

# IDE Assignment

Mohammed Riyazuddin  
FWC220110

**Abstract**—This document contains the procedure to solve the given problem using Karnaugh Map and verify the result using Arduino.

**Q.** To find the prime implicants of function whose sum of products is given as  $f(X,Y,Z) = \sum(2,3,4,5)$ .

**Solution:** Sum of Products: Sum of Products is a form of expression in Boolean algebra in which different product terms of inputs are being summed together. This product is Boolean logical AND and the Sum is Boolean logical OR. The SOP expression usually takes the form of two or more variables ANDed together. Each product term is known as minterm or implicants. The SOP expression is used most often because it tends itself nicely to the development of the truth table and timing diagrams. SOP circuits can also be constructed easily by using a special combinational logic gates called the "AND-OR-INVERTER" gate.

**Prime Implicants:** Prime implicant is a minterm, which are obtained by combining maximum possible adjacent cells in K-map.

**Karnaugh Map:** The "Karnaugh map" is a graphical method which provides a systematic method for simplifying and manipulating the Boolean expression or to convert a truth table to its corresponding logic circuit in a simple, orderly process. In this technique, the information contained in a truth table or available in SOP or POS form is represented on K-map. The given problem can be solved K-Map.

**Complete Simplification Rules:**

1. Construct the K-map and places 1's 2,3,4 and 5 cells. Place 0's in the other cells.
2. Examine the map for adjacent 1's and loop those 1's which are not adjacent to any other 1's. These are called isolated 1's. Here we find no such 1's.
3. Next, look for those 1's which are adjacent to only one other 1. Loop any pair containing such a 1. Here we have two such pairs.
4. Loop any octet even it contains some 1's that have already been looped. Here we find no such octet's.
5. Loop any quad that contains one or more 1's which have not already been looped, making sure to use the minimum number of loops. Here we find no such quad's.
6. Loop any pairs necessary to include any 1's that have not been looped, making sure to use the minimum number of loops. Here we find no such loops.
7. Form the OR sum of all the terms generating by each loop.

## I. K-MAP

Using the boolean logic, output F can be expressed in terms of the inputs X and Y with the help of the following Kmap.

		YZ			
		00	01	11	10
X	0	0	0	1	1
	1	1	1	0	0

The boolean expression for the output F, is obtained in the form of SOP after minimizing the Kmap mixterm implicants. Here in above K-map we notice that there are two pairs. Let us name them as P1 and P2.

$$P1 = X'YZ + X'YZ'$$

$$P2 = XY'Z' + XY'Z$$

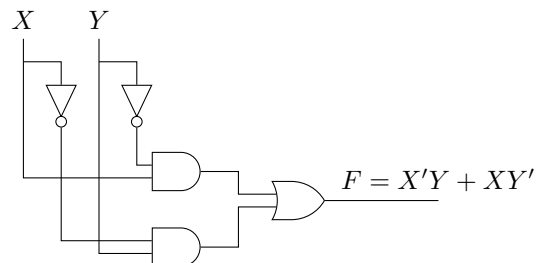
If we minimize the above equations we get

$$P1 = X'Y$$

$$P2 = XY'$$

$$\text{Since } F = P1 + P2$$

$$F = X'Y + XY'$$



Internal diagram of EX-OR gate.

## II. TRUTH TABLE

The following is the truth table of EX-OR gate

X	Y	$F = X'Y + XY'$
0	0	0
0	1	1
1	0	1
1	1	0

TABLE I

## III. COMPONENTS

Component	Value	Quantity
Arduino	UNO	1
Resistor	220ohm	1
Bread board	-	1
Jumper wires	M-M	20
LED	-	1

#### IV. HARDWARE CONNECTION

The inputs 'X' and 'Y' are read from the breadboard which are connected to pins '2','3' in Port D of Arduino and the output is take from the pin '5' of Arduino. The output is connected to LED. The respective connections are shown in the Table II.

Arduino	2	3	5
Inputs	X	Y	-
Output	-	-	F

TABLE II

#### V. PROCEDURE

1. Give the connections as per Table 2. For taking the inputs connect 5V of arduino to +ve line of bread board to consider it as logic 'HIGH',connect GND pin of arduino to -ve line of bread board to consider it as logic 'LOW'.
2. Since we know the output of EX-OR gate is '1', if and only if only one of the inputs is '1' and '0' in rest conditions. The same condition can be verified changing the inputs in breadboard.
3. Download the following code  
<https://github.com/riyaziith/FWC-Module1>
4. Upload the code into the Arduino board using USB cable.
5. The output '1' is represented as the state: 'LED ON' and '0' is represented as the state 'LED OFF'