chapter 3

Combinational circuit.

DATE _____ PAGE _____

Combinational circuit is a circuit which consist of logic gate whose outputs at any instant of time are determined directly from the present combination of input without regard to previous input.

~~There will be two combination of inputs~~

↓ The combinational circuit doesnot use any member.

The procedure involves the following steps:

- The problem is stated.
- The number of available input variables and required output variable is determined.
- The input and output variable are assigned their letter symbol.
- The truth table that defines the required relationship between the input and the output is derived.
- The simplified Boolean function for each output is obtained.
- The logic ~~dia~~ diagram is drawn.

Adder:

Adder are the combinational circuit which is used to add two or more than ~~bit~~ two bit at a time.

Type of adder

- Half adder.
- full adder.

DATE _____ PAGE _____

1. Half Adder

A combinational circuit that perform the addition of bit is called half adder. The circuit needs two binary input and two binary outputs. The input variable designate the augend 'A' and added 'B' bits; the output variable produce the sum (S) and carry (C).

Truth table to identify the function of half adder:

INPut		Output	
A	B	Sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

K-map:

for sum

A \ B	0	1
0	0	(1)
1	(1)	0

for carry

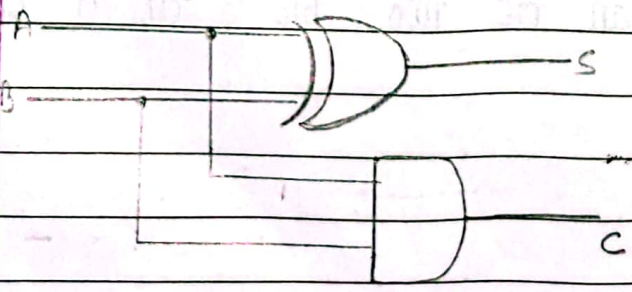
A \ B	0	1
0	0	0
1	0	1

from K-map the logic expression from sum & carry:

$C = AB$

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

logic diagram

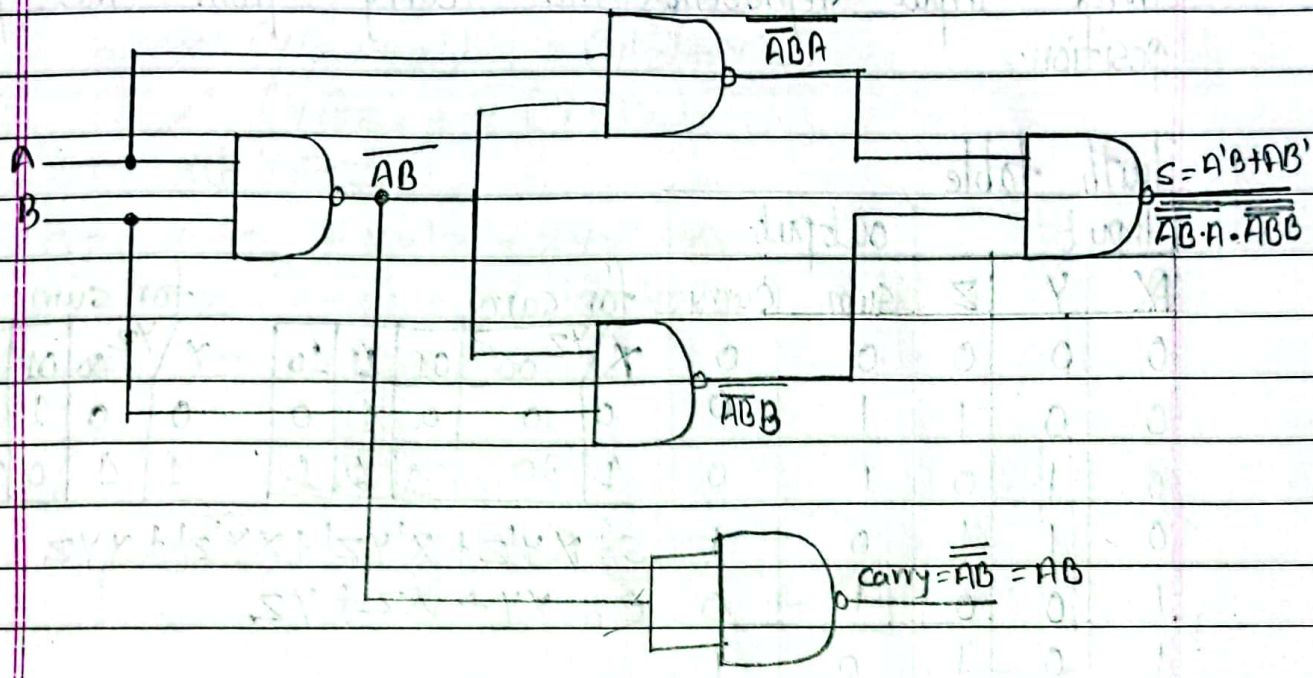


DATE _____ PAGE _____

Design a half adder using only NAND gate.

$S = A'B + AB'$

$C = AB$



Design a half adder using NOR gate.

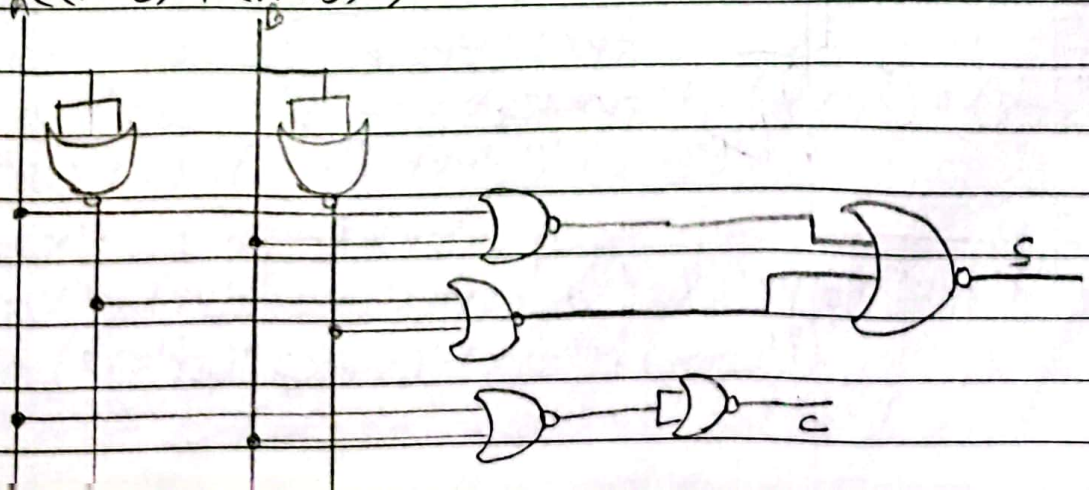
$S = (A+B).(A'+B')$

$C = \cancel{\overline{AB}}\ \cancel{AB}(A+B)$

$S = (A+B).(A'+B')$

$S' = ((A+B).(A'+B'))' = ((A+B)' + (A'+B')')$

$S'' ((A+B)' + (A'+B')')'$ $\quad C = ((A+B)')'$



DATE _____ PAGE _____

* Full Adder: A combinational circuit that perform the addition of three bit at a time is called full adder. It consists of three inputs and two outputs, two inputs are the bits to be added, the third input represent the carry from the previous position.

Truth table

Input			output	
X	Y	Z	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

for carry

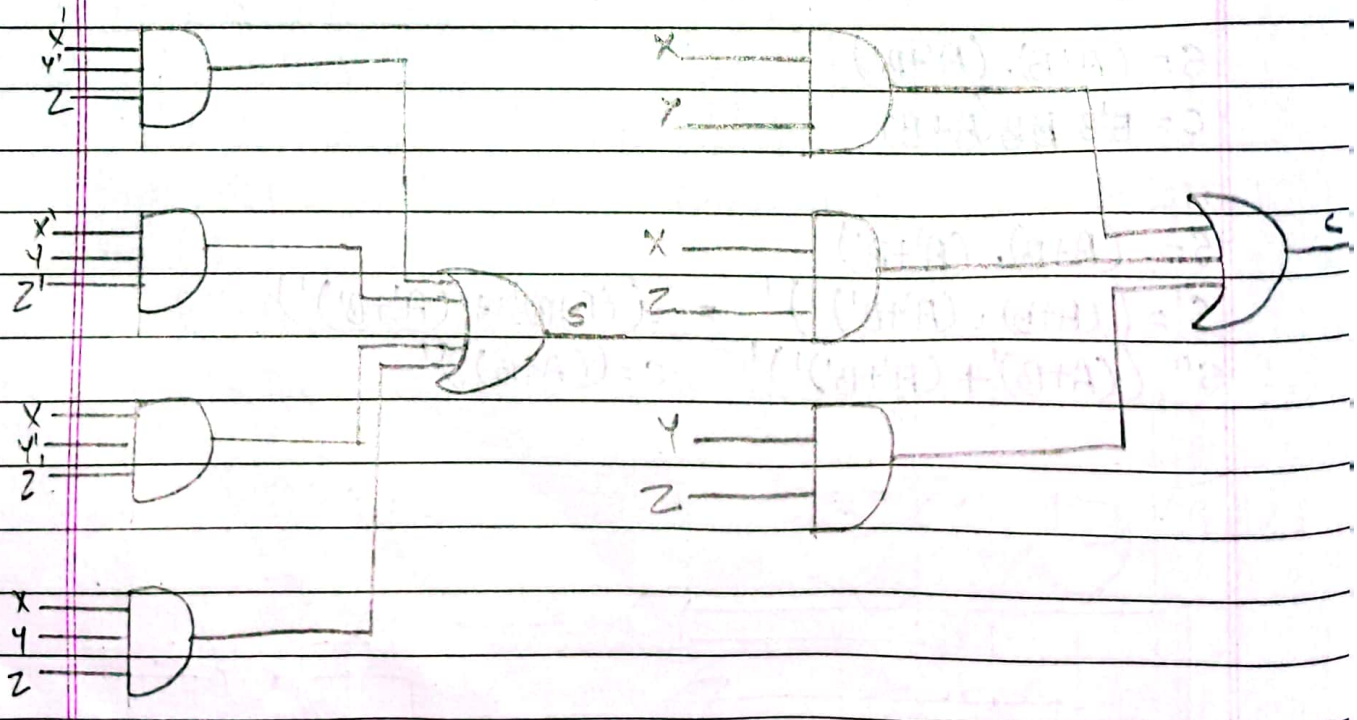
X \ YZ	00	01	11	10
0	0	0	1	0
1	0	1	1	1

for sum

X \ YZ	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$S = X'Y'Z + X'YZ' + XY'Z' + XYZ$

$C = XY + XZ + YZ$



DATE _____ PAGE _____

⊕ Design fulladder with two half adder and an OR gate

~~x'y'z+~~ S = x'y'z + x'yz' + xy'z' + xyz

C = xy + xz + yz [x'yz + xy'z + xyz' + xyz)

S = x'y'z + x'yz' + xy'z' + xyz

= x' (y'z + yz') + x (y'z' + yz)

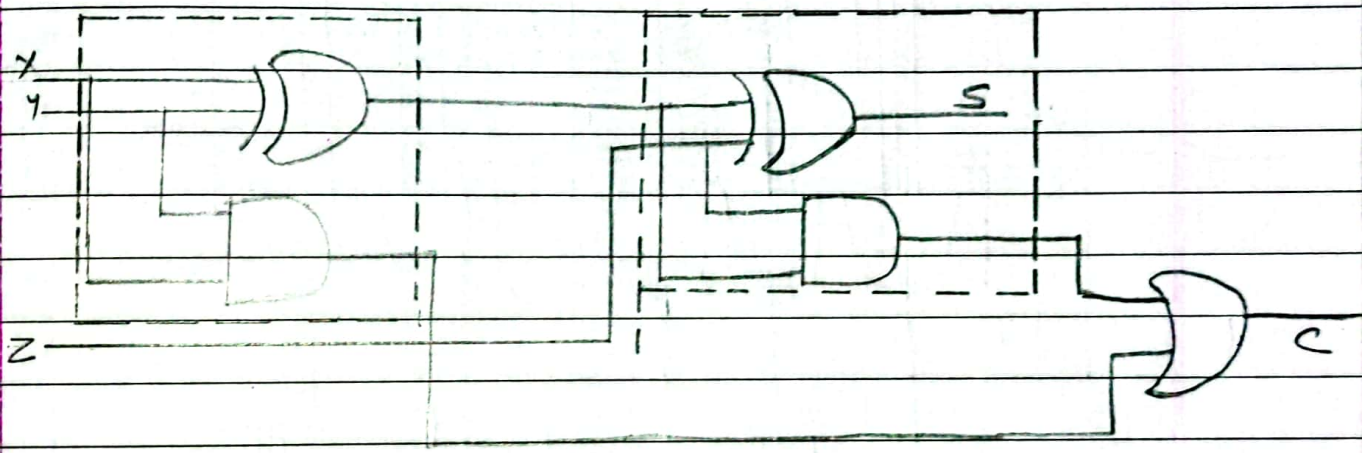
= x' (y ⊕ z) + x (y ⊕ z)

= x ⊕ y ⊕ z

C = x'yz + xy'z + xyz' + xyz

= yz (x + x') + x (y'z + yz')

= yz + x (y ⊕ z)



⊕ Design full adder using NAND gate:

Sum = x'y'z + x'yz' + xy'z' + xyz

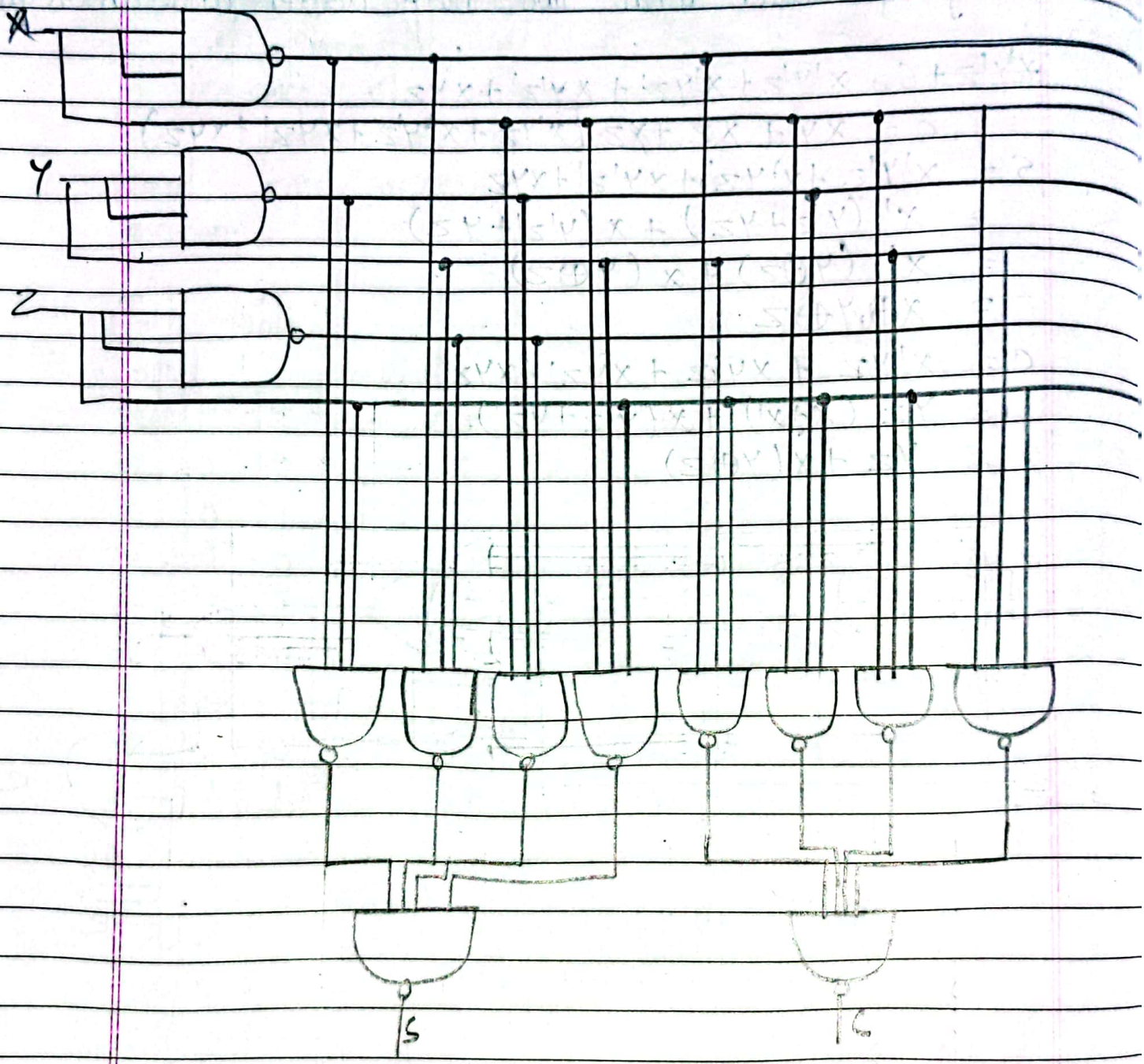
S = (x'y'z + x'yz' + xy'z' + xyz)' = ((x'y'z)' + (x'yz')' + (xy'z')' + (xyz)')

S'' = ((x'y'z)' . (x'yz')' . (xy'z')' . (xyz)')

C = x'yz + xy'z + xyz' + xyz

C' = (x'yz + xy'z + xyz' + xyz)' = ((x'yz)' + (xy'z)' + (xyz')' + (xyz)')

C'' = ((x'yz)' (xy'z)' (xyz')' (xyz)')



Design full adder using NOR gate

$S = x'y'z + x'yz' + xy'z' + xyz$ $C = x'yz + xy'z + xyz' + xyz$

x \ yz	00	01	11	10
0	0		0	
1		0		0

x \ yz	00	01	11	10
0	0	0		0
1	0			

$S = (x+y+z).(x'+y+z')\ (x+y'+z')\ (x'+y'+z)$ $C = (y+z)\ (x+y)\ (x+$

DATE _____ PAGE _____

$S' = (x+y+z) \cdot (x'+y+z') \cdot (x+y+z') \cdot (x+y'+z))'$

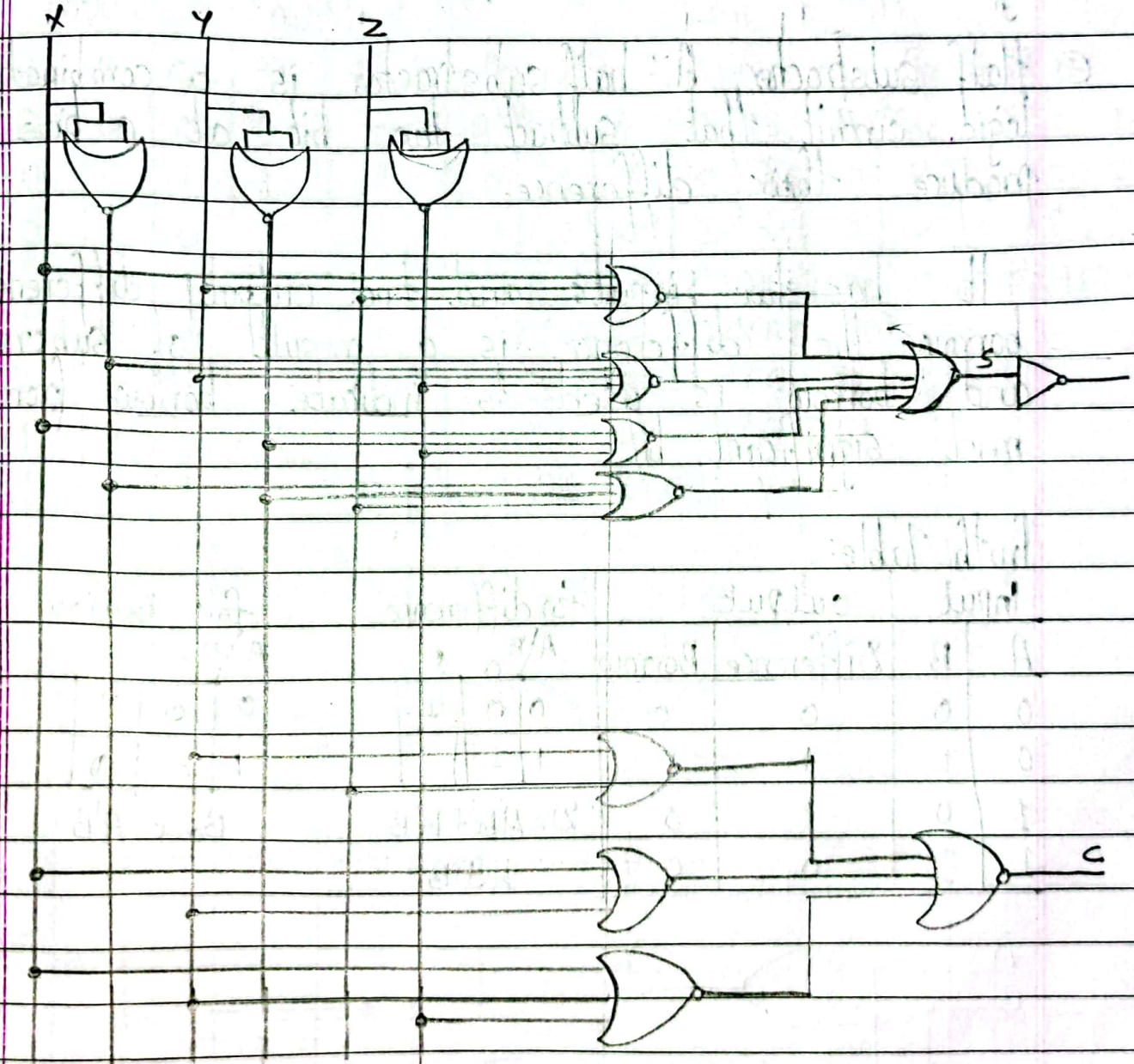
$= ((x+y+z)' + (x'+y+z)' + (x+y'+z')' + (x+y'+z)')$

$S'' (x+y+z)' + (x+y+z')' + (x+y'+z')' + (x'+y'+z)')'$

$C = (y+z) \cdot (x+y)(x+y'+z)$

$C' = ((y+z) \cdot (x+y) \cdot (x+y'+z))' = (y+z)' + (x+y)' + (x+y'+z)'$

$C'' = ((y+z)' + (x+y)' + (x+y'+z)')'$



DATE _____ PAGE _____

Subtractors:

Subtractor is a combinational logic circuit which is used to subtract two or more than two bits at a time and provides difference and borrow as an output.

Type of substractor:

⊕ Half substractor: A half substractor is a combinational logic circuit that subtract two bit at a time and produce their difference.

It has two inputs and two output difference and borrow. The difference is a result of subtraction and borrow is used to indicate borrow from next most significant bit.

Truth Table:

Input		output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

for difference

A \ B	0	1
0	0	1
1	1	

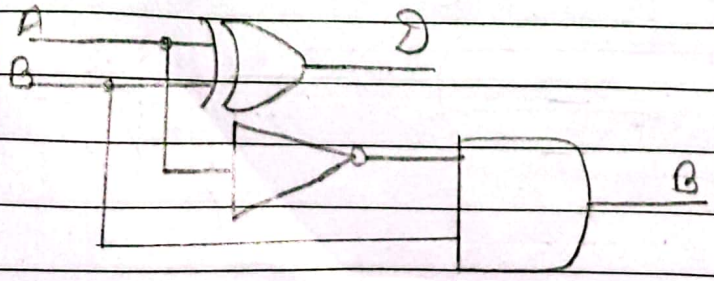
$D = AB' + A'B$

$= A \oplus B$

for Borrow

A \ B	0	1
0	0	1
1	0	0

Bout $A'B$



DATE _____ PAGE _____

⊕ Full Substractor: A combinational logic circuit used to substract three binary digit at a time is called full substractor. The circuit has three input and two output. There three input are A, B, C & two output D and Bout respectively.

Truth table for full substractor:

Input			output	
A	B	C	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

for D

A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

for Bout

A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

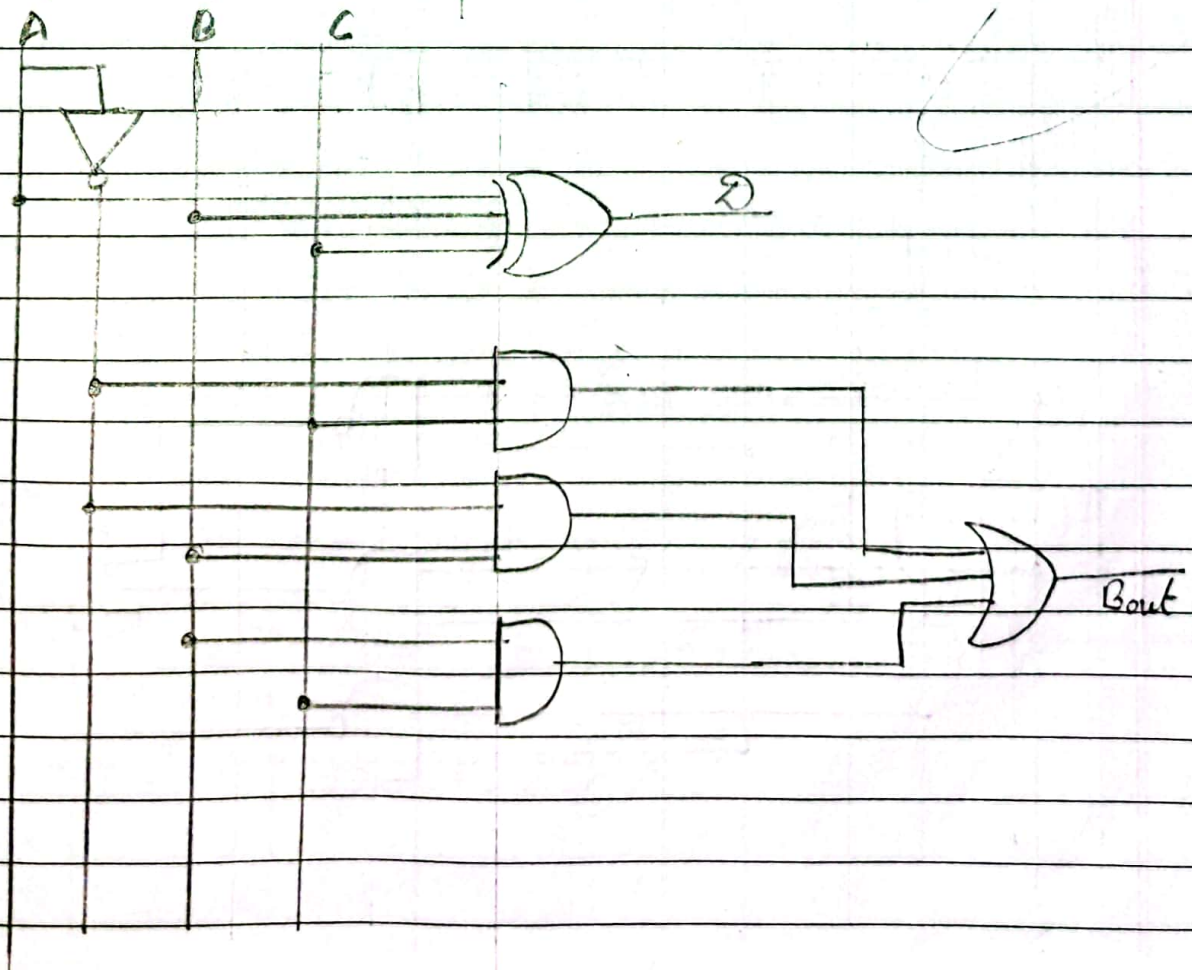
$D = A'B'C + A'BC' + AB'C' + ABC$ Bout $A'C + A'B + BC$

$= C(A'B' + A'B) + C'(A'B + A'B)$

$= C(A \odot B) + C'(A \oplus B)$

$= C(A \oplus B)' + C'(A \oplus B)$

$= A \oplus B \oplus C$



DATE _____ PAGE _____

Draw and implement full substractor using NAND gate.

$D = A'B'C + A'BC' + AB'C' + ABC$

$D' = (A'B'C + A'BC' + AB'C' + ABC)'$

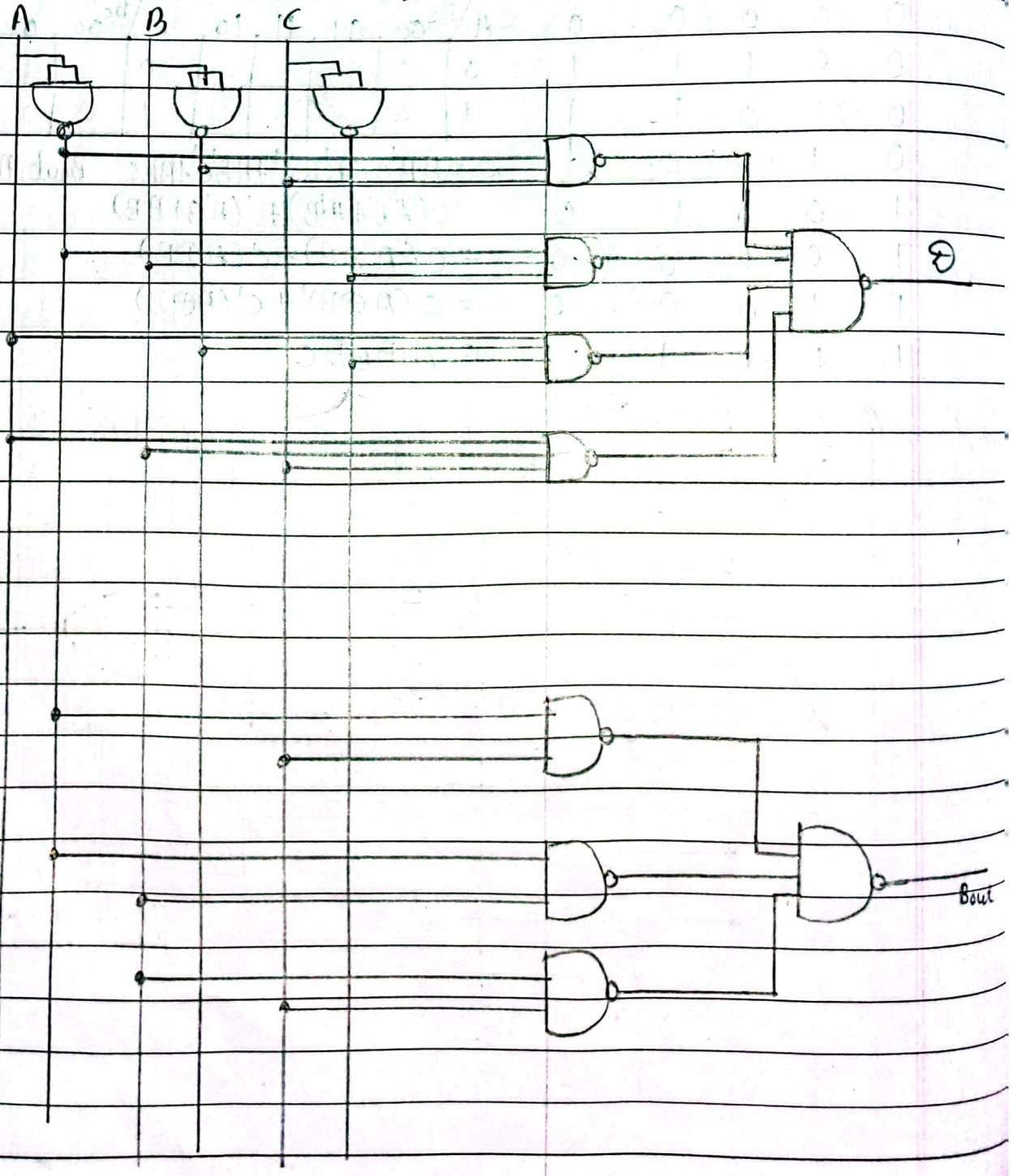
$' = ((A'B'C)' + (A'BC')' + (AB'C')' + (ABC)')$

$D'' = ((A'B'C)' \cdot (A'BC')' \cdot (AB'C')' \cdot (ABC)')'$

$B_{out} = A'C + AB + BC$

$B'_{out} = (A'C + A'B + BC)' = ((A'C)' + (A'B)' + (BC)')$

$B'_{out} = ((A'C)' \cdot (A'B)' \cdot (BC)')'$



DATE _____ PAGE _____

Draw and implement full substractor using NOR gate.

$D = A'B'C + A'BC' + AB'C' + ABC$

A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$D = (A+B+C)(A+B'+C')(A'+B+C')(A'+B'+C)$

$D' = ((A+B+C)(A+B'+C')(A'+B+C')(A'+B'+C))'$

$= ((A+B+C)' + (A+B'+C')' + (A'+B+C')' + (A'+B'+C)')$

$D'' = ((A+B+C)' + (A+B'+C')' + (A'+B+C')' + (A'+B'+C)')'$

$B_{out} = A'C + A'B + BC$

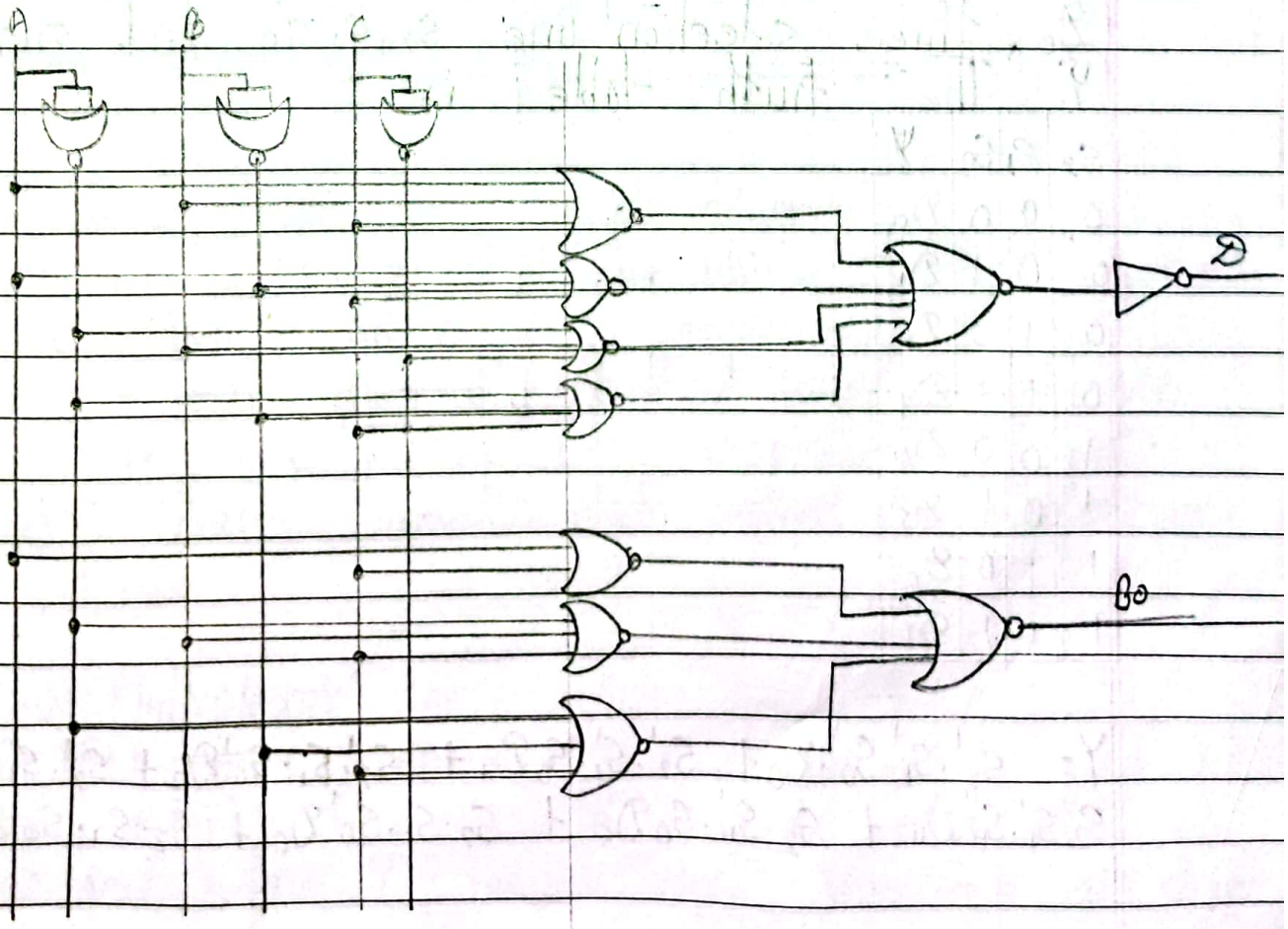
A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$B_{out} = (B+C)(A'+B+C')(A'+B'+C)$

$B' = ((B+C)(A'+B+C')(A'+B'+C))'$

$= ((B+C)' + (A'+B+C)' + (A'+B'+C)')$

$B'' = ((B+C)' + (A'+B+C)' + (A'+B'+C)')'$



DATE _____ PAGE _____

Multiplexer and Demultiplexer

1. Multiplexer: Multiplexer means a circuit used to select and route any one of several input signals to a single output. It has maximum 2^n data input, 'n' selection of line and

1. Multiplexer: Multiplexer is a combinational circuit that has maximum of 2^n data input, 'n' sechional line and single output line. There are 'n' selection line, there will be 2^n possible combinations of zeros and ones. It is also called as Mux.

8x1 Multiplexer

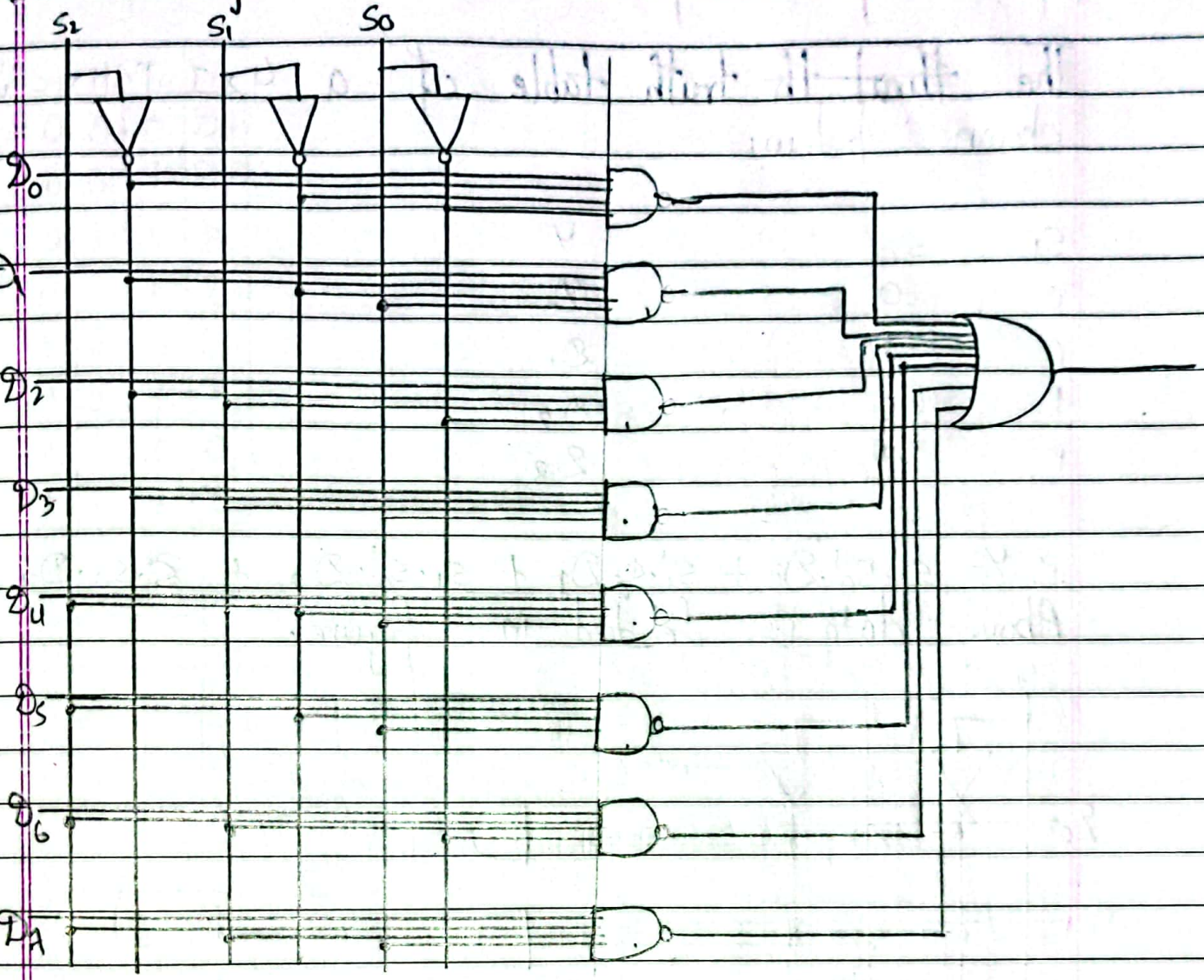
let the 8x1 Multiplexer has eight data input D_7 to D_0, three selection line S_2, S_1 S_0 and one output Y. The truth table is

S_2	S_1	S_0	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

$Y = S_2' \cdot S_1' \cdot S_0' \cdot D_0 + S_2' \cdot S_1' \cdot S_0 D_1 + S_2' \cdot S_1 \cdot S_0' D_2 + S_2' \cdot S_1 S_0 D_3 + S_2 \cdot S_1' \cdot S_0' D_4 + S_2 \cdot S_1' \cdot S_0 D_5 + S_2 \cdot S_1 \cdot S_0' D_6 + S_2 \cdot S_1 \cdot S_0 D_7$

DATE _____ PAGE _____

Circuit diagram



In above figure the logical diagram of 8 to 1 mux is implemented by using 8 AND gate three Not gate for complement of Input and ~~and~~ one OR gate. When enable pin 1 is set mux will disable & if it is zero then input data pass through output.

4x1 Multiplexer

A 4x1 mux consist four data input as D_0 to D_3 two select line as S_0 and S_1 and single out put line Y. The select line S_0 & S_1 one

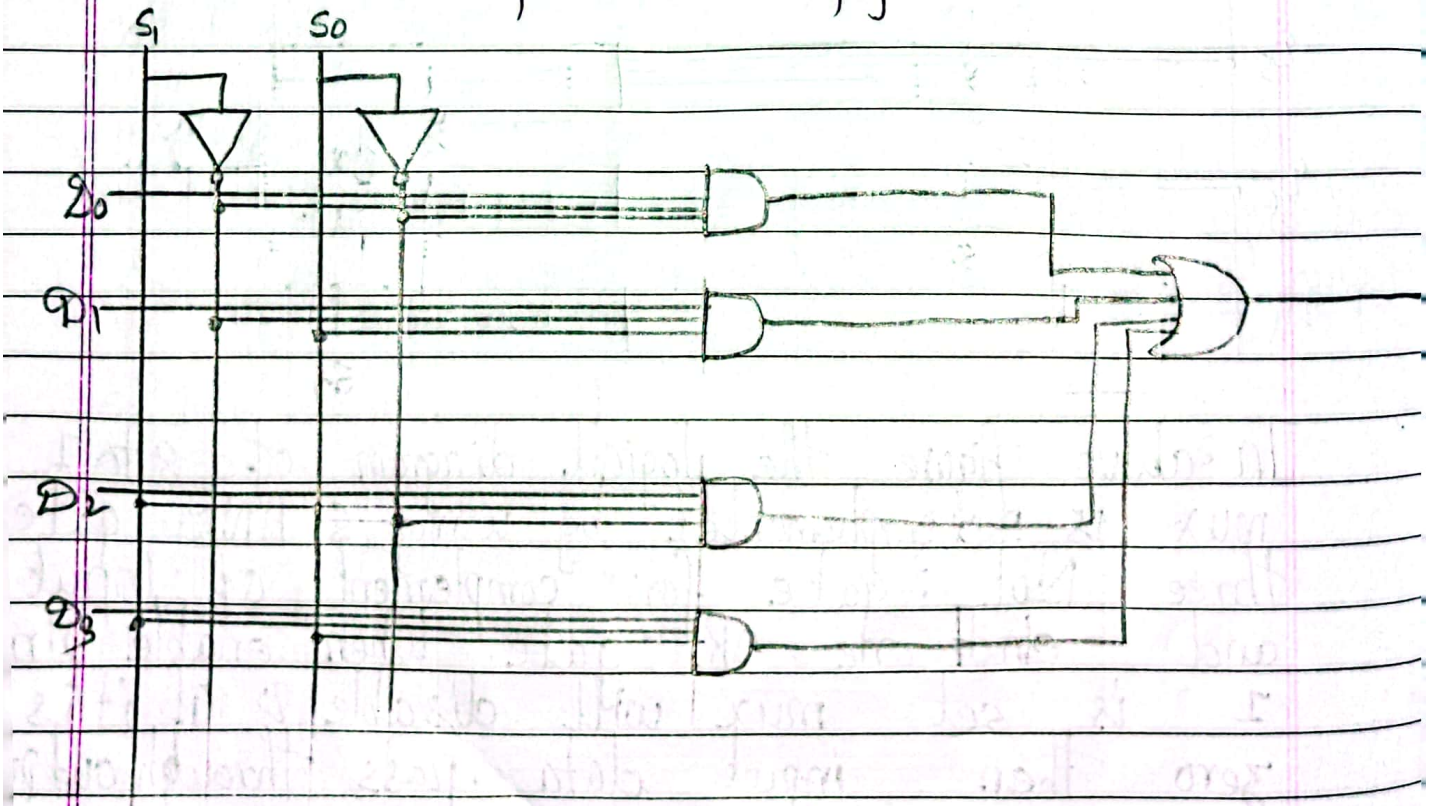
of the four input line to connect the output line.

The truth table of a 4x1 MUX is shown below:

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$\therefore Y = S_1' \cdot S_0' \cdot D_0 + S_1' \cdot S_0 D_1 + S_1 \cdot S_0' \cdot D_2 + S_1 \cdot S_0 \cdot D_3$

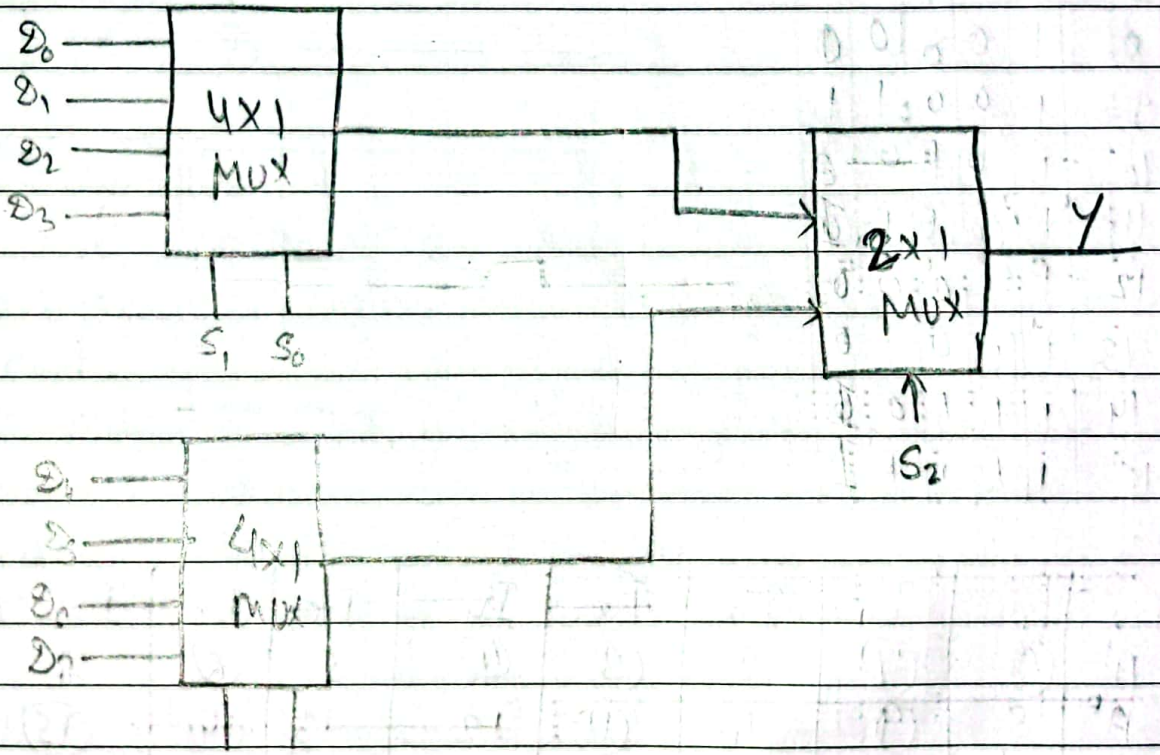
Above data is fielded in figure.



DATE _____ PAGE _____

Design 8x1 mux using lower mux

S_2	S_1	S_0	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



→ In figure selection line $S_1 S_0$ are applied in 4x1 Multiplexers. The data input in upper MUX from D_0 to D_3 and data input in lower Mux from D_4 to D_7. Each 4x1 Multiplexer produce an output based on value of selection line. S_2 is applied to 2x1 Mux.

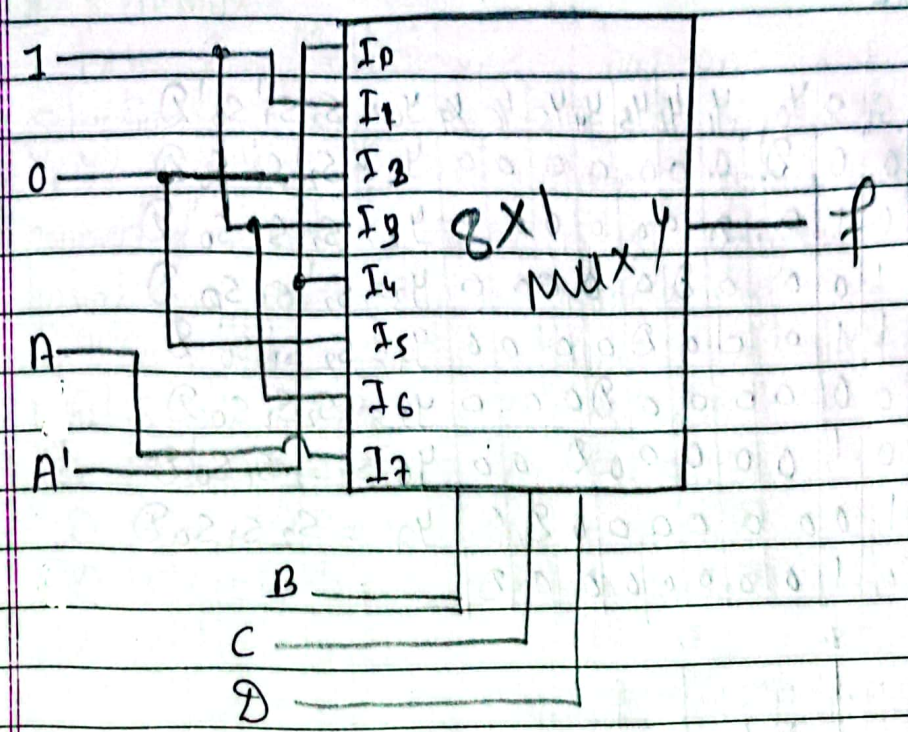
DATE _____ PAGE _____

Design Multiplexer for give boolean function $f(A,B,C,D)$

$\Sigma(0,1,3,4,6,7,8,9,11,14,15)$

Min	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

	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)
	A'	1	0	1	A'	0	1	A



Demultiplexer:

A combinational circuit that performs the reverse operation of Multiplexer. It has single input, 'n' selection line and maximum of 2^n outputs. The address input determine which data output is going to have the same value as the data input. The other data outputs will have the value 0.

1x8 Demultiplexer

A 1 to 8 (1x8) demux consist of one input and 8 output line and 3 select line. Let the input be D, S_1 and S_2 are two select line and eight output from Y_0 to Y_7

DATE _____ PAGE _____

Truth table

Input	S_2	S_1	S_0	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

$y_0 = S_2' S_1' S_0' D$

$y_1 = S_2' S_1' \cdot S_0 D$

$y_2 = S_2' S_1 S_0' D$

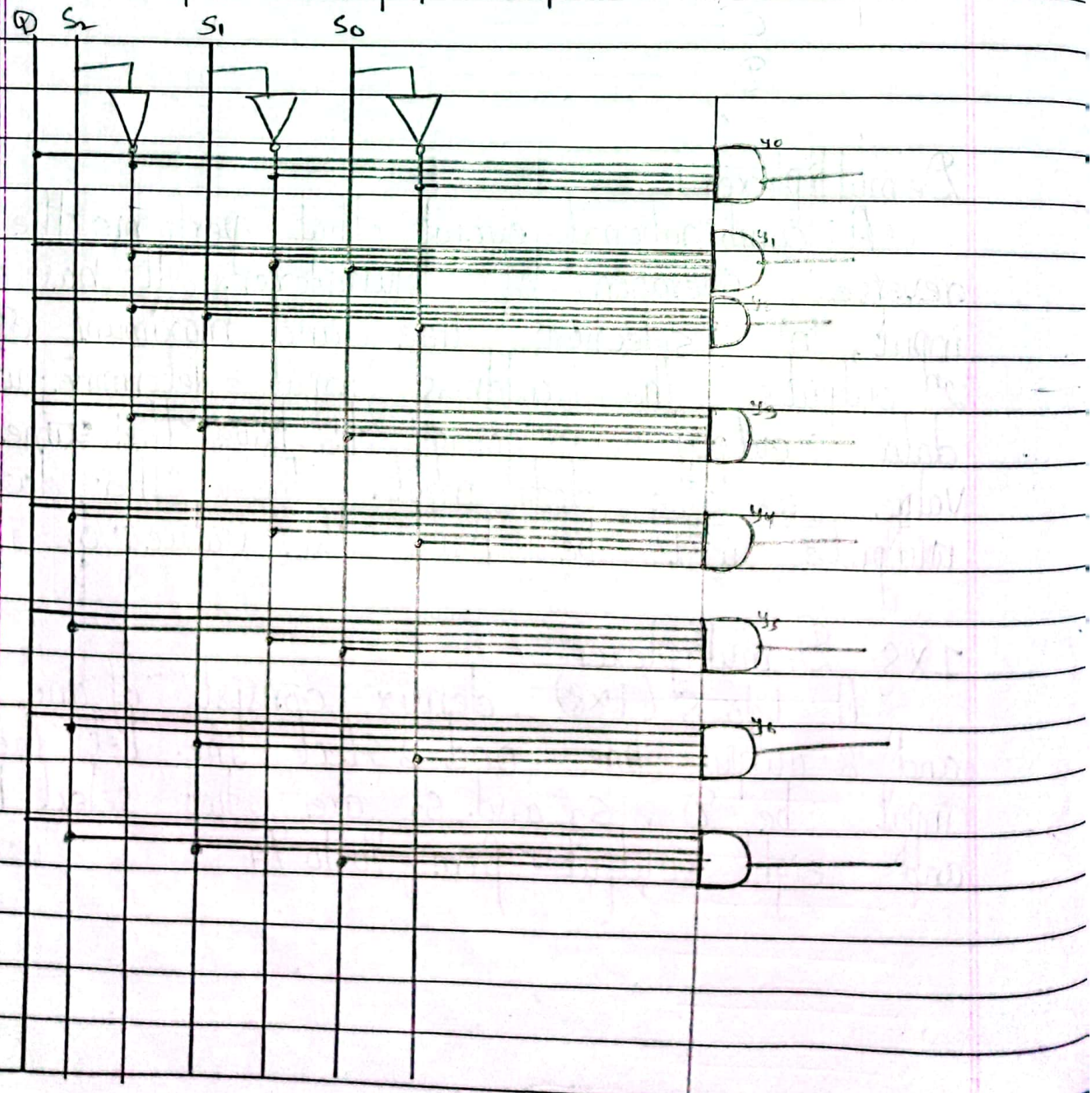
$y_3 = S_2' S_1 S_0 D$

$y_4 = S_2 S_1' S_0' D$

$y_5 = S_2 S_1' S_0 D$

$y_6 = S_2 S_1 S_0' D$

$y_7 = S_2 S_1 S_0 D$



DATE _____ PAGE _____

1x4 demux :

1x4 De-Multiplexer has one input D, two selection lines S_1 & S_0 and four output Y_3 Y_2 Y_1 & Y_0 The single input D will be connected to one of four output, Y_3 to Y_0 based on the value of selection line S_1 & S_0. The truth table of 1x4 DeMux is:

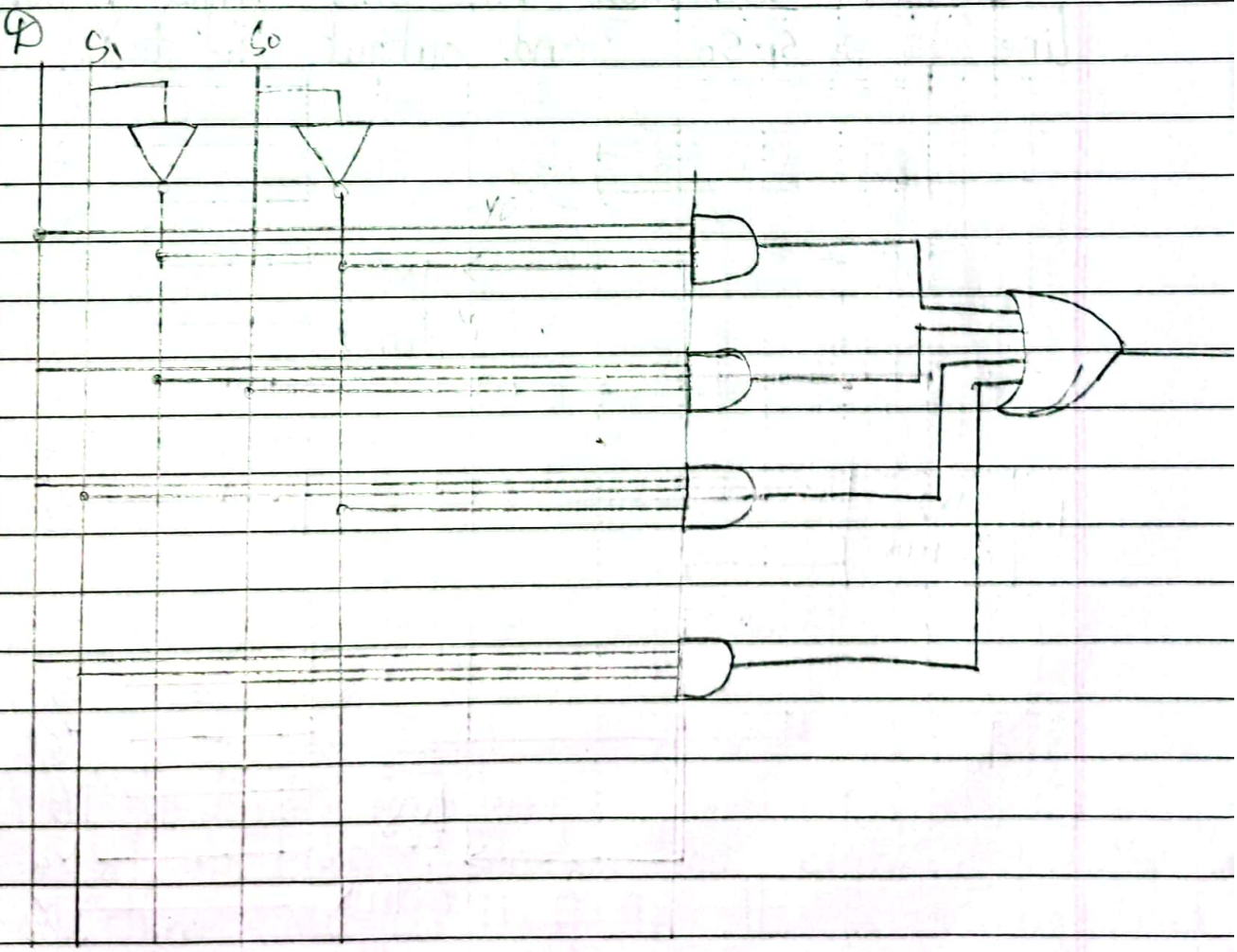
Input	S_1	S_0	Y_0	Y_1	Y_2	Y_3
D	0	0	D	0	0	0
D	0	1	0	D	0	0
D	1	0	0	0	D	0
D	1	1	0	0	0	D

$Y_0 = S_1' S_0' D$

$Y_1 = S_1' S_0 D$

$Y_2 = S_1 S_0' D$

$Y_3 = S_1 S_0 D$

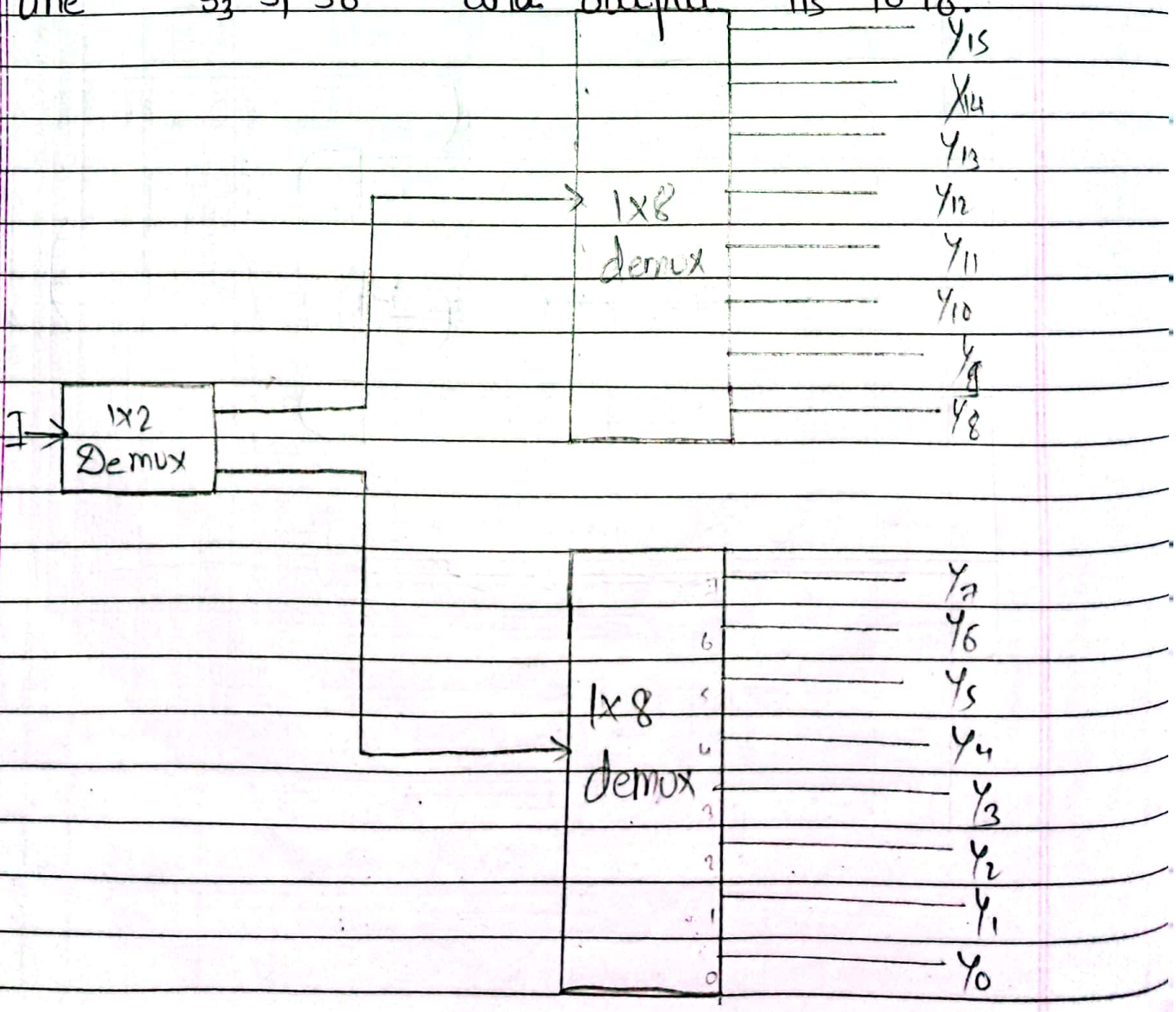


DATE _____ PAGE _____

1x16

In this ~~selec~~ section let us implement 1x16 Demux using 1x8 Demux and 1x2 De mux. ~~There are thr~~ In 1x16 de Mux has single input, four selection line and sixteen output. for 1x8 demux in second stage in order to get the final sixteen output. The number of input in second stage is two We require 1x2 demux in first stage so that the output of first stage will be input of second stage.

let 1x16 De-mux has one input, four selection line S_3 S_1 S_0 and output Y_{15} to Y_0.



DATE _____ PAGE _____

- Encoder

An Encoder is combinational circuit that perform the reverse operation of decoder. It has maximum of 2^n input lines and 'n' output lines. It will produce a binary code which active high.

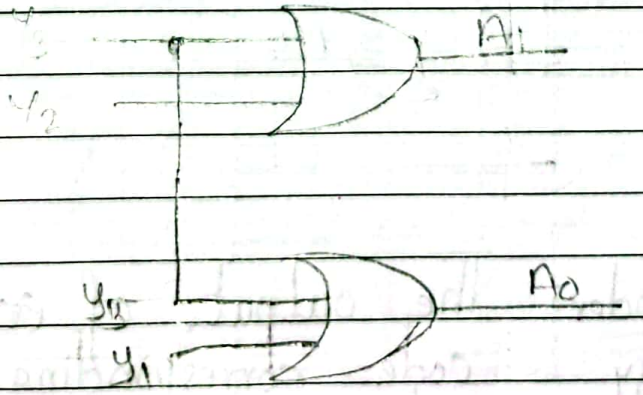
4 to 2 encoder

Y_3	Y_2	Y_1	Y_0	A_1	A_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

from Above table

$A_1 = Y_3 + Y_2$ $A_0 = Y_3 + Y_1$

The circuit diagram is



Octal to Binary Encoder

Octal to Binary Encoder has eight input, Y_7 to Y_0 and three output $A_2 A_1 A_0$. It is nothing but 8 to 3 encoder. Only one of these eight input can be '1' in order to get the respective binary code.

DATE _____ PAGE _____

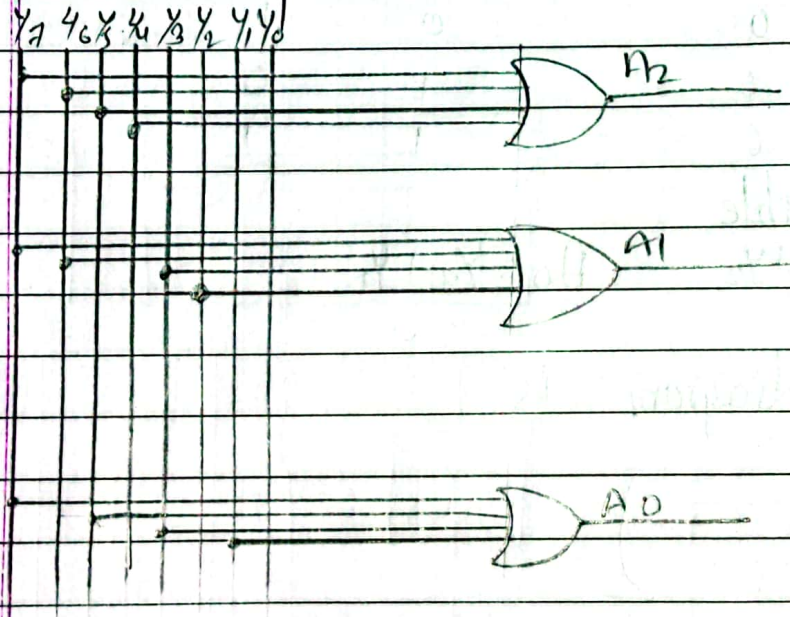
Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$A_2 = Y_7 + Y_6 + Y_5 + Y_4$

$A_3 = Y_7 + Y_6 + Y_3 + Y_2$

$A_0 = Y_7 + Y_5 + Y_3 + Y_1$

We can implement it in circuit diagram



Priority encoder: The output of encoder will be the binary code corresponding to active high input and which has higher priority is called as priority encoder.

DATE _____ PAGE _____

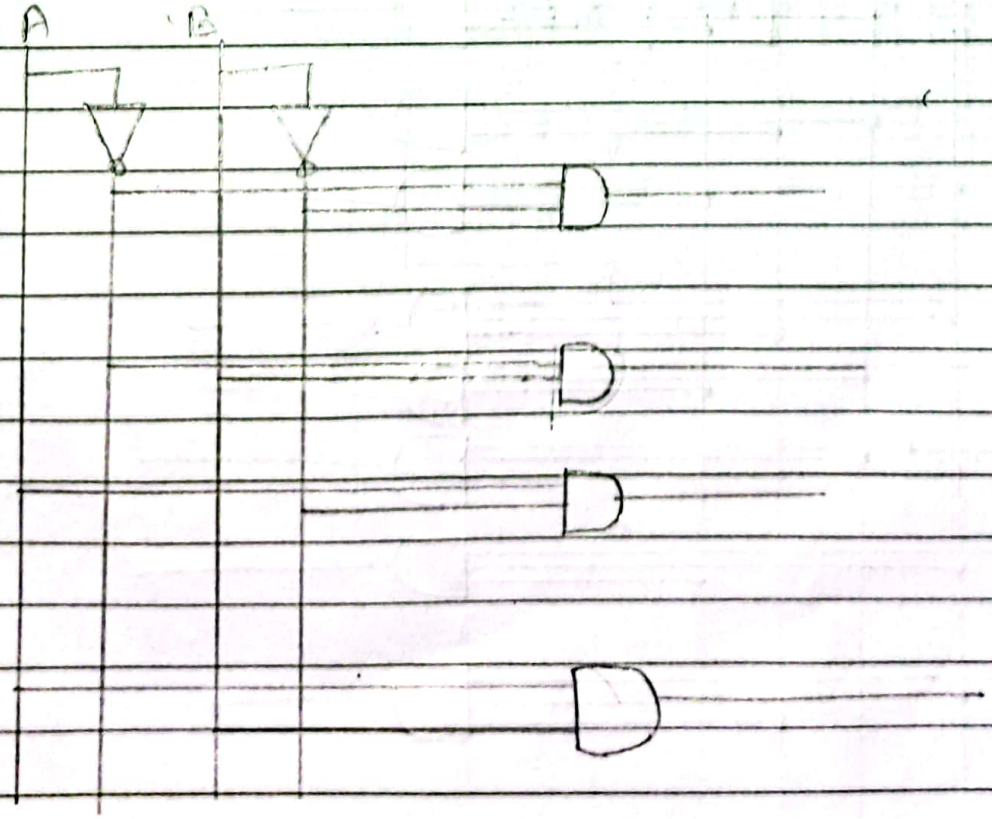
Decoder: Decoder is a combinational circuit that has 'n' input line and maximum of 2^n output line. One of these outputs will be active High based on the combination of inputs present, When the decoder is enabled. That means decoder detects a particular code.

* 2 to 4 decoder: let 2 to 4 decoder has two input A_1 and A_0 and four output P_3, P_2, P_1 & P_0. One of these four output will be '1' for each combination of input when enable E is 1.

Truth table

A	B	P_0	P_1	P_2	P_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

$P_0 = A'B'$
$P_1 = A'B$
$P_2 = AB'$
$P_3 = AB$

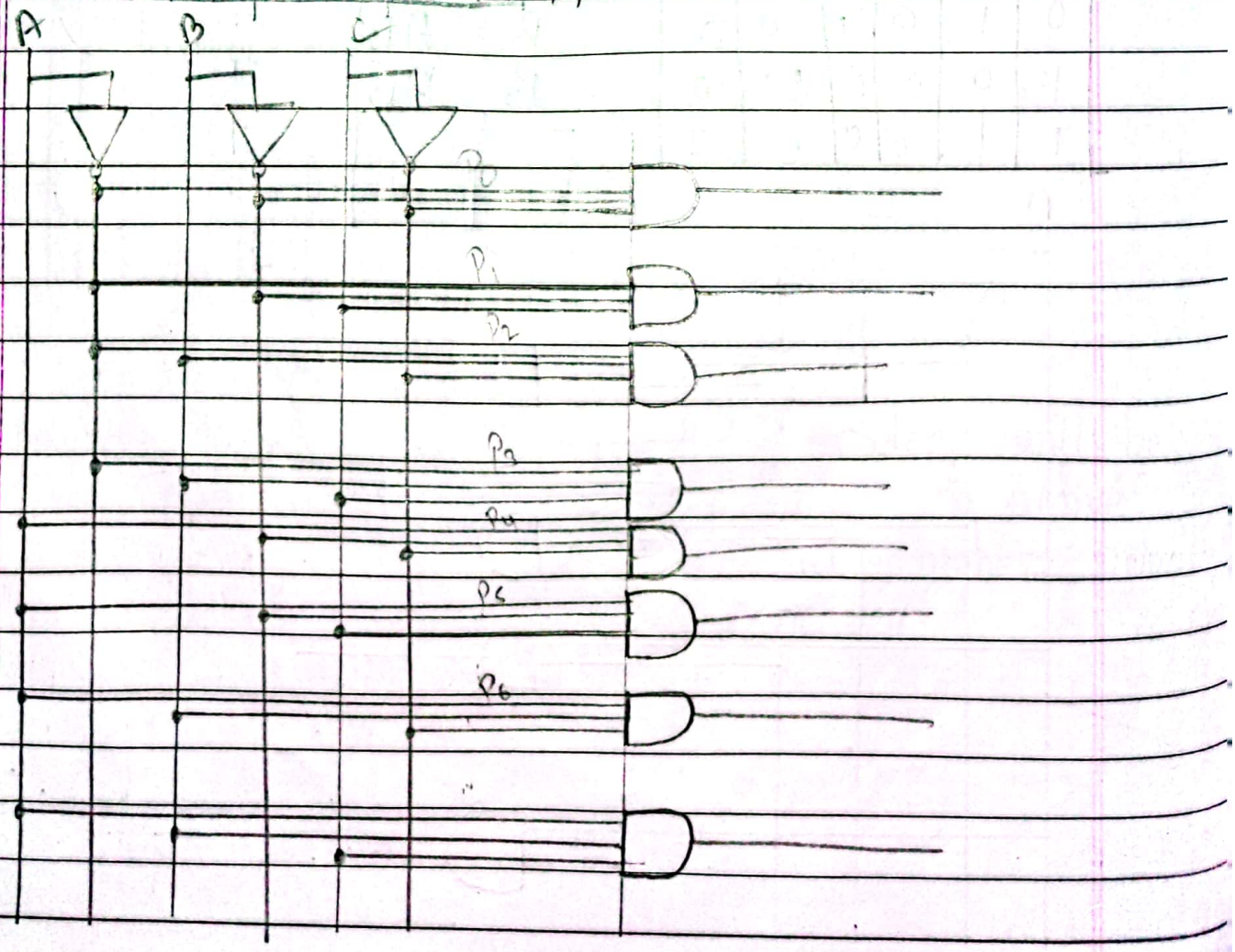


DATE _____ PAGE _____

- 3-8 line decoder:

let 3 to 8 line decoder has three input A, B, C, and eight output put P_0 to P_7. one of these eight output will be 1. Therefore the abov Truth table is structured as :

A	B	C	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	
0	0	0	1	0	0	0	0	0	0	0	$P_0 = A'B'C'$
0	0	1	0	1	0	0	0	0	0	0	$P_1 = A'B'C$
0	1	0	0	0	1	0	0	0	0	0	$P_2 = A'BC'$
0	1	1	0	0	0	1	0	0	0	0	$P_3 = A'BC$
1	0	0	0	0	0	0	1	0	0	0	$P_4 = AB'C'$
1	0	1	0	0	0	0	0	1	0	0	P_5 $AB'C$
1	1	0	0	0	0	0	0	0	1	0	P_6 ABC'
1	1	1	0	0	0	0	0	0	0	1	$P_7 = ABC$



DATE _____ PAGE _____

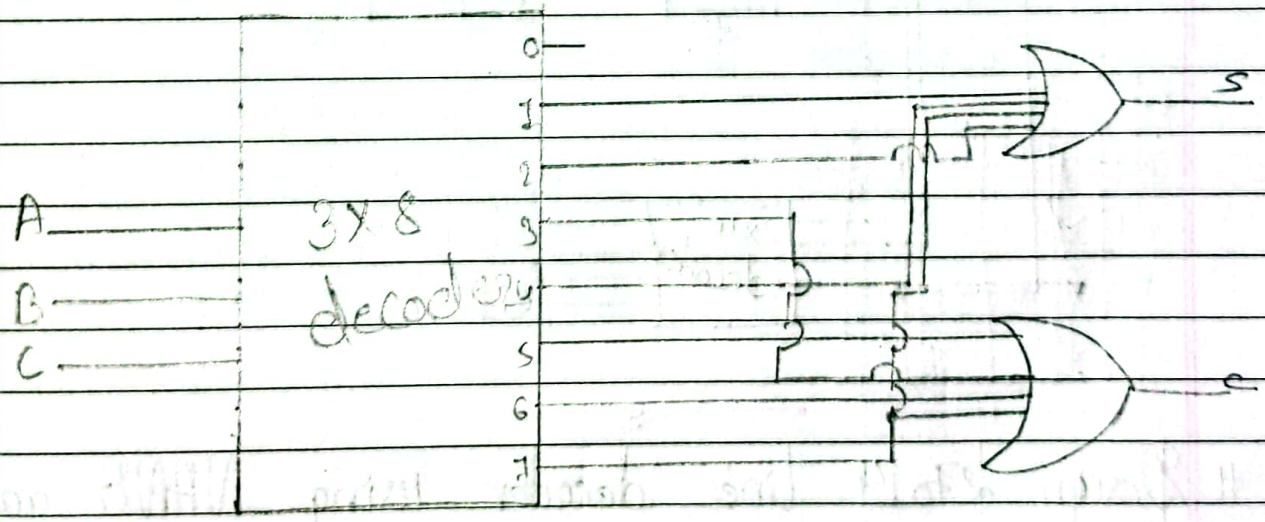
* Design a full adder using decoder and two OR gate

A	B	D	S	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

from truth table $S(ABD) = \Sigma(1,2,4,7)$

$C(ABD) = \Sigma(3,5,6,7)$

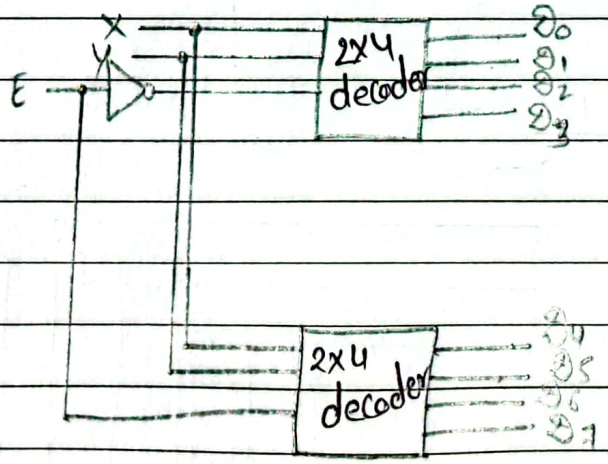
Since, there are three input and total of eight minterms. So, we need 3 to 8 line decoder. The decoder generate the eight minterms for ABD. The OR gate for output sum (S) & carry is (C).



DATE _____ PAGE _____

Design 3 to 8 line decoder using lower decoder

X	Y	E	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	0	1

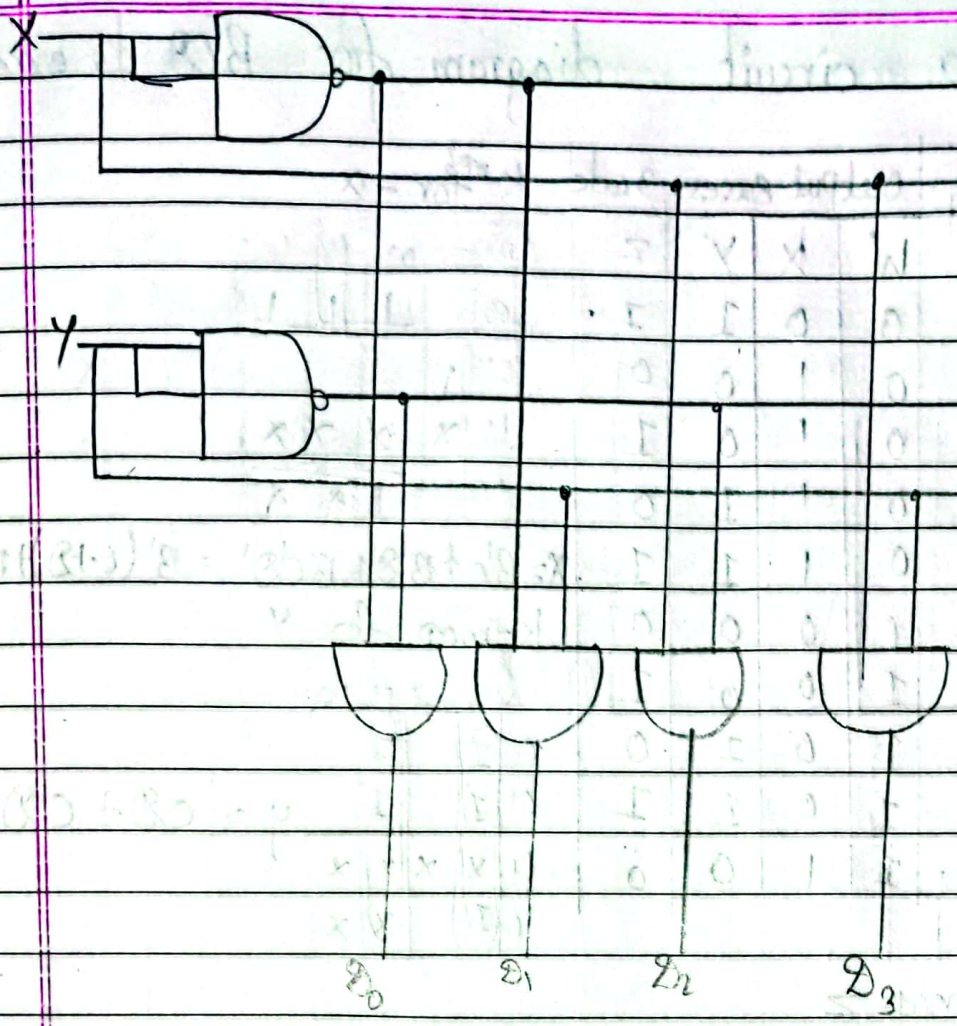


Design 2 to 4 line decoder using NAND gate

Truth table

X	Y	D_0	D_1	D_2	D_3		
0	0	1	0	0	0	D_0	$x'y'$
0	1	0	1	0	0	D_1	$x'y$
1	0	0	0	1	0	D_2	xy'
1	1	0	0	0	1	D_3	xy

DATE _____ PAGE _____



DATE _____ PAGE _____

Design a circuit diagram for BCD to excess-3 code converter.

	Input BCD				Output Excess-3 code			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	1	1	1	1	0	1	1
9	1	0	0	1	1	1	0	0

K-map for - X

AB \ CD	00	01	11	10
00		1	1	1
01	1			
11	x	x	x	x
10		1	x	x

$X = B'C + B'D + BC'D' = B'(C+D) + BC'D'$

K-map for Y

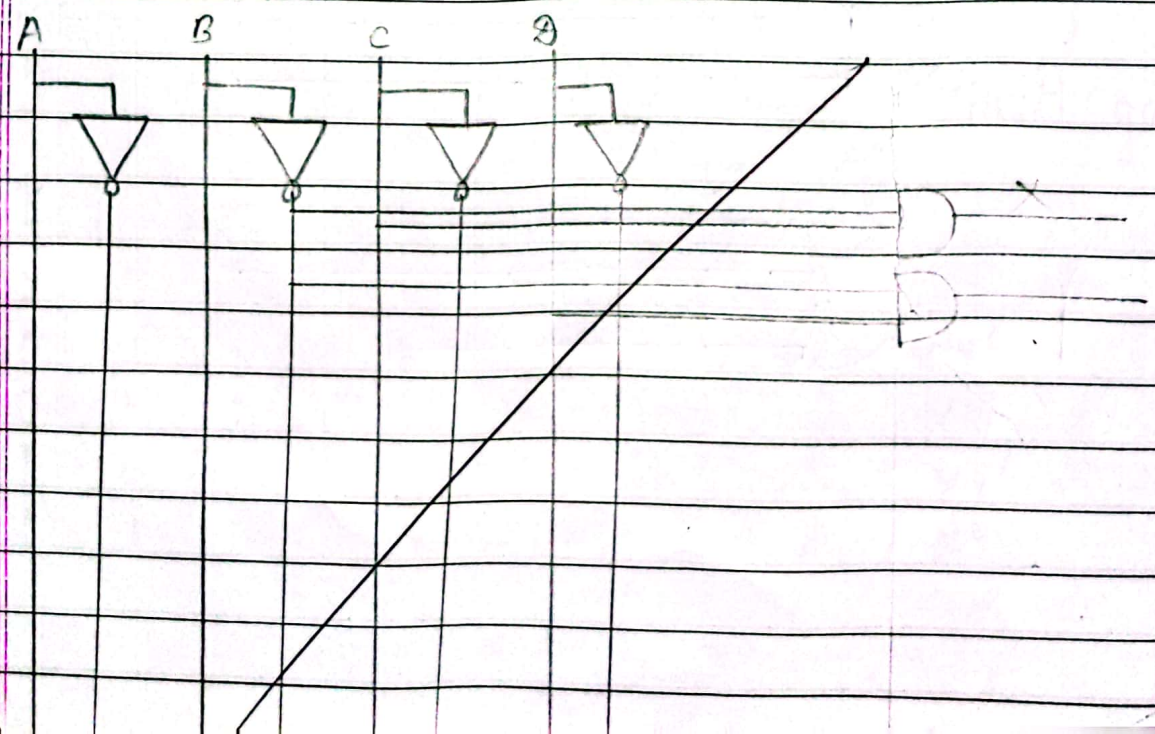
AB \ CD	00	01	11	10
00	1		1	
01	1		1	
11	x	x	x	x
10	1		x	x

$Y = CD + C'D'$

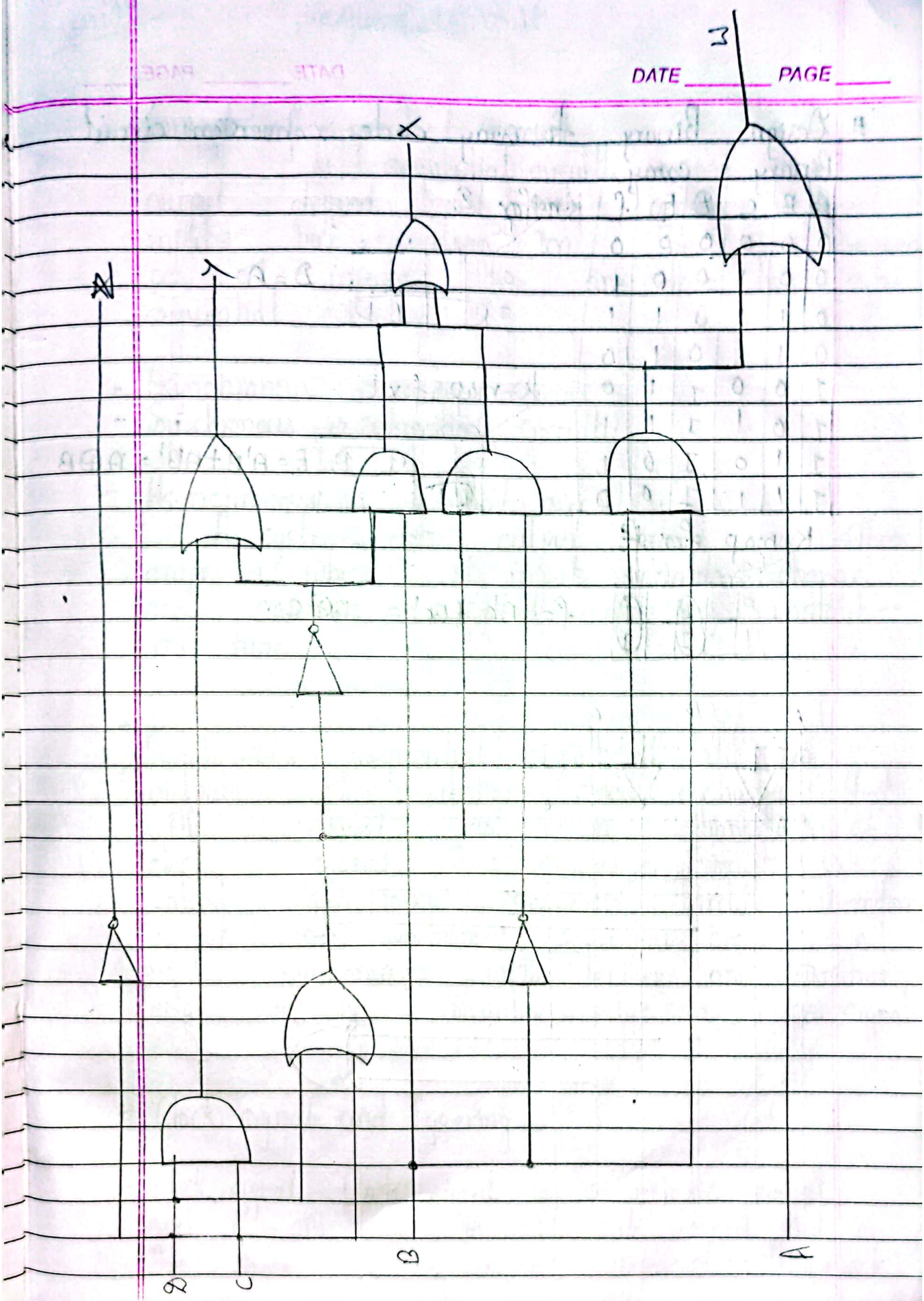
for K-map Z

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	x	x	x	x
11	x			x

$Z = D'$



DATE _____ PAGE _____

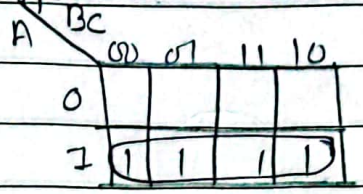


DATE _____ PAGE _____

Design Binary to Gray code conversion circuit

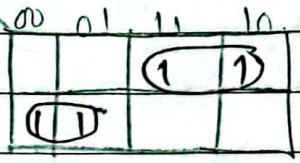
Binary			Gray		
A	B	C	D	E	f
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

K-map for D



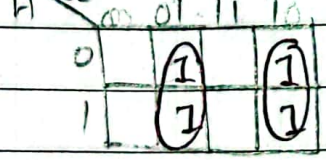
$D = A$

K-map for E



$E = A'B + AB' = A \oplus B$

K-map for f



$f = B'C + BC' = B \oplus C$