II Bre Lab

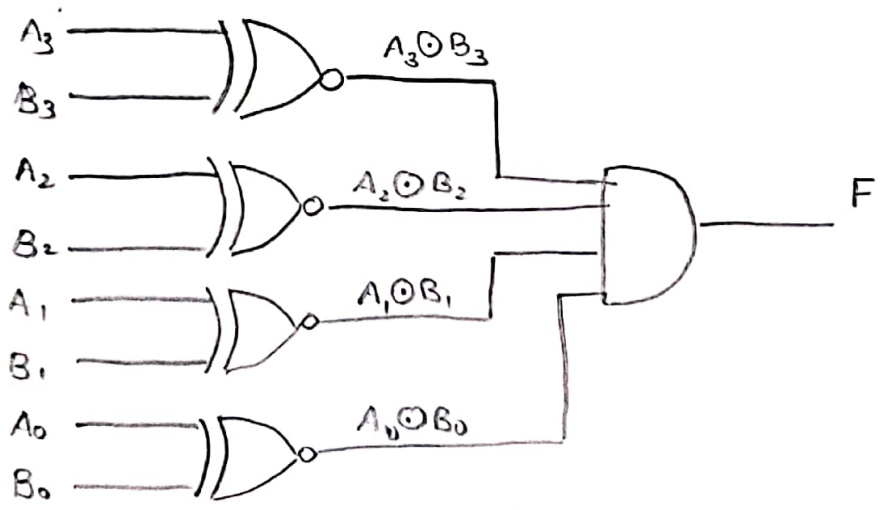
Objective 1

a)

A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	$A = B$
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	0	1
0	0	1	1	0	0	1	1	1
0	1	0	0	0	1	0	0	1
0	1	0	1	0	1	0	1	1
0	1	1	0	0	1	1	0	1
0	1	1	1	0	1	1	1	1
1	0	0	0	1	0	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	0	1
1	0	1	1	1	0	1	1	1
1	1	0	0	1	1	0	0	1
1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1	1

b) $F = (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$

c)



d)

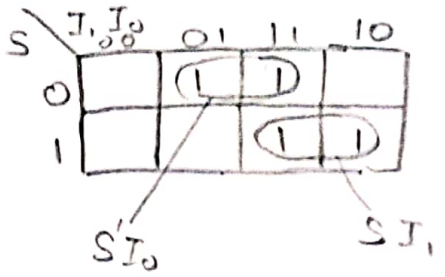
```
module 4bit_comparator(
    input A3;
    input A2;
    input A1;
    input A0;
    input B3;
    input B2;
    input B1;
    input B0,
    output F
    );
assign F = (A3 ~^ B3) & (A2 ~^ B2) & (A1 ~^ B1) & (A0 ~^ B0);
endmodule.
```

Objective 2

a)

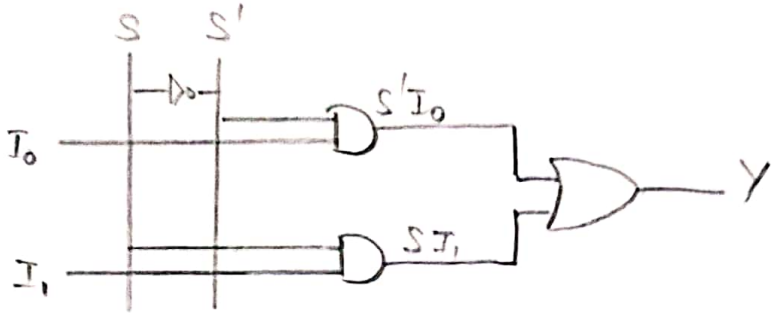
S	I_1	I_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

b)



$Y = S'I_0 + SI_1$

c)



d)
```
module mod(
    input S, I0, I1;
    output Y
    );
~~wire w~~
assign Y = ((~S & A0) || (S & A1));
endmodule
```

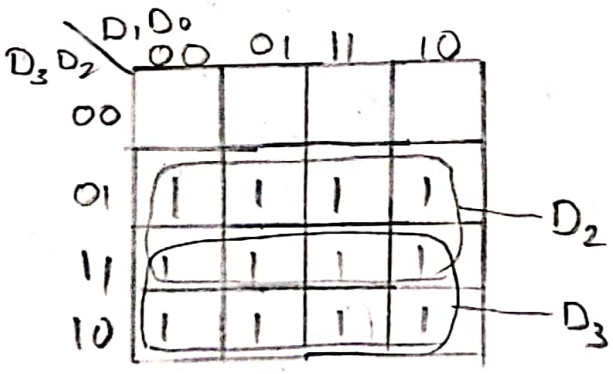
Objective 3

a)

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

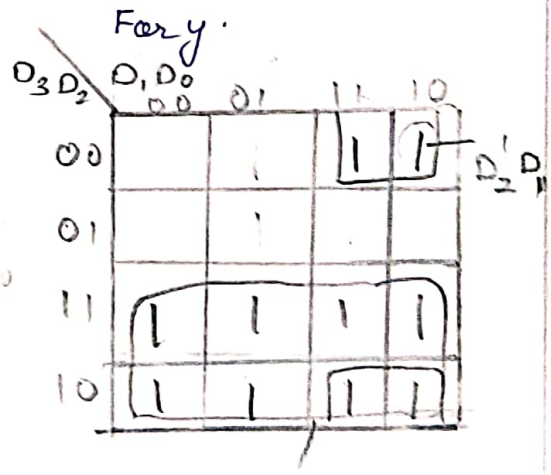
D_3	D_2	D_1	D_0	x	y	v
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	

b) For x

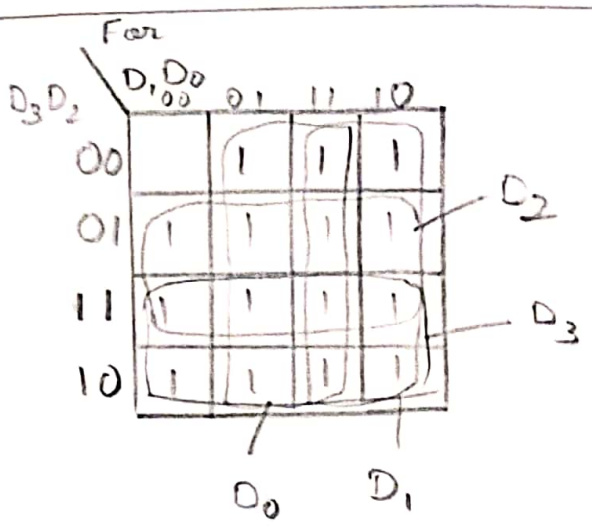


$x = D_2 + D_3$

For y.

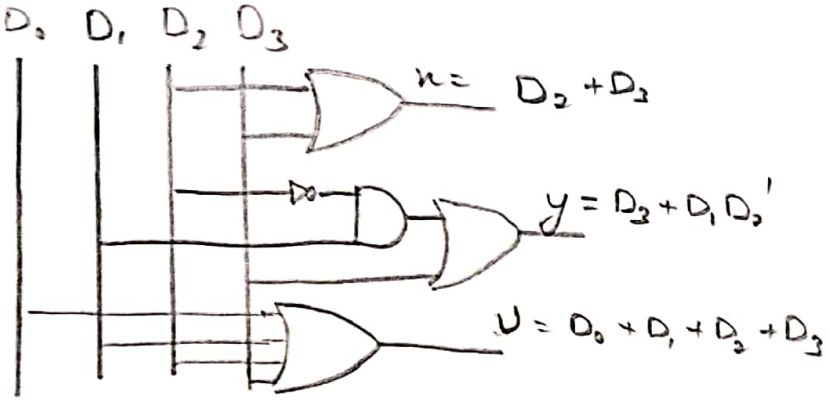


$y = D_3 + D_2' + D_1$



$v = D_0 + D_1 + D_2 + D_3$

c)



d)

```
module pri_enc(
  input D3, D2, D1, D0,
  output x, y, v
  );
assign v = D3 | D2 | D1 | D0;
assign x = D3 | (D1 & (~D2))
assign y = D2 | D3;
end module.
```

Objective 4

a)

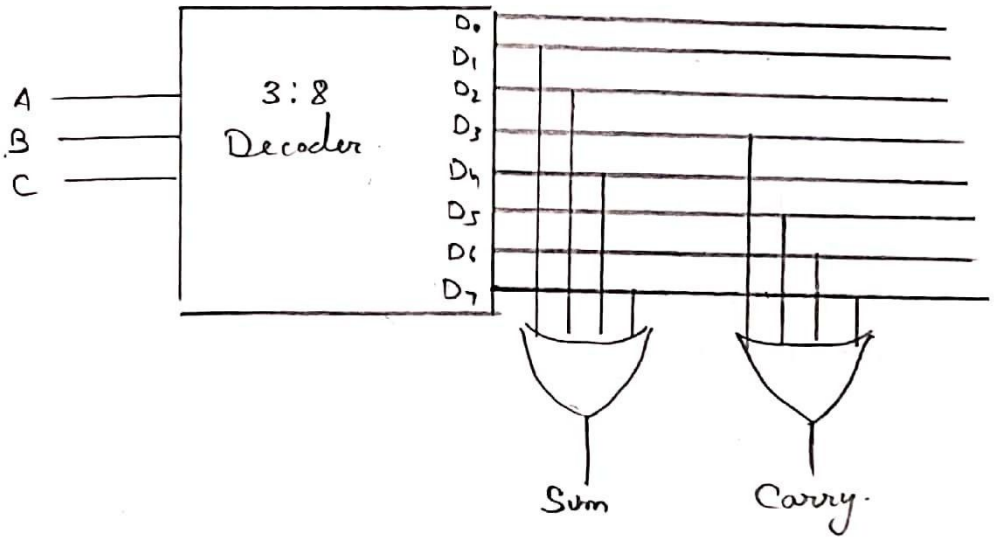
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

b)

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

$Carry = \Sigma(3,5,6,7)$

c)



d) module lab6Q4;

```
input A, B, C;
output Sum, Carry, D0, D1, D2, D3, D4, D5, D6, D7;
);
assign D0 = ~A && ~B && ~C;
assign D1 = ~A && ~B && C;
assign D2 = ~A && B && ~C;
assign D3 = ~A && B && C;
assign D4 = A && ~B && ~C;
assign D5 = A && ~B && C;
assign D6 = A && B && ~C;
assign D7 = A && B && C;
assign Sum = ~A && ~B && C || ~A && B && ~C || A && ~B && ~C ||
             A && B && C;
assign Carry = B && C || A && C || A && B;
end module.
```

III LAB

Component Required:

Objective 1

S.No	Item	Specification	Quantity
1	XOR gate	IC-7486	1
2	NOT gate	IC-7404	1
3	AND gate	IC-7408	1
4	Connecting Wire	23 SWG	As per required

Objective 2

S. No	Item	Specification	Quantity.
1	NOT gate	IC-7404	1
2	AND gate	IC-7408	1
3	OR gate	IC-7432	1
4	Connecting Wire	23SWG	As per required.

Objective 3

S.No	Item	Specification	Quantity.
1	NOT gate	IC-7404	1
2	AND gate	IC-7408	1
3	OR gate	IC-7432	1
4	Connecting Wire	23SWG	As per required.

Objective 4.

S. No	Item	Specification	Quantity.
1	3 to 8 Decoder	IC 74139	1
2	OR gate	IC-7432	1
3	Connecting Wire	23SWG	As per required.

Observation:

Objective 1

A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	A = B
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	0	1
0	0	1	1	0	0	1	1	1
0	1	0	0	0	1	0	0	1
0	1	0	1	0	1	0	1	1
0	1	1	0	0	1	1	0	1
0	1	1	1	0	1	1	1	1
1	0	0	0	1	0	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	0	1
1	0	1	1	1	0	1	1	1
1	1	0	0	1	1	0	0	1
1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1	1

Objective 2

S	I_1	I_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Objective 3

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

Objective 4

A	B	C	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

Conclusion:

Objective 1: It can be concluded that to design a combinational circuit for a 4 bit magnitude comparator 4 XNOR and 3 AND gates are required

Objective 2: It can be concluded that to design a combinational circuit for 2×1 Multiplexer 2 AND and 1 OR gates are required and circuit leads to the function

$$Y = S'I_0 + SI_1$$

Objective 3: It can be concluded that to design a combinational circuit for 4 bit priority encoder 3 OR, 1 AND, 1 NOT gates are required and circuit leads to the function.

$$x = D_2 + D_3$$
$$y = D_3 + D_1D_2'$$
$$v = D_0 + D_1 + D_2 + D_3$$

Objective 4: It can be concluded that to implement a full adder using 3-to-8 line decoder and external OR gates a decoder and 2 or gates are required and circuit leads to function.

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

IVPOST LAB

1. $A > B: A_3 B_3' + (A_3 \odot B_3) A_2 B_2' + (A_3 \odot B_3)(A_2 \odot B_2) A_1 B_1' + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1) A_0 B_0'$

$A = B: (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$

$A < B: A_3' B_3 + (A_3 \odot B_3) A_2' B_2 + (A_3 \odot B_3)(A_2 \odot B_2) A_1' B_1 + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1) A_0' B_0$

2. Multiplexer is known as data selector because it selects which data input is to be sent.

3

x	y	z	C	S	C	S
0	0	0	0	0	} 0	} z
0	0	1	0	1		
0	1	0	0	1	} z	} $\bar{z}$
0	1	1	1	0		
1	0	0	0	1	} z	} $\bar{z}$
1	0	1	1	0		
1	1	0	1	0	} 1	} z
1	1	1	1	1