Department of Electrical and Computer Engineering

Project Report

Course: CSE331 (Microprocessor Interfacing and Embedded System)

Section: 04

Summer: FALL 2022

Group No: 05

Project: Implementation of an Encryption Table Using Microcontroller

Submitted To:

Dr. Dihan Md. Nuruddin Hasan

Submission Date: 1/1/2023

GROUP MEMBER:

Name	ID	Deed
Sadik Ittesaf Abir	1931595042	Sketch code burning, generating hex, proteus simulation, hardware set up
Sumit Saha	1931415042	Sketch code burning, generating hex, proteus simulation, hardware set up
Sabbir Ahmed	1931098042	Report writing, hardware implementation
Faisal Hasan Tanjil	2013162042	Hardware Implementation
Jannatul Ferdous Sristy	1931533042	Kmap, Logisim
Tasfia Aktar	1911490642	Logisim,K-map,circuit build

Table of Contents

Abstract

Our projects main goal is to Implement the given encryption table using microcontroller. Using single pole, double throw switch to configure the inputs for high and low conditions and to use LED lights represent the corresponding output statuses.Arduino IDE To code the Arduino we used the Arduino IDE.The Arduino Integrated Development Environment (IDE). It connects to the Arduino hardware to upload and communicate with programs..Proteus Sketch is used to test and virtually prototyping embedded system designs based on Microchip Technologies microcontrollers to simulate the ciruit.Logisim is a logic simulator that allows circuits to be designed and simulated through the use of a graphical user interface.

General Description

Microprocessor

A microprocessor is the controlling unit of a microcomputer, which is constructed on a tiny chip and can perform ALU (Arithmetic Logical Unit) functions as well as communicate with other devices connected to it. A control unit, an ALU, and a register array make up the microprocessor.

- ★ **Control Unit:** The control unit controls the computer's data and command flow
- ★ **ALU:** An ALU is a combinational digital circuit that performs arithmetic and logical operations on data from a memory or input system.
- ★ **Register Array:** The register array is made up of registers with letters like B, C, D, E, H, L, and accumulator.

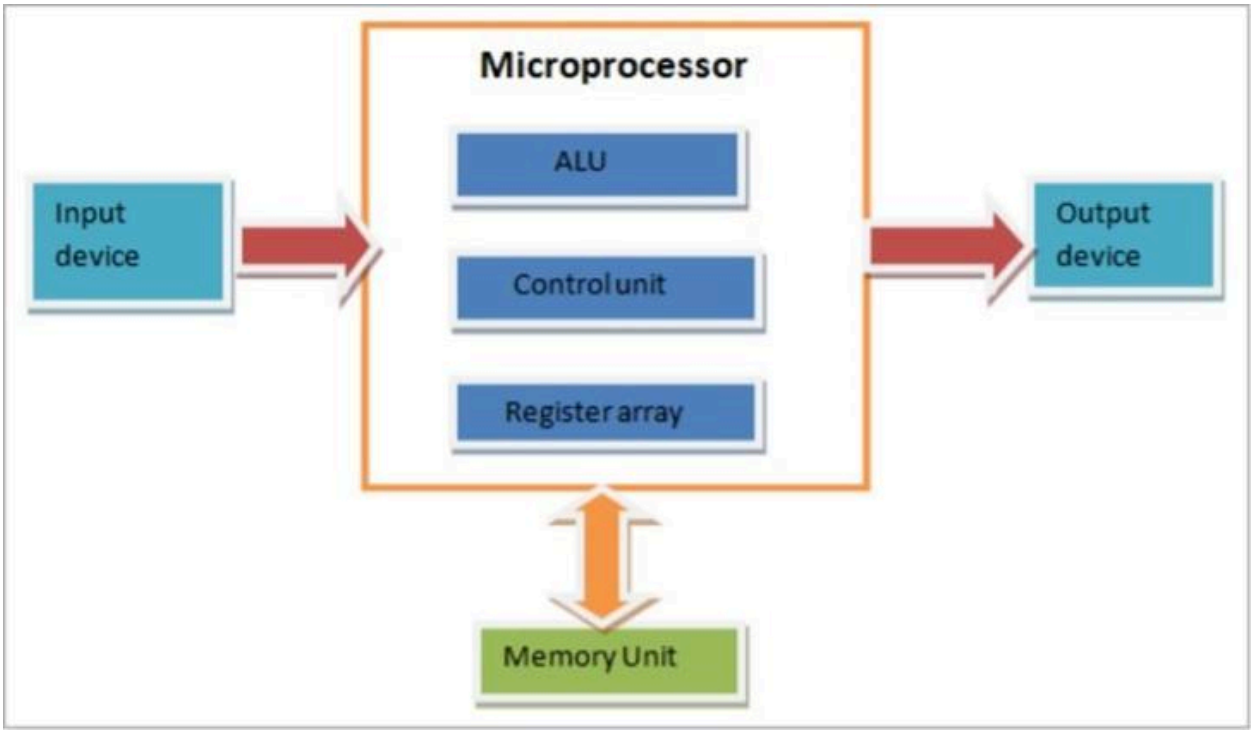


Figure: Block Diagram of a microcomputer

8085 Microprocessor

Intel introduced the Intel 8085, an 8-bit microprocessor, in March 1976. With the exception of two small instructions introduced to handle the interrupt and serial input/output features, It's a software-binary compatible with the more well-known Intel 8080.

8086 Microprocessor

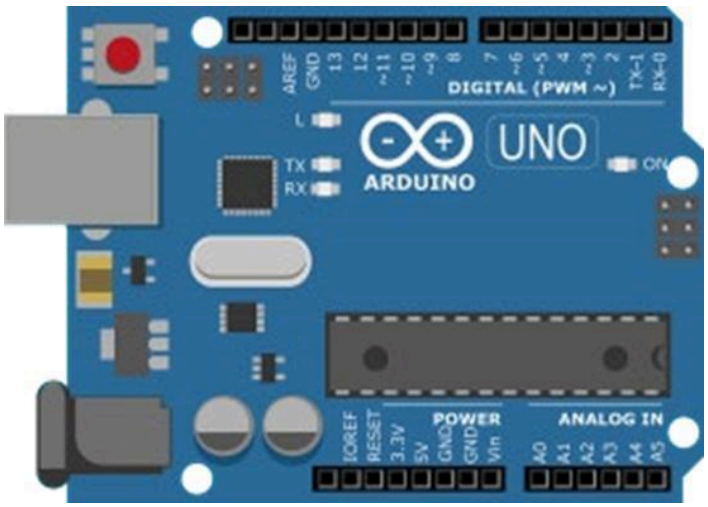
The Intel 8086 Microprocessor is a more powerful version of the Intel 8085 Microprocessor, which was first released in 1976. It's a 16-bit microprocessor with 20 address lines and 16 data lines and a memory capacity of 1MB. It has a flexible instruction set that makes operations like multiplication and division, easier. It has two operating modes: maximal and minimal. Maximum mode is ideal for multi-processor computers, whereas Minimum mode is appropriate for single-processor systems.

Microcontroller

A microcontroller is a compact and low-cost microprocessor designed to handle specialized functions of embedded systems such as displaying microwave information or receiving distant signals, among other things.

1.1 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 (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. The Uno board is the first in a series of USB-based Arduino boards;it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases.[4] The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter



1.2 Arduino IDE

The software used for writing, compiling & uploading code to Arduino boards is called Arduino IDE (Integrated Development Environment). It is a cross-platform software that is available for every Operating System like Windows, Linux, macOS. This software can be used with any Arduino board. We write our code in this compiler and then connect the microcontroller via USB to upload the code in it. Arduino IDE consists of different sections such as WindowBar, MenuBar, ShortcutButtons, Text Editor, Output Panel.

1.3 Proteus 8 pro

Proteus 8 Professional is software that can be used to draw schematics, PCB layout, code, and even simulate the schematic. The Proteus Design Suite is a software suite containing schematic, simulation as well as PCB designing. 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 simulate our circuit in Proteus 8 Pro using Arduino Uno, resistors, single pole double throw switches, LED lights.

2.Equipments

Hardware

- Resistors
- Arduino UNO R3
- Arduino Nano
- Wires
- LED light
- Push Buttons
- Breadboard

Software

- Arduino IDE
- Proteus
- Logisim

3.Method of derivation:

We used the given table of Group-7 and we built it into a truth table for inputs and outputs. Then we used the concept of SOP to produce the KMAPs below and then found the logical expression for each of the outputs using KMAP.

3.1 Truth Table

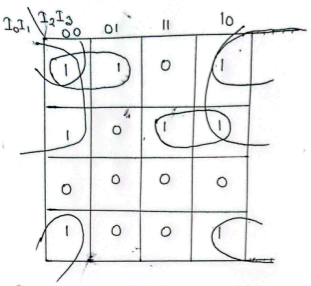
Input				Output			
I3	I2	I1	I0	O3	O2	O1	O0
0	0	0	0	0	1	0	1
1	0	0	0	1	1	0	0
0	1	0	0	1	1	1	0
1	1	0	0	0	0	0	1
0	0	1	0	0	1	1	1
1	0	1	0	0	0	1	0
0	1	1	0	0	1	1	0
1	1	1	0	0	1	0	0
0	0	0	1	1	1	0	0
1	0	0	1	0	0	0	0
0	1	0	1	0	1	1	0
1	1	0	1	1	0	1	1
0	0	1	1	0	0	0	0
1	0	1	1	0	0	1	1
0	1	1	1	1	0	1	1
1	1	1	1	0	0	1	1

3.2 Derived Result:

We use the K-Map to derive the expression for the output.

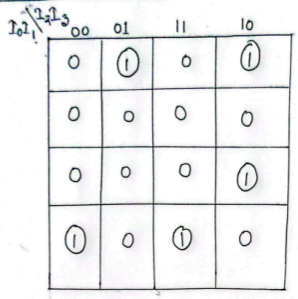
KMAP:

Equation For O_2 :



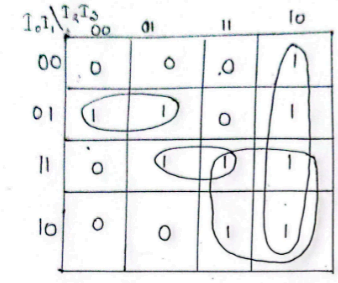
$$F_{O_2} = I_0'I_1'I_2' + I_0'I_3' + I_0'I_1I_2 + I_1'I_3'$$
$$= I_0'(I_1'I_2' + I_1I_2) + I_0'I_3' + I_1'I_3'$$
$$= I_0'(I_1 \odot I_2) + I_0'I_3' + I_1'I_3'$$

Equation For O_3 :



$$F_{O_3} = I_0'I_1'I_2'I_3 + I_0'I_1'I_2I_3' + I_0I_1I_2I_3' + I_0I_1'I_2'I_3' + I_0I_1'I_2I_3$$
$$= I_0'I_1'(I_2'I_3 + I_2I_3') + I_0I_1I_2I_3' + I_0I_1'(I_2'I_3' + I_2I_3)$$
$$= I_0'I_1'(I_2 \oplus I_3) + I_0I_1I_2I_3' + I_0I_1'(I_2 \odot I_3)$$

Equation For O_1 :



$$F_{O_1} = I_0'I_1I_2' + I_0I_1I_3 + I_0I_2$$
$$= I_1(I_0'I_2' + I_0I_3) + I_2I_3' + I_0I_2$$

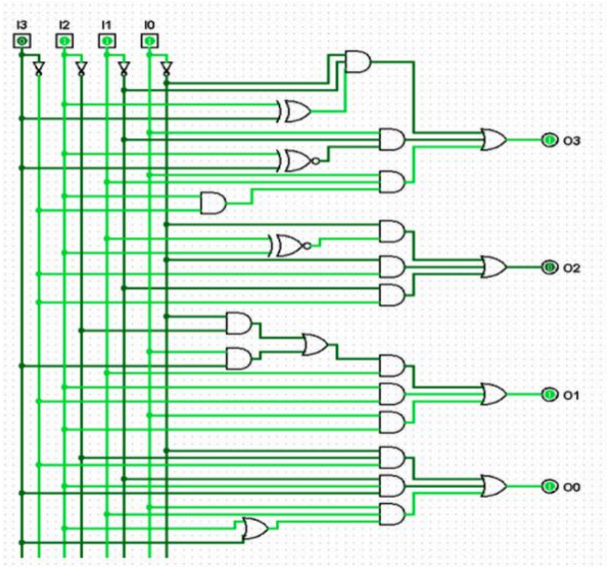
Equation for O_0 :

I_0I_1 \ I_2I_3	00	01	11	10
00	1	0	1	0
01	1	0	0	0
11	0	1	1	1
10	0	0	1	0

$$F_{O_0} = I_0' I_2' I_3' + I_1' I_2 I_3 + I_0 I_1 I_2 + I_0 I_1 I_3$$

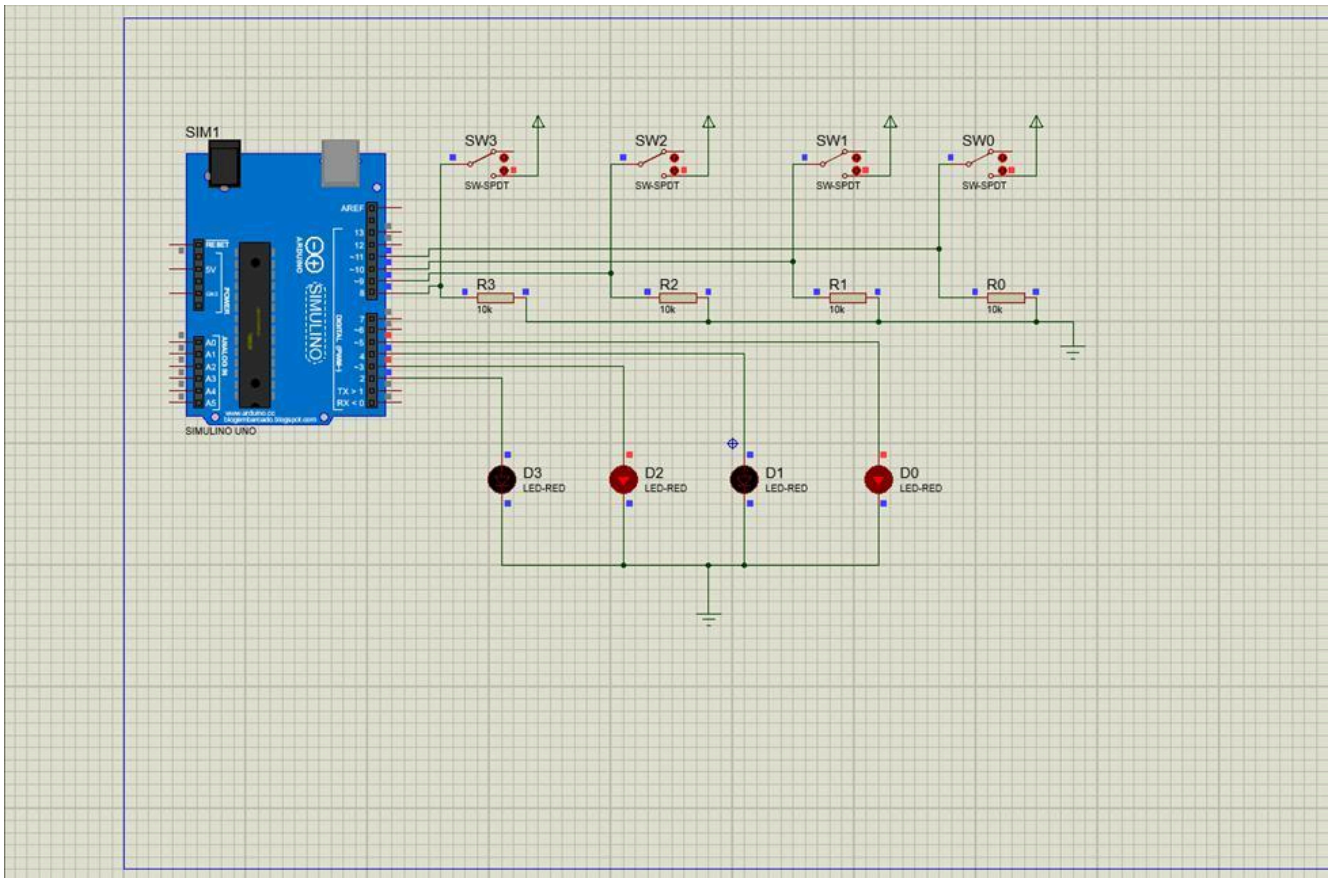
$$= I_0' I_2' I_3' + I_1' I_2 I_3 + I_0 I_1 (I_2 + I_3)$$

Circuit Diagram



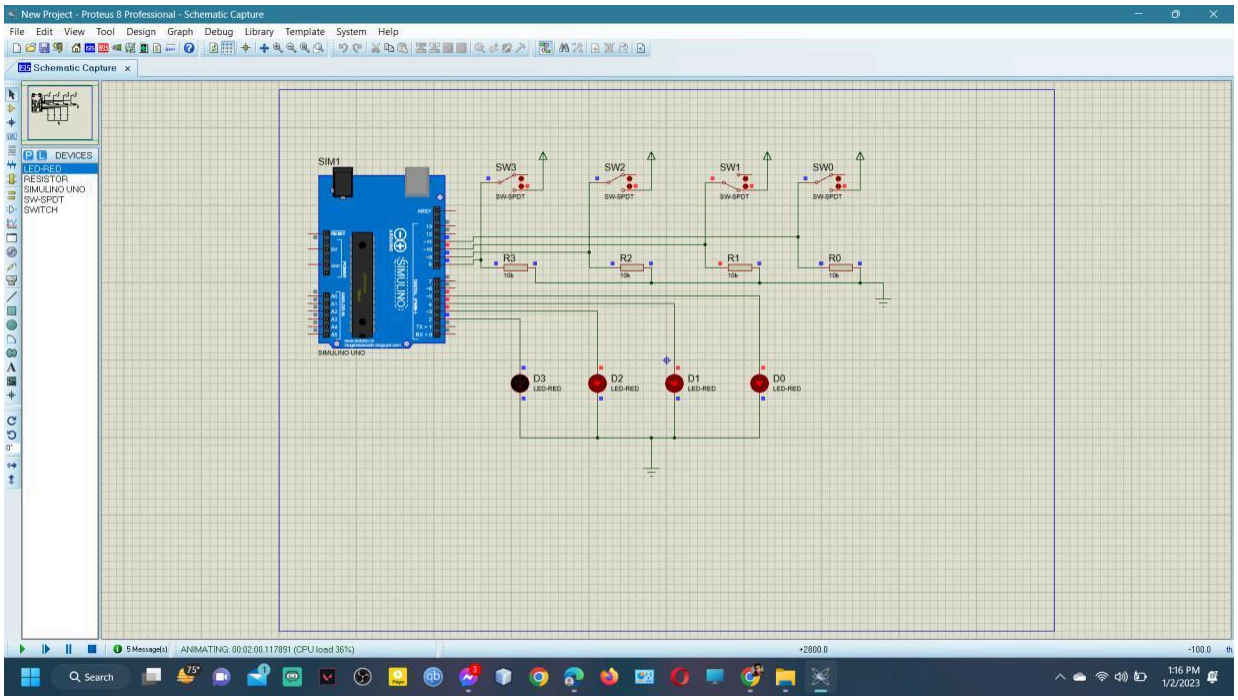
4. Circuit Diagram with Values of Electrical Components

4.1 Figure 1:



Here we can see the circuit in Proteus 8, for the inputs are I3=0, I2=0, I1=0, I0=0 the LED outputs are O4=0, O3=1, O2=0, O1=1

4.1 Figure 2:



Here we can see the circuit in Proteus 8, for the inputs are I3=0, I2=0, I1=1, I0=0 the LED outputs are O4=0, O3=1, O2=1, O1=1

5. Circuit Operation

Principles: Arduino Uno:

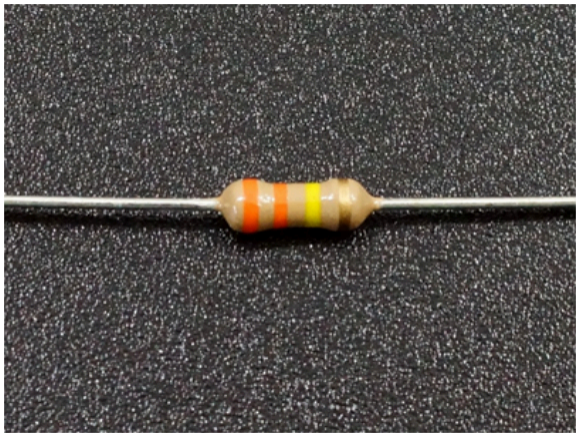
We have used Proteus 8 pro to build the hardware circuit diagram, the components used in proteus are Arduino UNO, this is the microcontroller we used in the simulation.

Button Switch:

We have used button switches to connect with Arduino pins. Then we connected the wire with the circuit.

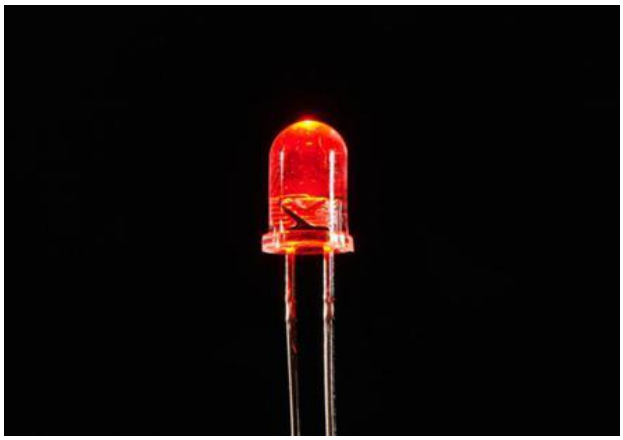
Resistors:

We used 4 10K -ohm resistors to connect with the switches and 4 330 -ohm resistors to connect with the LED outputs



LEDs:

We connected 4 LED outputs O0, O1, O2, O3 to Arduino Nano pin no D12, D11, D10, D9 respectively. After building the circuit we uploaded the Arduino code to the Nano and simulated the project. We uploaded a video on Drive to demonstrate the Proteus simulation of the circuit.



6. Circuit operation principles:

Our project is built with Arduino. The Arduino microcontroller board is based on the ATmega328P microprocessor (datasheet). It contains 14 digital I/O pins (six of which are PWM outputs), 6 analog input pins, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power connector, an ICSP header, and a reset button. It provides everything required to support the microcontroller.

The Four Input Pins are :

Arduino I/O pin 8 – Input Switch I3 Arduino I/O

pin 9 – Input Switch I2 Arduino I/O pin 10 – Input

Switch I1 Arduino I/O pin 11 – Input Switch I0

The Four Output Pins are :

Arduino I/O pin 2 – Output LED O3

Arduino I/O pin 3 – Output LED O2

Arduino I/O pin 4 – Output LED O1

Arduino I/O pin 5 – Output LED O0

The operation of an Arduino circuit is straightforward. To link switches in the Input area, we added resistors. Resistors regulate the flow of electricity via other materials. In the Output section, we added our LED lights. Because we placed the Arduino code's hex file into the Arduino board, the entire operation is dependent on our Arduino code. When we have 13 finished all of the connections,We will implement Input Switches, which will act according to our code and generate the required output via LED lights.

Arduino Program Code

```
int pinOut1 = 5;

int pinOut2 = 4;

int pinOut3 = 3;

int pinOut4 = 2;


int pinA = 11;

int pinB = 10;

int pinC = 9;

int pinD = 8;


void setup()

{

pinMode(pinOut1, OUTPUT);

pinMode(pinOut2, OUTPUT);

pinMode(pinOut3, OUTPUT);
```

```
pinMode(pinOut4, OUTPUT);

pinMode(pinA, INPUT);

pinMode(pinB, INPUT);

pinMode(pinC, INPUT);

pinMode(pinD, INPUT);

}


void loop()

{

boolean pinAState = digitalRead(pinA);

boolean pinBState = digitalRead(pinB);

boolean pinCState = digitalRead(pinC);

boolean pinDState = digitalRead(pinD);

boolean pinorOutState;

boolean pinorOutState1;

boolean pinorOutState2;

boolean pinorOutState3;


pinorOutState =
((!pinAState)&(!pinCState)&(!pinDState))|((!pinBState)&(pinCState)&(pinDState))|((pinA
State)&(pinBState)&(pinCState))|((pinAState)&(pinBState)&(pinDState));

digitalWrite(pinOut1, pinorOutState);
```

pinorOutState1 =((!pinAState)&(pinBState)&(!pinCState))|((pinAState)&(pinBState)&(pinDState))|((pinCState)&(!pinDState))|((pinAState)&(pinCState));

digitalWrite(pinOut2, pinorOutState1);

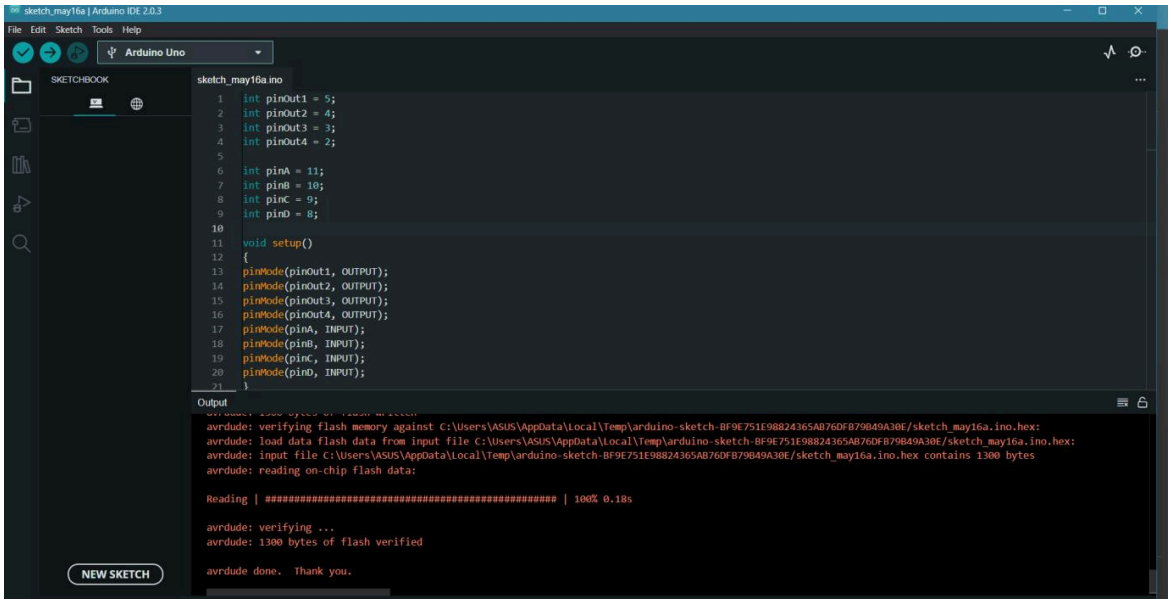
pinorOutState2 = ((!pinAState)&(!pinBState)&(!pinCState))|((!pinAState)&(!pinDState))|((!pinAState)&(pinBState)&(pinCState))|((!pinBState)&(!pinDState));

digitalWrite(pinOut3, pinorOutState2);

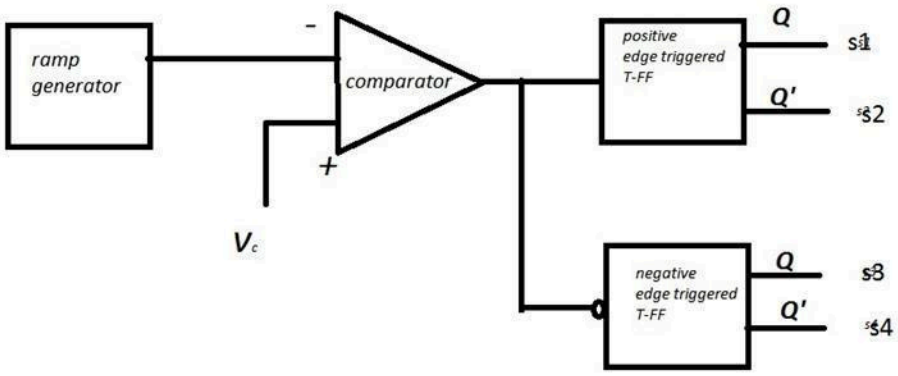
pinorOutState3 = ((!pinAState)&(!pinBState)&(!pinCState)&(pinDState))|((!pinAState)&(!pinBState)&(pinCState)&(!pinDState))|((pinAState)&(pinBState)&(pinCState)&(!pinDState))|((pinAState)&(!pinBState)&(!pinCState)&(!pinDState))|((pinAState)&(!pinBState)&(pinCState)&(pinDState));

digitalWrite(pinOut4, pinorOutState3);

}



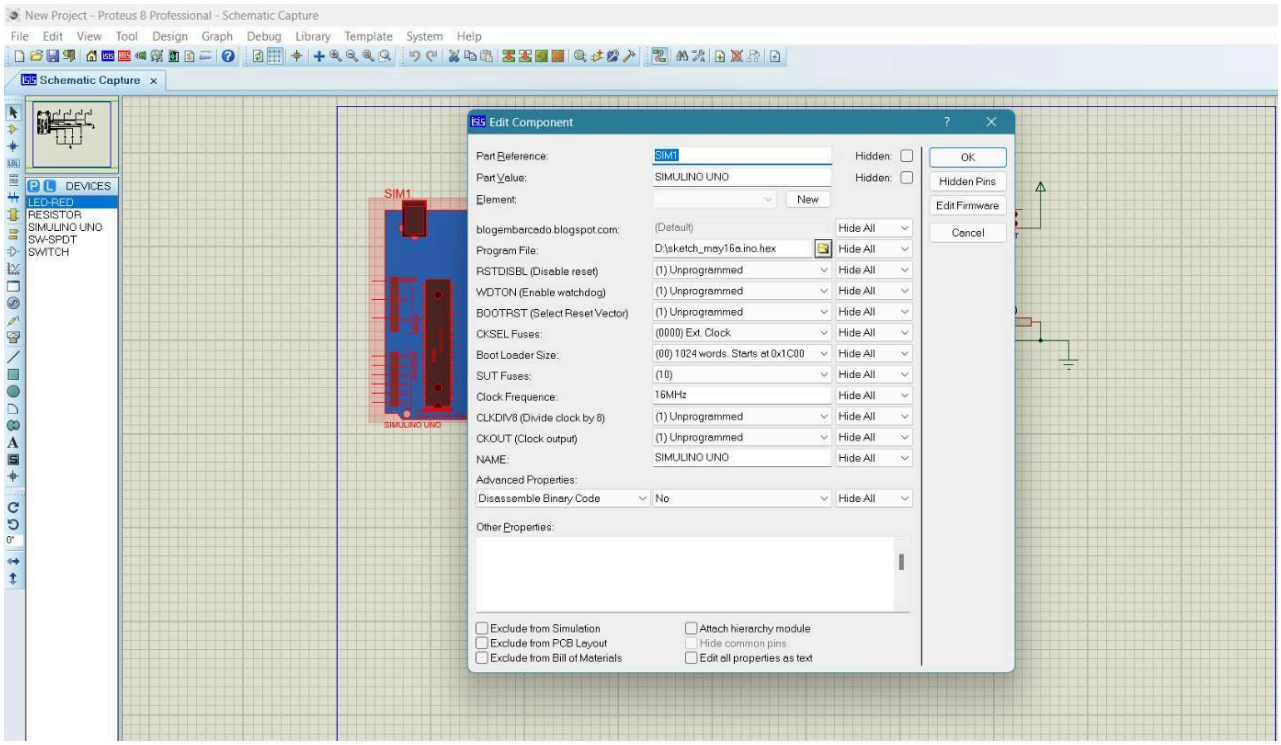
7. Proteus flow chart



HEX CODE:

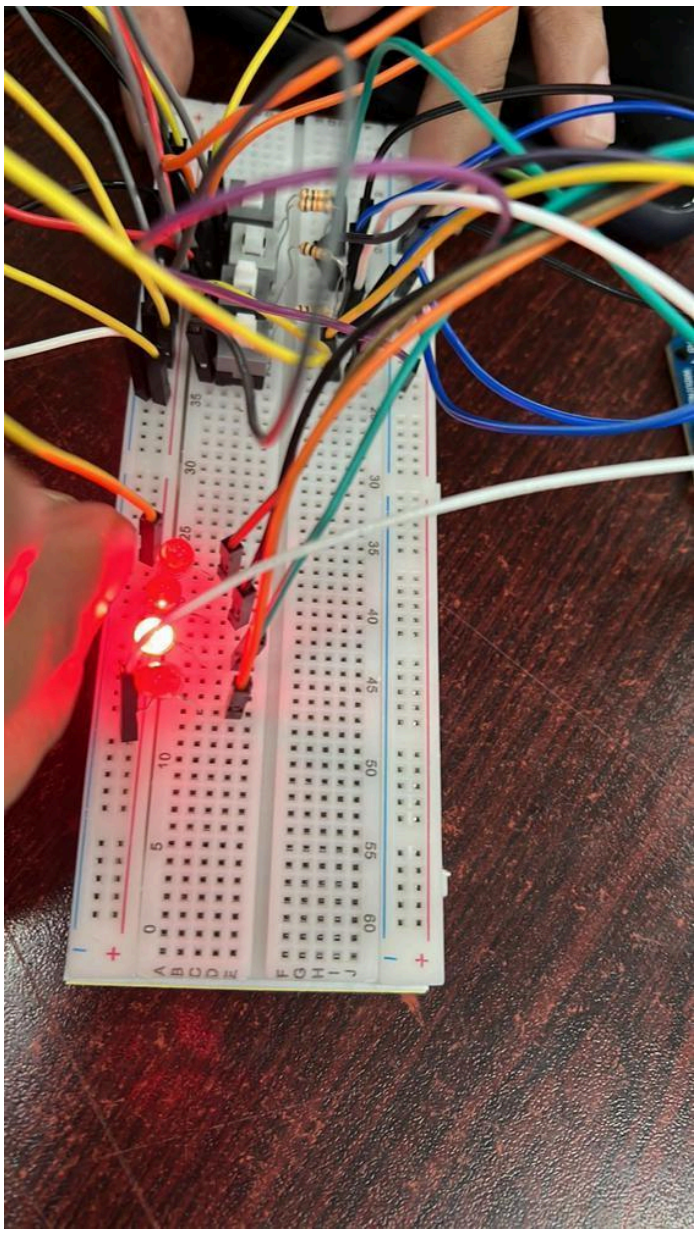
```
100000000C9461000C9473000C9473000C947300B6
:100010000C9473000C9473000C9473000C94730094
:100020000C9473000C9473000C9473000C94730084
:100030000C9473000C9473000C9473000C94730074
:100040000C9426010C9473000C9473000C947300B0
:100050000C9473000C9473000C9473000C94730054
:100060000C9473000C947300000000240027001F
:100070002A0000000000250028002B0000000000DE
:10008000230026002900040404040404040202DA
:10009000020202020303030303030301020408102007
:1000A000408001020408102001020408102000012
:1000B000000800020100000304070000000000027
:1000C000000011241FBECFEFD8E0DEBFCDBF21E07E
:1000D000A0E0B1E001C01D92AF30B207E1F70E948D
:1000E00070010C9484020C940000833081F028F499
:1000F000813099F08230A9F008958730A9F08830D6
:10010000C9F08430B1F4809180008F7D03C080916C
:1001100080008F7780938000089584B58F7784BDA9
:10012000089584B58F7DFBCF8091B0008F77809349
:10013000B00008958091B0008F7DF9CFCF93DF9309
:10014000282F30E0F901E255FF4F8491F901E6567E
:10015000FF4FD491F901EA57FF4FC491CC23A1F08E
:1001600081110E947500EC2FF0E0EE0FFF1FE458A4
:10017000FF4FA591B491EC91ED2381E090E009F45B
:1001800080E0DF91CF91089580E090E0FACF1F9357
:10019000CF93DF93282F30E0F901E255FF4F849190
:1001A000F901E656FF4FD491F901EA57FF4FC49188
:1001B000CC23A9F0162F81110E947500EC2FF0E0DE
:1001C000EE0FFF1FEE58FF4FA591B4918FB7F89433
:1001D000EC91111108C0D095DE23DC938FBFDF9125
:1001E000CF911F910895DE2BF8CFCF93DF9390E04E
:1001F000FC01E656FF4F24918A579F4FFC018491E2
:100200008823D1F090E0880F991FFC01E859FF4F37
:10021000A591B491FC01EE58FF4FC591D4916111A5
:100220000EC09FB7F8948C91E22FE0958E238C93AB
:100230002881E223E8839FBFDF91CF9108958FB794
:10024000F894EC91E22BEC938FBFF6CF1F920F92B4
:100250000FB60F9211242F933F938F939F93AF93D9
:10026000BF9380910B0190910C01A0910D01B09171
:100270000E0130910A0123E0230F2D3758F5019626
:10028000A11DB11D20930A0180930B0190930C01D5
```

:10029000A0930D01B0930E0180910601909107018A
:1002A000A0910801B09109010196A11DB11D809393
:1002B000060190930701A0930801B0930901BF9133
:1002C000AF919F918F913F912F910F900FBE0F9003
:1002D0001F90189526E8230F0296A11DB11DD2CFBD
:1002E000789484B5826084BD84B5816084BD85B511
:1002F000826085BD85B5816085BD80916E0081601D
:1003000080936E0010928100809181008260809 3C2
:10031000810080918100816080938100809180 00C4
:1003200081608093800080 91B10084608093B100EF
:100330008091B00081608093B00080917A008460E9
:1003400080937A0080917A00826080937A00809115
:100350007A00816080937A0080917A00806880932F
:100360007A001092C10060E082E00E94F50060E037
:1003700083E00E94F50060E084E00E94F50060E008
:1003800085E00E94F50061E08AE00E94F50061E0EE
:100390008BE00E94F50061E08CE00E94F50061E0D6
:1003A0008DE00E94F50080E0C82E80E0D82E82E02B
:1003B0000E949E007C0190930501809304018 3E0DC
:1003C0000E949E008C019093030180930201 84E0BF
:1003D0000E949E00EC01909301018093000185E052
:1003E0000E949E00E114F10409F058C001151105A6
:1003F00009F5209781F4009751F460E08AE00E94AB
:10040000C70060E08BE00E94C70061E023C0019755
:1004100071F561E0F3CF219751F5009709F070C0B5
:1004200060E08AE00E94C70061E08BE00E94C700A4
:1004300061E026C001301105D1F42097A1F40097A6
:1004400071F461E08AE00E94C70061E08BE00E94E5
:10045000C70060E08CE00E94C70061E015C0019712
:1004600031F460E0EFCF219711F4892B51F360E074
:100470008AE00E94C70060E08BE00E94C70060E055
:100480008CE00E94C70060E08DE00E94C700C114AC
:10049000D10409F48CCF0E94000089CFEA94EF28A0
:1004A00031F701151105D1F4209761F4009711F48B
:1004B00061E0DECF0197D9F661E08AE00E94C700D3
:1004C00061E0DACF219799F6009759F2019779F612
:1004D00060E08AE00E94C70061E094CF013011051E
:1004E00031F6209741F4009759F460E08AE00E94C9
:1004F000C70060E09ACF219709F0B9CF009741F388
:0C050000019709F0B4CF79CFF894FFCF39 :00000001FF


New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
LED-RED
RESISTOR
SIMULINO UNO
SW-SPDT
SWITCH
SIM1
SIMULINO UNO
Edit Component
Part Reference: SIM1
Hidden:
Part Value: SIMULINO UNO
Hidden:
Element:
New
OK
Hidden Pins
Edit Firmware
Cancel
blogembarcado.blogspot.com: (Default)
Program File: D:\sketch_may16a.ino.hex
RSTDISBL (Disable reset): (1) Unprogrammed
WDTON (Enable watchdog): (1) Unprogrammed
BOOTRST (Select Reset Vector): (1) Unprogrammed
CKSEL Fuses: (0000) Ext. Clock
Boot Loader Size: (00) 1024 words. Starts at 0x1C00
SUT Fuses: (10)
Clock Frequence: 16MHz
CLKDIV8 (Divide clock by 8): (1) Unprogrammed
CKOUT (Clock output): (1) Unprogrammed
NAME: SIMULINO UNO
Advanced Properties:
Disassemble Binary Code
No
Hide All
Other Properties:
Exclude from Simulation
Exclude from PCB Layout
Exclude from Bill of Materials
Attach hierarchy module
Hide common pins
Edit all properties as text

8. Hardware Implementation

8.1 Figure 1:



Here we can see the circuit, for the inputs are I3=0, I2=0, I1=0, I0=0 the LED outputs are O4=1, O3=1, O2=0, O1=0

9. Questions and answers:

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

Ans : The Clock frequency of the microcontroller used is 16 MHz

* **What is the data bus width of the microcontroller**

used ? Ans: The width of the microcontroller's data bus is 8

bit

*** What is the size of your hex file generated ?**

Ans: The generated hex file is 4 KB in size and has been used in the project.

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

Ans: Yes, interrupt can be used to implement the project. It is mainly used by application systems to support hardware clocks, data transportation and communication devices.

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

Ans: Yes, the main routine had to run in an infinite loop, as the value was not updated when it is not looped simultaneously.

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

Ans: Yes, the main difference between edge and level triggering is that the yield of consecutive circuits changes between high and low voltage cycles in terms of edge activating.

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

Ans: The project refers to encryption from input to output as the main motive of encryption is to convert a message by changing it into a code. The project accomplished this by encryption.

***Exclusively comment on the input-output pair that will draw the maximum power assuming a single power supply is used for the system?**

Ans: The input output pair (1101 - 1011) and (0111 - 1011) will draw the maximum power . We got to this result by the following procedure.

Input : When the input is high (1) . The Current flow through the external resistor is equal to VCC /R . And the Battery power consumption is equal to VCC*current .

22 When the input is low (0) . The Current flow through the external resistor is 20 micro amperes . And the Battery power consumption is equal to VCC*20 micro amperes .

In Our Input Circuit the VCC is 3.3 V and the Resistor R is 10Kohm.

So Using this we calculated all the input power consumption.

Output : When the input is high (1) . The Current flow through the external resistor is equal to VCC /R . And the Battery power consumption is equal to VCC*current .

When the input is low (0) . The Current flow through the external resistor is 0 amperes . And the Battery power consumption is equal to 0 (Zero) .

In Our output Circuit the VCC is 3.3 V and the Resistor R is 220ohm.

So Using this we calculated all the output power consumption.

The Full Power Consumption table of input and output is given below

POWER CONSUMPTION TABLE				
Input	Output	Input Power(IP)	Output Power(OP)	Power Consumption
0000	0101	0.000268W	0.099W	0.0993W
1000	1100	0.0013W	0.099W	0.1003W
0100	1110	0.0013W	0.1485W	0.1498W
1100	0001	0.00231W	0.0495W	0.0518W
0010	0111	0.0013W	0.1485W	0.1498W
1010	0010	0.00231W	0.0495W	0.0518W
0110	0110	0.00231W	0.099W	0.1013W
1110	0100	0.0034	0.0495W	0.0529W
0001	1100	0.0013W	0.099W	0.1013W
1001	0000	0.00231W	0	0.0023W
0101	0110	0.00231W	0.099W	0.1013W
1101	1011	0.0034W	0.1485W	0.1518W
0011	0000	0.00231W	0	0.0023W
1011	0011	0.0034W	0.099W	0.1023W
0111	1011	0.0034W	0.1485W	0.1518W
1111	0011	0.00436W	0.099W	0.1033W

10. References:

Adding arduino library : https://projecthub.arduino.cc/

Loading hex file in proteus environment :
https://www.instructables.com/How-to-Simulate-Arduino-in-Proteus/

Consumption :
https://maker.pro/arduino/projects/how-to-simulate-arduino-projects-using-proteus