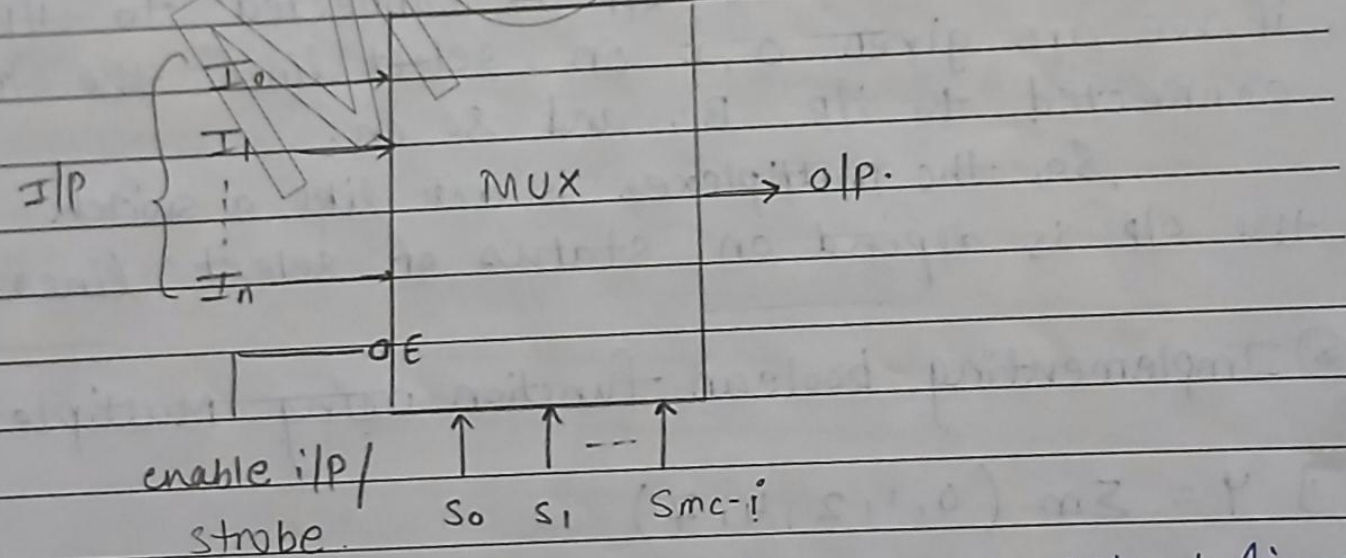
Name :- Akshata Pritish Zarjary
Class :- SY-IT-3 (UDICT)
Roll no :- 2305164

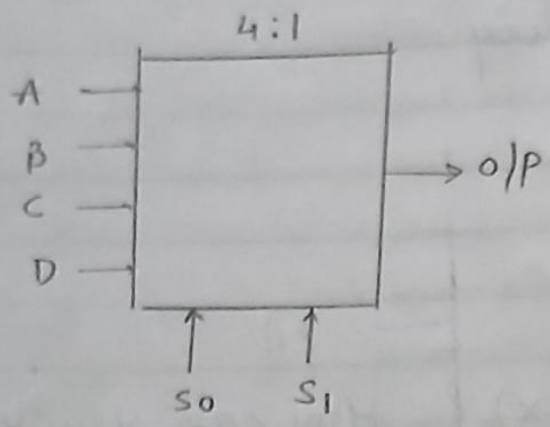
CA-III

Q.1. What is multiplexer (MUX)? How can you use a multiplexer in combinational logic design?

Ans: In multiplexer we have multiple i/p and single o/p. where the o/p is depend on select lines which decides the o/p.
It is represented as,



The relationship b/w i/p lines and select lines is given by $n = 2^m$ where, m are select lines and n are i/p lines.



S_0	S_1	o/p
0	0	A
0	1	B
1	0	C
1	1	D

If we consider 4:1 multiplexer we have 4 i/p lines and 2 select lines. the o/p is depend on select lines as we vary the i/p on select lines the o/p is also change.

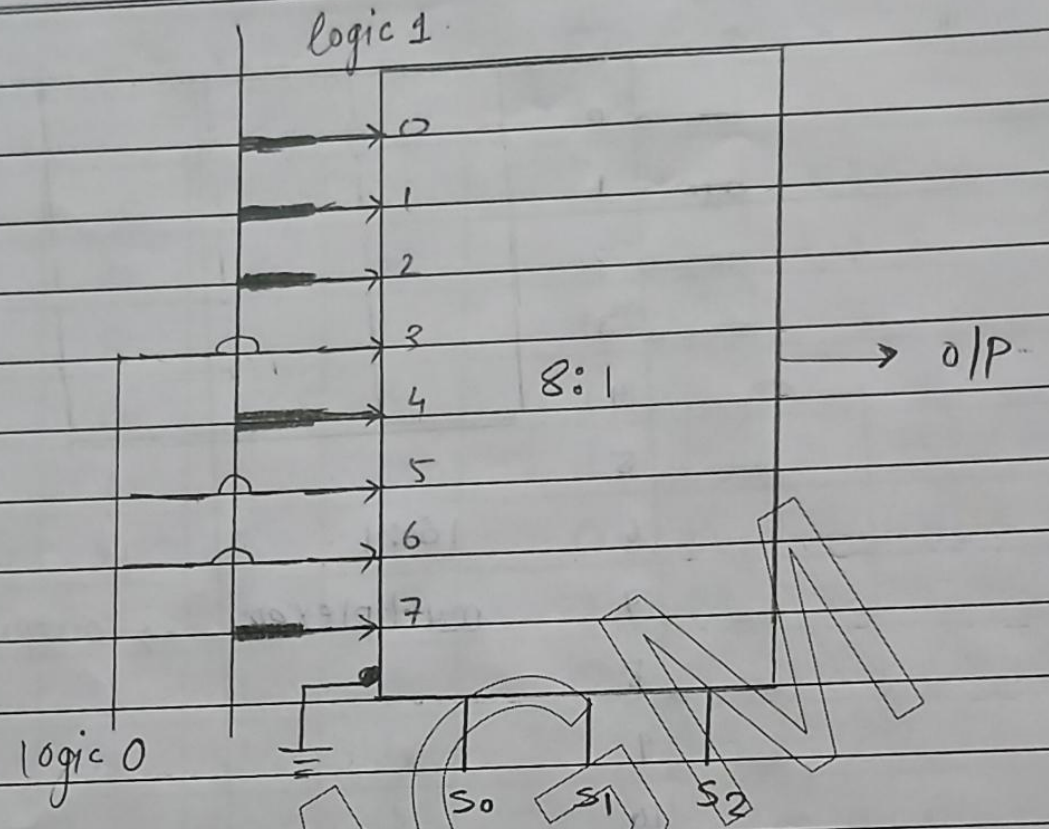
In 4:1 multiplexer if we are given 0,0 on select lines the o/p is connected to i/p A, if we are given 0,1 on select lines the o/p is connected to i/p B, and so on.

So, the multiplexer is act like a switch where the o/p is depend on status of select lines.

⊛ Implementing boolean function using Multiplexer.

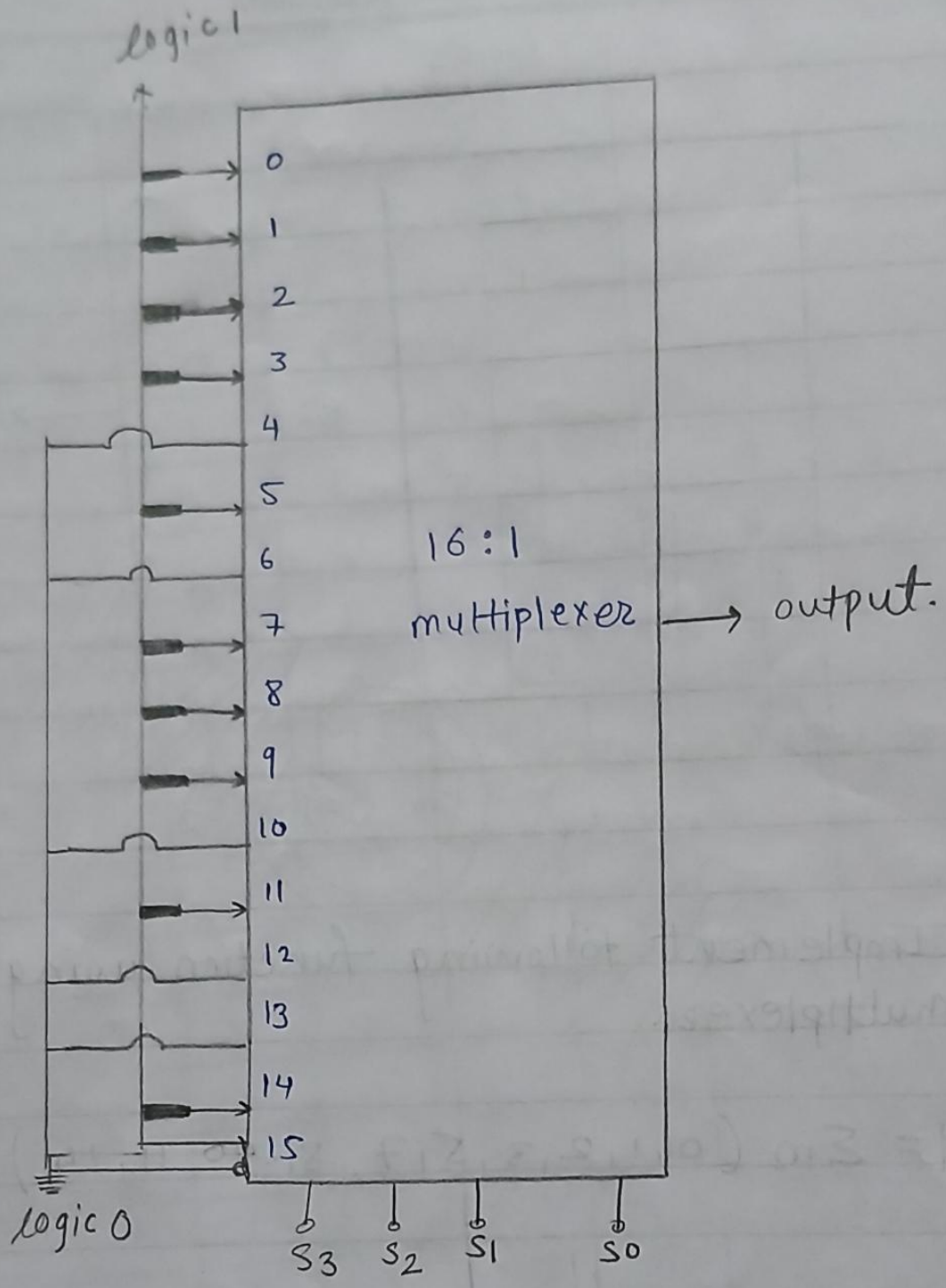
1] $Y = \Sigma m\ (0,1,2,4,7)$

Now, here maximum no. is 7. therefore we are going to use 8:1 multiplexer.



8.2 Implement following function using suitable multiplexer.

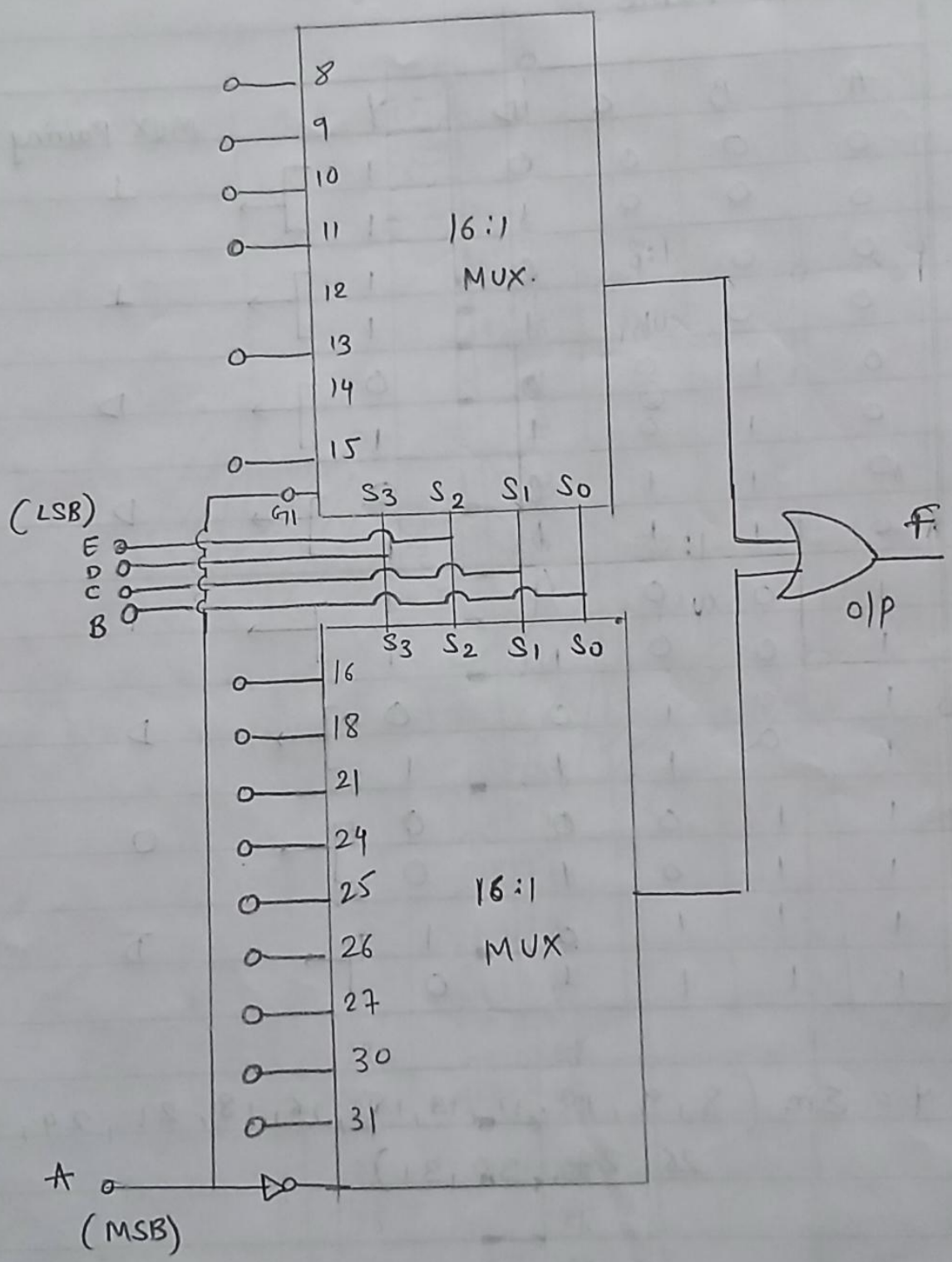
1] $Y = \Sigma m (0,1,2,3,5,7,8,9,11,14)$


logic 1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 : 1
multiplexer
output.
logic 0
S3
S2
S1
S0

Truth Table :-

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

b] $Y = \Sigma m (8, 9, 10, 11, 13, 15, 16, 18, 21, 24, 25, 26, 27, 30, 31)$.


8
9
10
11
16:1
MUX
12
13
14
15
S3 S2 S1 S0
G1
(LSB)
E
D
C
B
F
o/p
S3 S2 S1 S0
16
18
21
24
25
26
27
30
31
16:1
MUX
A
(MSB)

Q.3 Implement following function using suitable multiplexer

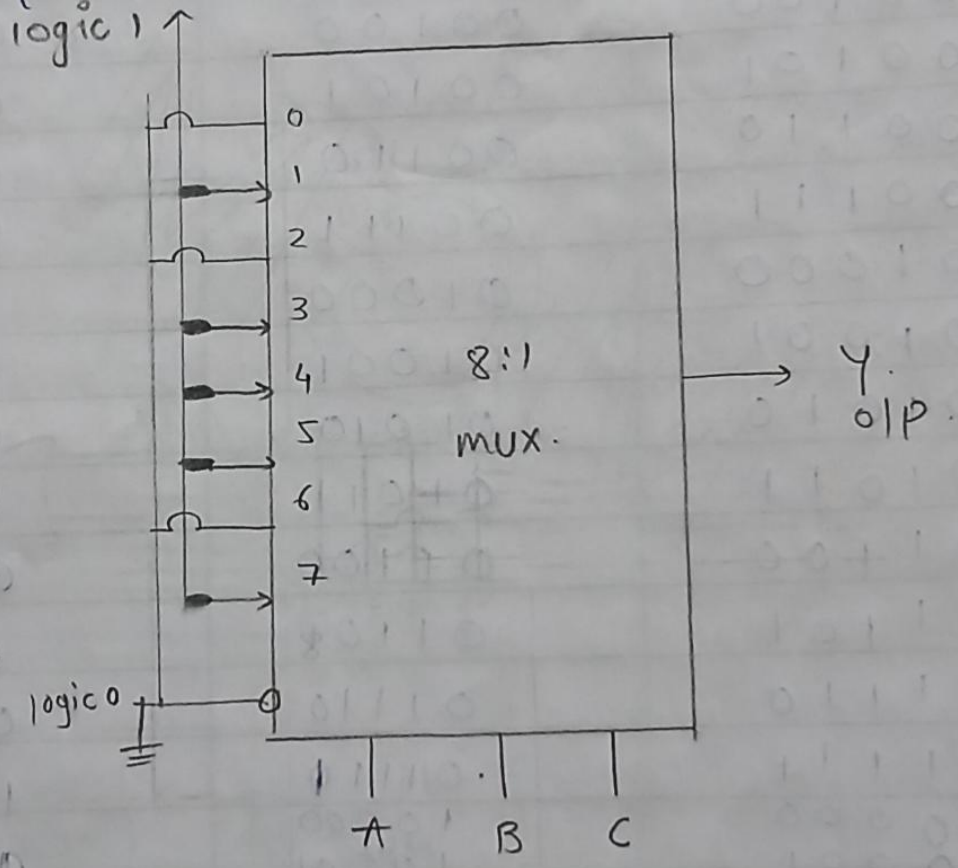
$Y = A'B'C + ABC + A'BC + AB'C + AB'C'$

$001 + 111 + 011 + 101 + 100$

Ans:

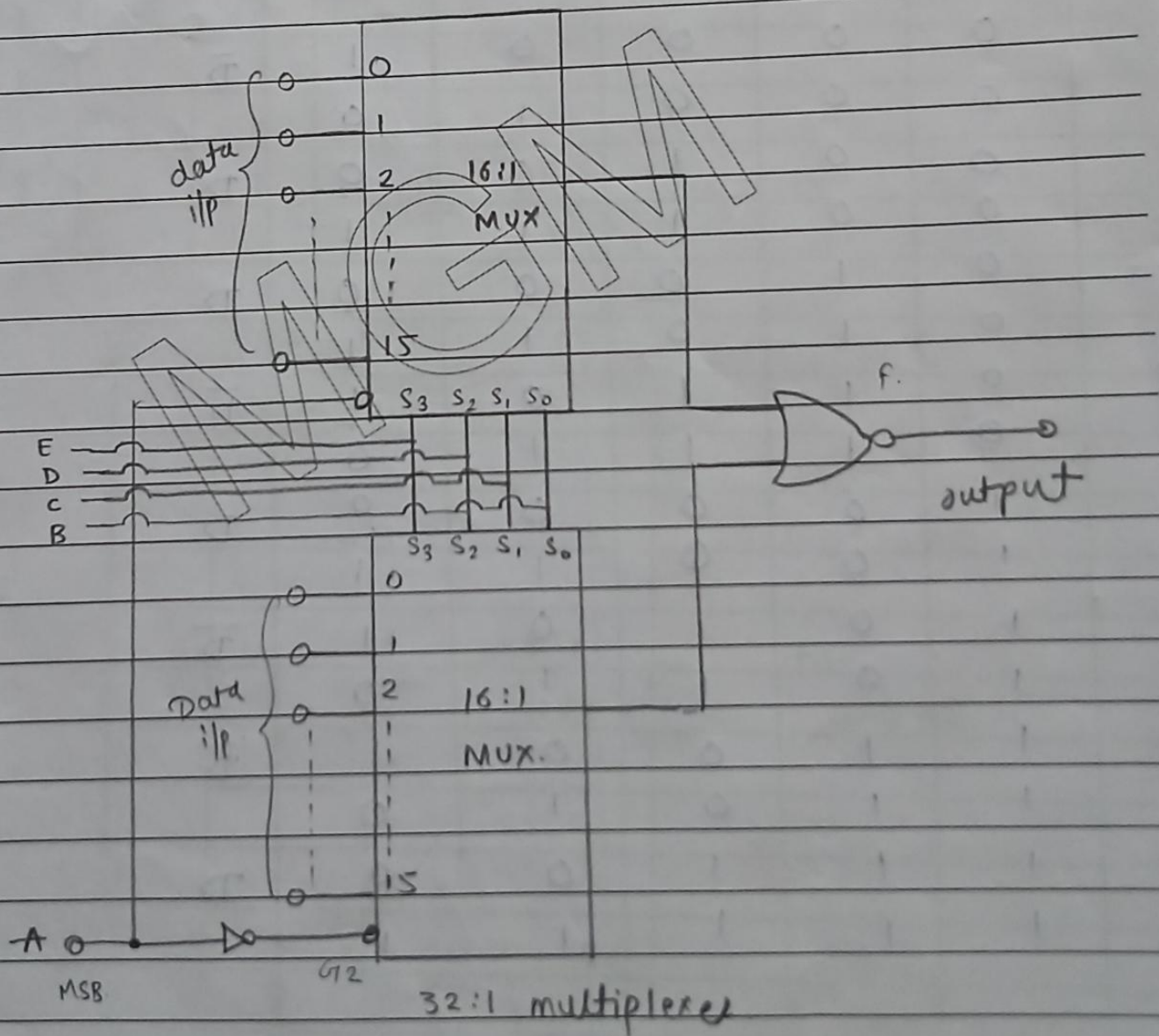
$\therefore Y = (1, 7, 3, 5, 4)$

$\therefore Y = \Sigma m\,(1, 7, 3, 5, 4).$



Q.4 Design a 32:1 multiplexer using two 16:1 multiplexer.

To design 32:1 multiplexer using two 16:1 multiplexers, we will use a hierarchical approach. Essentially we can split the 32 i/ps into ~~two~~ 2 groups of 16 i/ps and then use the 16:1 MUX to select from each grp. A third MUX will then choose b/w the o/ps of the 2 16:1 MUX.



32:1 multiplexer

Q.5. Realize the following functions of 4 variable using 8:1 multiplexer.

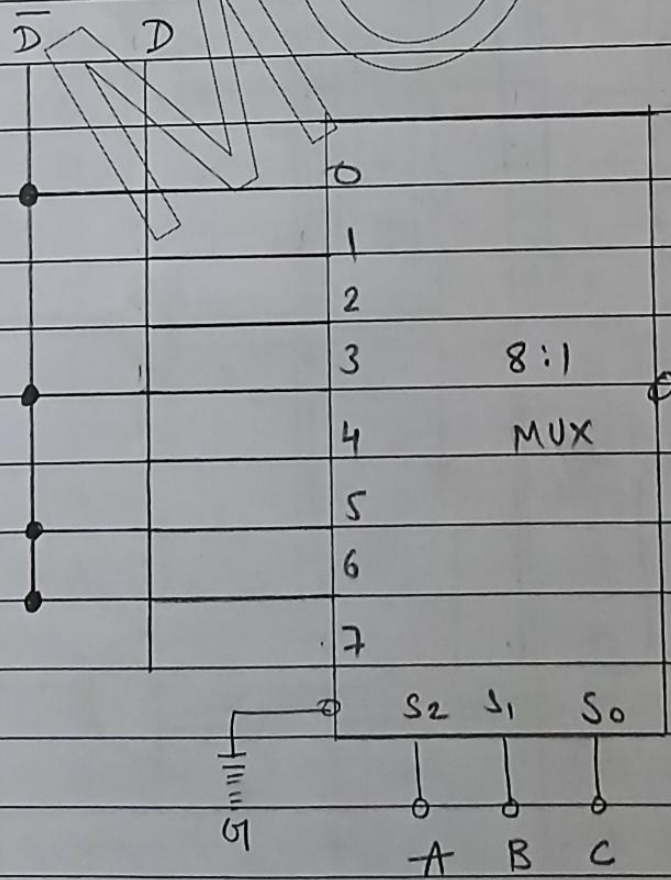
$F1 = \Sigma m (0, 3, 5, 6, 9, 10, 12, 15)$.

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

Truth table for 16:1 Multiplexer.

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

Truth table for 8:1 Multiplexer.



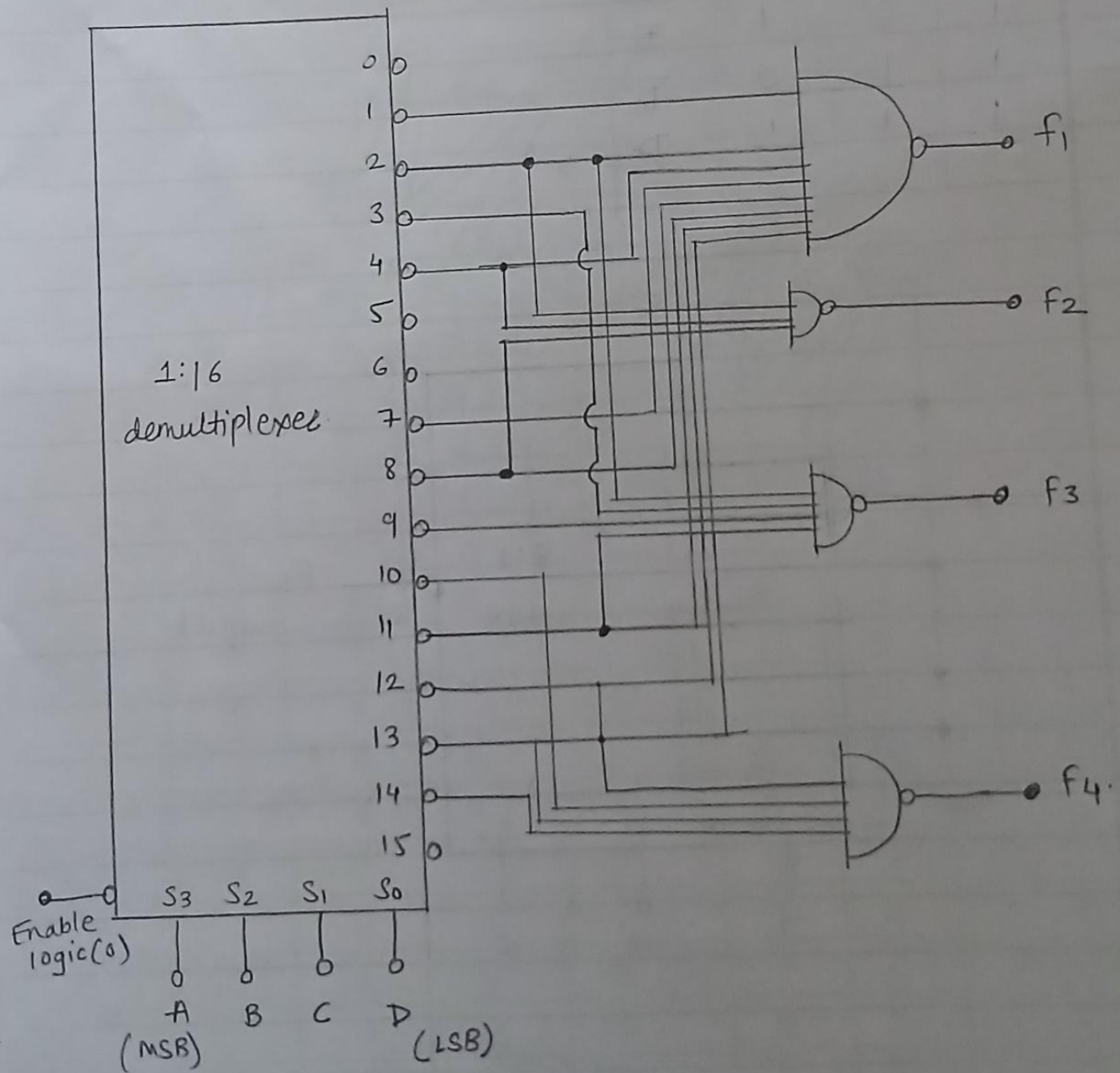
Q.6. Implement the following combinational using demultiplexer

$F_1 = \Sigma m\ (1, 2, 4, 7, 8, 11, 12, 13)$

$F_2 = \Sigma m\ (2, 3, 9, 11)$

$f_3 = \Sigma m\ (10, 12, 13, 14)$

$F_4 = \Sigma m\ (2, 4, 8)$



Q.7. explain 2 bit Comparator along with truth-table logical expression and circuit diagram.

→ A Comparator used to compare two binary numbers each of 2 bits is c/d a 2-bit Comparator it consists of four i/p and three o/p to generate less than, equal to and greater than b/w 2 binary numbers.

— Truth-Table for 2-bit Comparator.

Inputs				Outputs.		
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	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

logical expression for the above truth table.

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

$$A=B = \bar{A_1}\bar{A_0}\bar{B_1}\bar{B_0} + \bar{A_1}A_0\bar{B_1}B_0 + A_1A_0B_1B_0 + A_1\bar{A_0}B_1\bar{B_0}$$

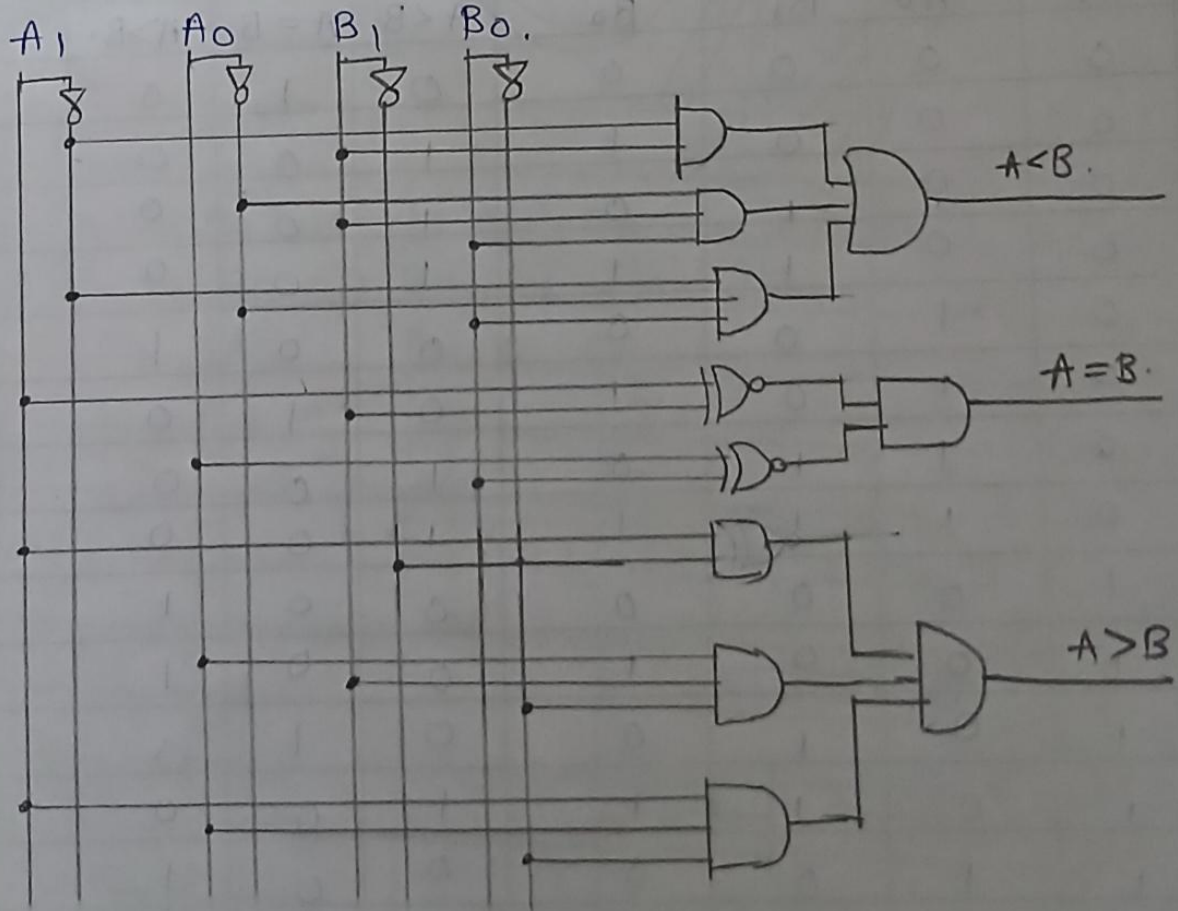
$$= \bar{A_1}\bar{B_1}(\bar{A_0}\bar{B_0} + A_0B_0) + A_1B_1(A_0B_0 + \bar{A_0}\bar{B_0})$$

$$= (A_0B_0 + \bar{A_0}\bar{B_0})(A_1B_1 + \bar{A_1}\bar{B_1})$$

$$= (A_0 \text{ Ex-Nor } B_0)(A_1 \text{ Ex-Nor } B_1)$$

$$A<B = \bar{A_1}B_1 + A_0\bar{B_1}B_0 + \bar{A_1}\bar{A_0}B_0.$$

Logic Circuit :-



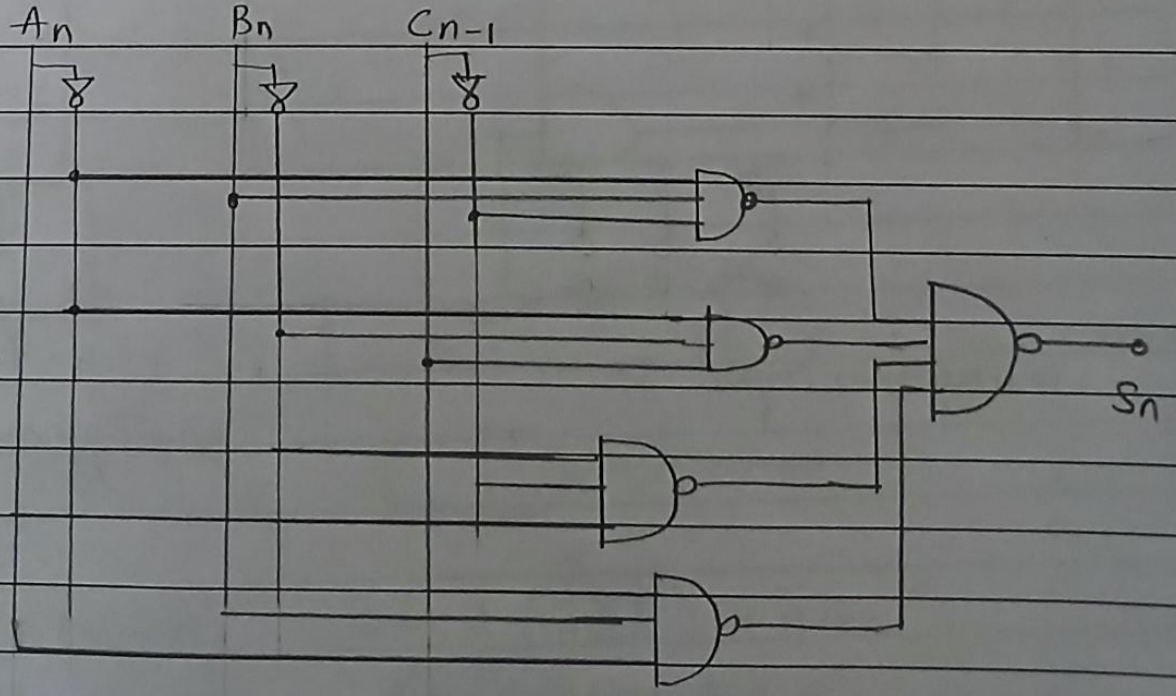
Q.8. Design full adder & full substractor circuit using suitable multiplexer.

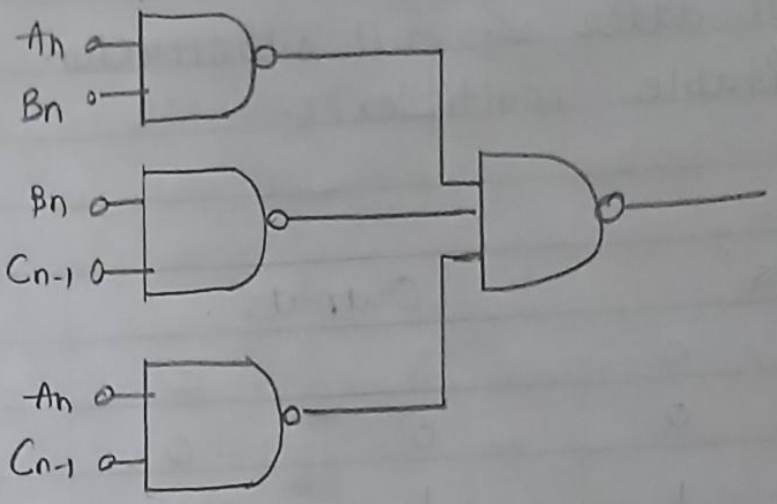
→ Full adder.

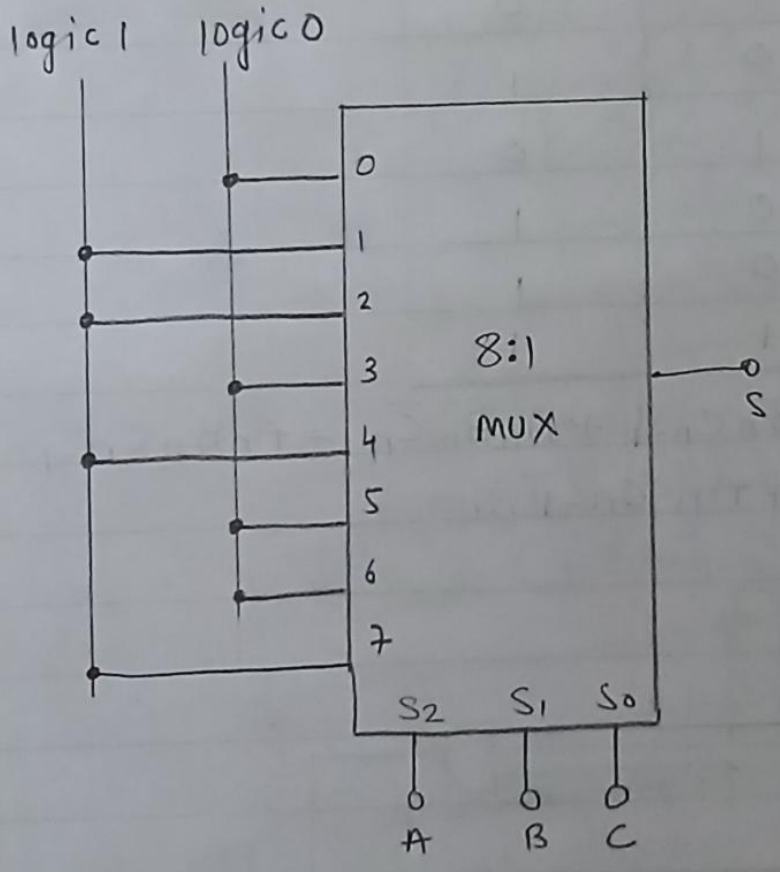
Inputs			Outputs.	
A	B	C	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S_n = \bar{A}_n B_n \bar{C}_{n-1} + \bar{A}_n \bar{B}_n C_{n-1} + A_n \bar{B}_n \bar{C}_n + A_n B_n C_{n-1}$

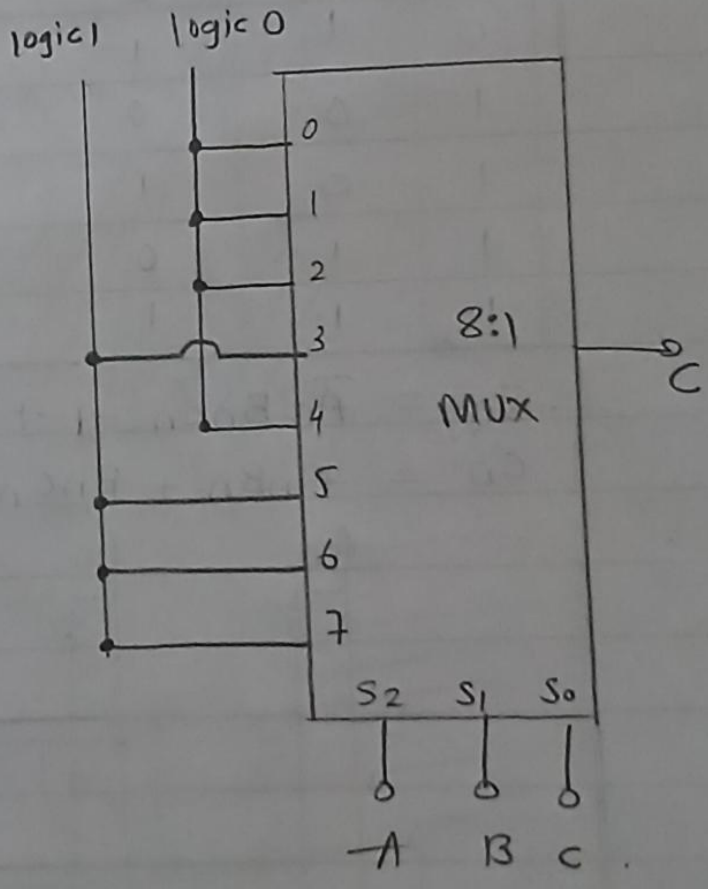
$C_n = A_n B_n + B_n C_{n-1} + A_n C_{n-1}$.







Multiplexer for S.



Multiplexer for C.

b) Full Substractor

Inputs			Outputs	
A	B	C	S	C
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

$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$C = \bar{A}\bar{B}C + \bar{A}BC + \bar{A}BC + ABC.$

$S = C(\bar{A}\bar{B} + AB) + \bar{C}(\bar{A}B + A\bar{B})$

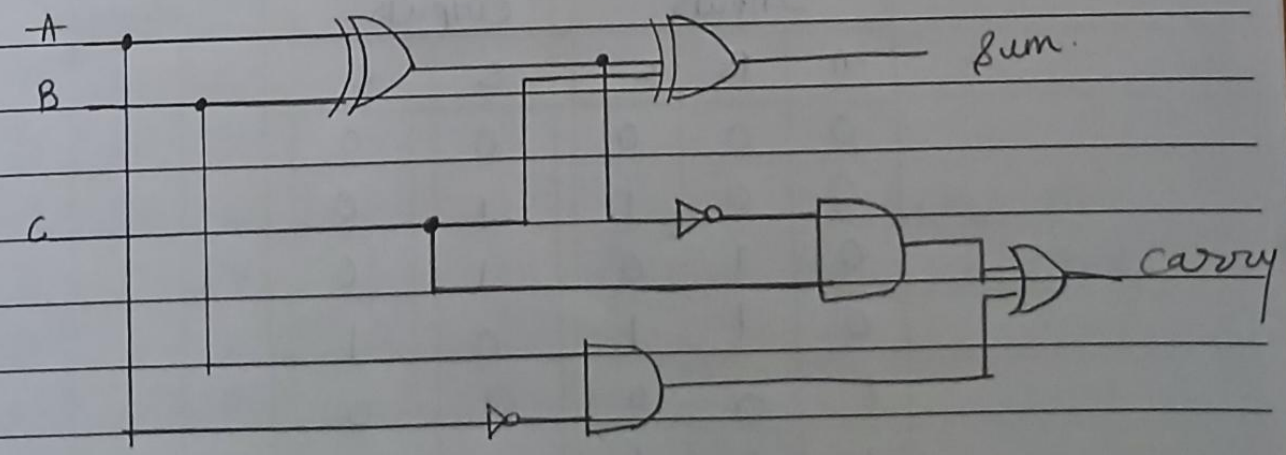
$= C(\overline{A \oplus B}) + \bar{C}(A \oplus B)$

$S = C \oplus (A \oplus B).$

$C = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC.$

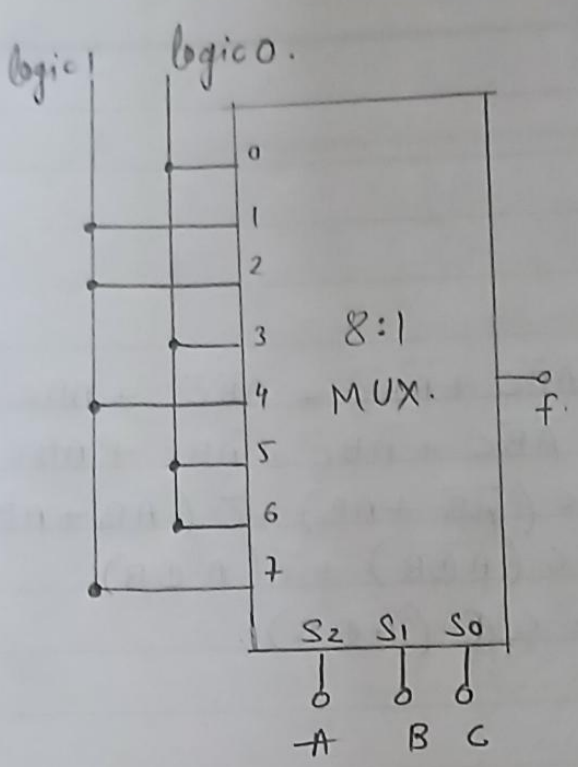
$= C(\bar{A}\bar{B} + AB) + (\bar{A}B(\bar{C} + C))$

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

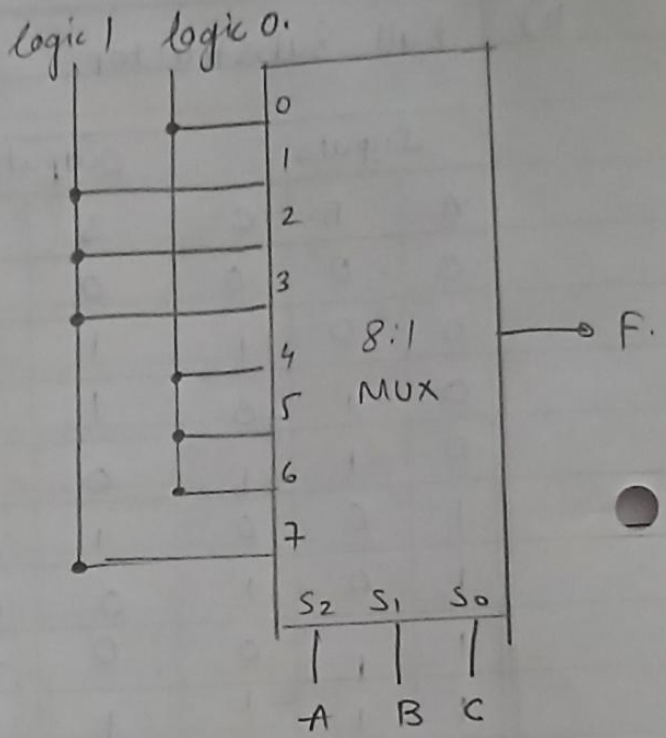


$S = C \oplus (A \oplus B)$ $C = C(\overline{A \oplus B}) + \bar{A}B.$

Circuit diagram for full substractor.



① MUX for sum.



② MUX for carry.

Q.9. design full adder & full substractor circuit using suitable demultiplexer.

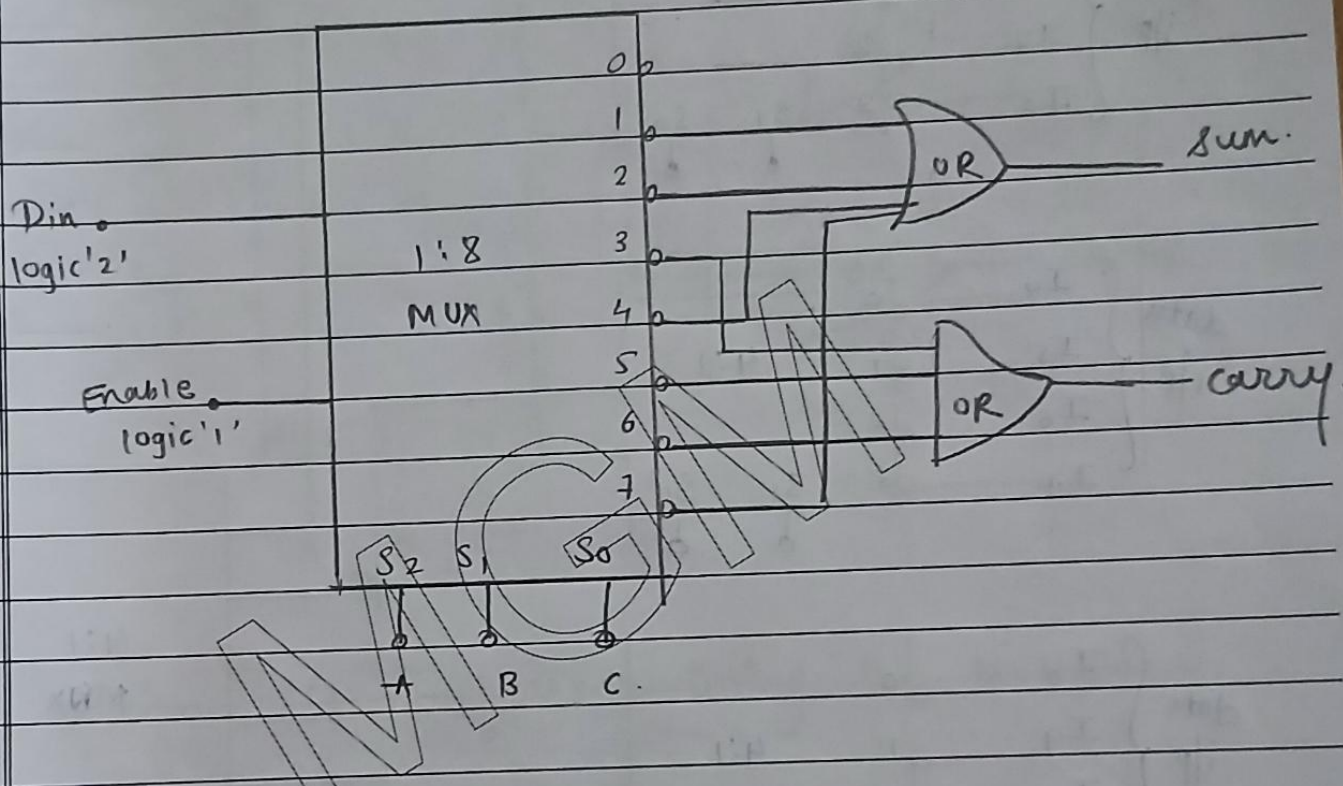
→ full adder.

Inputs			Outputs	
A	B	C	S	C
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

MGMUNIVERSITY

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

$C = \Sigma m\ (4,3,5,6,7)$.

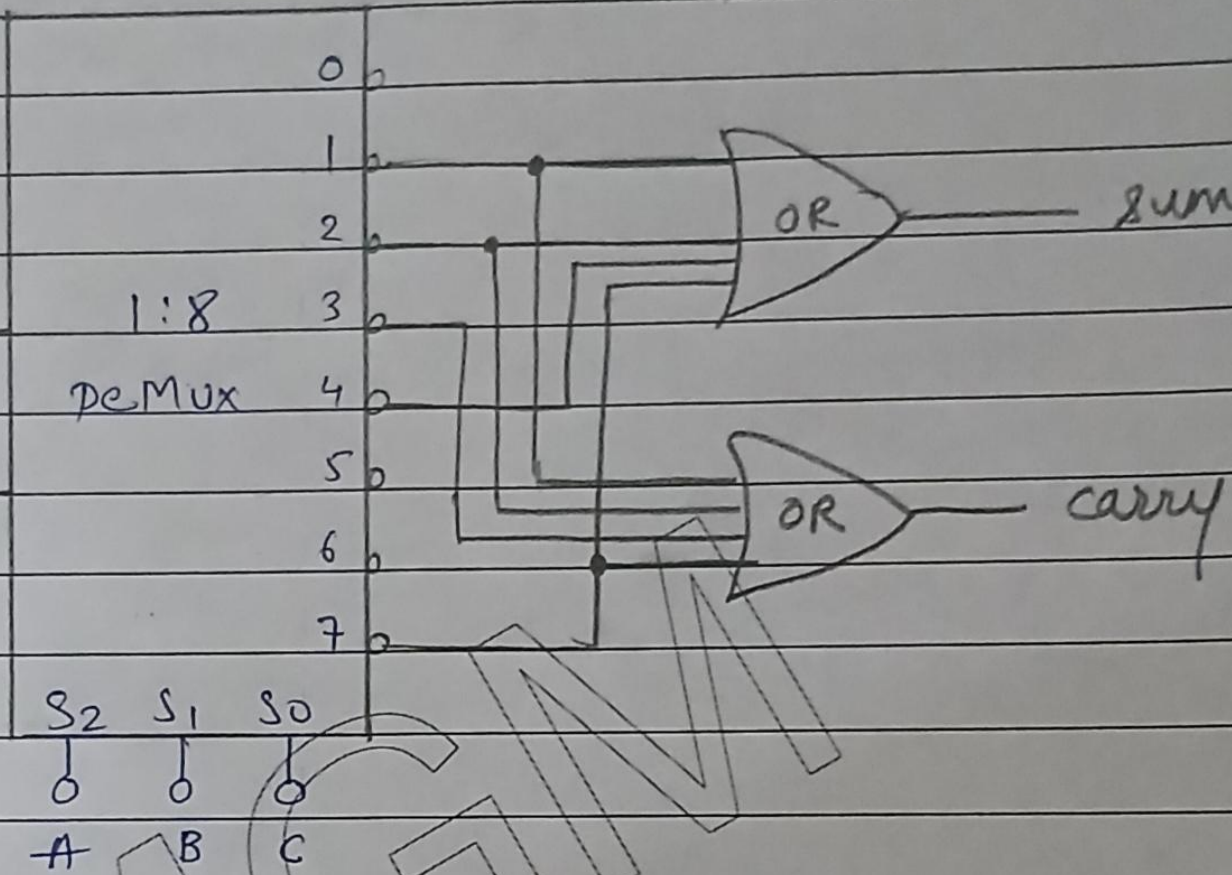


full Substractor.

Inputs			Outputs	
A	B	C	S	C
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

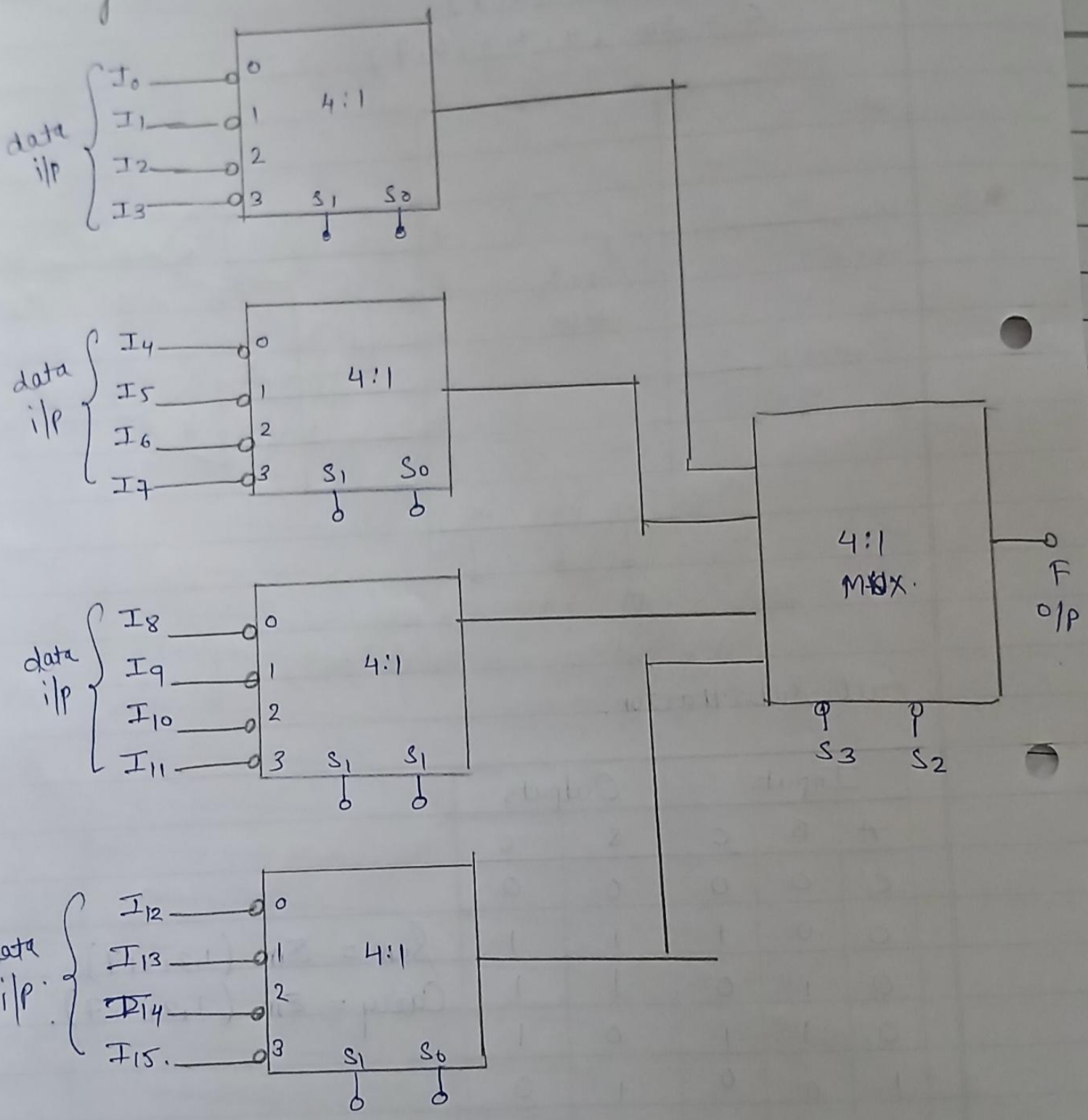
$Sum = \Sigma m\ (1,2,4,7)$

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



~~Q.10~~ ~~Design 16:1 multiplexer using 4:1 Multiplexer.~~

10. design 16:1 Multiplexer using 4:1 Multiplexer.



16:1 MUX using 4:1