1) PRE-LAB →

For Obj. 1 → Design a combinational circuit with four inputs A, B, C & D and one output F. The output F value is 0 if three or four of the inputs are 1; otherwise the value of F is 1.

a) Truth table →

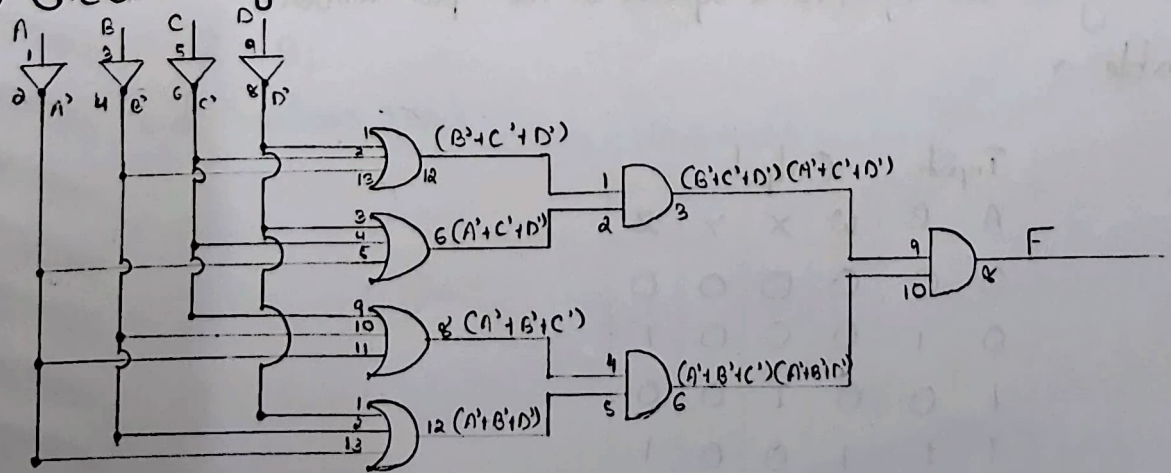
Inputs				Outputs
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

b) Minimised Boolean Function →

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	0	1
11	1	0	0	0
10	1	1	0	1

$$F_{max} = (B'+C'+D')(A'+C'+D')(A'+B'+C')(A'+B'+D')$$

c) Circuit Diagram →



d) HDL code →

```
library ieee;
use ieee.std_logic_1164.all;
entity test10 is
    port (A, B, C, D : in std_logic;
                                    F : out std_logic);
    end test10;
architecture boolean_eq of test10 is
    signal a, b, c, d, e, f, g, h, i, j : std_logic;
    Begin
        a <= NOT A;
        b <= NOT B;
        c <= NOT C;
        d <= NOT D;
        e <= b OR c OR d;
        f <= a OR c OR d;
        g <= a OR b OR c;
        h <= a OR b OR d;
        i <= e AND f;
        j <= g AND h;
        F <= i AND j;
end boolean_eq;
```

For Obj. 2 → Design a combinational circuit that accepts a 2-bit number and gene an output binary number equal to the square of the input number.

a) Truth table →

Inputs		Outputs			
A	B	W	X	Y	Z
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1

b) Minimised Boolean Expression →

For W →

A \ B	0	1
0	0	0
1	0	1

$W = AB$

For X →

A \ B	0	1
0	0	0
1	1	0

$X = AB'$

For Y →

A \ B	0	1
0	0	0
1	0	0

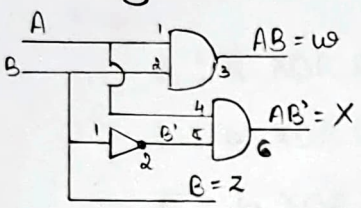
$Y = 0$

For Z →

A \ B	0	1
0	0	1
1	0	1

$Z = B$

c) Circuit Diagram →



d) HDL code →

```
library ieee;
use ieee.std_logic_1164.all;
entity test2 is
    port ( A, B : in std_logic;
           W, X, Z : out std_logic );
end test2;
architecture boolean_eq of test2 is
    signal a : std_logic;
Begin
    a <= NOT B;
    W = A AND B;
    X = A AND a;
    Z = B;
end boolean_eq;
```

For Obj. 3 → Design, construct and test a circuit that generates an even parity bit from four message bits.

a) Truth table →

Input				Output
A	B	C	D	F
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

b) Minimized Boolean Circuit →

For F →

AB \ CD	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$F = A'B'C'D + A'B'CD' + A'BC'D' + A'BCD + ABC'D + ABCD' + ABCD + AB'C'D' + AB'CD$

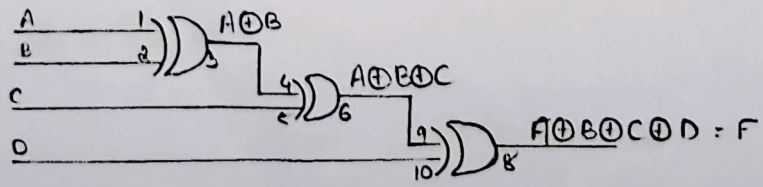
$= A'B'(C'D + CD') + A'B(C'D' + CD) + AB(C'D + CD') + AB'(C'D' + CD)$

$= (A'B' + AB)(C'D + CD') + (A'B + AB')(C'D' + CD)$

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

$= A \oplus B \oplus C \oplus D$

c) Circuit Design →



d) HDL code →

```
library ieee;
use ieee.std_logic_1164.all;
entity test3 is
    port (A, B, C, D : in std_logic;
          F : out std_logic;)
end test3;
architecture boolean_eq of test3 is
    signal a, b : std_logic;
Begin
    a : A XOR B;
    b : a XOR C;
    F : b XOR D;
end boolean_eq;
```

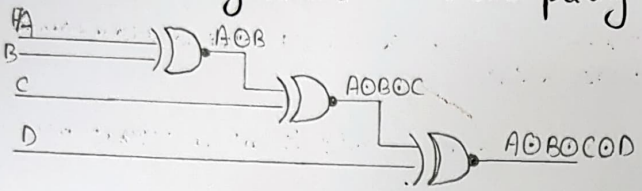
IV) POST LAB →

Q1) What do you understand by the term majority logic?

Ans→ Majority logic, a type of Boolean logic, is defined to be true if more than half of the n inputs are true, where n is odd.

Q2) Suggest a suitable modification to be made in existing even parity circuit that can be used to generate bit for odd parity.

Ans→



Q3) What is the function of a magnitude comparator circuit?

Ans→ A digital comparator or magnitude comparator ~~circuit~~ is a hardware electronic device that takes two numbers as input in binary form & determines whether one number is greater than, less than or equal to the other number.

III) LAB →

Obj.1→ It can be concluded for this combinational circuit we need 4 not gates, 4 ~~ex~~ or gates and 3 and gates.

Obj.2→ It can be concluded for this combinational circuit we need 2 and gates and 1 not gate.

Obj.3→ It can be concluded for this combinational circuit we need 3 ex-or gates to generate a even parity bit from 4 message bits.