Digital Logic Design Assignment 9 - EC2018-31

Priyansh Agrahari

December 21, 2020

1 Question:

A four-variable Boolean function is realized using 4x1 multiplexers as shown in the figure:

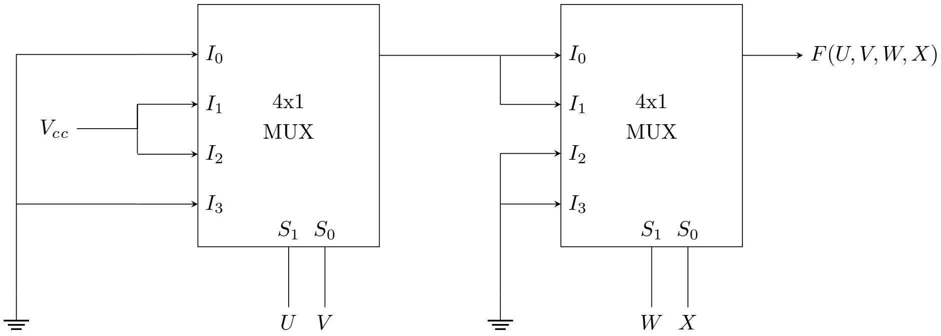


Figure 1: *Question figure*

The minimized expression for F(U,V,W,X) is

(A) $(U\ V + \overline{U}\ \overline{V})\,\overline{W}$

(B) $(U\ V + \overline{U}\ \overline{V})\,(\overline{W}\ \overline{X} + \overline{W}\ \overline{X})$

(C) $(U\ \overline{V} + \overline{U}\ V)\,\overline{W}$

(D) $(U\ \overline{V} + \overline{U}\ V)\,(\overline{W}\ \overline{X} + \overline{W}\ X)$

2 Solution:

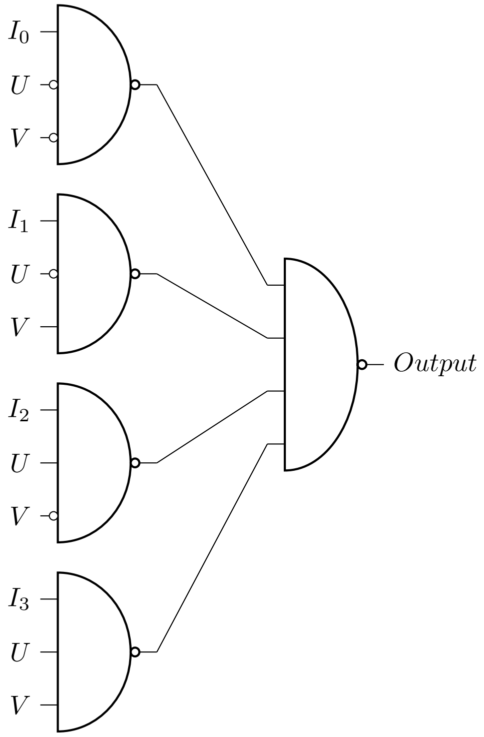


Figure 2: *Logic circuit equivalent of* 4x1 *MUX used to solve this problem*

Since we have I_0 and I_3 grounded, we can take their boolean equivalents to be 0. Then, we get the following equation:

$$Output = \overline{(U + \overline{V})(\overline{U} + V)} \tag{1}$$

which can be further simplified (using de Morgan's law) to obtain:

$$Output = U\overline{V} + \overline{U}V \tag{2}$$

Moving further, the same logic can be used to obtain the result of the second MUX. Since in this case, I_2 and I_3 are grounded; hence by taking their boolean equivalents to be 0, we get the following equation from the second MUX:

$$F = \overline{\overline{(Output.\overline{W}.\overline{X})}.\overline{(Output.\overline{W}.X)}} \tag{3}$$

simplifying, we get:

$$F = Output.\overline{W}.\overline{X} + Output.\overline{W}.X \tag{4}$$

after placing the value of *Output* from eq.(2), and performing a few more manipulations, we get:

$$F = (U\overline{V} + \overline{U}V)\overline{W}(X + \overline{X}) \tag{5}$$

Since X+$\overline{X}$=1, we finally get the desired equation:

$$F = (U\overline{V} + \overline{U}V)\overline{W} \tag{6}$$

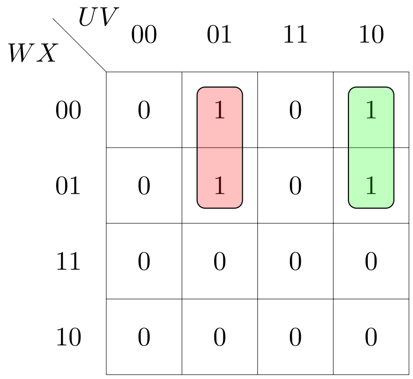
Hence, the answer the given question is (C).

3 Truth Table

$\boldsymbol{U}$	$\boldsymbol{V}$	$\boldsymbol{W}$	$\boldsymbol{X}$	$\boldsymbol{F}$	**Term**
0	0	0	0	0	-
0	0	0	1	0	-
0	0	1	0	0	-
0	0	1	1	0	-
0	1	0	0	1	$\overline{U}\ V\overline{W}\ \overline{X}$
0	1	0	1	1	$\overline{U}\ V\overline{W}\ X$
0	1	1	0	0	-
0	1	1	1	0	-
1	0	0	0	1	$U\overline{V}\ \overline{W}\ \overline{X}$
1	0	0	1	1	$U\overline{V}\ \overline{W}\ X$
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	0	-

Table 1: Truth Table for eq.(6)

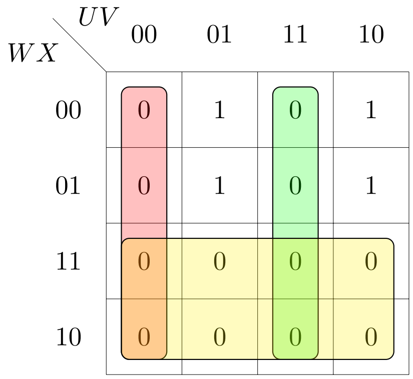
4 K-map for the function F(U,V,W,X)



WX \ UV	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

Figure 3: K-map for SOP expression

The expression obtained using the K-map is the same as the one obtained earlier in eq.(6). Alternatively, we can also make a K-map for obtaining the POS expression:



WX \ UV	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

Figure 4: K-map for POS expression

The POS expression hence obtained is:

$$F = (U + V)(\overline{U} + \overline{V})\overline{W} \tag{7}$$

5 C implementation to verify Table 1

5.1 C code:

```c
#include <stdio.h>
int main(void)
{
    unsigned char U = 0x00, V = 0x01, W = 0x00, X = 0x00;
    unsigned char F, one = 0x01;

    F = (~W) & ( (U & (~V)) | ((~U) & V) );
    printf("Using SOP form: F = %x\n", (F & one));

    F = (U | V) & ((~U) | (~V)) & (~W);
    printf("Using POS form: F = %x\n", (F & one));
}
```

5.2 Output:

```
Using SOP form: F = 1
Using POS form: F = 1
```