Difference between combinational logic circuit and sequential logic circuit

Combinational	Sequential
1. output of any instance of time depends only upon the present input only	Output is generated dependent upon the present input as well as past output of these inputs.
2. Memory unit is not required.	Memory unit is required to store past output.
3. Faster	Slower
4. Easy to design	difficult
5. e.g. Half & full adder, Half & full subtractor, MUX, DEMUX	e.g. Flip flops, shift register, counters

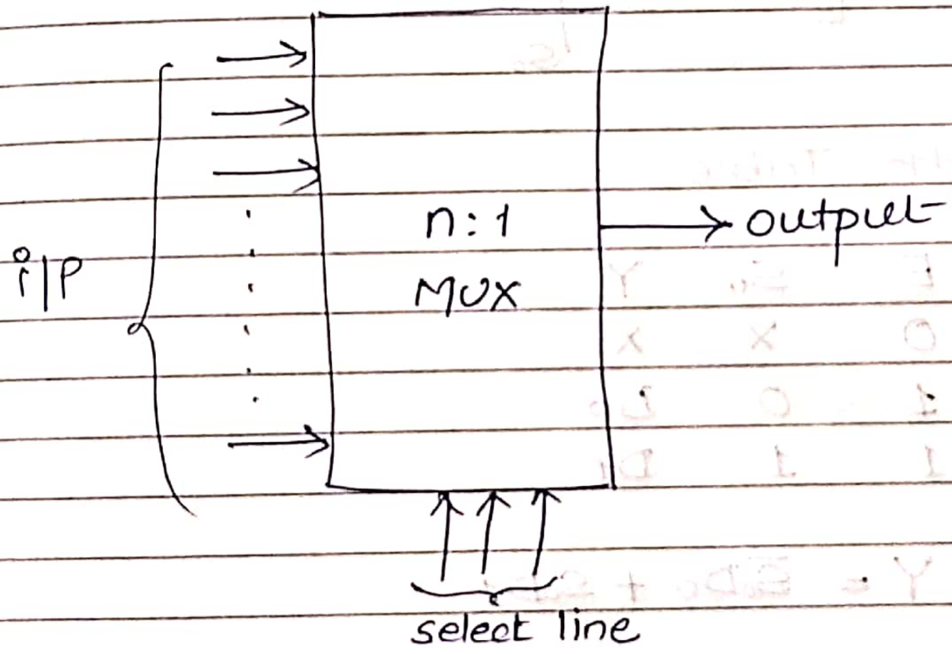
Scanned with CamScanner

Multiplexer

Multiplexer is a combinational logic circuit having many inputs and one output.

Multiplexer is many to one.

In multiplexer select line select the one input as a output



Types of Multiplexer
1. 2:1 Multiplexer
2. 4:1 Multiplexer
3. 8:1 Multiplexer
4. 16:1 Multiplexer

Select lines

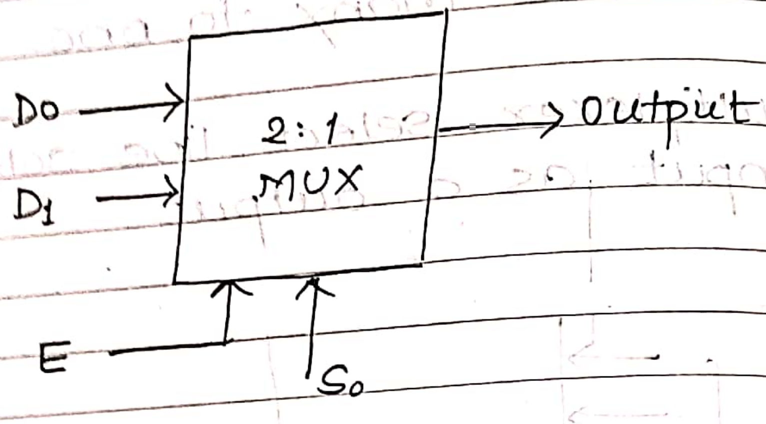
$2^m = n$

m = Select lines

n = inputs

2:1 Multiplexer

Inputs : 2
output : 1
Select lines : 1

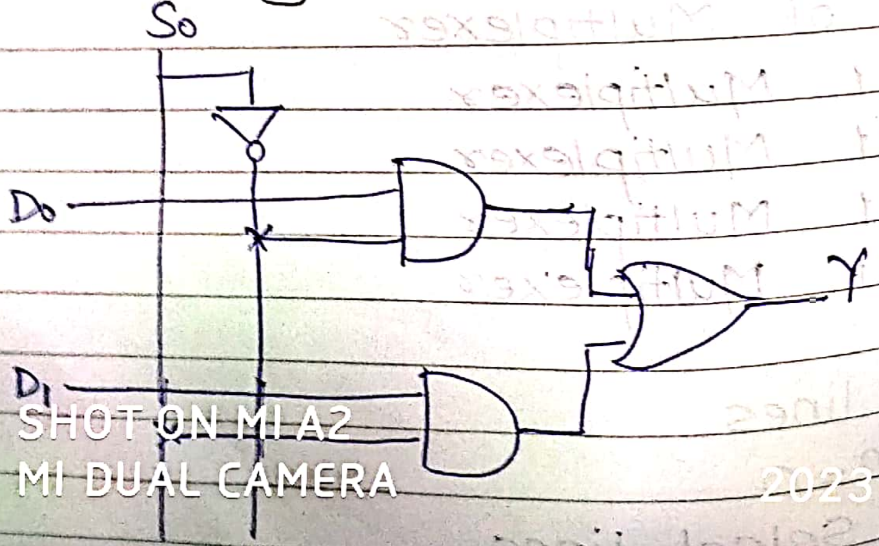


Truth Table

E	S_0	Y
0	X	X
1	0	Do
1	1	D1

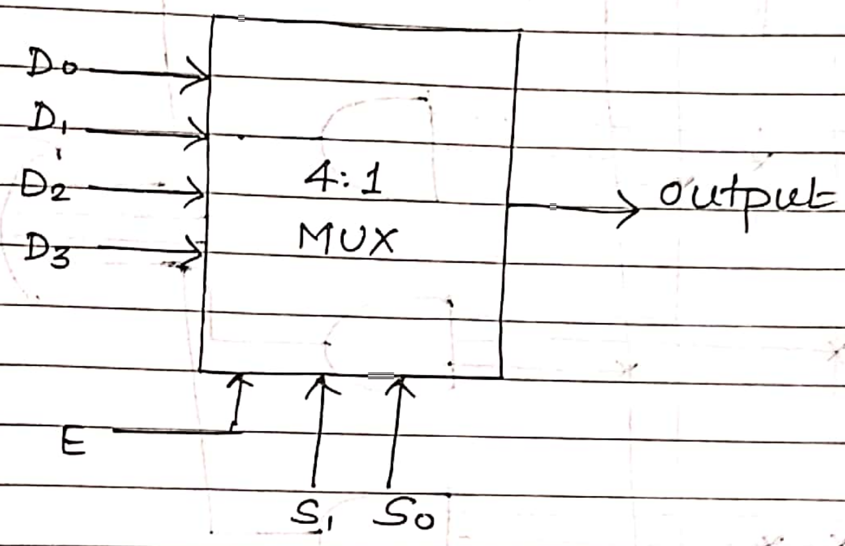
$$Y = \overline{S_0}D_0 + S_0D_1$$

Circuit diagram



4:1 Multiplexer

Inputs : 4
Outputs : 1
Select lines : 2



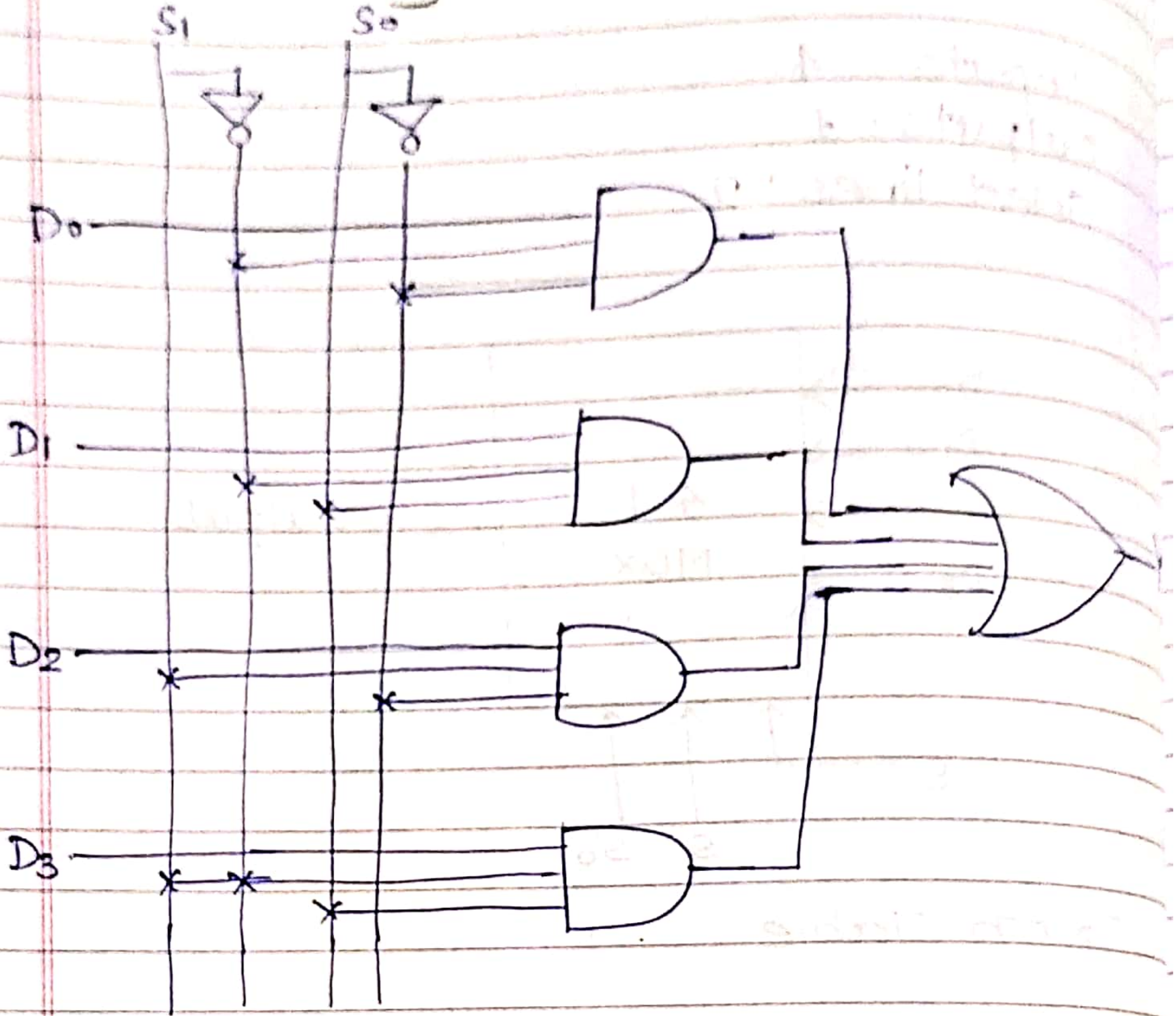
Truth Table

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

Equation :

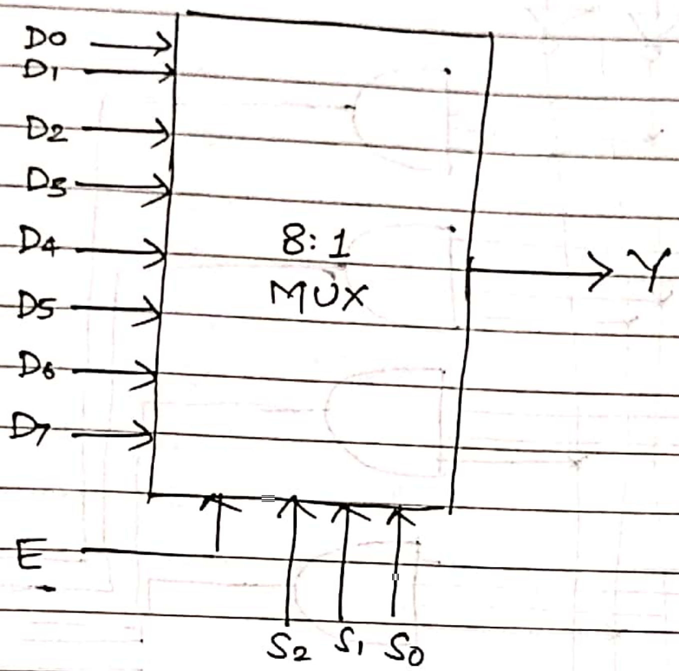
$$Y = \overline{S_1}\,\overline{S_0}\,D_0 + \overline{S_1}\,S_0 D_1 + S_1 \overline{S_0} D_2 + S_1 S_0 D_3$$

Circuit diagram:



8:1 Multiplexer

Inputs : 08
Outputs : 01
Select lines : 03

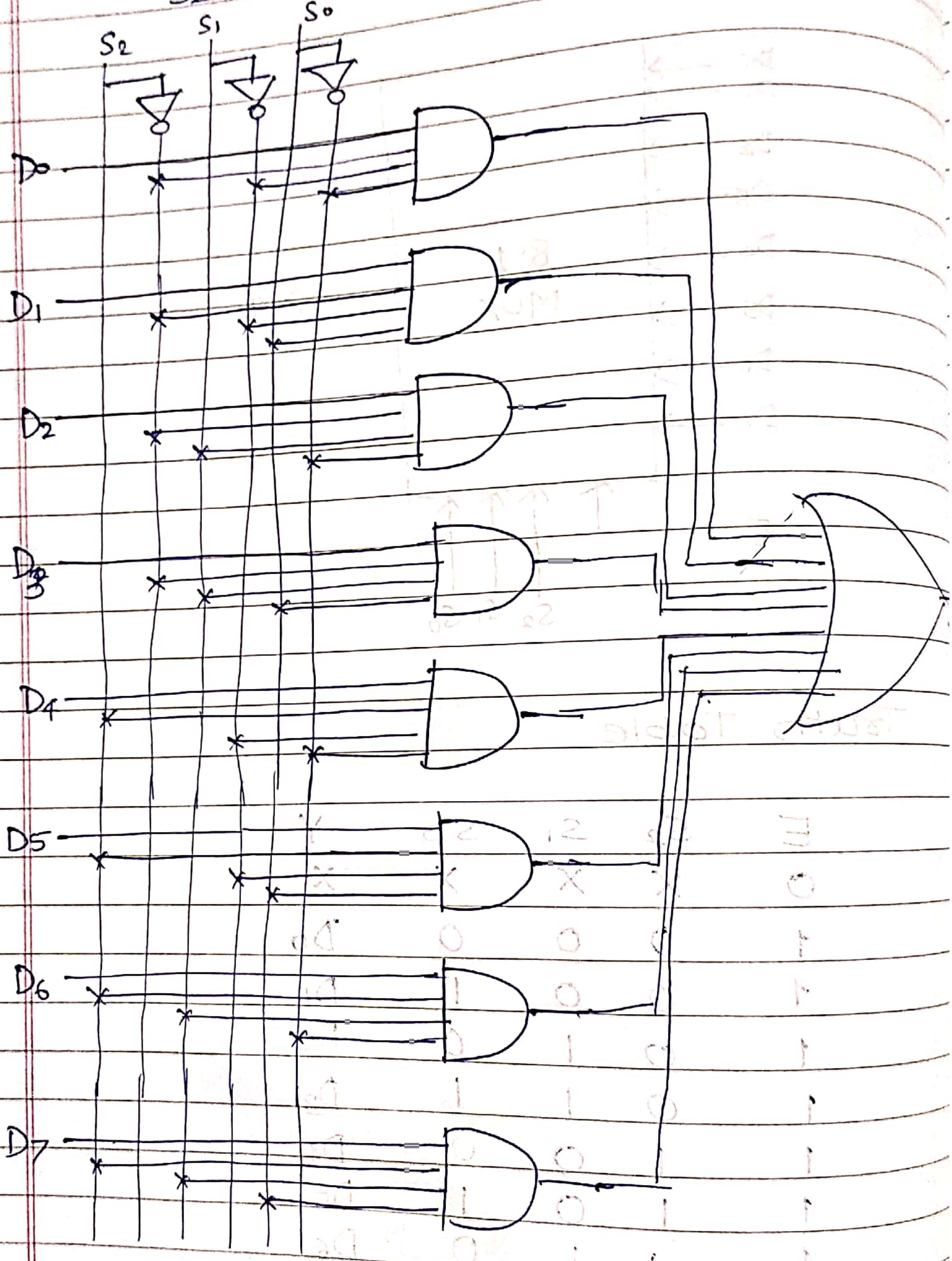


Truth Table

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

Equation:

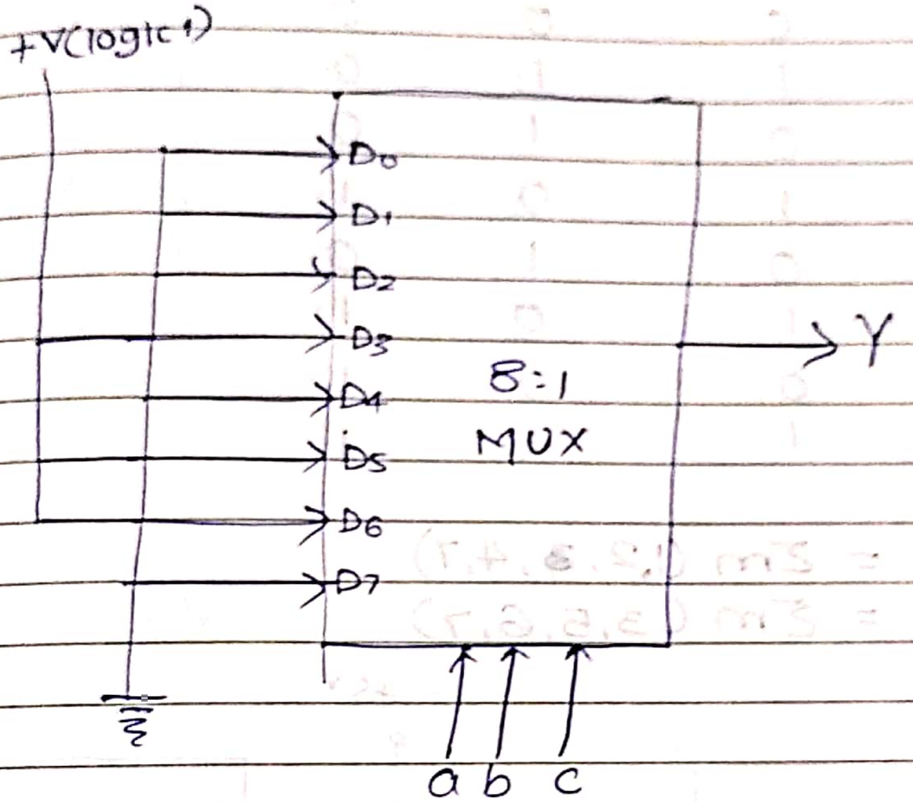
$Y = \overline{S_2}\,\overline{S_1}\,\overline{S_0} D_0 + \overline{S_2}\,\overline{S_1} S_0 D_1 + \overline{S_2} S_1 \overline{S_0} D_2 + \overline{S_2} S_1 S_0 D_3 + S_2 \overline{S_1}\,\overline{S_0} D_4 + S_2 \overline{S_1} S_0 D_5 + S_2 S_1 \overline{S_0} D_6 + S_2 S_1 S_0 D_7$



Multiplexer as a function generator

Implement the following function using 8:1 Multiplexer.

$$F(a,b,c) = \Sigma m(3,5,6)$$

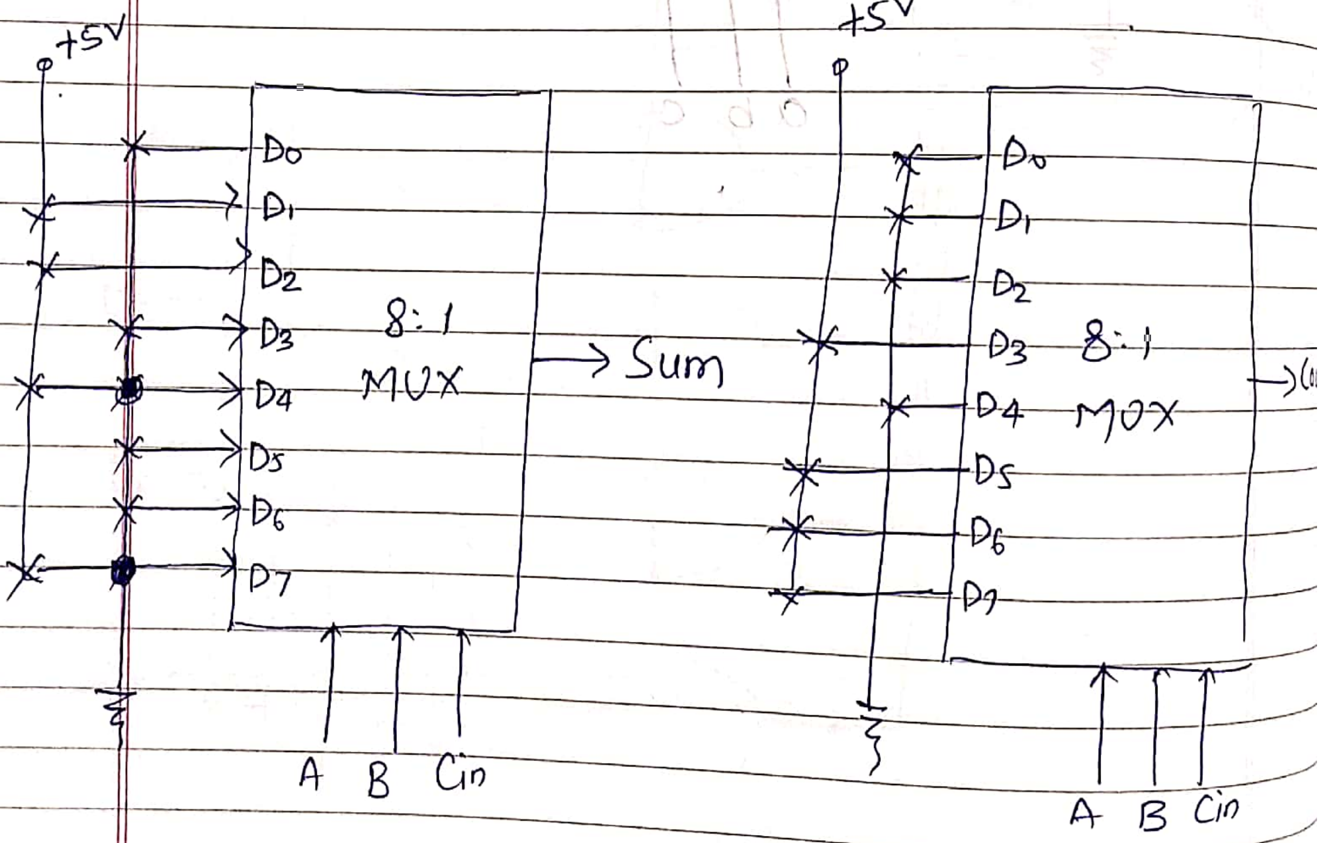


Implement full Adder using 8:1 Mux.

Truth Table:

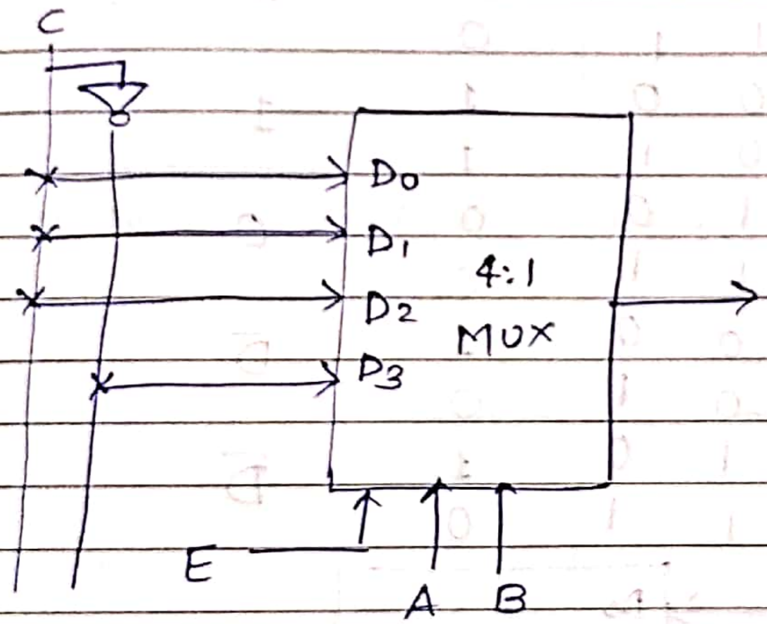
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Sum = \Sigma m(1,2,4,7)$$
$$Carry = \Sigma m(3,5,6,7)$$



Implement following function using 4:1 Mux
$F(A,B,C) = \Sigma m(1,3,5,6)$

	A	B	C	Y	Y
	0	0	0	0	
D_0	0	0	1	1	C
	0	1	0	0	
D_1	0	1	1	1	C
	1	0	0	0	
D_2	1	0	1	1	C
	1	1	0	1	
D_3	1	1	1	0	$\overline{C}$

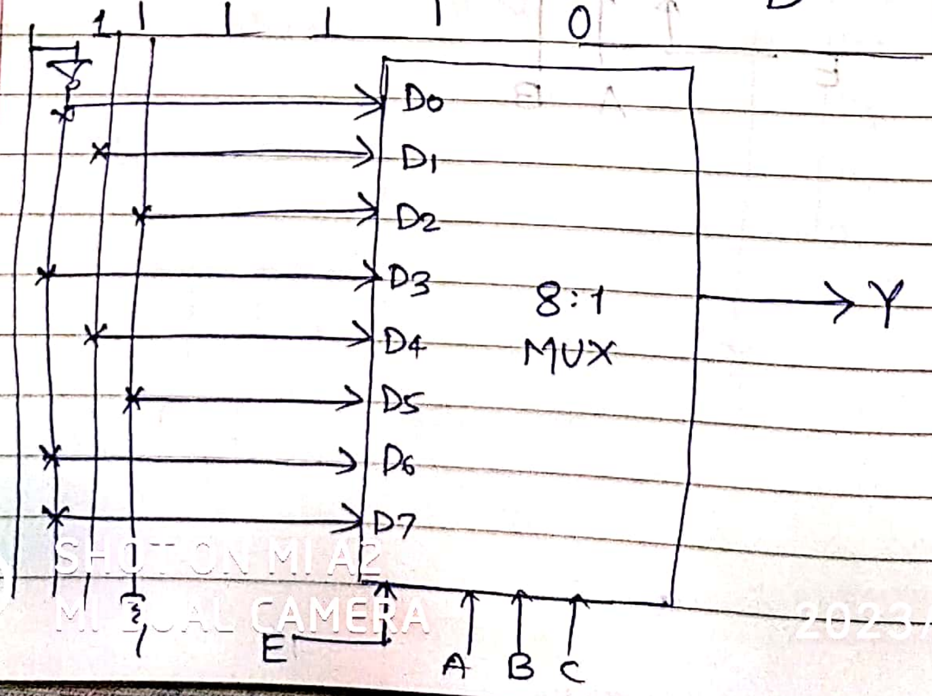


Implement expression using 8:1 MUX

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

Truth Table

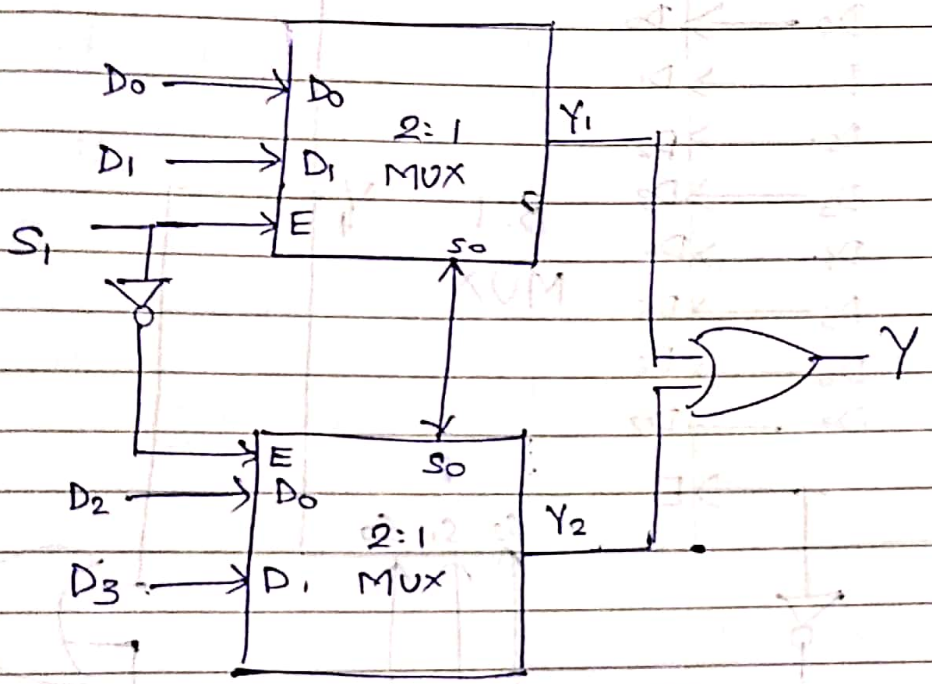
A	B	C	D	Y	Y
0	0	0	0	1	$\bar{D}$
0	0	0	1	0	
0	0	1	0	1	1
0	0	1	1	1	
0	1	0	0	0	0
0	1	0	1	0	
0	1	1	0	1	$\bar{D}$
0	1	1	1	0	
1	0	0	0	1	1
1	0	0	1	1	
1	0	1	0	0	0
1	0	1	1	0	
1	1	0	0	1	$\bar{D}$
1	1	0	1	0	
1	1	1	0	1	$\bar{D}$
1	1	1	1	0	



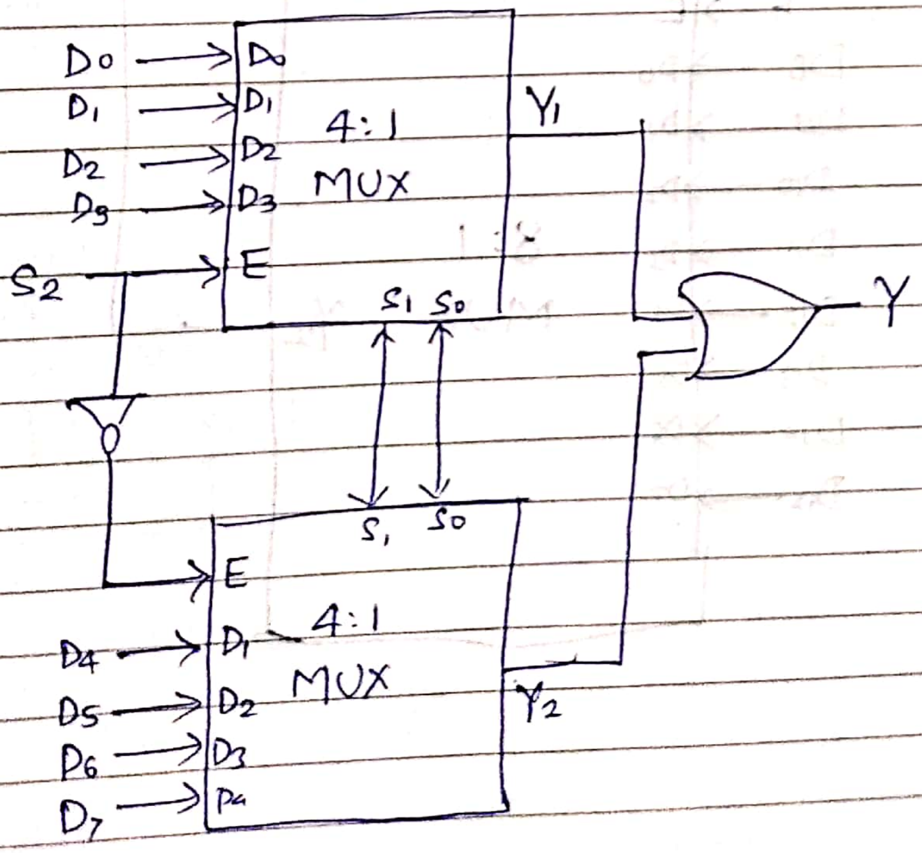
Date
Page

Cascading Multiplexer

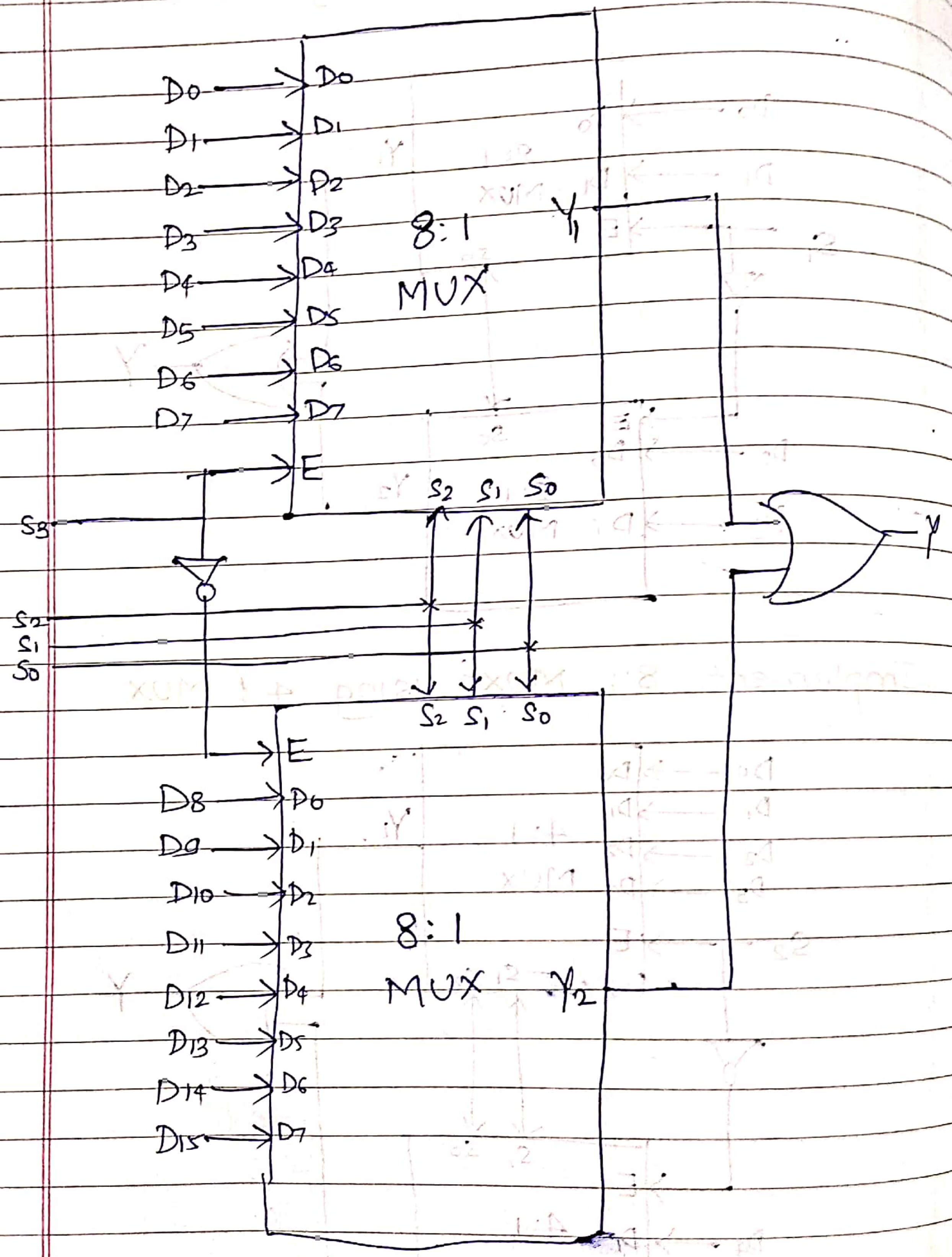
Implement 4:1 MUX using 2:1 MUX



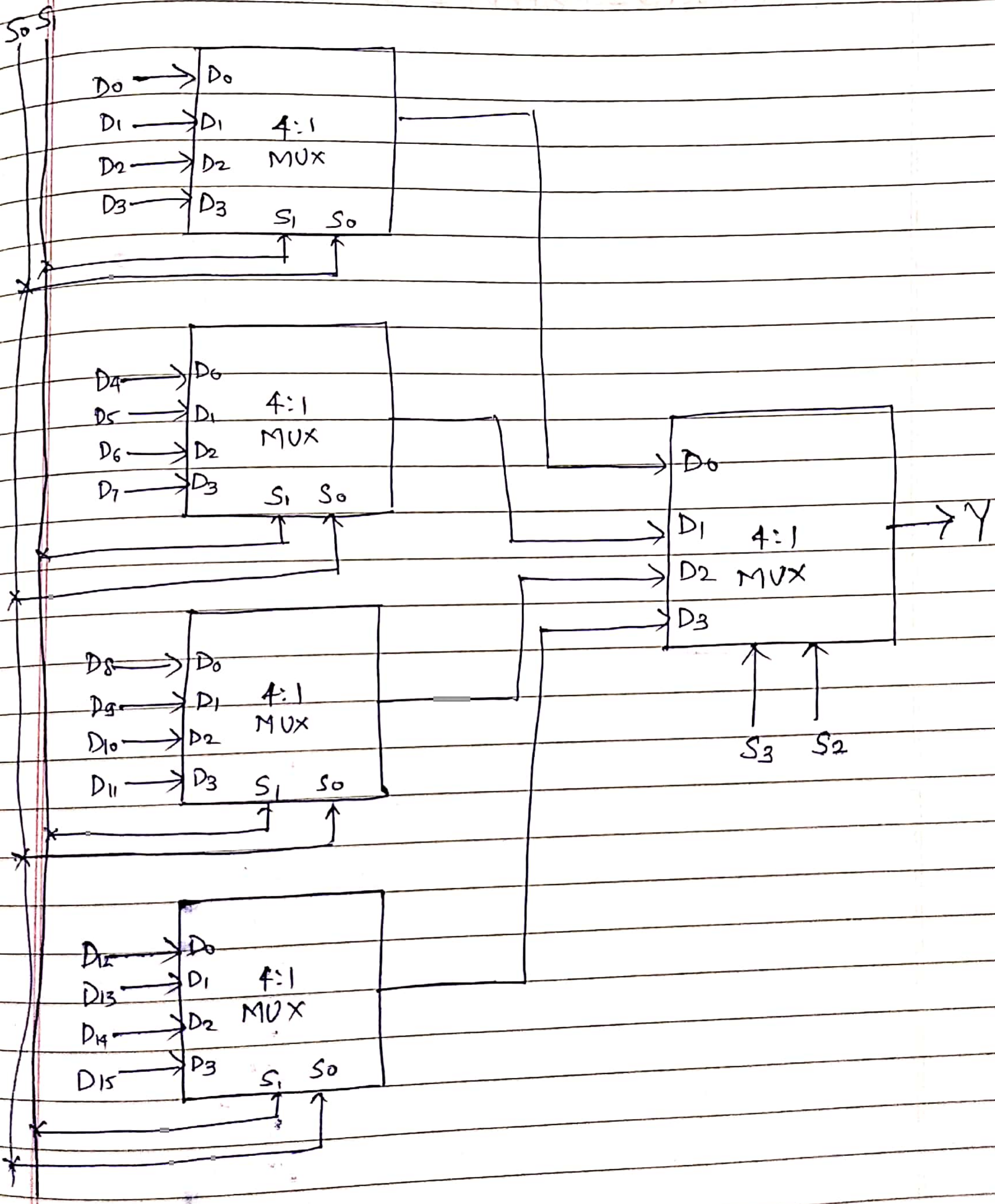
Implement 8:1 MUX using 4:1 MUX



Implement 16:1 Multiplexer using 8:1 MUX



Design 16:1 Multiplexer using 4:1 MUX



Implement function using 4:1 MUX.

$F(A,B,C) = \Sigma m(1,3,5,6)$

Scanned with CamScanner

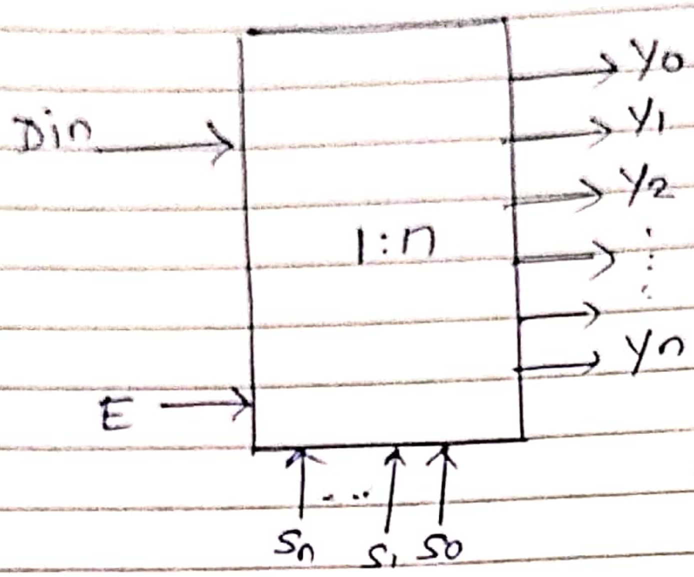
Implement function using 8:1 MUX

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

Scanned with CamScanner

Demultiplexer

Demultiplexer is one input and many output

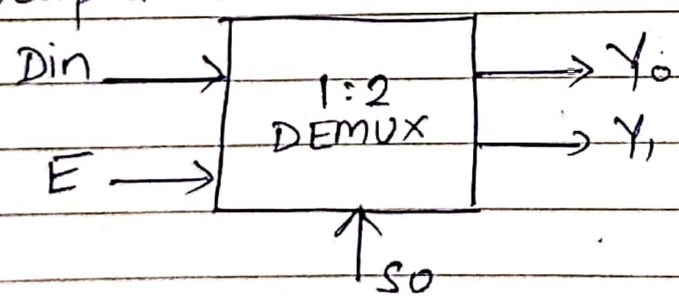


Type of demultiplexers

1. 1:2 demultiplexer
2. 1:4 demultiplexer
3. 1:8 demultiplexer
4. 1:16 demultiplexer

1:2 Demultiplexer

input - one   Select line - 1
output - two

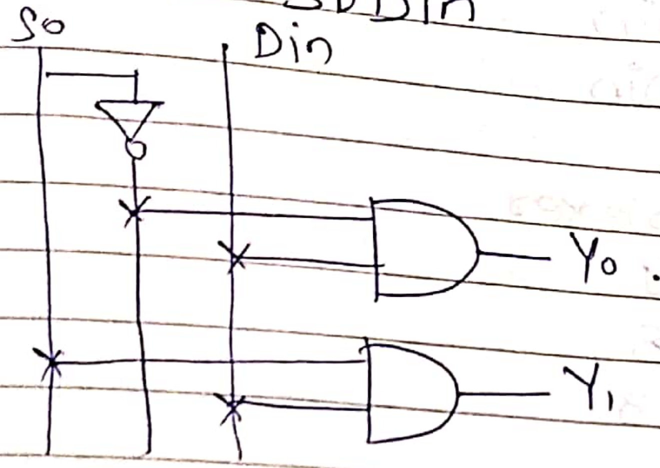


E	S_0	Y_0	Y_1
0	X	0	0
1	0	Din	0
1	1	0	Din

The logical expression

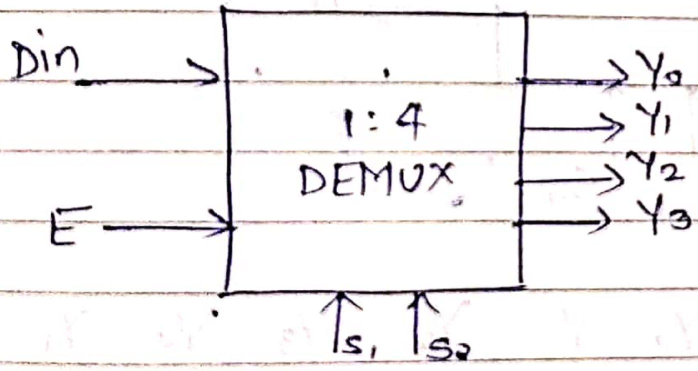
$$Y_0 = \overline{S_0}\,D_{in}$$

$$Y_1 = S_0\,D_{in}$$



1:14 Demultiplexer

inputs - 1
outputs - 4
selectline - 2



E	S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	x	x	0	0	0	0
1	0	0	Din	0	0	0
1	0	1	0	Din	0	0
1	1	0	0	0	Din	0
1	1	1	0	0	0	Din

The logical expression of the term of Y

$Y_0 = \bar{S_1}\bar{S_0}\,Din$

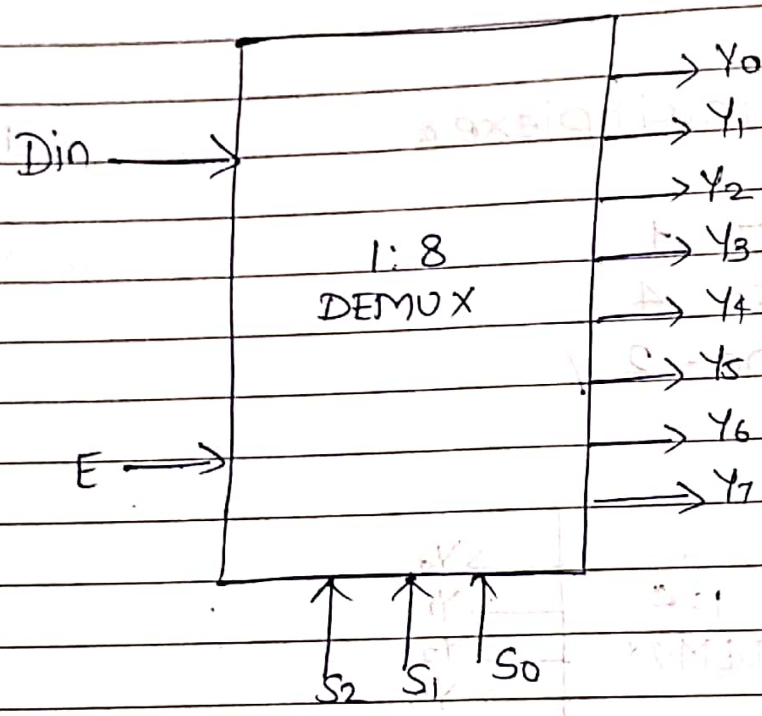
$Y_1 = \bar{S_1}S_0\,Din$

$Y_2 = S_1\bar{S_0}\,Din$

$Y_3 = S_1 S_0\,Din$

1:8 Demultiplexer

inputs - 1
outputs - 8
select lines - 3



E	S_2	S_1	S_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	Din							
1	0	0	1		Din						
1	0	1	0			Din					
1	0	1	1				Din				
1	1	0	0					Din			
1	1	0	1						Din		
1	1	1	0							Din	
1	1	1	1								Din

Scanned with CamScanner

Logical Expression

$Y_0 = \bar{S_2}\bar{S_1}\bar{S_0}\, D_{in}$

$Y_1 = \bar{S_2}\bar{S_1}S_0\, D_{in}$

$Y_2 = \bar{S_2}S_1\bar{S_0}\, D_{in}$

$Y_3 = \bar{S_2}S_1S_0\, D_{in}$

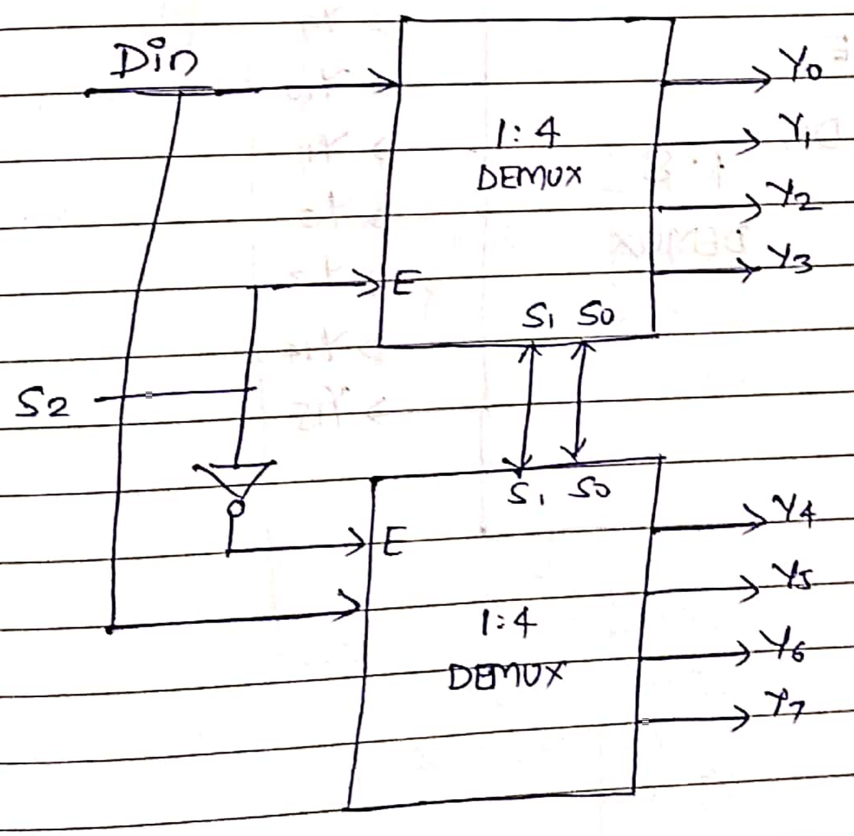
$Y_4 = S_2\bar{S_1}\bar{S_0}\, D_{in}$

$Y_5 = S_2\bar{S_1}S_0\, D_{in}$

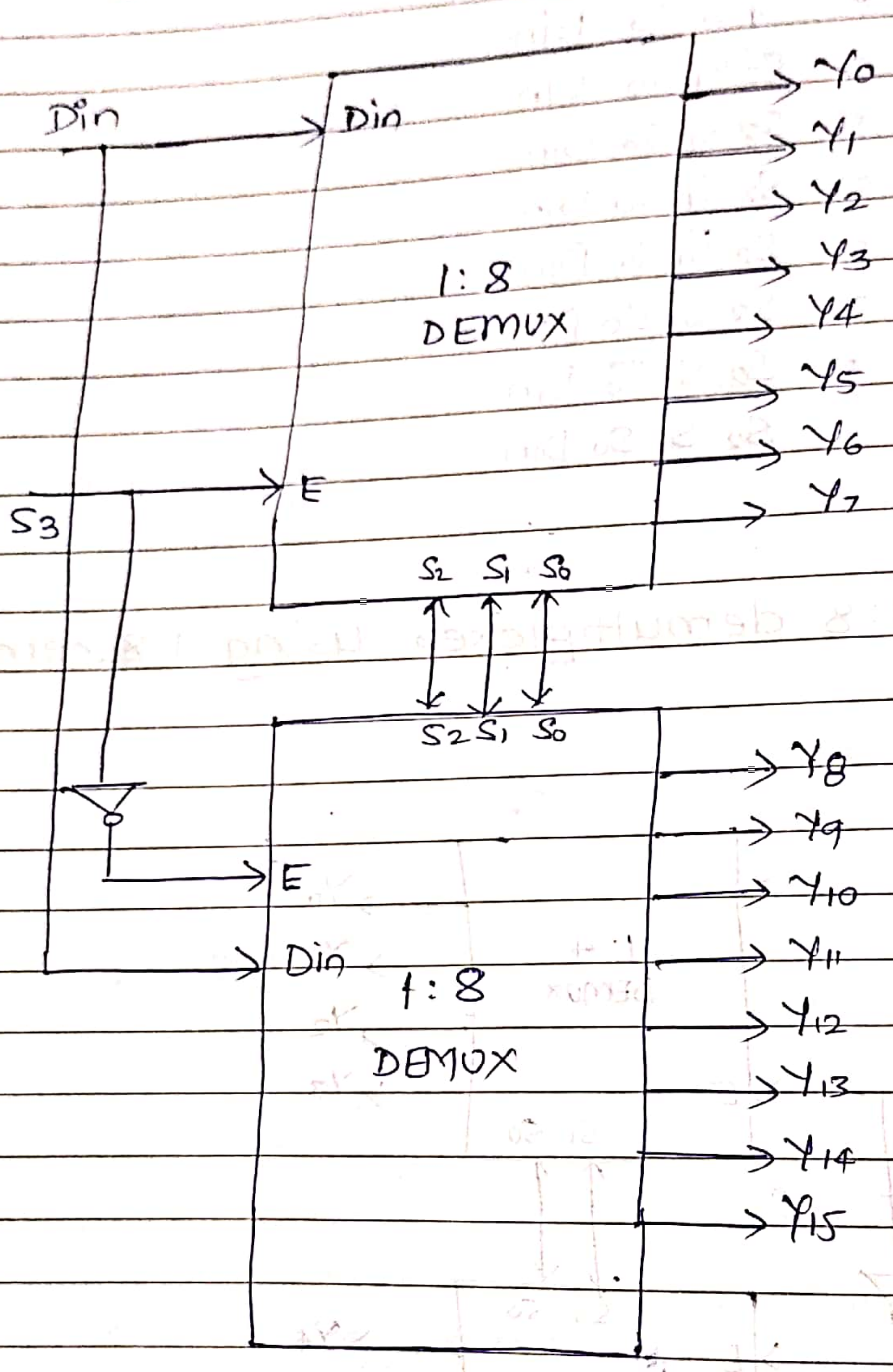
$Y_6 = S_2S_1\bar{S_0}\, D_{in}$

$Y_7 = S_2S_1S_0\, D_{in}$

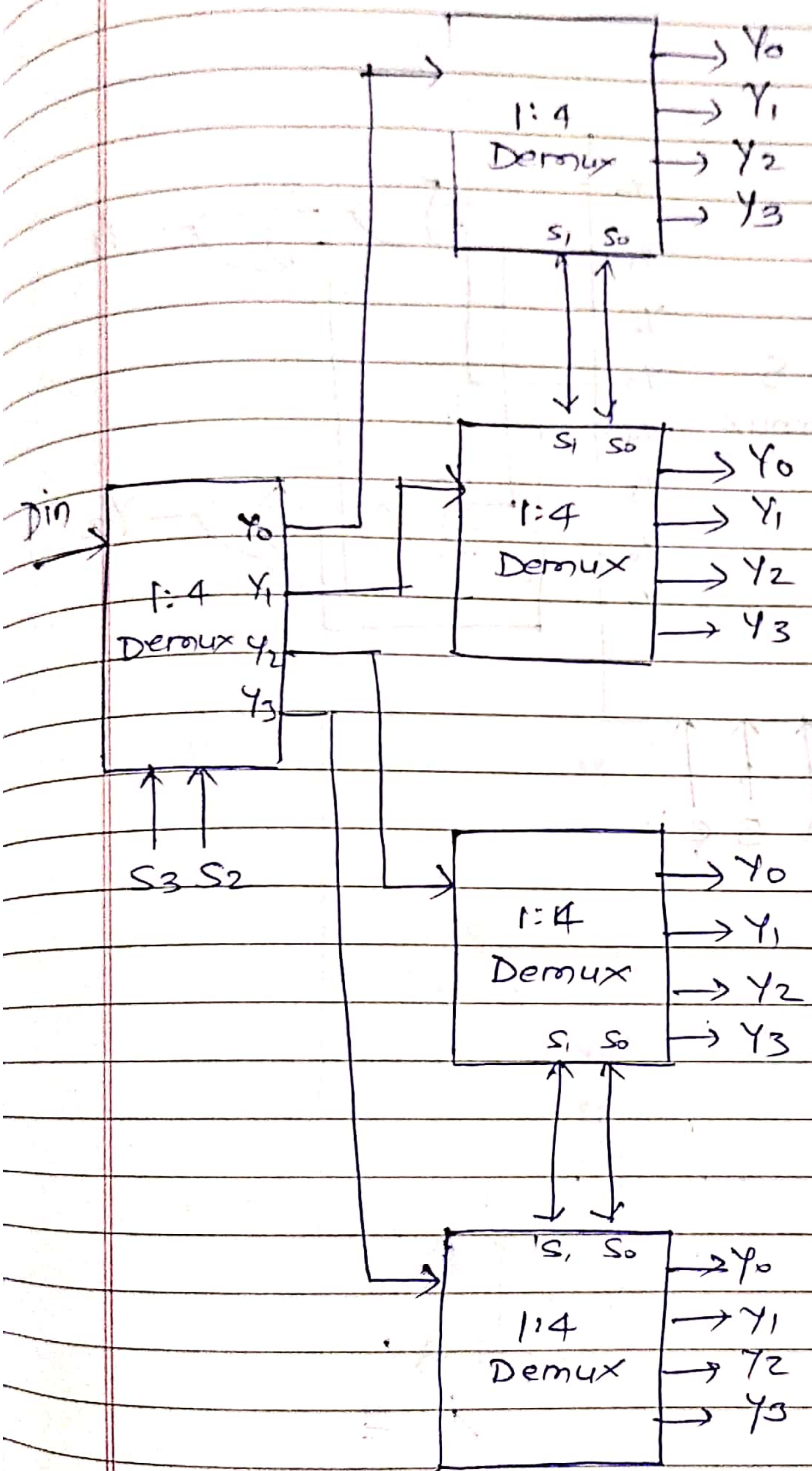
Design 1:8 demultiplexer using 1:4 demux



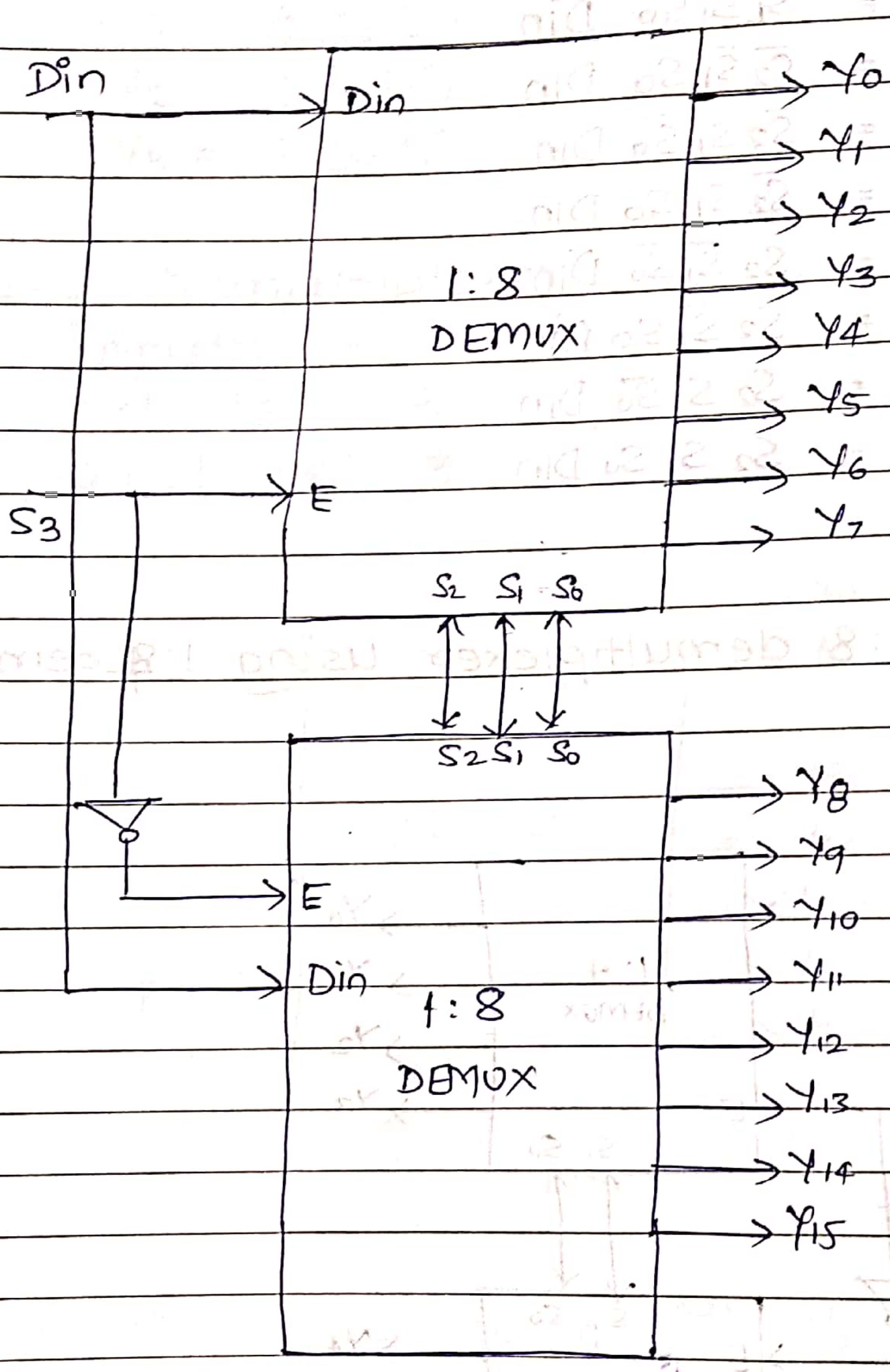
Design 1:16 demultiplexer using 1:8 Demux



Design 1:16 Demultiplexer using 1:4 Demux

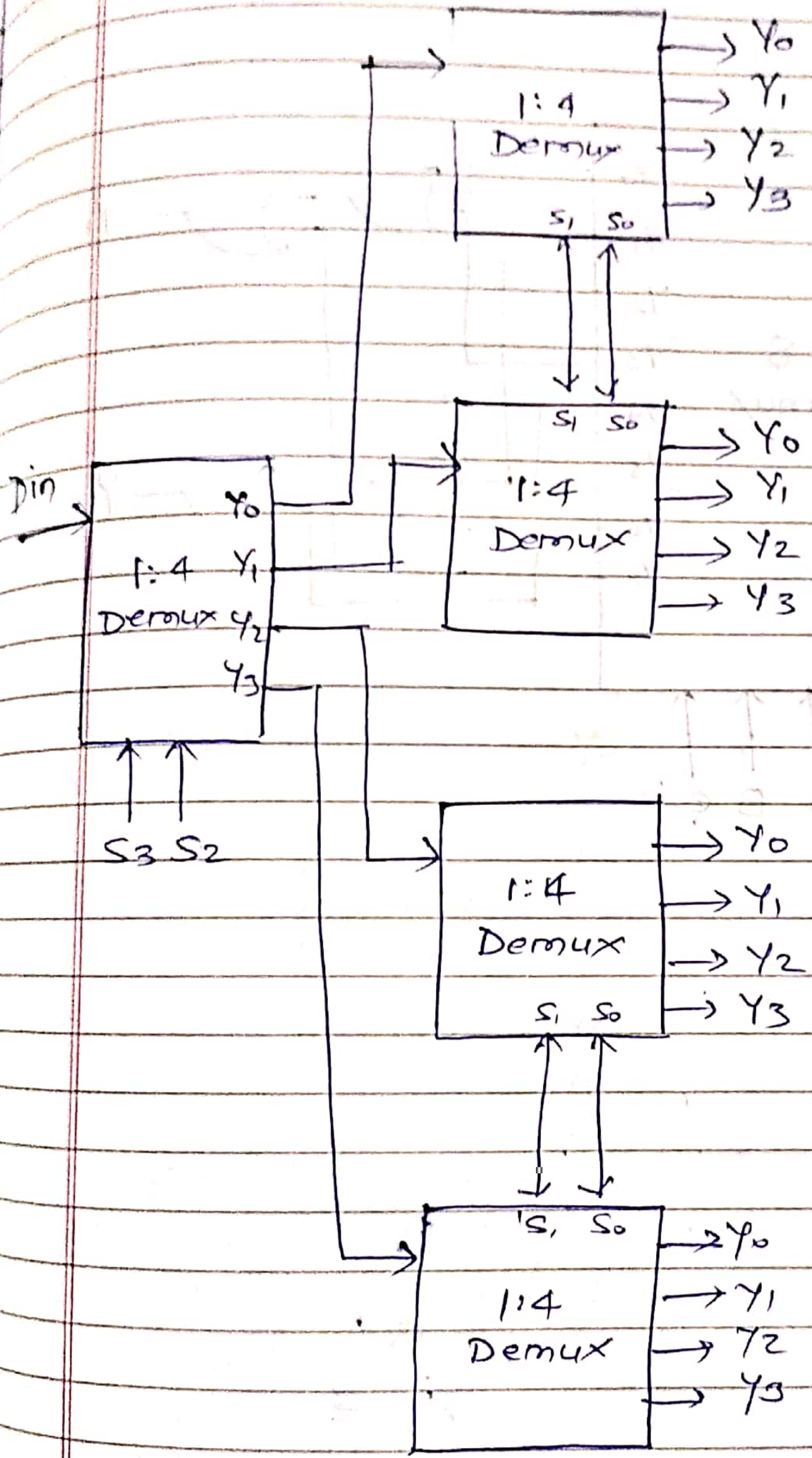


Design 1:16 demultiplexer using 1:8 Demux



Scanned with CamScanner

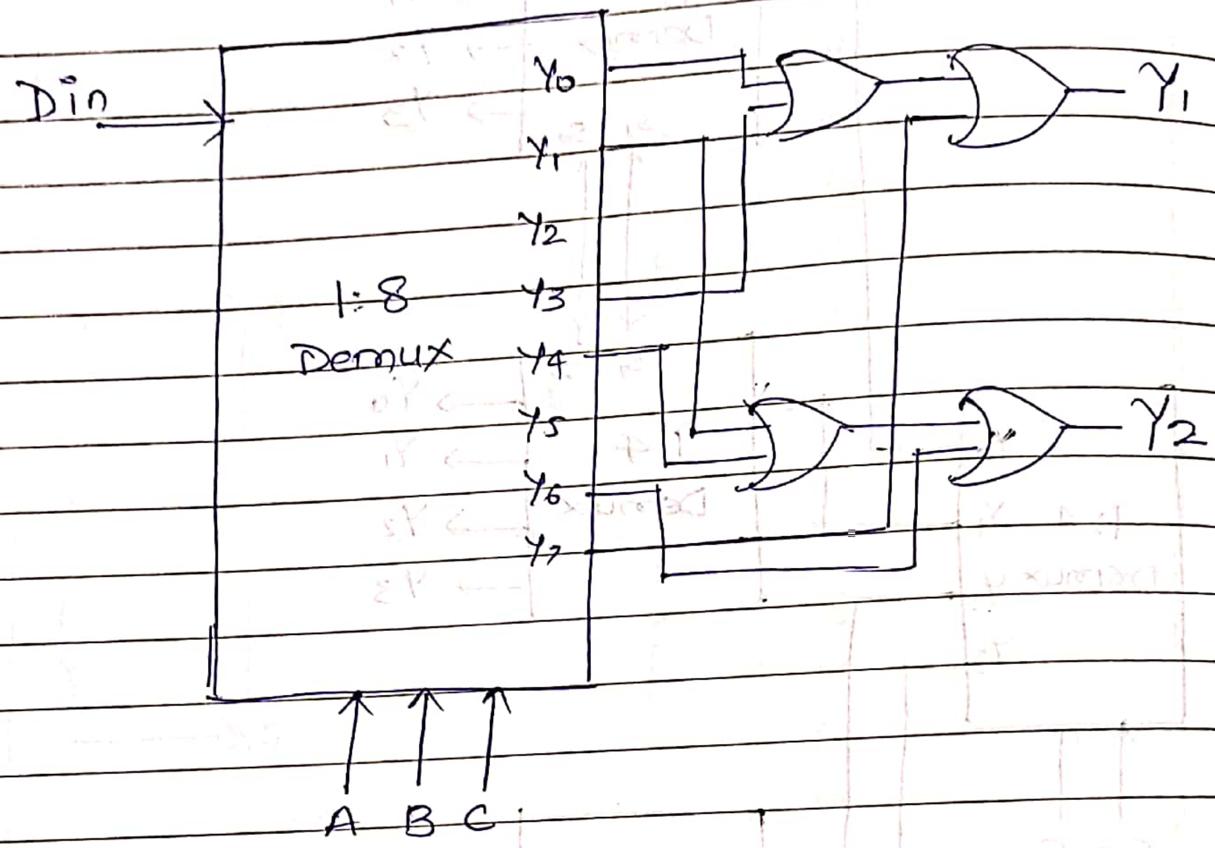
Design 1:16 Demultiplexer using 1:4 Demux



Implement following function using Demux

$Y_1 (A, B, C) = \Sigma m (0, 3, 7)$

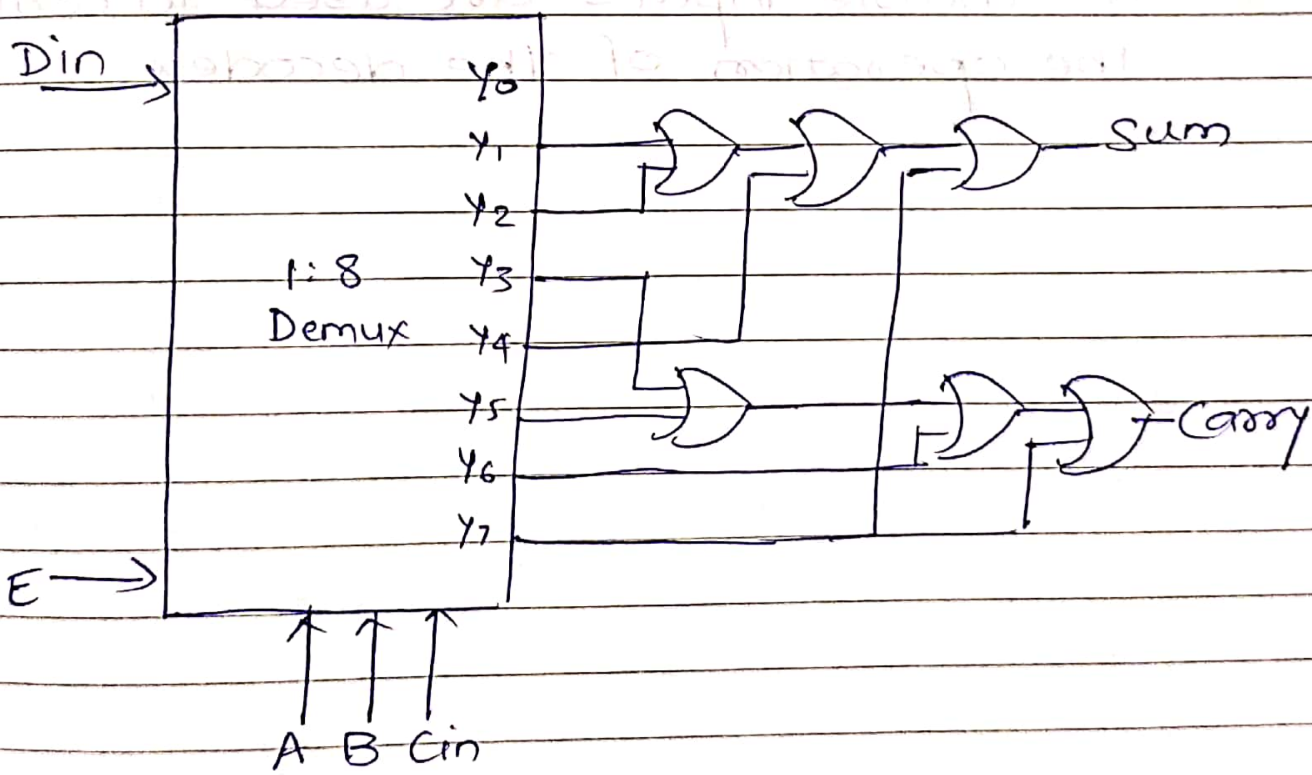
$Y_2 (A, B, C) = \Sigma m (1, 4, 6)$



Implement full Adder using demultiplexer

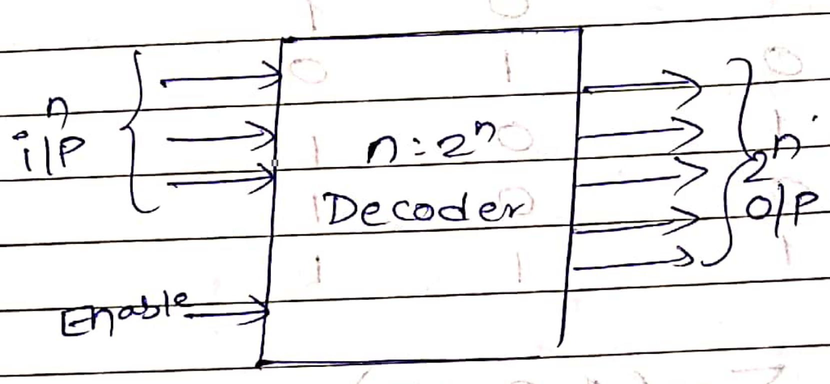
A	B	Cin	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Sum = \Sigma m(1,2,4,7)$$
$$Carry = \Sigma m(3,5,6,7)$$



Decoder

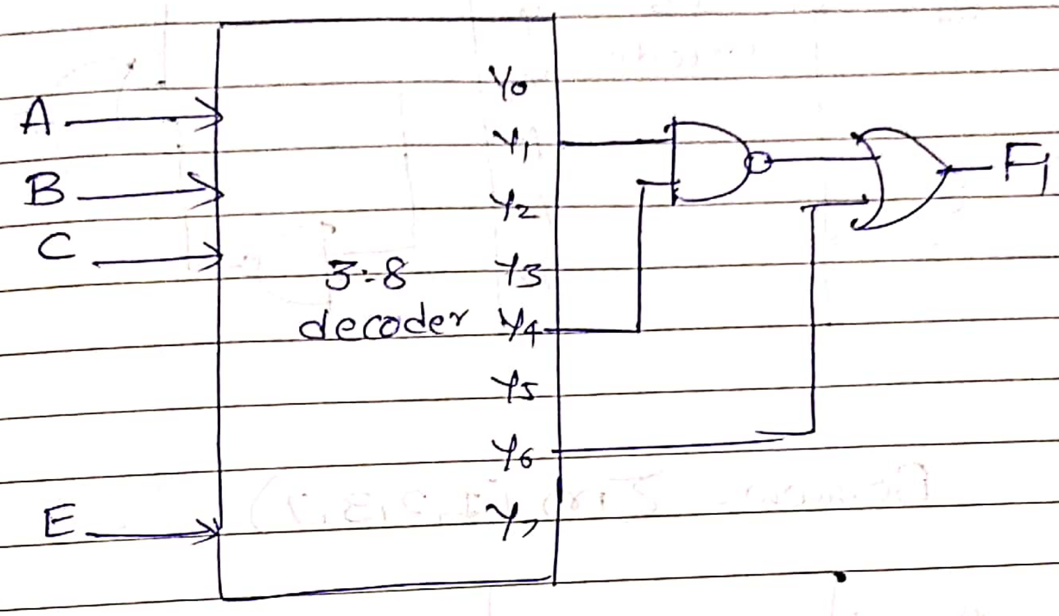
A decoder is a combinational logic circuit having multiple inputs & multiple output.



The decoder has n inputs & 2^n output

The Enable inputs are used to control the operation of the decoder

Implement Boolean function
$F_1 = \Sigma m(1,4,6)$ using 3:8 decoder

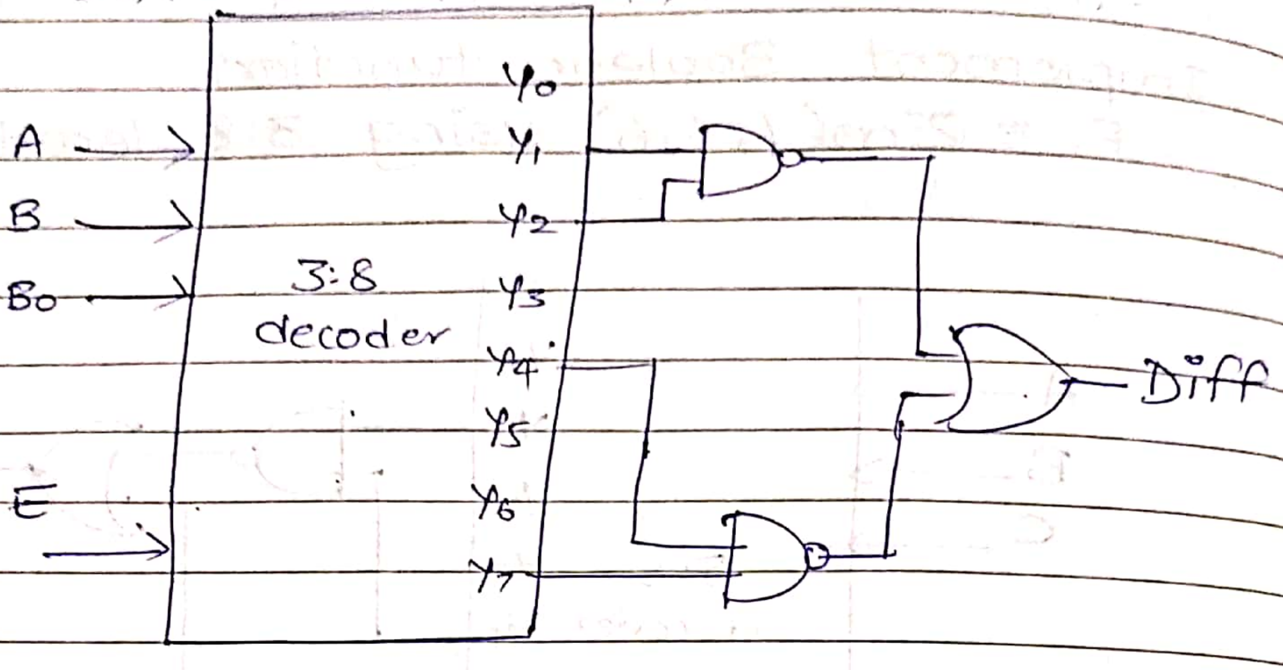


Implement full subtractor using 3:8 deoder

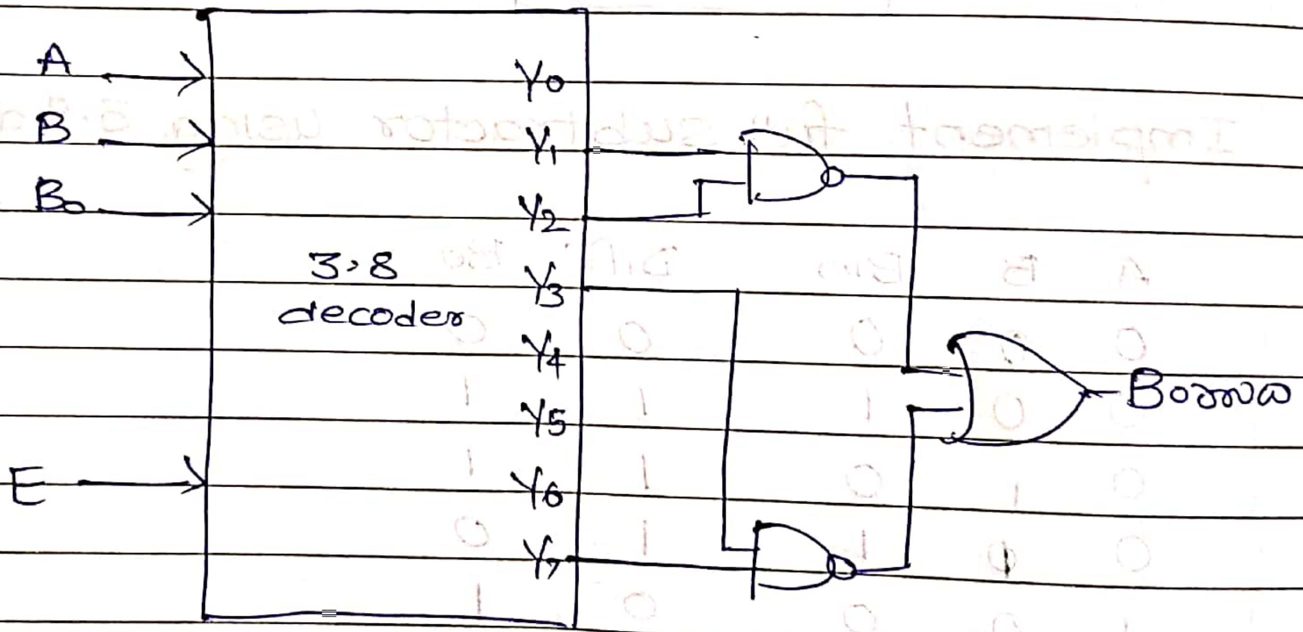
A	B	Bin	Diff	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	ϕ	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Diff = $\Sigma m(1,2,4,7)$
Bout = $\Sigma m(1,2,3,7)$

Diff = $\Sigma m(1,2,4,7)$



Borrow = $\Sigma m(1,2,3,7)$



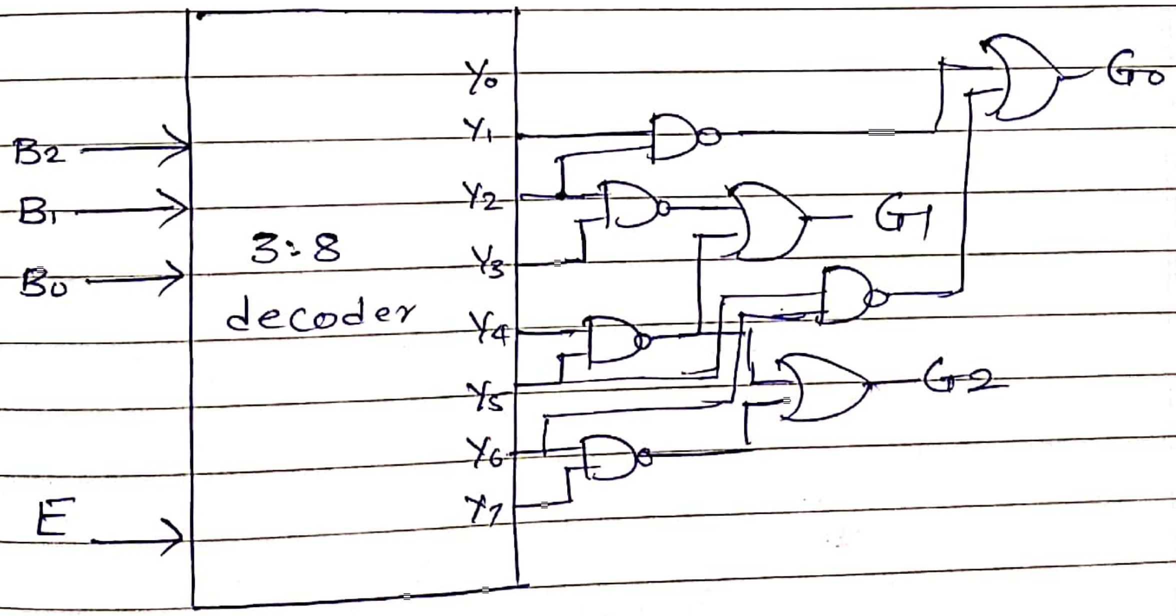
Design 3 bit binary to 3 bit gray code converter using 3:8 decoder.

I/P			O/P		
B_2	B_1	B_0	G_2	G_1	G_0
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

$G_2 = \Sigma m(4,5,6,7)$

$G_1 = \Sigma m(2,3,4,5)$

$G_0 = \Sigma m(1,2,5,6)$



Design one bit magnitude comparator

Truth Table

I/P		O/P		
A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Step 2

K-Map for A>B

A \ B	$\bar{B}$	B
$\bar{A}$	(0)	(1)
A	1 (2)	(3)

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

K-Map for A=B

A \ B	$\bar{B}$	B
$\bar{A}$	1 (0)	(1)
A	(2)	1 (3)

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

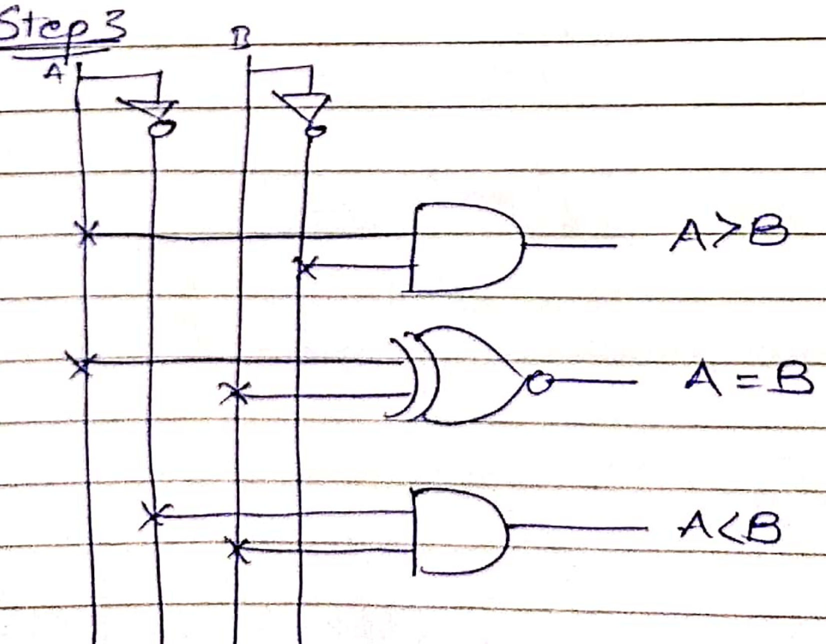
$= \overline{A \oplus B}$

K-Map for A<B

A \ B	$\bar{B}$	B
$\bar{A}$	(0)	1 (1)
A	(2)	(3)

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

Step 3



Scanned with CamScanner

Design 2 bit Magnitude comparator

Step1
Truth table:

i/p				o/p		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Step2 K-Map for A>B

$A_1A_0 \backslash B_1B_0$	$\overline{B_1}\,\overline{B_0}$	$\overline{B_1}B_0$	B_1B_0	$B_1\overline{B_0}$
$\overline{A_1}\,\overline{A_0}$	(0)	(1)	(3)	(2)
$\overline{A_1}A_0$	1 (4)	(5)	(7)	(6)
A_1A_0	1 (12)	1 (13)	(15)	1 (14)
$A_1\overline{A_0}$	1 (8)	1 (9)	(11)	(10)

$$A>B = A_0\overline{B_1}\,\overline{B_0} + A_1A_0\overline{B_0} + A_1\overline{B_1}$$

K-Map for A<B

$A_1A_0 \backslash B_1B_0$

	1	1	1
		1	1
		1	

K-Map for A=B

$A_1A_0 \backslash B_1B_0$

1			
	1		
		1	
			1

$Y = A<B = \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0 + \bar{A}_1B_1$

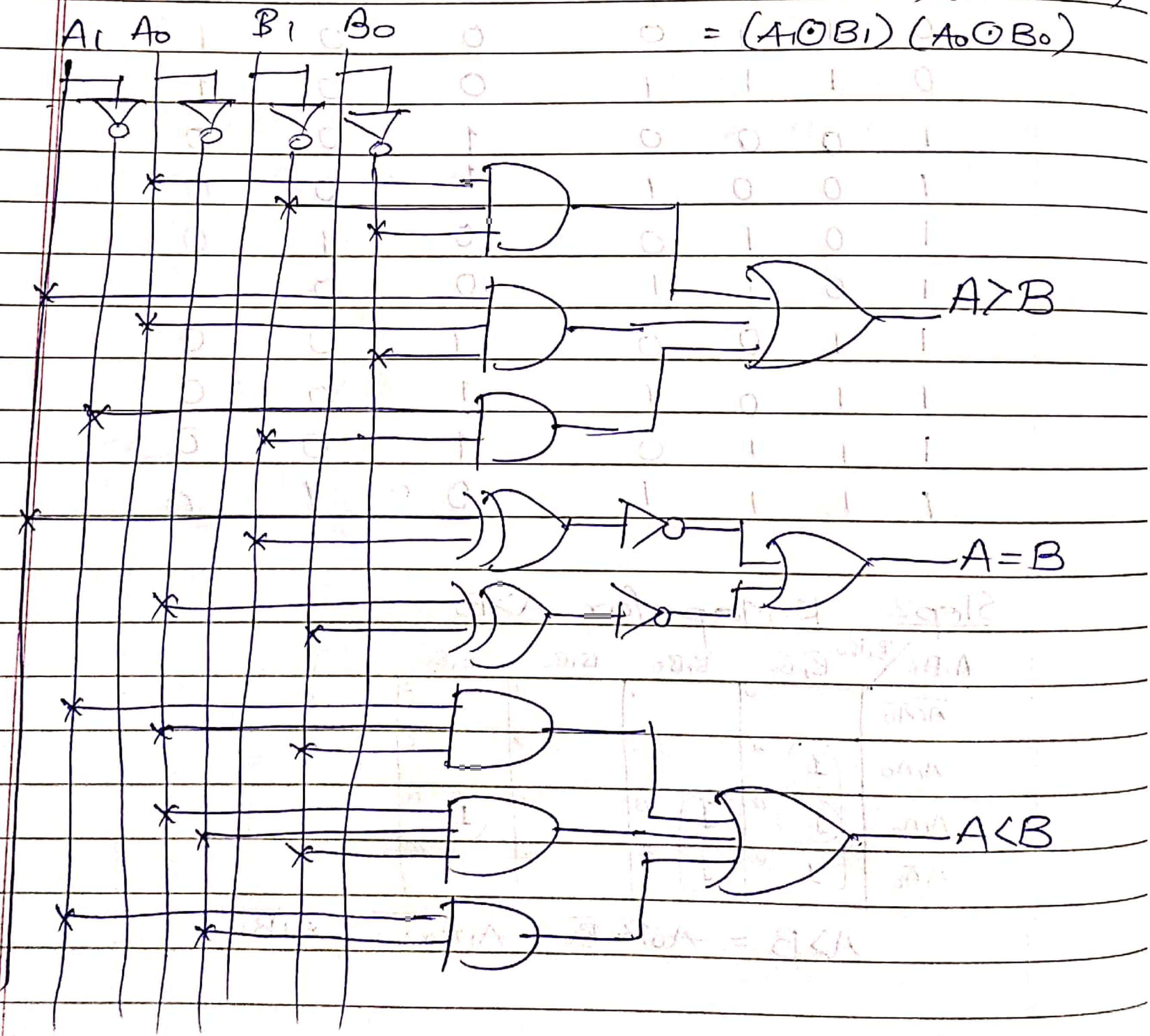
$Y_{A=B} =$

$= \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$

$= \bar{A}_1\bar{B}_1(\bar{A}_0\bar{B}_0 + A_0B_0) + A_1B_1(A_0B_0 + \bar{A}_0\bar{B}_0)$

$= (\bar{A}_1\bar{B}_1 + A_1B_1)(\bar{A}_0\bar{B}_0 + A_0B_0)$

$= (A_1 \odot B_1)(A_0 \odot B_0)$



Design Even parity generator & checker.

Truth Table

Truth-table for even parity generator

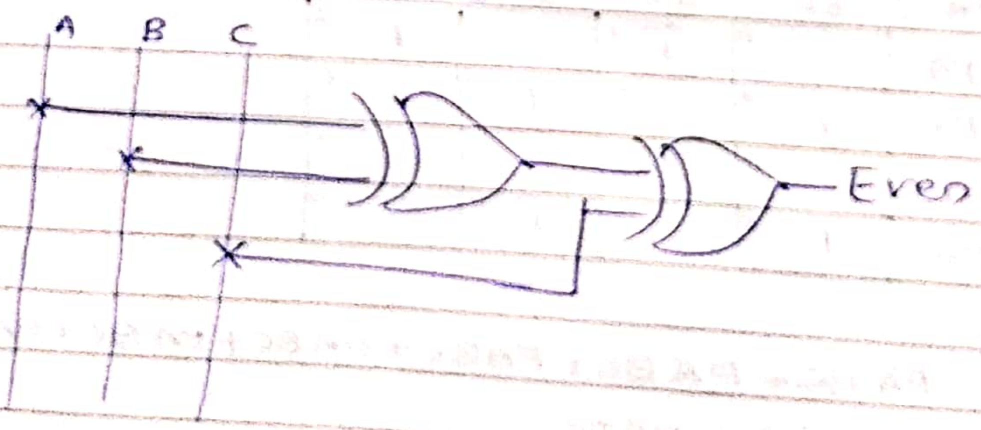
A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Step 2

K-Map for P

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1 1	3	1 2
A	1 4	5	1 7	6

$$P = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$$
$$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$
$$= A \oplus B \oplus C$$



Step 3

Truth Table for even parity checker

P	A	B	C	PEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Step 4

K-Map for PEC

PA \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{P}\bar{A}$	0	1 1	3	1 2
$\bar{P}A$	1 4	5	1 7	6
PA	12	1 13	15	1 14
$P\bar{A}$	1 8	9	1 11	10

$$PEC = \bar{P}\bar{A}\bar{B}C + \bar{P}\bar{A}B\bar{C} + \bar{P}A\bar{B}\bar{C} + \bar{P}ABC + PA\bar{B}C + PAB\bar{C} +$$
$$P\bar{A}\bar{B}\bar{C} + P\bar{A}BC$$

$= \bar{P}\bar{A}(\bar{B}C + B\bar{C}) + \bar{P}A(\bar{B}\bar{C} + BC) + PA(\bar{B}C + B\bar{C}) + P\bar{A}(\bar{B}\bar{C} + BC)$

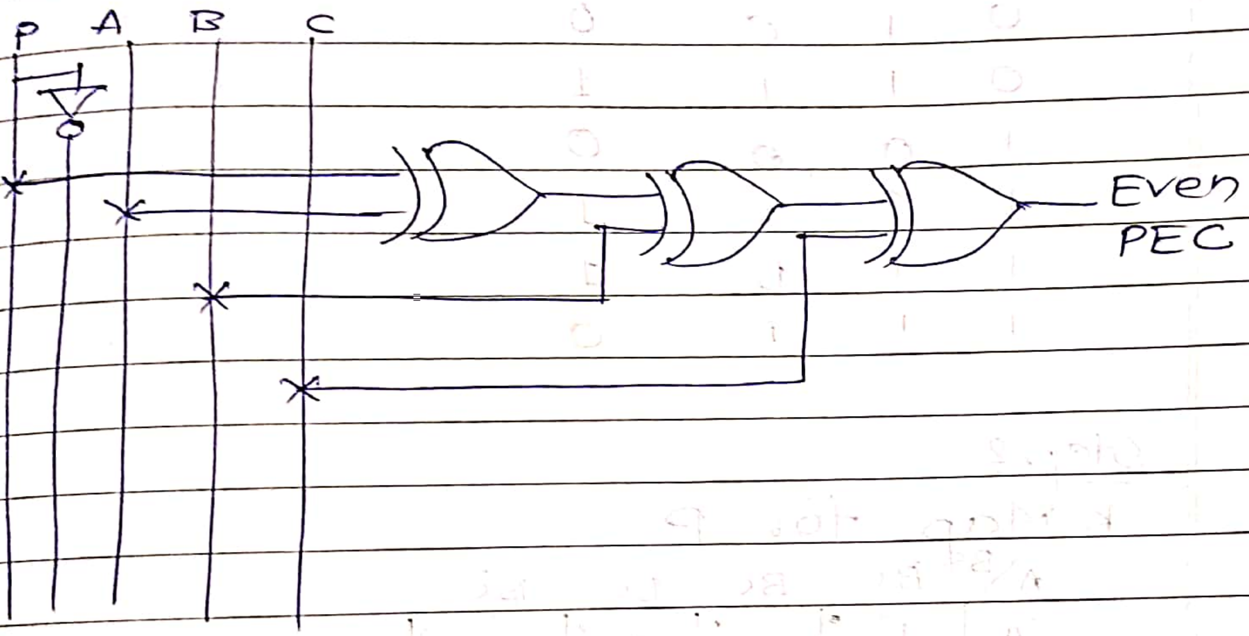
$= \bar{P}\bar{A}(B \oplus C) + \bar{P}A(\overline{B \oplus C}) + PA(B \oplus C) + P\bar{A}(\overline{B \oplus C})$

$= (B \oplus C)(\bar{P}\bar{A} + PA) + (\overline{B \oplus C})(\bar{P}A + P\bar{A})$

$= (B \oplus C)(\overline{P \oplus A}) + (\overline{B \oplus C})(P \oplus A)$

$= B \oplus C \oplus P \oplus A$

$= A \oplus B \oplus C \oplus P$



Design odd parity generator & checker.

Step 1

Truth table for odd parity generator

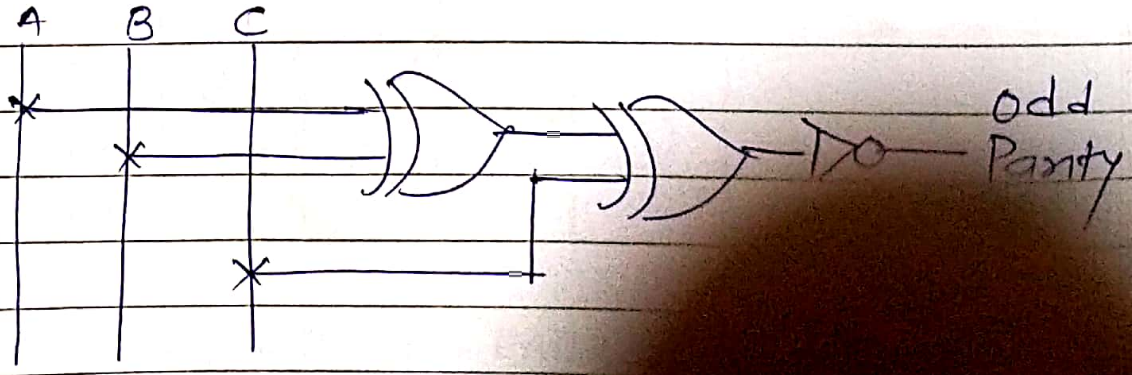
A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	D	1
1	1	1	0

Step 2

K-Map for P

A \ BC	$\bar{B}\bar{C}$	BC	BC	$B\bar{C}$
$\bar{A}$	1 ^{0}	^{1}	1 ^{3}	^{2}
A	^{4}	1 ^{5}	D ^{7}	1 ^{6}

$$P = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$$
$$= \bar{A}(\bar{B}\bar{C} + BC) + A(\bar{B}C + B\bar{C})$$
$$= \bar{A}(\overline{B \oplus C}) + A(B \oplus C)$$
$$= \overline{A \oplus B \oplus C}$$



Step 4

Truth Table for odd parity checker

P	A	B	C	PEC
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Step 5

K-Map for PEC

PA \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{P}\bar{A}$	1 (0)	(1)	1 (3)	(2)
$\bar{P}A$	(4)	1 (5)	(7)	1 (6)
PA	1 (12)	(13)	1 (15)	(14)
$P\bar{A}$	(8)	1 (9)	(11)	1 (10)

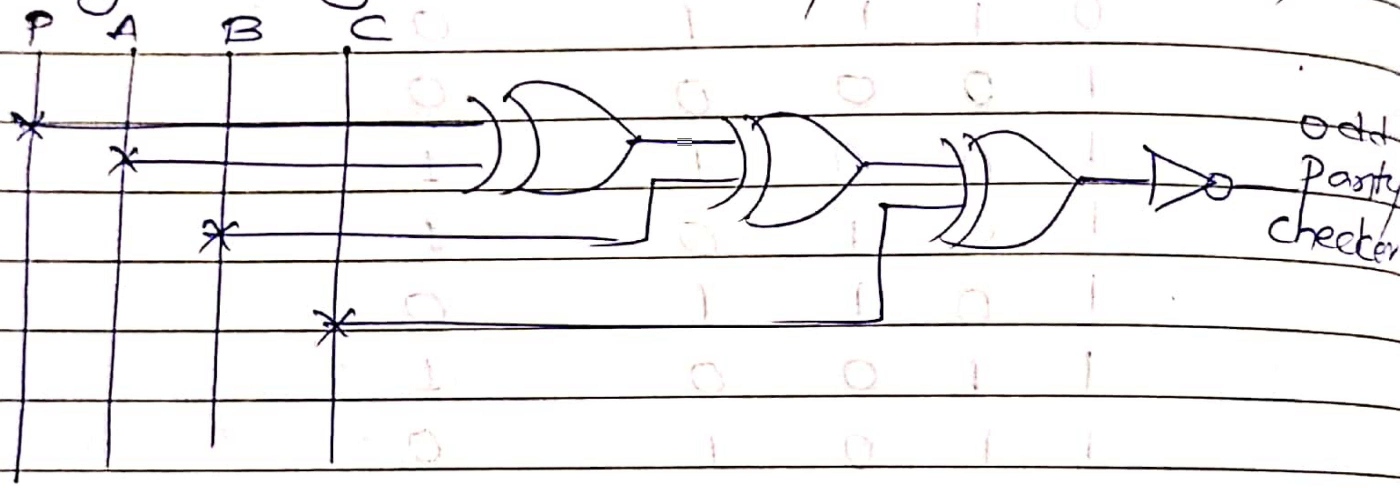
$$PEC = \bar{P}\bar{A}\bar{B}\bar{C} + \bar{P}\bar{A}BC + \bar{P}A\bar{B}C + \bar{P}AB\bar{C} + PA\bar{B}\bar{C} + PABC + P\bar{A}\bar{B}C + P\bar{A}B\bar{C}$$

$$= \bar P\bar A\cdot(\bar B\bar C + BC) + \bar P A(\bar B C + B\bar C) +$$
$$PA(\bar B\bar C + BC) + P\bar A(\bar B C + B\bar C)$$

$$= \bar P\bar A(\overline{B\oplus C}) + \bar P A(B\oplus C) + PA(\overline{B\oplus C}) + P\bar A(B\oplus C)$$
$$= (\overline{B\oplus C})(\bar P\bar A + PA) + (B\oplus C)(\bar P A + P\bar A)$$
$$= (\overline{B\oplus C})(\overline{P\oplus A}) + (B\oplus C)(P\oplus A)$$
$$= \overline{P\oplus A\oplus B\oplus C}$$

Step 6

Logic diagram of parity checker (odd)



Programmable Logic Devices

A programmable logic device is an integrated circuit capable of implementing logic functions.

The main advantage of PLD is that they can replace a number of circuits in many applications.

Classification of PLD'S

1. Programmable Logic Array (PLA)
2. Programmable Array Logic (PAL)
3. Programmable Read Only Memory (PROM)

Programmable Logic Array (PLA)

PLA is a logic device with both AND and OR arrays programmable.

i/p → [Programmable AND array] → [Programmable OR Array] → o/p

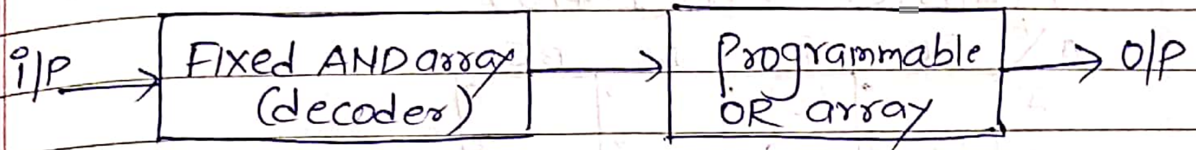
Programmable Array Logic (PAL)

PAL is a logic device with programmable AND array & a fixed OR array.

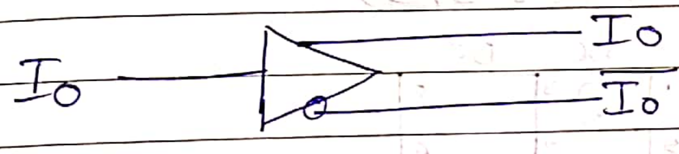
i/p → [Programmable AND array] → [Fixed OR array] → o/p

Programmable Read Only Memory (PROM)

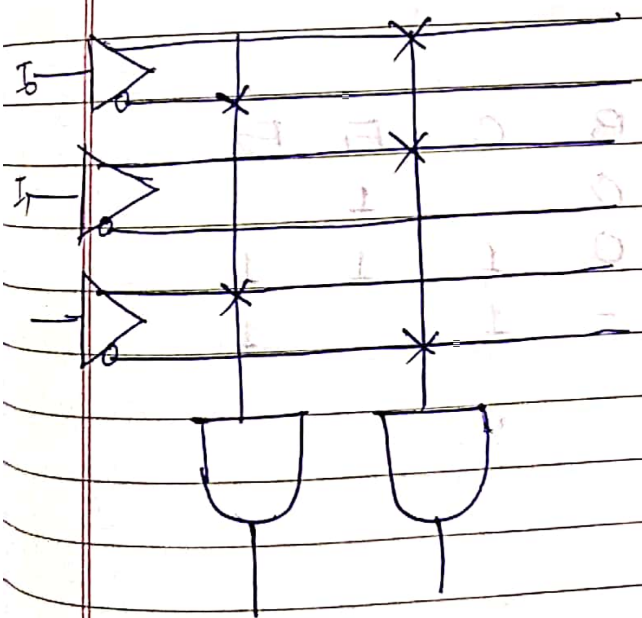
It has a fixed AND array that is constructed as a decoder and a programmable OR array.



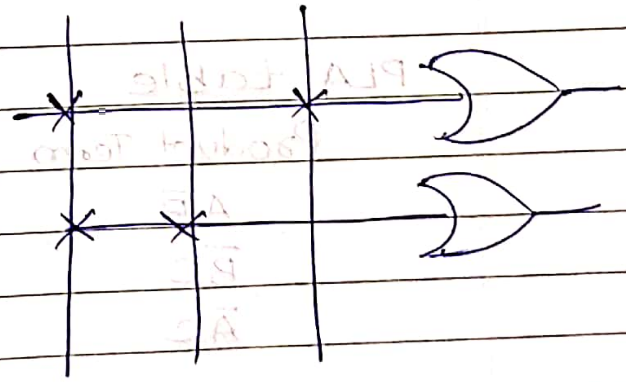
Input buffer



AND matrix



OR matrix



Programmable Logic Array (PLA)

A combinational circuit is defined by functio

$F_1(A,B,C) = \Sigma m(1,4,5)$

$F_2(A,B,C) = \Sigma m(1,3,5)$

Implement circuit with PLA.

$F_1(A,B,C) = \Sigma m(1,4,5)$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1 1	3	2
A	1 4	1 5	7	6

$F_1 = A\bar{B} + \bar{B}C$

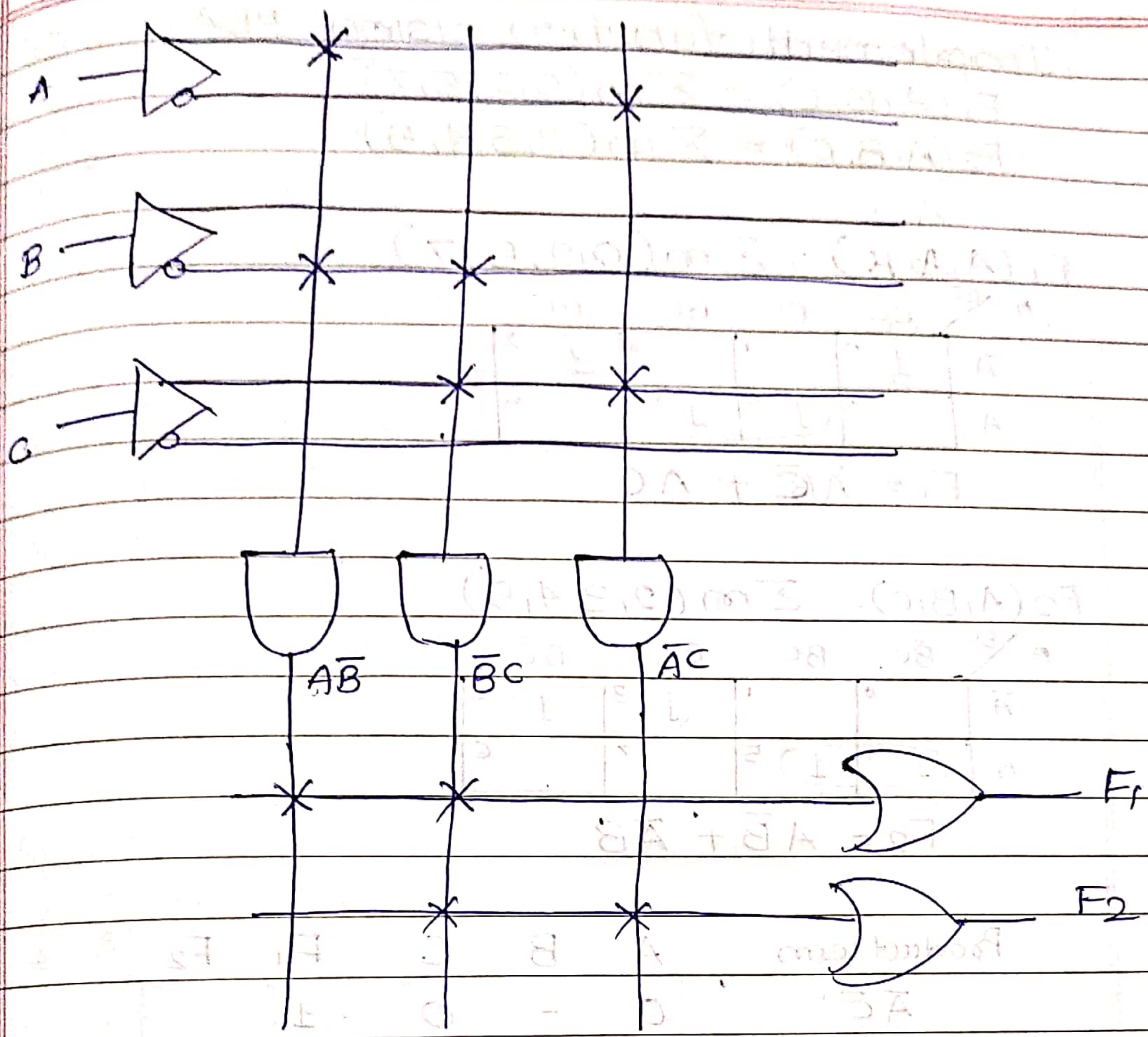
$F_2(A,B,C) = \Sigma m(1,3,5)$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1 1	1 3	2
A	4	1 5	7	6

$F_2 = \bar{B}C + \bar{A}C$

PLA table

Product Term	A	B	C	F_1	F_2
$A\bar{B}$	1	0		1	
$\bar{B}C$	-	0	1	1	1
$\bar{A}C$	0	-	1		1



A
B
C
$A\bar{B}$
$\bar{B}C$
$\bar{A}C$
F_1
F_2

Implement function using PLA

$F_1(A,B,C) = \Sigma m(0,2,5,7)$

$F_2(A,B,C) = \Sigma m(2,3,4,5)$

$F_1(A,B,C) = \Sigma m(0,2,5,7)$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1 (0)	(1)	(3)	1 (2)
A	(4)	1 (5)	1 (7)	(6)

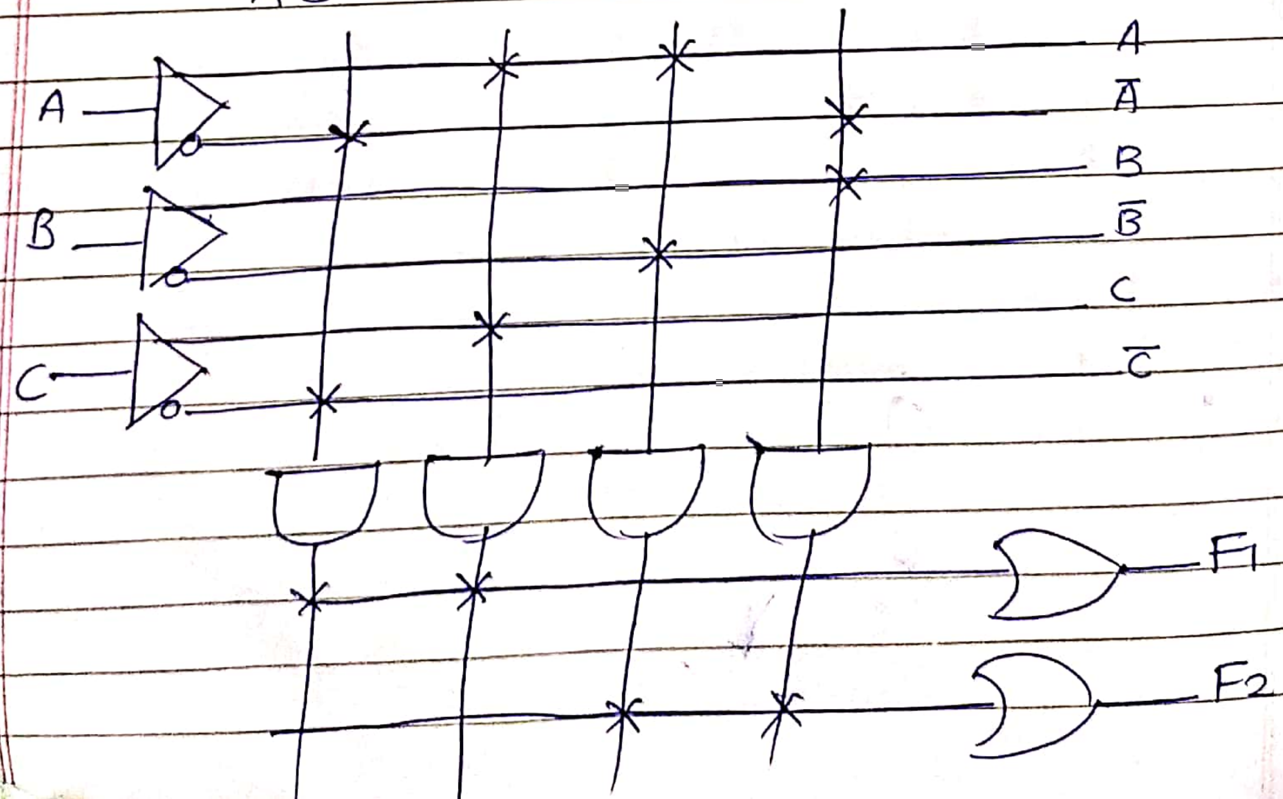
$F_1 = \bar{A}\bar{C} + AC$

$F_2(A,B,C) = \Sigma m(2,3,4,5)$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	(0)	(1)	1 (3)	1 (2)
A	1 (4)	1 (5)	(7)	(6)

$F_2 = A\bar{B} + \bar{A}B$

Product Term	A	B	C	F_1	F_2
$\bar{A}\bar{C}$	0	-	0	1	
AC	1	-	1	1	
$A\bar{B}$	1	0	-		1
$\bar{A}B$	0	1	-		1



Implement full Adder using PLA

Step 1

Truth Table

A	B	Cin	Sum	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

Step 2

K-Map for Sum

A \ B Cin	$\bar{B}\bar{C}in$	$\bar{B}Cin$	$BCin$	$B\bar{C}in$
$\bar{A}$	0	1 (1)	3	1 (2)
A	1 (4)	5	1 (7)	6

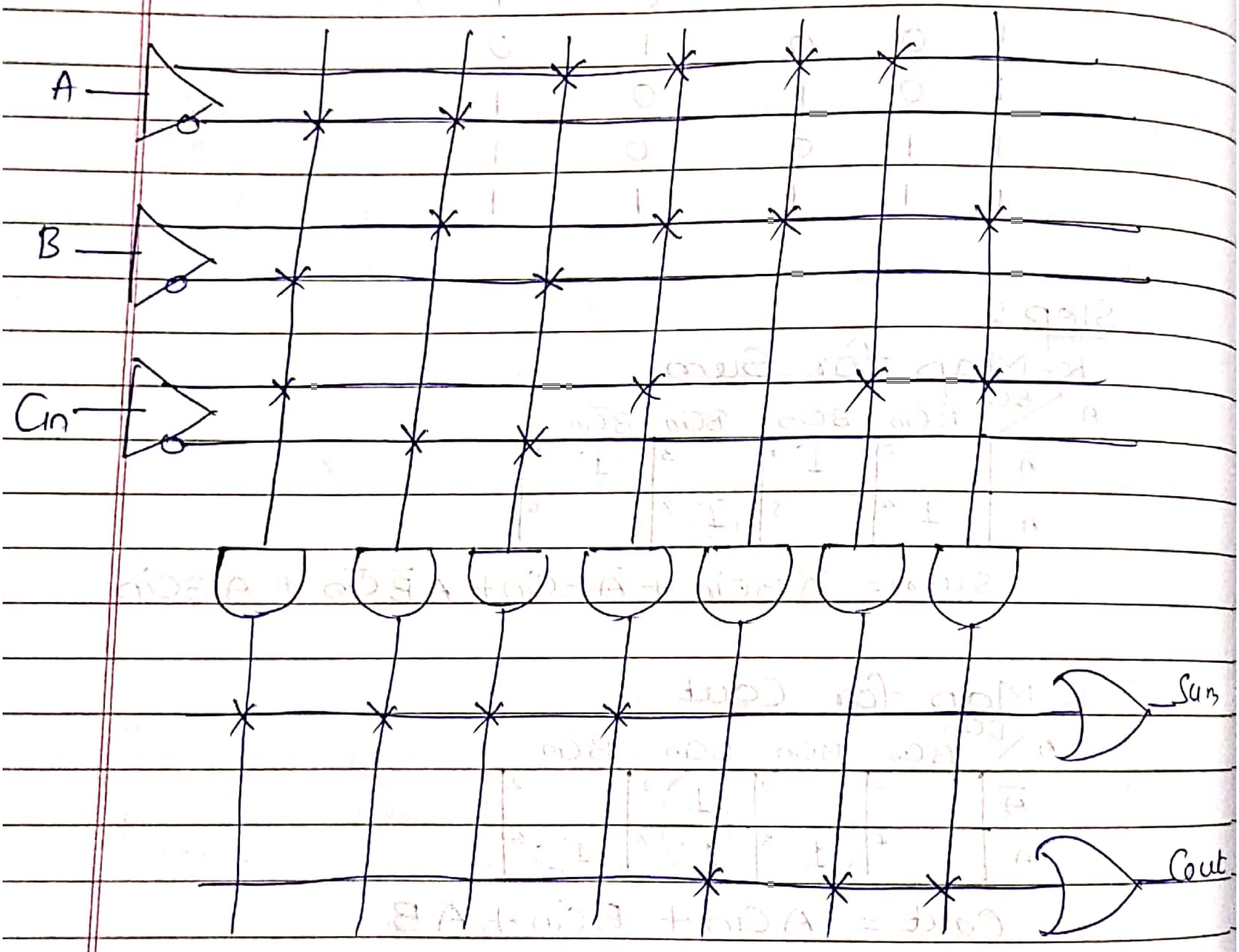
$$Sum = \bar{A}\bar{B}Cin + \bar{A}B\bar{C}in + A\bar{B}\bar{C}in + ABCin$$

K-Map for Cout

A \ B Cin	$\bar{B}\bar{C}in$	$\bar{B}Cin$	$BCin$	$B\bar{C}in$
$\bar{A}$	0	1	1 (3)	2
A	4	1 (5)	1 (7)	1 (6)

$$Cout = ACin + BCin + AB$$

Product Term	A	B	Cin	Sum	Cout
$\bar{A}\,\bar{B}\,Cin$	0	0	1	1	-
$\bar{A}\,B\,\overline{Cin}$	0	1	0	1	-
$A\,\bar{B}\,\overline{Cin}$	1	0	0	1	-
$A\,B\,Cin$	1	1	1	1	-
AB	1	1	-	-	1
$A\,Cin$	1	-	1	-	1
$B\,Cin$	-	1	1	-	1



Implement following function using PLA

$F_1(A,B,C,D) = \Sigma m(3,7,8,9,11,15)$

$F_2(A,B,C,D) = \Sigma m(3,4,5,7,10,14,15)$

Step 1

K-Map for F_1

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1 (3)	2
$\bar{A}B$	4	5	1 (7)	6
AB	12	13	1 (15)	14
$A\bar{B}$	1 (8)	1 (9)	1 (11)	10

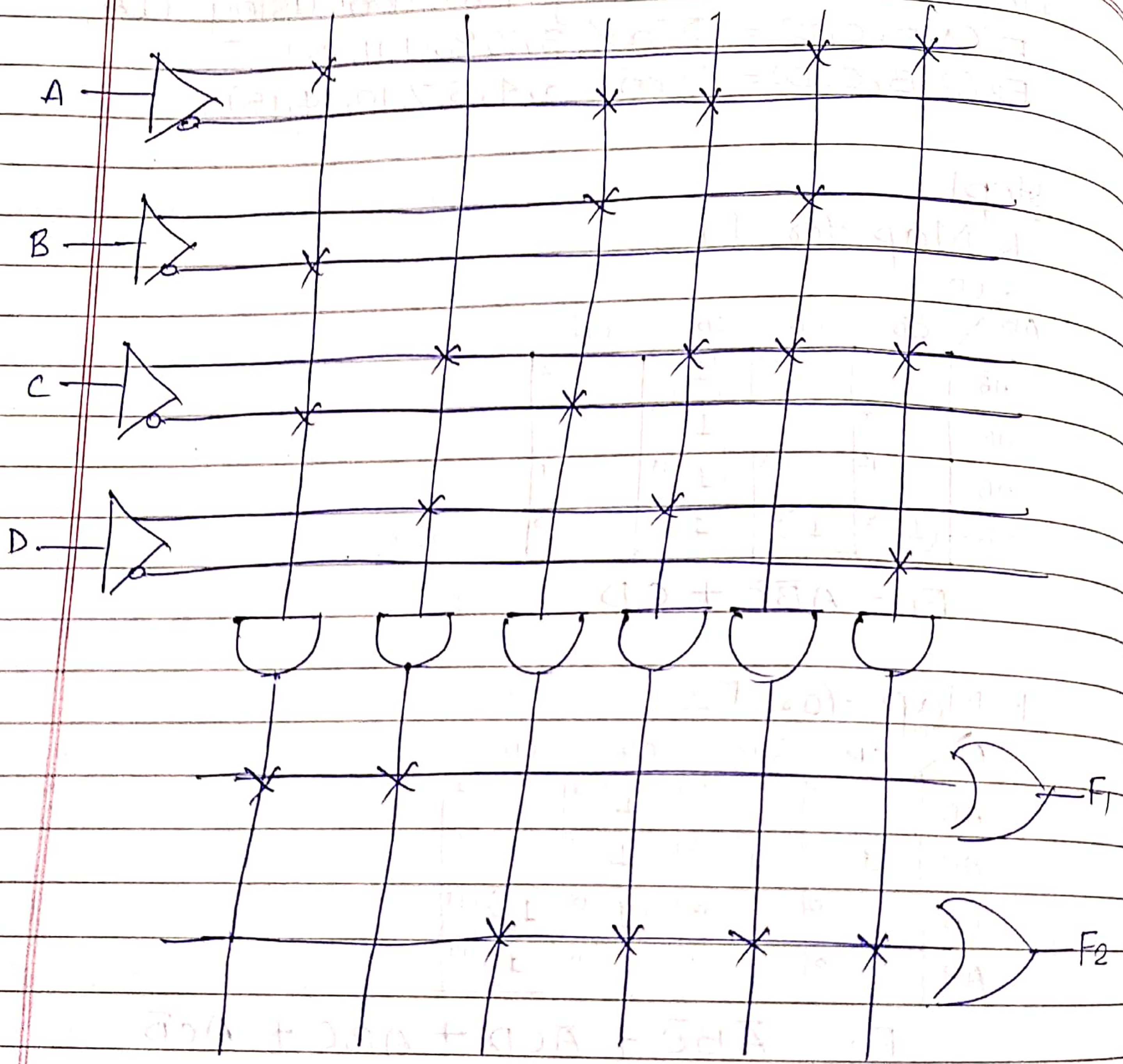
$F_1 = A\bar{B}\bar{C} + CD$

K-Map for F_2

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1 (3)	2
$\bar{A}B$	1 (4)	1 (5)	1 (7)	6
AB	12	13	1 (15)	1 (14)
$A\bar{B}$	8	9	11	1 (10)

$F_2 = \bar{A}B\bar{C} + \bar{A}CD + ABC + AC\bar{D}$

Product Term	A	B	C	D	F_1	F_2
$A\bar{B}\bar{C}$	1	0	0	-	1	
CD	-	-	1	1	1	
$\bar{A}B\bar{C}$	0	1	0	-	-	1
$\bar{A}CD$	0	-	1	1	-	1
ABC	1	1	1	-	-	1
ACD	1	-	1	1	-	1


A
B
C
D
F1
F2

Programmable Array Logic (PAL)

It is a programmable logic device that has programmable AND array & fixed OR Array

i/p → | Programmable AND array | → | Fixed OR array | → o/p

Implement the following function using PAL

$F_1(A,B,C) = \Sigma m(0,1,3,4)$

$F_2(A,B,C) = \Sigma m(3,4,5,\cancel{7}6)$

Step 1

K-Map for F_1

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1 (0)	1 (1)	1 (3)	(2)
A	1 (4)	(5)	(7)	(6)

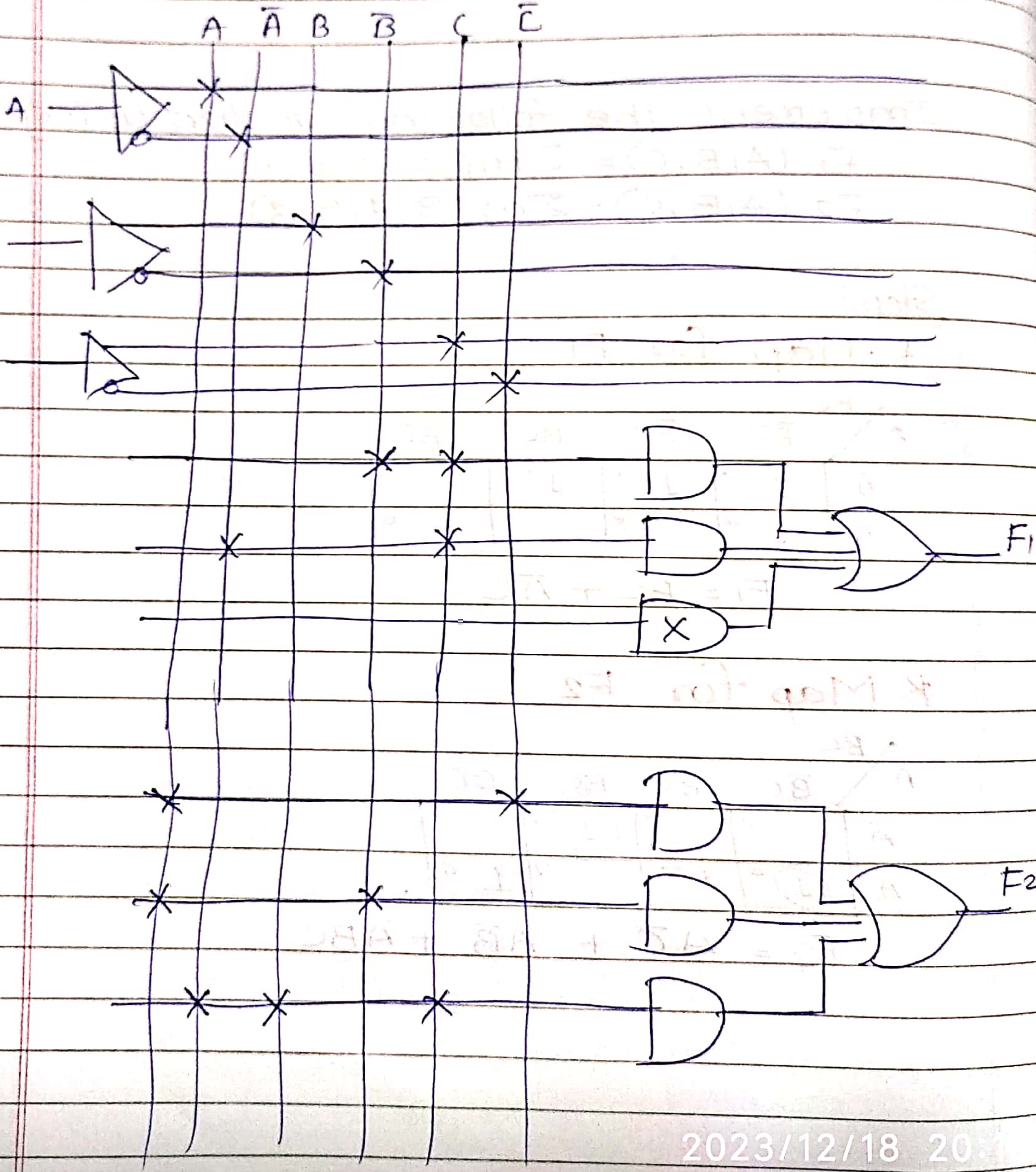
$F_1 = \bar{B}\bar{C} + \bar{A}C$

K-Map for F_2

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	(0)	(1)	1 (3)	(2)
A	1 (4)	1 (5)	(7)	1 (6)

$F_2 = A\bar{C} + A\bar{B} + \bar{A}BC$

Product Term	A	B	C	O/P
$\bar{B}C$	-	0	1	$F_1 = \bar{B}C + \bar{A}C$
$\bar{A}C$	0	-	1	
$A\bar{C}$	1	-	0	
$A\bar{B}$	1	0		$F_2 = A\bar{C} + A\bar{B} + \bar{A}BC$
$\bar{A}BC$	0	1	1	



Design BCD to Excess-3 code converter using PAL

Step 1 Truth Table

Decimal	BCD B_3	B_2	B_1	B_0	Excess-3 E_3	E_2	E_1	E_0
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	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

~~10~~

Step 2

K-Map for E_3

$B_3B_2 \backslash B_1B_0$	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$	(0)	(1)	(3)	(2)
$\bar{B_3}B_2$	(4)	(5) 1	(7) 1	(6) 1
B_3B_2	(12) X	(13) X	(15) X	(14) X
$B_3\bar{B_2}$	(8) 1	(9) 1	(11) X	(10) X

$$E_3 = B_3 + B_2B_0 + B_2B_1$$

K-Map for E_2

$B_3B_2 \backslash B_1B_0$	$\bar{B_1}\bar{B_0}$	$\bar{B_1}B_0$	B_1B_0	$B_1\bar{B_0}$
$\bar{B_3}\bar{B_2}$	(0)	(1) 1	(3) 1	(2) 1
$\bar{B_3}B_2$	(4) 1	(5)	(7)	(6)
B_3B_2	(12) X	(13) X	(15) X	(14) X
$B_3\bar{B_2}$	(8) ~~X~~	(9) 1	(11) X	(10) X

$$E_2 = B_2\bar{B_1}\bar{B_0} + \bar{B_2}B_0 + \bar{B_2}B_1$$

K-Map for E_1

$B_3B_2 \backslash B_1B_0$	$\overline{B_1}\,\overline{B_0}$	$\overline{B_1}B_0$	B_1B_0	$B_1\overline{B_0}$
$\overline{B_3}\,\overline{B_2}$	1 (0)	(1)	1 (3)	(2)
$\overline{B_3}B_2$	1 (4)	(5)	1 (7)	(6)
B_3B_2	X (12)	X (13)	X (15)	X (14)
$B_3\overline{B_2}$	1 (8)	(9)	X (11)	X (10)

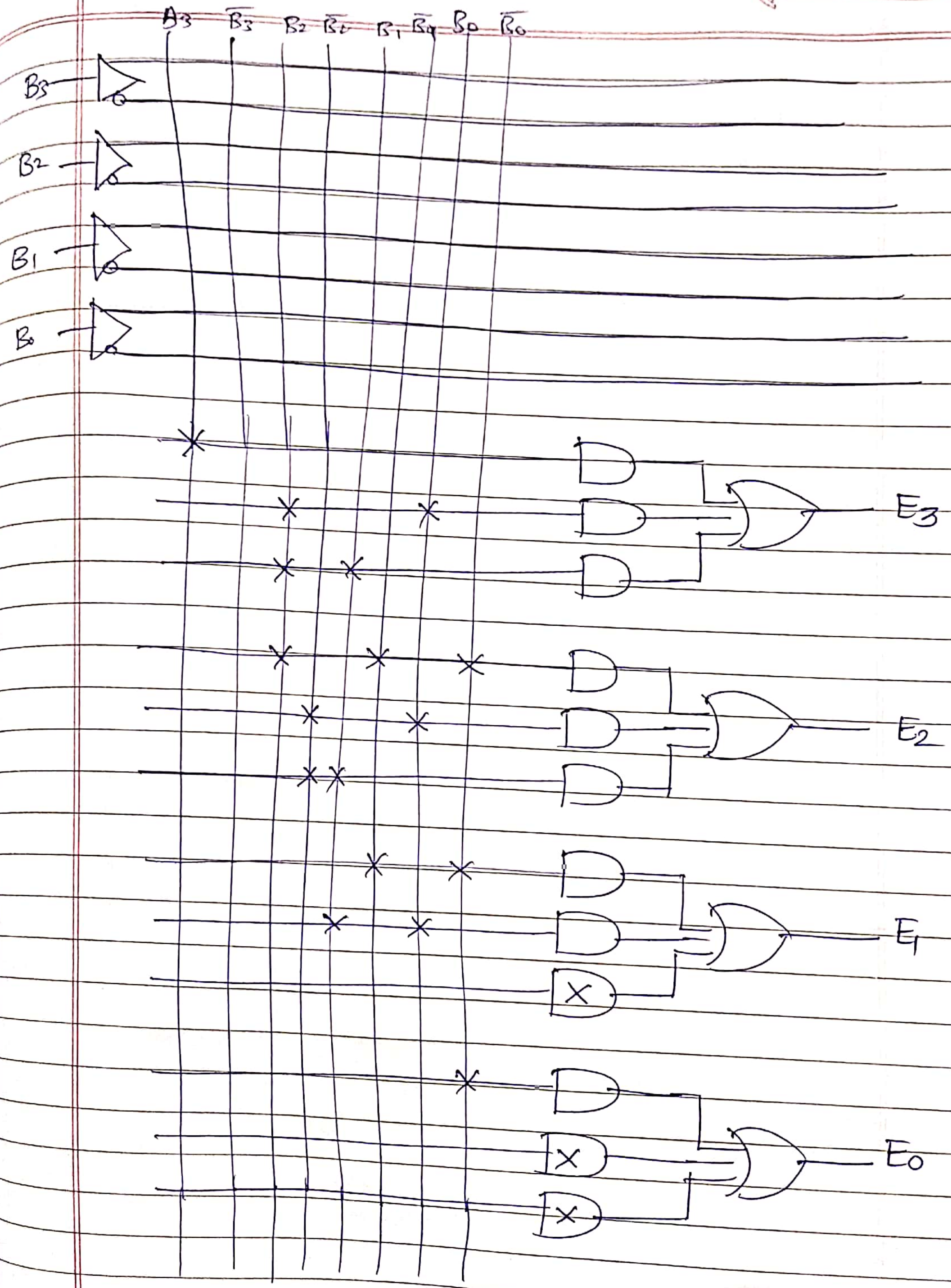
$$E_1 = \overline{B_1}\,\overline{B_0} + B_1B_0$$

K-Map for E_0

$B_3B_2 \backslash B_1B_0$	$\overline{B_1}\,\overline{B_0}$	$\overline{B_1}B_0$	B_1B_0	$B_1\overline{B_0}$
$\overline{B_3}\,\overline{B_2}$	1 (0)	(1)	(3)	1 (2)
$\overline{B_3}B_2$	1 (4)	(5)	(7)	1 (6)
B_3B_2	X (12)	X (13)	X (15)	X (14)
$B_3\overline{B_2}$	1 (8)	(9)	X (11)	X (10)

$$E_0 = \overline{B_0}$$

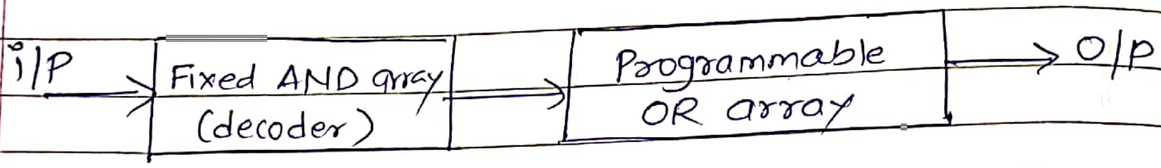
Product Term	B_3	B_2	B_1	B_0	O/P
B_3	1	-	-	-	$E_3 = B_3 + B_2B_0 + B_2B_1$
B_2B_0	-	1	-	1	
B_2B_1	-	1	1	-	
$B_2\overline{B_1}\,\overline{B_0}$	-	1	0	0	
$\overline{B_2}B_0$	-	0	-	1	$E_2 = B_2\overline{B_1}\,\overline{B_0} + \overline{B_2}B_0 + \overline{B_2}B_1$
$\overline{B_2}B_1$	-	0	1	-	
$\overline{B_1}\,\overline{B_0}$	-	-	0	0	$E_1 = \overline{B_1}\,\overline{B_0} + B_1B_0$
B_1B_0	-	-	1	1	
$\overline{B_0}$	-	-	-	0	$E_0 = \overline{B_0}$



A3
B̄3
B2
B̄2
B1
B̄1
B0
B̄0
B3
B2
B1
B0
E3
E2
E1
E0

Programmable ROM

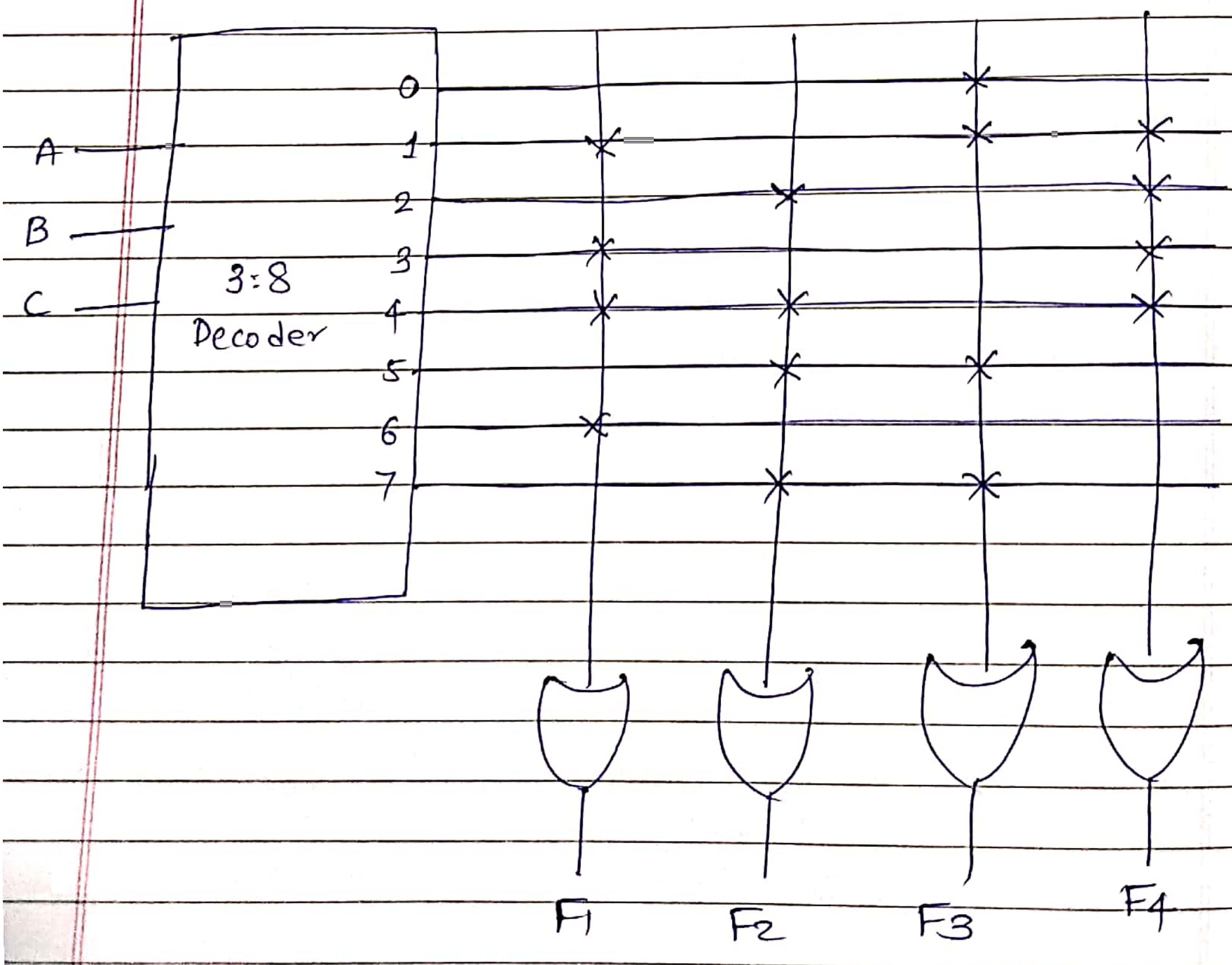
It has fixed AND array that is constructed as a decoder and programmable OR array.



Implement following function using PROM.

$F_1 = \Sigma m(1,3,4,6)$ $F_2 = \Sigma m(2,4,5,7)$

$F_3 = \Sigma m(0,1,5,7)$ $F_4 = \Sigma m(1,2,3,4)$



Design 3 bit Binary to Gray code converter using PROM.

Step 1 Truth Table

	Binary Code			Gray Code		
	B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

$G_2 = \Sigma m(4,5,6,7)$ $G_1 = \Sigma m(2,3,4,5)$
$G_0 = \Sigma m(1,2,5,6)$