CSE331

Team Project

Summer 2021

Section # 3

Group # 8

Submitted By

Member Names	ID
Mirza Sabbir Ahmed	1812680042
A.K.M. Ahsanul Habib	1822041042
Kamrujjaman	1821927642
Mohseu Minhaj Niloy	1821274042
MD.Nayeem Hossen Joardar	1821560042

Submitted To

Dr. Dihan Md. Nurudddin Hasan (DMH)

General Description

The dedicated software tools that we are going to use to implement the whole system are

1. Logisim
2. Proteus 8 Professional
3. Arduino UNO

A brief introduction of each of the tools is given below.

Logisim: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller subcircuits, and to draw bundles of wires with a single mouse drag, Logisim can be used to design and simulate entire CPUs for educational purposes.

Our objective is to design a logical circuit using Logisim from the truth table which will later be used to write the Arduino code.

Proteus 8 Professional: The Proteus Design Suite is a proprietary software tool suite used primarily for electric design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing painted circuit boards.

Schematic Capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project. It is therefore a core component and is included with all product configurations.

We have designed our Circuit Diagram With the help of Proteus 8 Professional Software, by using different components like Arduino, Single pole double throw switches, LEDs and Resistors.

To implement our circuit, we will be using Proteus 8 Professional. The microcontroller that we are going to use is Arduino UNO. The other components that we will use are SPDT Switches, LEDs and Resistors.

Arduino UNO: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs, light on a sensor, a finger on a button and turn it into an output, activating a motor, turning on an LED, publishing something online. Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It will be connected with sensors and the GSM and a code will be uploaded to the Arduino UNO for the other components to work properly.

It contains everything needed to support the microcontroller, simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Arduino Uno board can be powered via the USB connection or with an external power supply. The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V then the 5V pin may supply less than five volts and the board may become unstable. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.

Equipment List

1. **Proteus 8 Professional Design Suite**
2. **Logisim**
3. **1 x Arduino Board**
4. **4 x SPDT (Single Pole Double Throw) Switch**
5. **4 x RED-LED**
6. **4 x 330 ohm Resistors**
7. **4 x 10k ohm Resistors**

Method of Derivation

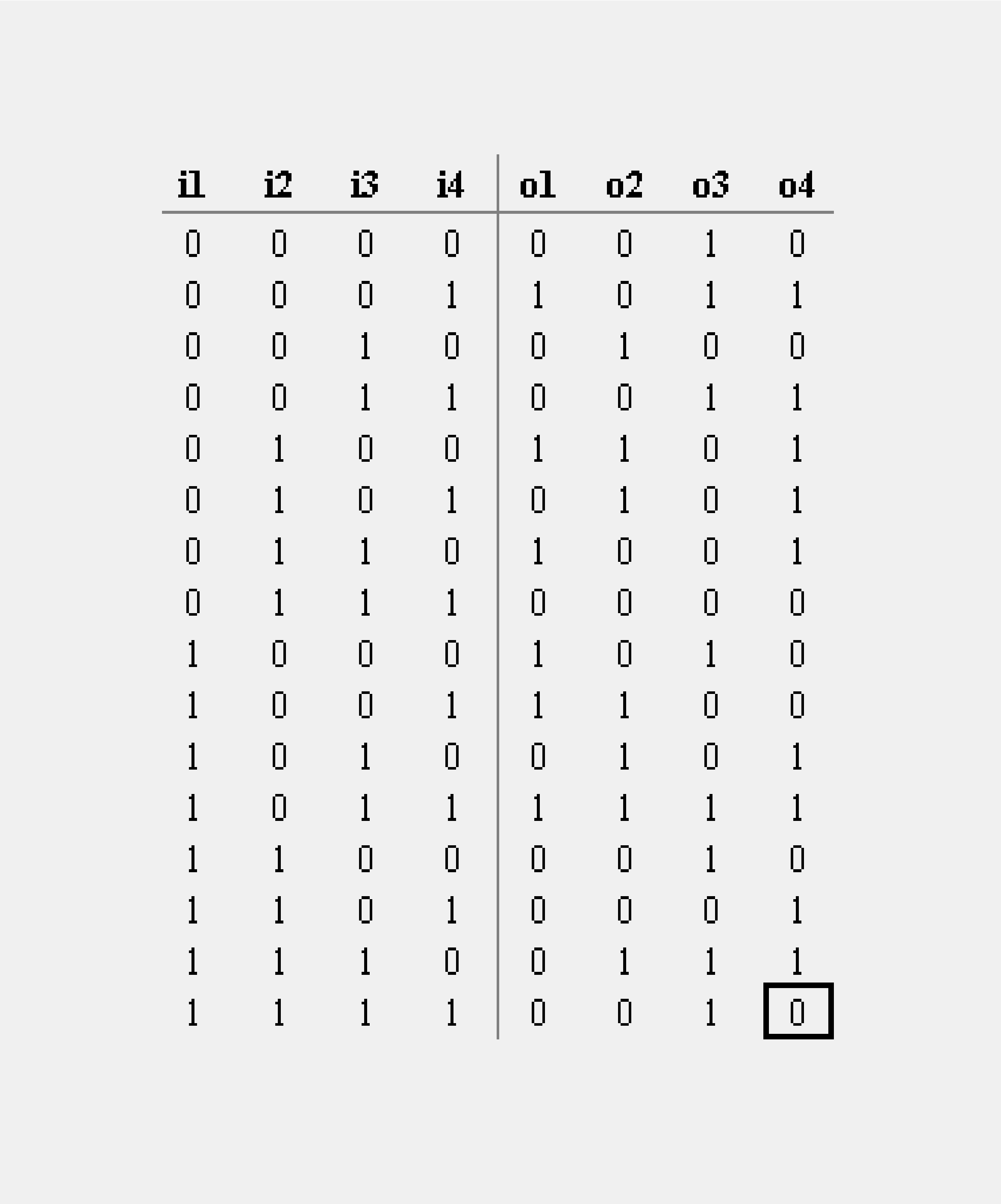
For the given encryption table, we first constructed and solved the k-maps for each of the outputs which are O1, O2, O3 and O4. Using K-maps we derived SOP (Sum of Products) expressions for the outputs with respect to the inputs which are I1, I2, I3 and I4. Using the SOP expression, we built our Logisim circuit. In our case we only used AND, OR and NOT gates to build our Logisim circuit.

According to the Logisim circuit we wrote our Arduino Code. First, we defined the input and output pins and then wrote the code accordingly. After the successful execution of the Arduino code, we got our hex file which was used in the Arduino microcontroller in the circuit which was implemented in proteus.

Finally, we compiled our circuit in proteus and checked the outputs with respect to the inputs.

Derived Results

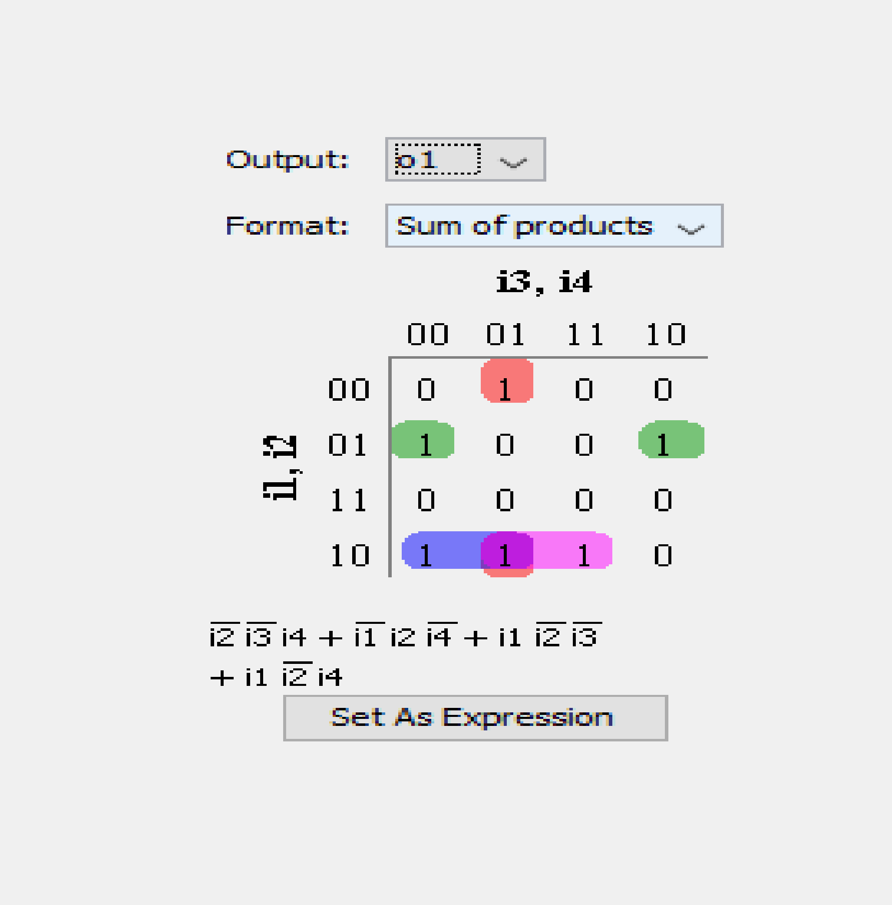
Truth Table



i1	i2	i3	i4	o1	o2	o3	o4
0	0	0	0	0	0	1	0
0	0	0	1	1	0	1	1
0	0	1	0	0	1	0	0
0	0	1	1	0	0	1	1
0	1	0	0	1	1	0	1
0	1	0	1	0	1	0	1
0	1	1	0	1	0	0	1
0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	0
1	0	0	1	1	1	0	0
1	0	1	0	0	1	0	1
1	0	1	1	1	1	1	1
1	1	0	0	0	0	1	0
1	1	0	1	0	0	0	1
1	1	1	0	0	1	1	1
1	1	1	1	0	0	1	0

K-Maps

1. **Output O1**



Output: o1

Format: Sum of products

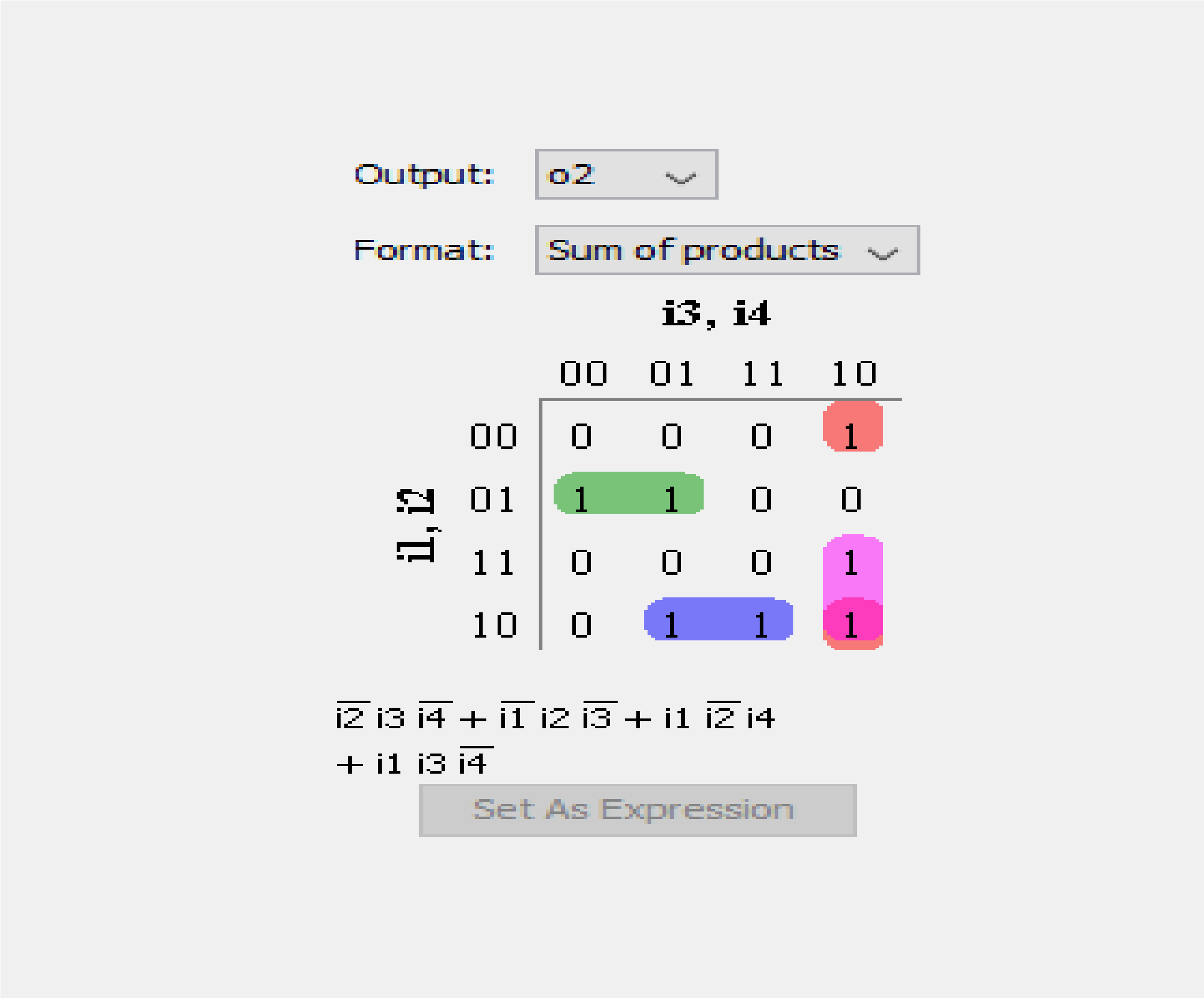
i3, i4

i1, i2	00	01	11	10
00	0	1	0	0
01	1	0	0	1
11	0	0	0	0
10	1	1	1	0

$\overline{i2}\ \overline{i3}\ i4 + \overline{i1}\ i2\ \overline{i4} + i1\ \overline{i2}\ \overline{i3}$
$+ i1\ \overline{i2}\ i4$

Expression: O1 = **I2’ I3’I4 + I1’ I2 I4’ + I1 I2’ I3’+I1 I2’ I4**

2. Output O2



Output: o2

Format: Sum of products

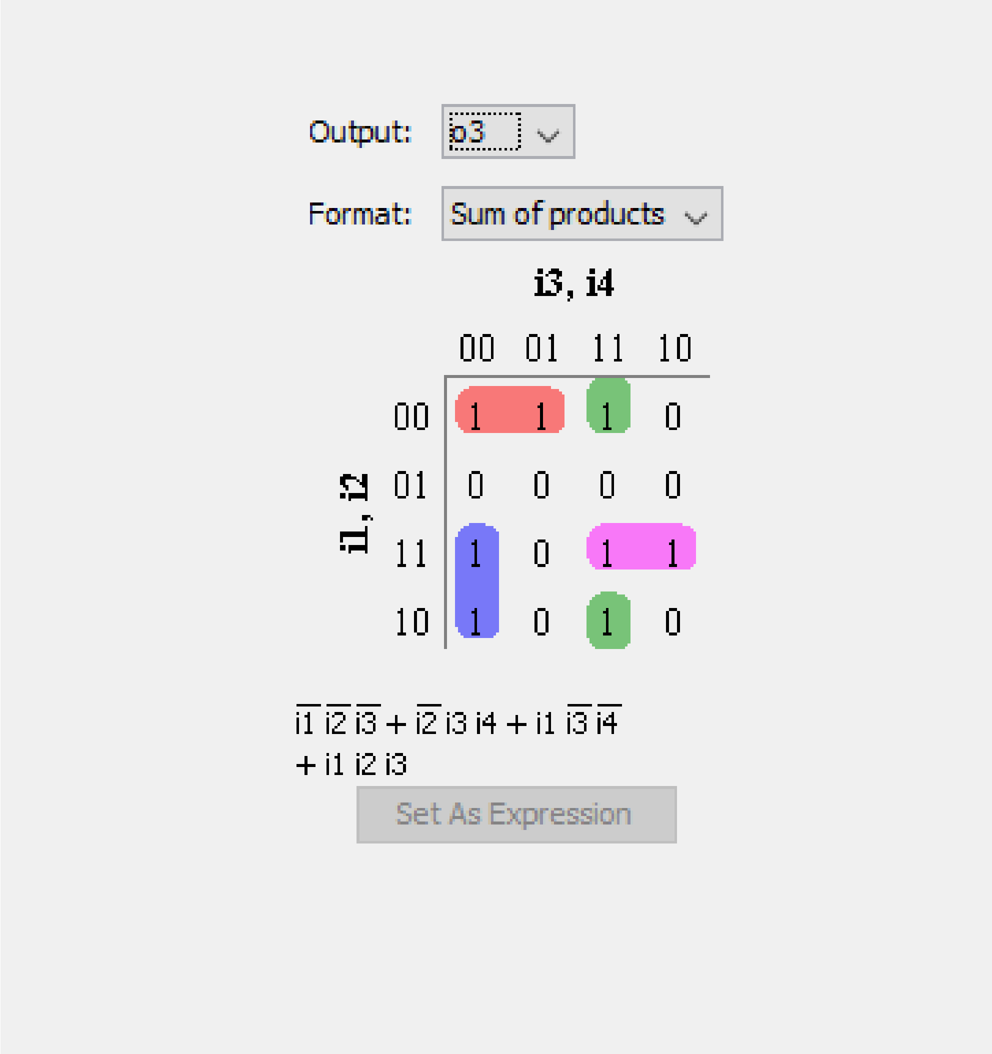
i3, i4

		00	01	11	10
i1, i2	00	0	0	0	1
	01	1	1	0	0
	11	0	0	0	1
	10	0	1	1	1

$\overline{i2}\ i3\ \overline{i4} + \overline{i1}\ i2\ \overline{i3} + i1\ \overline{i2}\ i4$
$+ i1\ i3\ \overline{i4}$

Expression: O2 **= I2’ I3 I4’ + I1’ I2 I3’ + I1 I2’ I4+I1 I3 I4’**

3. Output O3



Output: o3

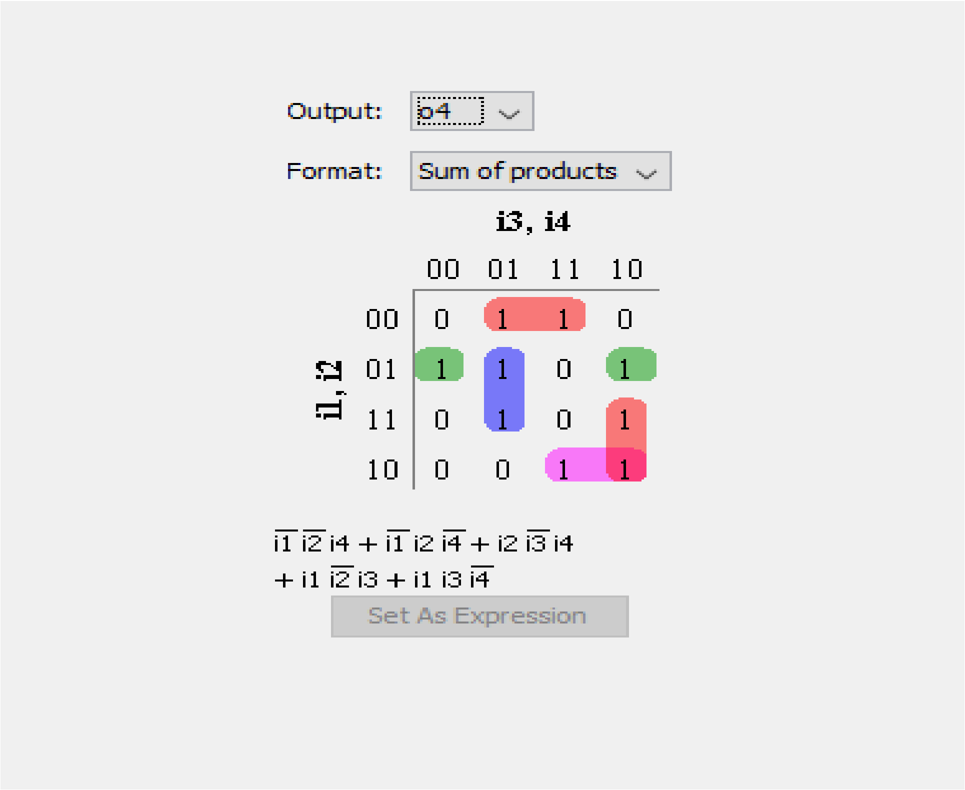
Format: Sum of products

i1, i2 \ i3, i4	00	01	11	10
00	1	1	1	0
01	0	0	0	0
11	1	0	1	1
10	1	0	1	0

$\overline{i1}\ \overline{i2}\ \overline{i3} + \overline{i2}\ i3\ i4 + i1\ \overline{i3}\ \overline{i4}$
$+ i1\ i2\ i3$

Expression: O3 = **I1’ I2’ I3’ + I2’ I3 I4 + I1 I3’ I4+I1 I2 I3**

4. Output O4



Output: o4

Format: Sum of products

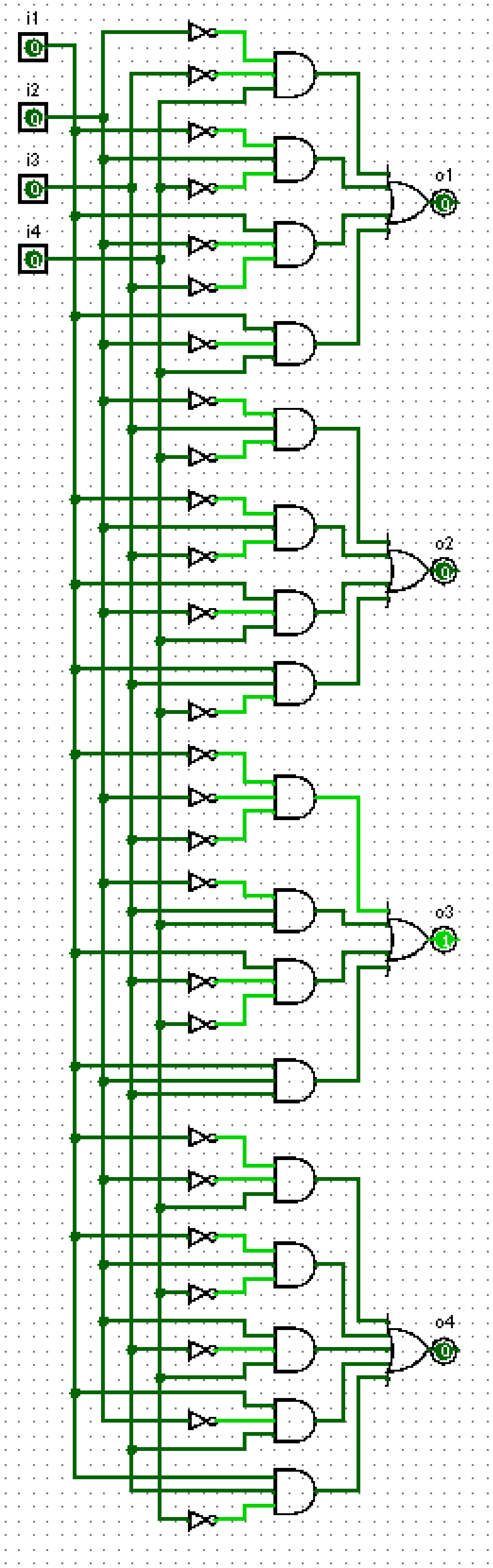
i3, i4

i1, i2	00	01	11	10
00	0	1	1	0
01	1	1	0	1
11	0	1	0	1
10	0	0	1	1

$\overline{i1}\ \overline{i2}\ i4 + \overline{i1}\ i2\ \overline{i4} + i2\ \overline{i3}\ i4$
$+ i1\ \overline{i2}\ i3 + i1\ i3\ \overline{i4}$

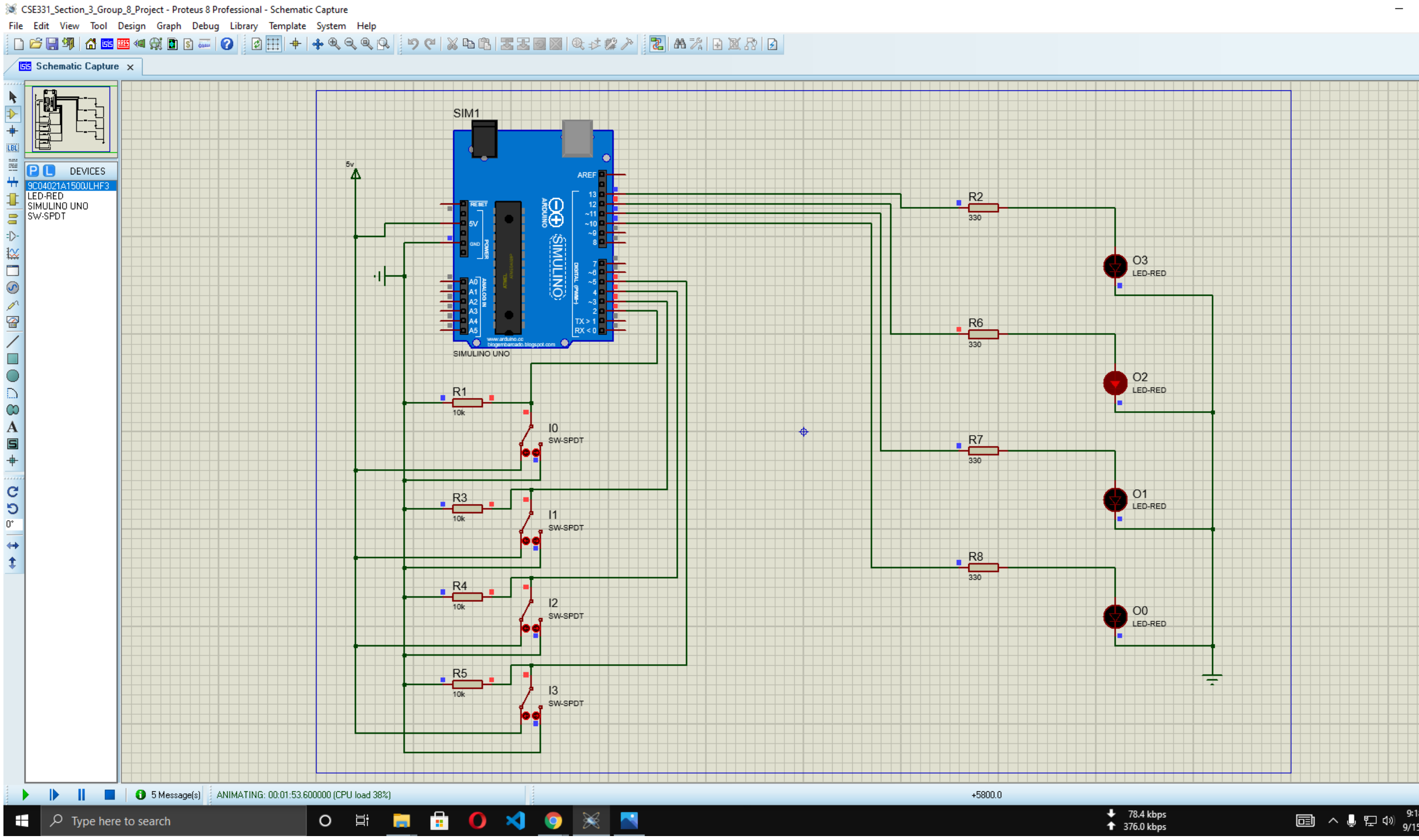
Expression: O3 = **I1’ I2’ I4 + I1’ I2 I4’ + I2 I3’I4 + I1 I2’ I3+I1 I3 I4’**

Logisim Circuit



Circuit Diagram

Figure – 1

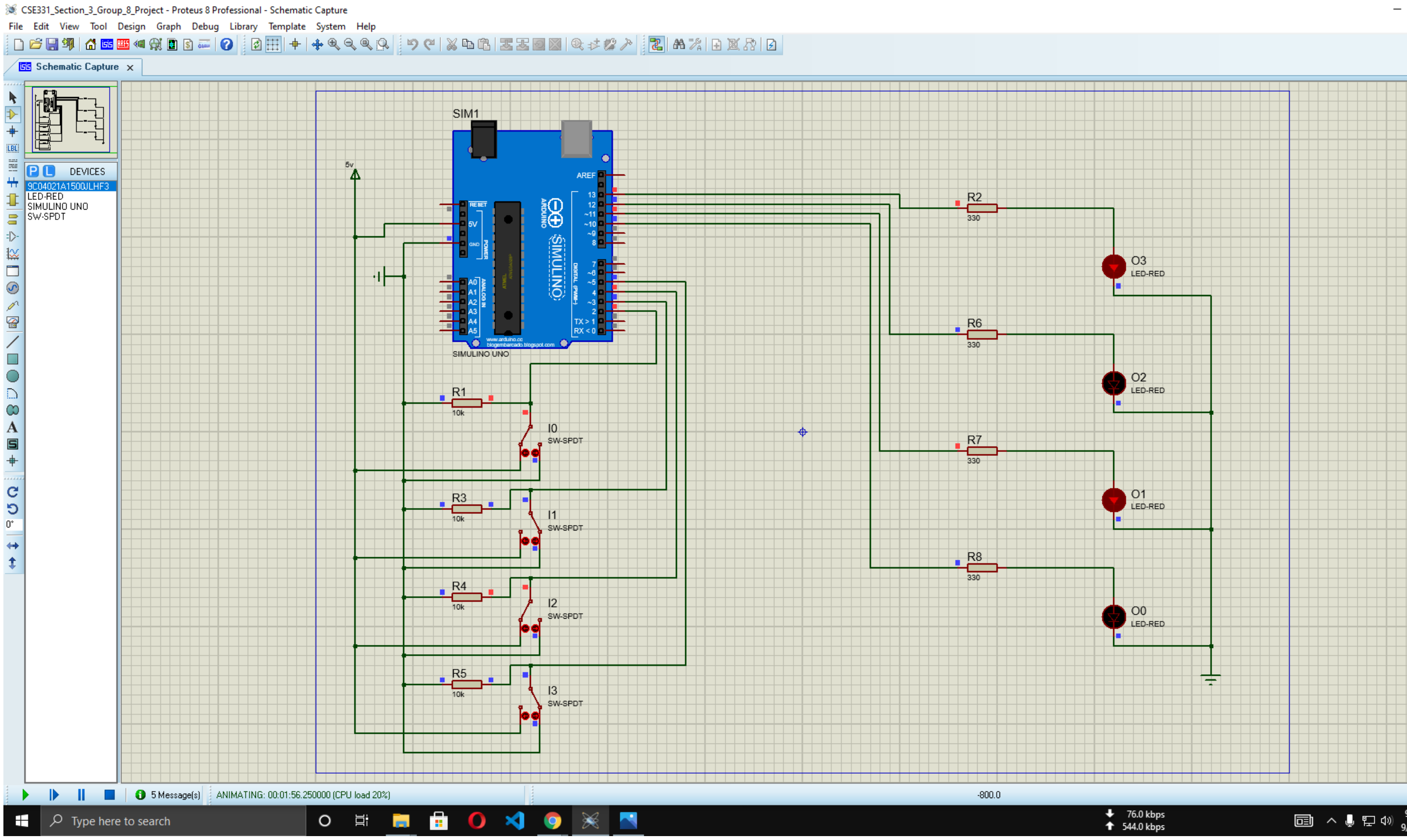


5 Message(s)

ANIMATING: 00:01:53.600000 (CPU load 38%)

In this diagram, the input SPDT Switches are set to **I0 = 0, I1 = 0, I2 = 0, I3 = 0** and the output that are coming through the Red LEDs are **O0 = 1, O1 = 1, O2 = 0, O3 = 1**

Figure – 2



In this diagram, the input SPDT Switches are set to **I0 = 1, I1 = 0, I2 = 1, I3 = 0** and the output that are coming through the Red LEDs are **O0 = 0, O1 = 0, O2 = 1, O3 = 0**

Values of Electrical Component

In our circuit we have used microcontroller Arduino UNO with Four SPDT (Single Pole Double Throw) Switches as Inputs and Four RED-LEDs as Outputs. The Switches and LEDs are connected in the Digital Pins of the Arduino UNO. We used total four 10k ohm resistors for the SPDT switches and four 330 ohm resistors for the RED-LEDs.

Circuit Operation Principles

At first, we have derived Karnaugh Map from the truth table and calculated output expressions for each of the outputs which are O0, O1, O2 and O3.

- O1 ❼ **I2’ I3’I4 + I1’ I2 I4’ + I1 I2’ I3’+I1 I2’ I4**
- O2 ❼ **I2’ I3 I4’ + I1’ I2 I3’ + I1 I2’ I4+I1 I3 I4’**

- O3 ❼ **I1’ I2’ I3’ + I2’ I3 I4 + I1 I3’ I4+I1 I2 I3**
- O4 ❼ **I1’ I2’ I4 + I1’ I2 I4’ + I2 I3’I4 + I1 I2’ I3+I1 I3 I4’**

With the help of these expressions, we have designed our logic circuit diagram using Logic Gates which consists of AND, OR and NOT gates only. The logic circuit diagram was built using Logisim.

The Arduino code was built with the help of those expressions and the truth table as well.

Finally, with the help of Proteus 8 Professional Software we have built our main Circuit.

Throughout the whole process, we have used the following components

1. **Arduino Uno**: We have used an Arduino Uno Model as our microcontroller which is the main part of this project. We connected the 5V pin of Arduino to a power supply of 5V. And we connected the ground pin of the Arduino to the Ground component.
2. **Resistors**: We have used a total 8 resistors of two values. We have used 330 ohm resistors with the RED-LEDs and we have used 10k ohm resistors with the SPDT Switches.
3. **SPDT (Single Pole Double Throw) Switches**: We used four of these Switches with our Arduino Digital pins. They are arranged in such a way:
    - **Arduino UNO Digital Pin No. 2 ❼ SPDT Switch 1 (I0)** •
      Arduino UNO Digital Pin No. 3 ❼ SPDT Switch 2 (I1) •
      Arduino UNO Digital Pin No. 4 ❼ SPDT Switch 3 (I2) •
      Arduino UNO Digital Pin No. 5 ❼ SPDT Switch 4 (I3)

Also, we connected the 10k Resistor to one side of the Digital Pin that is connected to the Switch and another side to Ground. Like this orientation, we placed all the four Switches with all the four 10k resistors.

4. **LEDs**: We have connected Four LEDs to the Digital Pins of Arduino to receive Output through those Pins. They are arranged in such way:
    - **Arduino UNO Digital Pin No. 10 ❼ LED1 (O0)** •
      Arduino UNO Digital Pin No. 11 ❼ LED2 (O1) •
      Arduino UNO Digital Pin No. 12 ❼ LED3 (O2)
    - **Arduino UNO Digital Pin No. 13 ❼ LED4 (O3)**

After we connected all the wires in place, we executed the Arduino Code which generated the HEX file that we used in the Arduino in our circuit in Proteus.

The detail explanation can be found on the video demonstration. The link is added below.

Arduino Code with no Compiling Error

```
int O0 = 10;
int O1 = 11;
int O2 = 12;
int O3 = 13;

int I0 = 2;
int I1 = 3;
int I2 = 4;
int I3 = 5;

void setup()
{
  pinMode(I0,INPUT);
  pinMode(I1,INPUT);
  pinMode(I2,INPUT);
  pinMode(I3,INPUT);
  pinMode(O0,OUTPUT);
  pinMode(O1,OUTPUT);
  pinMode(O2,OUTPUT);
  pinMode(O3,OUTPUT);
}
void loop()
{
  boolean I0State = digitalRead(I0);
  boolean I1State = digitalRead(I1);
  boolean I2State = digitalRead(I2);
  boolean I3State = digitalRead(I3);
  boolean O0State;
  boolean O1State;
  boolean O2State;
  boolean O3State;

 O0State= ((!I1State&!I2State&I3State))|((!I0State&I1State&!I3State))|((I0State&!I1State&!I2State))|((I0State&!I1State&I3State));
 O1State= ((!I1State&I2State&!I3State))|((!I0State&I1State&!I2State))|((I0State&!I1State&I3State))|((I0State&I2State&!I3State));
 O2State= ((!I0State&!I1State&!I2State))|((!I1State&I2State&I3State))|((I0State&!I2State&!I3State))|((I0State&I1State&I2State));
 O3State= ((!I0State&!I1State&I3State))|((!I0State&I1State&!I3State))|((I1State&!I2State&I3State))|((I0State&!I1State&I2State))|((I0State&I2State&!I3State));

  digitalWrite(O0,O0State);
  digitalWrite(O1,O1State);
  digitalWrite(O2,O2State);
  digitalWrite(O3,O3State);
}
```

Question and Answers

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

Answer: 16Mhz

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

Answer: 8-Bits

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

Answer: Sketch uses 2346 bytes (7%) of program storage space. Maximum is 32256 bytes.

Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

4. Can the project be implemented by using interrupt?

Answer: Yes.

Interrupts in Arduino: Interrupts are a mechanism by which an I/O or instruction can suspend the normal execution of the processor and gets itself serviced like it has higher priority. It has two types of interrupts.

External Interrupt: These interrupts are interpreted by hardware and are very fast. These interrupts can be set to trigger on the event of RISING or FALLING or LOW levels.

Pin Change Interrupts: Arduinos can have more interrupt pins enabled by using pin change interrupts. In ATmega168/328 based Arduino boards any pins or all the 20 signal pins can be used as interrupt pins. They can also be triggered using RISING or FALLING edges.

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

Answer: Yes. The main routine is required to be an infinite loop because If we want the program to execute once only, we should end it with a while loop.

For instance:

```
main()

{
        ……….
……….
while(1)
        {
                ……….
                ……….
        }
}
```

Otherwise, it automatically restarts.

To elaborate, main is a function called by the startup code, when main terminates it returns. The startup code then executes a reset. This startup code is added by the linker.

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

Answer: Edge Triggering is a type of triggering that allows a circuit to become active at the positive edge or the negative edge of the clock signal. And Level Triggering is a type of triggering that allows a circuit to become active when the clock pulse is on a particular level.

In this given project with the Microcontroller Arduino UNO, the Edge Triggering and the Level Triggering is the same.

7. **7.** Is the project referring to encryption or decryption from input to output?

Answer: The project refers to decryption from input to output.

Google Drive Link

https://drive.google.com/drive/u/1/folders/1HJMoGXHJ78RvyWifGEb-ITGSJ2Zo0wmA?fbclid=IwAR0Z0L6W_DjDkX-WGLl_V-IKjfdEMII2mDaKsN5Njw8lif02QNLfztGlMZA