* Boolean f^n Implementation of MUX.

* If we have Boolean f^n of $n+1$ variables, we take n of these variable as selection line.
* remaining variables of the f^n is used as i/p of MUX.

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

$n+1$ variable $= 2+1$ variable

so take, 2 selection line & 1 i/p

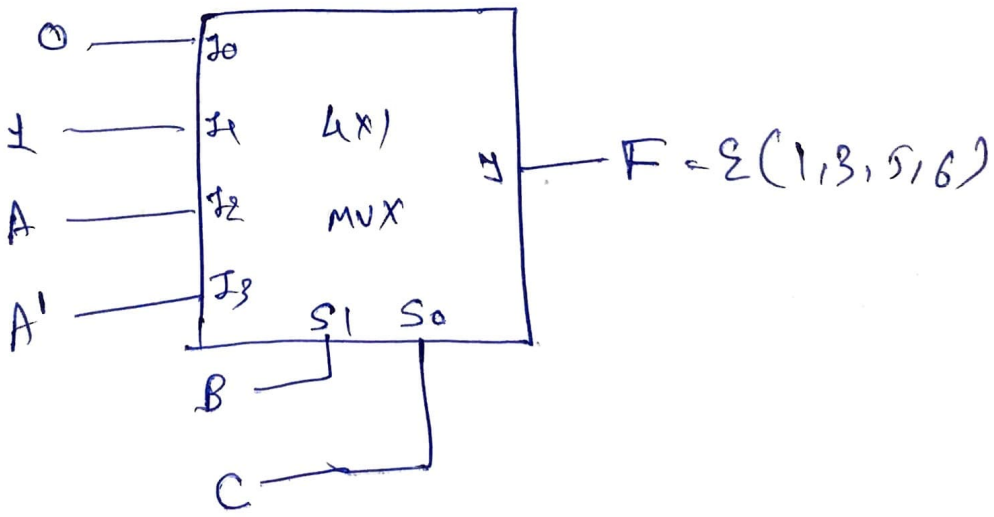
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

here take B & C as selection line & A i/p line

→ · Implementation Table

	I_0	I_1	I_2	I_3
A'	0	(1)	2	(3)
A	4	(5)	(6)	7
	0	1	A	A'

→ MUX Implementation

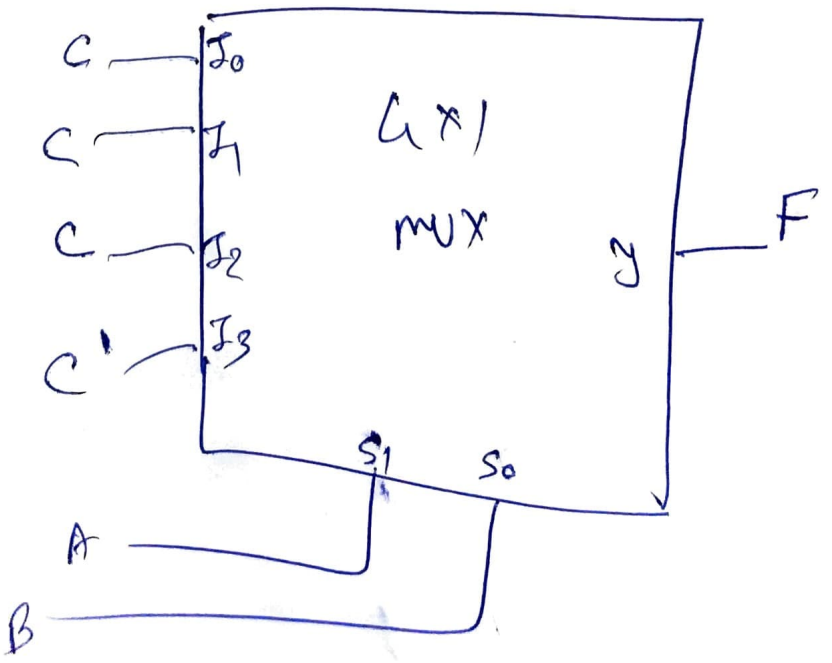


→ Same Example but now take A & B as selection line & C is I/P

→ Implementation table

	I_0	I_1	I_2	I_3
C'	0	2	4	(6)
C	(1)	(3)	(5)	7
	C	C	C	C'

→ MUX Implementation



Ex

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

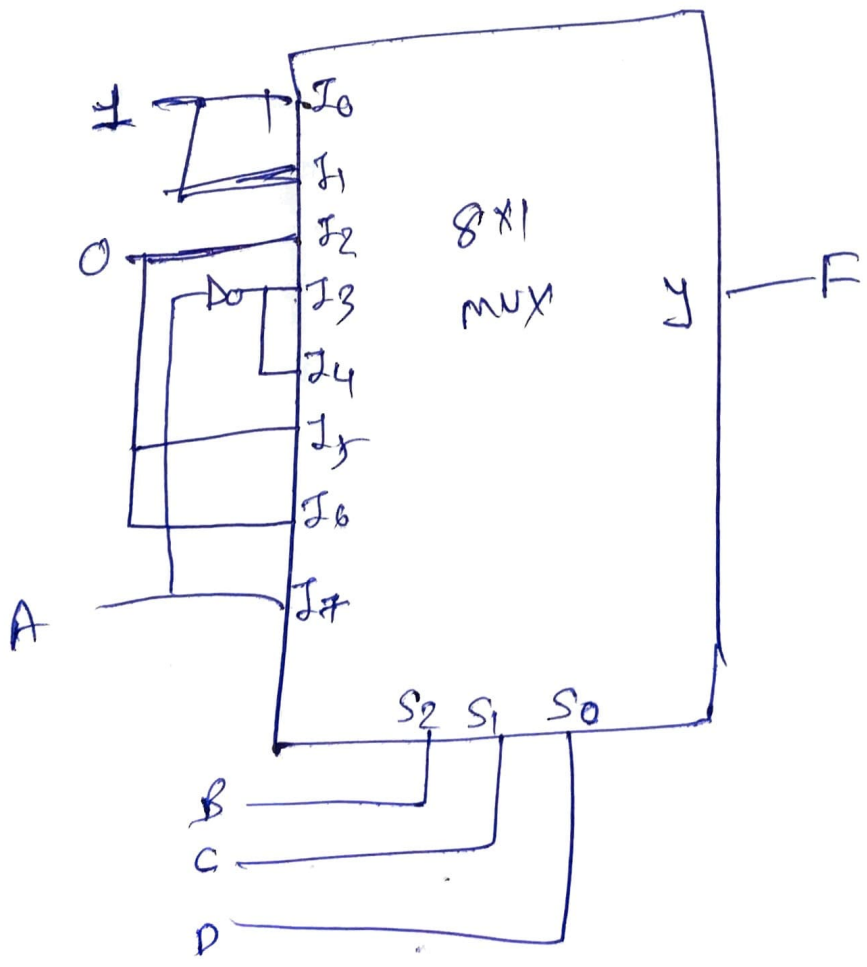
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

→ take A as I/P & B C D as selection line

→ Implementation table

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A1	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	1	0	A1	A1	0	0	A

→ 8:1 MUX Design

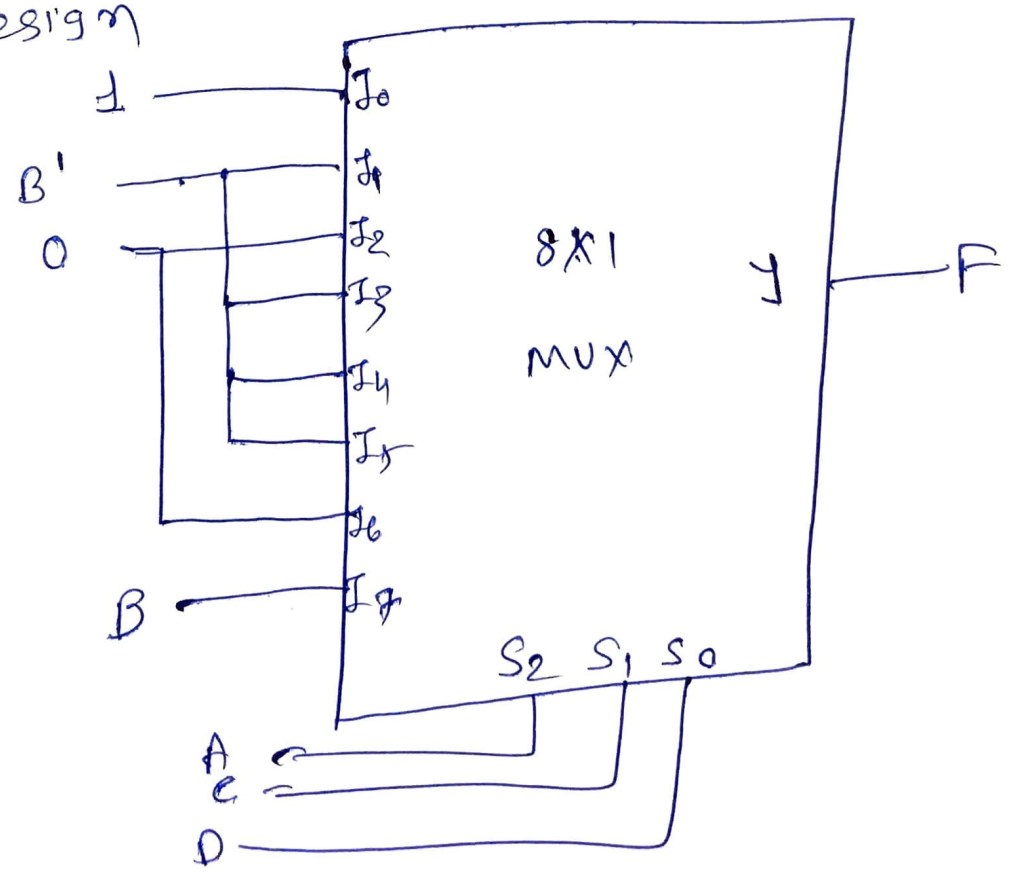


Take B as i/p & A, C, D as Selection line

→ Implementation table

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
B'	(0)	(1)	2	(3)	(8)	(9)	10	11
B	(4)	5	6	7	12	13	14	(15)
	1	B'	0	B'	B'	B'	0	B

→ Design



Ch-5

Examples

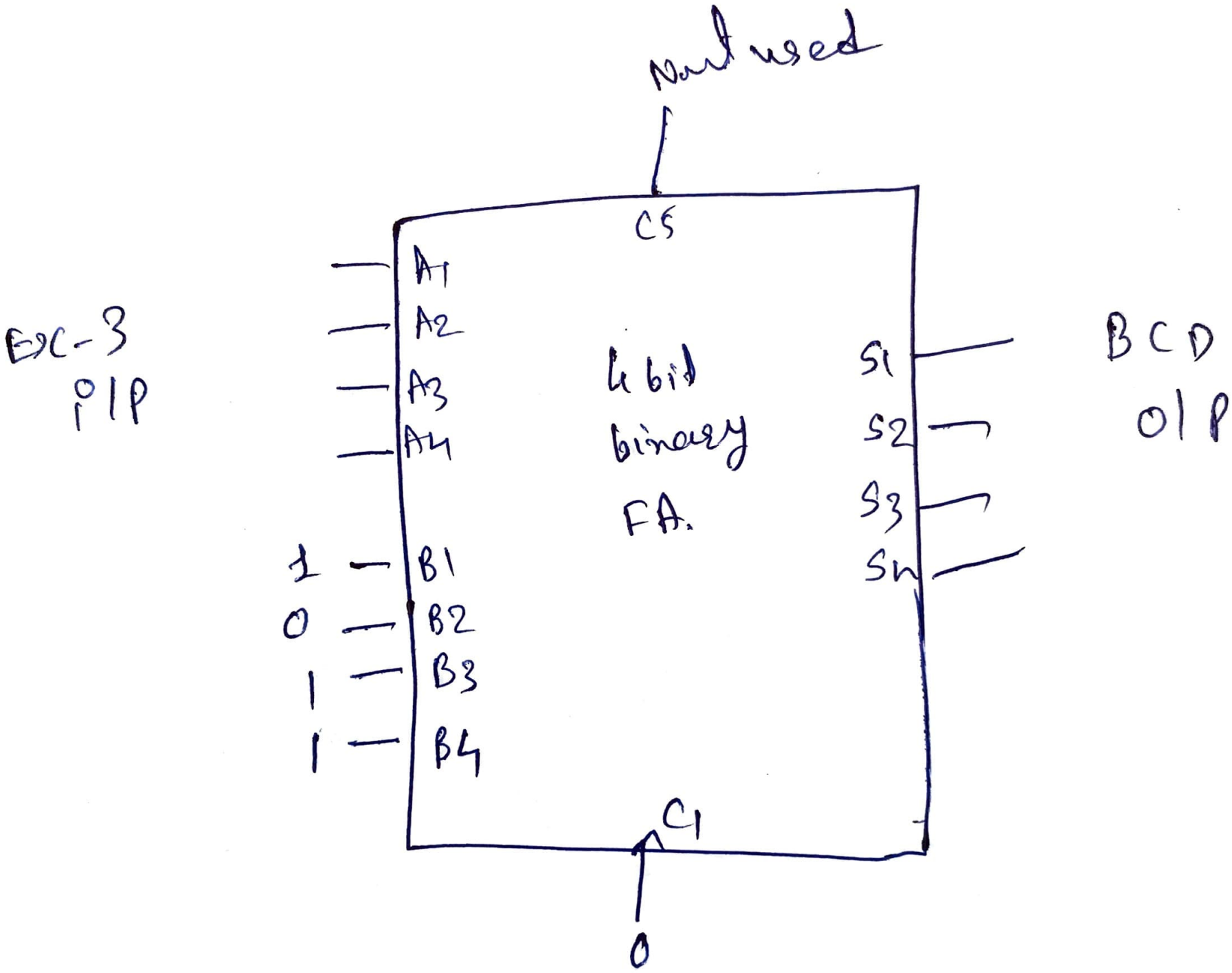
Ex-1 – excess-3 to BCD Code Converter using 4-bit full-adder MSI Ckt.

Excess-3	BCD
A B C D	W X Y Z
0 0 1 1	0 0 0 0
0 1 0 0	0 0 0 1
0 1 0 1	0 0 1 0
0 1 1 0	0 0 1 1
0 1 1 1	0 1 0 0
1 0 0 0	0 1 0 1
1 0 0 1	0 1 1 0
1 0 1 0	0 1 1 1
1 0 1 1	1 0 0 0
1 1 0 0	1 0 0 1

Excess-3 i/p + $(1101)_2$ = BCD Number

$$\begin{array}{r} 1\ 1\ 1 \\ 0\ 0\ 1\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline 0\ 0\ 0\ 0 \end{array}$$

$$\begin{array}{r} 1\ 0\ 0\ 0 \\ +\ 1\ 1\ 0\ 1 \\ \hline 0\ 1\ 0\ 1 \end{array}$$


Not used
C5
A1
A2
A3
A4
4 bit
binary
FA.
B1
B2
B3
B4
C1
0
S1
S2
S3
S4
1
0
1
1
EXC-3
I/P
BCD
O/P

Ex:2 How many don't care i/p are there in a BCD adder?

- There are 9 i/ps to the BCD adder.

A + B + carry

4 bit 4bit 1 bit = 9 bit i/ps

$2^9 = 512$ binary combinations.

But there are only $10 \times 10 = 100$ valid addition (0 to 9) & with carry of 0 or 1 it gives total $100 \times 2 = 200$ valid i/p comb^n.

So, $512 - 200 = 312$ don't care i/ps are there.

Ex-3 Design a BCD to excess-3 code converter with a BCD to decimal decoder & four OR gates

BCD

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

Excess-3

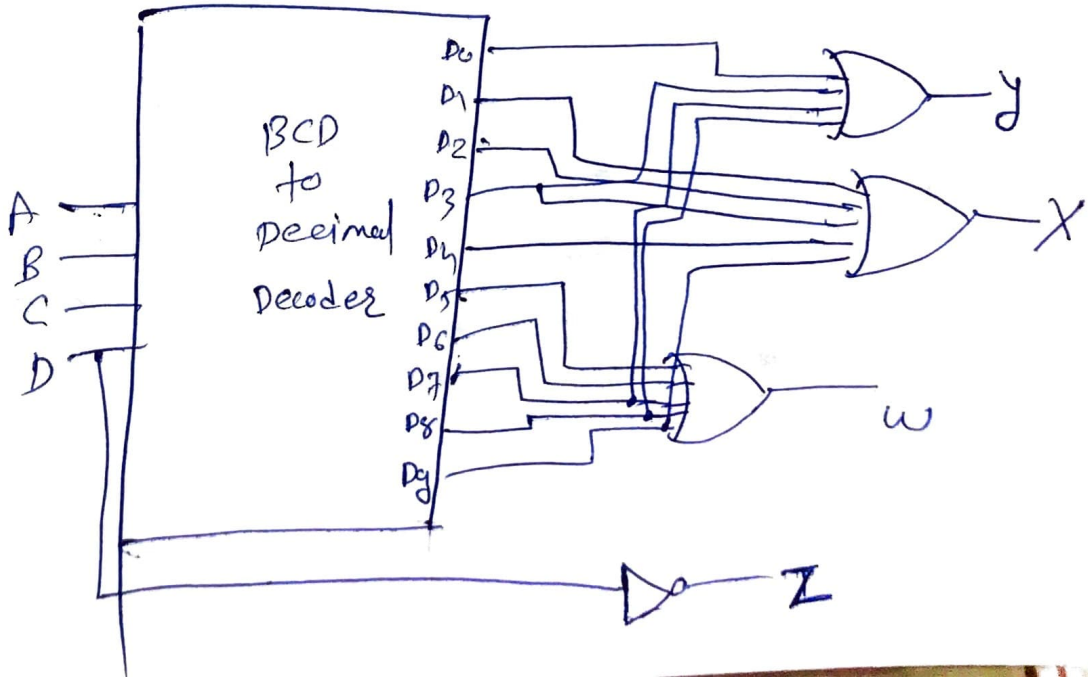
w	x	y	z
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0

$w = \Sigma(5,6,7,8,9)$

$x = \Sigma(1,2,3,4,9)$

$y = \Sigma(0,3,4,7,8)$

$z = D'$



Ex-4 A combinational ckt is defined by following 3 f^n's.

$F_1 = x'y' + xyz'$

$F_2 = x'+y$

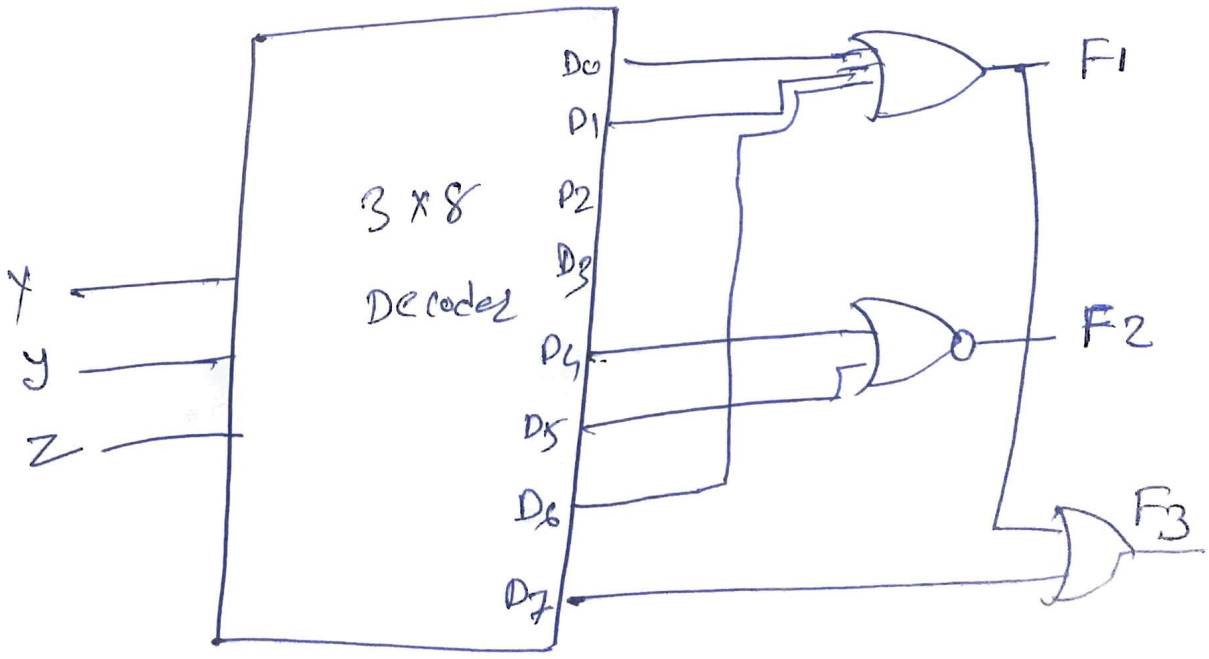
$F_3 = xy + x'y'$

Design the ckt with decoder & External gates.

→ $F_1 = \Sigma(0,1,6)$

$F_2 = \Sigma(0,1,2,3,6,7)$ or $F_2' = \Sigma(4,5)$

$F_3 = \Sigma(0,1,6,7) = F_1 + m_7$



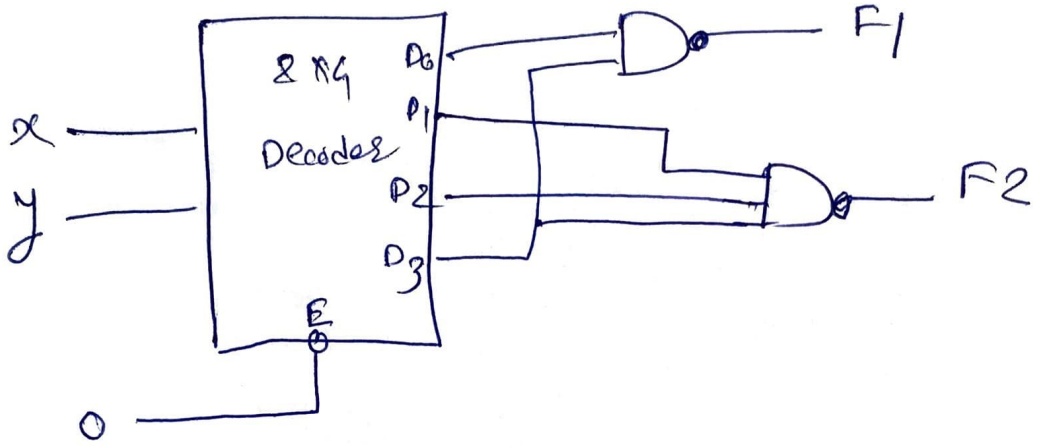
Ex-5 A comb^n ckt is defined by the following two F^{M}s.

$F_1(x,y) = \Sigma(0,3)$

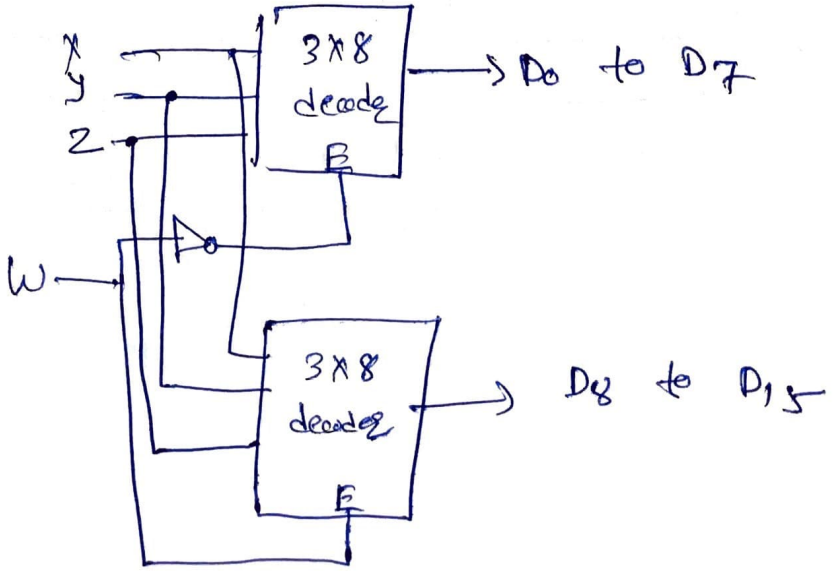
$F2(x,y) = \Sigma(1,2,3)$

Implement the comb^n ckt by means of the decoder & external gates.

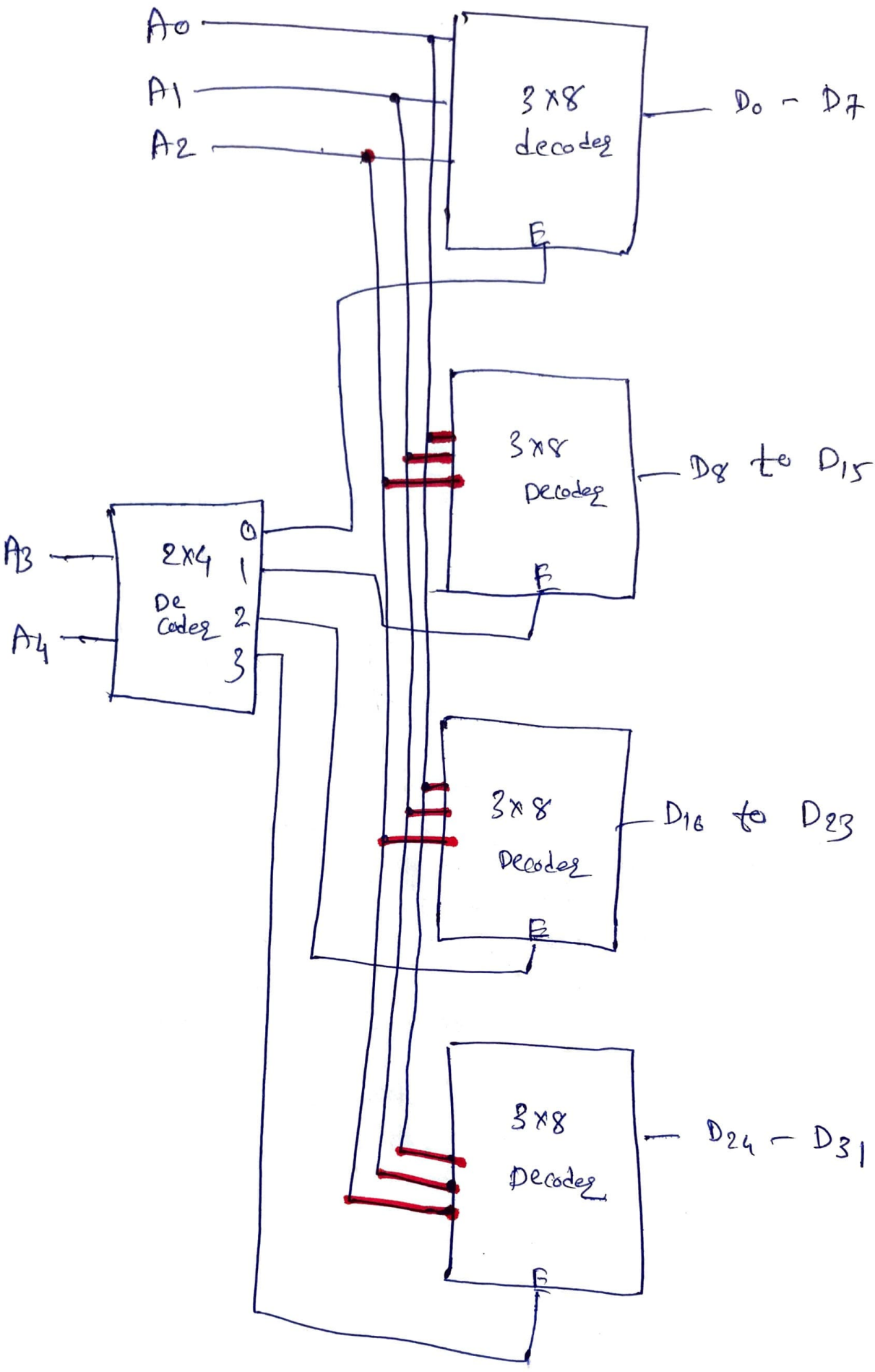
NAND



Ex-6 Construct a 5x32 decoder with four 3x8 decoder / demultiplexer & 2x4 decoder. Use Block diagram.



4x16 Decoder with 2 → 3x8 decoder



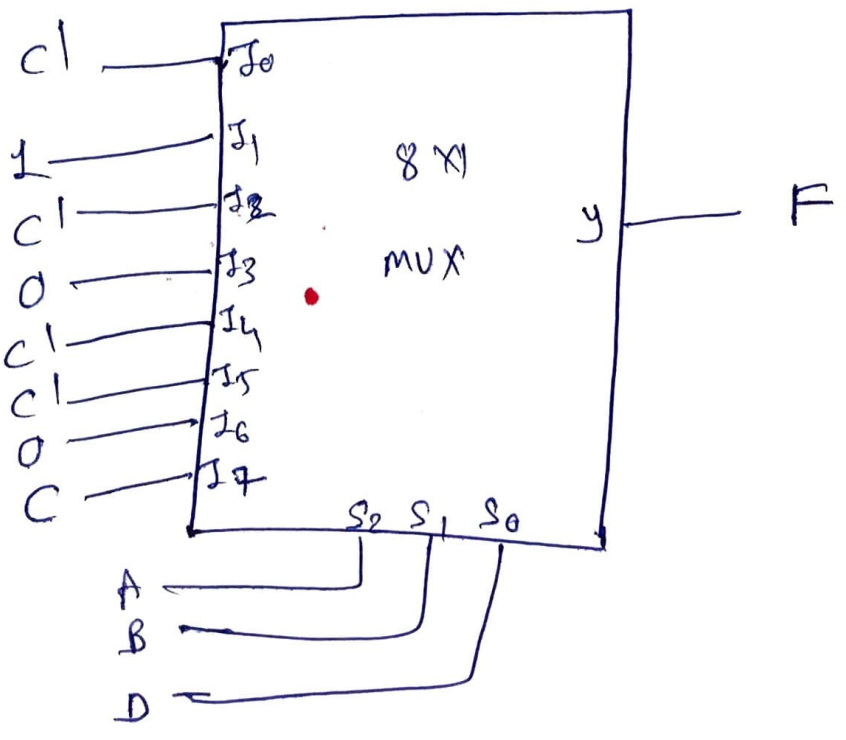
5X32 decoder with 4 → 3x8 & 1 → 2x4 decoder

Ex-7 Implement the Boolean fn with an 8×1 MUX with A, B & D connected to selection lines S_2, S_1 & S_0 respectively.

$$F = \Sigma(0, 1, 3, 4, 8, 9, 15)$$

→ Implementation Table

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
	(0)	(1)	(4)	5	(8)	(9)	12	13
C'								
	2	(3)	6	7	10	11	14	(15)
C								
	C'	1	C'	0	C'	C'	0	C



Ex 6.8 Implement the Comb^n ckt specified by these F^n with a dual 4-line to 1 line mux & an OR gate & inverter

$F_1 = x'y' + xyz' = \Sigma(0,1,6)$

$F_2 = x' + y$

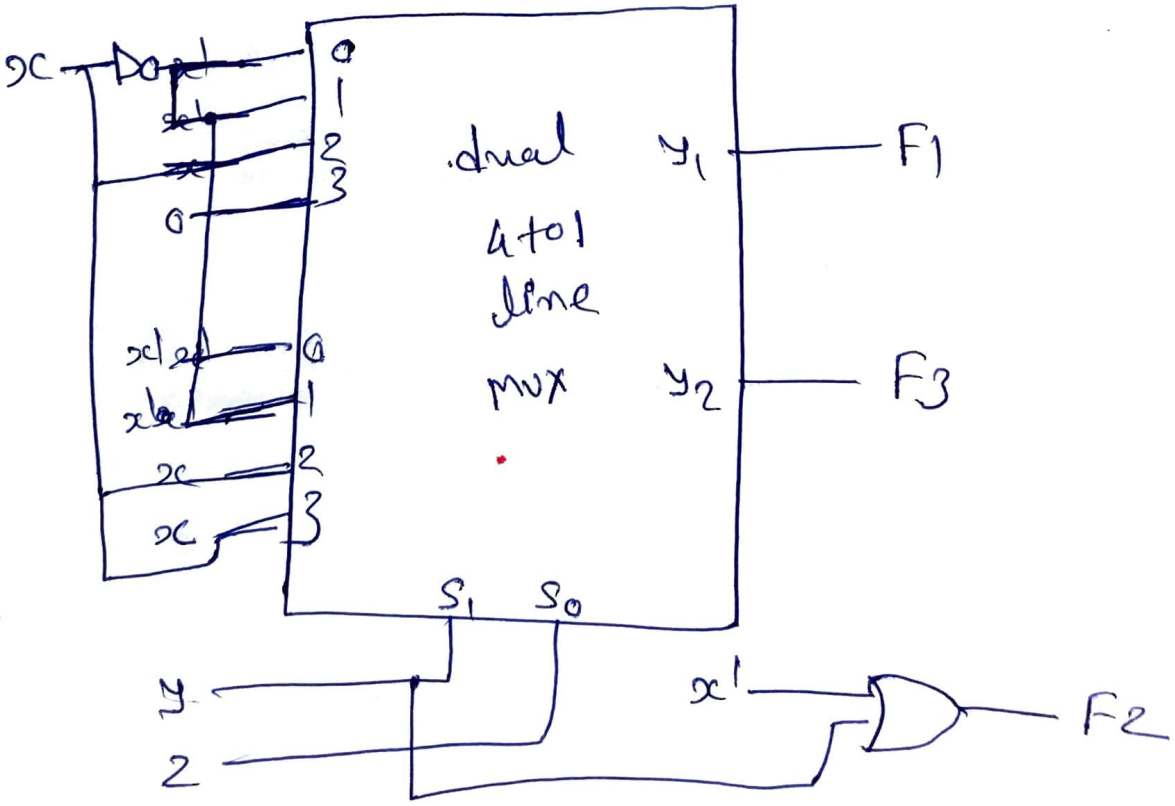
$F_3 = xy + x'y' = \Sigma(0,1,6,7)$

F_1

	I_0	I_1	I_2	I_3
x'	(0)	(1)	2	3
x	4	5	(6)	7
	x'	x'	x	0

F_3

	I_0	I_1	I_2	I_3
x'	(0)	(1)	2	3
x	4	5	(6)	(7)
	x'	x'	x	x



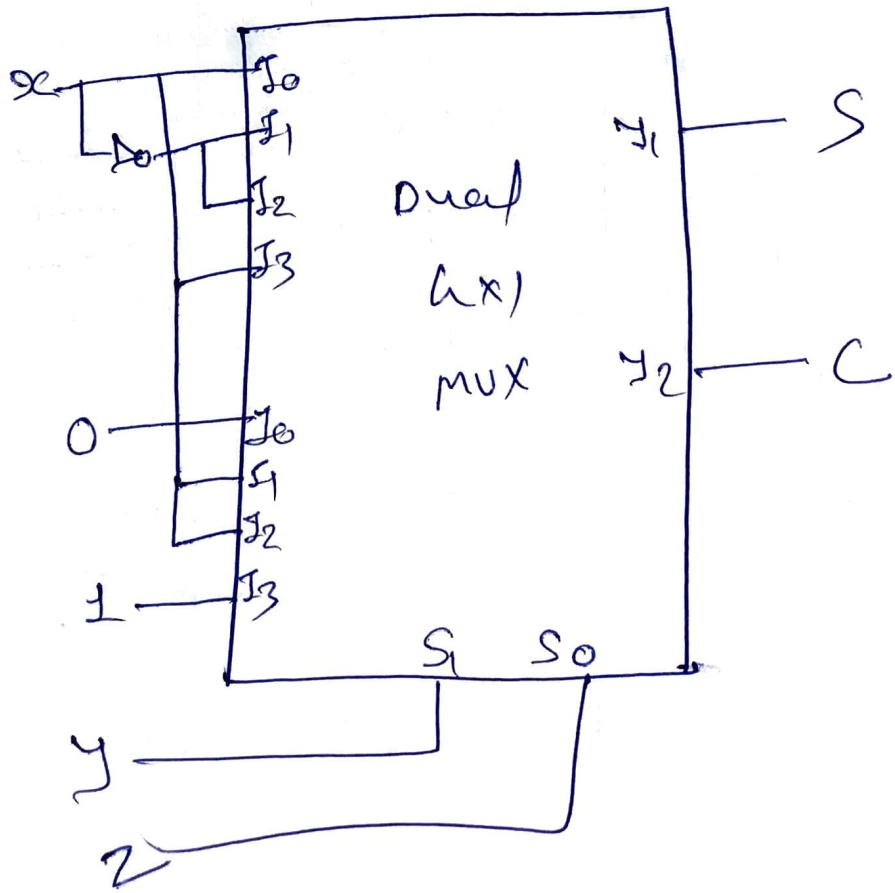
Ex-1 Implement full adder with MUX

$S(x,y,z) = \Sigma(1,2,4,7)$

$C(x,y,z) = \Sigma(3,5,6,7)$

S	I_0	I_1	I_2	I_3
x'	0	(1)	(2)	3
x	(4)	5	6	(7)
	x	x'	x'	x

C	I_0	I_1	I_2	I_3
x'	0	1	2	(3)
x	4	(5)	(6)	(7)
	0	x	x	1



Ex- 10

Obtain. an 8X1 mux with a dual 4 to 1 line mux having separate enable i/p but common selection lines. use Block diagram construction.