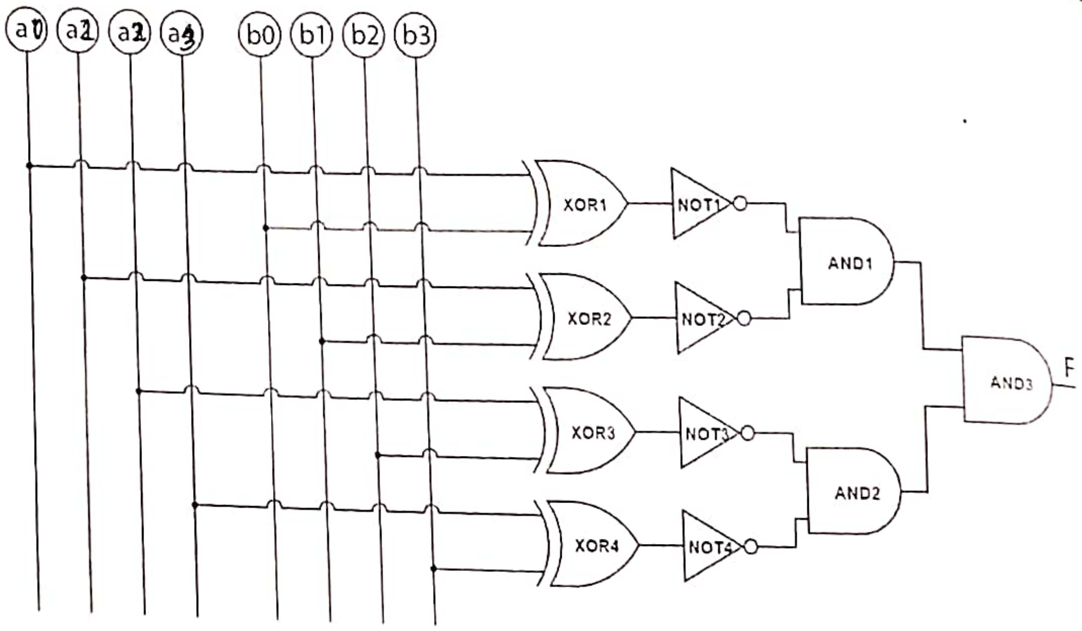
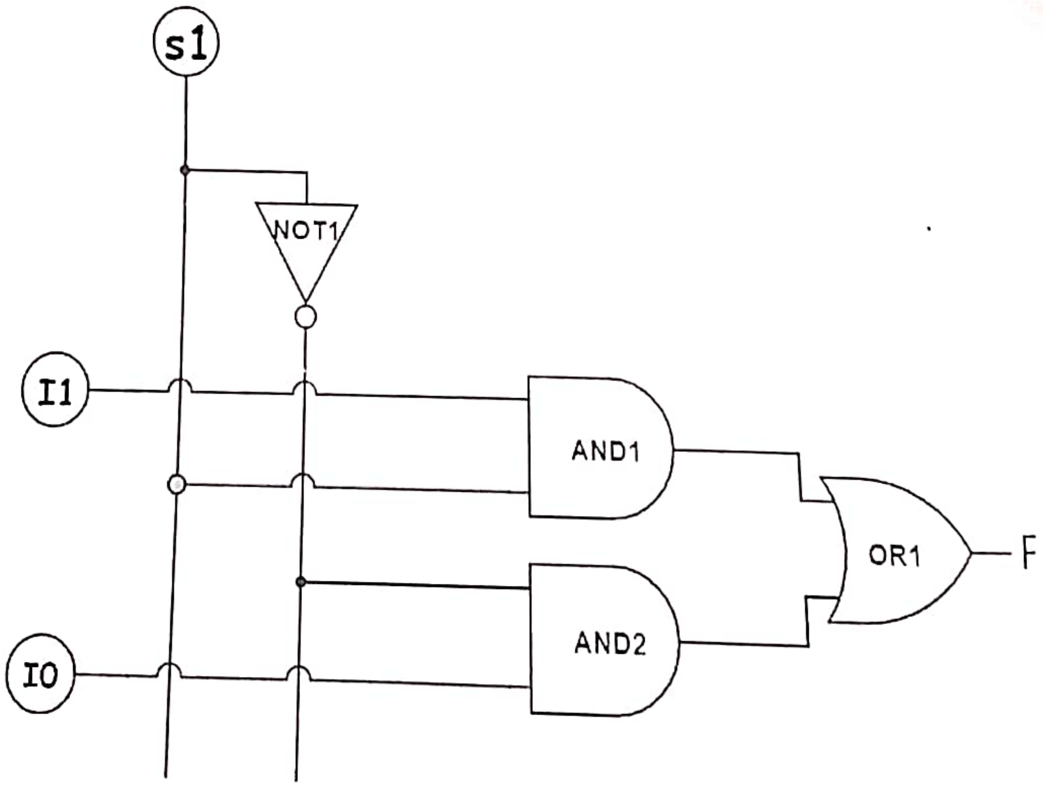
4 bit Comparator + 2x1 Mux



4-bit equality checker



2X1 Mux

PRE-LAB

Objective-1

1. Design a combinational circuit that compares 2-4bit numbers to chec
if they are equal. The circuit output is equal to 1 if the 2
numbers are equal and 0 otherwise.

ⓐ Truth Table

a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	F
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	1	0
1	0	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1

ⓑ Minimized boolean expression:

$$F = (\overline{a_0 \oplus b_0})(\overline{a_1 \oplus b_1})(\overline{a_2 \oplus b_2})(\overline{a_3 \oplus b_3})$$

Objective-2 :-

② Design a 2x1 Multiplexer that will select the binary info.
from one of the 2 input lines and direct it to a single
output line based on the value of selection line.

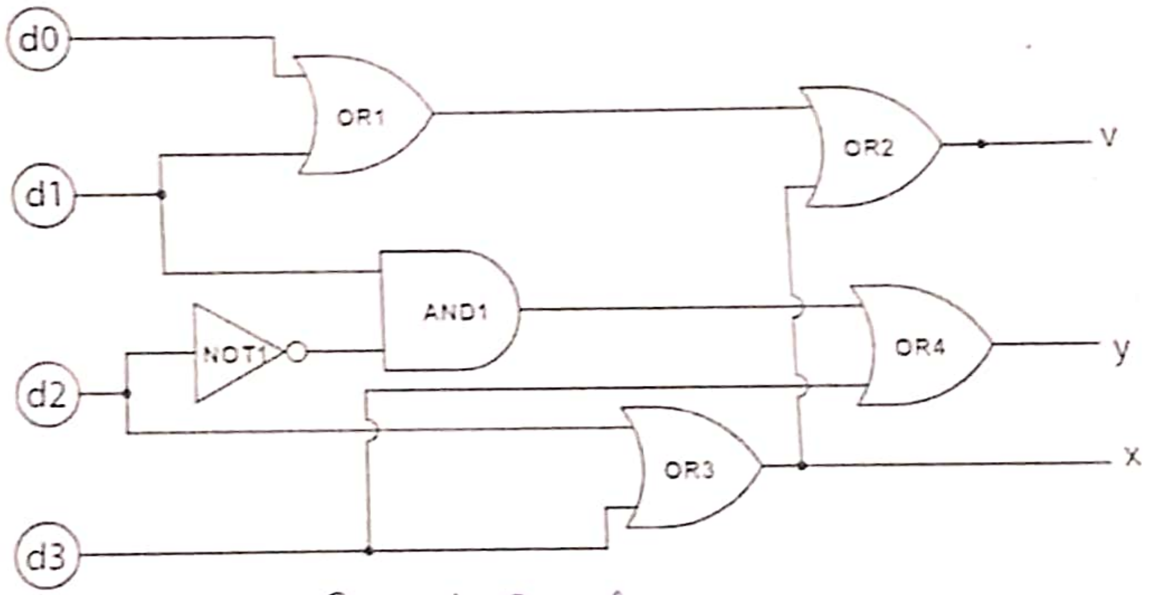
ⓐ Truth Table :-

S_1	F	
0	0	I_0
1	1	I_1

ⓑ Boolean expression :-

$$F = \bar{S} I_0 + S I_1$$

[illegible]



Priority Encoder

Obj-3

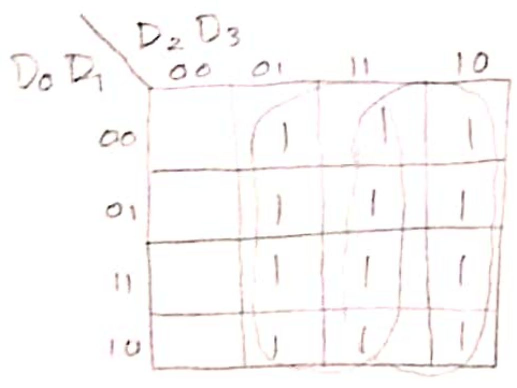
Design a 4-bit priority encoder with inputs D_3 (MSB), D_2, D_1 and D_0 (LSB) and outputs X, Y, and V. The priority assigned to inputs is $D_3 > D_2 > D_1 > D_0$. The output V shows a value 1 where one or more inputs are equal to one. If all inputs are 0, V is equal to 0. When V=0, then other 2 outputs are not inspected and are specified as don't care conditions.

ⓐ Truth Table :-

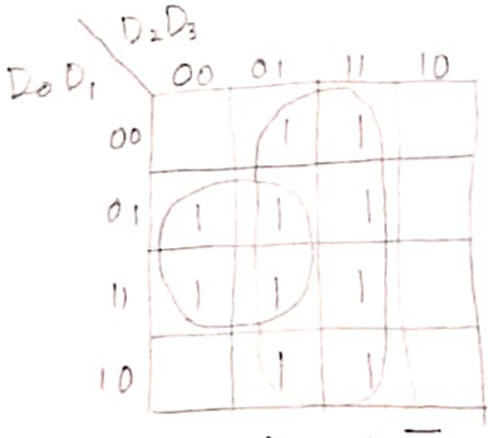
D_0	D_1	D_2	D_3	x	y	v
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

ⓑ Simplified boolean expressions:-

$$V = D_0 + D_1 + D_2 + D_3$$

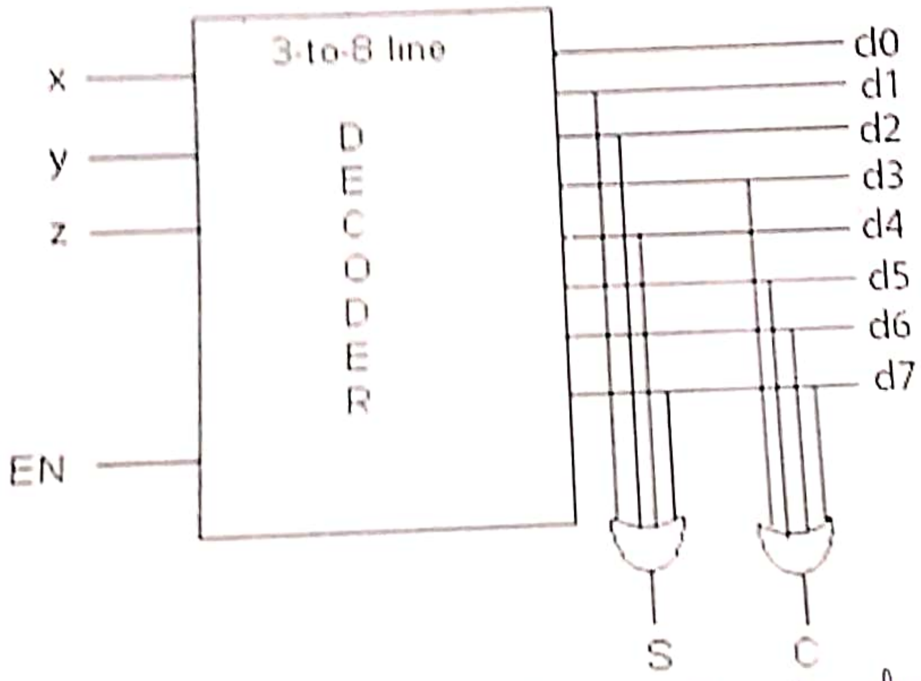


$$x = D_3 + D_2$$



$$y = D_3 + D_1 \overline{D_2}$$

FA using 3-8 line decoder



Full adder using 3-8 Line decoder

4 Obj:- 4

Design a full adder using 3 to 8 line decoder and external OR gates.

ⓐ Truth Table :-

x	y	z	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

ⓑ Simplified boolean expression :-

$S = d_1 + d_2 + d_4 + d_7$

$C = d_3 + d_5 + d_6 + d_7$

III LAB :-

Sl no.	Components	Specification	Quantity
1	XOR-Gate	7486 IC	1
2	NOT-Gate	7404 IC	1
3	AND-Gate	7408 IC	1
4	OR-Gate	7432 IC	1
5	3-inp-OR-Gate	4075 IC	1
6	Wires	23 SVG	As req

HDL Program :-

Obj-1 :-
```
module ckt_4bitChecker (a0, a1, a2, a3, b0, b1, b2, b3, F);
        input a0, a1, a2, a3, b0, b1, b2, b3;
        output F;
        assign F = (!(a0^b0))&&(!(a1^b1))&&(!(a2^b2))&&(!(a3^b3));
endmodule
```

Obj-2 :-
```
mode ckt_2x1Mux (s, i1, i0, F);
        input s, i1, i0;
        output F;
        assign F = (i0 &&(!s))||(I1 &&s);
endmodule
```

Obj-3 :-
```
module ckt_PriorityEncoder(d0, d1, d2, d3, x, y, v);
        input d0, d1, d2, d3;
        output x, y, v;
        assign x = (d3 || d2);
        assign y = (d3 || (d1&&(!d2)));
        assign v = (d0 || d1 || d2 || d3);
endmodule
```

Obj-4 :-
```
module ckt_8x8FA (d1, d2, d3, d4, d5, d6, d7, s, c);
        input d1, d2, d3, d4, d5, d6, d7;
        output s, c;
        assign s = (d1|| d2 || d4 || d7);
        assign c = (d3|| d5|| d6|| d7);
endmodule
```

Observation :-

Obj-1

Req. F	Actual F	Status
0	0	✓
0	0	✓
0	0	✓
0	0	✓
0	0	✓
1	1	✓
1	1	✓

Obj-2

Req F	Actual F	Status
0	0	✓
1	1	✓

Obj-3

Required			Actual			Status
x	y	v	x	y	v	
×	×	0	×	×	0	✓
0	0	1	0	0	1	✓
0	1	1	0	1	1	✓
1	0	1	1	0	1	✓
1	1	1	1	1	1	✓

Obj-4

Req. S	Req. C	Actual S	Actual C	Status
0	0	0	0	✓
1	0	1	0	✓
1	0	1	0	✓
0	1	0	1	✓
1	0	1	0	✓
0	1	0	1	✓
0	1	0	1	✓
1	1	1	1	✓

Conclusion

Obj-1
We've designed a circuit to compare 2 4-bit no. to check if they are equal or not and verified it

Obj-2
We've designed a 2x1 Mux and also verified it

Obj-3
We've designed a 4-bit priority encoder i.e $D_3 > D_2 > D_1 > D_0$ & hence also verified it.

Obj-4
We've designed a full adder using 3 to 8 line decoder and external OR gates and verified both sum and carry.

IV: POST-LAB

Q1 Logically derive the boolean expressions for the output variables of a 4-bit magnitude comparator.

$(A_0 \odot B_0)(A_1 \odot B_1)$ ↑

$Z(A=B) =$ ~~$A_3B_3 \cdot A_2B_2 \cdot A_1B_1 \cdot A_0B_0$~~

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

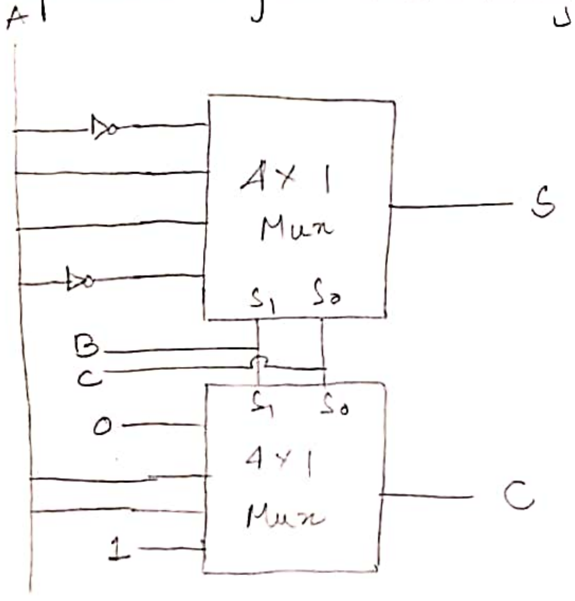
$X(A<B) = \bar{A}_1B_1 + \bar{A}_0B_1B_0 + \bar{A}_1\bar{A}_0B_0$

$Z(A>B) = A_1\bar{B}_1 + A_0\bar{A}_1\bar{B}_0 + A_1A_0\bar{B}_0$

4. Why is a multiplexer known as data selector?

The selection of a particular input data line for the output is decided on the basis of selection lines. The multiplexer is often called as data selector since it selects one one of many data inputs.

5. Implement a full adder using 2 4x1 multiplexers.



Sum →

A \ BC	00	01	10	11
0	0	①	②	3
1	④	5	6	⑦
	$\bar{A}$	A	A	$\bar{A}$

Carry →

A \ BC	00	01	10	11
0	0	1	2	③
1	4	⑤	⑥	⑦
	0	A	A	1