**Table of Contents**

**1. Introduction**

1.1Arduino

1.2 Common Components of Arduino Boards

1.3 Proteus 8 pro

1.4 Logisim

1.5 Equipment

* 1. **Component Description** 2.1 Arduino UNO

2.2 Breadboard

2.3 LED light

2.4 External Power source

2.5 Arduino Software

1. **Truth Table**
2. **Experimental Planning**
3. **Experimental Circuit Setup**
4. **Circuit Diagram**
5. **Arduino code**
6. **Observation**

8.1 Circuit Diagram

8.2 Circuit Operation

8.3 Program Flowchart

1. **Question Answer**
2. **Discussion**

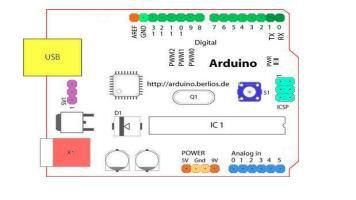
**1 .Introduction**

**1.1 Arduino**

Arduino comprises of both a physical programmable circuit board (commonly known as a microcontroller) and a programming software, or IDE (Integrated Development Environment) that can be run on a PC, used to compose and transfer PC code to the circuit board. It can be done by using the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Unlike other programmable circuit boards, the Arduino does not require a different equipment (called a software engineer) to upload code to the circuit board, one can essentially utilize a USB link.Also, the Arduino IDE utilizes a rearranged rendition of C, making it simpler to figure out how to program.

**1.2 Common Components on Arduino Boards**

There are different types of Arduino boards for different purposes. But all the boards have the majority of following components in common.



Starting clockwise from the top center:

* Analog Reference pin (orange)
* Digital Ground (light green)
* Digital Pins 2-13 (green)

* Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital I/O
* (digitalRead() and digitalWrite()). If serial communication is also being used (e.g. Serial.begin).
* Reset Button - S1 (dark blue)
* In-circuit Serial Programmer (blue-green)
* Analog In Pins 0-5 (light blue)
* Power and Ground Pins (power: orange, grounds: light orange)
* External Power Supply In (9-12VDC) - X1 (pink)
* Toggles External Power and USB Power (place jumper on two pins closest to desired supply) -SV1 (purple)
* USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

Digital Pins

The digital pins on an Arduino board can be used for general purpose input and output via the pinMode(), digitalRead(), and digitalWrite() commands. Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input.

* Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
* External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low
* value, a rising or falling edge, or a change in value.
* PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On
* boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.[1]
* Analog Pins

The analog input pins support 10-bit analog-to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19.

Power Pins

* 9V: The input voltage to the Arduino board when it’s using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). Different boards accept different input voltages ranges.

* 5V: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
* 3V: (Decimal-only) A 3.3-volt supply generated by the on-board FTDI chip.
* GND: Ground pins

**1.3 Proteus 8 Pro**

Proteus 8 Professional. Proteus 8 Professional is a software which can be used to draw schematics, PCB layout, code and even simulate the schematic. It is developed by Labcenter Electronic Ltd. We used Proteus 8 first to build our Arduino project. We installed Arduino library first. Then we build a project containing Arduino, LED, Buttons and Resistors. But first we derived out HEX file from Arduino IDE code. We burned the HEX file into Proteus 8 Arduino simulation and thus finishing our project simulation.

**1.4 Logisim**

Logisim is a logic simulator which permits circuits to be designed and simulated using a graphical user interface. Released under the GNU Public License, Logisim is free software designed to run under the Microsoft Windows, OS X, and Linux platforms. Its code is entirely in Java using the Swing graphical user interface library. The primary developer, Carl Burch, has worked on Logisim since its inception in 2001. We build our circuit diagram using Logisim simulation software and thus given below.

**1.5 Equipment**

We used One Arduino UNO, One Battery, One Breadboard, Four LEDs, 4x10k ohm Resisitors, Four Buttons, 18 Wires. For software we used Logisim, Proteus 8 Pro and Arduino IDE.

**2. Components Description**

**2.1 Arduino UNO**

The Uno is one of the most popular Arduino boards. It consists of 14-digital I/O pins, where 6-pins can be used as PWM (pulse width modulation outputs), 6-analog inputs, a reset button, a power jack, a USB connection and more. It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with an AC-to-DC adapter or battery.

**2.2 Breadboard**

A breadboard is a construction base for prototyping of electronics. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term” breadboard” is commonly used to refer to these. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

**2.3 LED Light**

A LED display is a flat panel display that uses an array of light-emitting diodes as pixels for a video display. Their brightness allows them to be used outdoors where they are visible in the sun for store signs and billboards. In recent years, they have also become commonly used in destination signs on public transport vehicles, as well as variable-message signs on highways. LED displays are capable of providing general illumination in addition to visual display, as when used for stage lighting or other decorative (as opposed to informational) purposes. LED displays can offer higher contrast ratios than a projector and are thus an alternative to traditional projection screens, and they can be used for large, uninterrupted (without a visible grid arising from the bezels of individual displays) video walls. microLED displays are LED displays with smaller LEDs, which poses significant development challenges.

**2.4 External Power Source**

For stand-alone operation, the board is powered by a battery rather than through the USB connection to the computer. While the external power can be anywhere in the range of 6 to 24 V (for example, you could use a car battery), a standard 9 V battery is convenient. While you could jam the leads of a battery snap into the Vin and Gnd connections on the board, it is better to connect the battery snap leads to a DC power plug and connect to the power jack on the board.

**2.5 Arduino Software**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board.

The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.

**3. Truth Table**

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

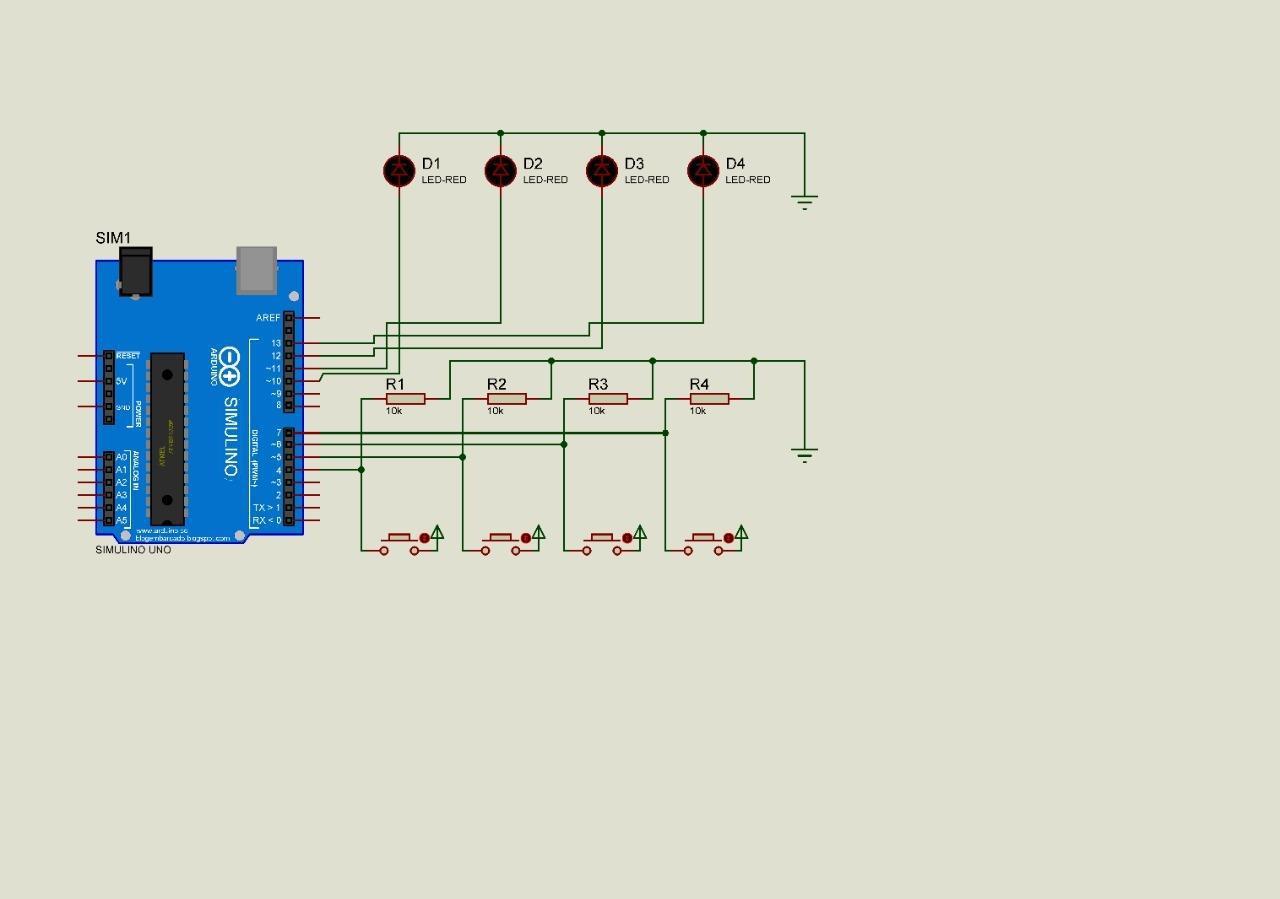


**4. Experimental Planning**

We have to implement an encryption table using microcontroller. Microcontroller is a small computer based on a single metal-oxide-semiconductor, in sort MOS, integrated circuit chip. It contains one or more CPUs with memory and programmable input/outputs. In this project, we are using Arduino Uno.To get the desired output, here we have written a code based on derived logical expressions. To get these expressions, we have derived K-map according to the given input table.

1. **Experimental Circuit Setup**
   * First, power and ground were connected to the breadboard through the Arduino.
   * Then placed the Arduino Board
   * Connected the wires with input and output pins
   * Placed the LEDs on breadboard
   * Connected the power supply to the Arduino

**6. Circuit Diagram**



**7. Arduino Code**

int i4 = 7;

int i3 = 6;

int i2 = 5;

int i1 = 4;

int o4 = 13;

int o3 = 12;

int o2 = 11;

int o1 = 10;

int val4 = 0;

int val3 = 0;

int val2 = 0;

int val1 = 0;

void setup() {

pinMode(o4, OUTPUT);

pinMode(o3, OUTPUT);

pinMode(o2, OUTPUT);

pinMode(o1, OUTPUT);

pinMode(i4, INPUT);

pinMode(i3, INPUT);

pinMode(i2, INPUT);

pinMode(i1, INPUT);

}

void loop() {

val4 =

((!digitalRead(i2)&&digitalRead(i3)&&!digitalRead(i4))||

(digitalRead(i2)&&!digitalRead(i3)&&!digitalRead(i4))||

(!digitalRead(i1)&&digitalRead(i2)&&digitalRead(i4))||

(digitalRead(i1)&&digitalRead(i3)&&digitalRead(i4)));

digitalWrite(o4, val4);

val3 = ((!digitalRead(i1)&&!digitalRead(i2)&&!digitalRead(i3)&&!digitalRead(i4))||

(digitalRead(i3)&&digitalRead(i4))||

(digitalRead(i2)&&digitalRead(i4))||(!digitalRead(i1)&&digitalRead(i2)&&digitalRead(i3)));

digitalWrite(o3, val3);

val2 =

((digitalRead(i2)&&!digitalRead(i4))||(digitalRead(i3)&&!digitalRead(i4))||(digitalRead(i1)&&d

igitalRead(i2))||(!digitalRead(i1)&&!digitalRead(i2)&&digitalRead(i4))||(!digitalRead(i2)&&!di

gitalRead(i3)&&digitalRead(i4)));

digitalWrite(o2, val2);

val1 =

((!digitalRead(i2)&&!digitalRead(i3))||(!digitalRead(i1)&&!digitalRead(i3)&&!digitalRead(i4))|

|(digitalRead(i1)&&!digitalRead(i2)&&!digitalRead(i4))||(digitalRead(i1)&&!digitalRead(i3)&

&digitalRead(i4)));

digitalWrite(o1, val1);

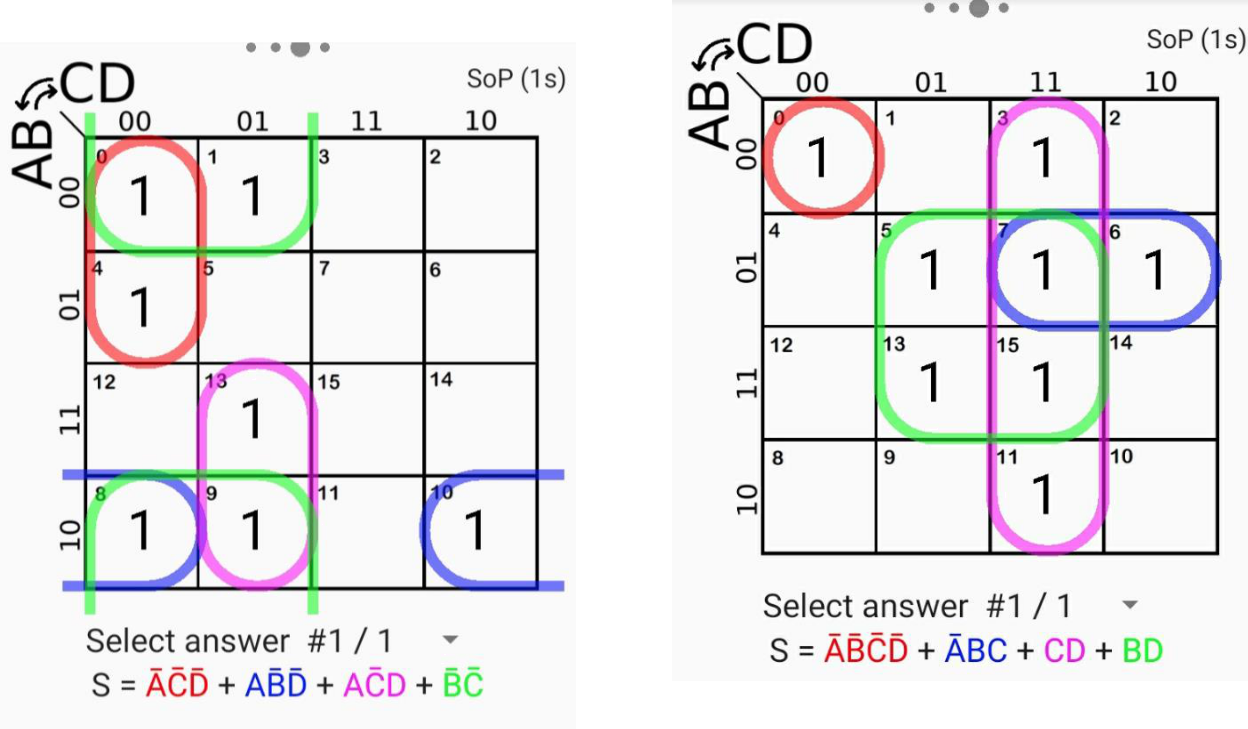
}

**8 Observation**

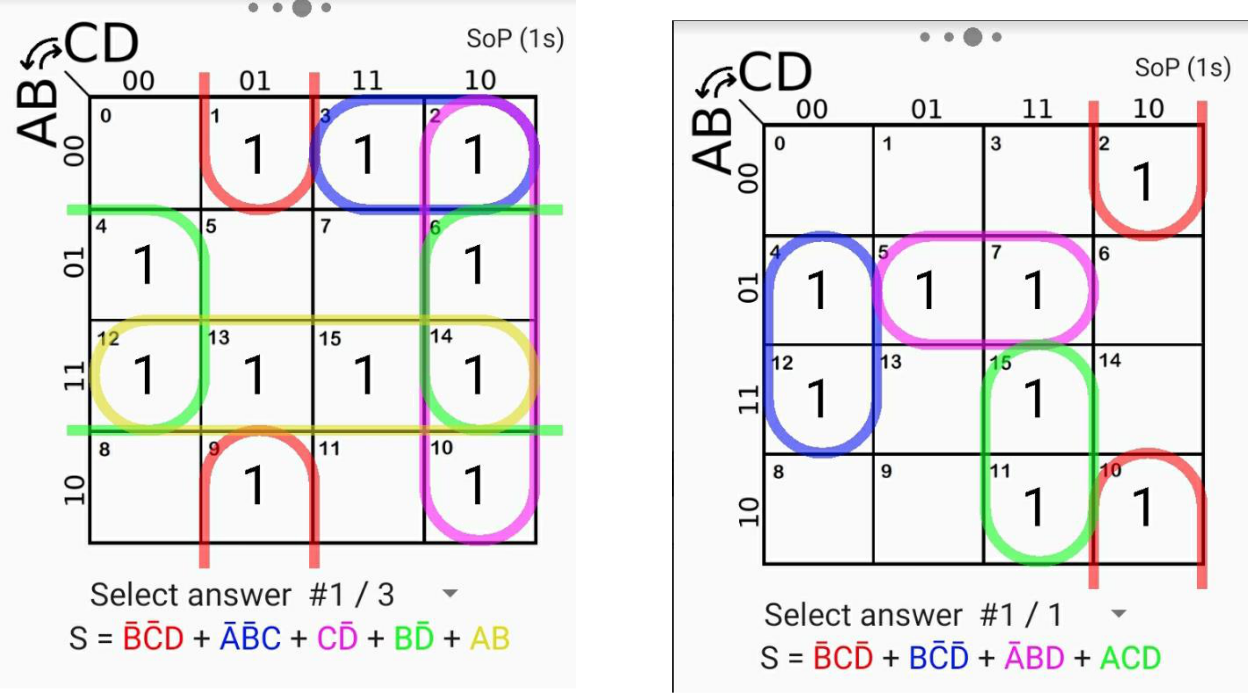
**8.1 Theoretical observation:**

K-Mapping

K-maps are used to simplify the logic requirements and provide a visual way to produce a much simpler formula for expressing the same logic derived from truth table. Here our task is to minimize Boolean expressions of four variables from the given encryption table.



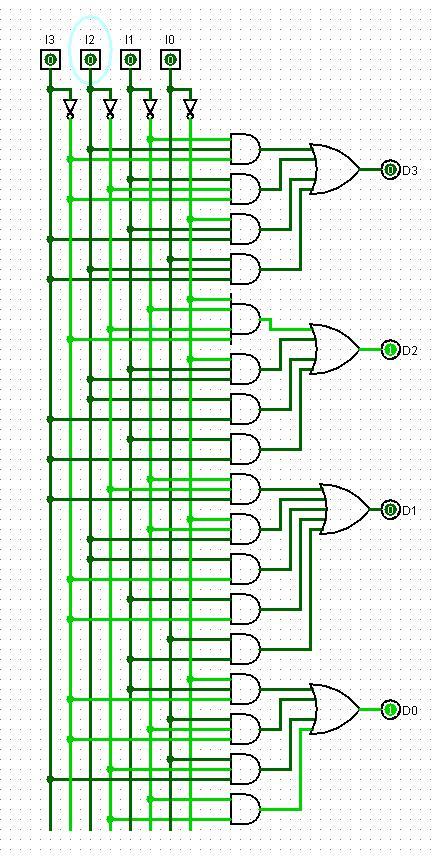
| Output- 1 | Output- 3 |  |
| --- | --- | --- |
|  |  |



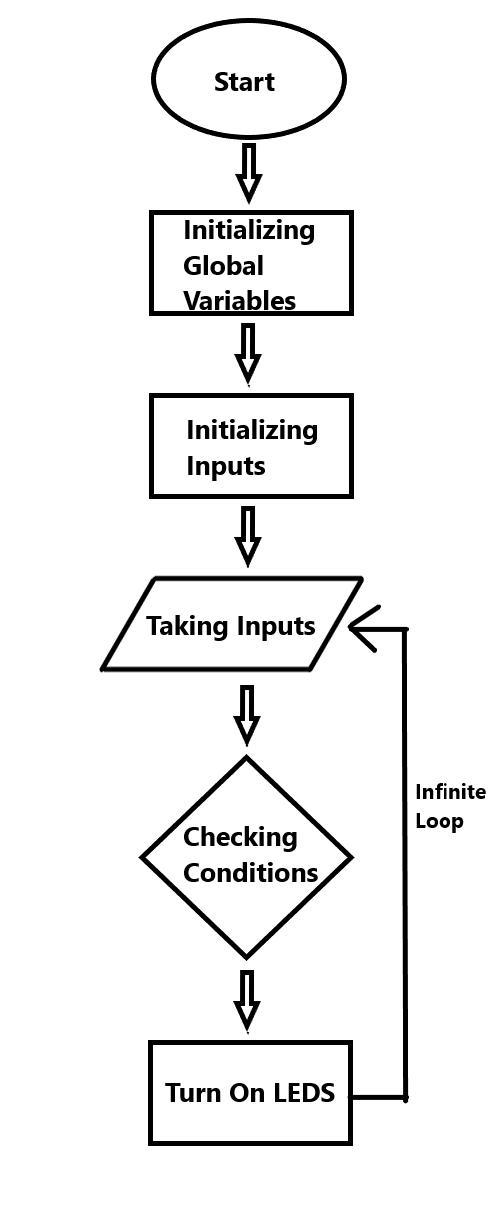
Output- 2 Output-4

**8.2 Circuit operation:**

We developed a program on Arduino IDE and imported the hex file to Proteus for simulation. So, when we set the switches to ground the value sets to 0 for that particular pin. Similarly, we set the switch to ON mode, input value for the particular pin sets to 1. Arduino IC receives the input values, analyzes and sets the output values for each output pin. So correspondent LED reflects the results.



**8.3 Program Flowchart:**



**9. Question-Answer:**

**1. What is the clock frequency of the microcontroller used?**

Answer: The clock frequency is 16 MHz.

**2. What is the data bus width of the microcontroller used?**

Answer: Data bus width is 6 bit

**3. What is the size of your hex file generated? Attach the hex codes in your report.**

Answer: The size of the hex file is 5628 bytes.

**4. Can the project be implemented by using interrupt?**

Answer: No, this project cannot be implemented by using interrupt. Interrupts are special routine code that will leave the main routine and run when they occur. As we do not need such operation anywhere for our project, there is no ISR (Interrupt Service Routine) written in code.

1. **Is the main routine required to be an infinite loop? Provide explanation in favor of your answer.**

Answer: Yes, the main routine needs to be an infinite loop in order to check the inputs

continuously and change the outputs according to that. As the loop is infinite it will run and check for inputs and outputs until we cut off power supply to Arduino. As infinite routine gives us real time update, we get our derived output from the given input in real time.

1. **Is there any difference between level triggered and edge triggered operation for the given project?**

Answer: In the given project, there’s no difference between level triggered and edge triggered operation. For the push button inputs, there’s both edge triggered and level triggered signal but output is consistent. So, there’s no difference.

**7. Is the project referring encryption or decryption from input to output?** Answer: The project is referring encryption from input to output.

**10. Discussion:**

We had to implement an encryption input LED with Arduino microcontroller. To implement the project, firstly he had to solve truth table and according to that truth we had to solve a K-Map. Then we used Arduino IDE to write the code according to the K-Map. After writing the code, we generated a Hex file for the Proteus software. Before implementing the hardware we built a simulation in Proteus and burned the Hex file just to see if the Code we wrote were okay or not. At first we were getting some outputs wrong because there were errors in the Arduino code. After examining, we found out that there was a small mistake in the K-map we had made. After fixing it and the code all the outputs were correct. We also drew a circuit diagram from the solved K-Map equation. We had to buy Arduino UNO, one breadboard, battery as we used it as external power source, buttons, wires, LEDs and 10k Ohms. There were many components that were faulty. The LEDs were not working properly, we had to buy more. After solving the software and components related problem we then moved to hardware implementation. During building the hardware we faced a lot of problem regarding setting the wires as there we so many. We also noticed that the voltages across the LEDs were too high. The LEDs were getting fused. But after changing many LEDs, the problem was solved so we didn't put any more resistance. Once finishing the hardware implementation checked all the given situation. As we mentioned that there was a problem with the Arduino code, after solving the problem there were no hardship at all. To bring the hardware from home was also another problem we faced as it was fragile hardware. Any broken wire or button or disruption could broke the hardware so we had to be extra careful bringing the hardware to showcase. Thus finishing our project.

