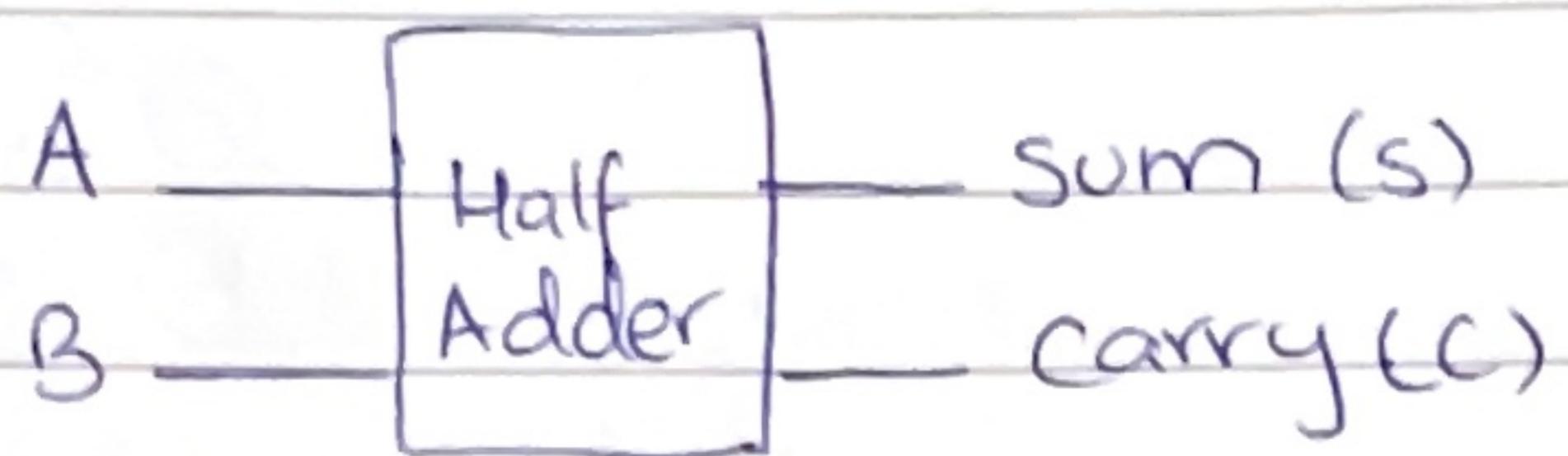


Date

## HALF ADDER:-

A combinational circuit that adds two bit and it gives the sum bit and carry bit as the output.



check last page  
using Jigs  
k-maps

(according to sir)  
→ this should be derived

## TRUTH TABLE

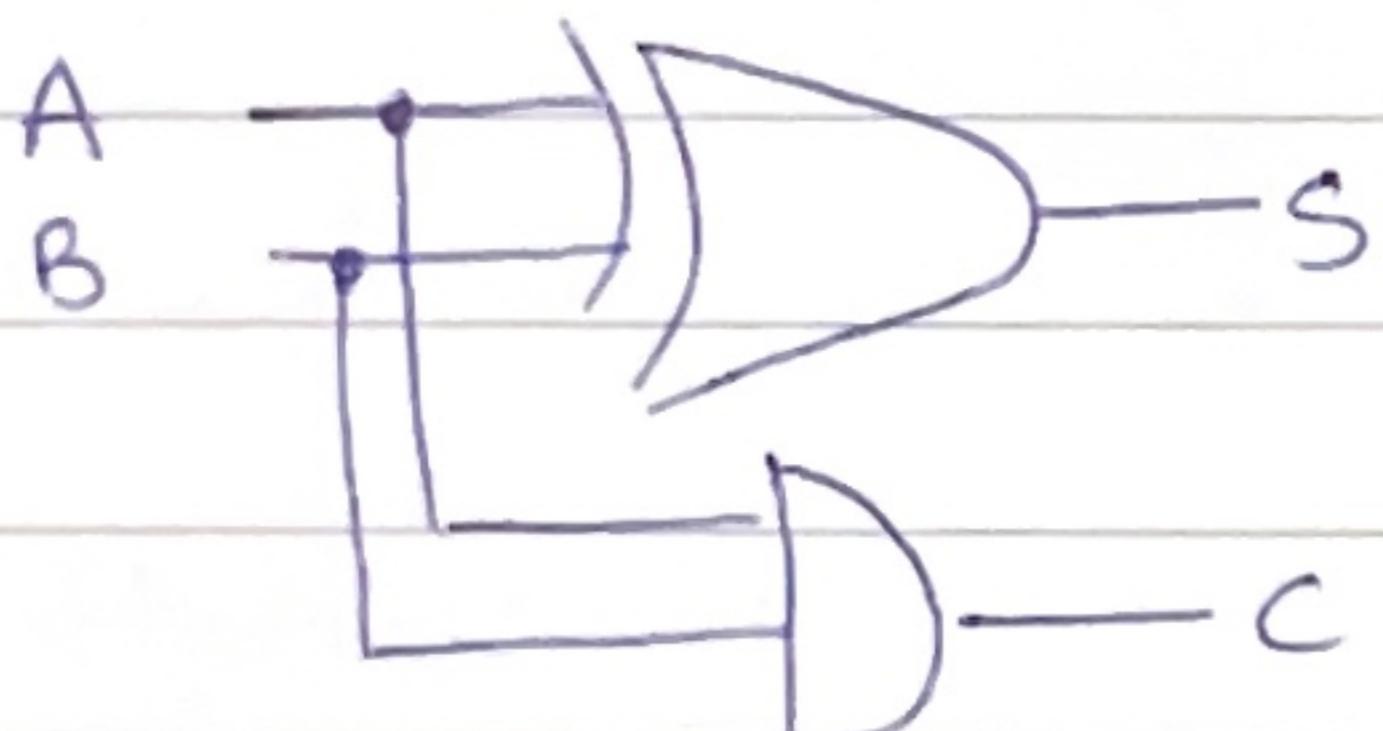
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	X0	1

## BOOLEAN EXPRESSION.

$$S = \bar{A}B + A\bar{B} \quad (\text{SUM}) \\ = A \oplus B$$

$$C = AB \quad (\text{CARRY})$$

## LOGIC CIRCUIT:-



The half adder circuit works perfectly for two bits when there is no incoming carry

\* But, if there is an incoming carry from previous addition then we cannot add that using the half adder

example:-

①

addition	0	1
cannot	+	1
be	1	1

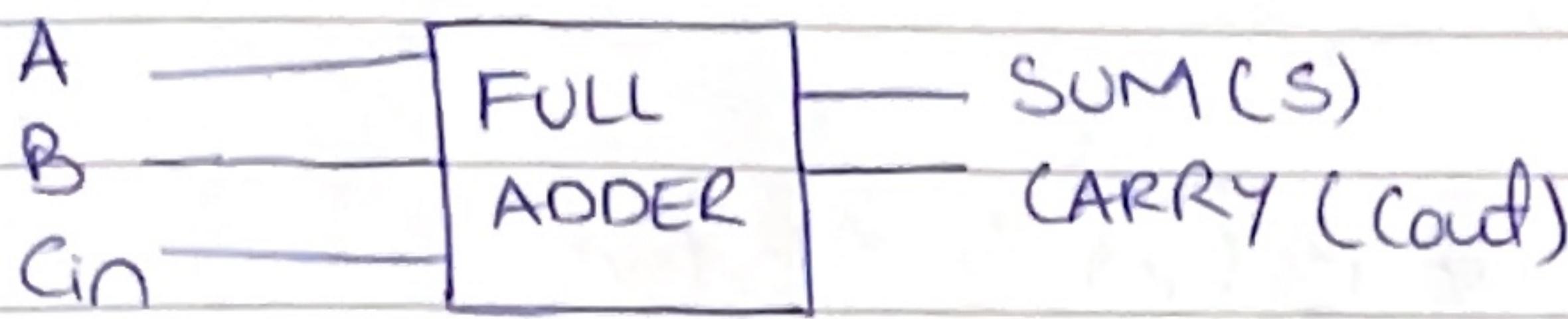
performed for the next bit using half adder. 1

## FULL ADDER:-

" A combinational Circuit which adds 2 bits and an incoming carry bit to generate outputs."

\* Overcomes the limitations of half adder circuit.

Date



A	B	Cin	SUM	CARRY (Cout)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Expression:- (" ; check note on same heading)  
(CSUM)

$$S = \overline{A} \overline{B} \text{Cin} + \overline{A} B \overline{C} \text{Cin} + A \overline{B} \text{Cin} + A B \text{Cin}$$

$$= \overline{A} (\overline{B} \text{Cin} + B \overline{\text{Cin}}) + A (\overline{B} \text{Cin} + \overline{B} \text{Cin})$$

$$= \overline{A} (B \oplus \text{Cin}) + A (\overline{B} \oplus \text{Cin})$$

let  $B \oplus \text{Cin} = X$

$$\therefore = \overline{A}(X) + A(\overline{X})$$

$$S = \overline{A} \oplus X$$

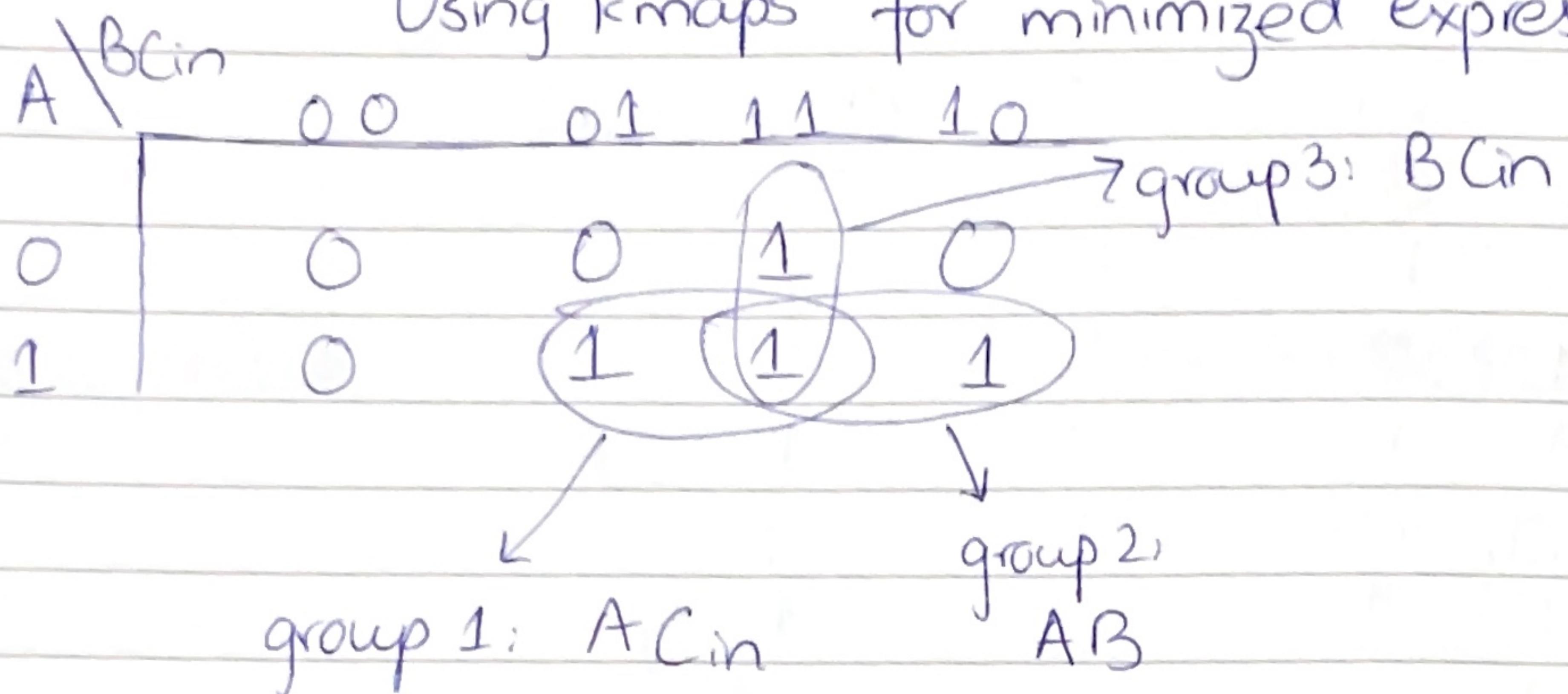
$$= \overline{A} \oplus B \oplus \text{Cin} \quad (\text{Minimal expression for sum output})$$

Date

(Cout / Carry)

$$\text{Cout} = \bar{A}\bar{B}C_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in} \quad \dots \textcircled{1}$$

Using kmaps for minimized expression.



$$\text{Cout} = AB + BC_{in} + AC_{in} \quad (\text{one way of writing simplified expression})$$

Simplifying ①

$$\text{Cout} = (C_{in} + \bar{C}_{in}) AB + C_{in} (\bar{A}B + A\bar{B})$$

$$= AB(1) + C_{in}(A \oplus B)$$

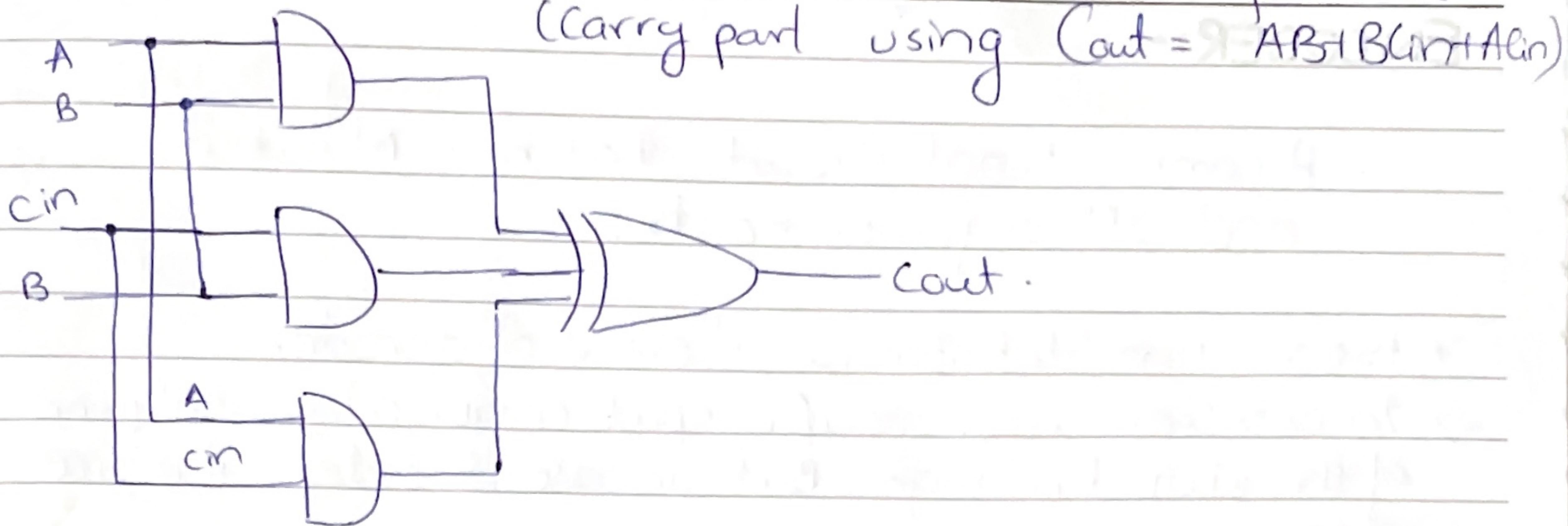
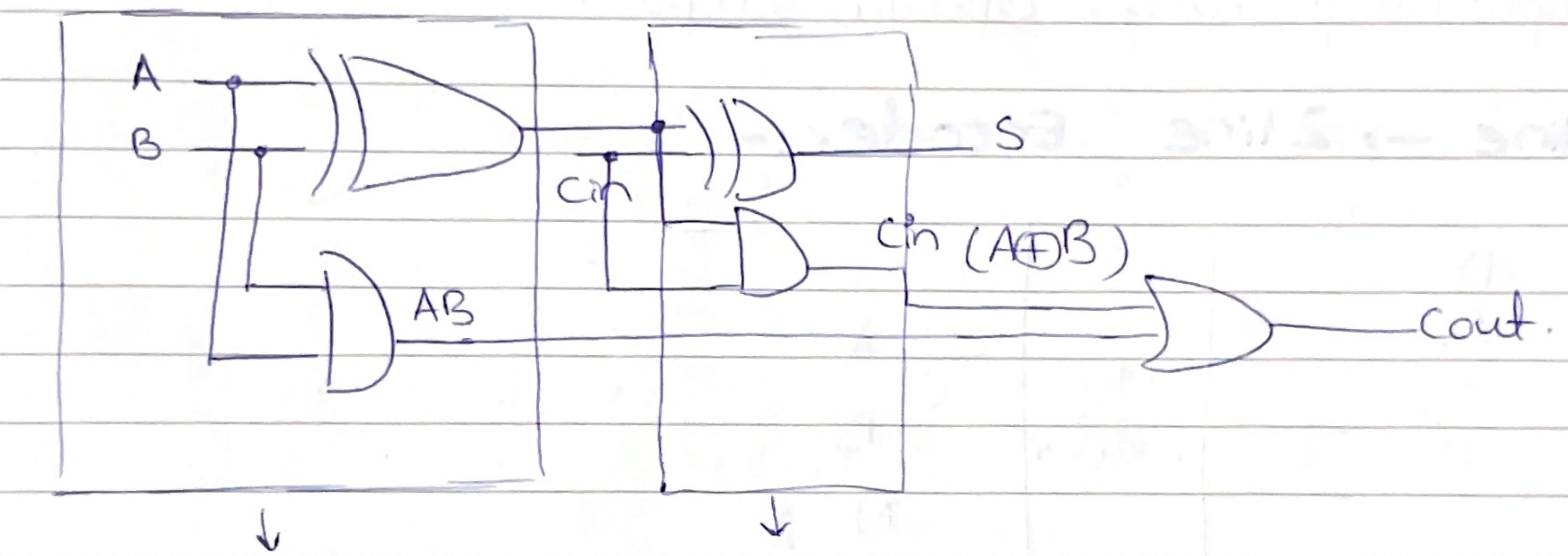
$$= AB + C_{in}(A \oplus B) \quad (\text{one way of writing simplified expression.})$$

LOGIC CIRCUIT:-



1-L 2:50-3 3-4

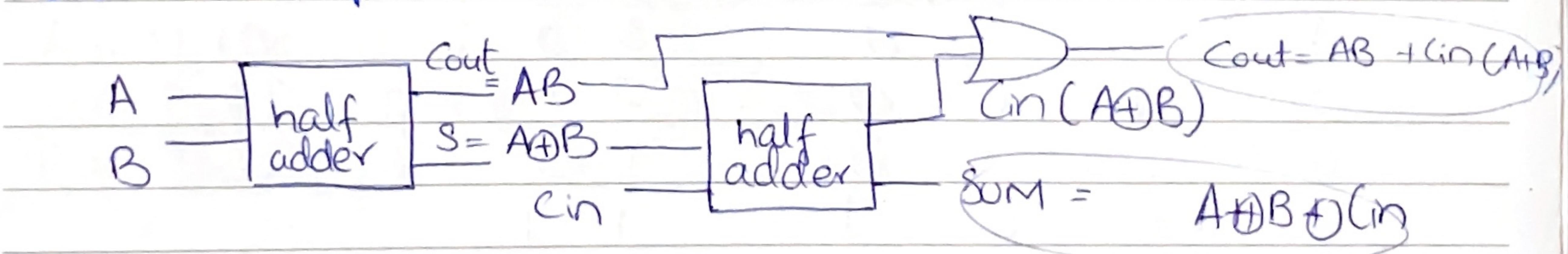
Date \_\_\_\_\_

2 Bit addition  $\rightarrow$  half adder3 Bit addition  $\rightarrow$  full adderAnother method

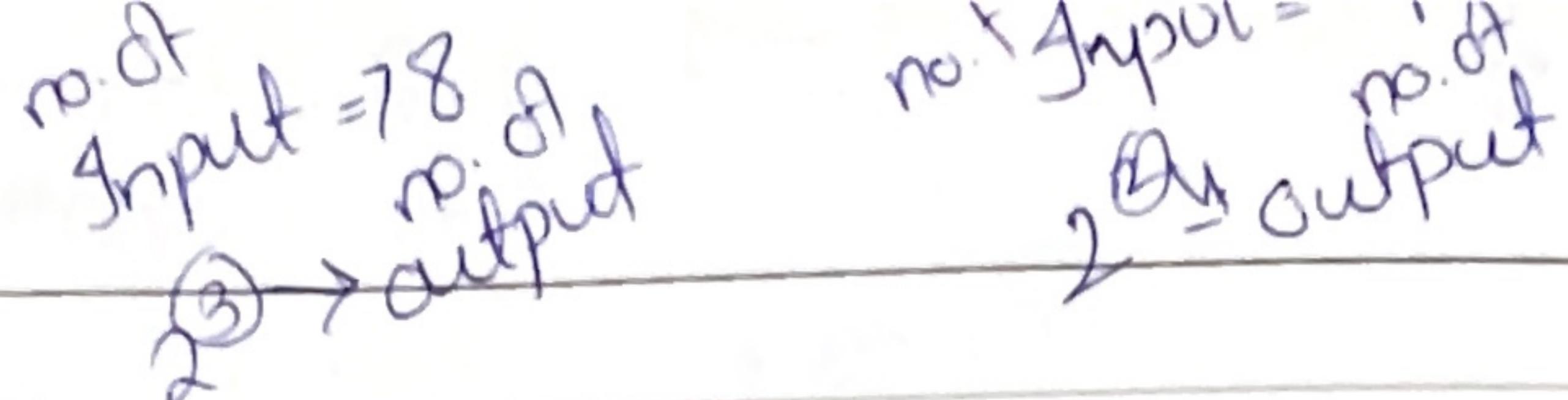
half adder  
(1 XOR & 1 andgate)

half adder.  $\Rightarrow$  full adder.  
(ii)

## BLOCK-DIAGRAM.



Date

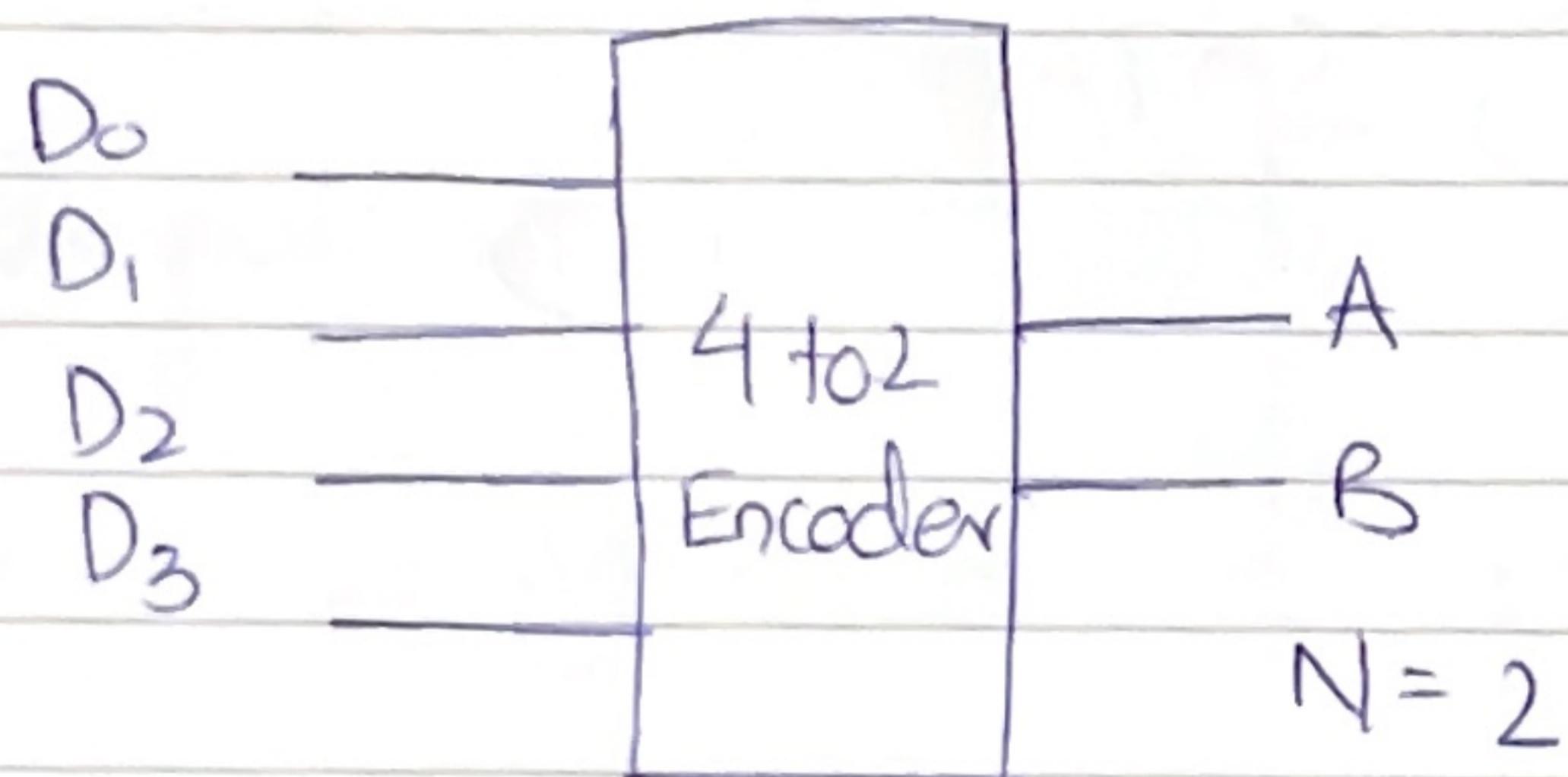


## ENCODER:-

A combinational Circuit that has N outputs; and  $2^N$  or fewer inputs.

- \* its operation/definition is viceversa of decoder.
- $\Rightarrow$  In Decoder, as per specific input combination only one of the output is high, But in case of encoder it is vice versa
- $\Rightarrow$  In encoder, it is assumed that at any given time only one of the input is high and depending upon which input is high we get a specific output.

## 4 line $\rightarrow$ 2 line Encoder:-



$$2^N = 4$$

D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

## Logical Expressions

$$A = D_2 + D_3$$

$$B = D_1 + D_3$$

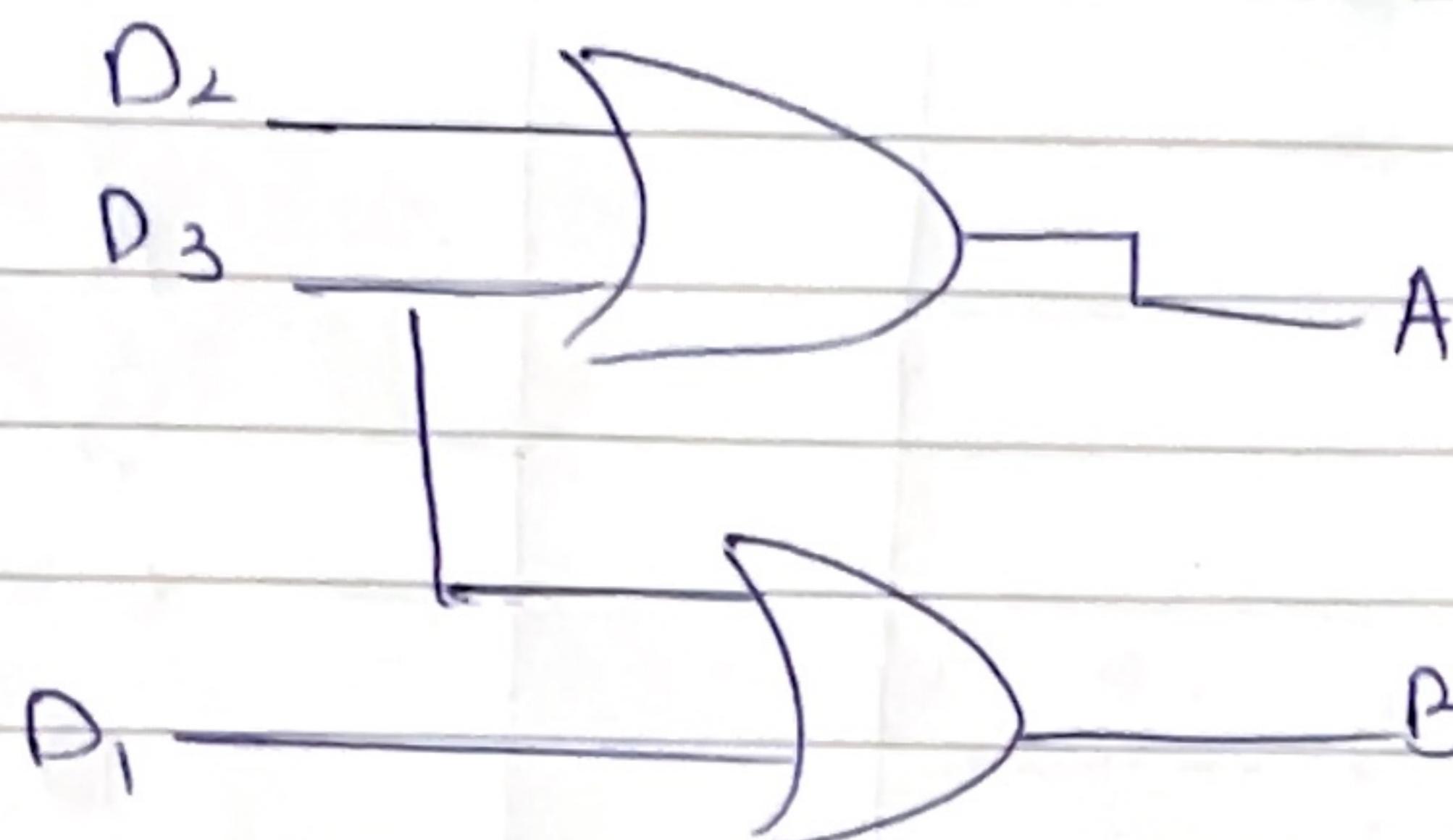
(Input at <sup>A</sup> which the outputs are high)

$\Rightarrow$  limitation of Encoder  
& priority Encoder!

Date \_\_\_\_\_

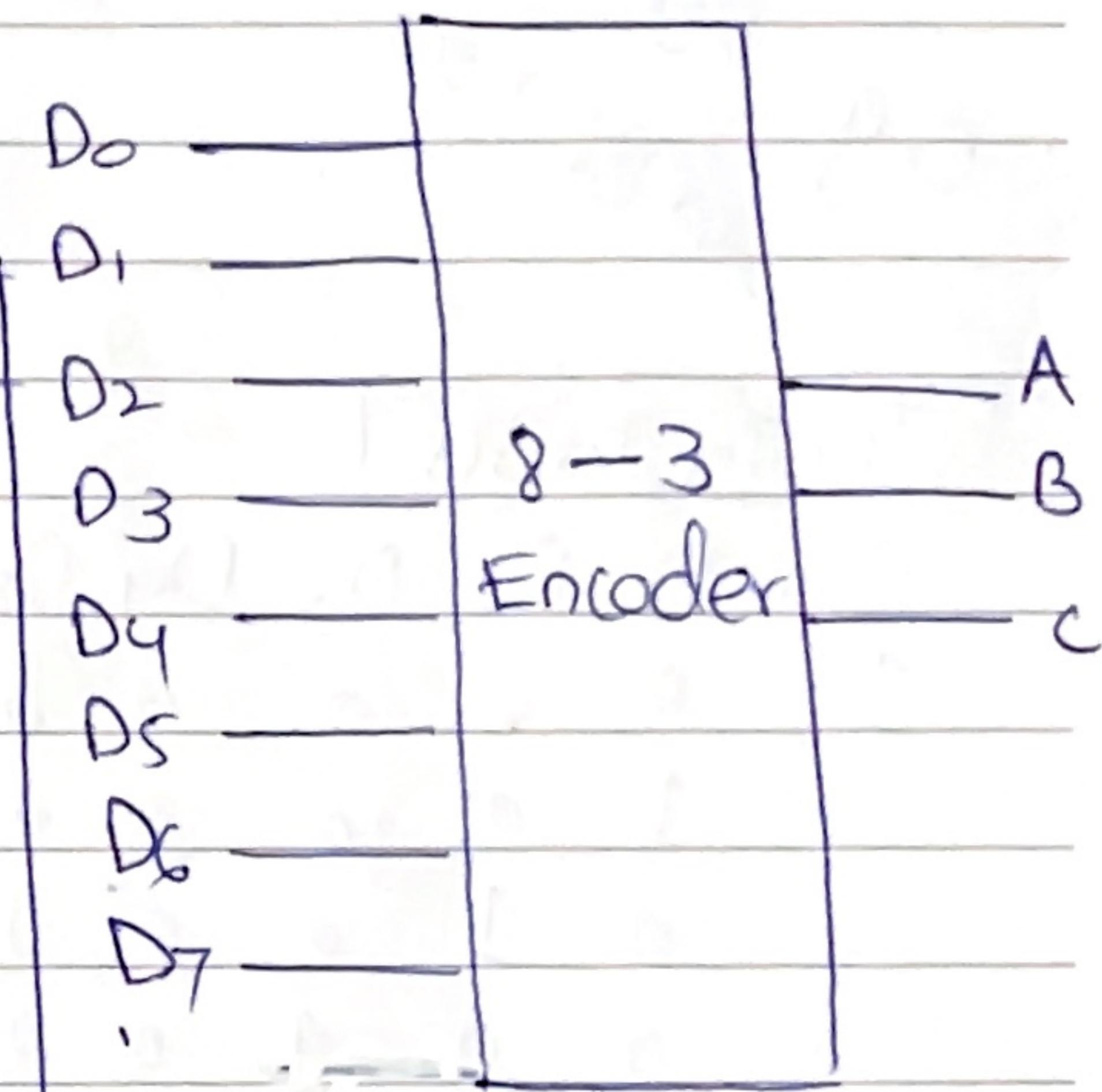


Logic Diagram.



8 line  $\rightarrow$  3 line Encoder.

D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0



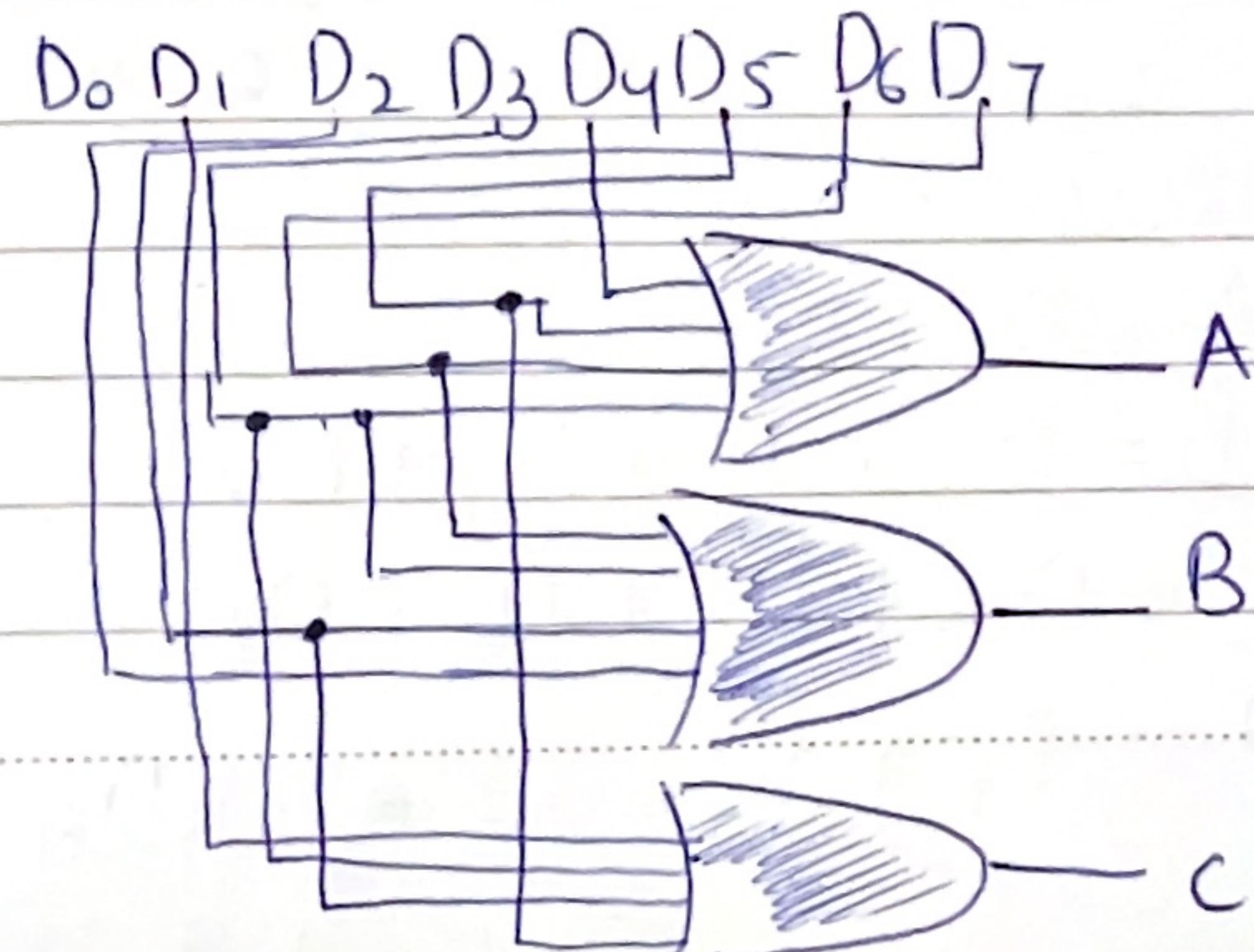
Boolean Expression

$$A = D_4 + D_5 + D_6 + D_7$$

$$B = D_2 + D_3 + D_6 + D_7$$

$$C = D_1 + D_3 + D_5 + D_7$$

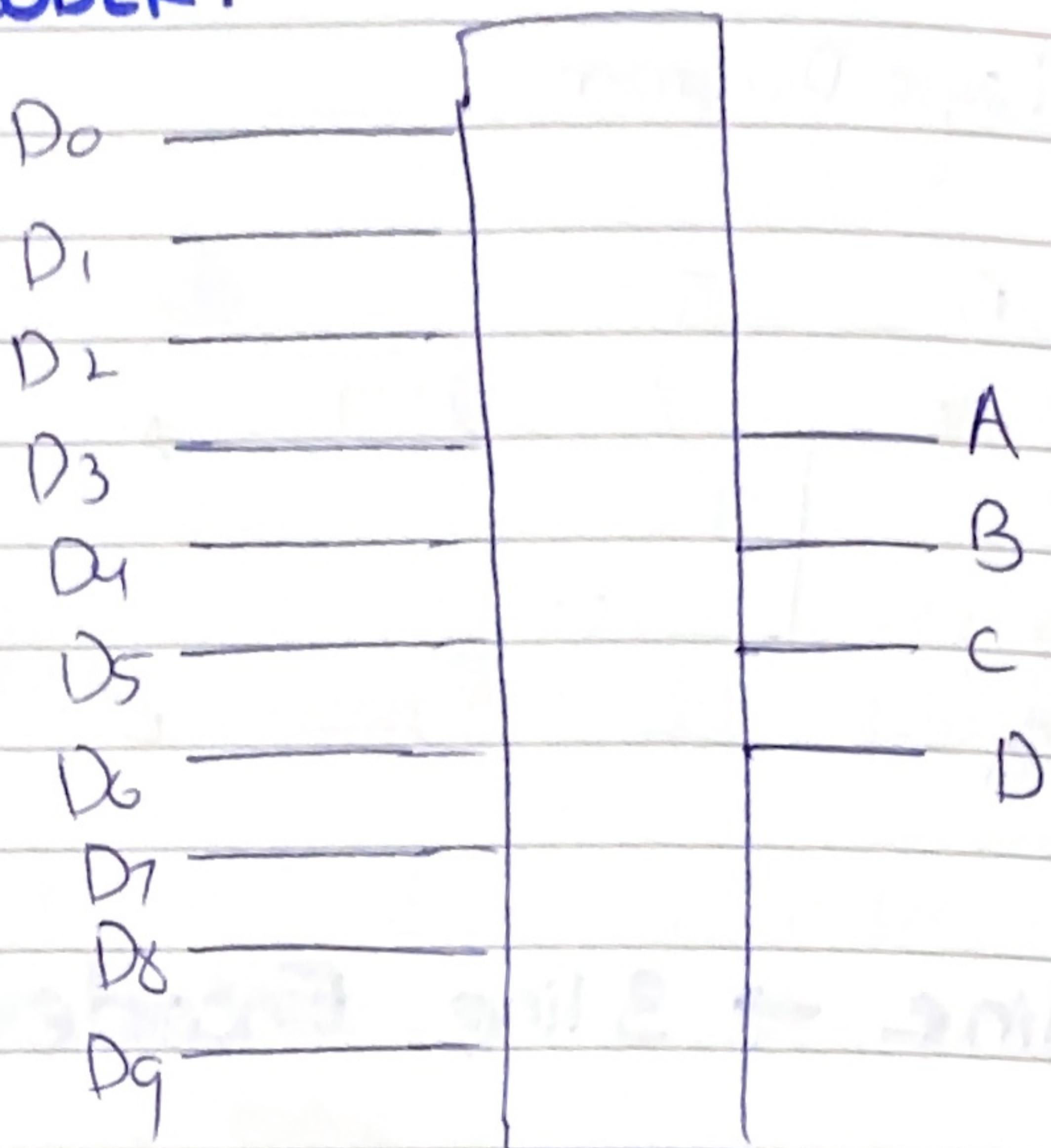
LOGIC DIAGRAM :-



Date

## DECIMAL TO BCD ENCODER.

each input in a decimal no. that is converted into BCD code.  
4 digit decimal 1 is coded e.g. in BCD as 0111



### TRUTH TABLE

D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>	D <sub>8</sub>	D <sub>9</sub>	A	B	C	D	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	
0	0	1	0	0	0	0	0	0	0	0	1	0		
0	0	0	1	0	0	0	0	0	0	0	1	1		
0	0	0	0	1	0	0	0	0	0	1	0	0		
0	0	0	0	0	1	0	0	0	0	1	0	1		
0	0	0	0	0	0	1	0	0	0	1	1	0		
0	0	0	0	0	0	0	1	0	0	1	1	1		
0	0	0	0	0	0	0	0	1	0	1	0	0		
0	0	0	0	0	0	0	0	0	1	1	0	0	1	

### BOOLEAN EXPRESSIONS:

$$A = D_8 + D_9$$

$$B = D_4 + D_5 + D_6 + D_7$$

$$C = D_2 + D_3 + D_6 + D_7$$

$$D = D_1 + D_3 + D_5 + D_7 + D_9$$

Similarly logic circuit can be drawn using the Boolean Expressions

ACTIVE HIGH LOGIC  
Date → depending upon the input combination, at a given time, only one of the output is high.

## DECODER:-

Decoder is a combinational circuit.

Example:-

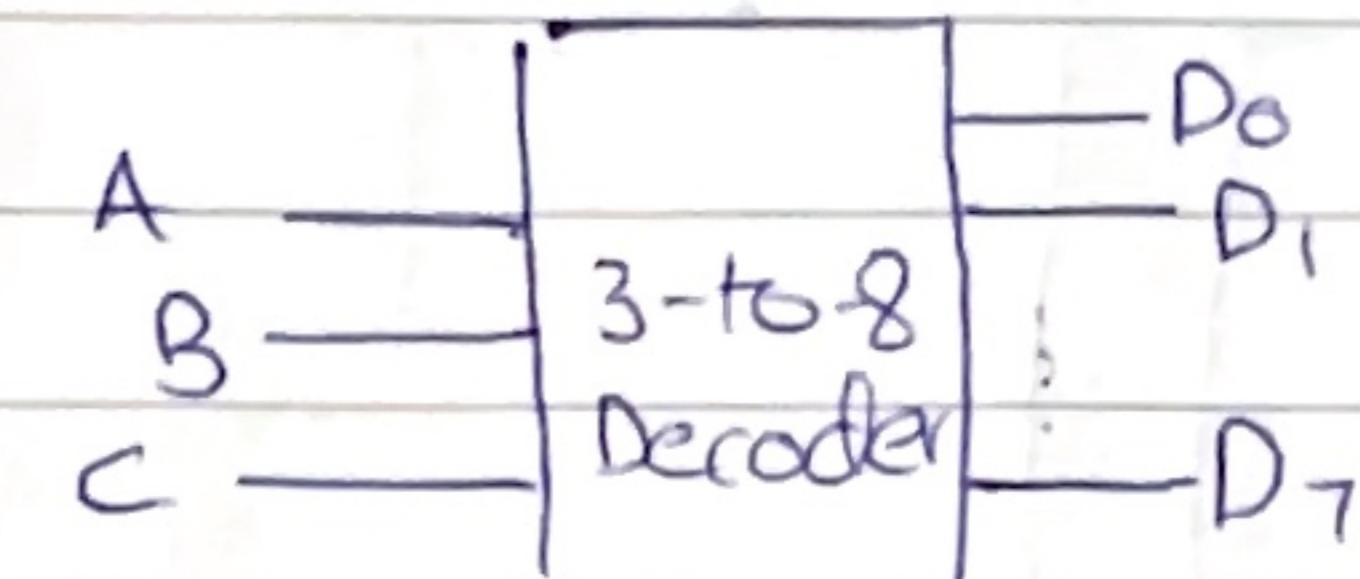
3 to 8 Decoder

4 to 16 Decoder

\* in some decoders the no. of output can be less than  $2^N$  e.g BCD to Decimal Decoder; here N inputs of  $2^N$  combinations have M outputs i.e  $M \leq 2^N$

### 3-to-8 Decoder:-

TRUTH TABLE



A	B	C	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
0	0	0	1							
0	0	1		1						
0	1	0			1					
0	1	1				1				
1	0	0					1			
1	0	1						1		
1	1	0							1	
1	1	1								1

### 2-to-4 Decoder:-

B <sub>1</sub>	B <sub>0</sub>	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

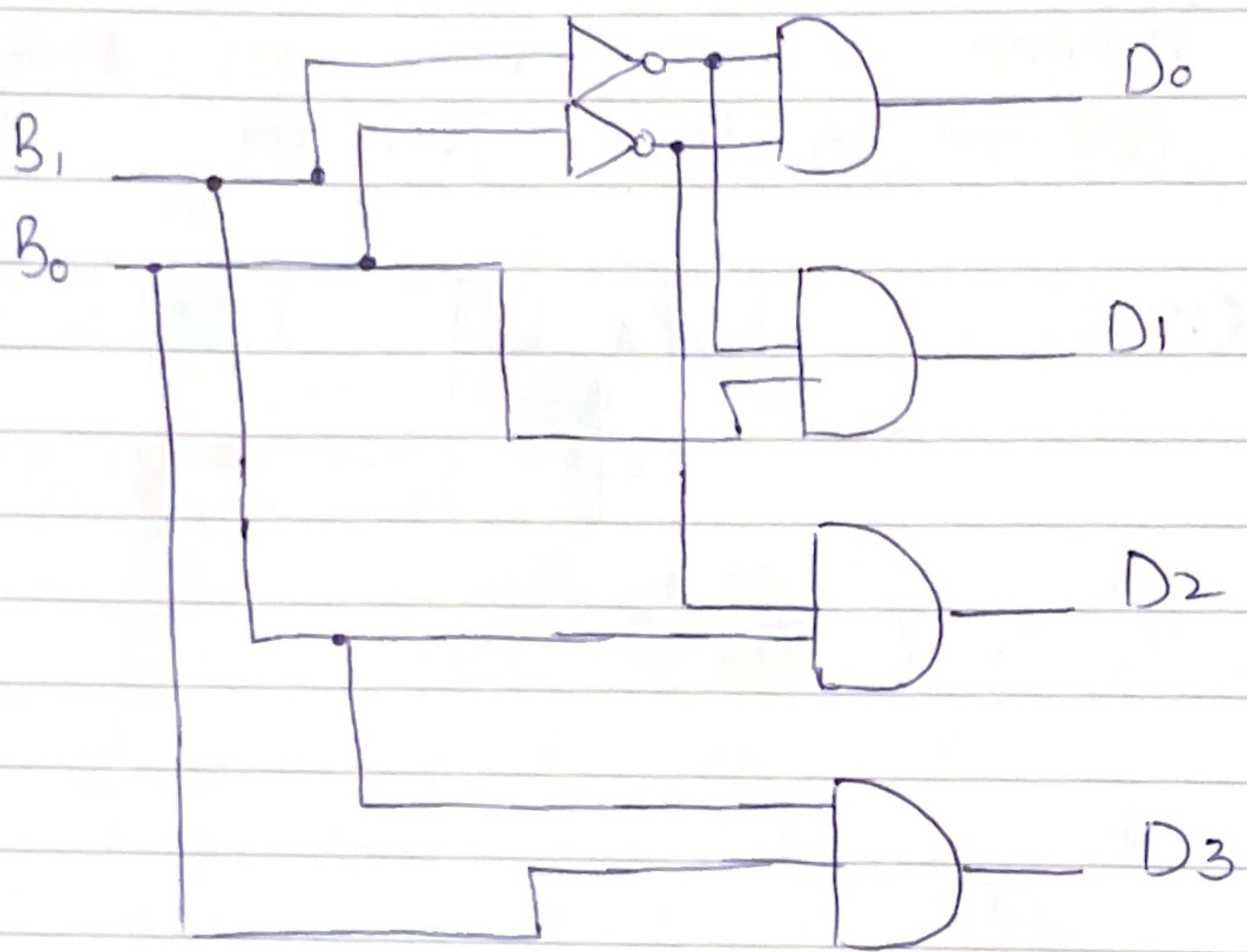
Date

- DECODER :-

## LOGIC CIRCUIT :-

while drawing the logic Circuit.

- (i) Take ones as it is;
- (ii) Take NOT gate on  $B_0$ ;



## CONSTRUCT A FULL ADDER CIRCUIT USING DECODER :-

=> Combine all the Outputs of the decoder for which the output of the logic circuit is high.

Step no. 1.

Draw Truth Table of Full adder.

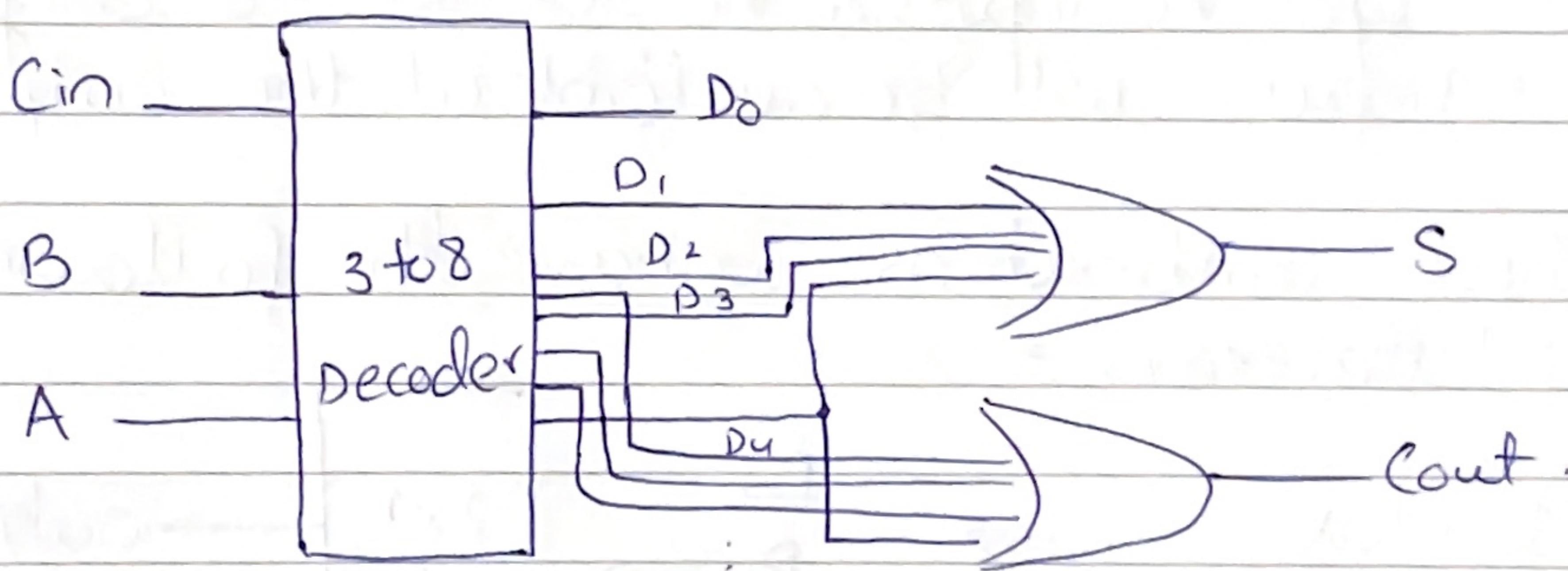
Date \_\_\_\_\_

- ADDITION

A	B	Cin	SUM	COUT	
0	0	0	0	0	D <sub>0</sub>
0	0	1	1	0	D <sub>1</sub>
0	1	0	1	0	D <sub>2</sub>
0	1	1	0	1	D <sub>3</sub>
1	0	0	1	0	D <sub>4</sub>
1	0	1	0	1	D <sub>5</sub>
1	1	0	0	1	D <sub>6</sub>
1	1	1	1	1	D <sub>7</sub>

Step no. 2.

Combine the terms using OR-gate for which both SUM & COUT are high respectively.

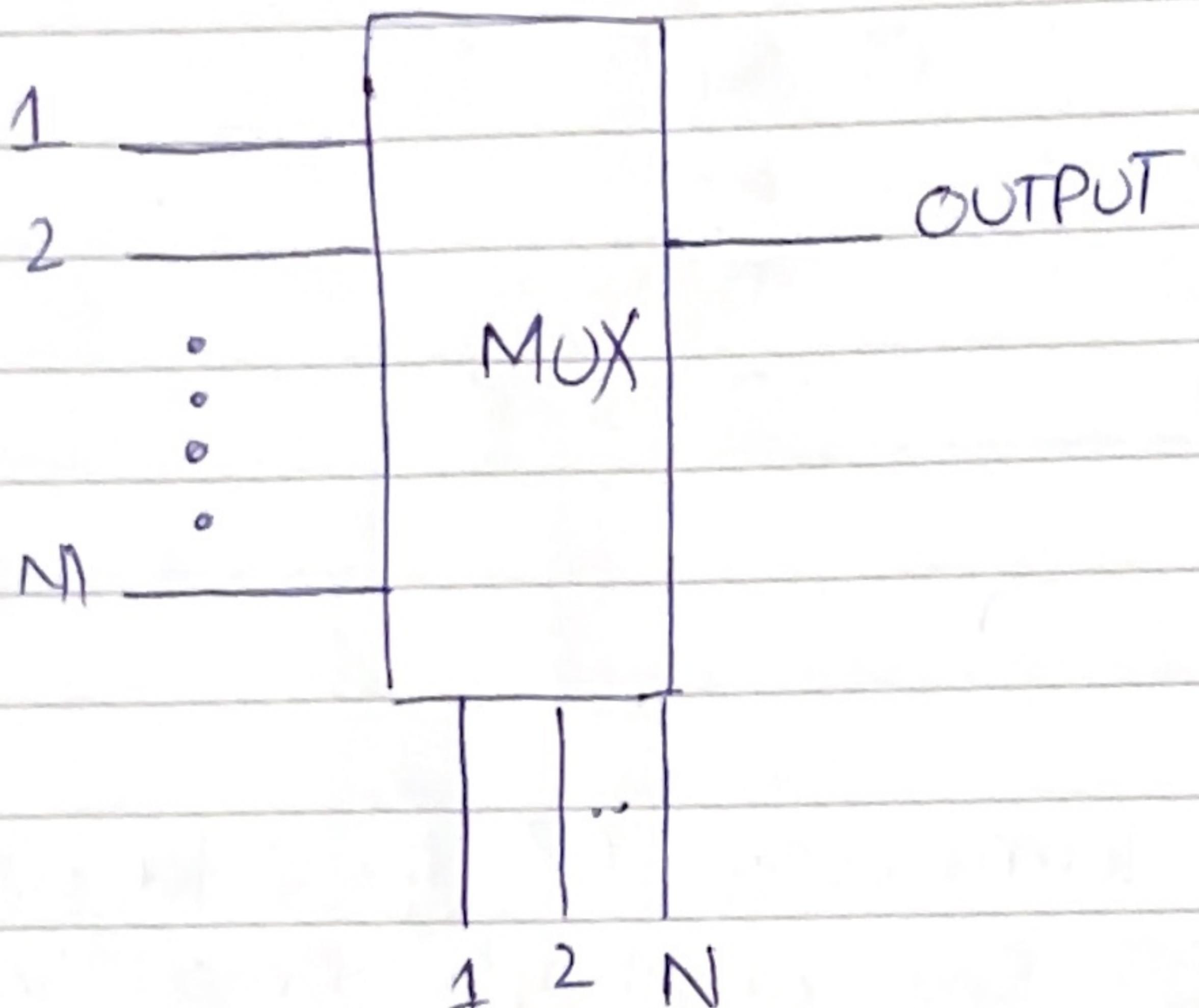


Date

## MULTIPLEXER:-

A type of Combinational Circuit that has  $M$  inputs and only 1 output.

It also has  $N$ . no. of selection lines.



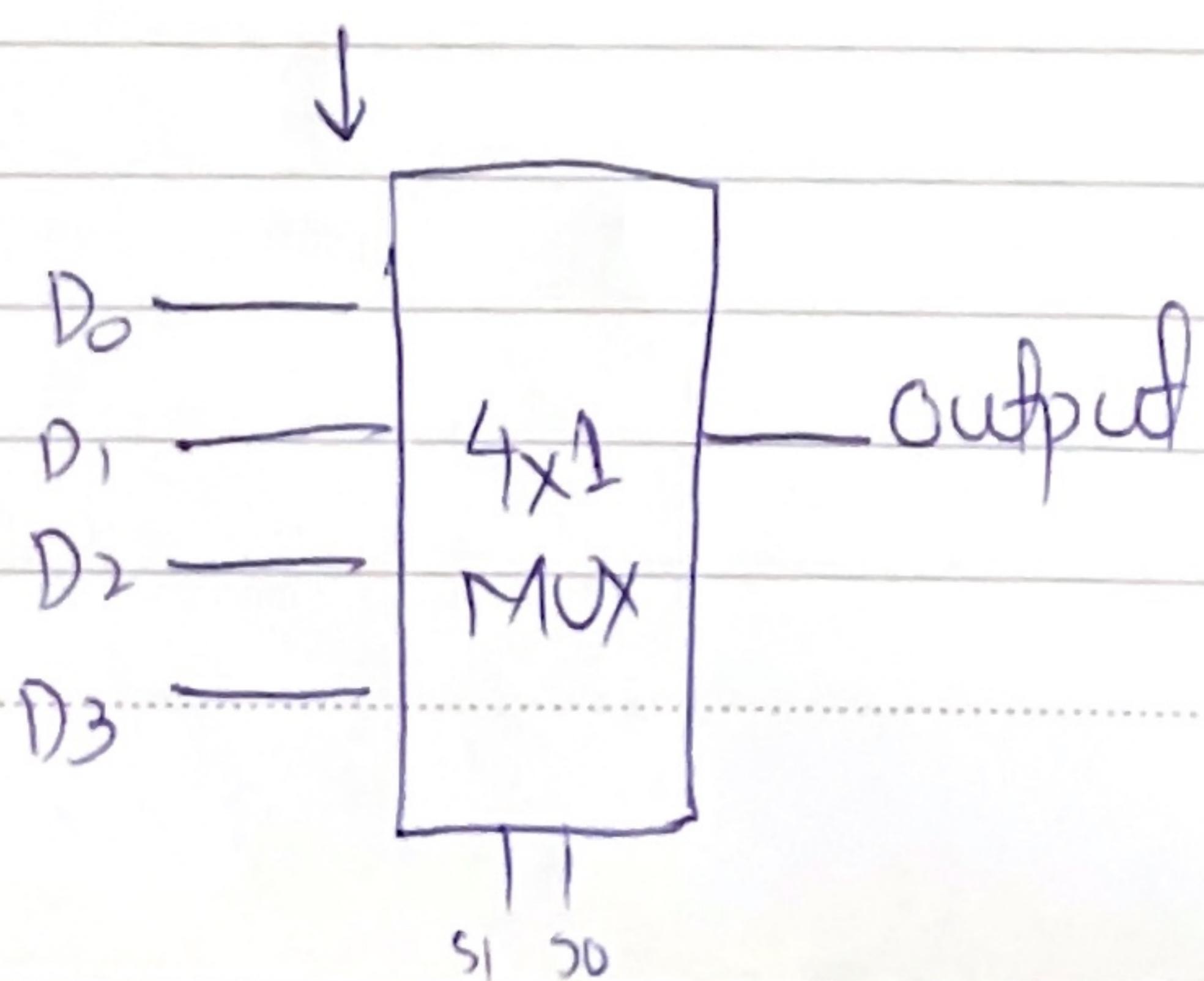
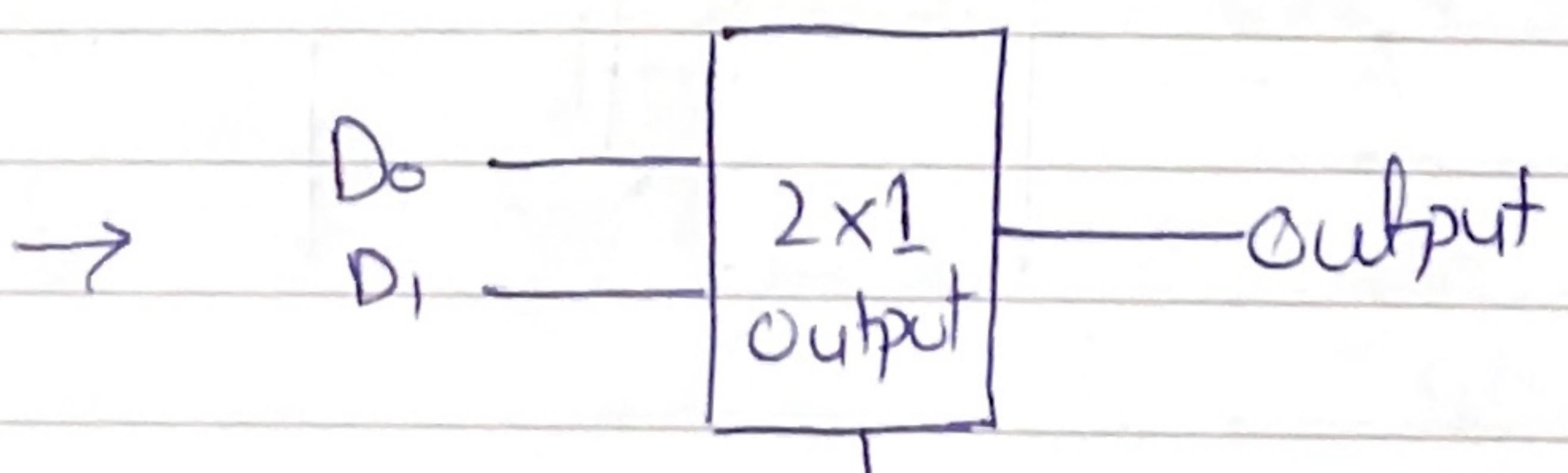
Where  $M = 2^N$  <sup>input</sup>  $\rightarrow$  <sup>no. of select lines</sup>

$$N = \log_2(M).$$

& Depending upon the inputs at the selection line only one of the  $M$  inputs will be available at the output.

Due to these combinations we have the following types of Multiplexers:-

- (i)  $2 \times 1$  MUX
- (ii)  $4 \times 1$  MUX



Date

## 2x1 MULTIPLEXER:-

selections

4x1  $\rightarrow$  Multiplexer.

8x1  $\rightarrow$  Multiplexer

16x1  $\rightarrow$  "

DEMUX

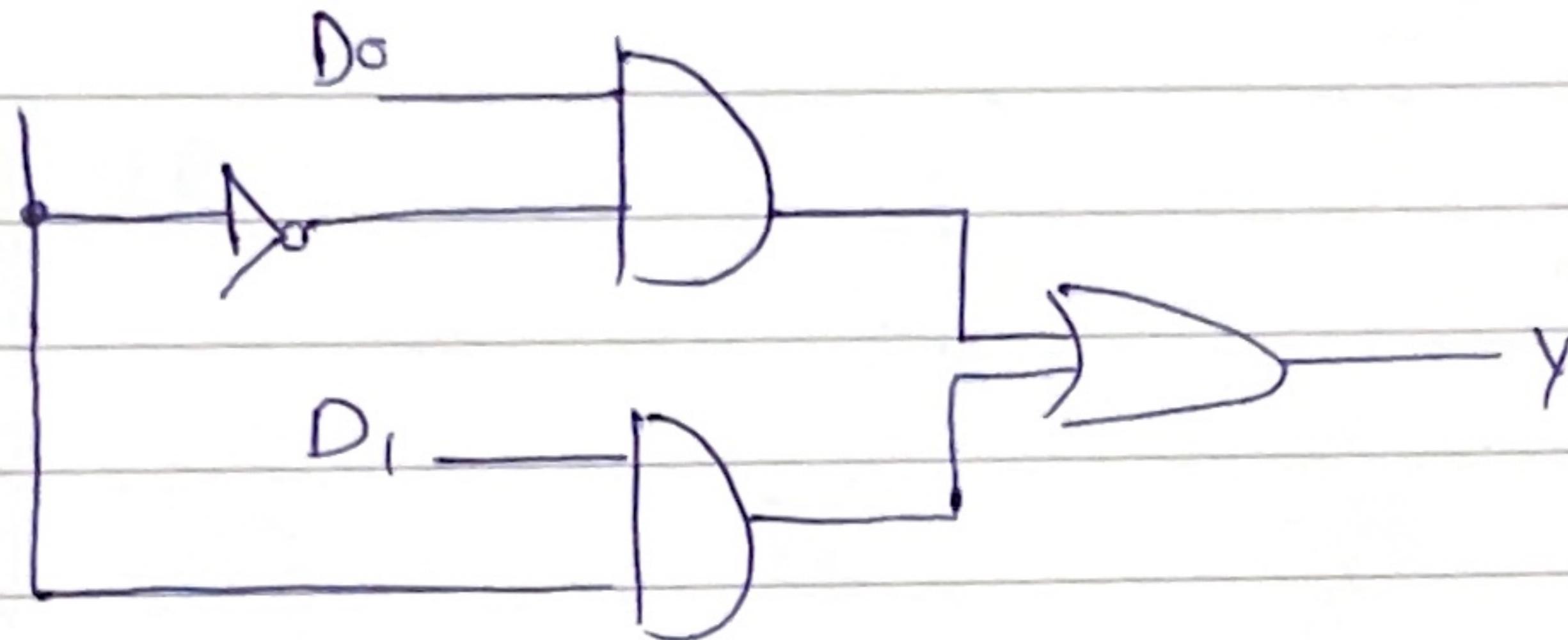
D <sub>1</sub>	D <sub>0</sub>	S	Y	Y
0	0	0	0	D <sub>0</sub>
0	0	1	0	D <sub>1</sub>
0	1	0	1	D <sub>0</sub>
0	1	1	0	D <sub>1</sub>
1	0	0	0	D <sub>0</sub>
1	0	1	1	D <sub>1</sub>
1	1	0	1	D <sub>0</sub>
1	1	1	1	D <sub>1</sub>

$\Rightarrow$

S	Y
0	D <sub>0</sub>
1	D <sub>1</sub>

simplified!

$$\therefore Y = \bar{S}D_0 + SD_1.$$



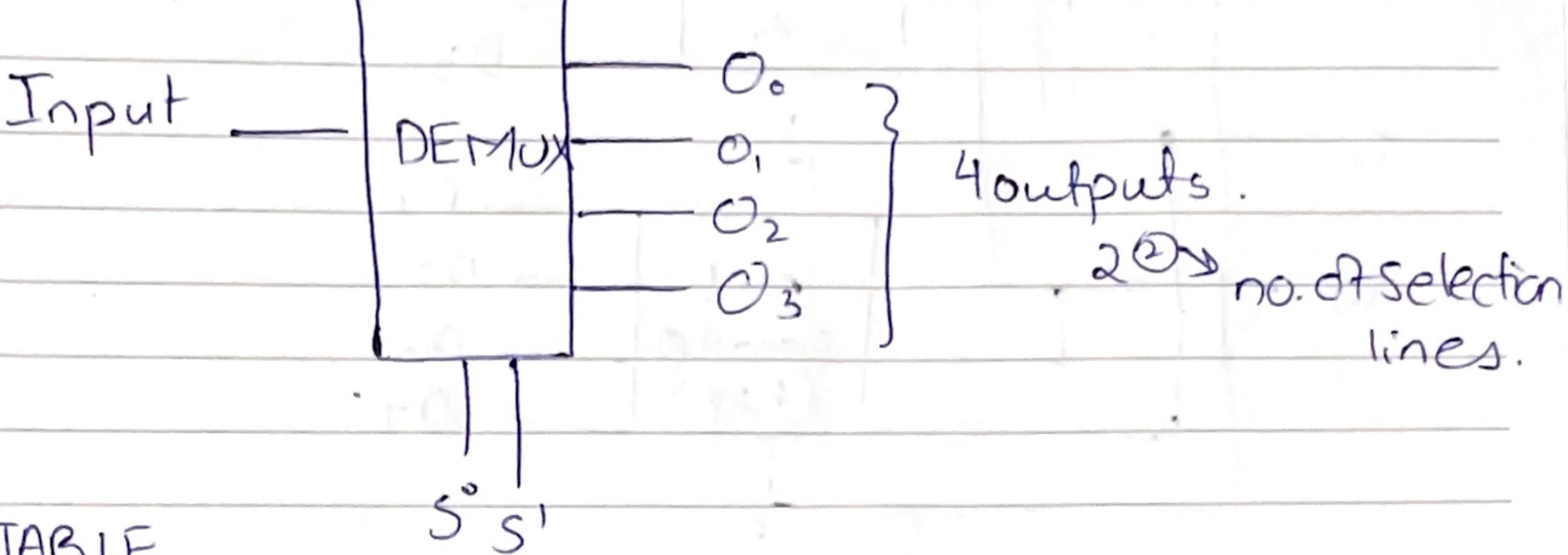
- case study included  
Date - 4/15 input K-map.  
Application:-

- learn to analyze combination  
of MUX / DEMUX.

## DEMUX :-

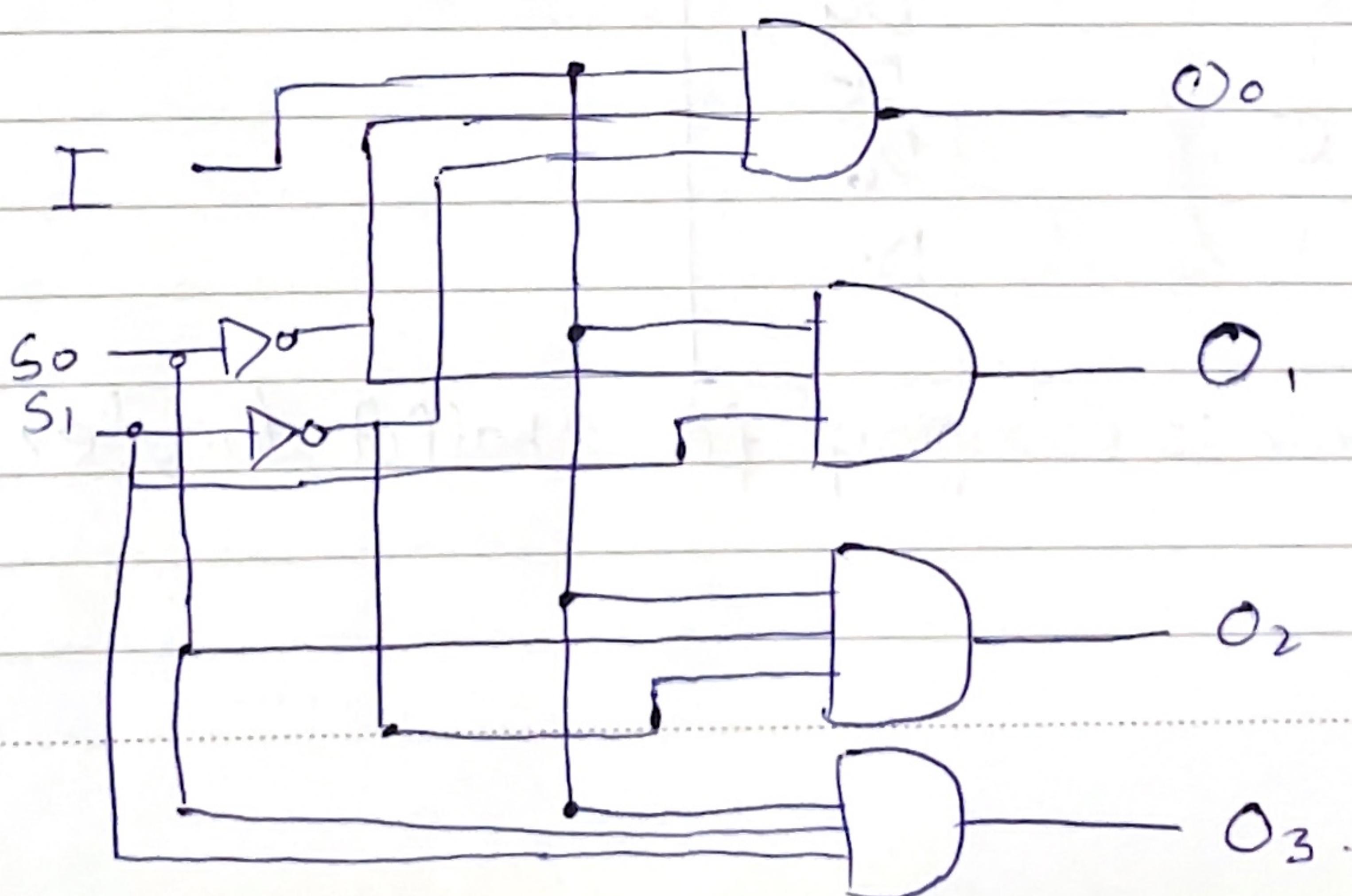
A type of combinational circuit that has 1 input and M outputs.

De-  
**1x4 Multiplexer.**



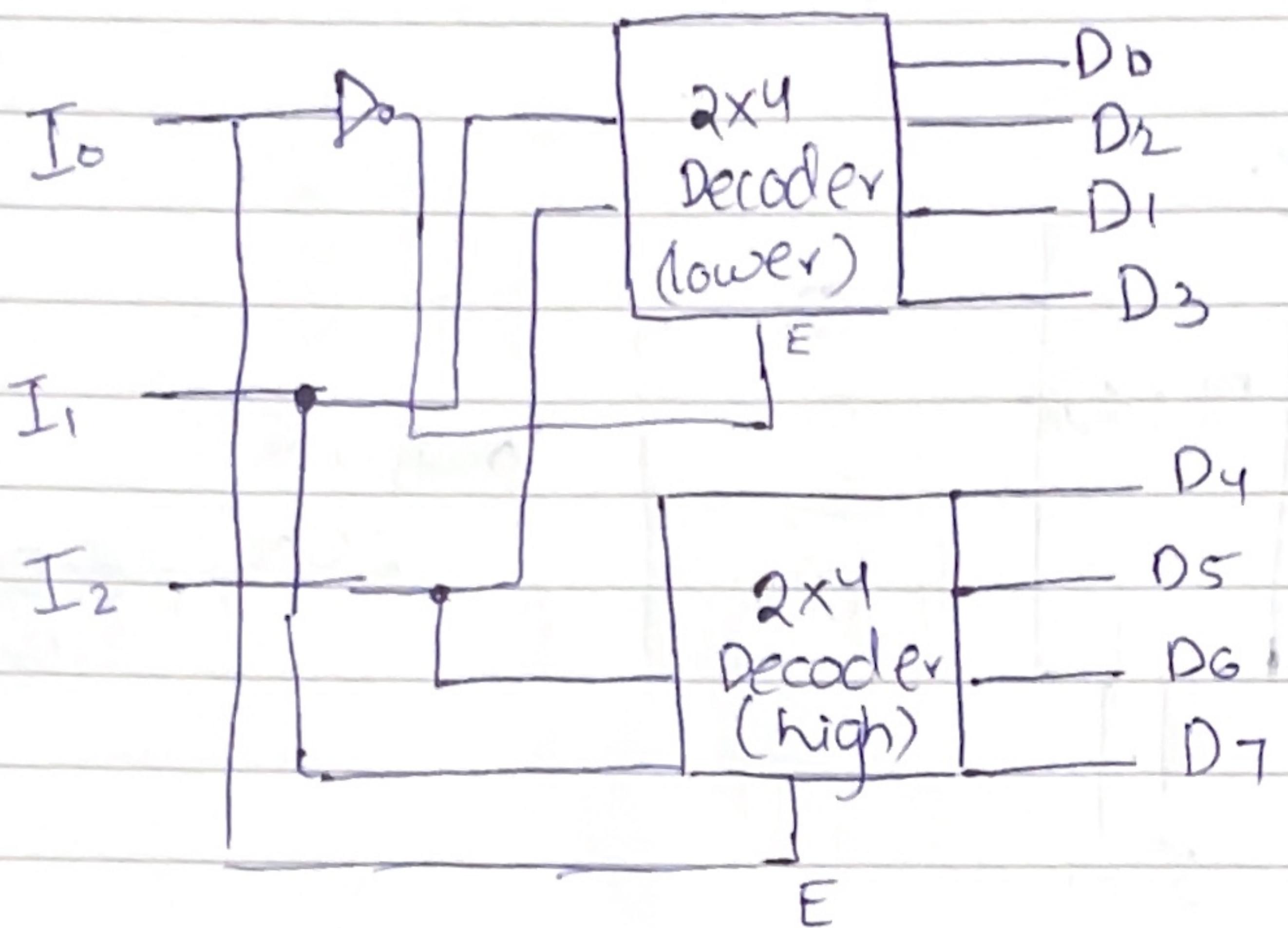
### TRUTH TABLE

$S_0$	$S_1$	
0	0	$O_0$
0	1	$O_1$
1	0	$O_2$
1	1	$O_3$



Date

## 3x8 Decoder using 2x4 Decoder.

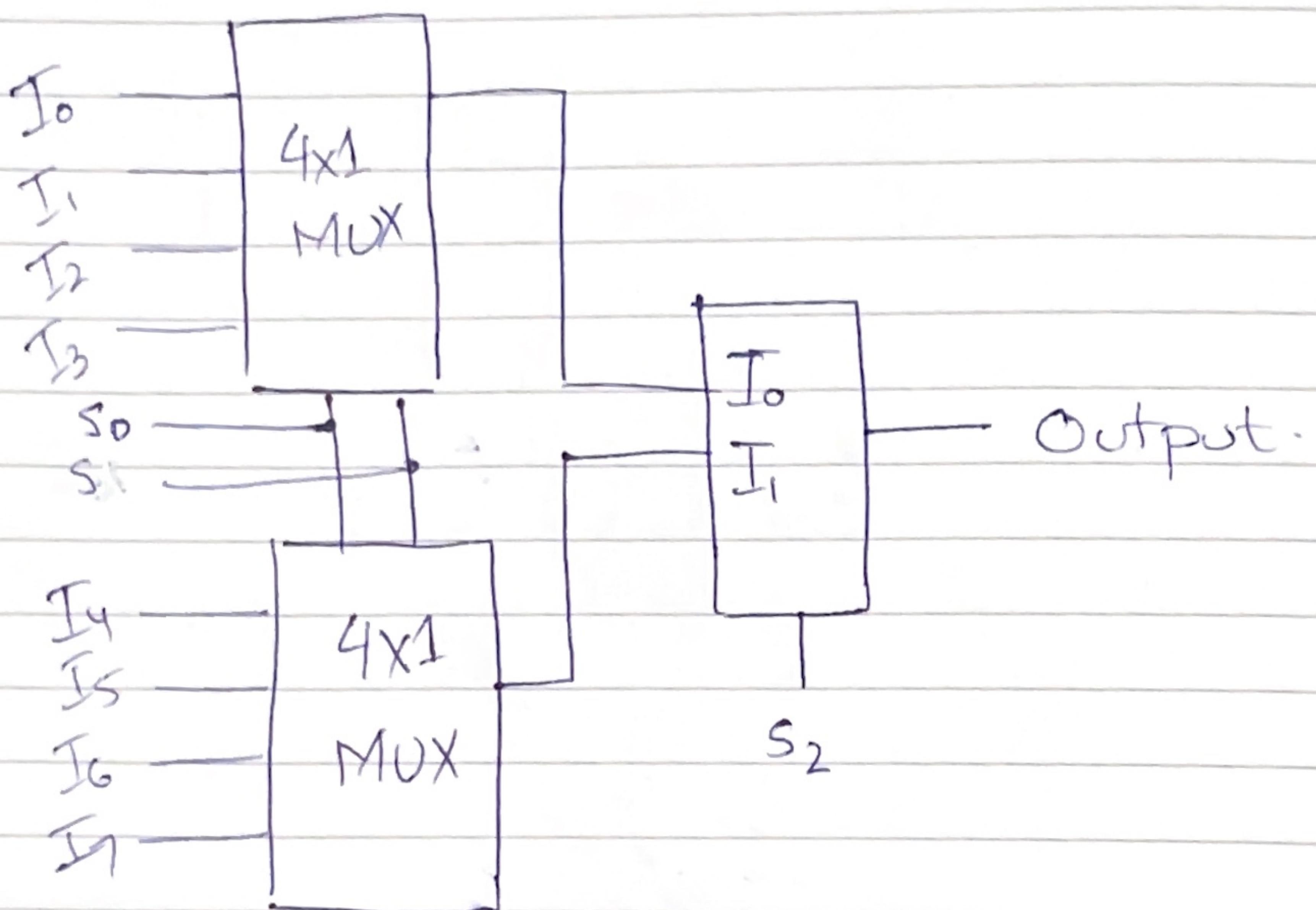


$I_0$	$I_1$	$I_2$	Outputs
0	0	0	$D_0$
0	0	1	$D_1$
0	1	0	$D_2$
0	1	1	$D_3$
1	0	0	$D_4$
1	0	1	$D_5$
1	1	0	$D_6$
1	1	1	$D_7$

only this pair is changing for 2 half of encoder

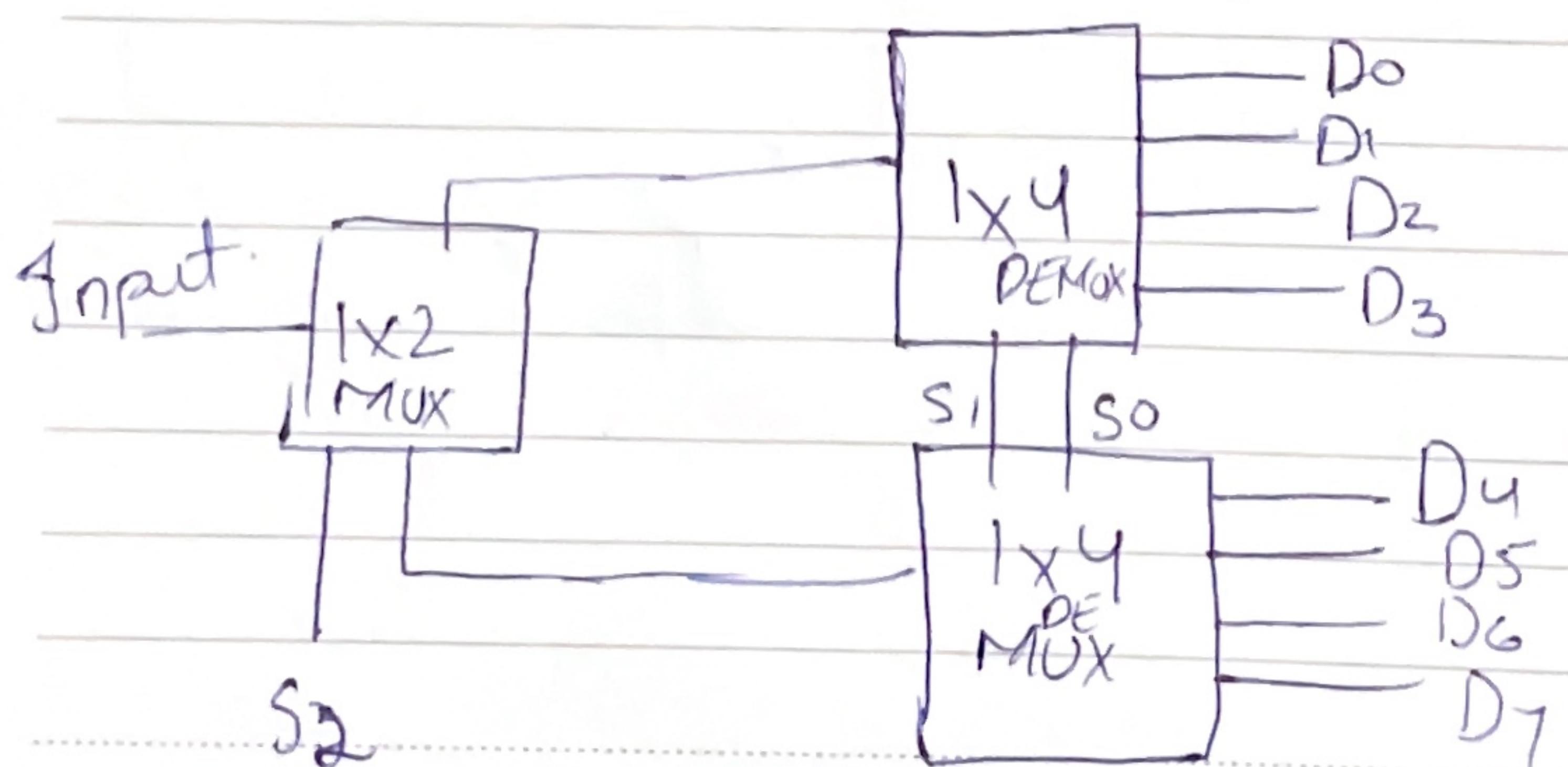
Date

## DESIGN 8x1 MUX USING 4x1 MUX.



Now the values of  $S_1, S_2, S_3$  will decide that which output will come.

## Design 1x8 DEMUX using 1x4 DEMUX:-



no.5.

Output		X	
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$X = \overline{AB} + \overline{A}\overline{B}$

$X = A \oplus B$

Bi-conditional - Exclusive NOR

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

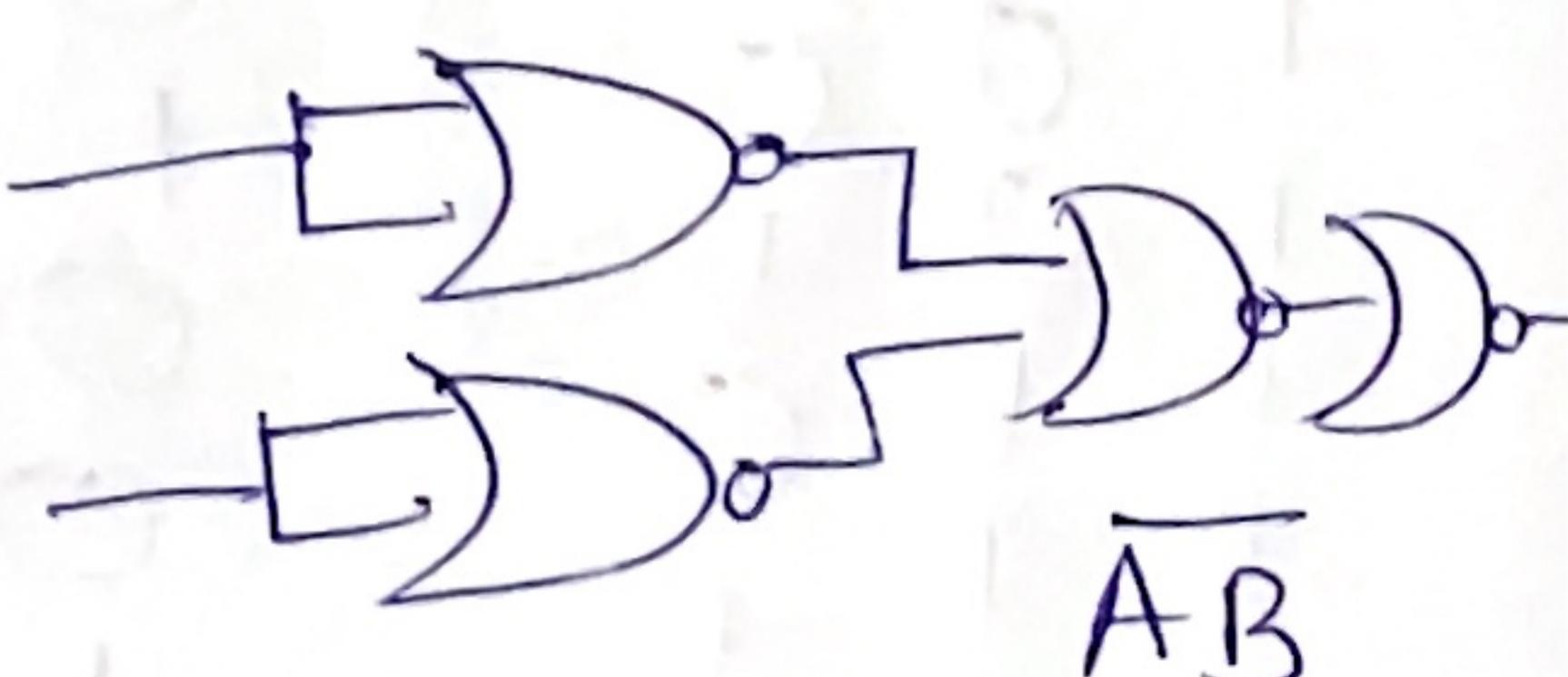
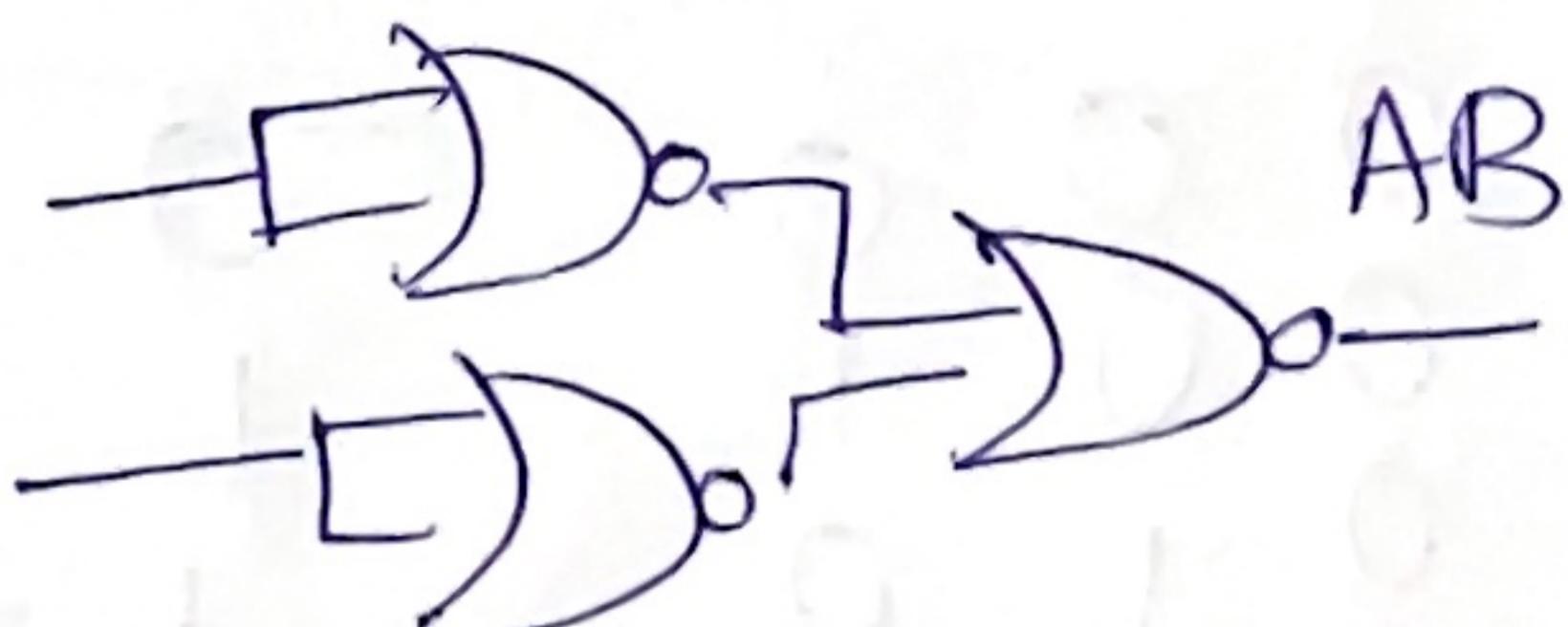
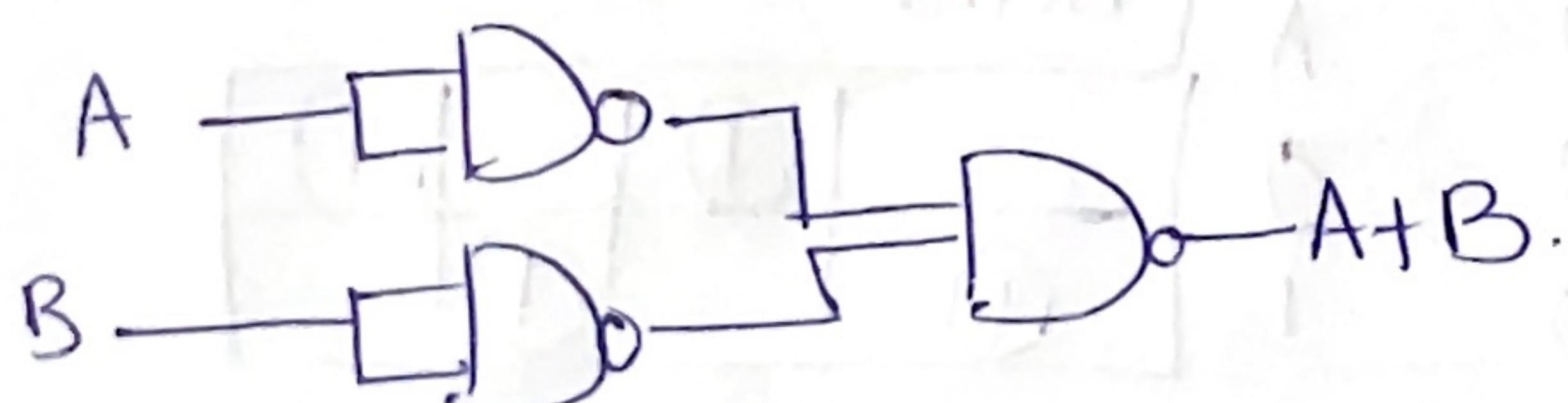
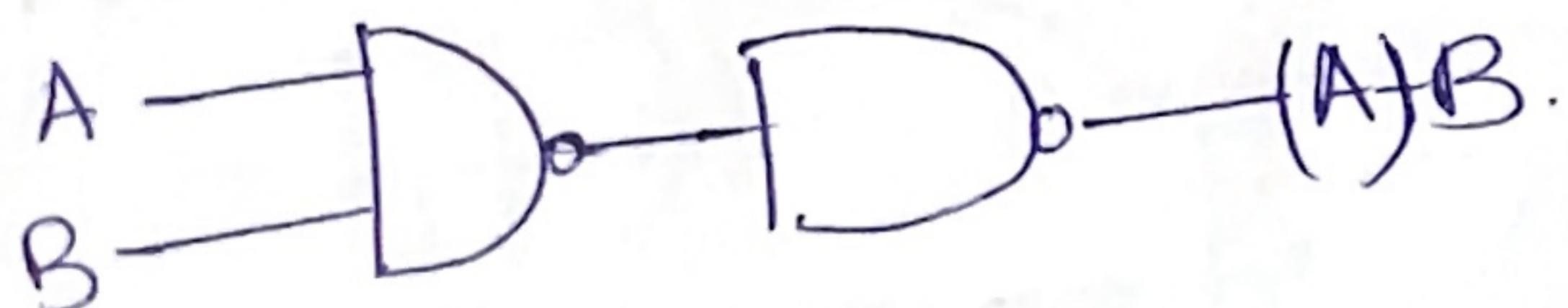
$X = \overline{\overline{AB} + AB}$

$X = (\overline{A} \oplus B)$

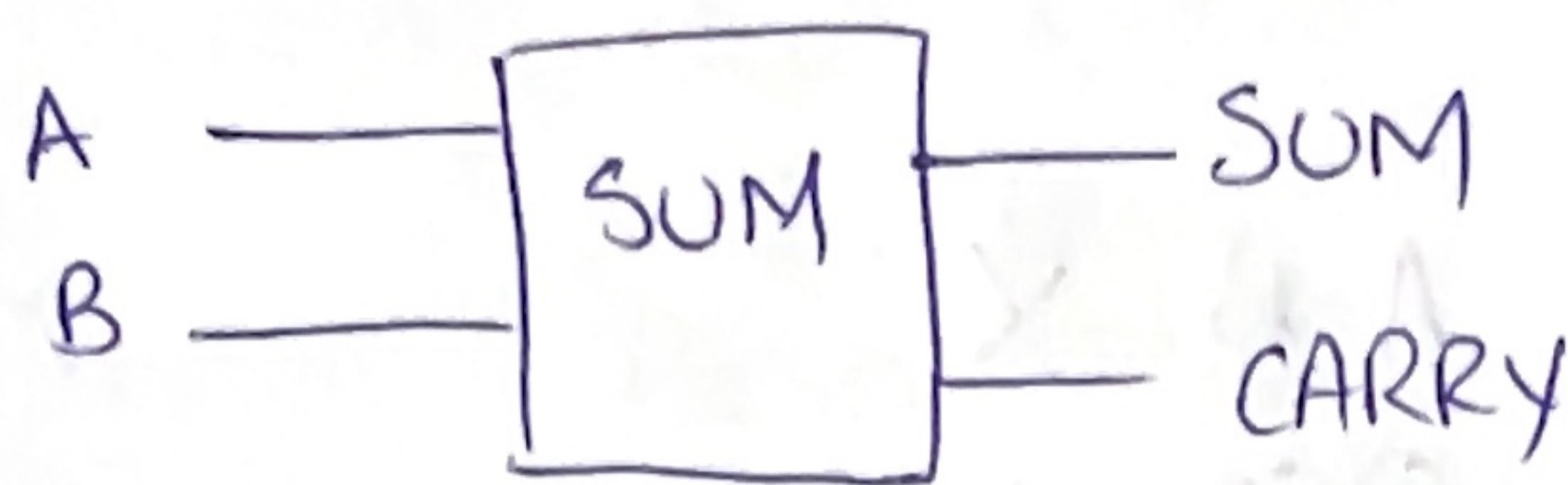
By Kmap we will derive the above given equations.  
(imp for simplification often)

$\sim X \sim$

Any gate can be implemented using NAND & NOR



# HALF ADDER & FULL ADDER.



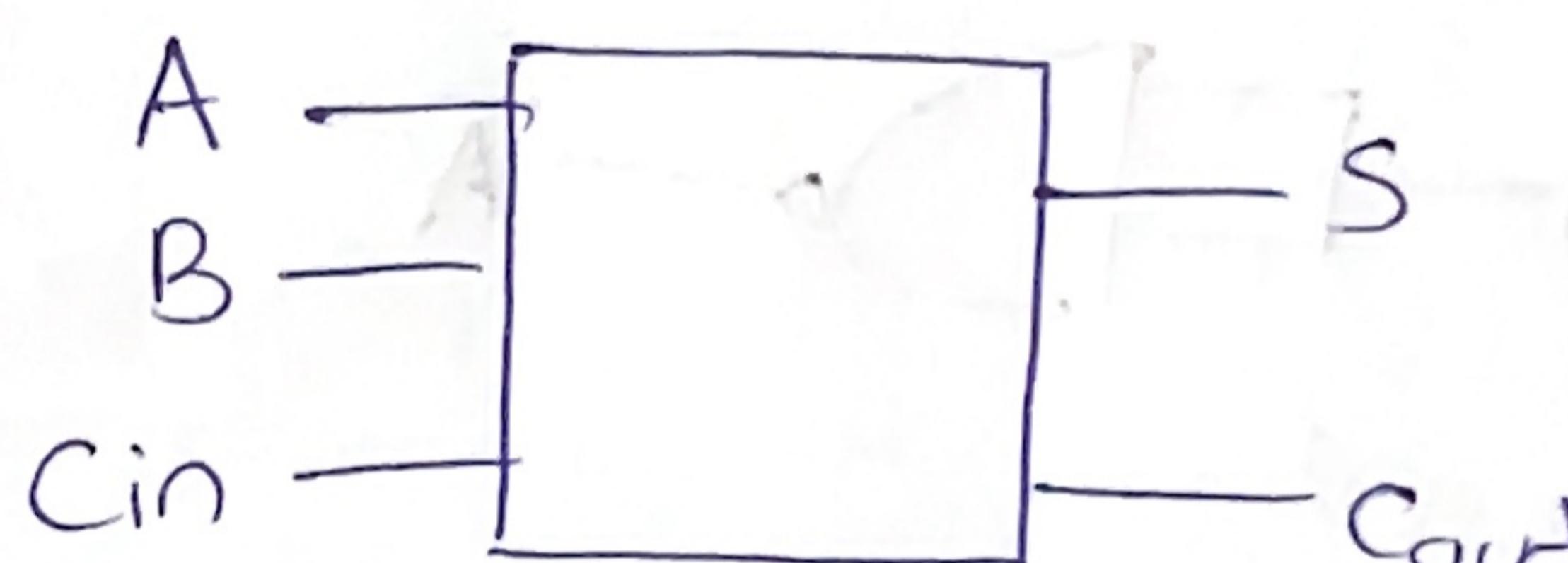
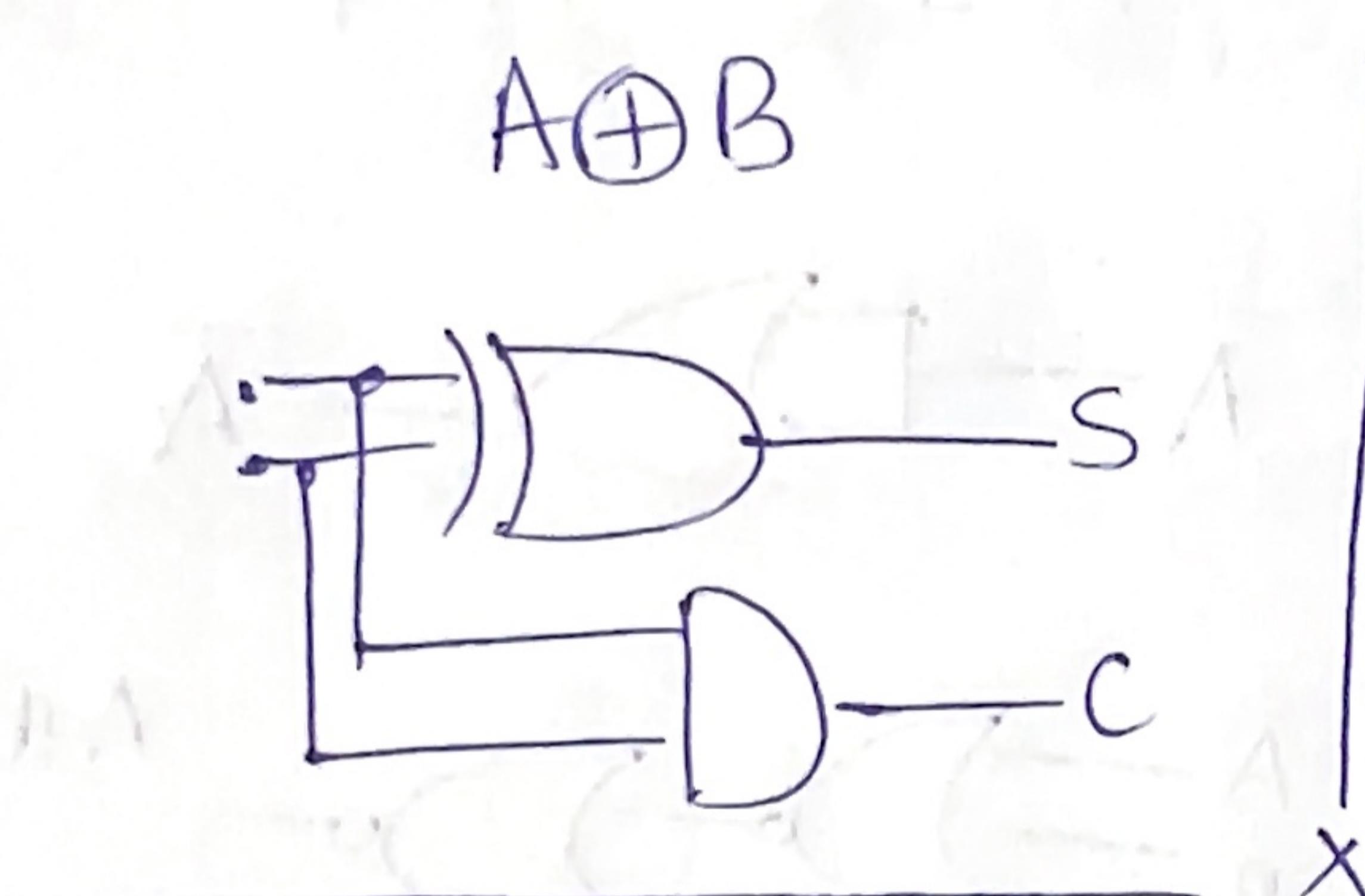
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Draw the equations of SUM & CARRY.

A \ B	0	1
0	0	1
1	1	0

Carry	A	B
0	0	1
1	0	0

$$\text{SUM} \Rightarrow \bar{A}\bar{B} + \bar{B}\bar{A}$$



A	B	Cin	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$A \oplus B \oplus \text{Cin}$

S	00	01	11	10
A	0	0	1	1
B	0	1	0	1
Cin	0	1	0	1

$$S = \bar{A}\bar{B}\text{Cin} + \bar{A}\bar{B}\bar{C}$$

$$\bar{A}\bar{B}\bar{\text{Cin}} + ABC$$

$$\Rightarrow \bar{A}(\bar{B}\text{Cin} + \bar{B}\text{Cin}) + A(\bar{B}\bar{\text{Cin}} + B\text{Cin})$$

$$\Rightarrow \bar{A}(A \oplus \text{Cin}) + A(B \oplus \text{Cin})$$

$B_{Cin}$

00	01	11	10
1	1	1	1

$$C_{out} = AC_{in} + BC_{in} + AB$$

or by min terms.

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} \leftarrow (a)$$

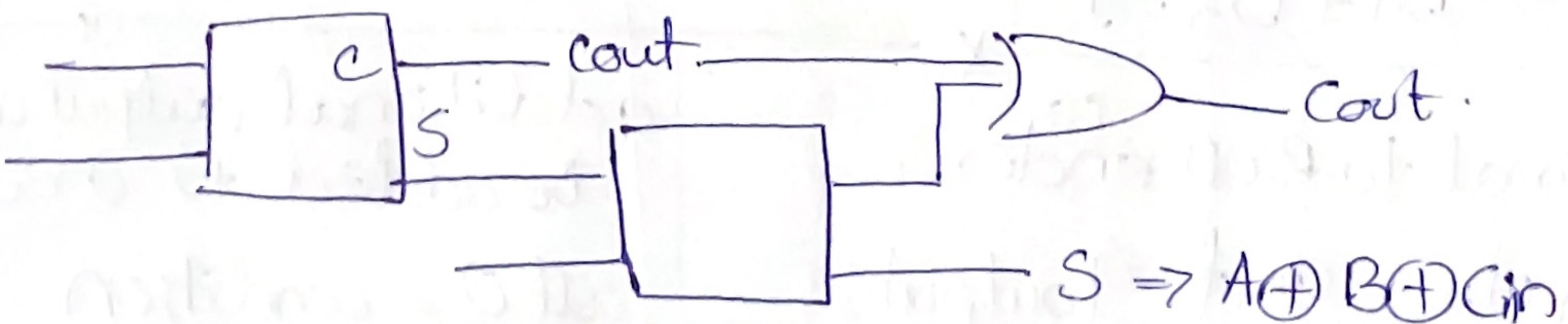
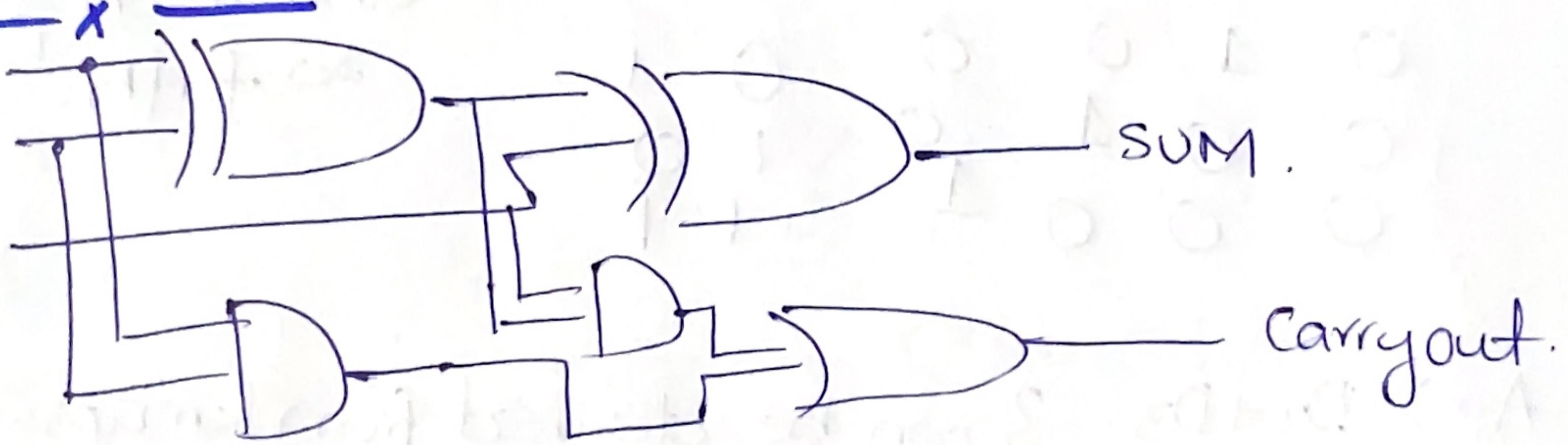
$$+ AB\bar{C}_{in} + ABC_{in}.$$

$\Rightarrow$

$$AB(\bar{C}_{in} + C_{in}) + C_{in}(\bar{A}B + \bar{A}\bar{B})$$

$$\Rightarrow AB + C_{in}(A \oplus B) \leftarrow (b).$$

## LOGICAL CIRCUITS



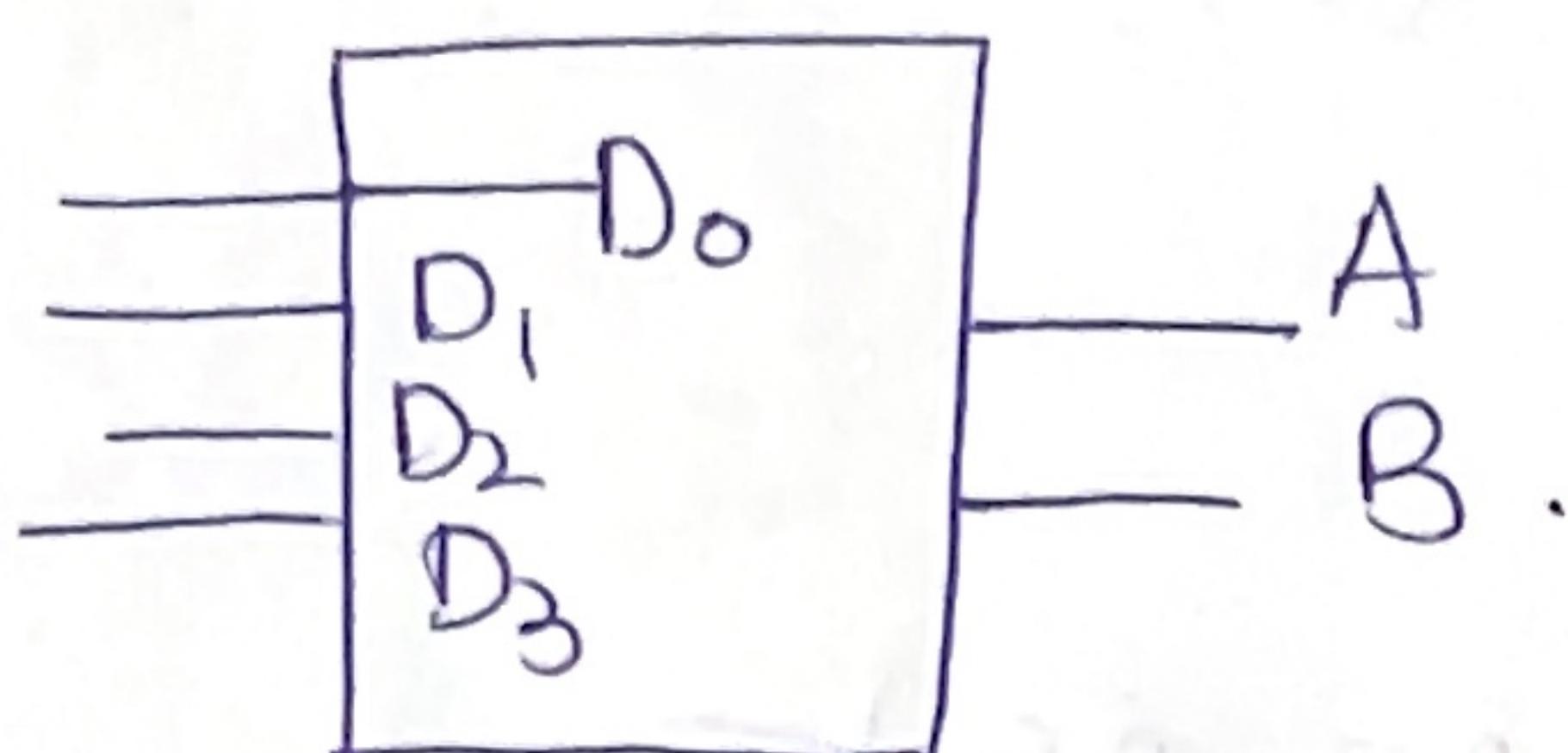
# ENCODER

N outputs and  $2^N$  inputs (reduces the selection)

Reverse of Decoder.

At an encoder at the particular moment only one input is high.

Example: 4 line to 2 line Encoder



- \* Outputs are normal 1T patterns.
- \* Inputs are high.

Input	Output	
D0 D1 D2 D3	A B	when D0 is high
1 0 0 0	0 0	the output is 00
0 1 0 0	0 1	and so on &
0 0 1 0	1 0	so forth.
0 0 0 1	1 1	

$$A \Rightarrow D_2 + D_3 \quad ? \text{ can be derived from kmaps.}$$

$$B \Rightarrow D_1 + D_3.$$

Decimal to BCD encoder.

8 inputs and 4 outputs.

additional output could be added to check all 0s condition.

given the input combination only 1 output is high, we add additional Enable input to keep the decoder is ON.



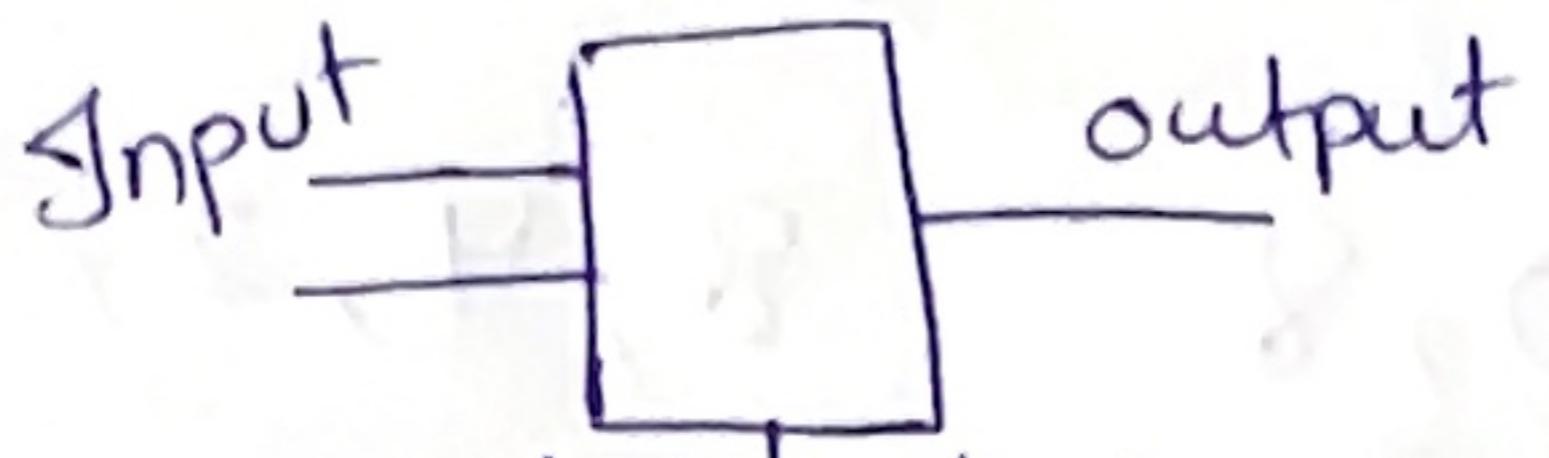
## MUX.

Decoder  
demUX-  
(similar to decoder)

$N$  selection lines } Relationship b/w selection line  
and  $2^N$  inputs } and inputs

\* So depending upon the selection line combination  
only one input is present at the output

$2 \times 1$  MUX.



$D_1 \quad D_0 \quad S .$

0 0 0

0 0 1

0 1 0

0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

$D_0 \Rightarrow 0$

$D_1 \Rightarrow 0$

$D_0 \Rightarrow 1$

$D_1 \Rightarrow 0$

$D_0 \Rightarrow 0$

$D_1 \Rightarrow 1$

$D_0 \Rightarrow 1$

$D_1 \Rightarrow 1$

simplified!

S	Y
0	$D_0$
1	$D_1$

$$Y = \bar{S}D_0 + SD_1$$

$4 \times 1$

simplified.

$D_3 \quad D_2 \quad D_1 \quad D_0$

$S_1, S_0$	Output
0 0	$D_0$
0 1	$D_1$
1 0	$D_2$
1 1	$D_3$

$$Y = \bar{S}_1 \bar{S}_0 D_0 + \bar{S}_1 S_0 D_1$$

$$+ S_1 \bar{S}_0 D_2 + S_1 S_0 D_3$$

For an 8 Bit BCD number  
the column weights are:

80, 40, 20, 10, 8, 4, 2, 1

largest possible no.  $\Rightarrow (2^{(k-1)} - 1)$   
Lowest possible no.  $\Rightarrow (2^{(k-1)})$

e.g.  $(\frac{1000}{8000}, \frac{0011}{4000}, \frac{0101}{2000}, \frac{1001}{1000})$  BCD.  
 $300, 100, 200, 100, 80 \rightarrow 40, 20, 10, 8, 4, 2, 1$

Now add column weights where there was 1.

$$8000 + 200 + 100 + 40 + 10 + 8 + 1 \\ \Rightarrow (8359)_{10}$$

Gray Code has a single bit change.

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

Binary  $\rightarrow$  BCD

Binary  $\rightarrow$  Gray code.