Lab Report-2

(Sanchit jalan,Group-3,2022101070,Table No:-40)

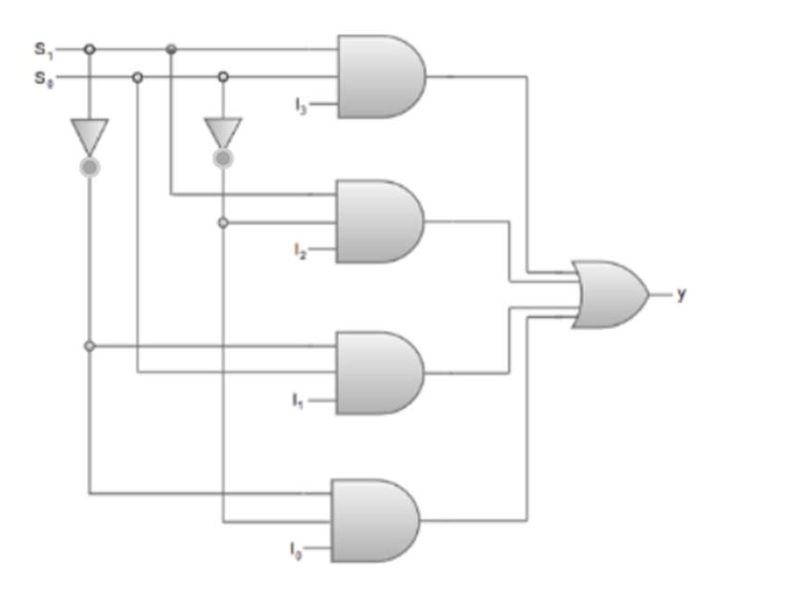
**Experiement 3:-**

Part A:- Designing 4:1 Multiplexer using basic logic gates

Electronic components required:-

1. Digital test kit
2. Wires
3. 3 input AND gate (7411 IC)
4. NOT gate (7404)
5. OR gate (7432)
6. Arduino UNO

Logic gate circuit:-



Procedure :-

1. Connect the Arduino pins of 5V and GND to the breadboard
2. Take the appropriate ICs and connect S0,S1,I0,I1,I2,I3 appropriately as shown in the logic gate diagram..
3. In my tinkerCAD simulation :-

|  |  |
| --- | --- |
| INPUT PIN | VALUE |
| 8 | I0 |
| 9 | I1 |
| 10 | I2 |
| 11 | I3 |
| 12 | S0 |
| 13 | S1 |

1. Connect the final output of OR gate to an LED ..
2. Write the appropriate arduino code and observe for different combinations of inputs and select lines ..

Arduino code:-

int a,b,c,d,e,f;

void setup()

{

Serial.begin(9600);

pinMode(13, OUTPUT);

pinMode(12, OUTPUT);

pinMode(11, OUTPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

pinMode(8, OUTPUT);

Serial.println("Inputs in order I0,I1,I2,I3,S0,S1");

}

void loop()

{

if(Serial.available())

{

a=Serial.parseInt();

Serial.println(a);

b=Serial.parseInt();

Serial.println(b);

c=Serial.parseInt();

Serial.println(c);

d=Serial.parseInt();

Serial.println(d);

e=Serial.parseInt();

Serial.println(e);

f=Serial.parseInt();

Serial.println(f);

if(a) digitalWrite(8,HIGH);else digitalWrite(8,LOW);

if(b) digitalWrite(9,HIGH);else digitalWrite(9,LOW);

if(c) digitalWrite(10,HIGH);else digitalWrite(10,LOW);

if(d) digitalWrite(11,HIGH);else digitalWrite(11,LOW);

if(e) digitalWrite(12,HIGH);else digitalWrite(12,LOW);

if(f) digitalWrite(13,HIGH);else digitalWrite(13,LOW);

delay(100);

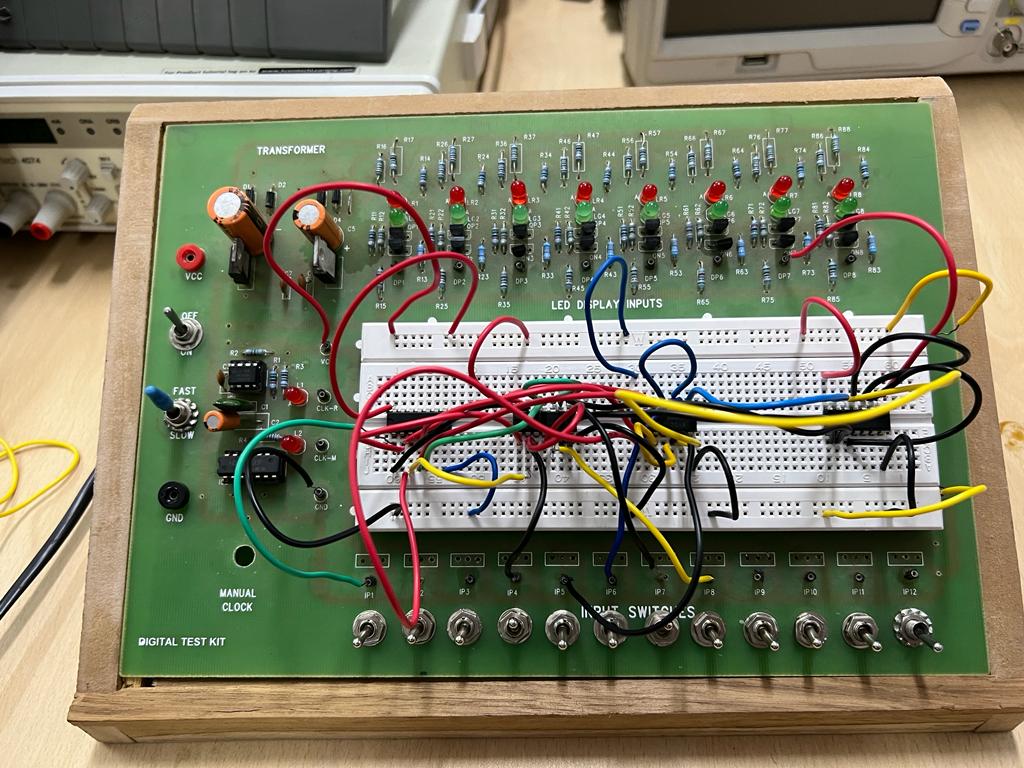
}

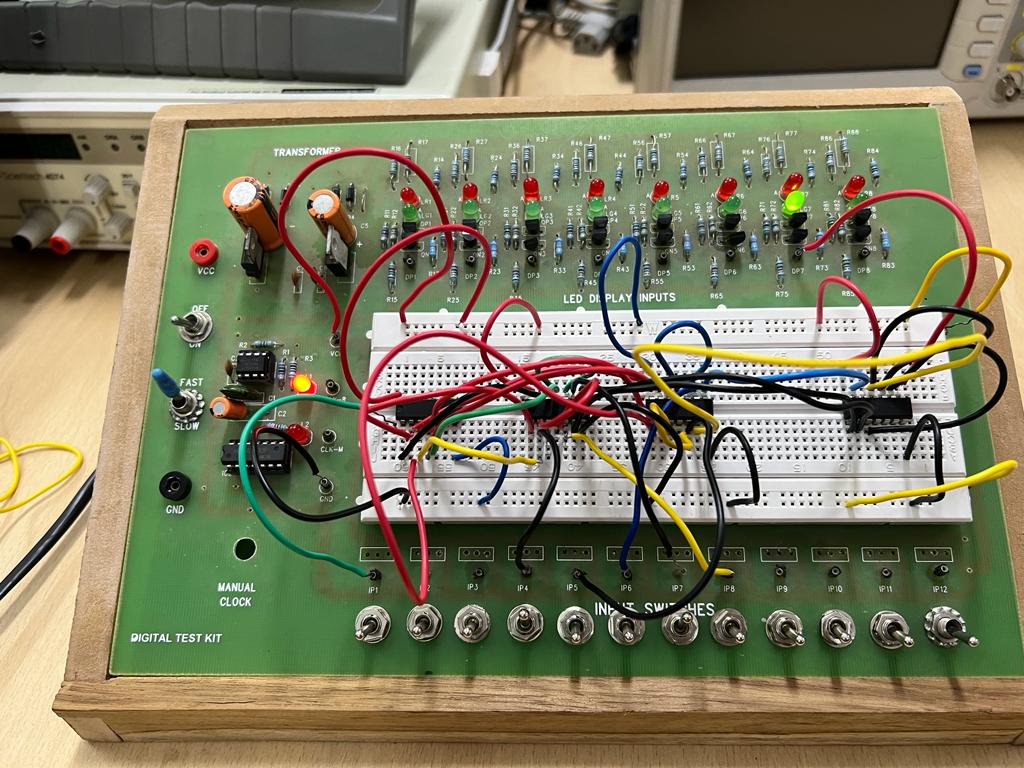
}

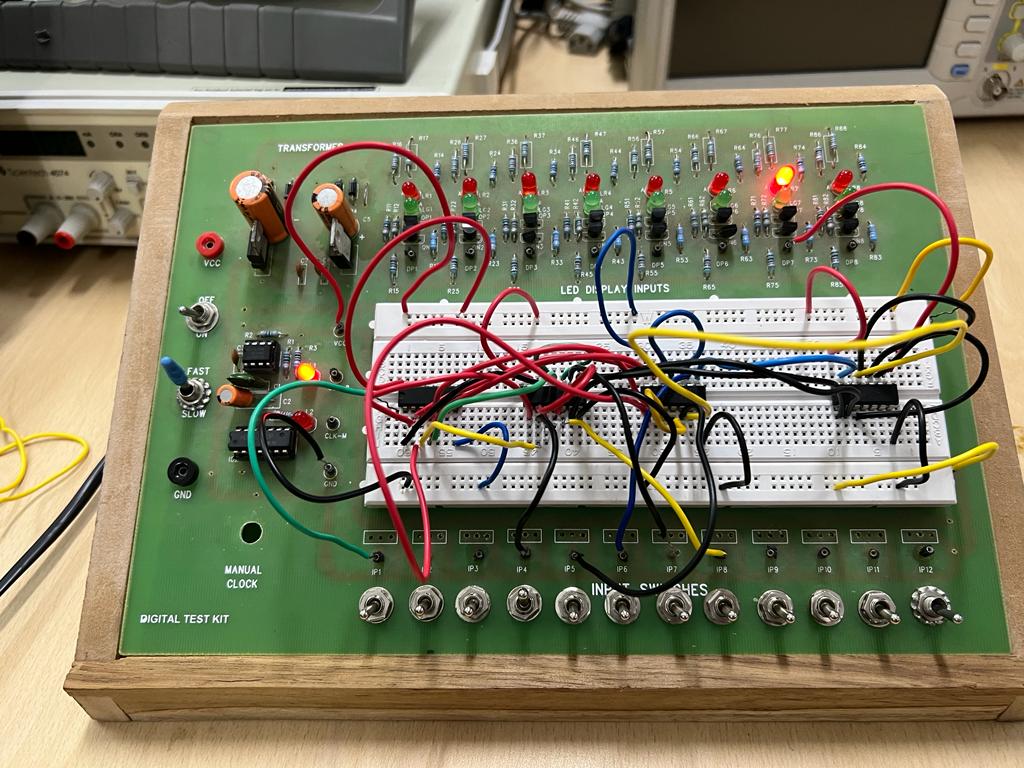
Observation :-

|  |  |  |
| --- | --- | --- |
| S0 | S1 | Y |
| 0 | 0 | I0 |
| 0 | 1 | I2 |
| 1 | 0 | I1 |
| 1 | 1 | I3 |

Output(Y) = I0 (S1)’(S0)’+I1 (S1)’S0+I2 S1(S0)’+I3 S1S0







Conclusion :-

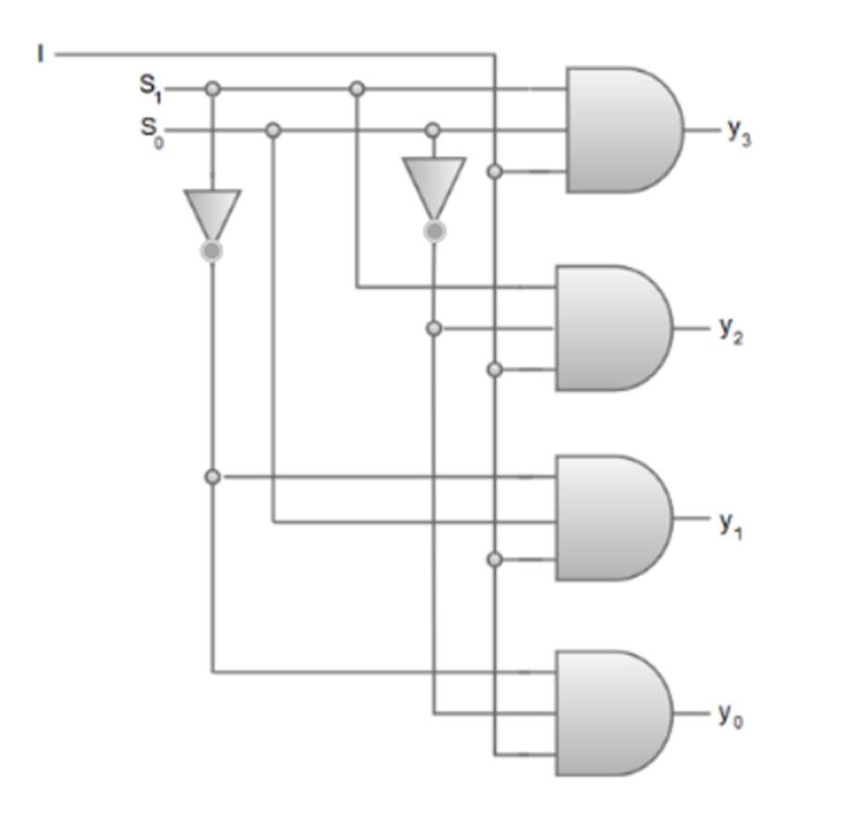
A (4:1) Multiplexer can be made in this way using combination of ORs ANDs gate and NOT gate .. We can get desired output using the combination of minterms which is successfully depicted in truth table and circuit diagram ..

PART B:- Designing a (1:4) DeMultiplexer using basic logic gates ..

Electric components required :-

1. Arduino UNO
2. Digital Test kit
3. Not gate (7404)
4. Triple input AND gate(7411)
5. Wires

Logic gate circuit:-



Procedure :-

1. Connect the Arduino 5V and GND to breadboard.
2. Take inputs of I,S0,S1 from Arduino and connect them appropriately to the ICs (7404 and 7411) as shown in the diagram
3. In my tinkerCAD simulation :-

|  |  |
| --- | --- |
| INPUT PIN | VALUE |
| 11 | I |
| 12 | S0 |
| 13 | S1 |

1. Connect the outputs of 4 3 input AND gates to 4 LEDs respectively..
2. Write the appropriate Arduino Code and observe for different combinations of select lines.

Arduino Code:-



Observations:-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S1 | S0 | Y3 | Y2 | Y1 | Y0 |
| 0 | 0 | 0 | 0 | 0 | i |
| 0 | 1 | 0 | 0 | i | 0 |
| 1 | 0 | 0 | i | 0 | 0 |
| 1 | 1 | i | 0 | 0 | 0 |

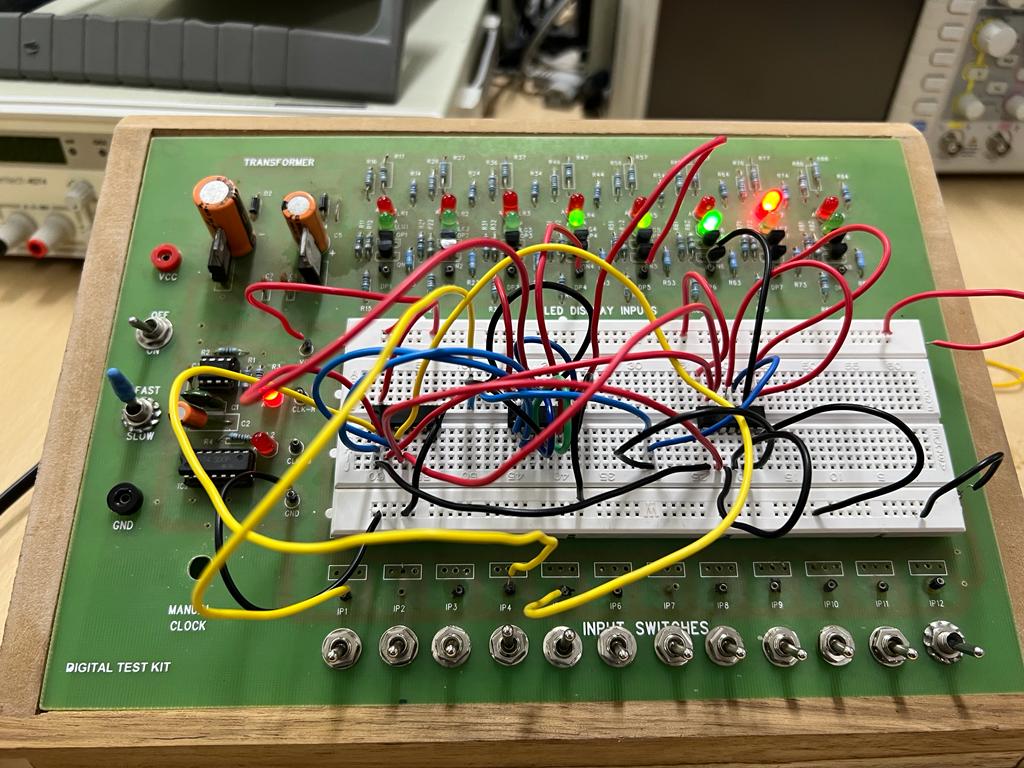
From truth table we can write output expressions as:-

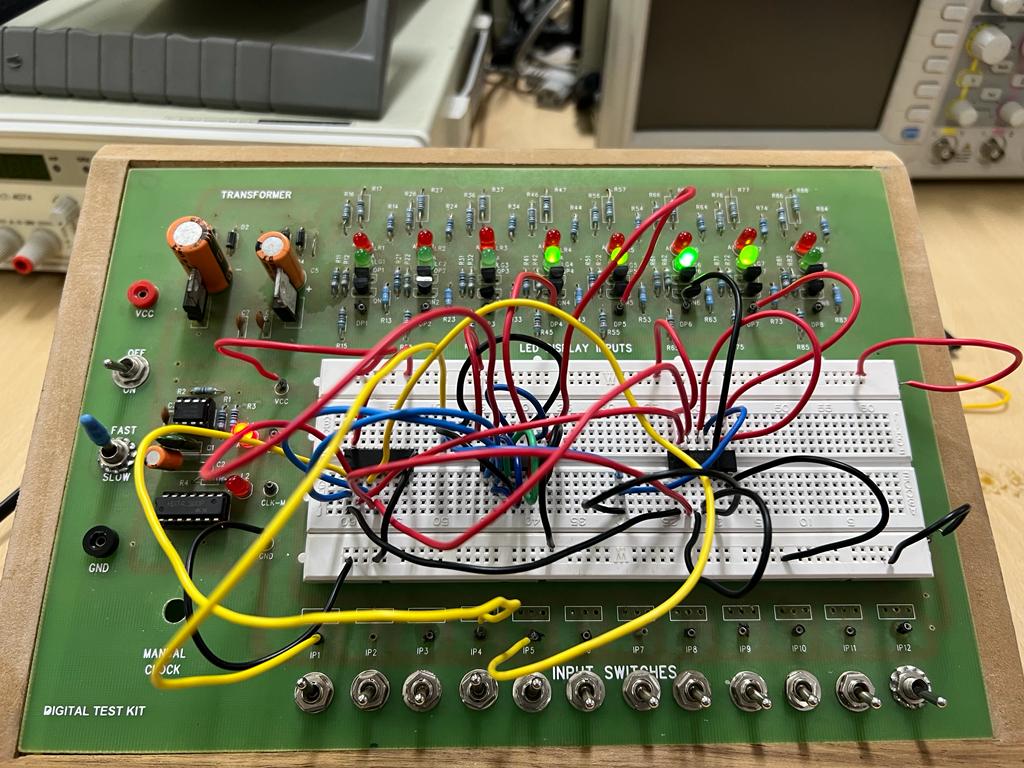
y0 = i (S1)’ (S0)’

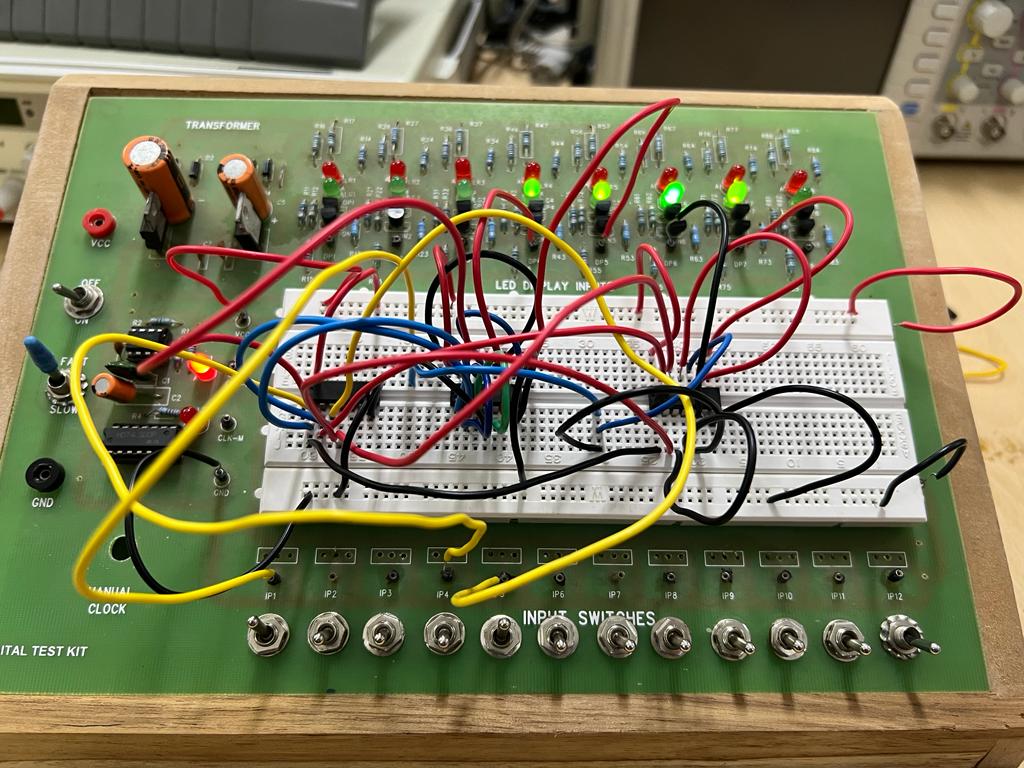
y1 = i (S1)’ S0

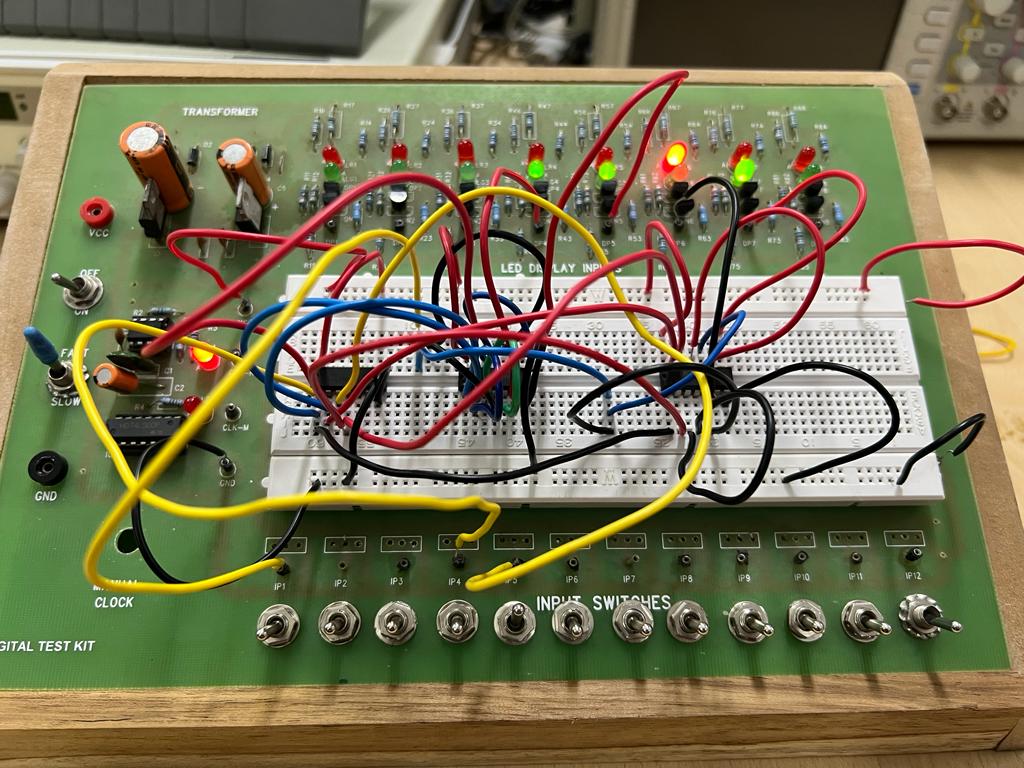
y2 = i S1 (S0)’

y3 = i S1 S0









Conclusions:-

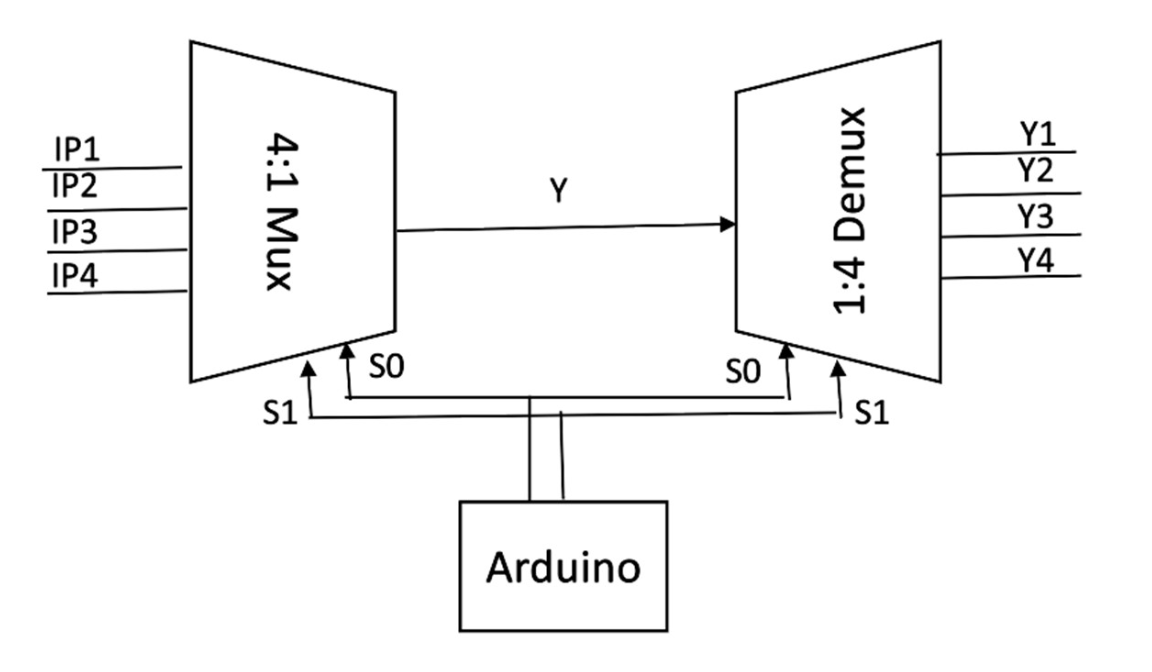
We can make a (1:4) Demultiplexer using the above combination of AND and NOTs gate …

PART C:- Assemble and test circuits made in PART A and B

Electronic components required:-

1. MUX circuit designed in Part A
2. DEMUX circuit designed in Part B
3. Breadboard

Block diagram:-



Procedure:-

1. Connect the 5V and GND of Arduino to MUX and DEMUX.
2. Connect 4 pins of Arduino as input to MUX .
3. Select lines of both MUX and DEMUX should be same .
4. Connect the output of MUX to input of DEMUX
5. In my tinkerCAD simulation :-

|  |  |
| --- | --- |
| INPUT PIN | VALUE |
| 8 | I0 |
| 9 | I1 |
| 10 | I2 |
| 11 | I3 |
| 12 | S0 |
| 13 | S1 |

1. Connect the 4 outputs of DEMUX to 4 LEDs..
2. Write appropriate Arduino Code to power up MUX

Arduino Code:-

int a,b,c,d,e,f;

void setup()

{

Serial.begin(9600);

pinMode(13, OUTPUT);

pinMode(12, OUTPUT);

pinMode(11, OUTPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

pinMode(8, OUTPUT);

Serial.println("Inputs in order I0,I1,I2,I3,S0,S1");

}

void loop()

{

if(Serial.available())

{

a=Serial.parseInt();

Serial.println(a);

b=Serial.parseInt();

Serial.println(b);

c=Serial.parseInt();

Serial.println(c);

d=Serial.parseInt();

Serial.println(d);

e=Serial.parseInt();

Serial.println(e);

f=Serial.parseInt();

Serial.println(f);

if(a) digitalWrite(8,HIGH);else digitalWrite(8,LOW);

if(b) digitalWrite(9,HIGH);else digitalWrite(9,LOW);

if(c) digitalWrite(10,HIGH);else digitalWrite(10,LOW);

if(d) digitalWrite(11,HIGH);else digitalWrite(11,LOW);

if(e) digitalWrite(12,HIGH);else digitalWrite(12,LOW);

if(f) digitalWrite(13,HIGH);else digitalWrite(13,LOW);

delay(100);

}

}

Observations:-

|  |  |  |
| --- | --- | --- |
| S1 | S0 | Y |
| 0 | 0 | I0 |
| 0 | 1 | I1 |
| 1 | 0 | I2 |
| 1 | 1 | I3 |

Output table of MUX

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S1 | S0 | Y3 | Y2 | Y1 | Y0 |
| 0 | 0 | 0 | 0 | 0 | i |
| 0 | 1 | 0 | 0 | i | 0 |
| 1 | 0 | 0 | i | 0 | 0 |
| 1 | 1 | i | 0 | 0 | 0 |

Output table of DEMUX

Conclusion:-

In this way we can connect MUX and DEMUX together and verify the working of both ….