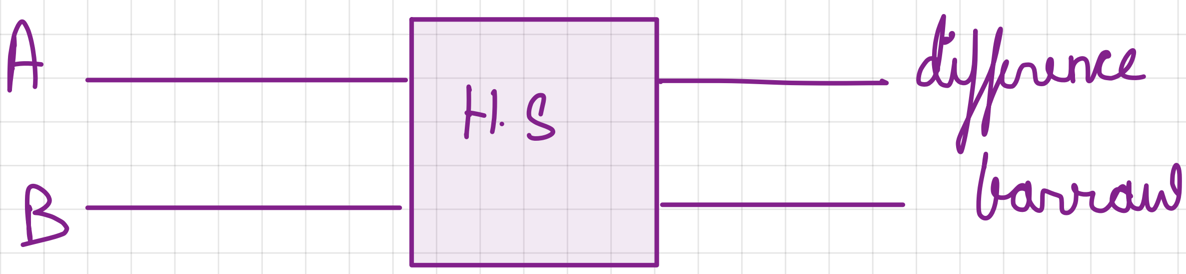
ADITYA KUNDU

08/Sep/2022

Half Subtractor :-

1. It is a combinational circuit
2. It performs the arithmetic subtraction of two binary digits
3. It produces two output (i) difference and (ii) barrow.

Block diagram :-



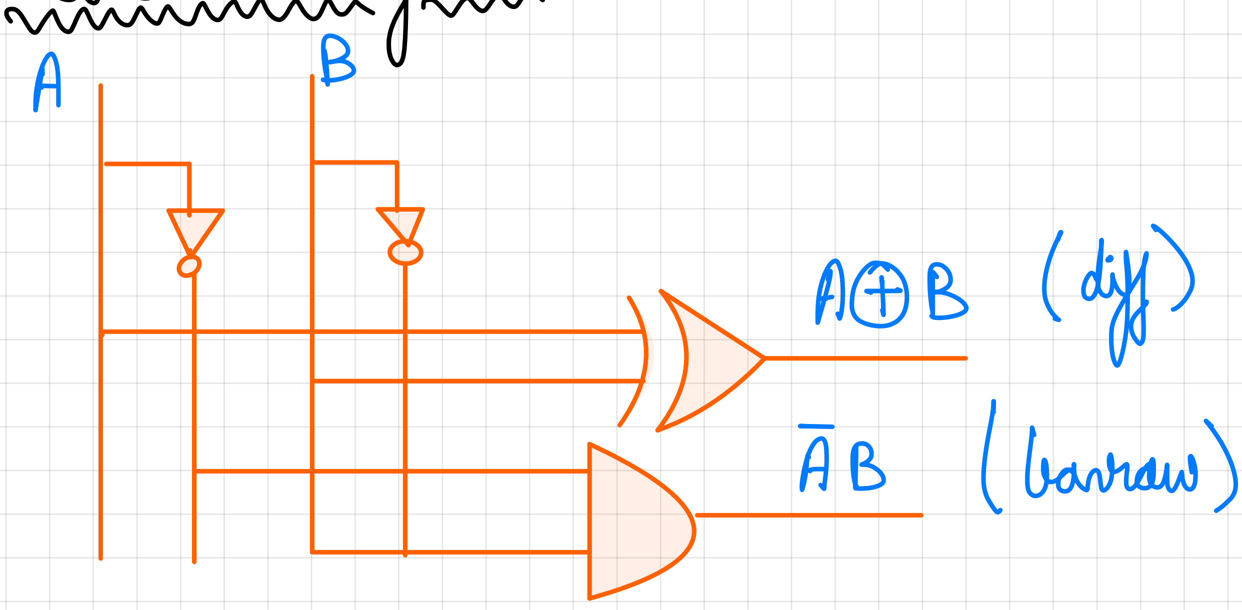
Truth Table :-

A	B	diff	barrow	minterm
0	0	0	0	$\bar{A}\bar{B}$
0	1	1	1	$\bar{A}B$
1	0	1	0	$A\bar{B}$
1	1	0	0	AB

$$\left.\begin{aligned} \text{difference} &= (\bar{A}B) + (A\bar{B}) \\ \text{barrow} &= \bar{A}B \end{aligned}\right\} \text{from truth table}$$

$A \oplus B$

Circuit diagram :-



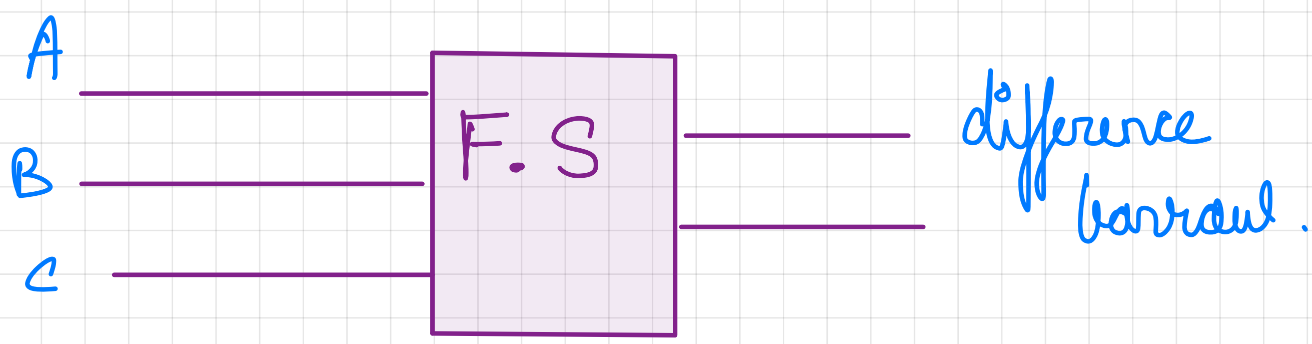
Full Subtractor :-

(i) It is a combinational circuit.

(ii) It performs arithmetic subtraction of three binary digits.

(iii) It also produces two outputs (1) difference and (2) borrow.

Block diagram :-



Truth Table :-

A	B	C	diff	borrow	minterm	
0	0	0	0	0	$\bar{A}\bar{B}\bar{C}$	
0	0	1	1	1	$\bar{A}\bar{B}C$	
0	1	0	1	1	$\bar{A}B\bar{C}$	
0	1	1	0	1	$\bar{A}BC$	$(0-1)-1$
1	0	0	1	0	$A\bar{B}\bar{C}$	
1	0	1	0	0	$A\bar{B}C$	$(1-0)-1$
1	1	0	0	0	$AB\bar{C}$	
1	1	1	1	1	ABC	

$$\text{difference} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

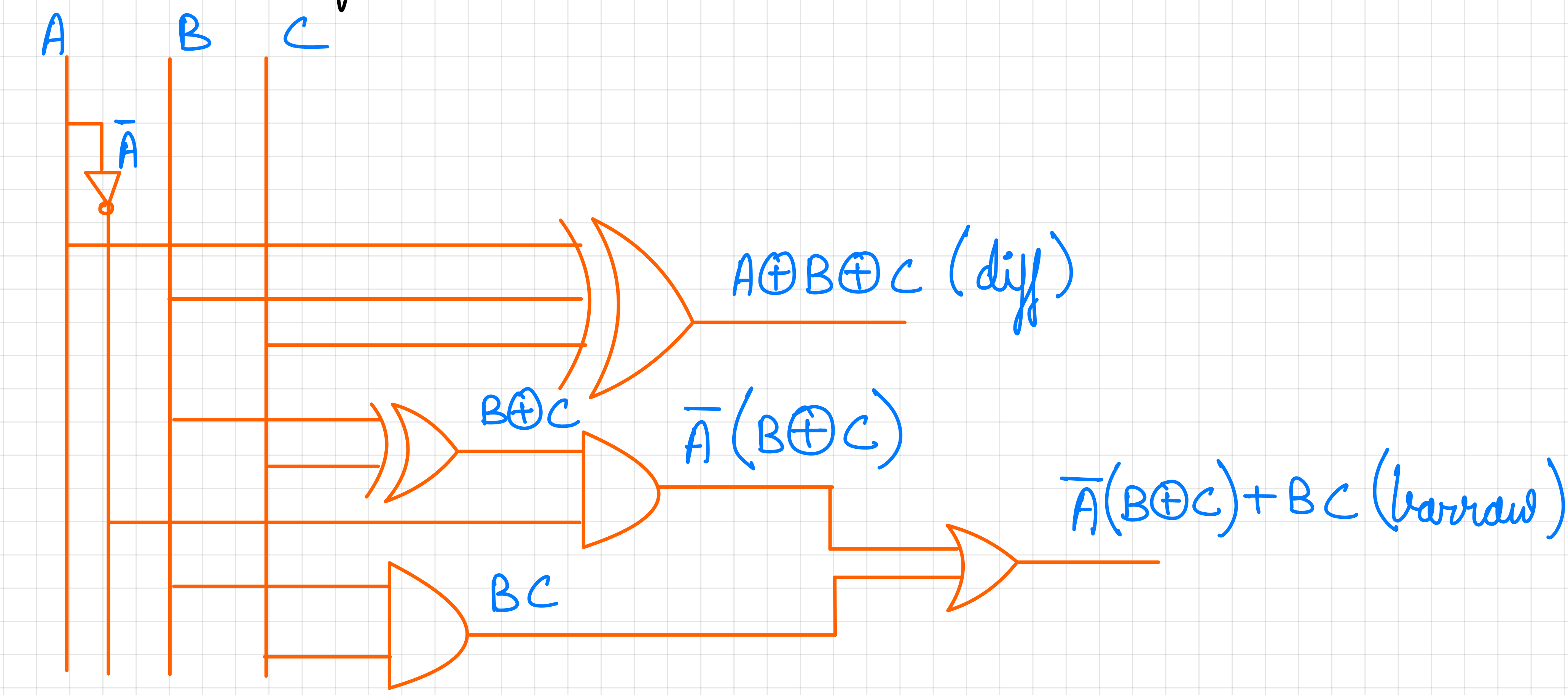
$$= \left[\bar{A}(\bar{B}C + B\bar{C})\right] + \left[A(\bar{B}\bar{C} + BC)\right]$$

$$= \left[\bar{A}(B \oplus C)\right] + \left[A(\overline{B \oplus C})\right]$$

$$= \boxed{A \oplus B \oplus C}$$

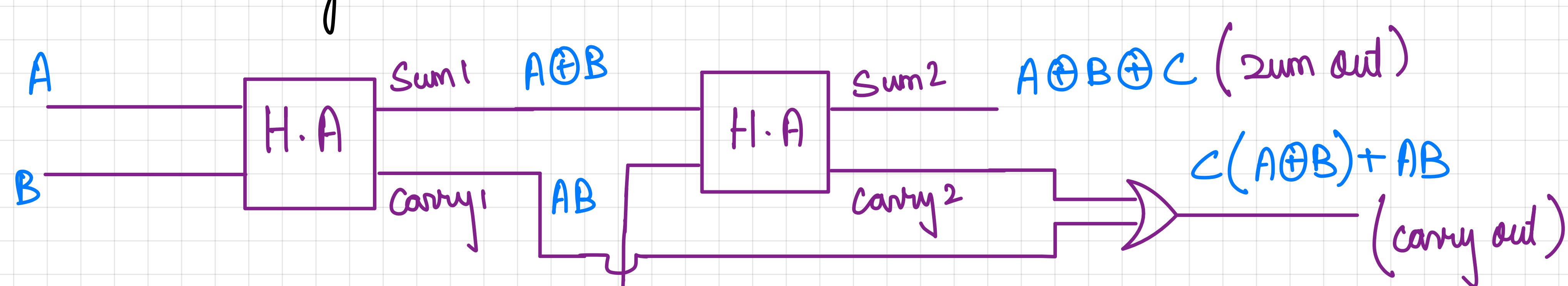
$$\text{Borrow} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + BC(\bar{A}+A) \quad [\because \bar{A}+A=1]$$
$$= \bar{A}(B \oplus C) + BC$$

Circuit diagram :-

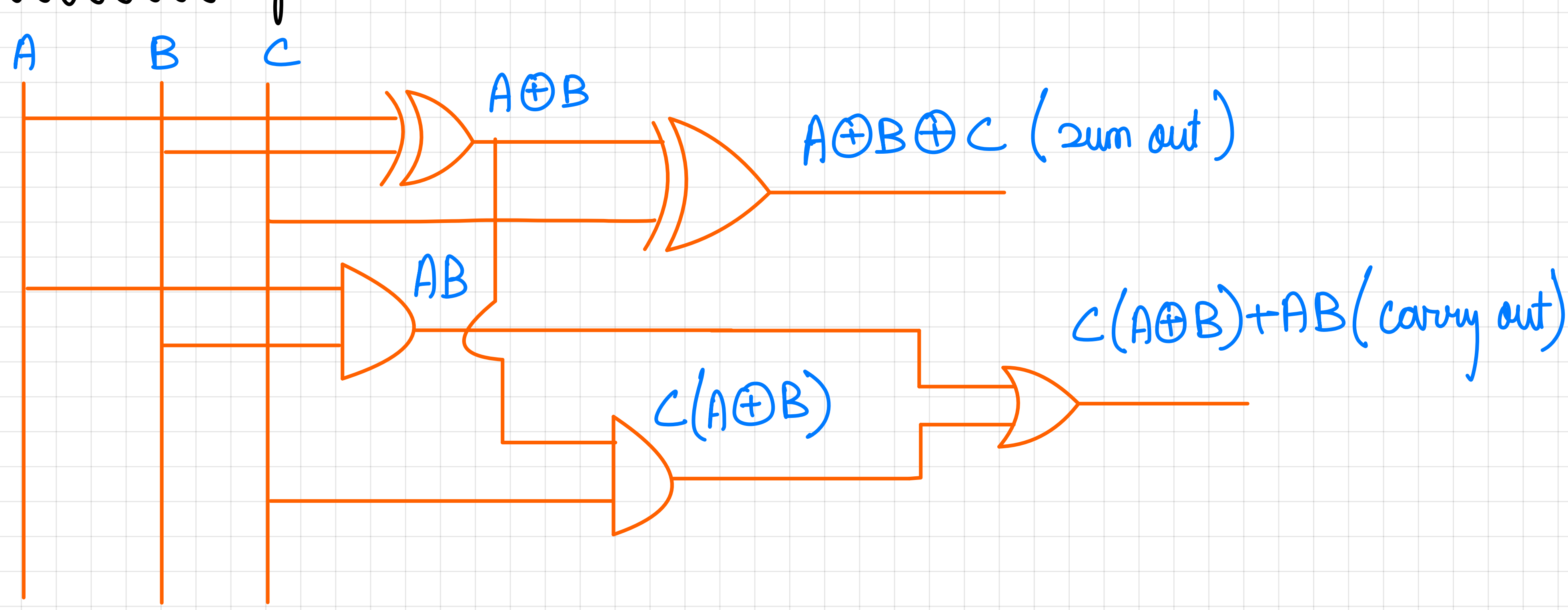


Full Adder Using Half Adder :-

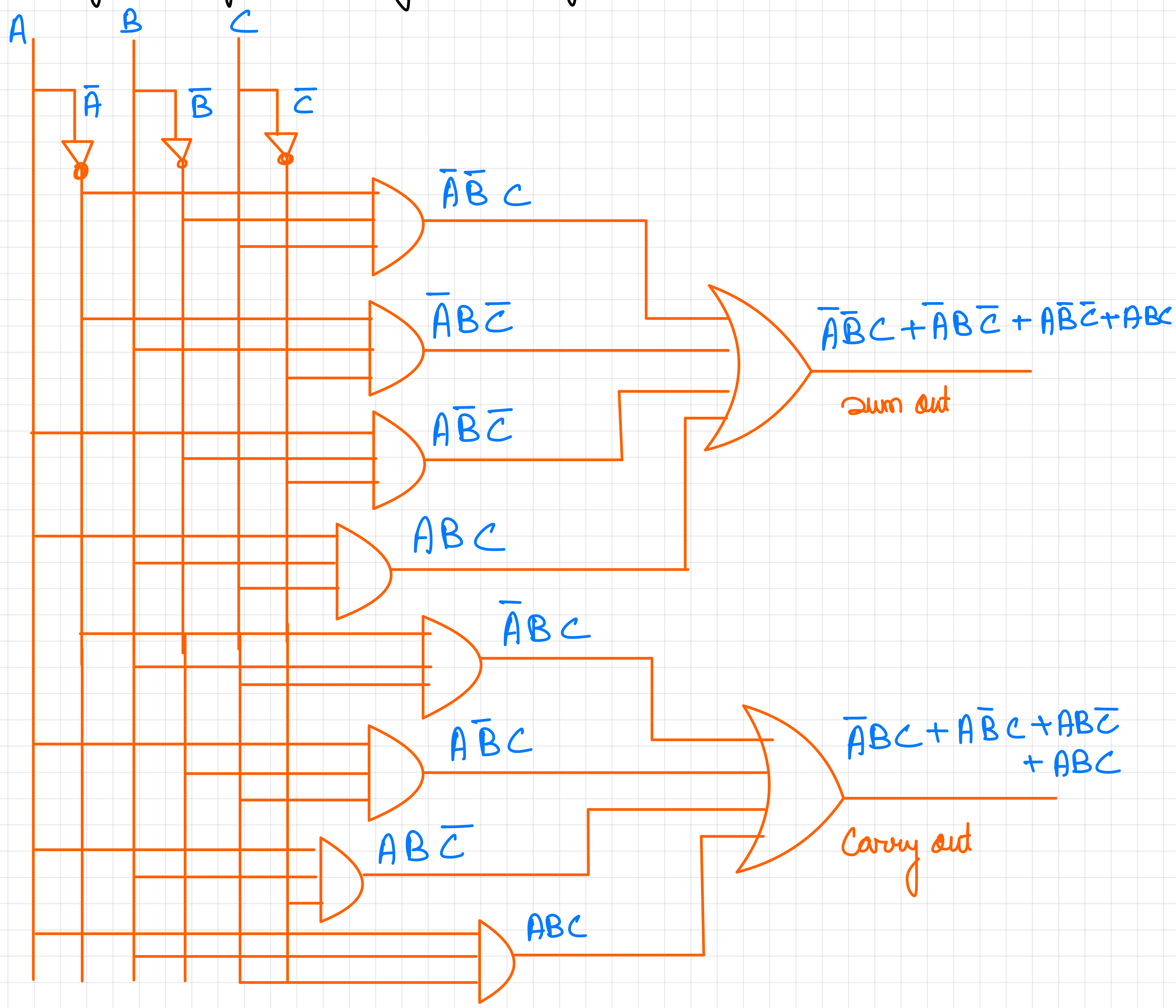
Block diagram :-



Circuit diagram :-



Logic diagram Using Basic gates :-



Karnaugh Map (K-Map)

15/Sep/2022

The Karnaugh map technique provide a systematic method for simplifying and manipulating switching expressions. In this technique the information contained in a truth table or available in the POS form or SOP form is represented on the K-Map.

$$Y = \sum m(0,3)$$

Important!

Variable values in K-Map follow gray code only! Not Binary

Step 1 : $2^n = 3 \simeq 4$

$n = 2$

Step-2 :

A \ B	0	1
0	0: 1 $\bar{A}\bar{B}$	1: $\bar{A}B$ 0
1	2: $A\bar{B}$ 0	3: AB 1

group 1

group 2

A	B	
0	0	$\rightarrow$ 0
0	1	$\rightarrow$ 1
1	0	$\rightarrow$ 2
1	1	$\rightarrow$ 3

Step-3 : Group Making

① The groups will be made with max no. of 1s (SOP), 0s (POS)

Remember

② The total no of 1s within the group must be the power of 2.

Step-4 :

	A	B	
group 1	0	0	$\rightarrow \bar{A}\bar{B}$
group 2	0	1	$\rightarrow AB$

$Y = \bar{A}\bar{B} + AB$

Q. $Y = \sum m(2, 3, 4, 5)$

Soln: Step-① : $2^n = 5 \simeq 8$

$n = 3$

Step-② :

A \ BC	00	01	11	10
0	0 (000): 0 $\bar{A}\bar{B}\bar{C}$	1 (001): 0 $\bar{A}\bar{B}C$	3 (011): 1 $\bar{A}BC$	2 (010): 1 $\bar{A}B\bar{C}$
1	4 (100): 1 $A\bar{B}\bar{C}$	5 (101): 1 $A\bar{B}C$	7 (111): 0 ABC	6 (110): 0 $AB\bar{C}$

Step-③ : Grouping.

Step-④

	A	B	C	
group 1 $\rightarrow$	0	1	1 / 0	$\rightarrow \bar{A}B$
group 2 $\rightarrow$	1	0	0 / 1	$\rightarrow A\bar{B}$

$Y = \bar{A}B + A\bar{B}$

Don't take the changing values, take the fixed values only.

Remember

Q.1. $Y = \sum m(0,2,3)$

Q.2. $Y = \sum m(0,1,4,5,7)$

Multiplexer :-

16/Sep/2022

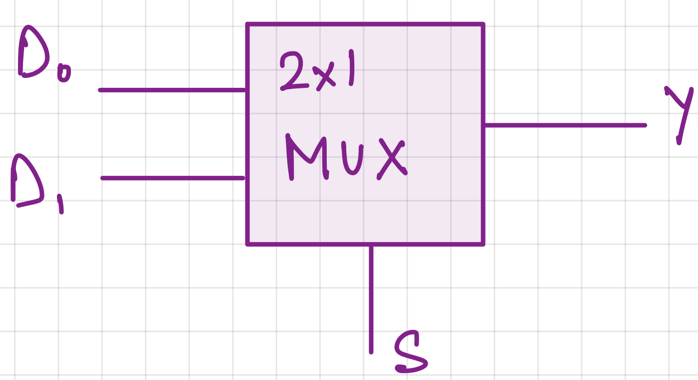
→ It is a combinational circuit.

→ It select one digital information from several source and transmits the selective information on a single output line.

(Many to One) → Multiple input source and one output source.

• 2×1 MUX :-

Block diagram :



$2^m = n$

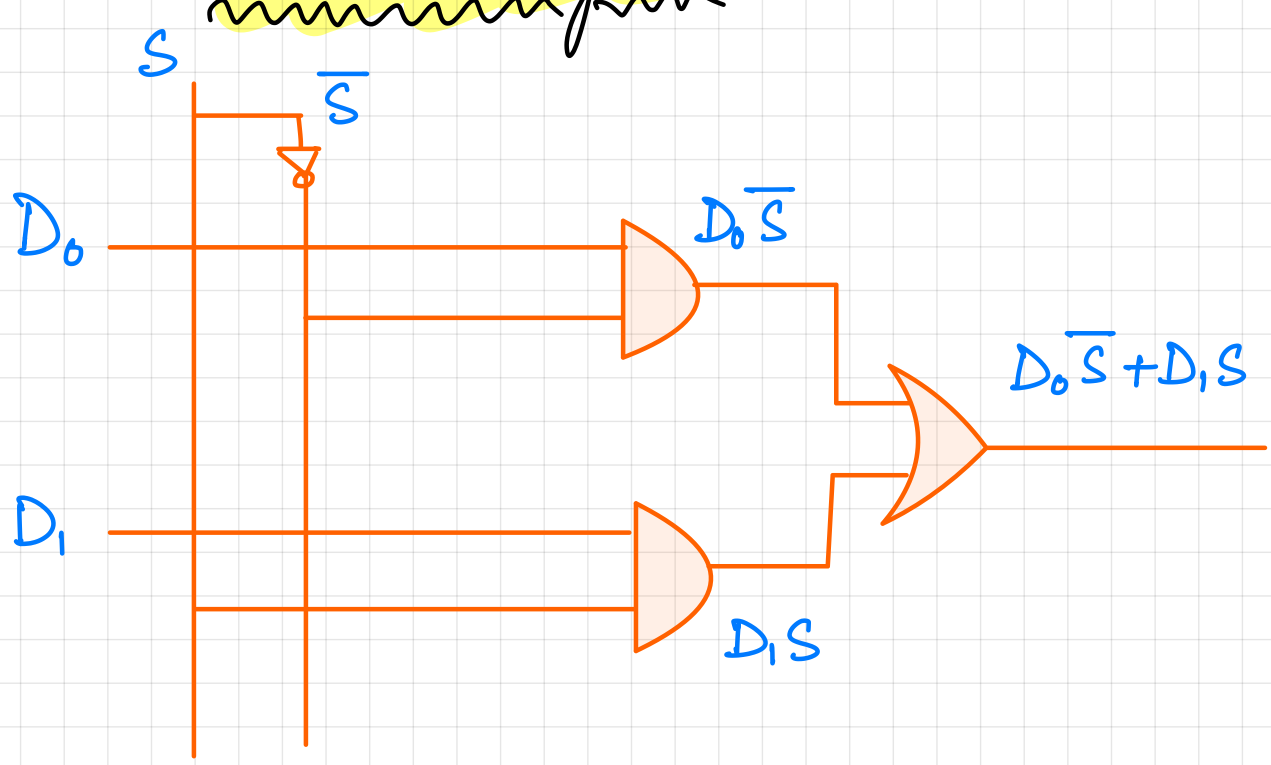
{ n = no of inputs }

Truth Table

S	Y
0	D_0
1	D_1

$Y = D_0\overline{S} + D_1 S$

• Circuit diagram :-



- # 8×1 MUX :-

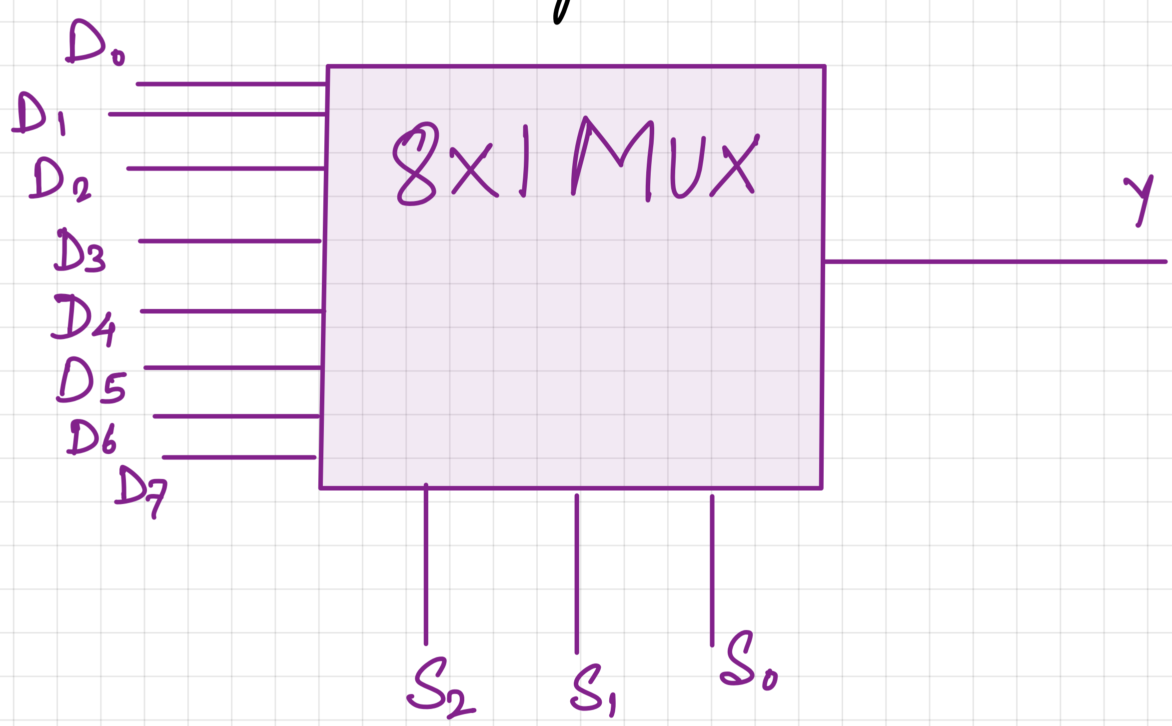
16/Sep/2022

$2^m = n$

$\Rightarrow 2^m = 8$

$\Rightarrow m = 3$

Block diagram



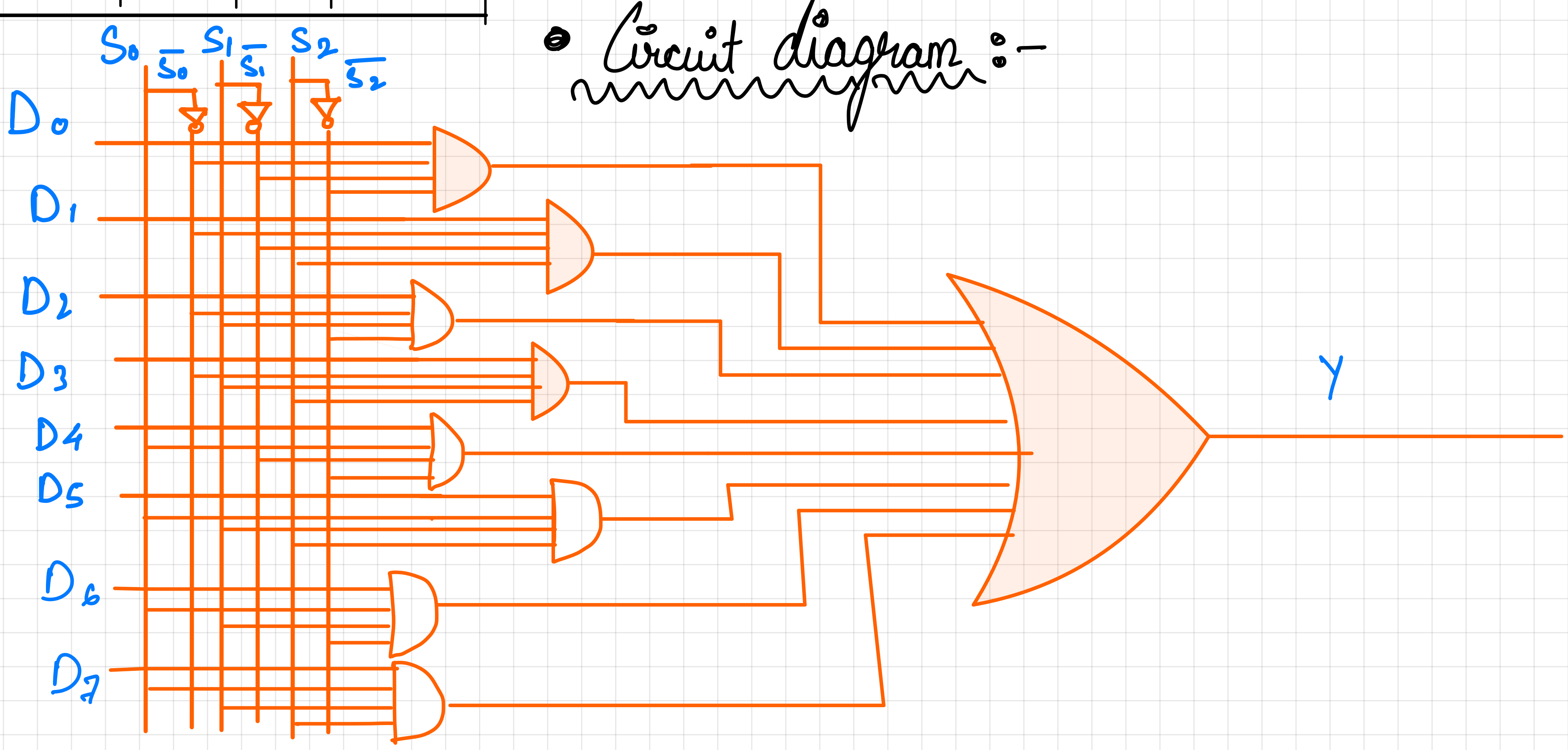
- ## Truth Table

S_0	S_1	S_2	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

- ## Logical expression

$$Y = D_0 \overline{S_0}\,\overline{S_1}\,\overline{S_2} + D_1 \overline{S_0}\,\overline{S_1}\, S_2 +$$
$$D_2 \overline{S_0}\, S_1 \overline{S_2} + D_3 \overline{S_0}\, S_1 S_2 + D_4 S_0 \overline{S_1}\,\overline{S_2}$$
$$+ D_5 S_0 \overline{S_1}\, S_2 + D_6 S_0 S_1 \overline{S_2} + D_7 S_0 S_1 S_2$$

- ## Circuit diagram :-



Q. $Y = \sum m(0, 2, 3)$

Soln: Step-① : $2^n = 3 \simeq 4$

$n = 2$

Step-② :

A \ B	0	1
0	0: 00, $\bar{A}\bar{B}$ — 1	1: 01, $\bar{A}B$ — 0
1	2: 10, $A\bar{B}$ — 1	3: 11, AB — 1

group 2

group 1

Step-③ : Grouping done in K-Map.

Step-④ :

	A	B	
group 1 →	1 1	0 1	→ A
group 2 →	0 1	0 0	→ $\bar{B}$

$\therefore Y = A + \bar{B}$

Q. $Y = \sum m(0, 1, 4, 5, 7)$

Soln: Step-① : $2^n = 7 \simeq 8$

$n = 3$

Step-② :

A \ BC	00	01	11	10
0	0: 000, $\bar{A}\bar{B}\bar{C}$ — 1	1: 001, $\bar{A}\bar{B}C$ — 1	3: 011, $\bar{A}BC$ — 0	2: 010, $\bar{A}B\bar{C}$ — 0
1	4: 100, $A\bar{B}\bar{C}$ — 1	5: 101, $A\bar{B}C$ — 1	7: 111, ABC — 1	6: 110, $AB\bar{C}$ — 0

group 1

group 2

Step-③ : Grouping done in K-Map

Step-④ :

	A	B	C	
group 1 →	0 0 1 1	0 0 0 0	0 1 0 1	→ $\bar{B}$
group 2 →	1 1	0 1	1 1	→ AC

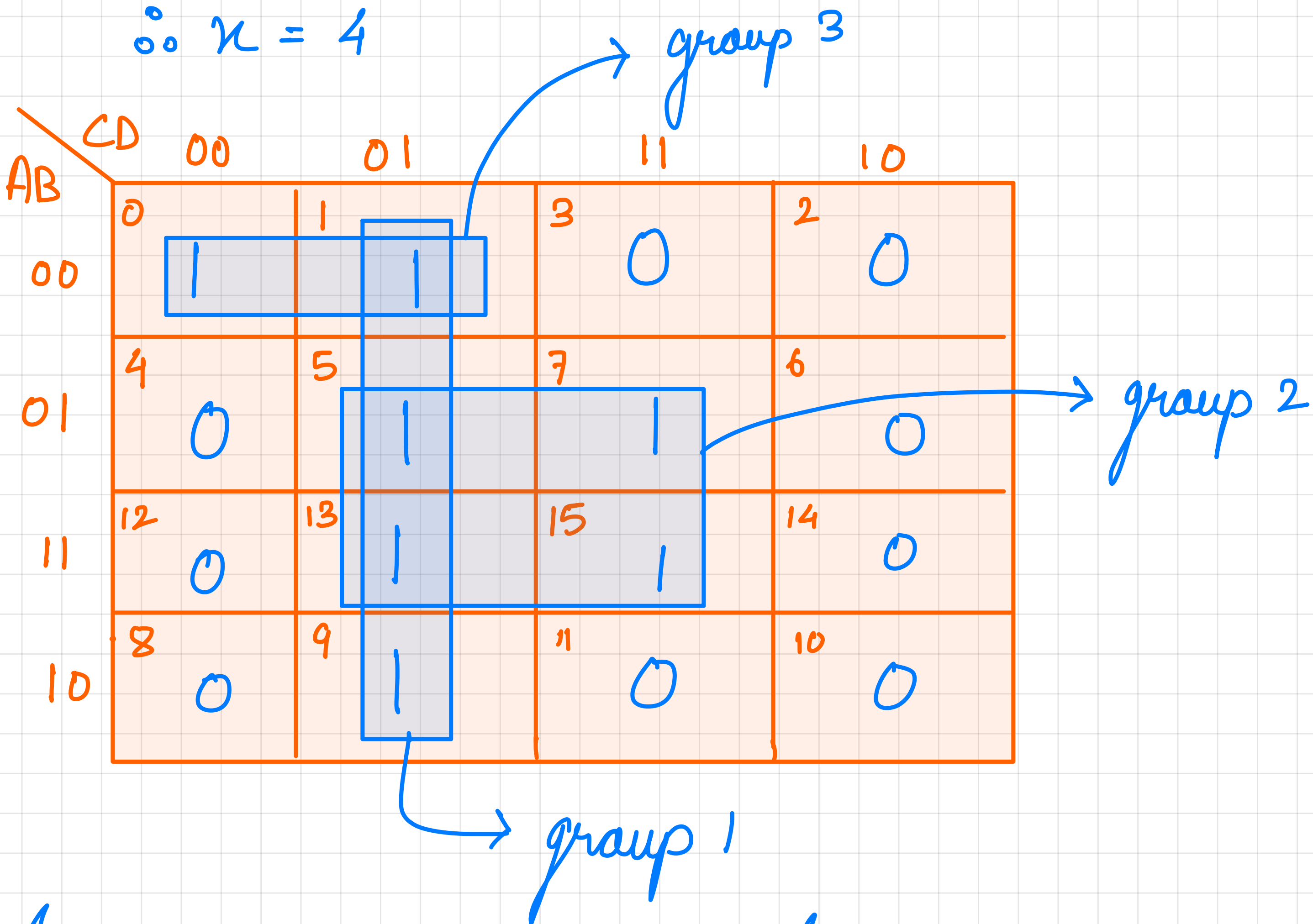
$\therefore Y = \bar{B} + AC$

Q. $Y = \sum m(0, 1, 5, 7, 9, 13, 15)$

Soln: Step ①: $2^n = 15 \simeq 16$

$\therefore n = 4$

Step ②:



Step ③: Grouping as done in diagram.

Step-④:

	A	B	C	D	
Group 1 →	0 0 1 1	0 1 1 0	0 0 0 0	1 1 1 1	→ $\bar{C}D$
Group 2 →	0 0 1 1	1 1 1 1	0 1 0 1	1 1 1 1	→ BD
group 3 →	0 0	0 0	0 0	0 1	→ $\bar{A}\bar{B}\bar{C}$

$\therefore Y = \bar{C}D + BD + \bar{A}\bar{B}\bar{C}$ Ans

Q. $Y = \sum m(0, 2, 8, 10)$

Soln: Step ①: $2^n = 10 \simeq 16$

$\therefore n = 4$

Step-②:

AB \ CD	00	01	11	10
00	(0) 1	(1) 0	(3) 0	(2) 1
01	(4) 0	(5) 0	(7) 0	(6) 0
11	(12) 0	(13) 0	(15) 0	(14) 0
10	(8) 1	(9) 0	(11) 0	(10) 1

Step-③: Corner grouping in K-Map. (of 4 elements)

Step-④:

	A	B	C	D
group 1 →	0	0	0	0
	0	0	1	0
	1	0	0	0
	1	0	1	0

$\longrightarrow \bar{B}\bar{D}$

$\therefore Y = \bar{B}\bar{D}$

TO DO

Q.① $Y = \sum m(1, 2, 5, 7, 9, 13, 15)$

Q.② $Y = \sum m(1, 3, 5, 7, 9, 11, 13)$

Q.③ $Y = \sum m(1, 3, 5, 7, 9, 11, 13, 15)$

Q.④ $Y = \sum m(0, 1, 2, 3, 4, 5, 8, 9, 10, 11)$

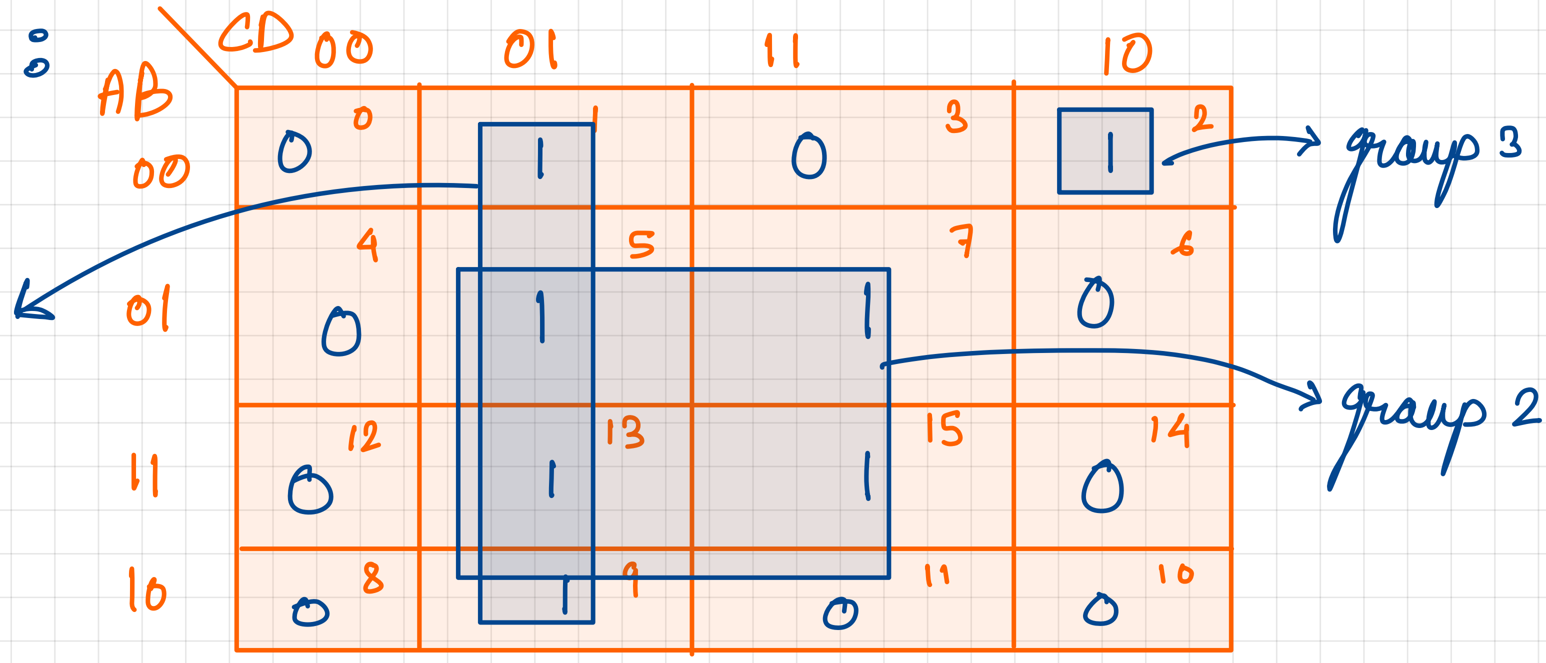
} K-Mapping

Soln ① Step ①

$2^n = 15 \simeq 16$

$\therefore n = 4$

Step-② :



Step-③ : Grouping in KMap.

Step-④ :

	A	B	C	D	
group → ①	0	0	0	1	
	0	1	0	1	→ $\bar{C}D$
	1	1	0	1	
	1	0	0	1	
group → ②	0	1	0	1	
	0	1	1	1	→ BD
	1	1	0	1	
	1	1	1	1	
group-③	0	0	1	0	→ $\bar{A}\bar{B}C\bar{D}$

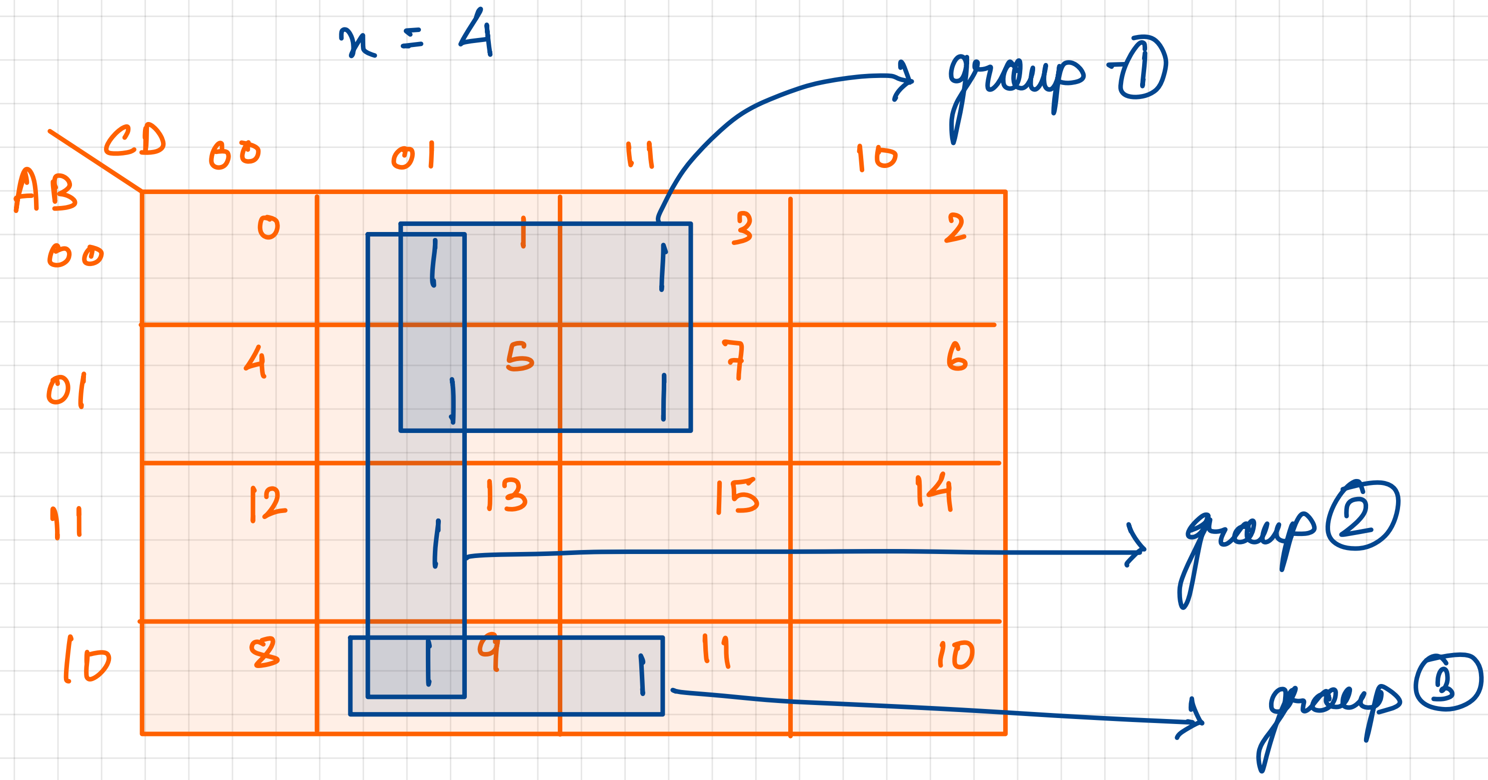
$\therefore Y = \bar{C}D + BD + \bar{A}\bar{B}C\bar{D}$ Ans

---

② **Step-①** : $2^x = 13 \simeq 16$

$x = 4$

Step-② :



Step ③ : Grouping in K-Map.

Step-④

	A	B	C	D	
group ① →	0	0	0	1	
	0	0	1	1	→ $\bar{A}D$
	0	1	0	1	
	0	1	1	1	
group ② →	0	0	0	1	
	0	1	0	1	→ $\bar{C}D$
	1	1	0	1	
	1	0	0	1	
group ③ →	1	0	0	1	→ $A\bar{B}D$
	1	0	1	1	

$\therefore Y = \bar{A}D + \bar{C}D + A\bar{B}D$ Ans

---

③ **Step-①:** $2^x = 15 \simeq 16$

$x = 4$

Step-②:

AB \ CD	00	01	11	10
00	0	1: 1	3: 1	2
01	4	5: 1	7: 1	6
11	12	13: 1	15: 1	14
10	8	9: 1	11: 1	10

→ group-① of 8 1s

Step-③: Grouping on K-Map

Step-④:

A	B	C	D
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	1	0	1
1	1	1	1
1	0	0	1
1	0	1	1

→ D

$\therefore \boxed{Y = D}$

④ **Step-①** : $2^n = 11 \simeq 16$

$x = 4$

Step-② :

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	1 (3)	1 (2)
01	1 (4)	1 (5)	(7)	(6)
11	(12)	(13)	(15)	(14)
10	1 (8)	1 (9)	1 (11)	1 (10)

group ②

group ①

Step-③ : Grouping on K-Map.

Step-④ :

	A	B	C	D	
group ① →	0	0	0	0	
	0	0	0	1	
	0	0	1	1	→ $\bar{B}$
	0	0	1	0	
	1	0	0	0	
	1	0	0	1	
	1	0	1	1	
	1	0	1	0	
group ② →	0	0	0	0	
	0	0	0	1	→ $\bar{A}\bar{C}$
	0	1	0	0	
	0	1	0	1	

$\therefore \boxed{Y = \bar{A}\bar{C} + \bar{B}}$

Q. $Y = \sum m(0,1,2,4,5,6,7,8)$

Soln : **Step-①** : $2^n = 8 \simeq 16$

$\therefore x = 4$

Step-② :

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	(3)	(2)
01	1 (4)	1 (5)	1 (7)	1 (6)
11	(12)	(13)	(15)	(14)
10	1 (8)	(9)	(11)	(10)

group 1

group 2

group 3

Step 3 : Grouping

Step 4 :

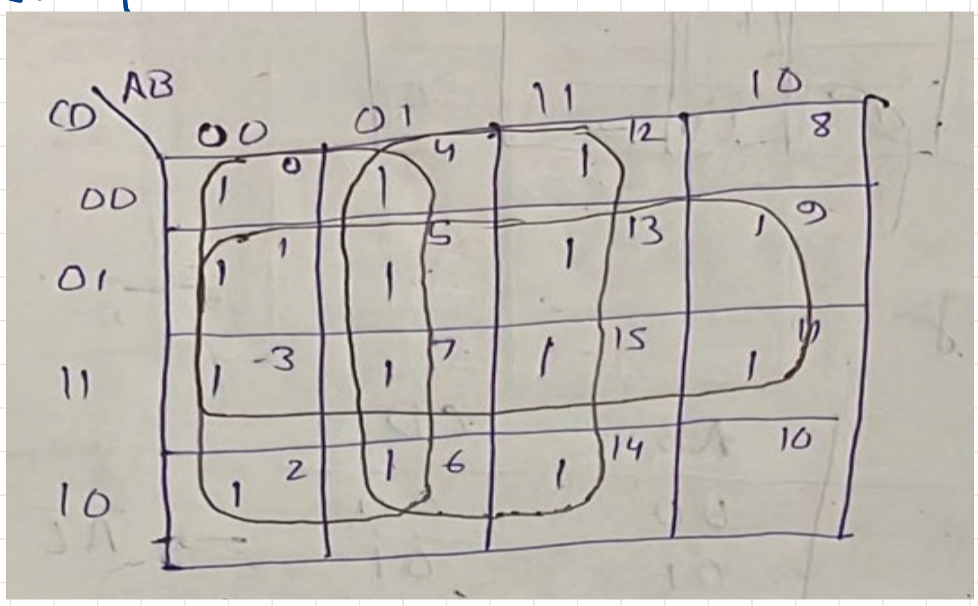
	A	B	C	D	
G_1 →	0	0	0	0	
	0	0	0	1	→ $\bar{A}\bar{C}$
	0	1	0	0	
	0	1	0	1	
G_2 →	0	1	0	0	
			0	1	→ $\bar{A}B$
			1	1	
			1	0	
G_3 →	0	0	0	0	→ $\bar{B}\bar{C}\bar{D}$
	1	0	0	0	
G_4 →	1	0	0	0	→ $A\bar{B}\bar{C}\bar{D}$

Q. $Y = \sum m(0,1,2,3,4,5,6,7,9,11,12,13,14,15)$

Step ①: $2^n = 15 \simeq 16$

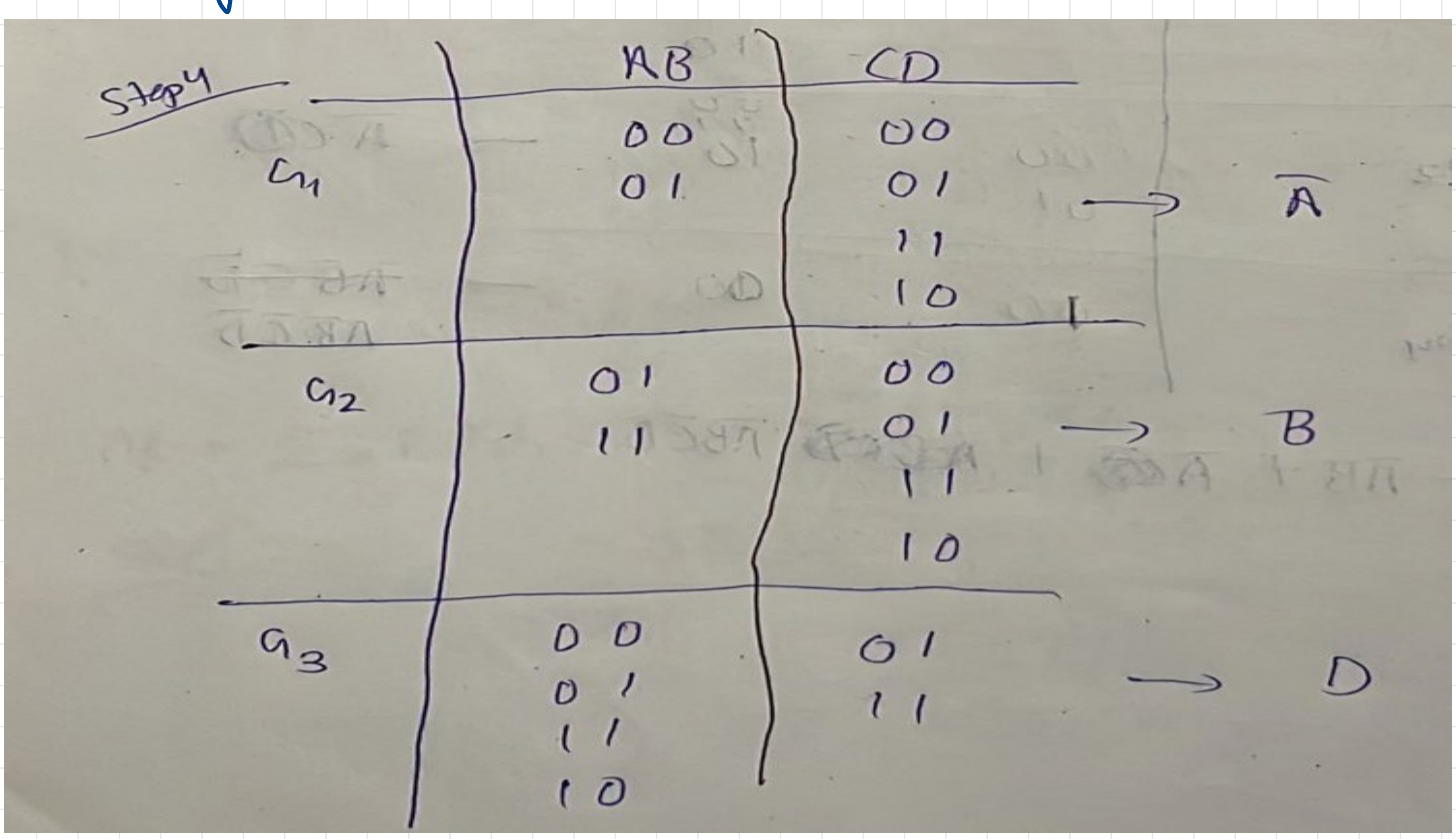
$\therefore n = 4$

Step-②:



Step-③: Grouping on K-Map.

Step-④:



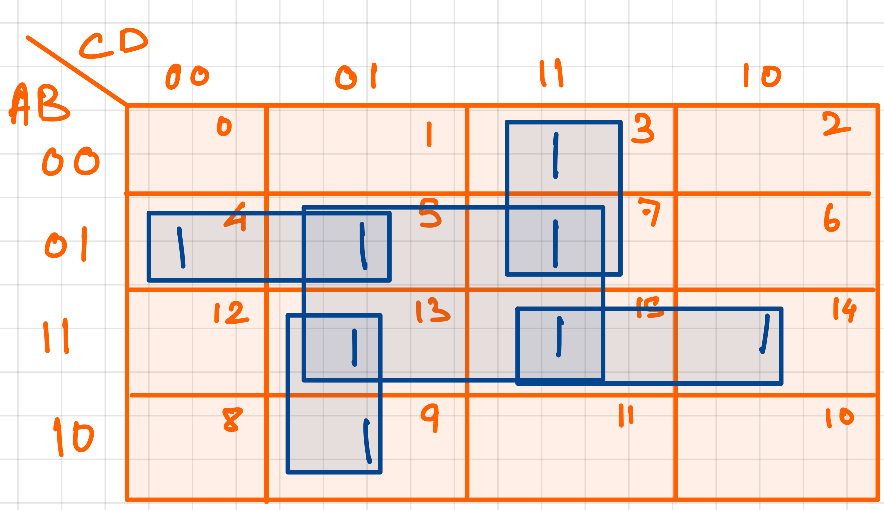
Step 4	AB	CD		
G_1	00, 01	00, 01, 11, 10	→	$\bar{A}$
G_2	01, 11	00, 01, 11, 10	→	B
G_3	00, 01, 11, 10	01, 11	→	D

Q. $Y = \sum m(3,4,5,7,9,13,14,15)$

REMEMBER

Soln: $2^n = 15 \simeq 16 \Rightarrow n = 4$

Step-②:



Special Case

Here all the 4 1s covered by quad group are also covered by pairs. So the quad grouping will be ignored.

Step-3: Grouping

Step-④ :

	A	B	C	D	
$G_1 \rightarrow$	0	0	1	1	$\rightarrow \bar{A}CD$
	0	1	1	1	
$G_2 \rightarrow$	0	1	0	0	$\rightarrow \bar{A}B\bar{C}$
	0	1	0	1	
$G_3 \rightarrow$	1	1	1	1	$\rightarrow ABC$
	1	1	1	0	
$G_4 \rightarrow$	1	1	0	1	$\rightarrow A\bar{C}D$
	1	0	0	1	

$$\boxed{Y = \bar{A}CD + \bar{A}B\bar{C} + ABC + A\bar{C}D}$$

• POS

1) $Y = \pi(3, 4, 6, 7, 11, 12, 13, 14, 15)$

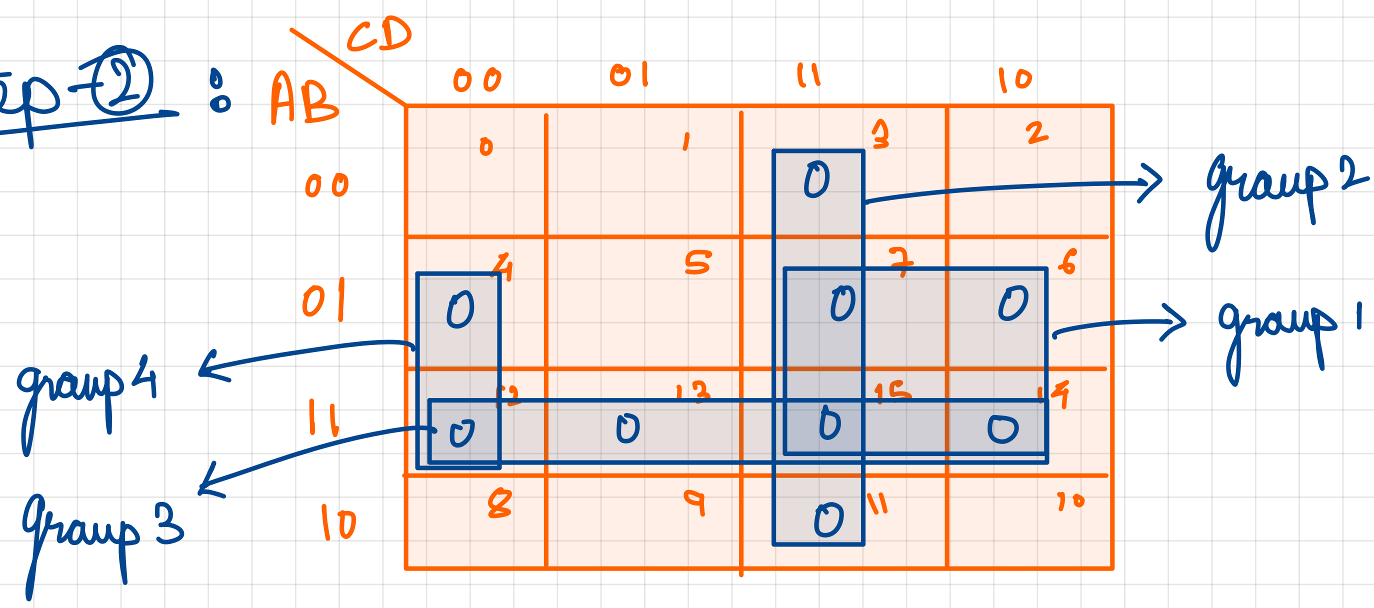
2) $Y = \pi(0, 1, 4, 5, 6, 8, 9, 12, 13, 14)$

24/Sep/2022

Soln ① **Step ① :** $2^n = 15 \simeq 16$

$n = 4$

Step-② :



Step-3 : Grouping on K-Map.

Step-4 :

	A	B	C	D	
$G_1 \rightarrow$	0	1	1	1	$\rightarrow \bar{B} + \bar{C}$
	0	1	1	0	
	1	1	1	1	
	1	1	1	0	
$G_2 \rightarrow$	0	0	1	1	$\rightarrow \bar{C} + \bar{D}$
	0	1	1	1	
$G_3 \rightarrow$	1	1	0	0	$\rightarrow \bar{A} + \bar{B}$
	:	:	:	:	

$G_4 \rightarrow$ 0 1 0 0 / 1 1 0 0 $\rightarrow \bar{B} + C + D$

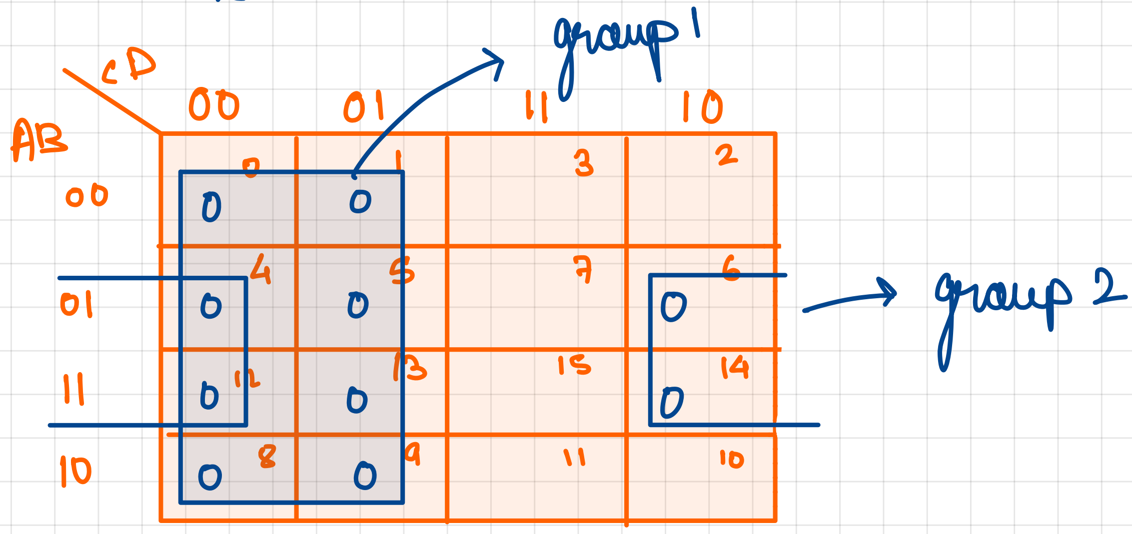
$$\therefore Y = (\bar{B} + \bar{C}).(\bar{C} + \bar{D})(\bar{A} + \bar{B}).(\bar{B} + C + D)$$

Ans

② Step-①: $2^n = 14 \simeq 16$

$n = 4$

Step-②:



Step③: Grouping on K-Map.

Step-④:

	A	B	C	D	
$G_1 \rightarrow$	0	0	0	0	
	0	0	0	1	
	0	1	0	0	$\rightarrow C$
	0	1	0	1	
	1	1	0	0	
	1	1	0	1	
	1	0	0	0	
	1	0	0	1	
	0	1	0	0	
$G_2 \rightarrow$	0	1	1	0	$\rightarrow (\overline{B}+D)$
	1	1	0	0	
	1	1	1	0	

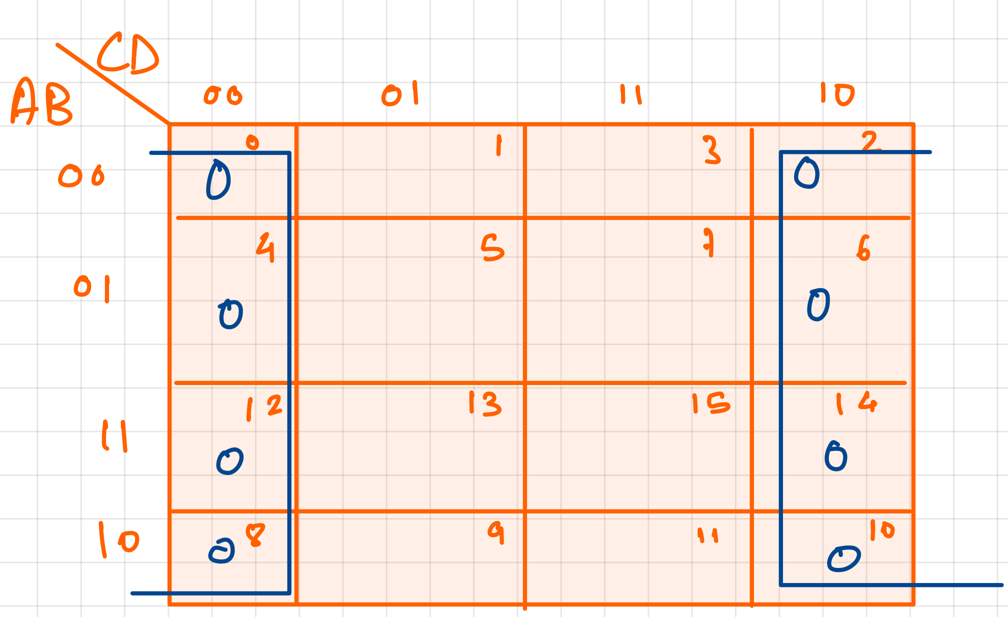
$\therefore \boxed{Y = C \cdot (\overline{B}+D)}$

$Y = \pi(0, 2, 4, 6, 8, 10, 12, 14)$

Soln: Step-①: $2^n = 14 \simeq 16$

$n = 4$

Step-②:



Step③: Grouping

Step-④:

	A	B	C	D
$G_1 \rightarrow$	0	0	0	0
	0	0	1	0
	0	1	0	0
	0	1	1	0
	1	1	0	0
	1	1	1	0
	1	0	0	0
	1	0	1	0

$\Rightarrow D$

$\therefore \boxed{Y = D}$

Don't Care Condition :-

- Usually we are representing the don't care form as 'd' or 'x'.
- Objective : not to take all the don't care form available in the K-Map table.
- We should take the combination of 1s and don't care 'd' (SOP) and the combination of 0s & 'd' in (POS), when it covers maximum number of 1s (SOP) & 0s (POS) in the adjacent cell.
- For SOP when we will take the don't care form along with the 1s to make a group then 'd' will be treated as 1 and for POS 'd' will be treated as 0.

$$Y = \Sigma m(1,3,7,11,15) + \Sigma d(0,2,5)$$

AB \ CD	00	01	11	10
00	d (0)	1 (1)	1 (3)	d (2)
01	(4)	d (5)	1 (7)	(6)
11	(12)	(13)	1 (15)	(14)
10	(8)	(9)	1 (11)	(10)

	A	B	C	D	
$G_1 \rightarrow$	0	0	:	:	$\rightarrow \bar{A}\bar{B}$
$G_2 \rightarrow$	:	:	1	1	$\rightarrow CD$

$\therefore$ $\boxed{Y = \bar{A}\bar{B} + CD}$

$$Y = \pi(0,1,2,3,8,9,11) + \Sigma d(5,10,15)$$

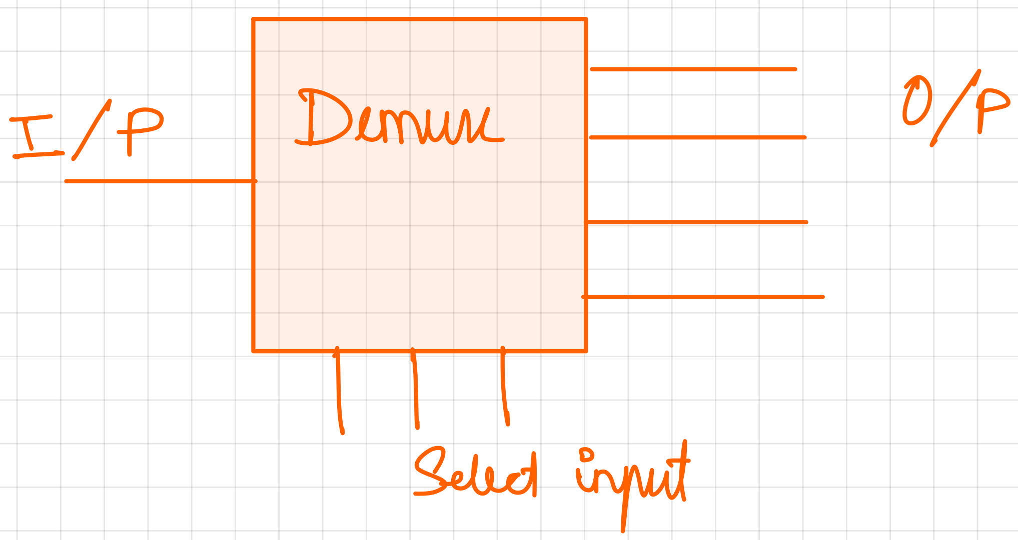
AB \ CD	00	01	11	10
00	0 (0)	0 (1)	0 (3)	0 (2)
01	(4)	d (5)	(7)	(6)
11	(12)	(13)	d (15)	(14)
10	0 (8)	0 (9)	0 (11)	d (10)

	A	B	C	D	
$G_1 \rightarrow$	0	0	0	0	
			:	:	$\rightarrow B$
	1	0	:	:	

$\therefore$ $\boxed{Y = B}$

• De Multiplexer :- (DeMUX)

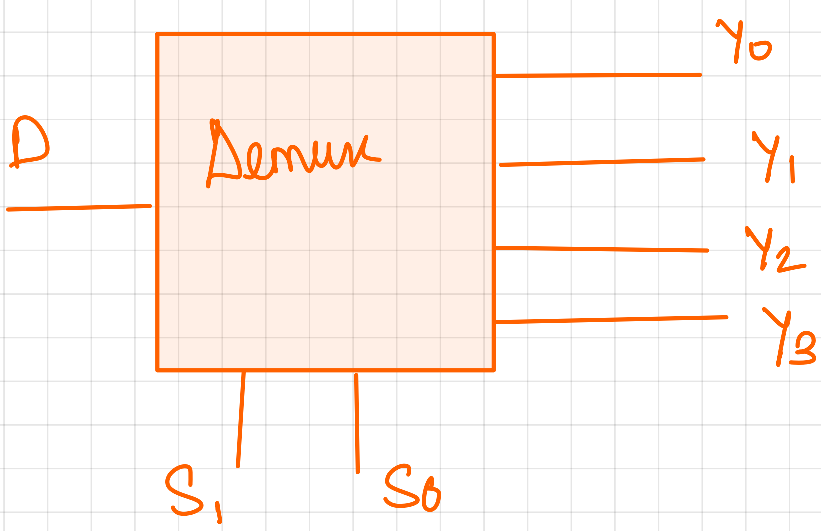
03/Nov/2022



$$\boxed{2^m = n}$$

m = select input

n = Output



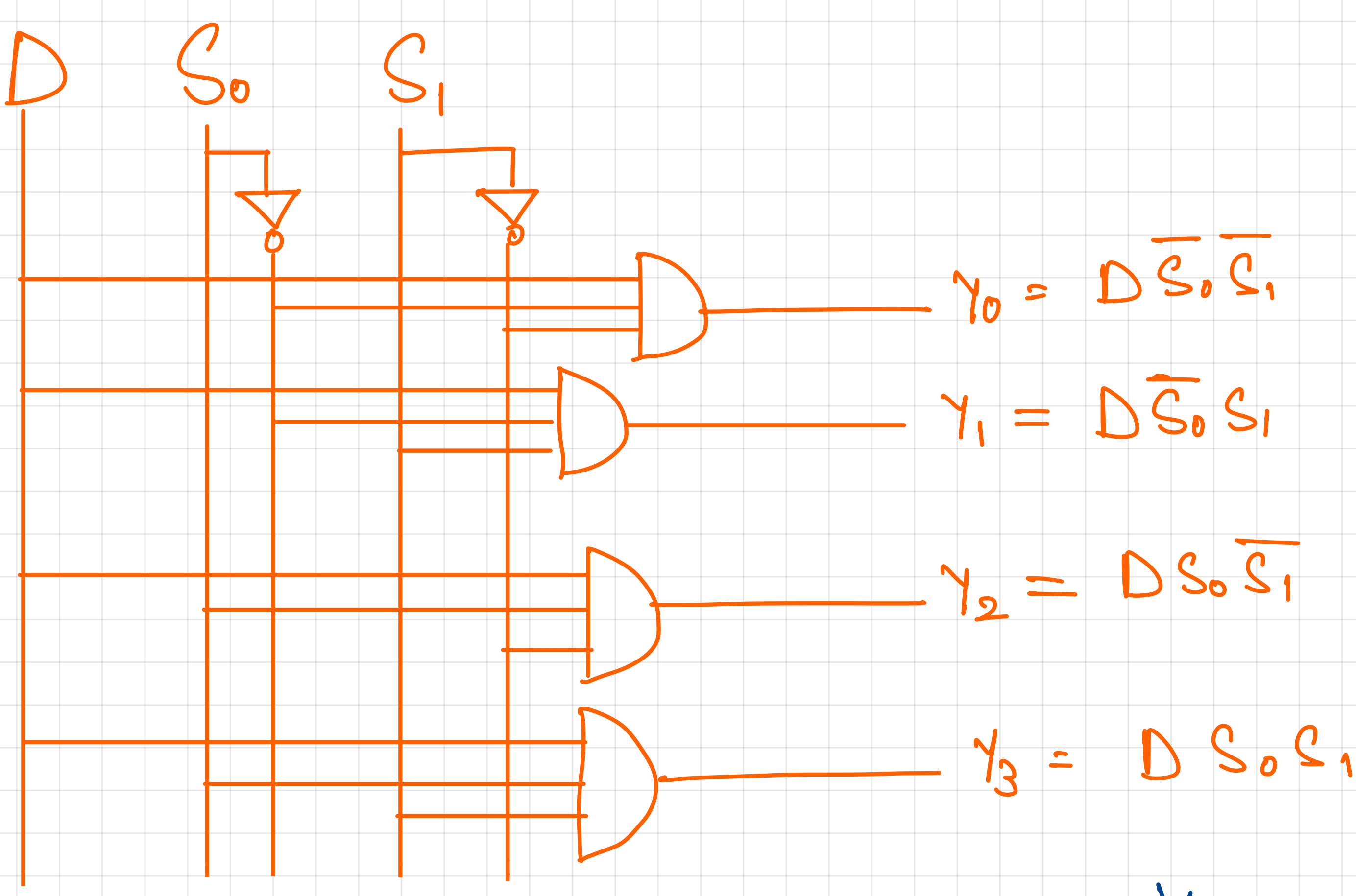
1 : 4

Truth Table

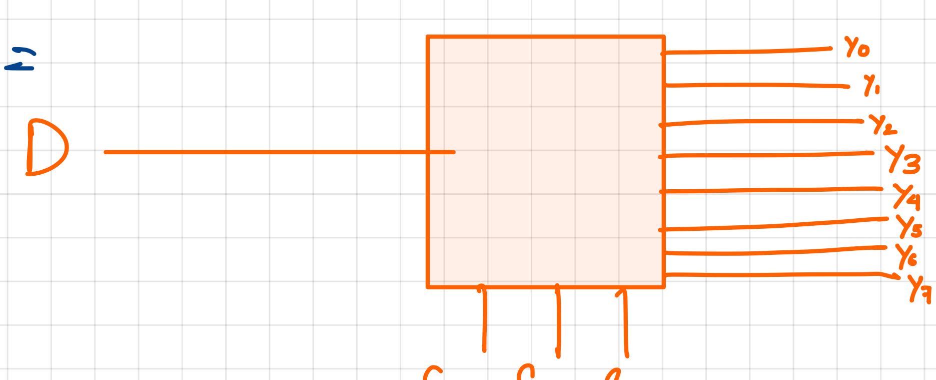
Data I/P (D)	Select I/P (S_0)	(S_1)	O/P Y_0	Y_1	Y_2	Y_3
D	0	0	D	0	0	0
D	0	1	0	D	0	0
D	1	0	0	0	D	0
D	1	1	0	0	0	D

$Y_0 = D\overline{S_0}\,\overline{S_1}$, $Y_2 = D S_0 \overline{S_1}$

$Y_1 = D\overline{S_0} S_1$, $Y_3 = D S_0 S_1$



• 1 : 8 =



$Y_0 = D\overline{S_0}\,\overline{S_1}\,\overline{S_2}$

$Y_1 = D\overline{S_0}\,\overline{S_1} S_2$

$Y_2 = D\overline{S_0} S_1 \overline{S_2}$

$Y_3 = D\overline{S_0} S_1 S_2$

$Y_4 = D S_0 \overline{S_1}\,\overline{S_2}$

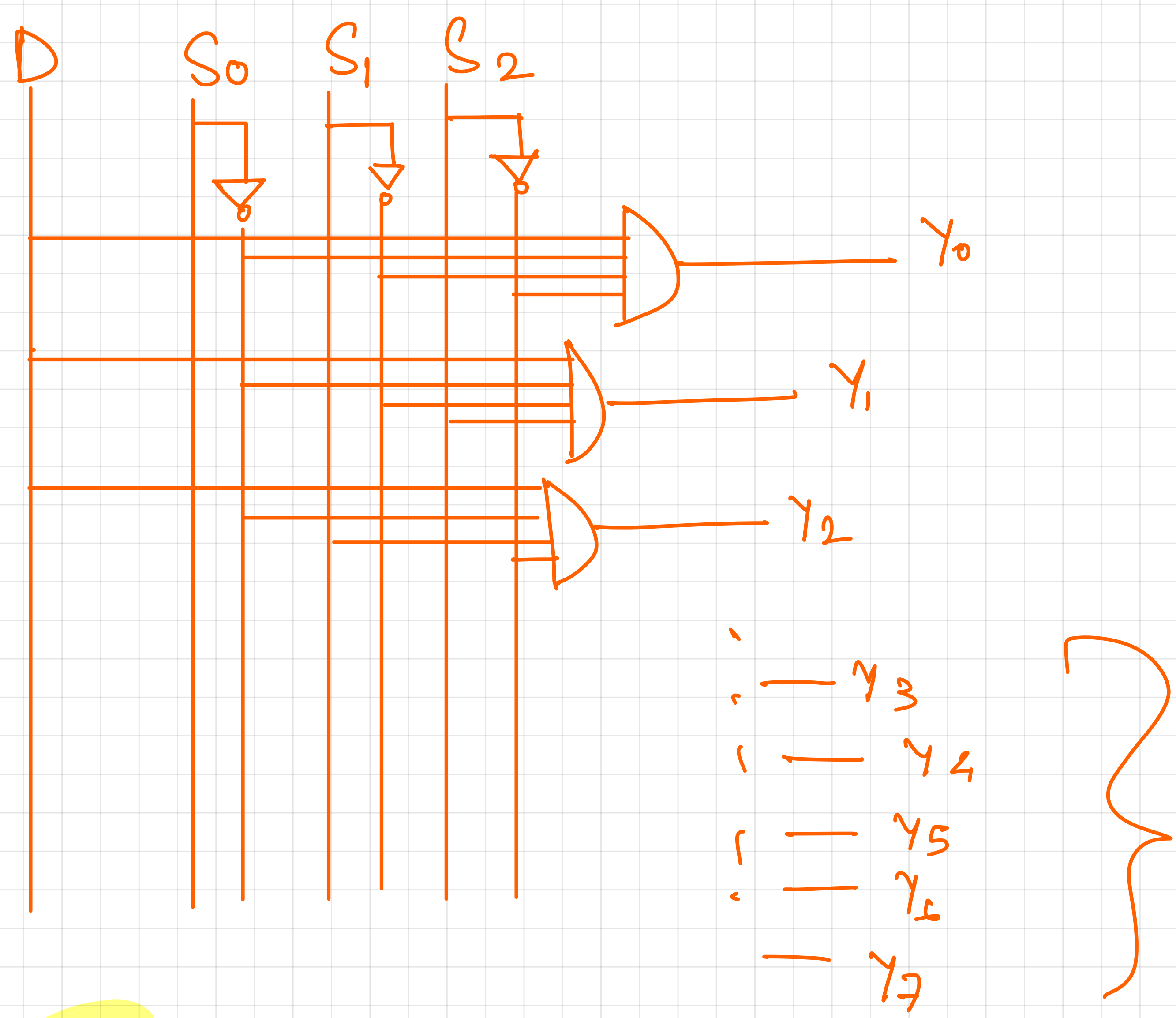
$Y_5 = D S_0 \overline{S_1} S_2$

$Y_6 = D S_0 S_1 \overline{S_2}$

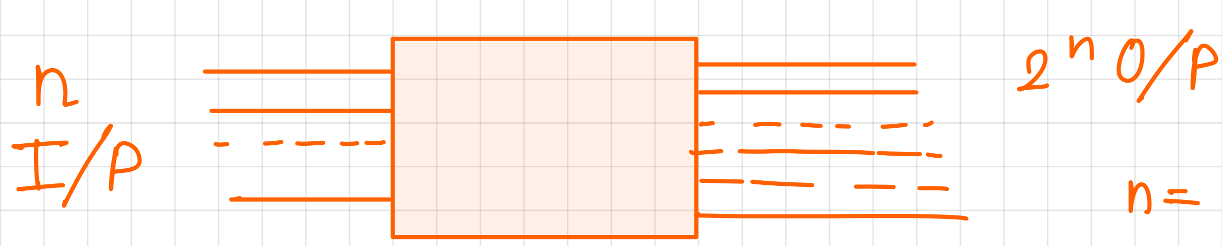
$Y_7 = D S_0 S_1 S_2$

- **Truth Table :-**

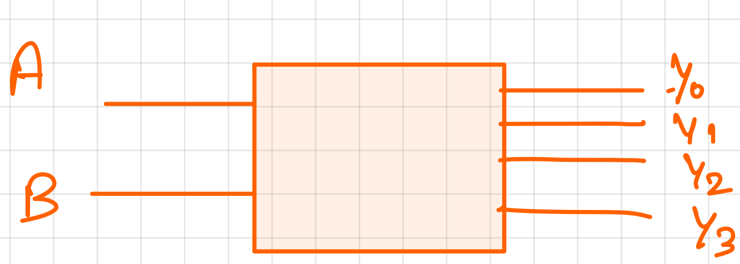
D(I/P)	S_0	S_1	S_2	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
D	0	0	0	D	0	0	0	0	0	0	0
D	0	0	1	0	D	0	0	0	0	0	0
D	0	1	0	0	0	D	0	0	0	0	0
D	0	1	1	0	0	0	D	0	0	0	0
D	1	0	0	0	0	0	0	D	0	0	0
D	1	0	1	0	0	0	0	0	D	0	0
D	1	1	0	0	0	0	0	0	0	D	0
D	1	1	1	0	0	0	0	0	0	0	D



- **De Coder :-**



n = 2^n O/P
I/P



Decode → code → original
Encode → original → code

• Truth Table

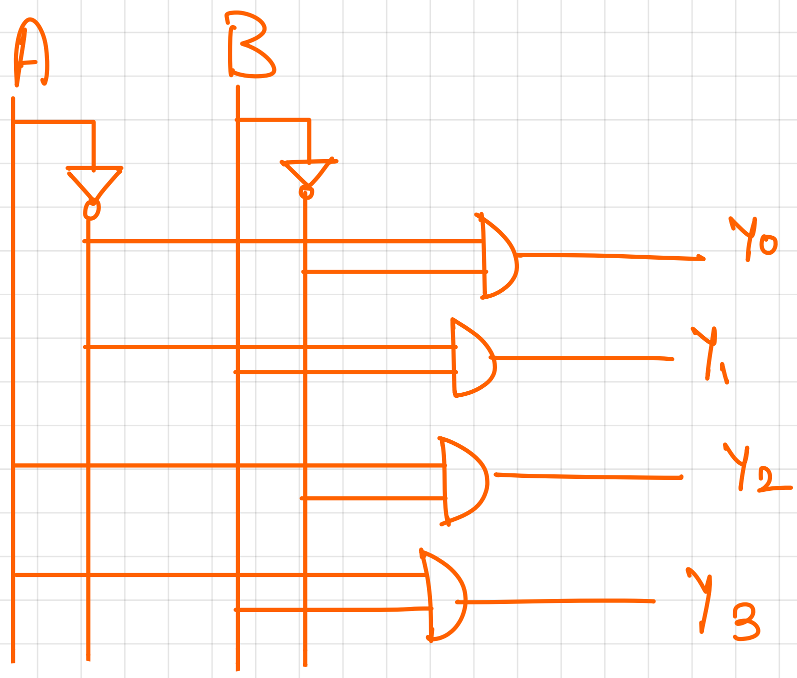
I/P		O/P			
A	B	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$Y_0 = \bar{A}\bar{B}$

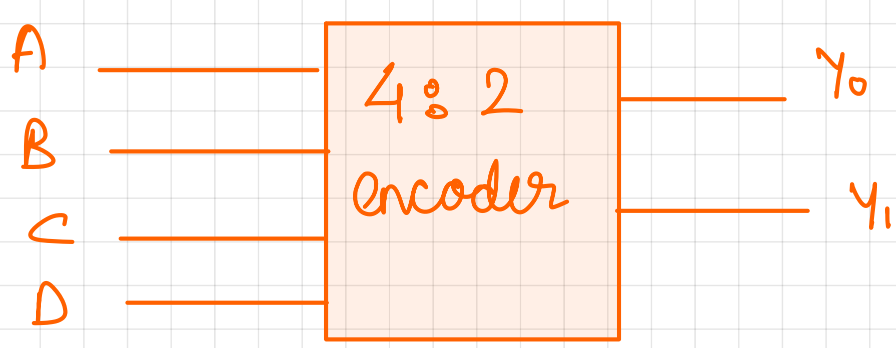
$Y_1 = \bar{A}B$

$Y_2 = A\bar{B}$

$Y_3 = AB$



• Encoder :-



$n = O/P$

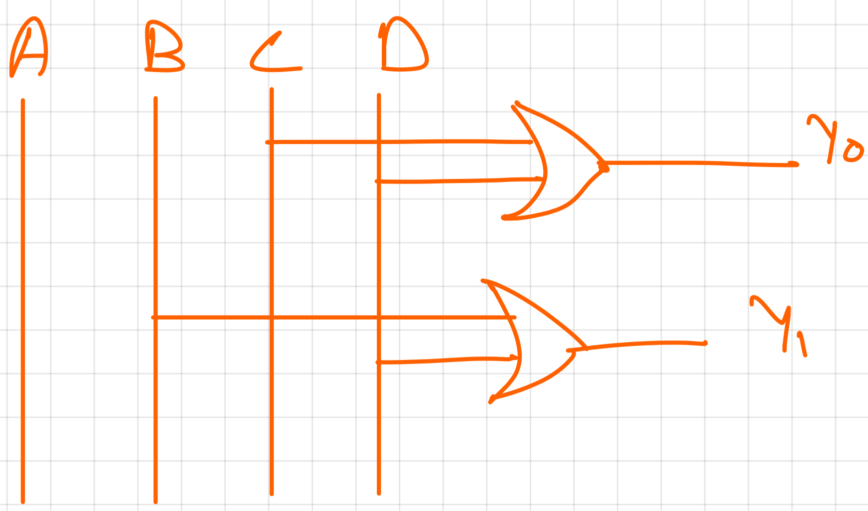
$2^n = I/P$

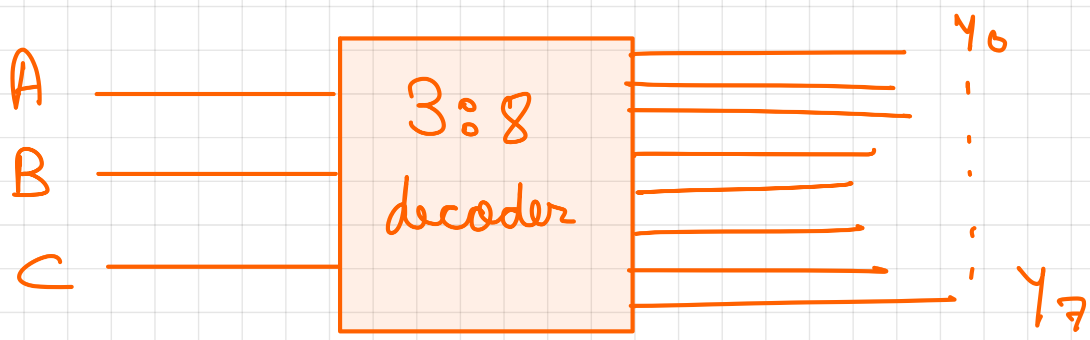
Truth Table :-

I/P				O/P	
A	B	C	D	Y_0	Y_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

$Y_0 = C + D$

$Y_0 = B + D$





Input			Output							
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$\Rightarrow$ $y_0 = \bar{A}\bar{B}\bar{C}$

$y_1 = \bar{A}\bar{B}C$

$y_2 = \bar{A}B\bar{C}$

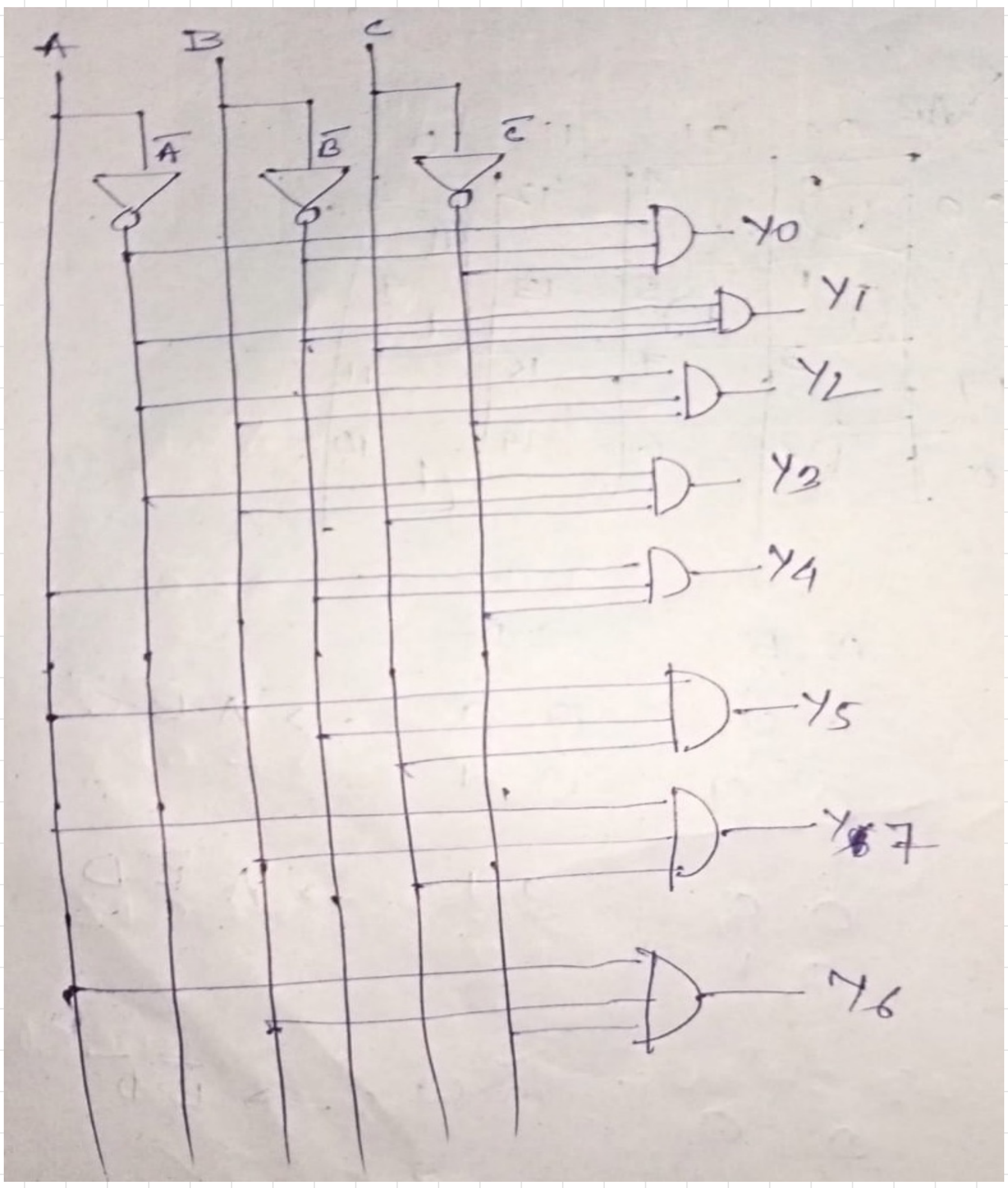
$y_3 = \bar{A}BC$

$y_4 = A\bar{B}\bar{C}$

$y_5 = A\bar{B}C$

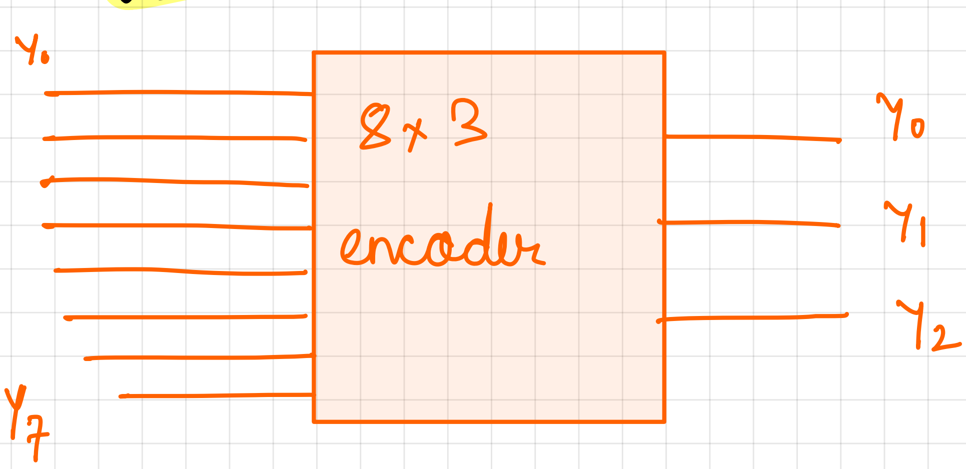
$y_6 = AB\bar{C}$

$y_7 = ABC$



$\Rightarrow$ logic diagram,

- **Encoder** :-

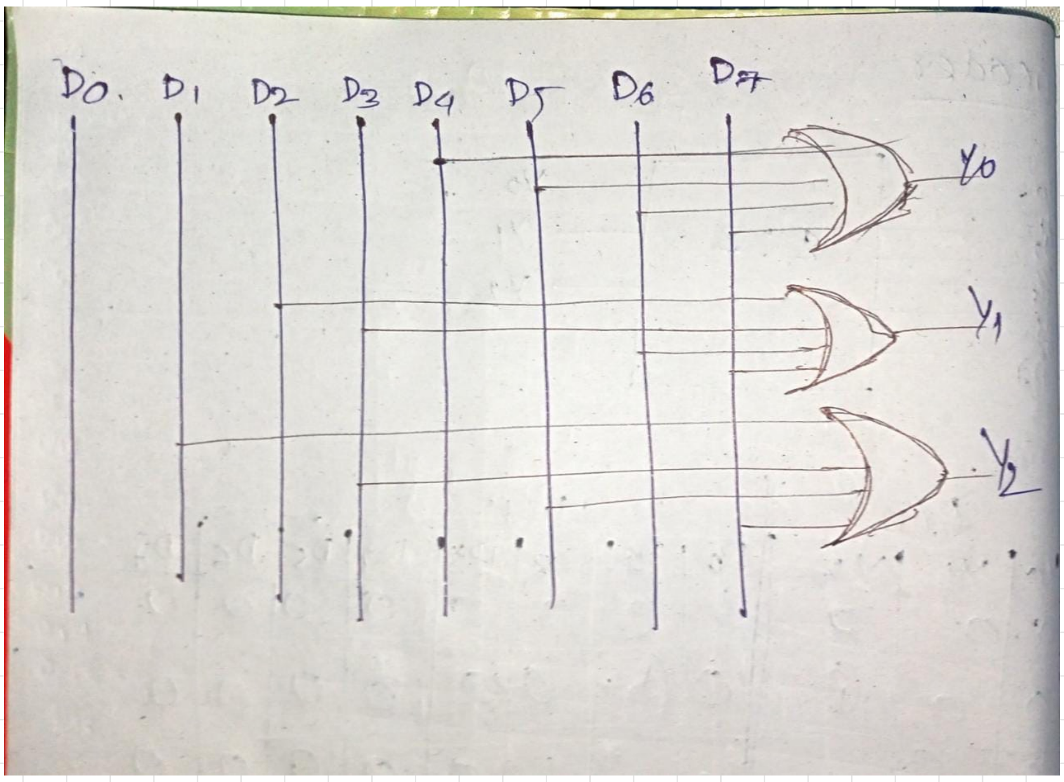


O/P			I/P							
Y_0	Y_1	Y_2	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0		1

$Y_0 = D_4 + D_5 + D_6 + D_7$

$Y_1 = D_2 + D_3 + D_6 + D_7$

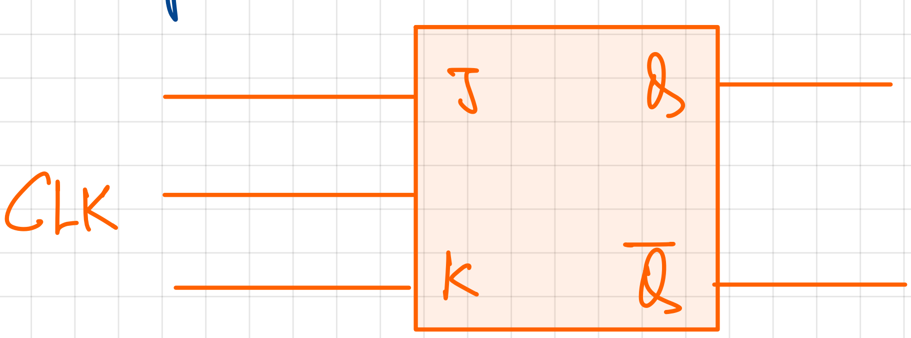
$Y_2 = D_1 + D_3 + D_5 + D_7$



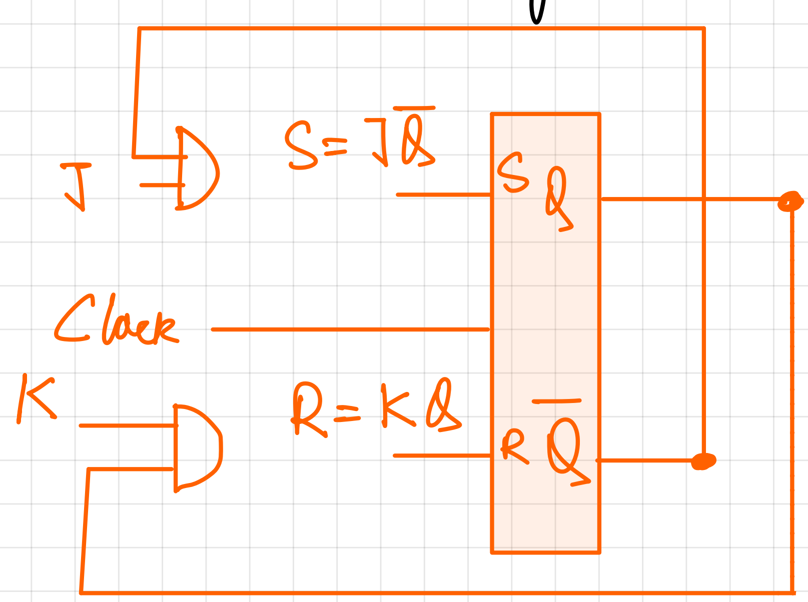
11/Nov/2022

• J-K Flip Flop :-

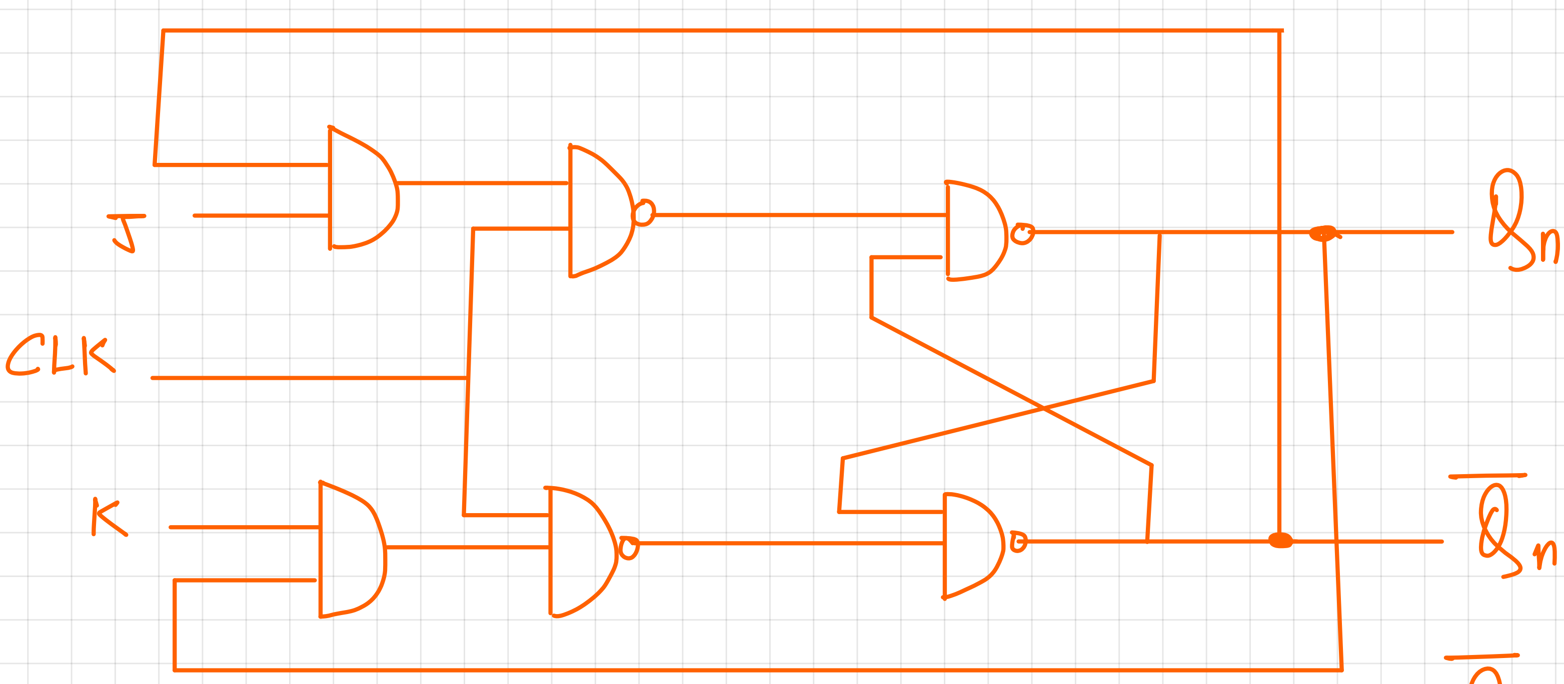
Symbol



Block diagram :-



Circuit diagram :-

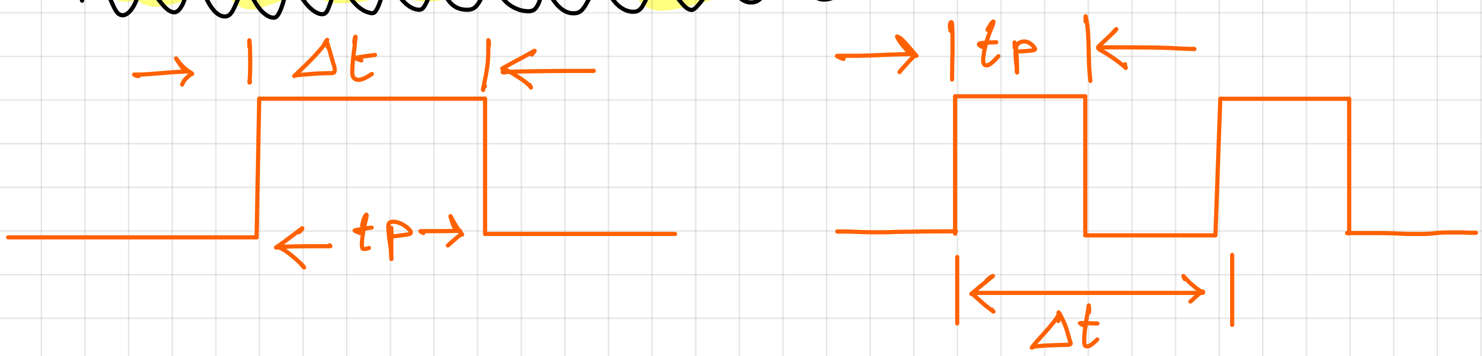


$\overline{Q}_{n+1} =$

Truth Table :-

CLK	Q_n	J K	Q_{n+1}	$\bar{Q}_{n+1}$	Action
1	0/1	0 0	Q_n	$\bar{Q}_n$	No change
1	0/1	0 1	0	1	Reset
1	0/1	1 0	1	0	Set
1	0/1	1 1	1/0	0/1	toggle
0	0/1	0 0			
0	0/1	0 1			
0	0/1	1 0			
0	0/1	1 1			

• Race around Condition :-



$\boxed{\Delta t < \Delta t_p}$ $\boxed{\Delta t > t_p}$

t_p = width of clock pulse

Δt = propagation delay.

When, $J = K = 1$, $Q_n = 0$ then $Q_{n+1} = 1$

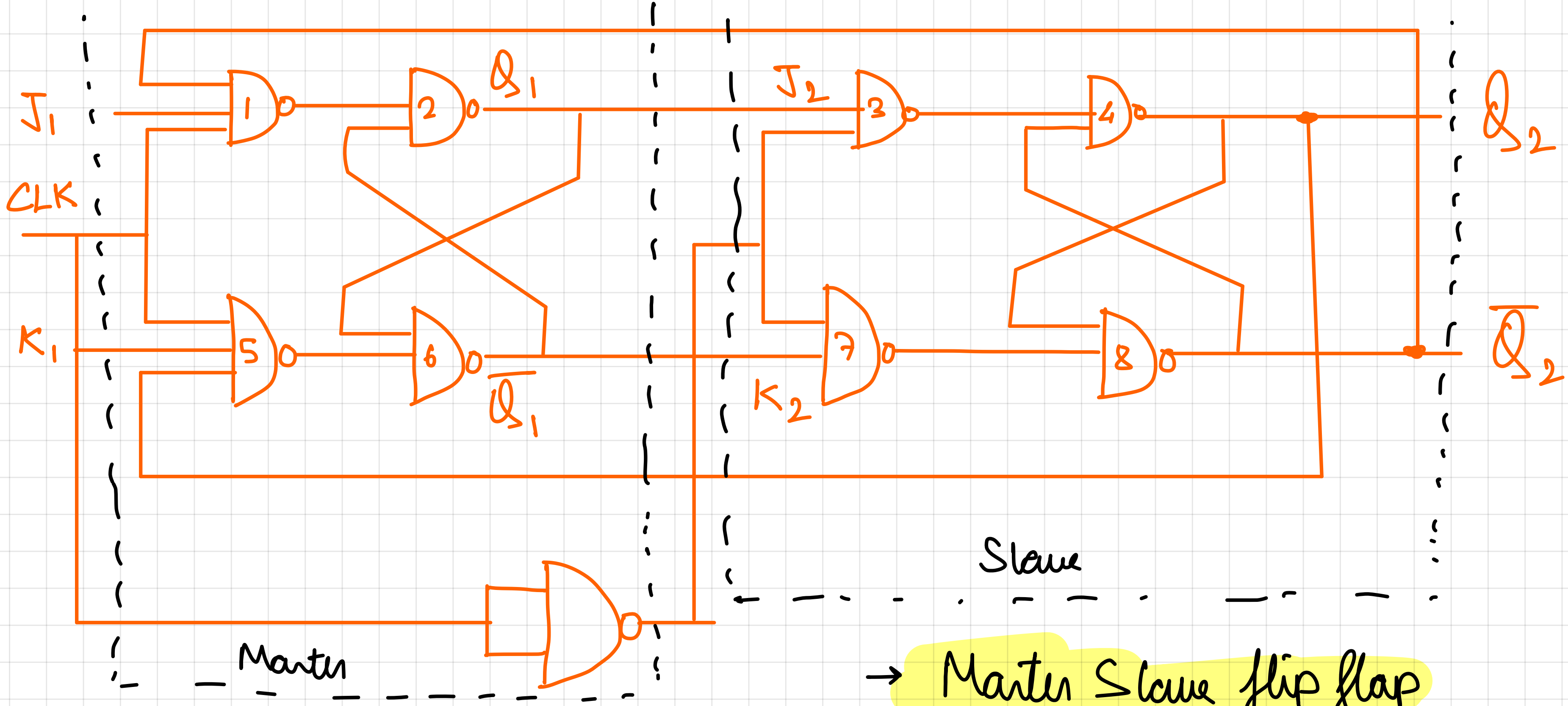
$J = K = 1$, $Q_n = 1$ then $Q_{n+1} = 0$

When, $J = K = 1$

After one propagation delay the output of J-K flip flop is 1. ($Q_n = 0$)

After another propagation time period the O/P of flip flop is 0.

So within the width of clock pulse both the O/P of J-K flip flop are changing.



→ Master Slave flip flop

→ Using 2 JK flip flops the master slave flip flop can be constructed.

→ The 2 J-K flip flops are connected in series.

→ 4 NAND are used to make master flip flop another 4 NAND used to make slave flip flop.

→ The master flip flop is triggered at the positive edge of the clock pulse. And the slave flip flop is triggered at negative edge of clock pulse.

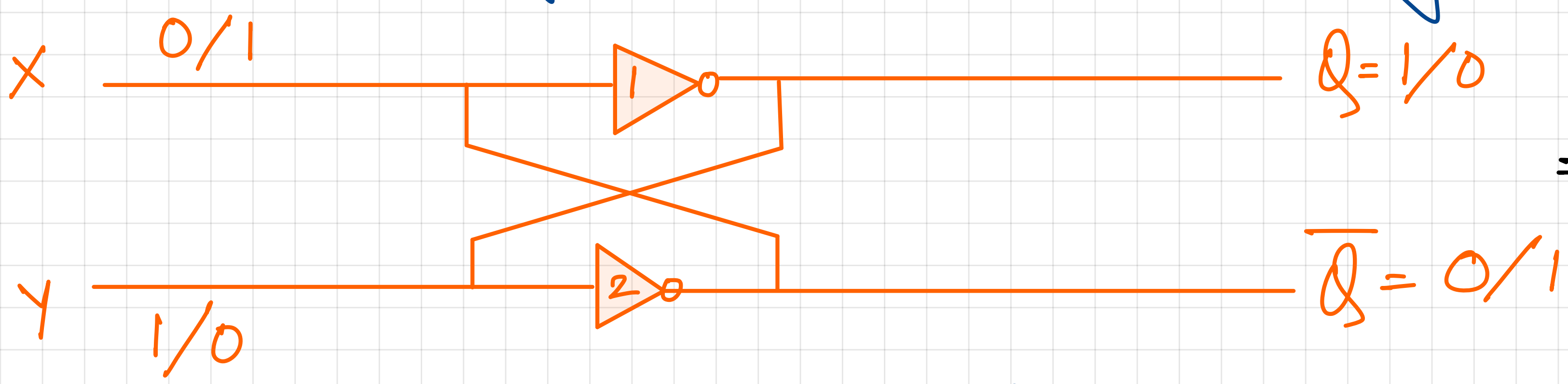
→ When $J_1 = 1$, $K_1 = 0$, the output of the master flip flop (Q_1) = 1 (at positive edge of clock pulse). The output $Q_1 = 1$ drives the input J_2 of the slave flip flop. At the negative edge of clock pulse the slave flip flop will copy the action of master flip flop.

MASTER-SLAVE

• Latch :-

10/Nov/2022

Both the O/P of two inverters are complementary to each other

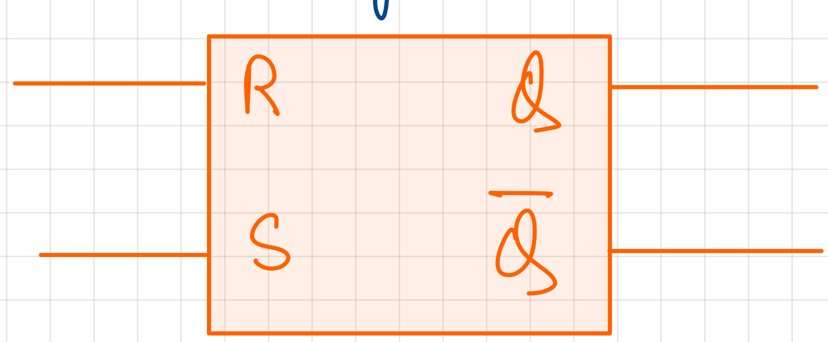


⇒ Basic latch

- The simplest kind of sequential circuit, has only two states
- 1 bit of info is stored.
- logic 0 or logic 1

• NOR-based Set-Reset latch (S-R latch) :-

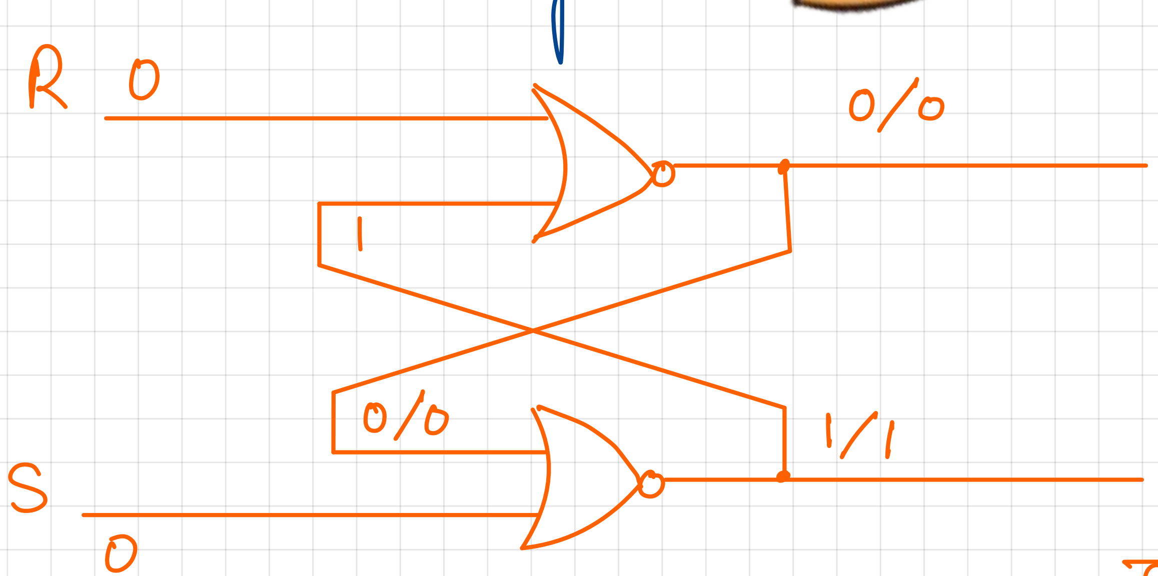
→ Block diagram



→ Truth Table

S	R	Q_{n+1}	$\overline{Q}_{n+1}$	Action
0	0	0/1/Q_n	1/0/$\overline{Q}_n$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	forbidden

→ Circuit diagram



Q_n → Present state

Q_{n+1} → Complement of present state

$\overline{Q}_n$ → Next state

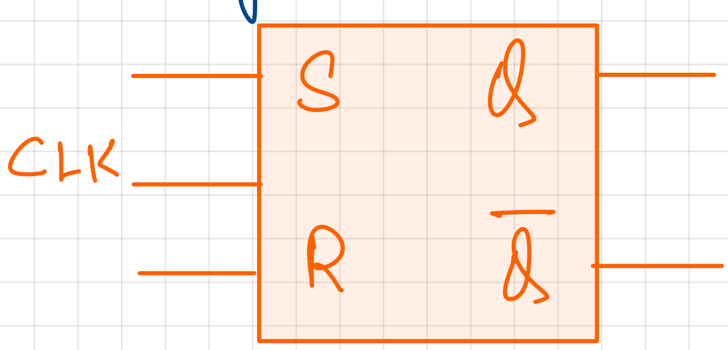
$\overline{Q}_{n+1}$ → Complement of next state.

Flip Flops :-

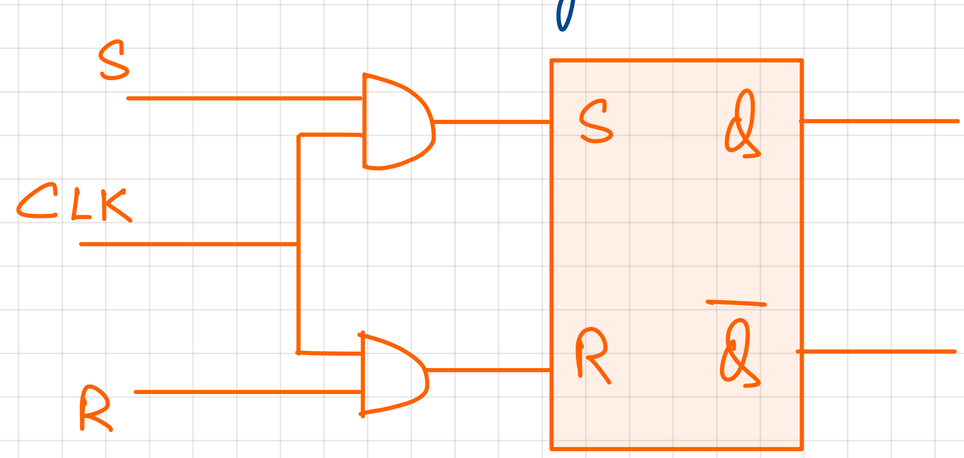
- This is a sequential circuit.
- If we provide an additional control input to the basic latch, then the state of the circuit will be changed.
- The latch with the additional control input is called flip flop.
- The additional control input may be clock or enable input.

S-R flip flop :-

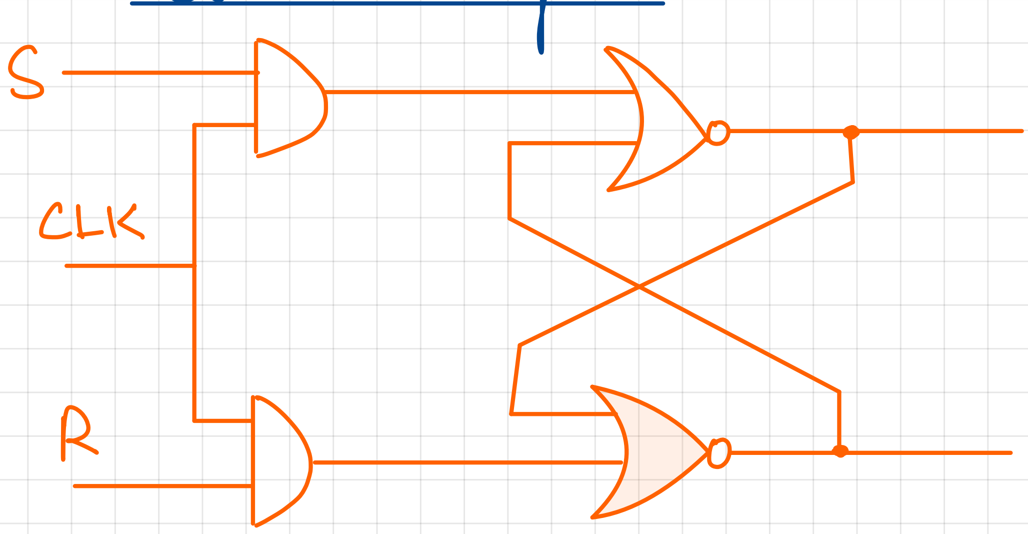
→ Symbol



→ Block diagram



→ Circuit diagram

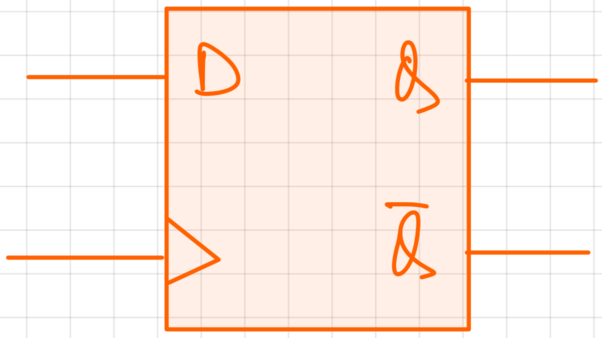


→ Truth Table

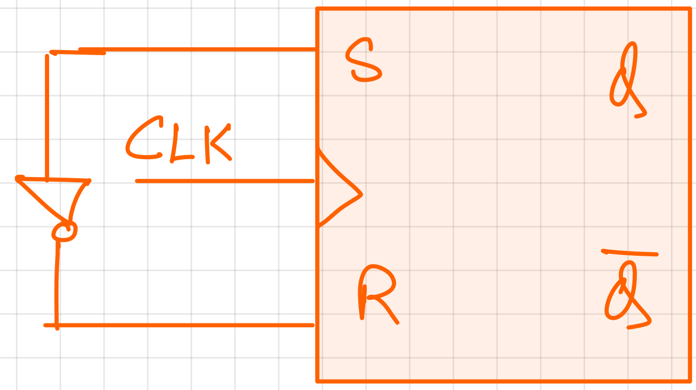
CLK	Q_n	S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Action
1	0/1	0	0	0/1	1/0	No change
1	0/1	0	1	1	0	Set
1	0/1	1	0	0	1	Reset
1	0/1	1	1	?	?	forbidden
0	0/1	0	0	0/1	1/0	No change
0	0/1	0	1	0/1	1/0	No change
0	0/1	1	0	0/1	1/0	No change
0	0/1	1	1	0/1	1/0	No change.

- **Delay flip-flops** (D- flip flops)

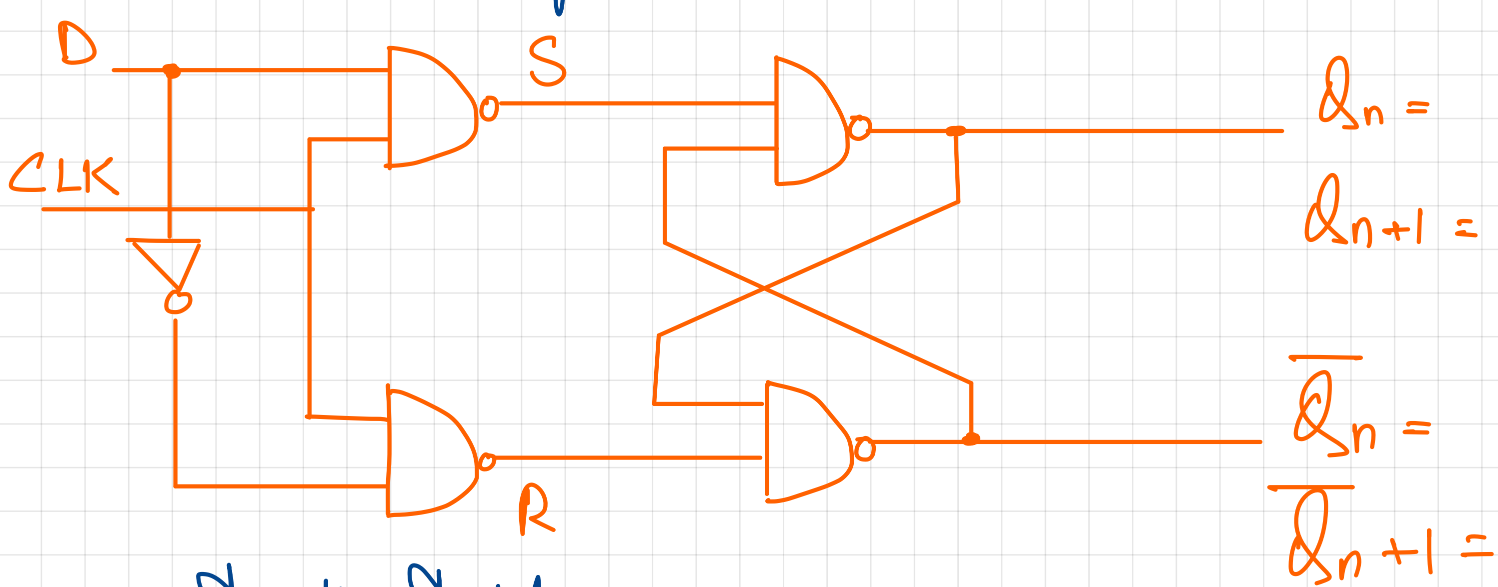
→ Symbol



→ Using S-R flip flop



→ Circuit diagram



→ Truth Table

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

17/Nov/2022

(i) When $J_1 = 0$ & $K_1 = 1$, the output of the master $Q_1 = 0$ and $Q_1' = 1$, $Q_1' = 1$, this will help to trigger the K_2 input of the slave flip flop.

(ii) At the negative edge of the clock pulse the slave flip flop will copy the same action of the master flip flop.

(iii) When $J_1 = K$ & $K_1 = 1$, the master flip flop toggles on the edge of the clockpulse & the slave flip flop toggles at the −ve edge of the clock pulse.

(iv) When $J_1 = 0$ & $K_1 = 0$, the master and slave flip flop will remain at the same state.

Implementation of Boolean expression using multiplexer :-

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

$(n+1) = 3+1$

Select input BCD

I/P of multiplexer $A = A, \overline{A}, 0, 1$

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\overline{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	1	0	$\overline{A}$	$\overline{A}$	0	0	A

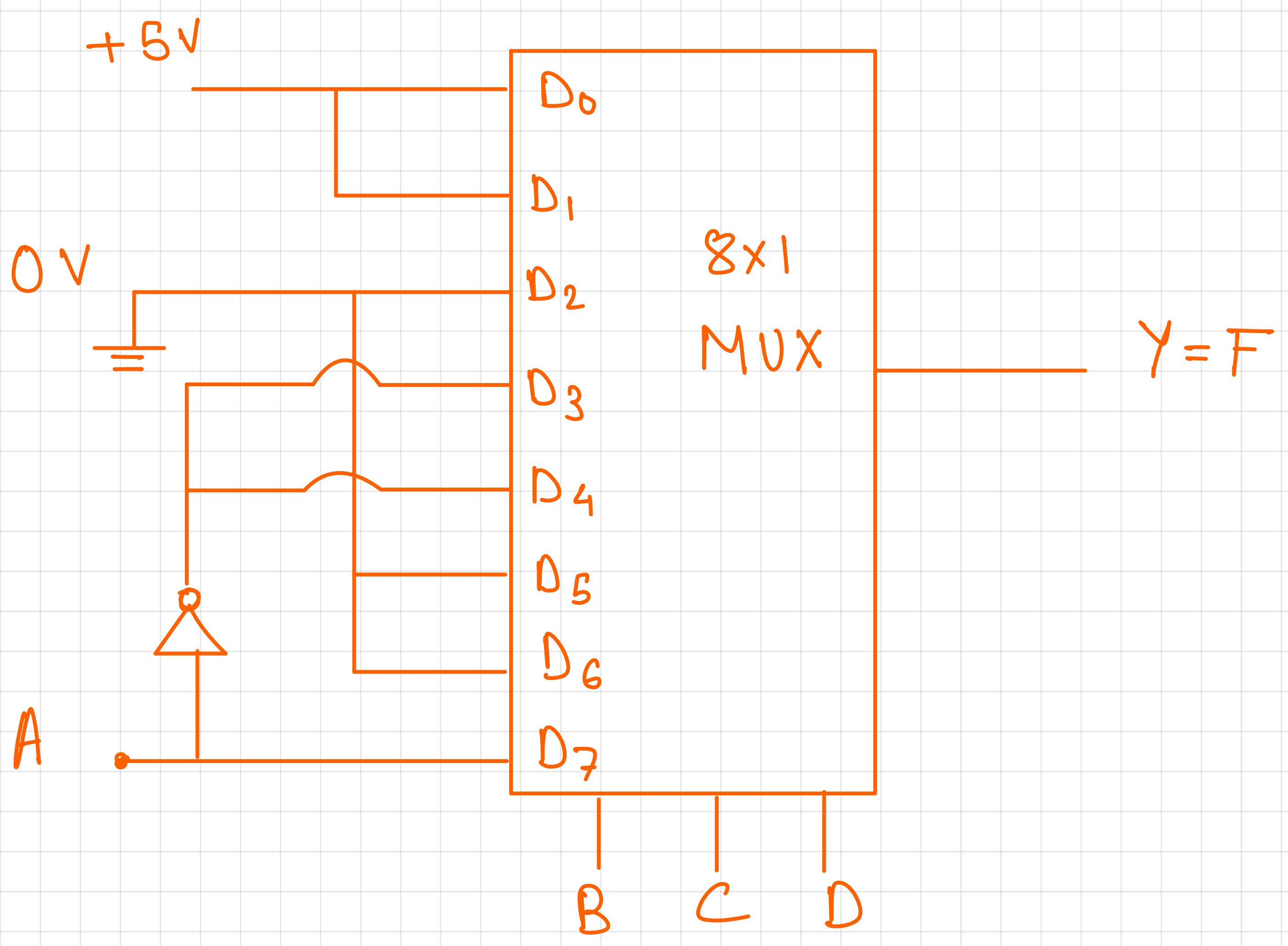
Rules :-

(i) If both the min terms in a column are not circled applied 0 to the corresponding I/P.

(ii) If both the min terms in a column are circled apply 1 to the corresponding I/P.

(iii) If bottom minterm is circled and top is not circled apply A to the corresponding I/P.

(iv) If the top min term is circled and the bottom is not, then apply $\overline{A}$ to the corresponding I/P.



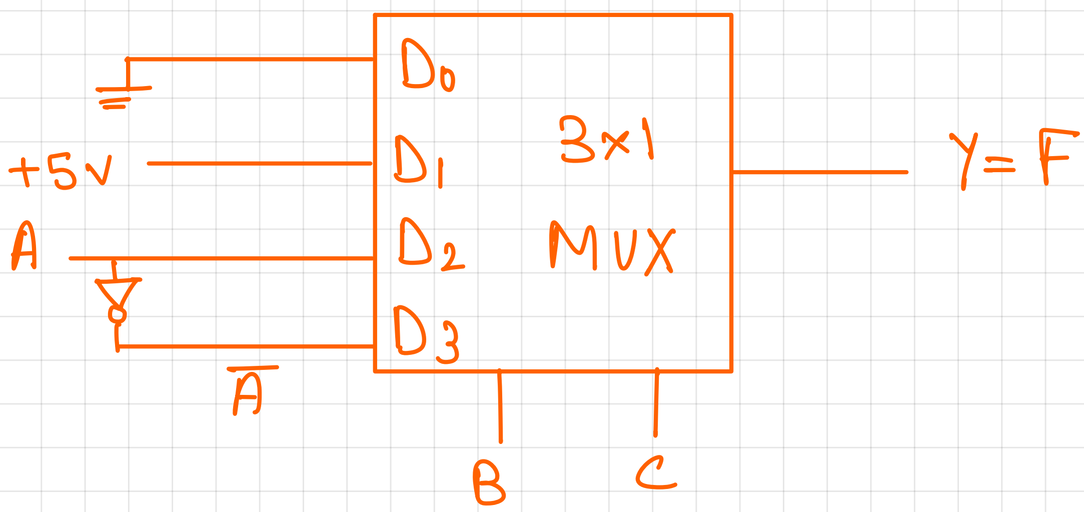
$$P(A,B,C) = \sum(1,3,5,6)$$

$$(n+1) = 2+1$$

$n \to BC$, $1 \to A$

$$2^2 = 4$$

	D_0	D_1	D_2	D_3
$\bar{A}$	0	1	2	3
A	4	5	6	7
	0	1	A	$\bar{A}$



Counters

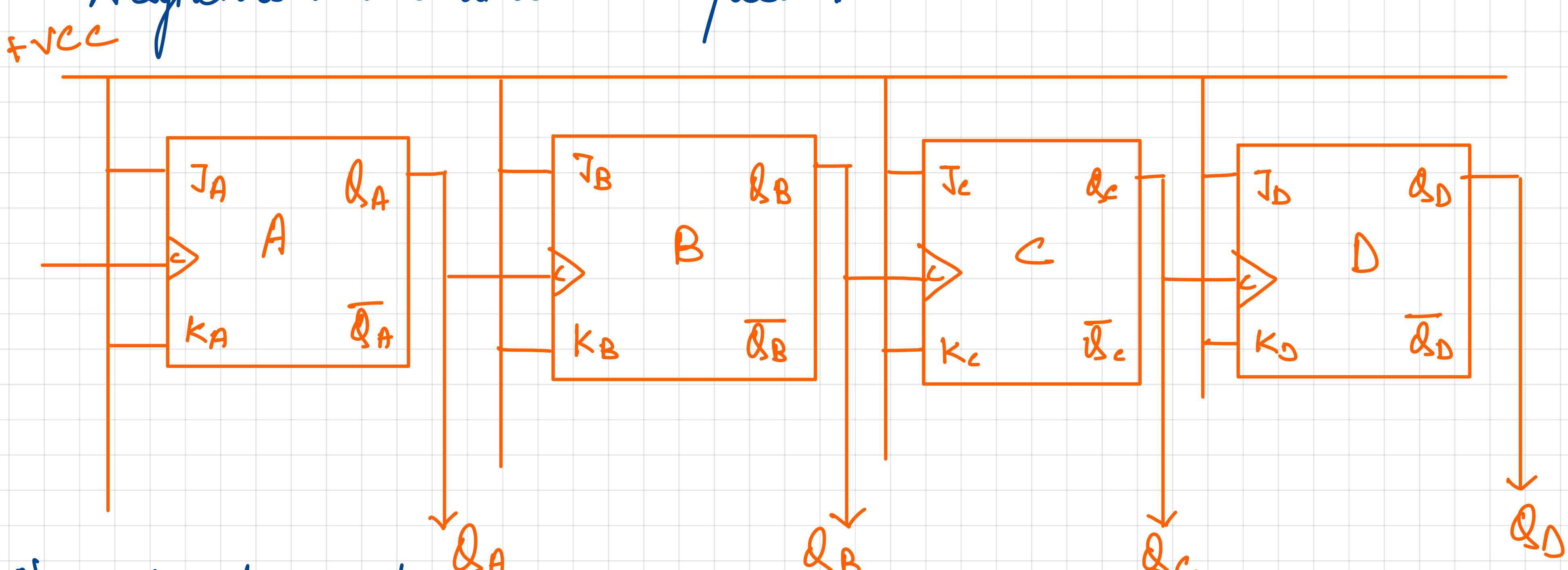
21/Nov/2022

i) It is a sequential circuit.

ii) It consists a set of flip flops.

iii) Counts the binary input sequence in digital form.

Asynchronous counter — speed ↓



i) A flip flop is triggered by clock input pulse.

ii) B flip flop is triggered by the Q_A, output of A flip flop.

iii) C flip flop is triggered by the Q_B, output of B flip flop.

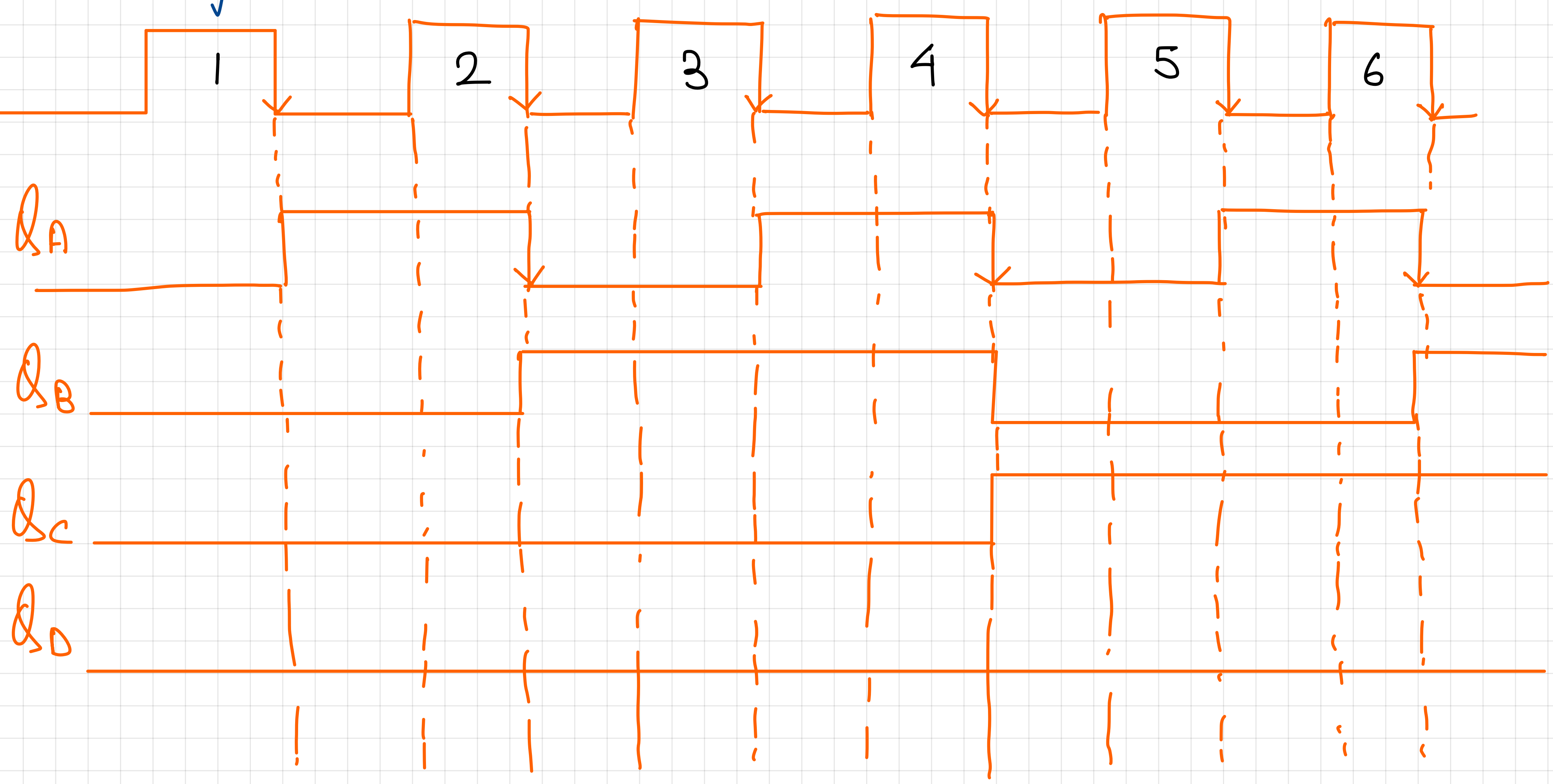
iv) D flip flop is triggered by the Q_C, output of C flip flop.

v) All J-K input of A, B, C, D are connected with +VCC.

vi) All flip flops are toggling as $J = K = 1$, at the −ve transition of clock pulse I/P.

- **Clock Pulse input** :-

Initially $Q_A = Q_B = Q_C = Q_D = 0$



- **Truth Table** :-

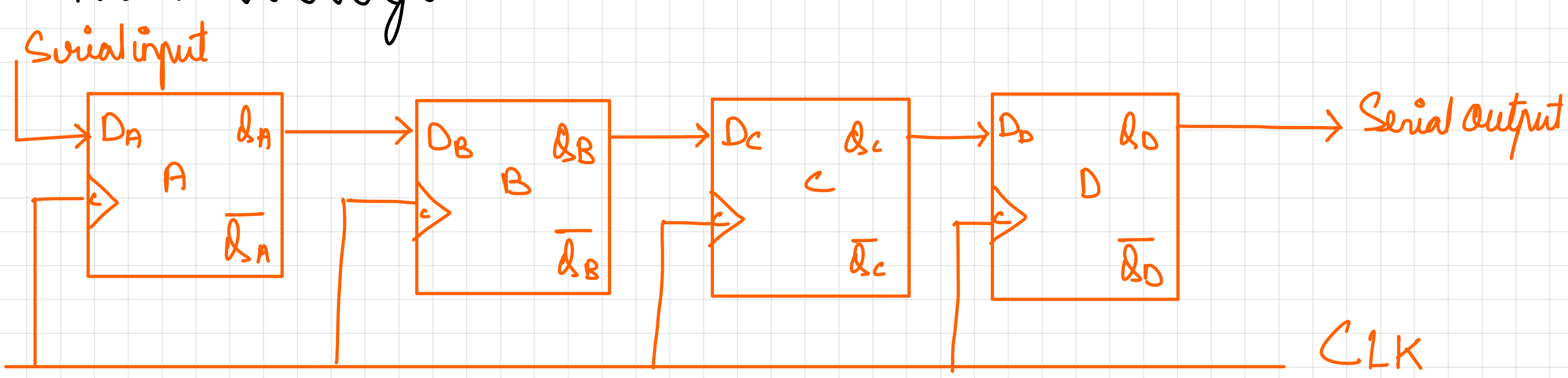
State	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Register :-

Function : Register consist a set of flipflops to store binary information in it.
Also registers can shift binary information from one side to another side.
There are 4 types of register :

1. SISO (Serial In Serial Out)
2. SIPO (Serial in Parallel Out)
3. PISO (Parallel in Serial Out)
4. PIPO (Parallel in Parallel Out)

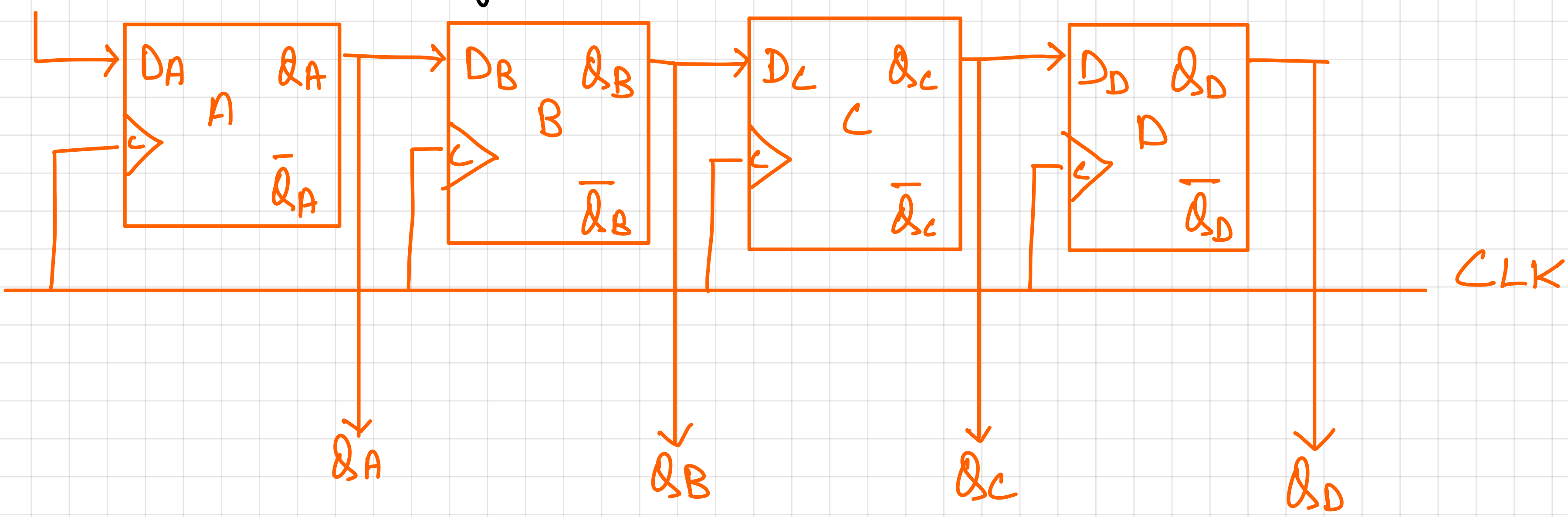
SISO Register :-



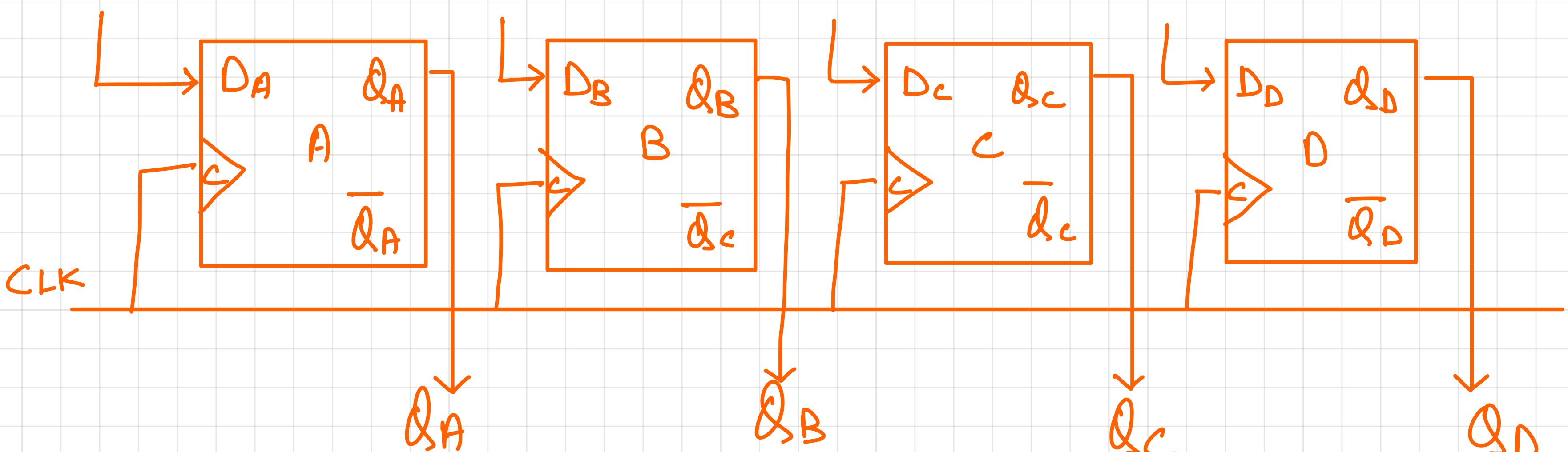
Truth Table :

CLK	Q_A	Q_B	Q_C	Q_D
X	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	0	1	1	0
4	1	0	1	1

SIPO Register :-

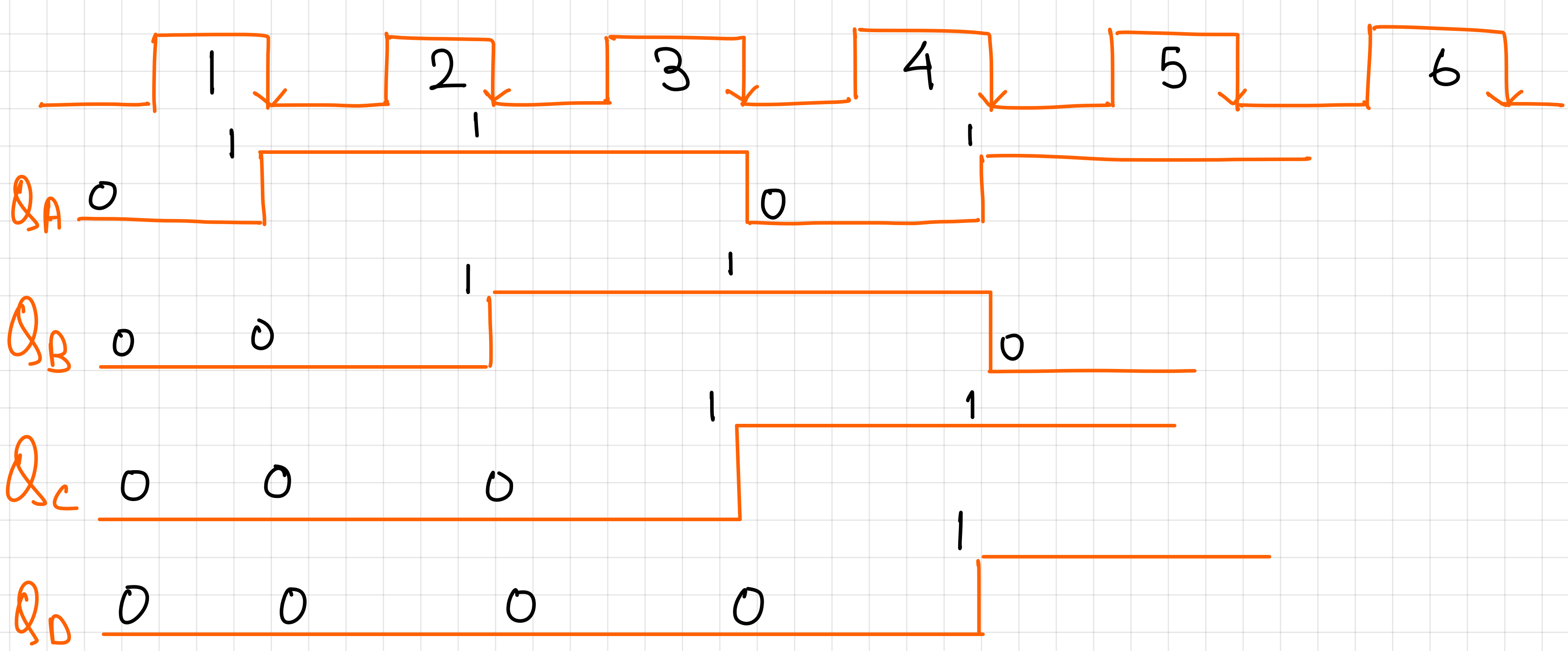


PIPO Register :-



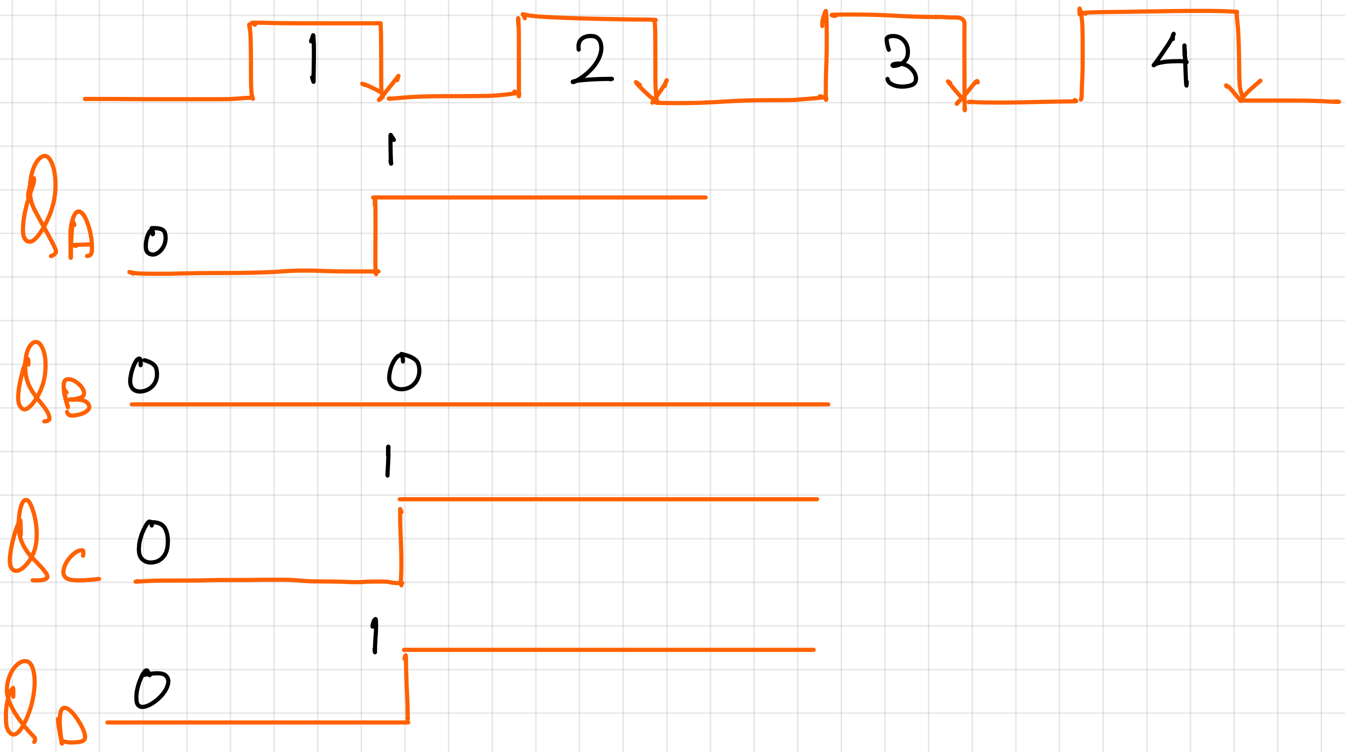
CLK	Q_A	Q_B	Q_C	Q_D
X	0	0	0	0
1	1	0	1	1

SISO / SIPO

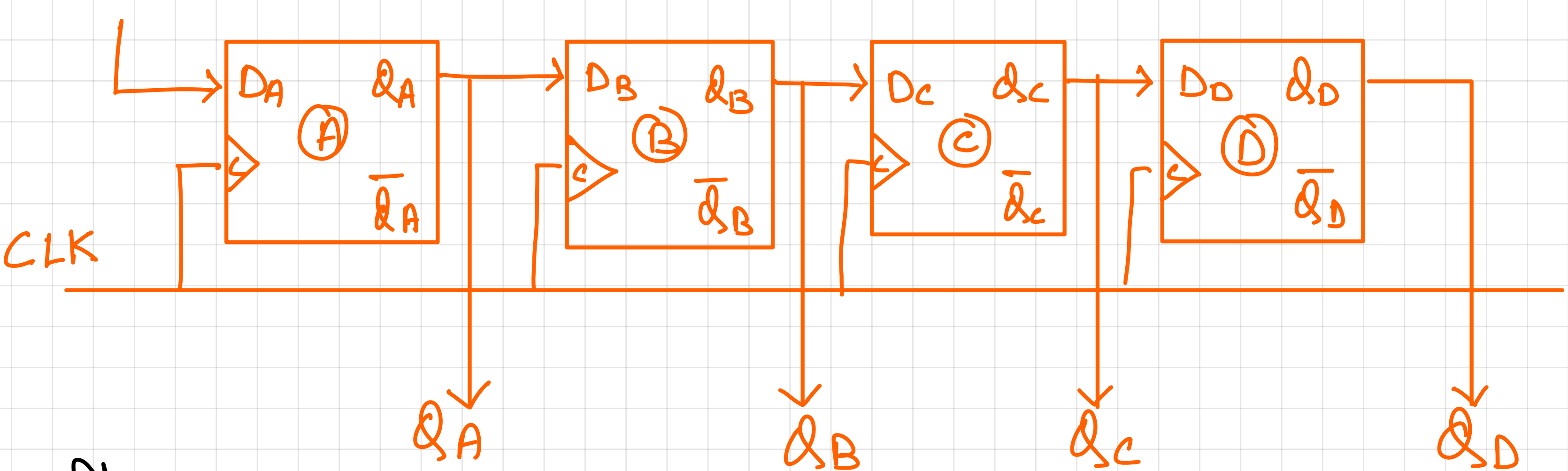


{ SISO → for storing & SIPO → for storing and getting output. }

PIPO :-



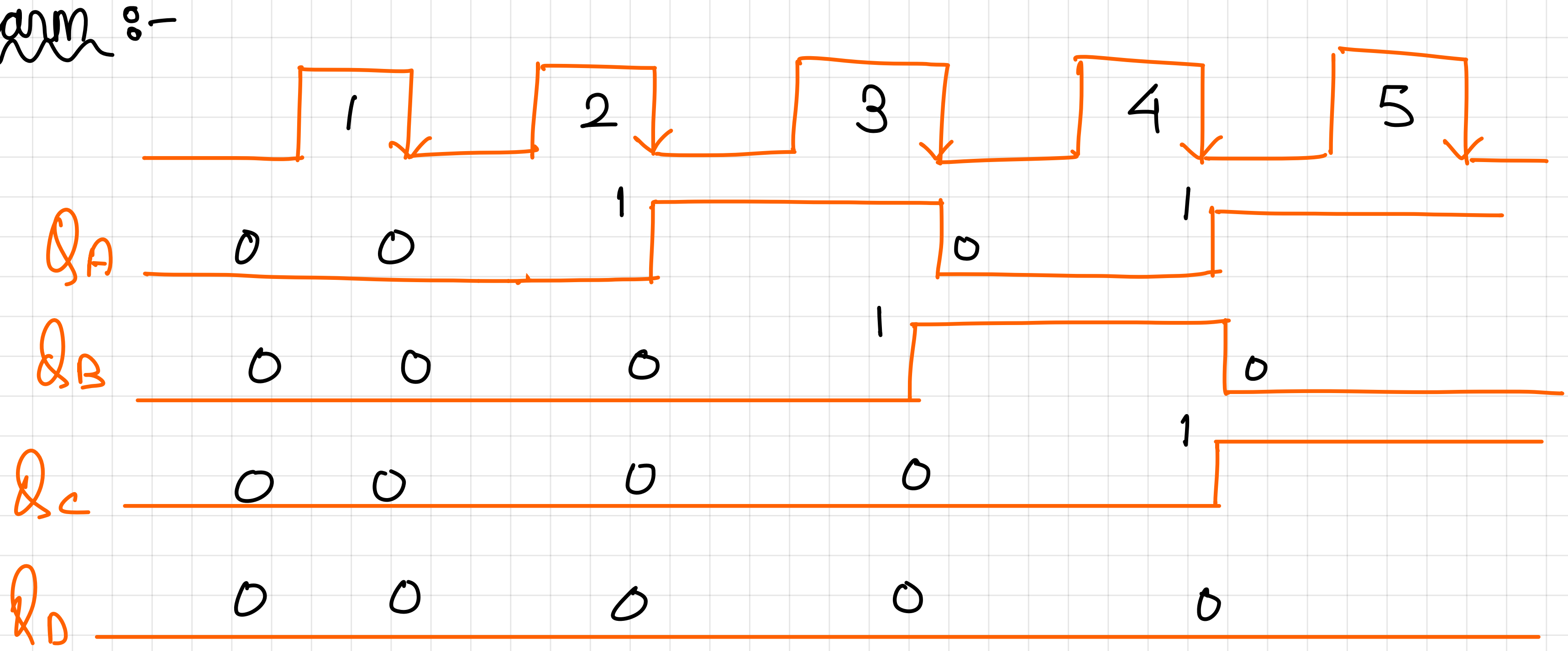
For SIPO (1010) :-

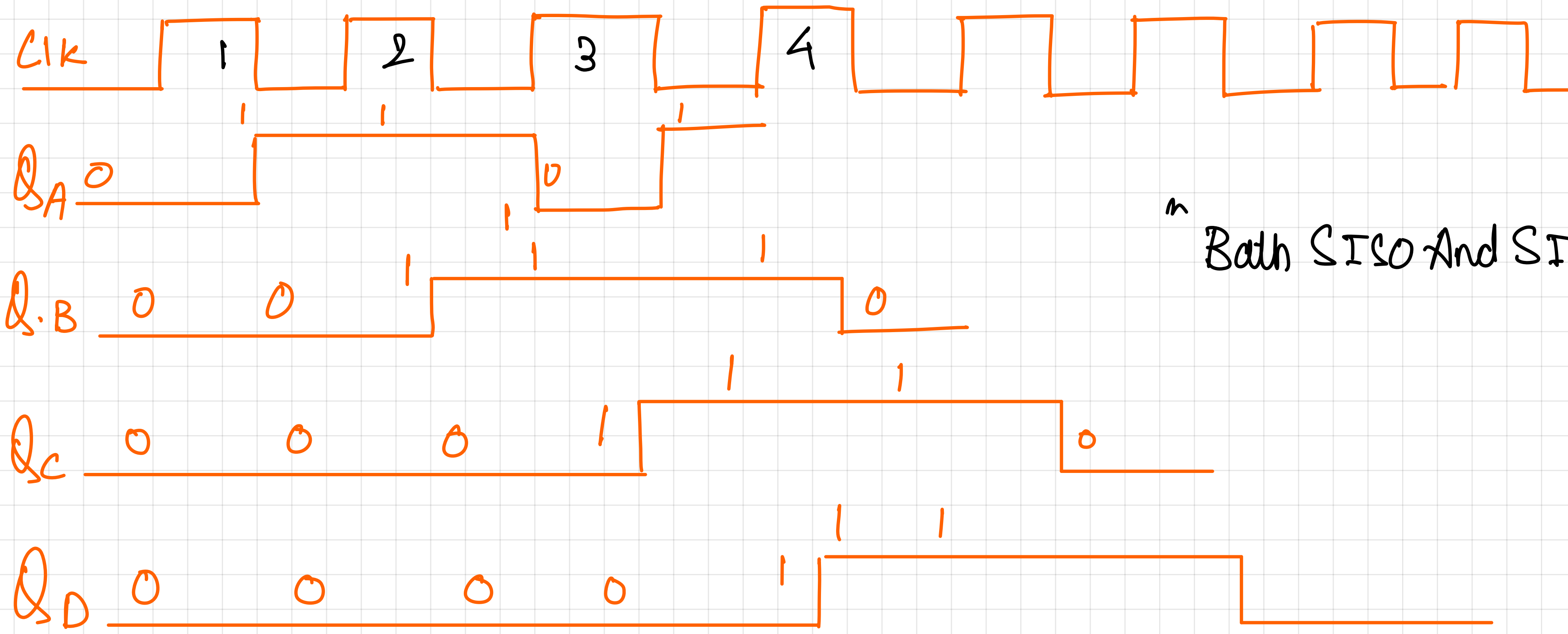


Truth Table :-

CLK	Q_A	Q_B	Q_C	Q_D
X	0	0	0	0
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	1	0	1	0

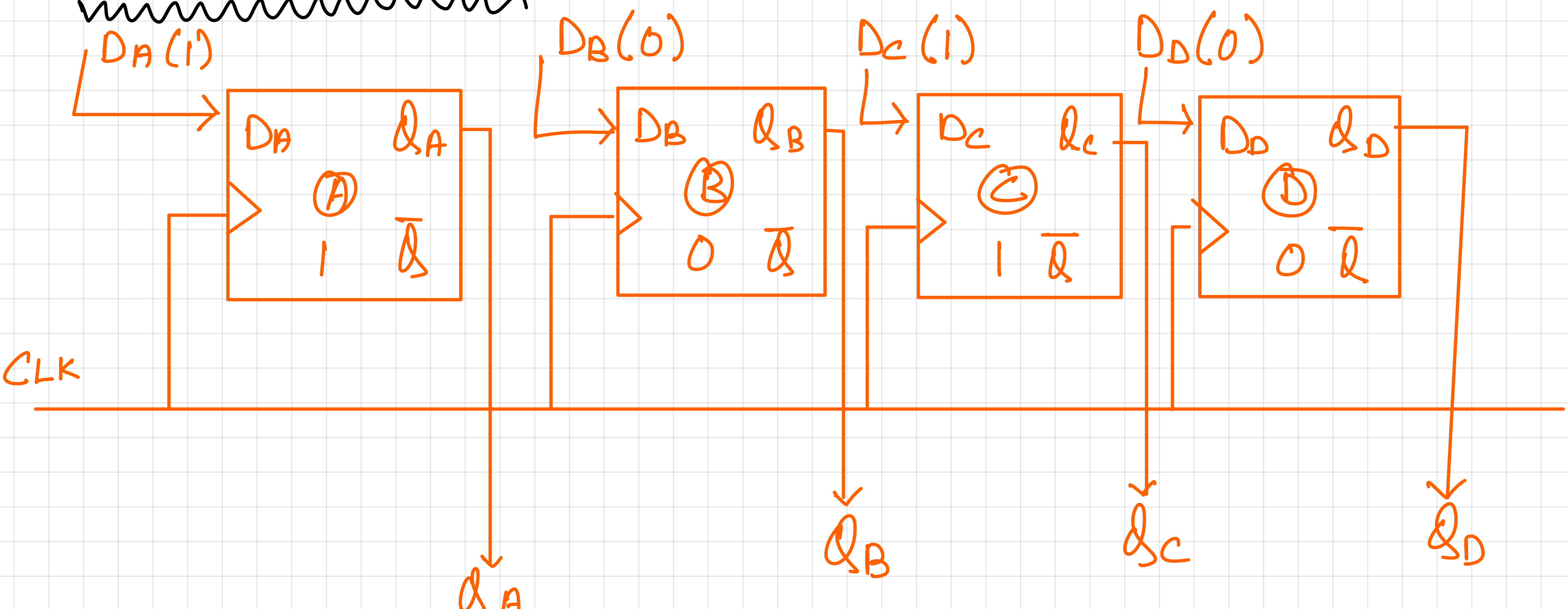
- **Waveform :-**





"Both SISO And SIPO"

- For PIPO → 1010



- Truth Table :-

CLK	Q_A	Q_B	Q_C	Q_D
X	0	0	0	0
1	1	0	1	0

- Waveform :-