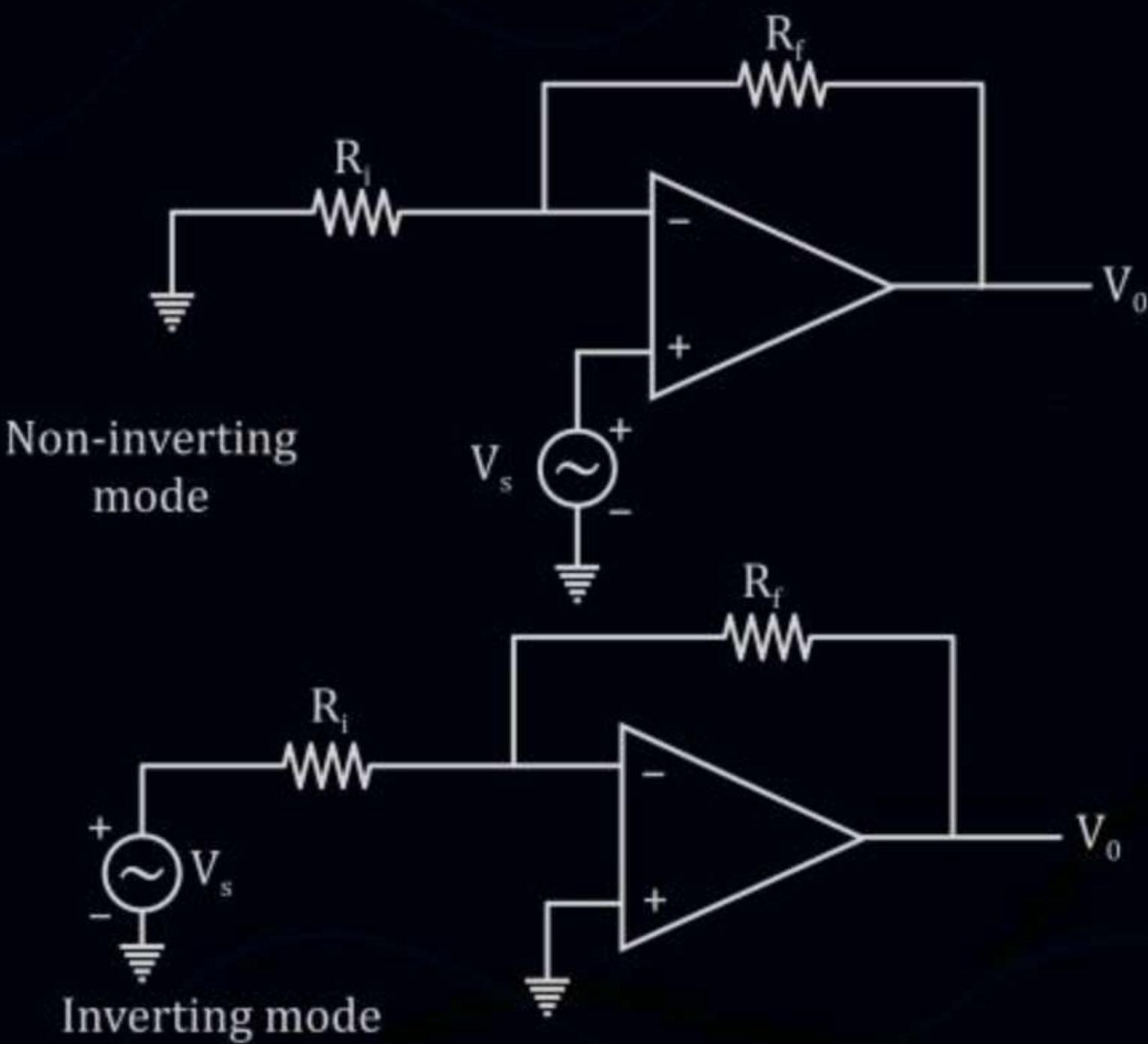
$$\text{Min conversion} = 2^n \cdot T_{\text{clk}}$$

$$\text{Maximum conversion} = (2^{n+1}) T_{\text{clk}}$$



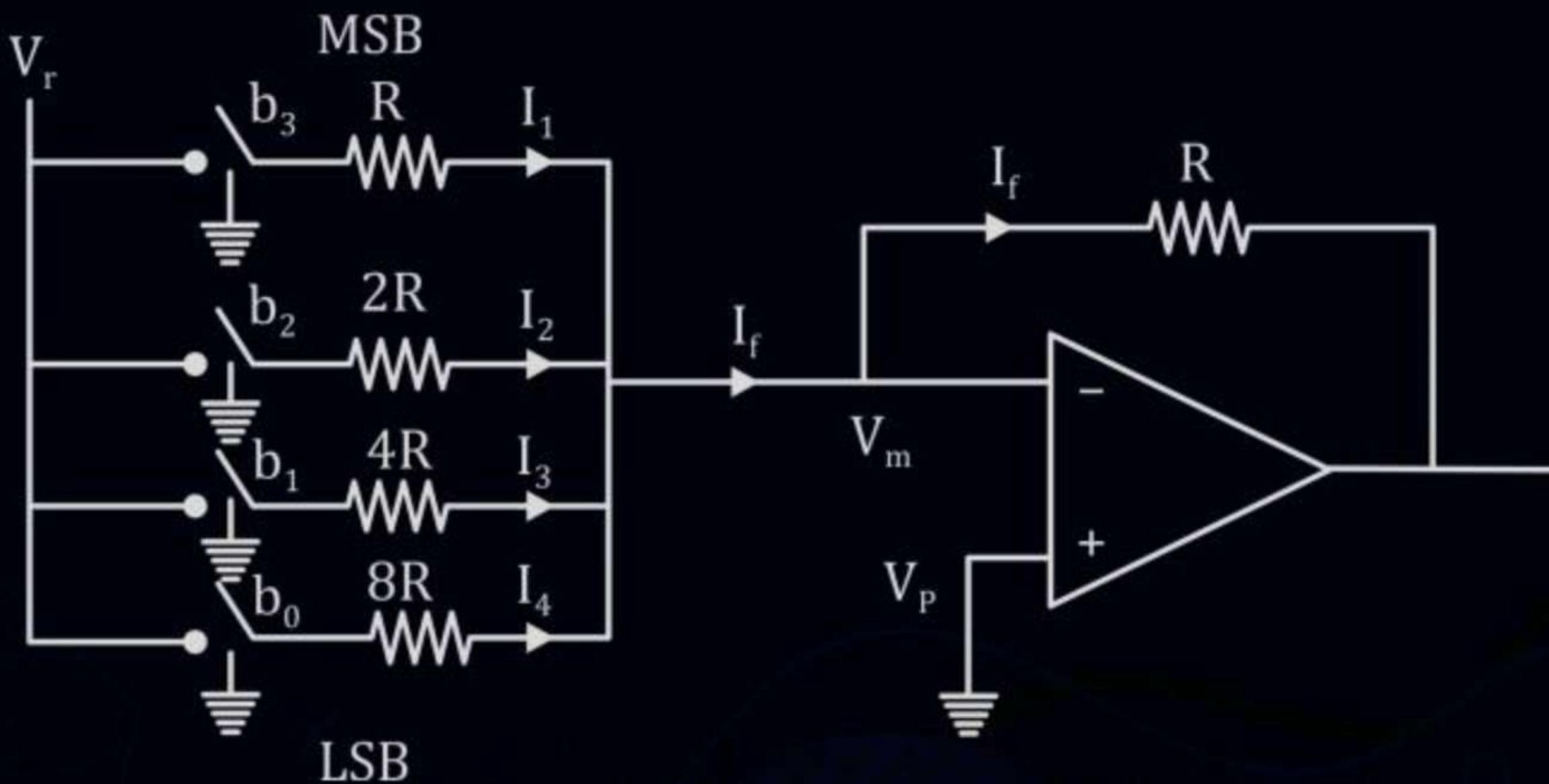
## Weighted Resistor and R-2R type DAC

Digital to Analog Converter



## ① Weighted Resistor and R-2R type DAC

Digital to Analog Converter



$V_o = \text{Resolution} \times \text{Decimal Equivalent} \times \text{Gain}$

$$V_o = \frac{V_r}{2^{n-1}} \times \sum_{i=0}^{n-1} 2^i b_i \times G$$

$$G = -\frac{R_f}{R_i} \leftarrow \text{Inverting}$$

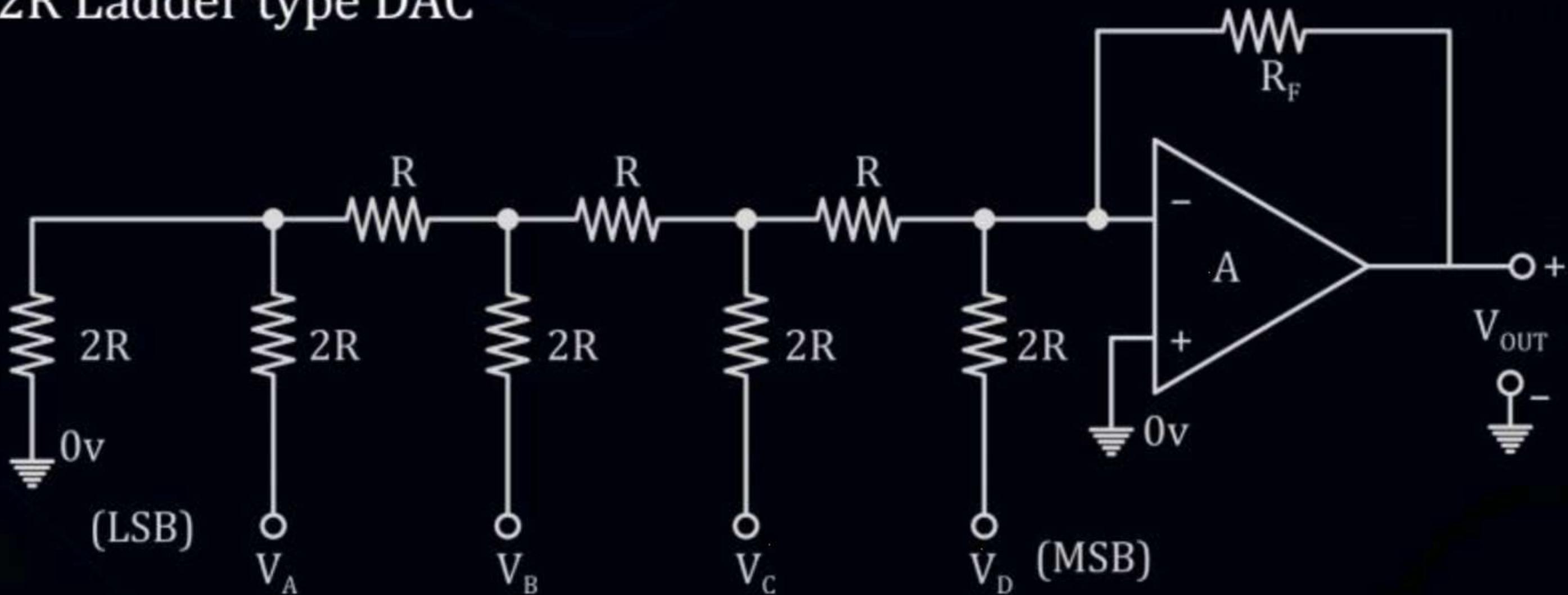
$$G = \left(1 + \frac{R_f}{R_i}\right) \rightarrow \text{Non inverting}$$

## Weighted Resistor and R-2R type DAC

Weighted Resistor type DAC

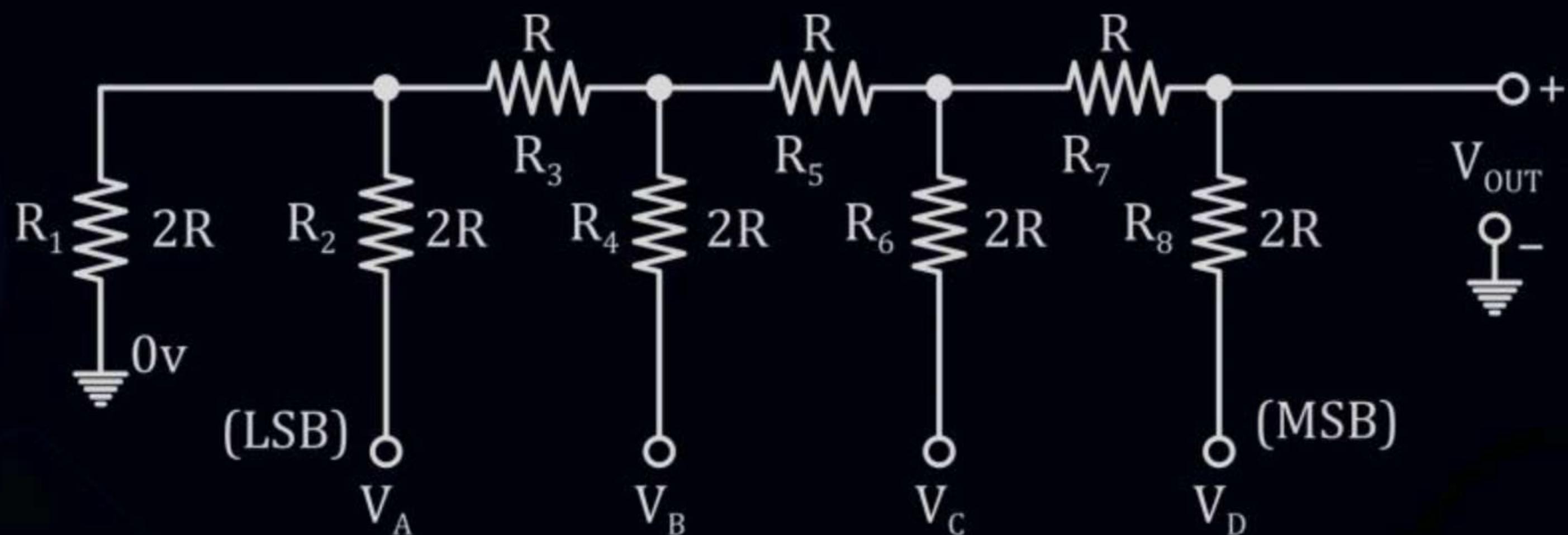
## Weighted Resistor and R-2R type DAC

R - 2R Ladder type DAC



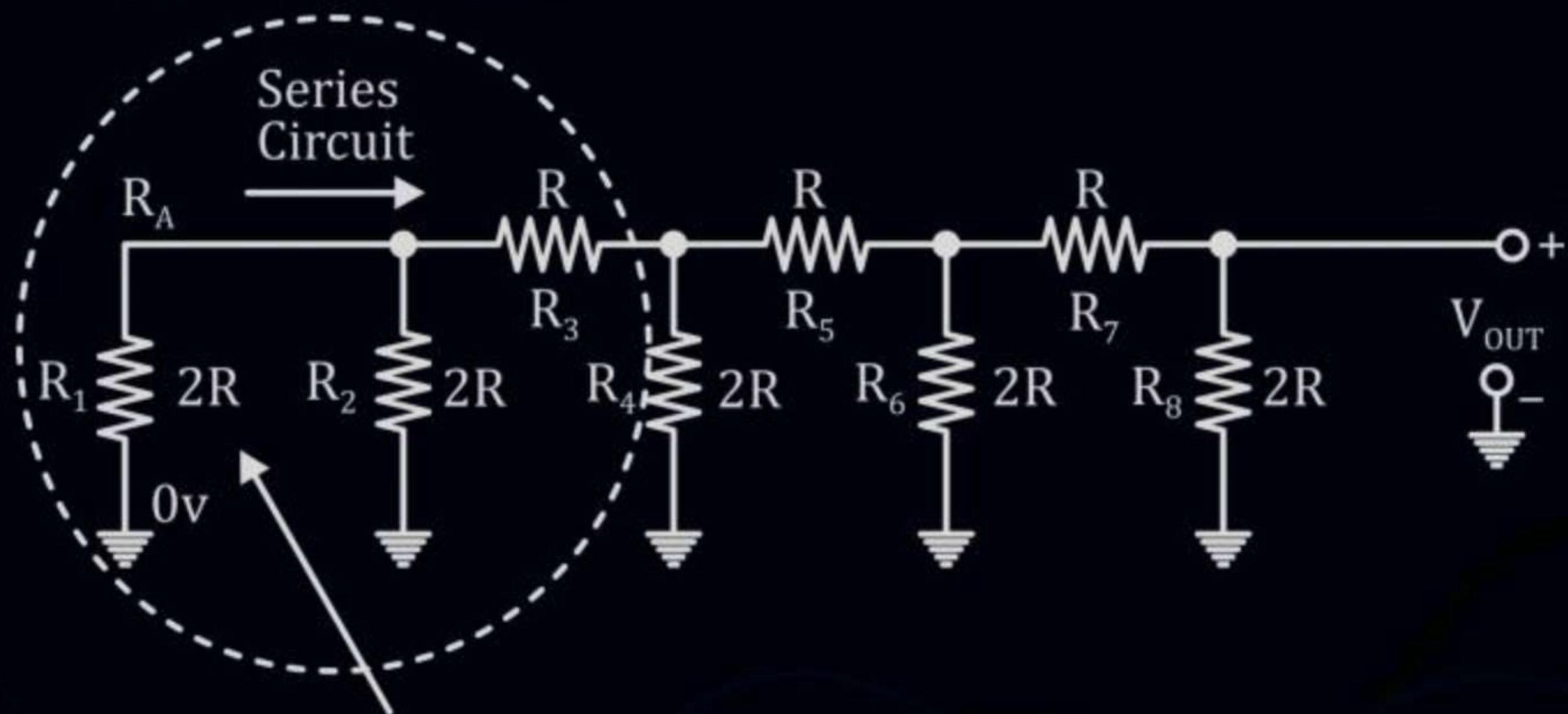
## Weighted Resistor and R-2R type DAC

R - 2R Ladder type DAC



## Weighted Resistor and R-2R type DAC

### R - 2R Ladder type DAC



Parallel Circuit

$$V_0 = R \times D \times G$$

$$V_0 = \frac{V_r}{2^n} \times \sum_{i=0}^{n-1} 2^i b_i \times G$$

DAC  
L

$$V_o = R \times D \times G$$

$$R = \frac{V_r}{2^n - 1}$$

DAC

$$V_o = R \times D \times G$$

$$R = \frac{V_r}{2^n - 1}$$

Weighted  
Resistor

$$V_o = R \times D \times G$$

$$R = \frac{V_r}{2^{n-1}}$$

R-2R

$$V_o = R \times D \times G$$

$$R = \frac{V_r}{2^n}$$



**THANK YOU**

