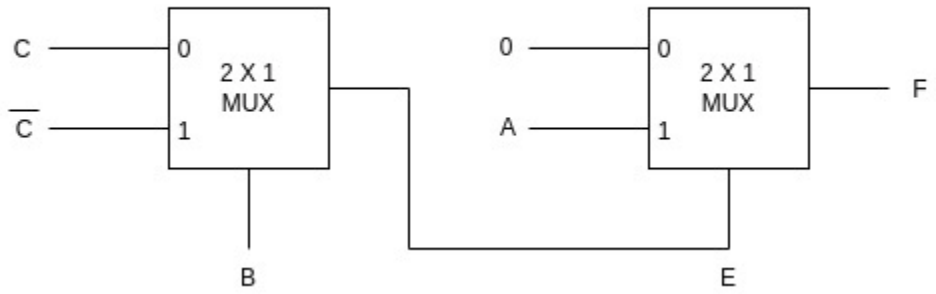
Module: R3: DLD + DSD
Section: Combinational Circuits **Task:** Assignment 2

Assignment 2

Combination Circuits

➢ **Question 1:**



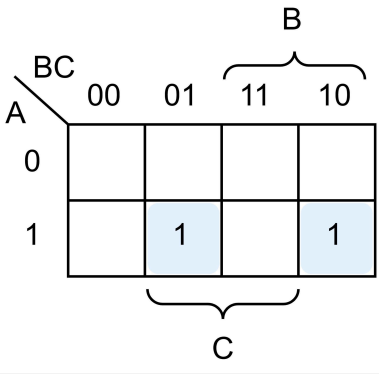
■ **Solution:**

a. Find the Minimal Expression using K-Map:

The truth table of the given circuit can be obtained by applying all the possible combinations for three inputs A, B and C. Here's the obtained truth table:

Truth table			
A	**B**	**C**	**F**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Using K-maps:



The simplified Boolean Expression is:

$$F = AB'C + ABC' = A\,(B \oplus C)$$

b. Verilog code of the circuit:

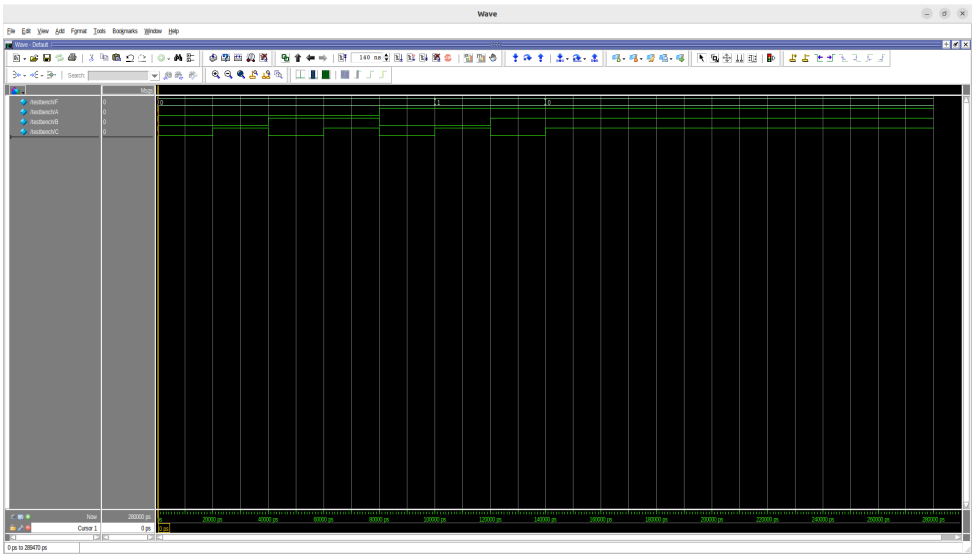
```
module mux_2x1 (input A, B, C, output reg F);


always @*
assign F = A & (B ^ C);
endmodule
```

c. Testbench:

```
      //Author: Noman Rafiq
//Dated: July 10, 2024

`timescale 1ns/10ps

module testbench;
wire F;
reg A, B, C;
mux_2x1 dut(.A(A), .B(B), .C(C), .F(F));

initial begin
      A=1'b0;
      B=1'b0;
      C=1'b0;
      #20;

      A=1'b0;
```

```
            B=1'b0;
            C=1'b1;
            #20;

            A=1'b0;
            B=1'b1;
            C=1'b0;
            #20;

            A=1'b0;
            B=1'b1;
            C=1'b1;
            #20;

            A=1'b1;
            B=1'b0;
            C=1'b0;
            #20;

            A=1'b1;
            B=1'b0;
            C=1'b1;
            #20;

            A=1'b1;
            B=1'b1;
            C=1'b0;
            #20;

            A=1'b1;
            B=1'b1;
            C=1'b1;
            #20;
      end
      endmodule
```

d. Output Waveform:

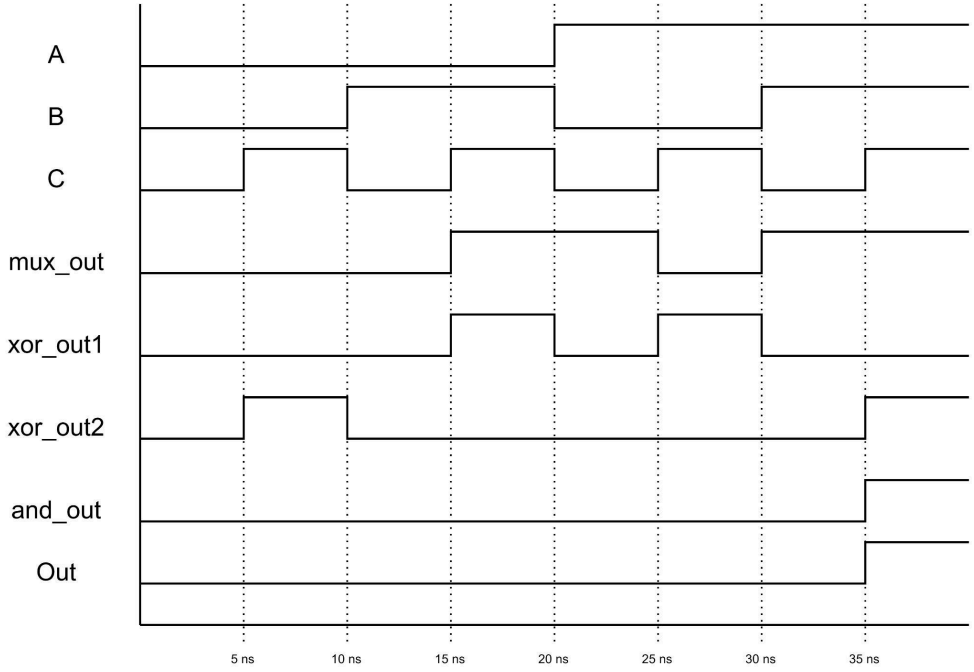


The output F is only 1 when the input combinations for A, B and C are either 101 (m5) or 110 (m6) as illustrated by the truth table in **part a**.

➢ **Question 2 - Timing Diagram:**

- **Solution:**

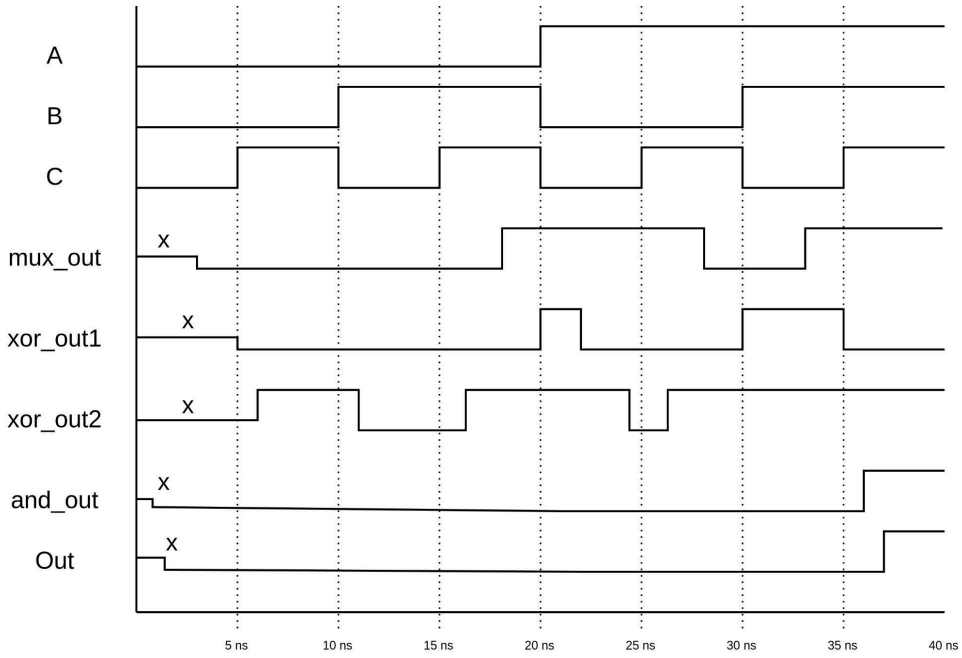
  Let's first draw the ideal timing diagram so we can identify the glitches in our circuit. Here's the idea diagram which doesn't take the delays into consideration.



Ideal Case

Let's cater the gate delays. Assuming the 2x1 mux has a delay of 3 ns (adding the 1ns delay from the not gate inside to the 2ns of and and or gates).

Here's the timing diagram. The diagram clearly shows the output has glitches which start arising at xor_out2 and propogates to final Out.



The glitches can be resolved using a set of not gates and buffers in lines C and A feeding to gates xor_out1 and xor_out2 in such a way that the the delays for each line of input for respective gates are equal. In other words, both of the inputs should be available to the gate at the same time.

- **Question 3 - Divisible by 3, 8421 weighted circuit Design:**
  - **Solution:**

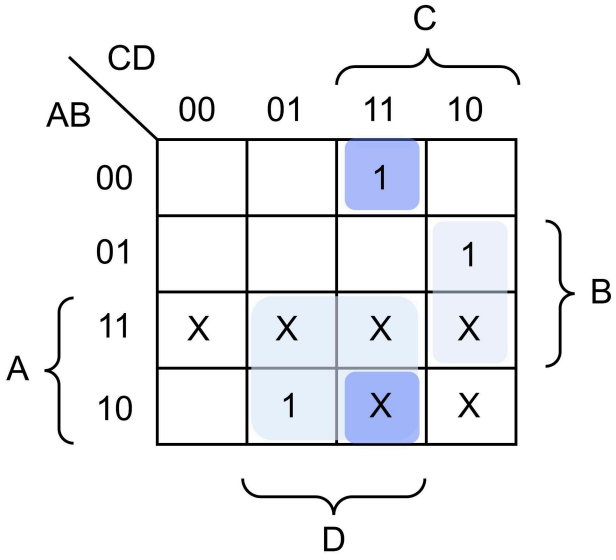
    The 4-bit 8421 weighted code is actually a BCD code which is used to represent decimal digits from 0 to 9 and this code is invalid for numbers greater than 9.

    Hence, we'll be using those invalid combinations as don't care conditions. Here's the truth table:

Truth table				
A	**B**	**C**	**D**	**F**

0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1

Using K-Maps:

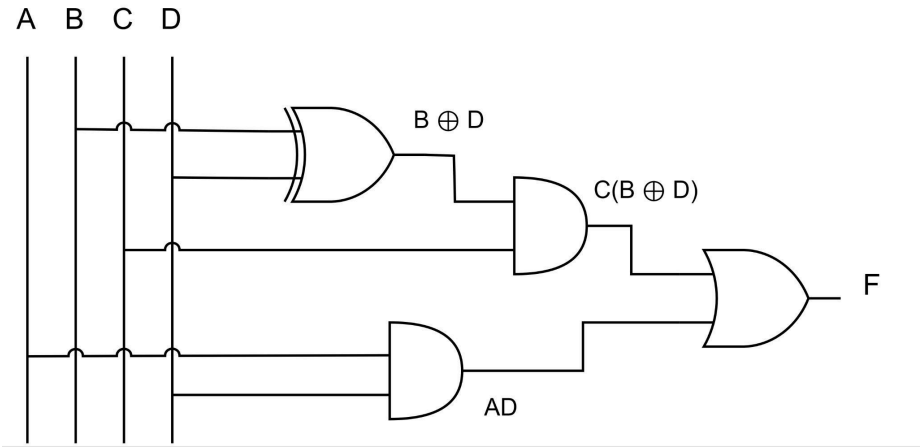


The simplified expression, hence, is given as:

F = B'CD + BCD' + AD = C (B'D + BD') + AD

F = C (B ⊕ D) + AD

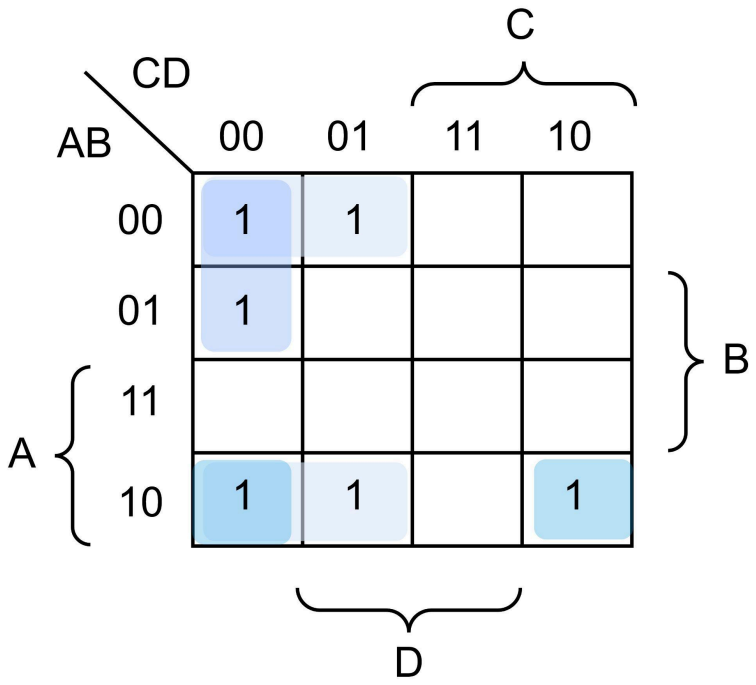
The logic Diagram of the function is:



➢ **Question 4 - Simplify the Boolean function using K-MAP technique:**

a. **F(A,B,C,D) = Σ(0,1,4,8,9,10)**

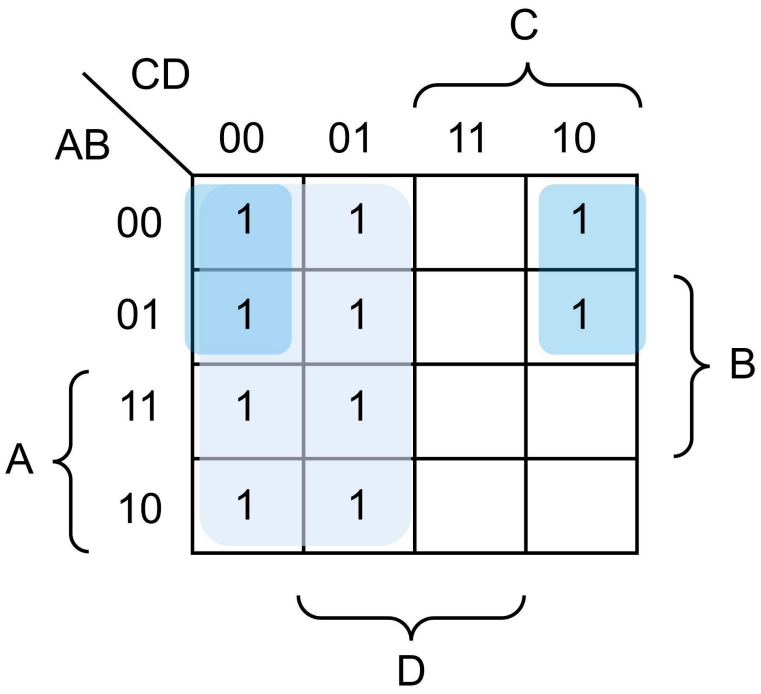
Using K-Map:



Essential Implicants: A'C'D', B'C', AB'D'
Non-Essential Implicants: None

b. **F(A,B,C,D) = ∑(0,1,2,4,5,6,8,9,12,13)**

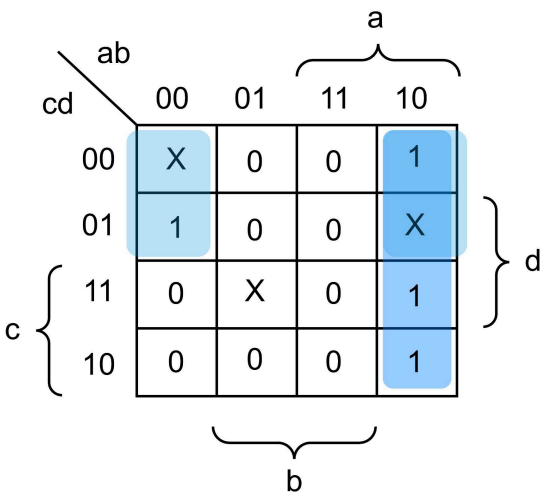
Using K-Map:



Essential Implicants: C', A'D'
Non-Essential Implicants: None

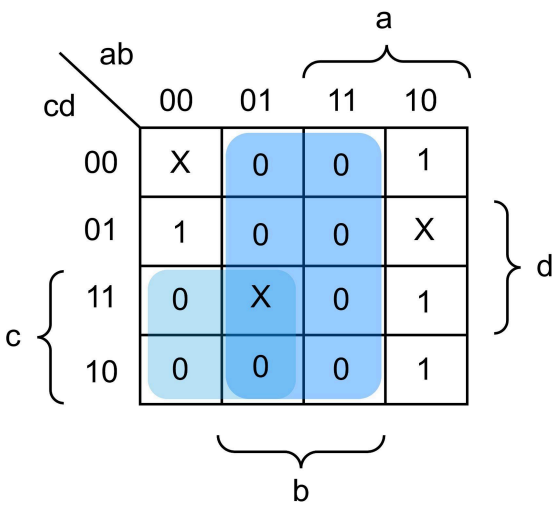
➢ **Question 5 - SOP and POS:**

1. **SOP:**



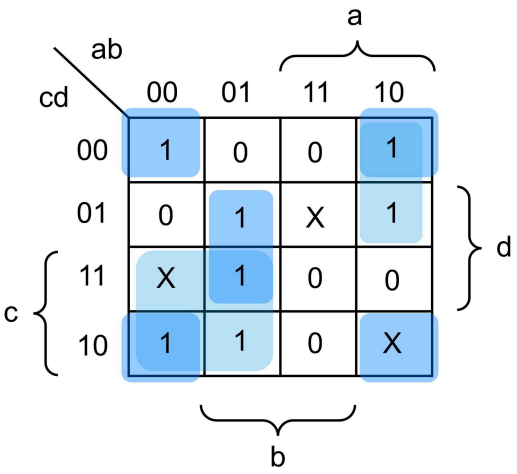
F = ab' + b'c' **...(SOP)**

POS:



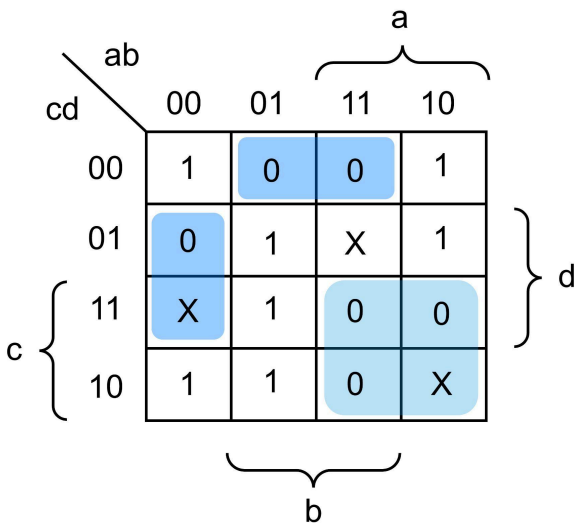
F = b'(a + c') **...(POS)**

2. **SOP:**



F = a'c + b'd' + a'bd + ab'c' **...(SOP)**

POS:



$$F = (a' + c')(b' + c + d)(a + b + d') \quad \textbf{...(POS)}$$