North South University

Department of Electrical and Computer Engineering

School of Engineering and Physical Sciences

Project

Semester: Spring 2020

Topic: Implementation of a given encrypted table using Arduino

Submitted By

Group-5

Md. Ruhul Amin

ID: 1512487042

Md Yeasir Arafat

ID: 1631007042

Course Title: Microprocessor Interfacing and Embedded System

Course Code: CSE331

Instructor: Dr. Dihan Md. Nuruddin Hasan (dmh)

Section: 06

Submission Date: 20 April,2020

Table of Contents

Title: 3

Description: 3

Tools Used: 3

- **Microcontroller: (Arduino UNO)** 3
- **LED light** 4
- **Breadboard, Wires** 4

K-Map Theory: 5

Steps to solve expression using K-map- 5

Methods of Derivation and Derived result: 7

Circuit Diagram & Electrical Components: 11

Components: 11

Circuit Diagram (Porteous): 11

K-Map Circuit Diagram: 12

Circuit Operation Principle: 13

Input Output Table (Truth Table): 14

Flow Chart: 15

Arduino Program: 16

Title:

Implementation of a given encrypted table using Arduino.

Abstract:

We are going to implement the given encryption table using microcontroller. We will 4 input DIP switches to configure the inputs for high and low conditions. And we are using LEDs to represent the corresponding output statues.

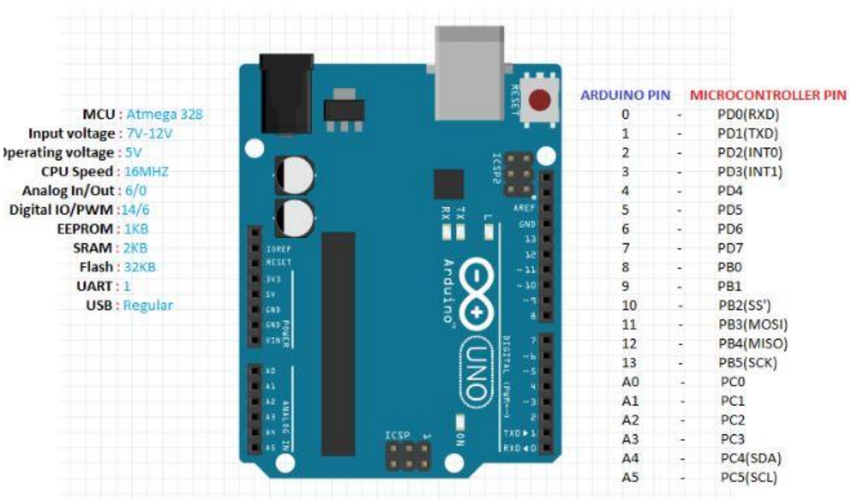
Description:

This project is based on a table input and output. Here it takes input from hardware components (Button) and shows corresponding out through LEDs. It takes four input from button and shows four output. This input encrypted by k-map program. An Arduino program which is using to encrypt that inputs. Porteous used to implement the circuit.

Tools Used:

- **Microcontroller: (Arduino UNO)**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.



1.Arduino Uno R3

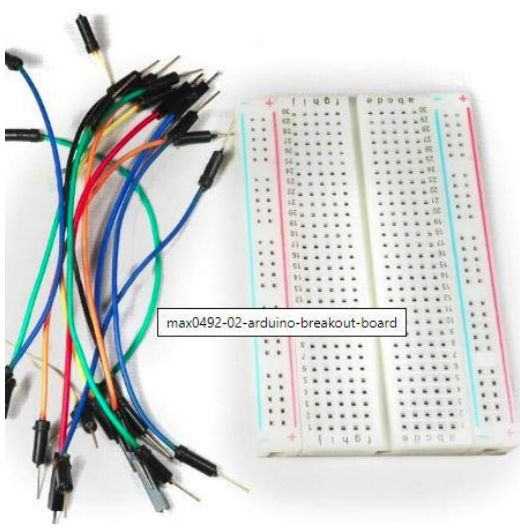
➢ LED light

LED stands for "light emitting diode." A diode is an electrical component with two terminals which conduct the electricity only in one direction. With an electrical current, the diode emits a bright light around the small bulb.

2. Led

➢ Breadboard, Wires

Breadboards are designed to work with through-hole electronic components. These components have long metal leads that are designed to be inserted through holes in a printed circuit board (PCB) that are plated with a thin copper coating, which allows the components' leads to be soldered to the board.



3.Breadboard and Wires

K-Map Theory:

Maurice Karnaugh, a broadcast communications engineer, built up the Karnaugh map at Bell Labs in 1953 while structuring advanced rationale-based phone exchanging circuits. A Karnaugh map (K-map) is a pictorial technique used to limit Boolean expressions without utilizing Boolean variable-based math hypotheses and condition controls. A K-map can be thought of as an exceptional version of a truth table. Utilizing a K-map, expressions with two to four variables are effectively limited. Expressions with five to six variables are progressively troublesome however attainable, and expressions with at least seven variables are amazingly troublesome (if not difficult) to limit utilizing a K-map. We can minimize Boolean expressions of 3, 4 variables very easily using K-map without using any Boolean algebra theorems. K-map can take two forms Sum of Product (SOP) and Product of Sum (POS) according to the need of problem. K-map is table like representation but it gives more information than TRUTH TABLE. We fill grid of K-map with 0's and 1's then solves it by making groups.

Steps to solve expression using K-map-

1. Select K-map according to the number of variables.
2. Identify minterms or maxterms as given in problem.
3. For SOP put 1's in blocks of K-map respective to the minterms (0's elsewhere).
4. For POS put 0's in blocks of K-map respective to the maxterms(1's elsewhere).
5. Make rectangular groups containing total terms in power of two like 2,4,8 ..(except 1) and try to cover as many elements as you can in one group.
6. From the groups made in step 5 find the product terms and sum them up for SOP form.

The diagram below illustrates the correspondence between the Karnaugh map and the truth table for the general case of a two-variable problem.

A	B	F
0	0	a
0	1	b
1	0	c
1	1	d

Truth Table.

B \ A	0	1
0	a	b
1	c	d

F.

The values inside the squares are copied from the output column of the truth table, therefore there is one square in the map for every row in the truth table. Around the edge of the Karnaugh map are the values of the two input variable. A is along the top and B is down the left hand side. The diagram below explains this:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table.

B \ A	0	1
0	0	1
1	1	1

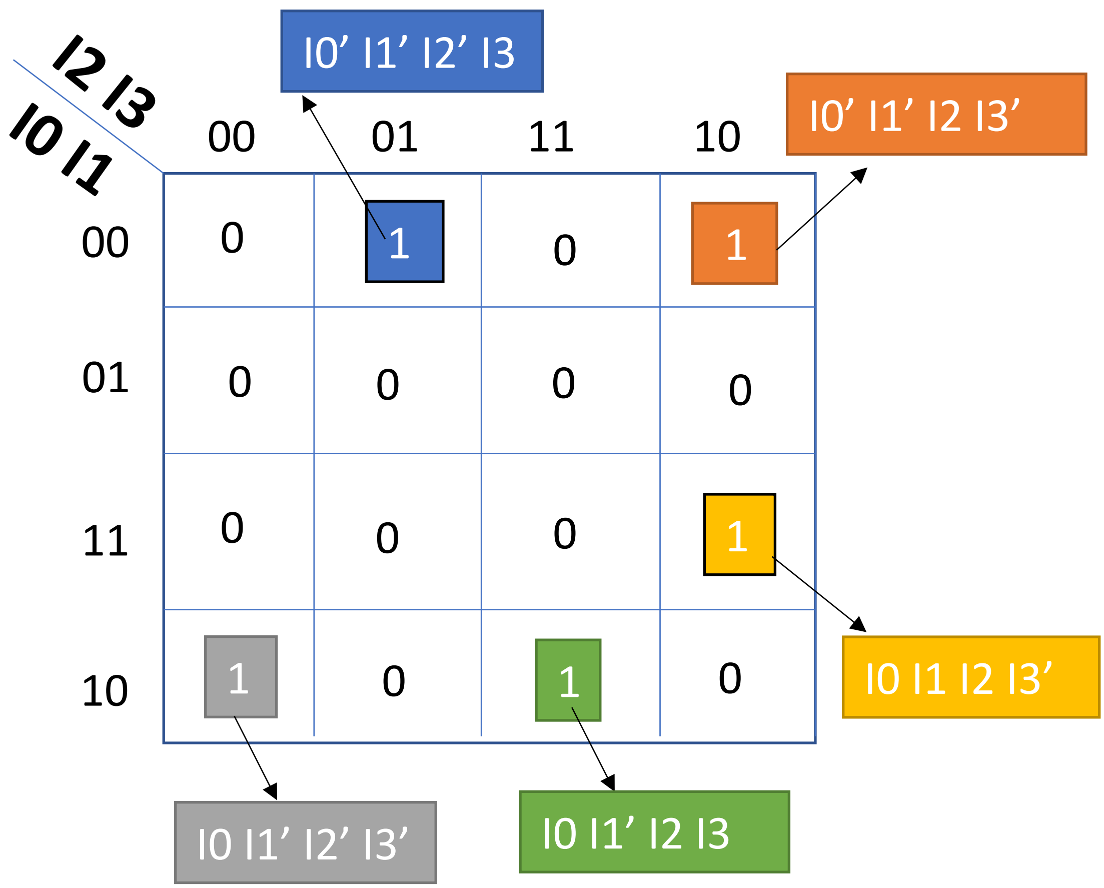
F.

The values around the edge of the map can be thought of as coordinates. So as an example, the square on the top right hand corner of the map in the above diagram has coordinates A=1 and B=0. This square corresponds to the row in the truth table where A=1 and B=0 and F=1. Note that the value in the F column represents a particular function to which the Karnaugh map corresponds.

Methods of Derivation and Derived result:

We used K-Map to encrypt out output. For 16 input we have four output stats. For each output condition we derived equation through k-map. Here we are showing the derivation process and derived result.

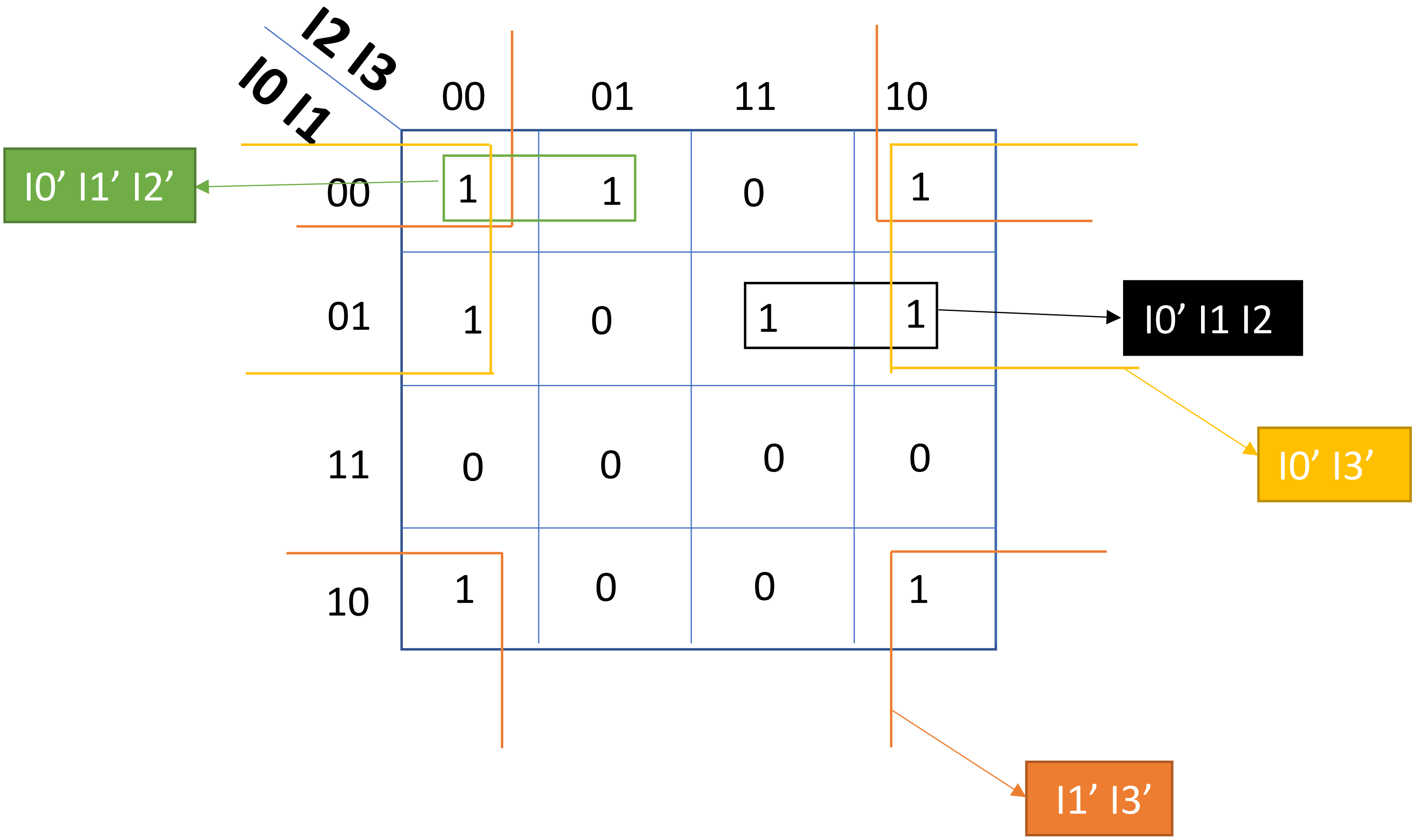
FOR O3



F= (IO+I1+I2+I3’) (I0+I1+I2’+I3) (I0’+I1’+I2’+I3) (I0’+I1+I2’+I3’) (I0’+I1+I2+I3)

F= IO’ I1’ I2’ I3 + I0’ I1’ I2 I3’ + I0 I1 I2 I3’ + I0 I1’ I2’ I3’ + I0 I1’ I2 I3

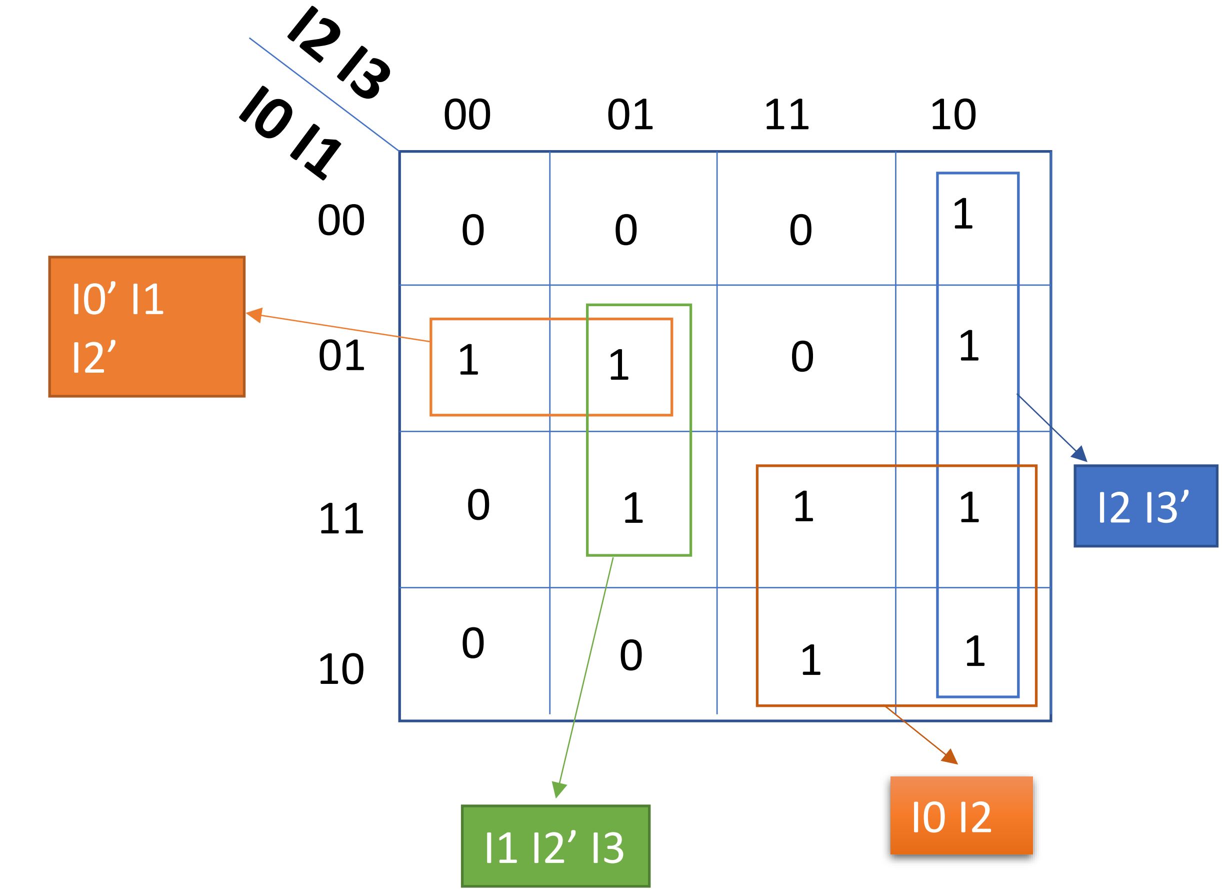
FOR 02



$$F = (I0+I1+I2)\,(I0+I1'+I2')\,(I0+I3)\,(I1+I3)$$

$$F = I0'\,I1'\,I2' + I0'\,I1\,I2 + I0'\,I3' + I1'\,I3'$$

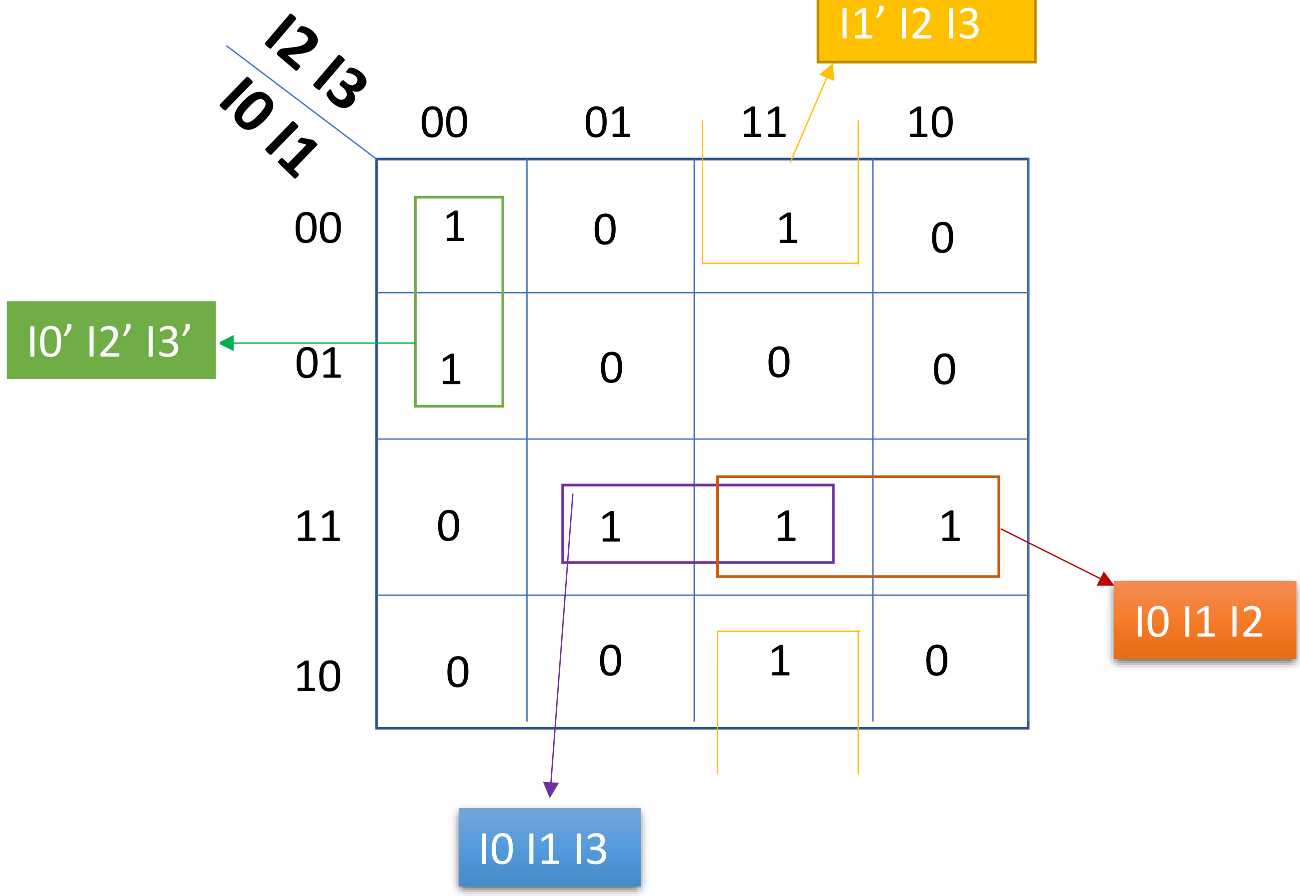
FOR 01



F= (I0+I1'+I2') (I1+I2'+I3) (I0'+I2') (I2'+I3)

F= I0' I1 I2' +I1 I2' I3 + I0 I2 + I2 I3'

FOR 00



F= (I0+I2+I3) (I0’+I1’+I3’) (I1+I2’+I3’) (I0’+I1’+I2’)

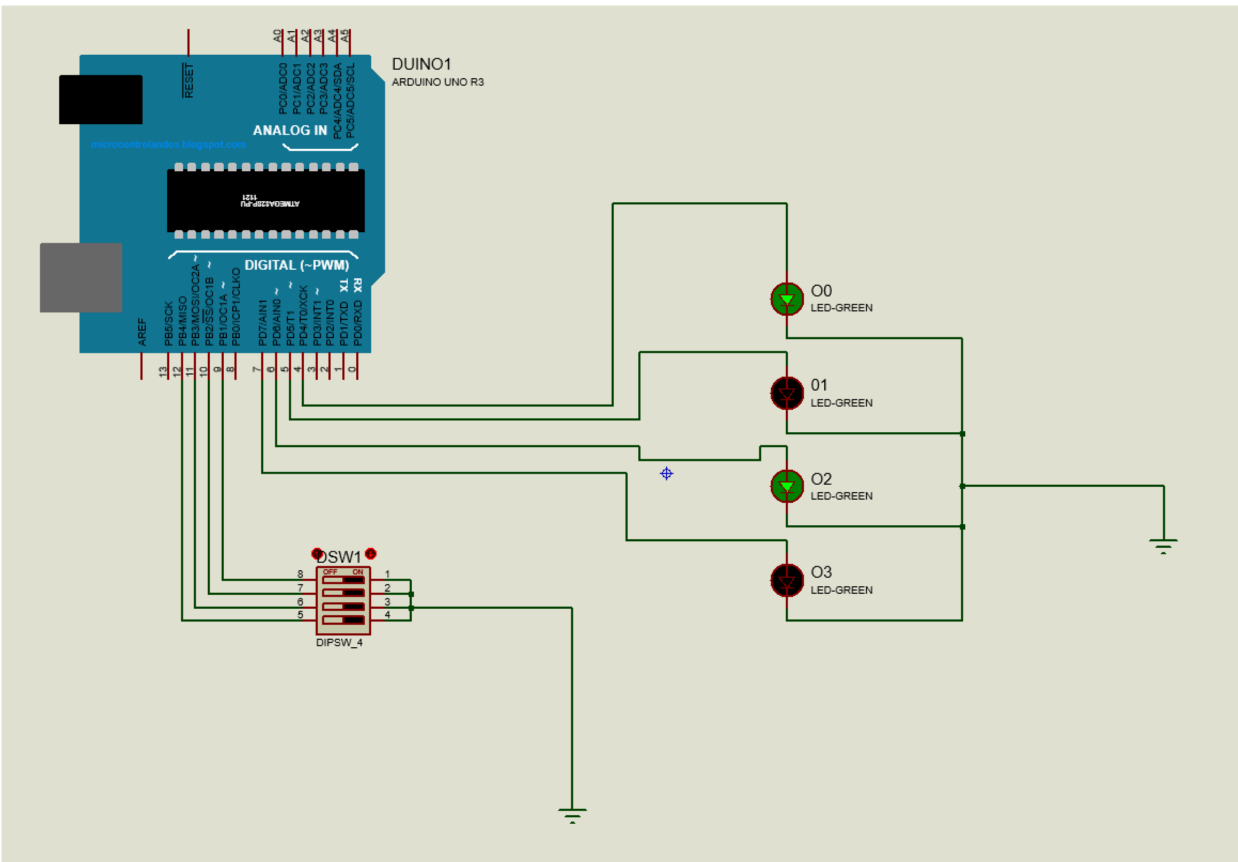
F= I0’ I2’ I3’ + I0 I1 I3 + I1’ I2 I3 + I0 I1 I2

Circuit Diagram & Electrical Components:

Components:

1. Arduino UNO R3
2. 4 input DIP Switch (DIPSW_4)
3. LED green
4. Connecting Wires

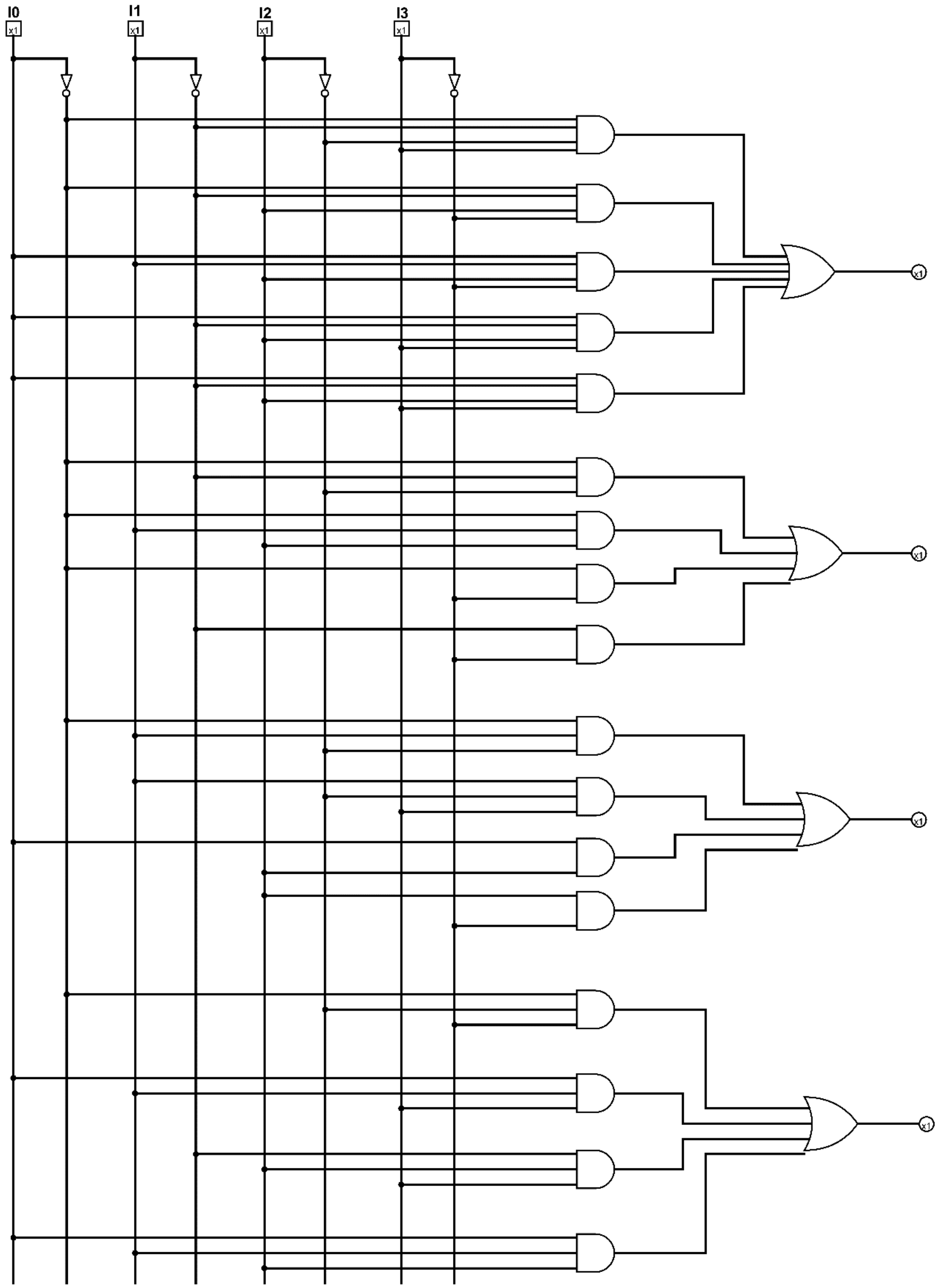
Circuit Diagram (Porteous):



4.Circuit Diagram

We also implemented circuit diagram for K-Map using Logisim. This shows the K-Map Input and output behaviors.

K-Map Circuit Diagram:



5. K-Map Circuit Diagram

Circuit Operation Principle:

As we implemented our circuit in proteus we have to run it from proteus. An Arduino program has loaded into the Arduino in proteus. In order to check or use that we will run our circuit. Then the program will execute into the Arduino and it will start giving output for corresponding input. The input serial is from bottom to upward in DIP switch. The output is also from bottom to upward.

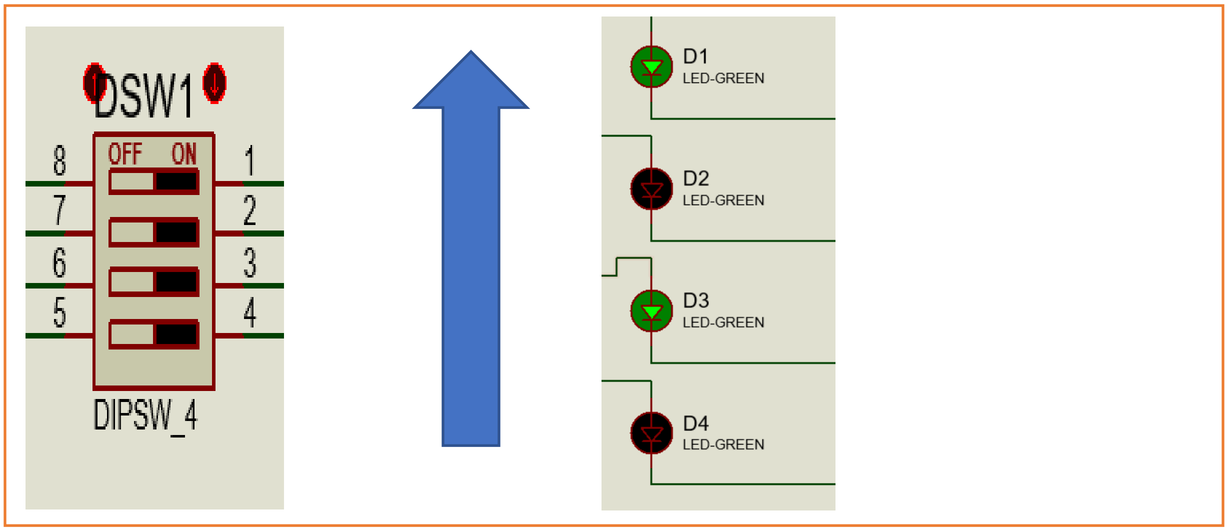
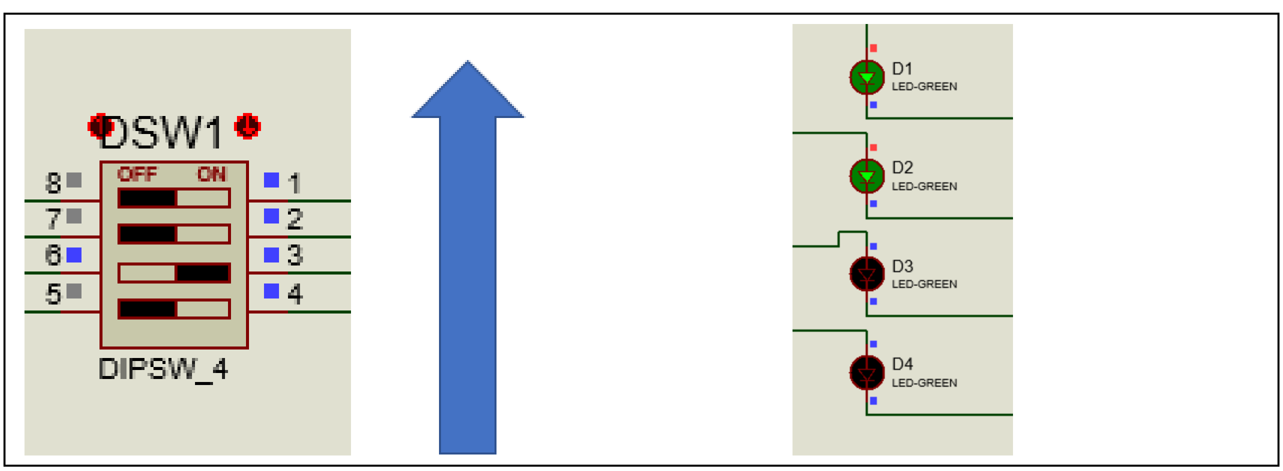


Figure 6Circuit Principle

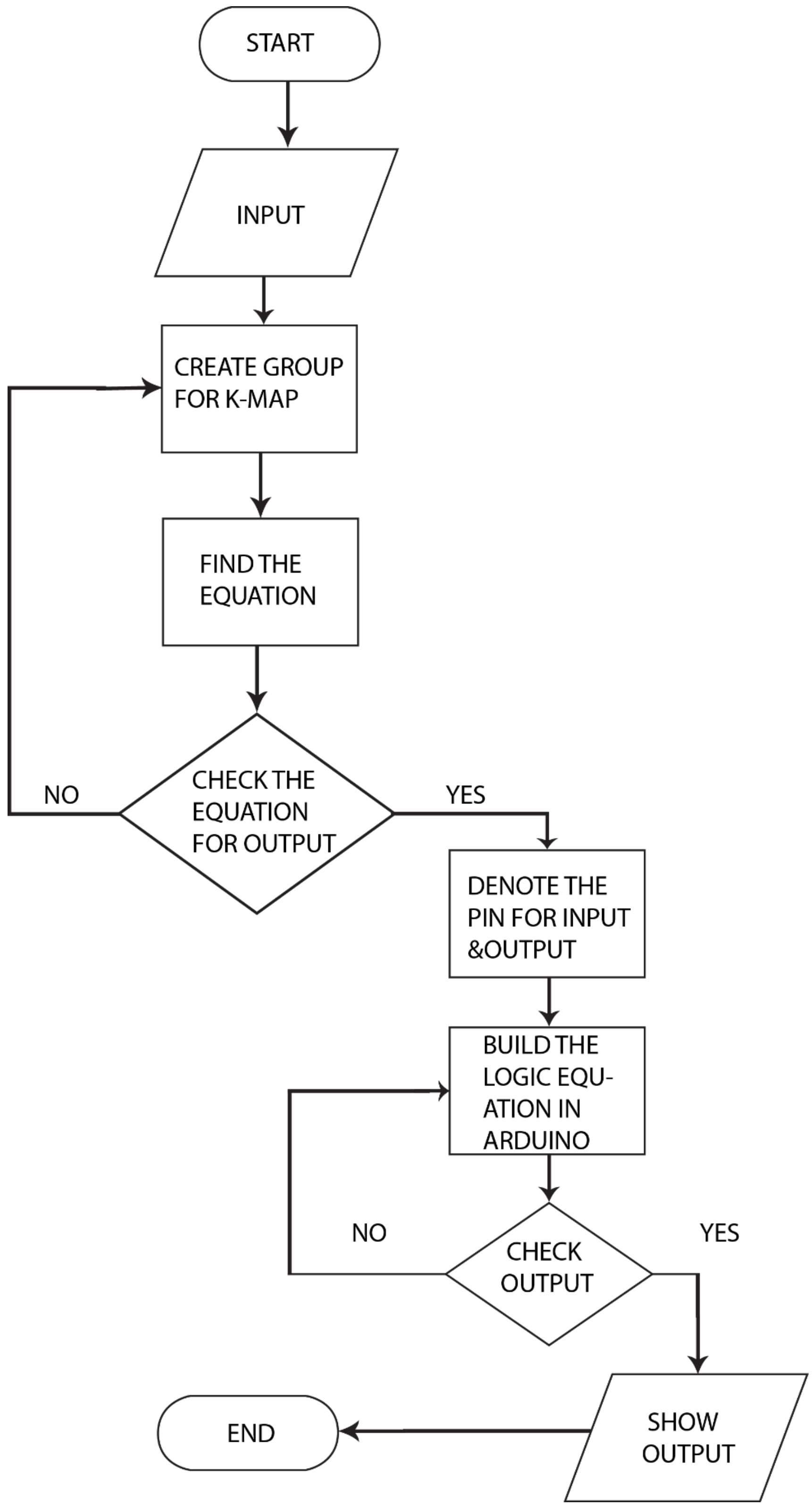
If we want to input 1011 then we have to turn on the switch from bottom. So, number 5,7,8 will be turned on and 6 will remain off. It will give us the input 1011. For output we will consider the led from bottom to top. So, the output is 0011.



Input Output Table (Truth Table):

Input				**Output**			
I3	I2	I1	I0	O3	O2	O2	O1
0	0	0	0	0	1	0	1
1	0	0	0	1	1	0	0
0	1	0	0	1	1	1	0
1	1	0	0	0	0	0	1
0	0	1	0	0	1	1	1
1	0	1	0	0	0	1	0
0	1	1	0	0	1	1	0
1	1	1	0	0	1	0	0
0	0	0	1	1	1	0	0
1	0	0	1	0	0	0	0
0	1	0	1	0	1	1	0
1	1	0	1	1	0	1	1
0	0	1	1	0	0	0	0
1	0	1	1	0	0	1	1
0	1	1	1	1	0	1	1
1	1	1	1	0	0	1	1

Flow Chart:


START
INPUT
CREATE GROUP FOR K-MAP
FIND THE EQUATION
CHECK THE EQUATION FOR OUTPUT
NO
YES
DENOTE THE PIN FOR INPUT &OUTPUT
BUILD THE LOGIC EQU-ATION IN ARDUINO
CHECK OUTPUT
NO
YES
SHOW OUTPUT
END

Arduino Program:

```
/*
 * Note:
 * Dots (.) denote AND operation.
 * Single quote (') denote low state.
 *
 * K-Map results:
 * OUT3 = IO’.I1’.I2’.I3+I0’.I1’.I2.I3’+I0.I1.I2.I3’+I0.I1’.I2’.I3’+I0.I1’.I2.I3
 * OUT2 = I0’.I1’.I2’+I0’.I1.I2+I0’.I3’+I1’.I3’
 * OUT1 = I0’.I1.I2’+I1.I2’.I3+I0.I2+I2.I3’
 * OUT0 = I0’.I2’.I3’+I0.I1.I3+I1’.I2.I3+I0.I1.I2
 */

/*
 * Constants defined so the variables defined for
 * the purpose of controlling the input and output
 * pins are not changed.
 */
// Built-in LED
const int builtInLED = 13;
// Digital pins for input
const int IN3=12; // pin 12 for I3
const int IN2=11; // pin 11 for I2
const int IN1=10; // pin 10 for I1
const int IN0=9;  // pin 9 for I0
```

```
// Digital pins for output
const int OUT3=7;  // pin 7 for O3
const int OUT2=6;  // pin 6 for O2
const int OUT1=5;  // pin 5 for O1
const int OUT0=4;  // pin 4 for O0

void setup() {
  // turn off built-in LED
  pinMode(builtInLED, OUTPUT);
  digitalWrite(13,LOW);

  // defining digital pins 9 to 12 as input
  pinMode(IN3, INPUT);
  pinMode(IN2, INPUT);
  pinMode(IN1, INPUT);
  pinMode(IN0, INPUT);

  //defining digital pins 4 to 7 as output
  pinMode(OUT3, OUTPUT);
  pinMode(OUT2, OUTPUT);
  pinMode(OUT1, OUTPUT);
  pinMode(OUT0, OUTPUT);

}
```

```
void loop() {
  // Reading digital input from input pins
  bool input3 = digitalRead(IN3);
  bool input2 = digitalRead(IN2);
  bool input1 = digitalRead(IN1);
  bool input0 = digitalRead(IN0);

  // OUT3 status
  if((input0==LOW&&input1==LOW&&input2==LOW&&input3==HIGH)|| //
breaking long conditional statement into multiple lines
  (input0==LOW&&input1==LOW&&input2==HIGH&&input3==LOW)||
  (input0==HIGH&&input1==HIGH&&input2==HIGH&&input3==LOW)||
  (input0==HIGH&&input1==LOW&&input2==LOW&&input3==LOW)||
  (input0==HIGH&&input1==LOW&&input2==HIGH&&input3==HIGH)){
    digitalWrite(OUT3, HIGH); // OUT3 set to high
  }else{
    digitalWrite(OUT3, LOW); // OUT3 set to low
  }

  // OUT2 status
  if((input0==LOW&&input1==LOW&&input2==LOW)|| // breaking long
conditional statement into multiple lines
  (input0==LOW&&input1==HIGH&&input2==HIGH)||
  (input0==LOW&&input3==LOW)||
  (input1==LOW&&input3==LOW)){
```

```
    digitalWrite(OUT2, HIGH); // OUT2 set to high
  }else{
    digitalWrite(OUT2, LOW); // OUT2 set to low
  }

  // OUT1 status
  if((input0==LOW&&input1==HIGH&&input2==LOW)||      // breaking long conditional statement into multiple lines
  (input1==HIGH&&input2==LOW&&input3==HIGH)||
  (input0==HIGH&&input2==HIGH)||
  (input2==HIGH&&input3==LOW)){
    digitalWrite(OUT1, HIGH); // OUT1 set to high
  }else{
    digitalWrite(OUT1, LOW); // OUT1 set to low
  }

  // OUT0 status
  if((input0==LOW&&input2==LOW&&input3==LOW)||       // breaking long conditional statement into multiple lines
  (input0==HIGH&&input1==HIGH&&input3==HIGH)||
  (input1==LOW&&input2==HIGH&&input3==HIGH)||
  (input0==HIGH&&input1==HIGH&&input2==HIGH)){
    digitalWrite(OUT0, HIGH); // OUT0 set to high
  }else{
    digitalWrite(OUT0, LOW); // OUT0 set to low
  }
```

```
  // adding delay to idle the arduino
  // and also for stability to allow the LEDs to discharge
  delay(50);

}
```

In this section we have attached GitHub links where all of codes, Circuit Diagrams Arduino programs and other files linked with this project can be found.

https://github.com/ruhulamin005/Encryption-with-k-map-using-arduino