Lab Report: 3

Name: Atharv Bhatt
Roll number: 2023101089
Group number: 3

Experiment 3: Multiplexing and Demultiplexing

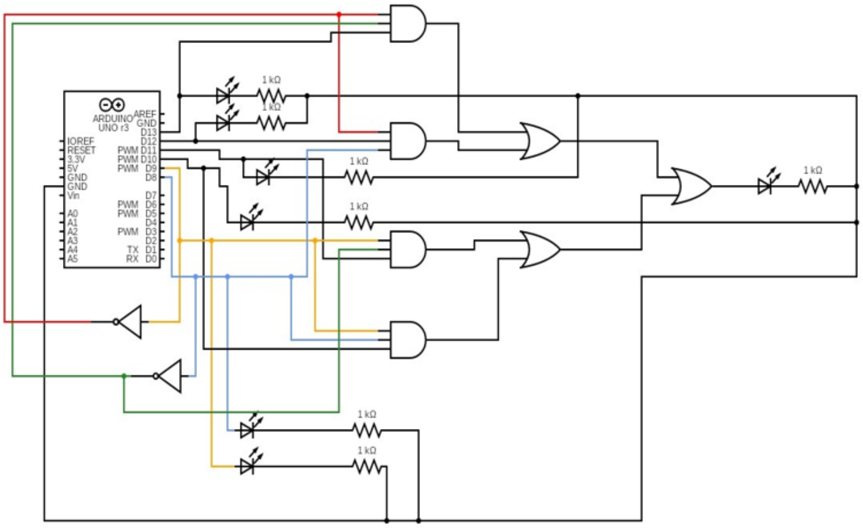
Part A: Design of 4:1 Multiplexer (MUX)

Aim/Objective: The objective of this experiment is to design, assemble, and test a (1:4) Multiplexer (MUX) using basic logic gates controlled by an Arduino.

Electronic Components Used:

1. Breadboard
2. 74HC04 Inverter IC
3. 74HC11 AND Gate ICs (2)
4. 74HC32 OR Gate ICs (2)
5. Arduino
6. Input Switches
7. LED Displays
8. Jumper Wires

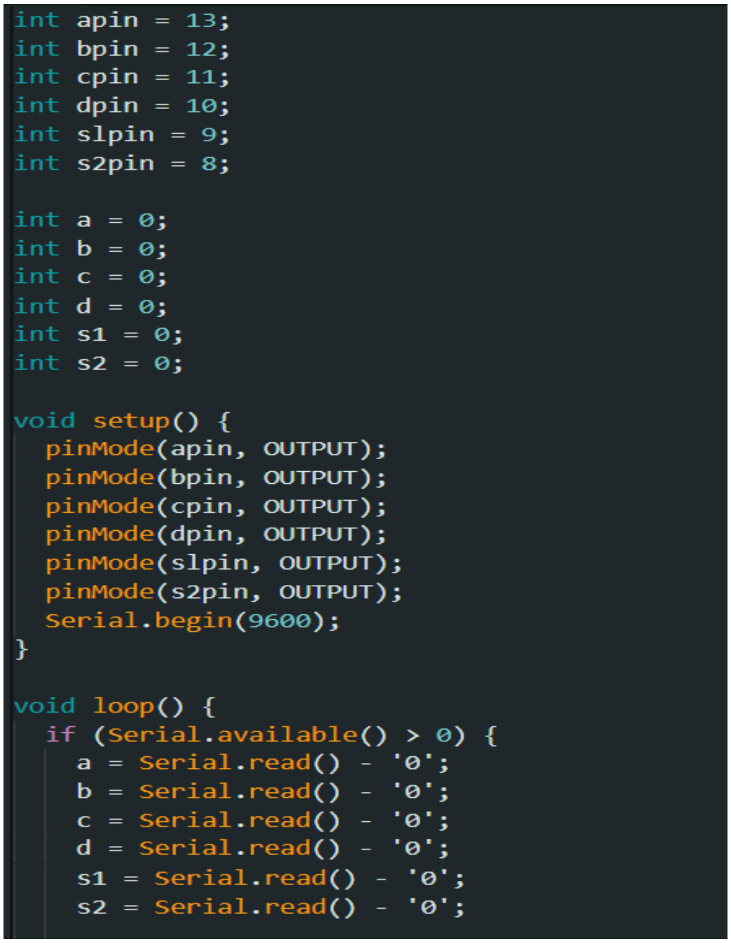
Reference Circuit



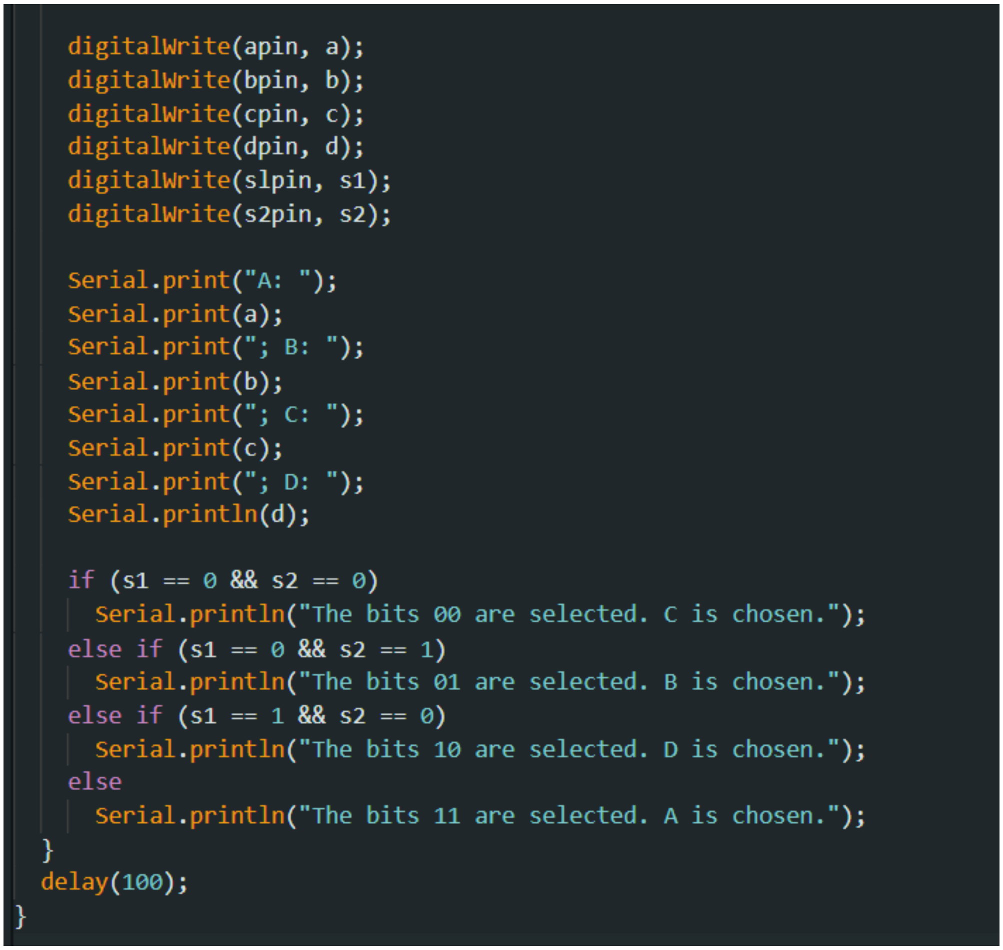
Procedure

1. A 4:1 MUX has four data input lines (I0 to I3), two select lines (S0 and S1), and a single output line (Y).
2. The truth table for a 4:1 MUX determines which input line is connected to the output line based on the select lines.
3. The logic circuit of the 4:1 MUX is implemented using an inverter IC (74HC04), two AND gate ICs (74HC11), and two OR gate ICs (74HC32).
4. The Arduino code is written to control the input combinations and observe the output on an LED.
5. The MUX function is verified by tabulating the output values for all input combinations.

Code



```
int apin = 13;
int bpin = 12;
int cpin = 11;
int dpin = 10;
int slpin = 9;
int s2pin = 8;

int a = 0;
int b = 0;
int c = 0;
int d = 0;
int s1 = 0;
int s2 = 0;

void setup() {
  pinMode(apin, OUTPUT);
  pinMode(bpin, OUTPUT);
  pinMode(cpin, OUTPUT);
  pinMode(dpin, OUTPUT);
  pinMode(slpin, OUTPUT);
  pinMode(s2pin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    a = Serial.read() - '0';
    b = Serial.read() - '0';
    c = Serial.read() - '0';
    d = Serial.read() - '0';
    s1 = Serial.read() - '0';
    s2 = Serial.read() - '0';
```

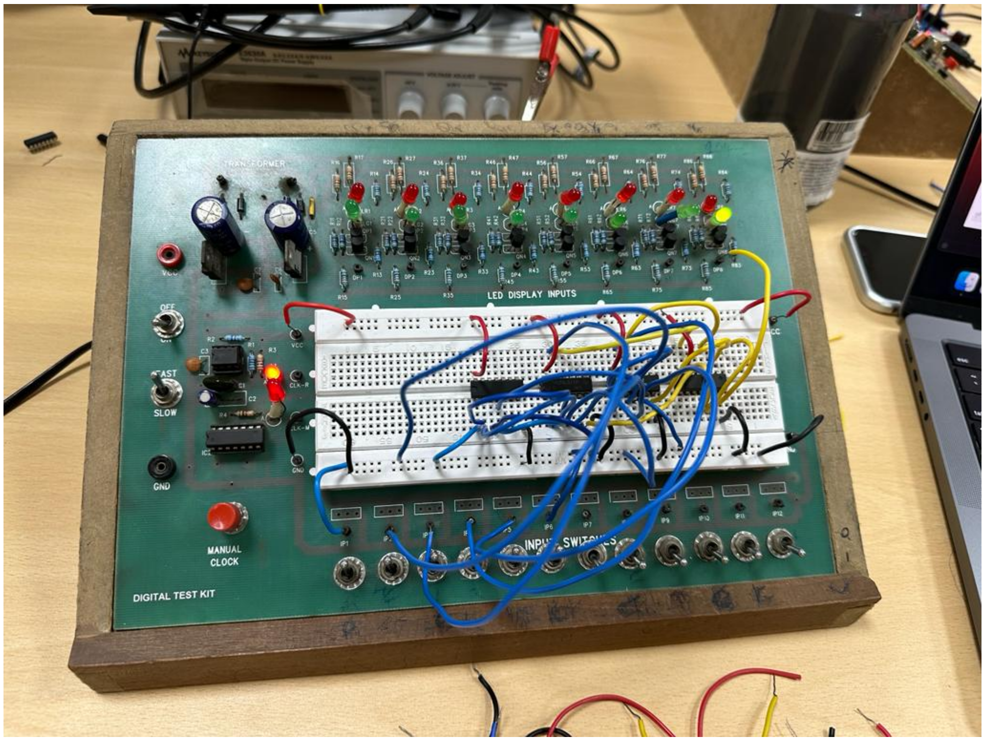


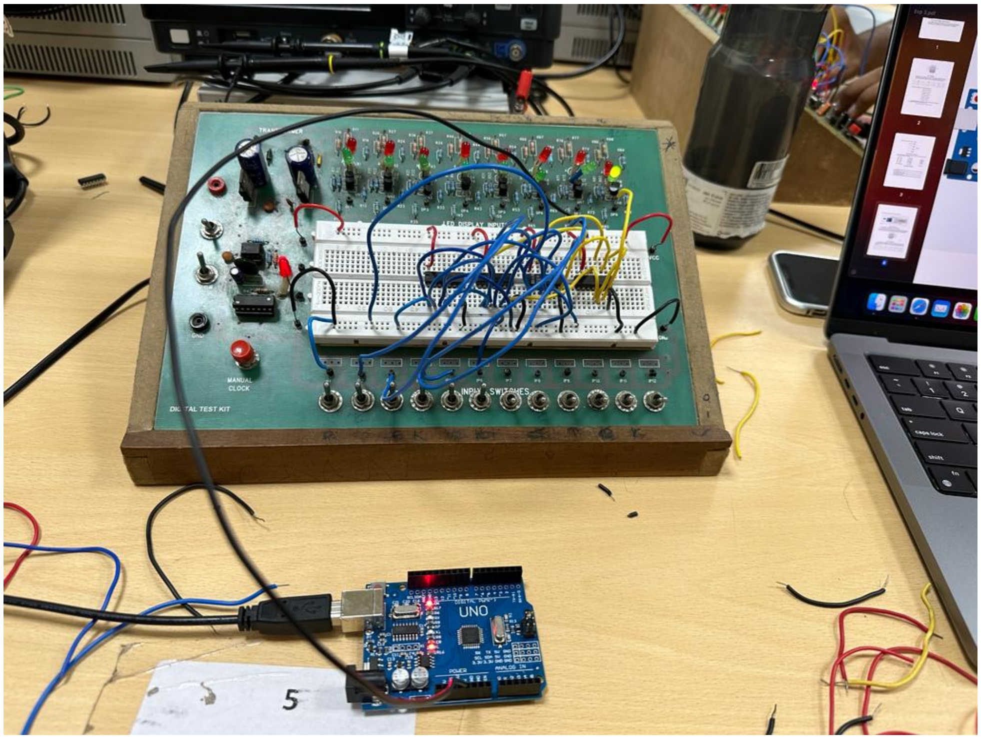
```
    digitalWrite(apin, a);
    digitalWrite(bpin, b);
    digitalWrite(cpin, c);
    digitalWrite(dpin, d);
    digitalWrite(slpin, s1);
    digitalWrite(s2pin, s2);

    Serial.print("A: ");
    Serial.print(a);
    Serial.print("; B: ");
    Serial.print(b);
    Serial.print("; C: ");
    Serial.print(c);
    Serial.print("; D: ");
    Serial.println(d);

    if (s1 == 0 && s2 == 0)
      Serial.println("The bits 00 are selected. C is chosen.");
    else if (s1 == 0 && s2 == 1)
      Serial.println("The bits 01 are selected. B is chosen.");
    else if (s1 == 1 && s2 == 0)
      Serial.println("The bits 10 are selected. D is chosen.");
    else
      Serial.println("The bits 11 are selected. A is chosen.");
  }
  delay(100);
}
```

Tinkercad

https://www.tinkercad.com/things/dku2kk0MCIc/editel?returnTo=%2Fclassrooms%2F65dBoBuNJYW%2Factivities%2F0vR1LpzK1ik&sharecode=BKuk1XP3IIy2S9Z6P97bmLORHauiiCWYstDSZLWGt4o

Images


TRANSFORMER
LED DISPLAY INPUTS
VCC
OFF
ON
FAST
SLOW
GND
MANUAL
CLOCK
INPUT SWITCHES
DIGITAL TEST KIT


LED DISPLAY INPUTS
MANUAL
CLOCK
INPUT SWITCHES
DIGITAL TEST KIT
UNO
5

Truth Table:

S1	S2	Y (Output)
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Conclusion

In this part of the experiment, a 4:1 Multiplexer (MUX) was designed and implemented using basic logic gates. The MUX function was successfully verified through Arduino-controlled input combinations and LED output observations. The logical expressions for the MUX were derived, and the truth table was validated, ensuring the proper operation of the 4:1 MUX circuit.

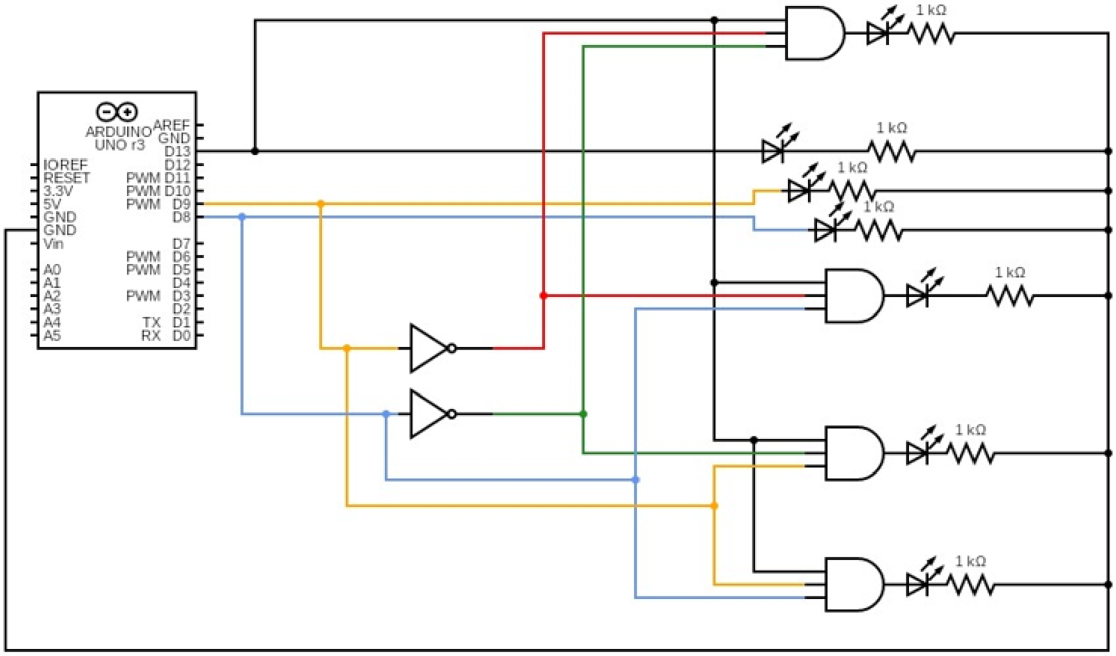
Part B: Design of 1:4 Demultiplexer (DEMUX)

Aim / Objective: To design a (1:4) Demultiplexer using basic logic gates.

Electronic Components Used:

1. Arduino Uno R3
2. Hex Inverter
3. Triple 3-Input AND gate
4. Digital Test Kit

Reference Circuit

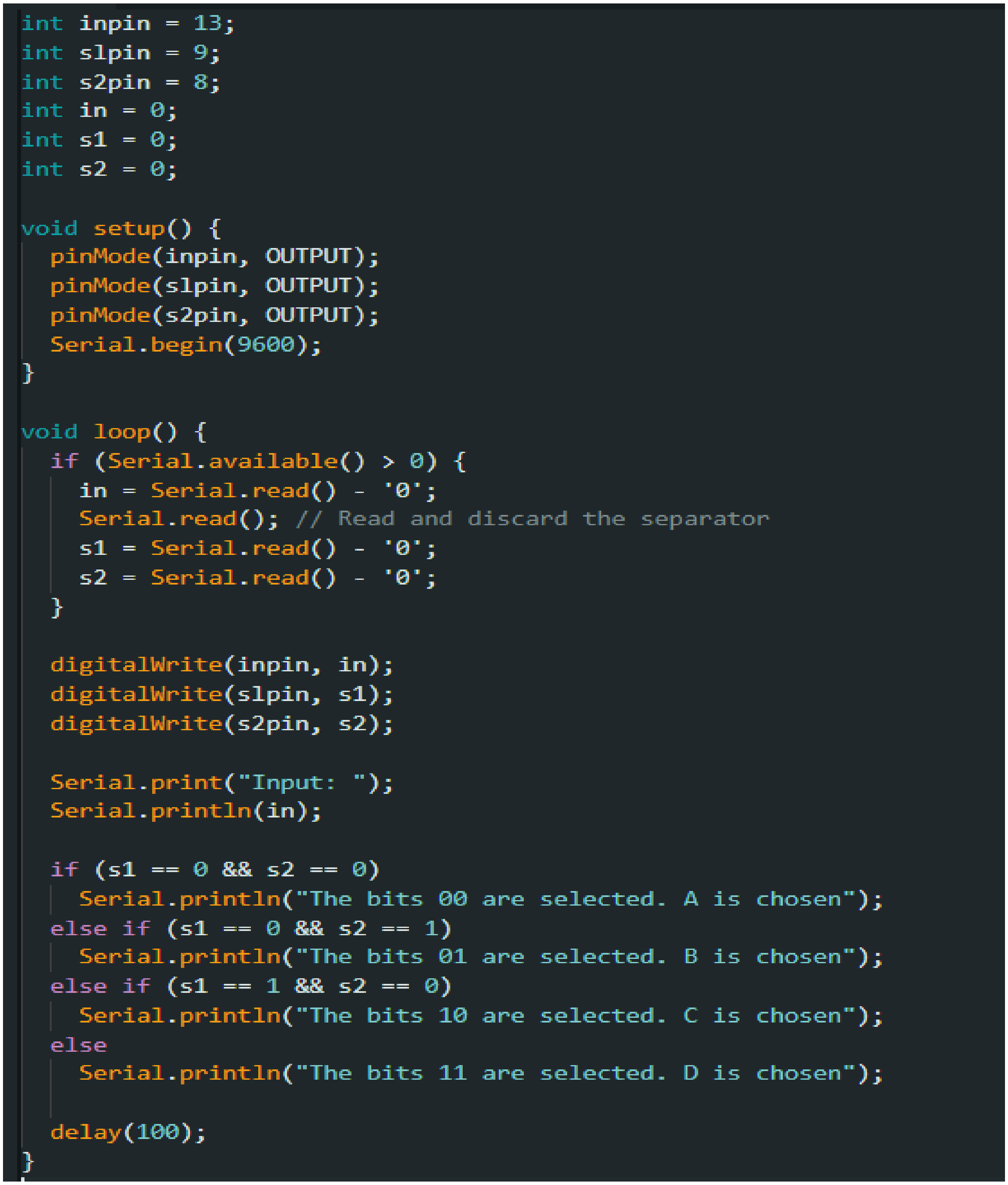


Procedure

1. A 1:4 DEMUX has one data input line (i), two select lines (S0 and S1), and four output lines (y0, y1, y2, y3).
2. The truth table for a 1:4 DEMUX determines which output line receives the input based on the select lines.
3. The logic circuit of the 1:4 DEMUX is implemented using AND gate ICs (74HC11) and an inverter IC (74HC04).

4. The Arduino code is written to control the input and select combinations and observe the output on LEDs.
5. The DEMUX function is verified by tabulating the output values for all input combinations.

<u>Code</u>



```
int inpin = 13;
int slpin = 9;
int s2pin = 8;
int in = 0;
int s1 = 0;
int s2 = 0;

void setup() {
  pinMode(inpin, OUTPUT);
  pinMode(slpin, OUTPUT);
  pinMode(s2pin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    in = Serial.read() - '0';
    Serial.read(); // Read and discard the separator
    s1 = Serial.read() - '0';
    s2 = Serial.read() - '0';
  }

  digitalWrite(inpin, in);
  digitalWrite(slpin, s1);
  digitalWrite(s2pin, s2);

  Serial.print("Input: ");
  Serial.println(in);

  if (s1 == 0 && s2 == 0)
    Serial.println("The bits 00 are selected. A is chosen");
  else if (s1 == 0 && s2 == 1)
    Serial.println("The bits 01 are selected. B is chosen");
  else if (s1 == 1 && s2 == 0)
    Serial.println("The bits 10 are selected. C is chosen");
  else
    Serial.println("The bits 11 are selected. D is chosen");

  delay(100);
}
```

Tinkercad

https://www.tinkercad.com/things/1V1lEjCWBNl/editel?returnTo=%2Fclassrooms%2F65dBoBuNJYW%2Factivities%2FhdfznnLdskP&sharecode=uyfBxObSUnOaJey3kSEYqDroV7yyKbbm5Bu9unAMU24

Truth Table

S1	S2	Y0	Y1	Y2	Y3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

Conclusion

In this part of the experiment, a 1:4 Demultiplexer (DEMUX) was designed and constructed using basic logic gates. The DEMUX function was effectively verified through Arduino-controlled input and select line combinations, along with LED output observations. The logical expressions for the DEMUX were derived, and the truth table was confirmed, establishing the correct functioning of the 1:4 DEMUX circuit.

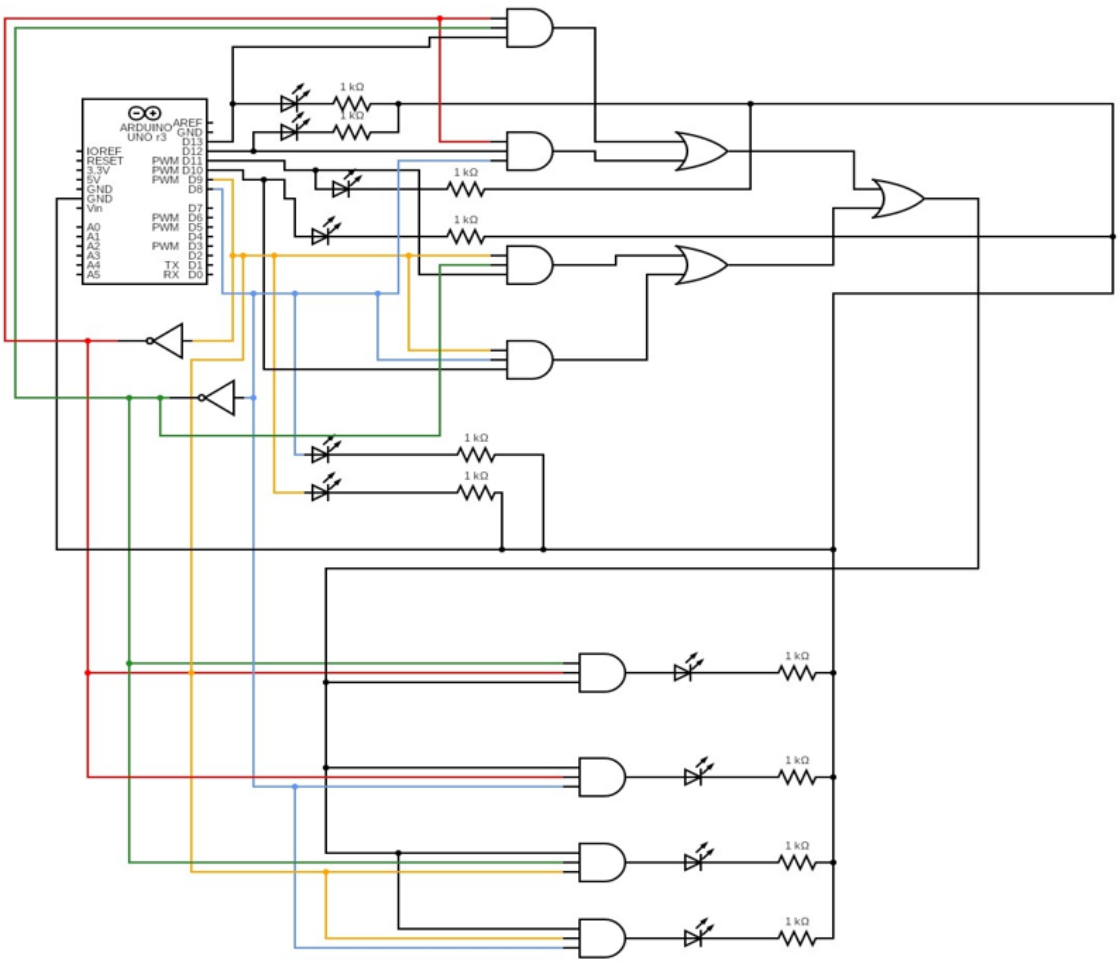
Part C: Assembling and Testing the Circuits

Aim/Objective: To assemble and test the (4:1) multiplexer and (1:4) demultiplexer.

Electronic Components Used:

1. Arduino Uno R3
2. Hex Inverter
3. Triple 3-Input AND gate
4. Quad OR gate
5. Digital Test Kit

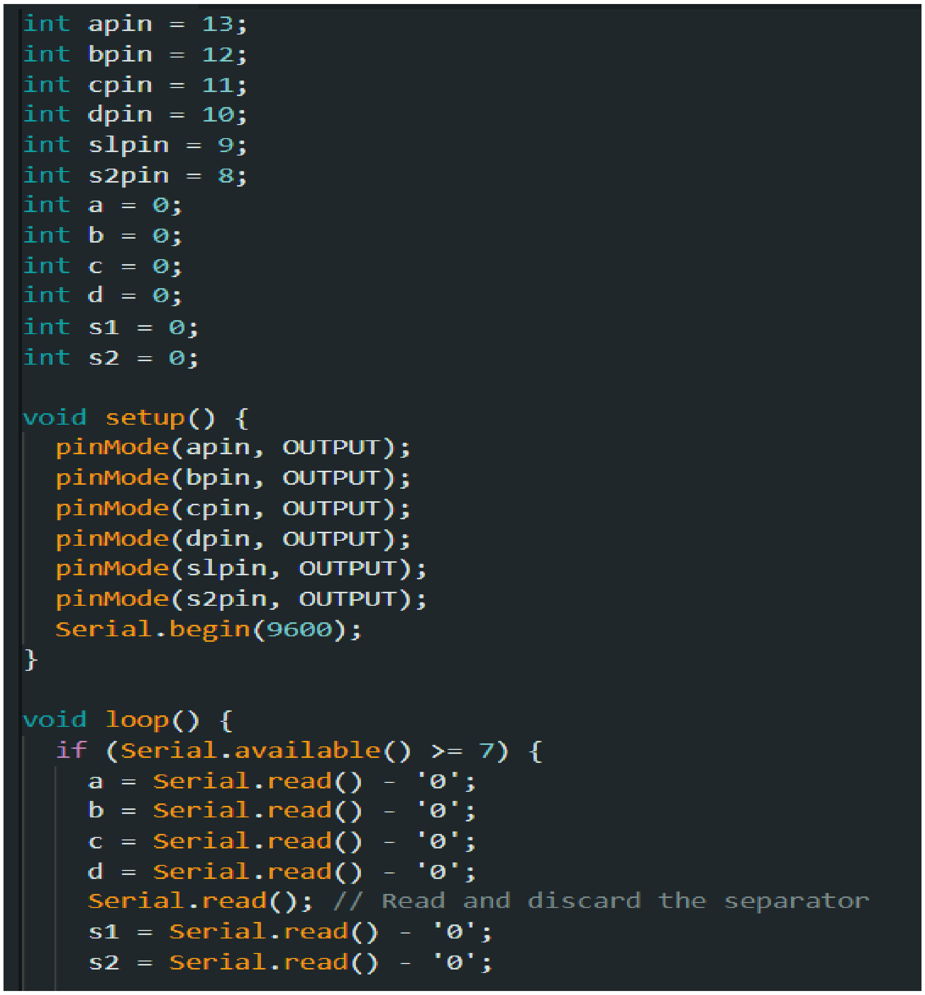
Reference Circuit



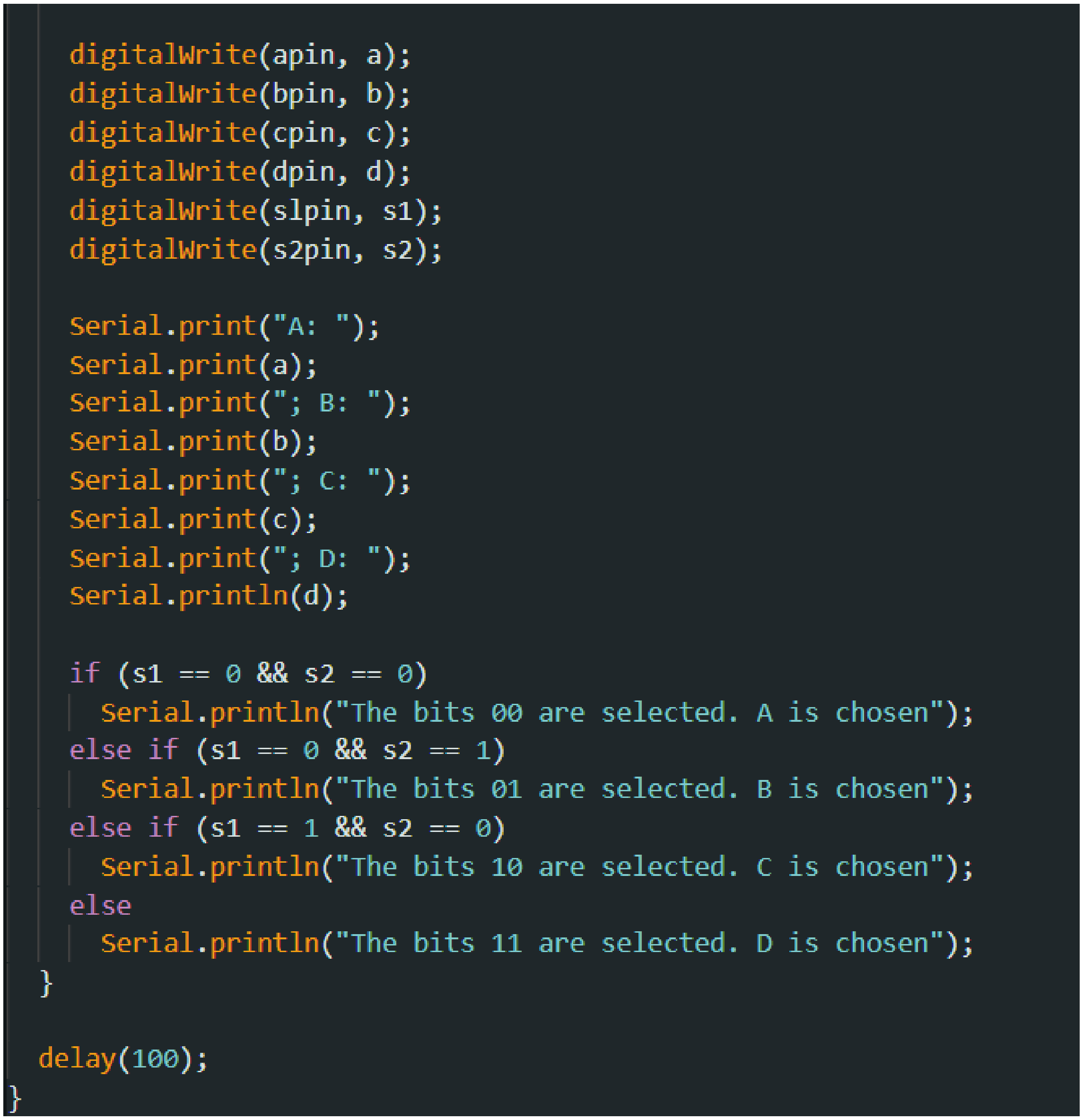
Procedure

1. A program is written to control the MUX input combinations and observe the DEMUX output using LEDs.
2. VCC and GND of both MUX and DEMUX are connected to the 5V and GND pins of the Arduino.
3. Four Arduino pins are connected to the MUX as input pins, and the MUX output is connected to the DEMUX input.
4. The select lines of both the MUX and DEMUX are connected to the Arduino.
5. The output of the DEMUX is connected to four LEDs.
6. The program controls the inputs and select lines, and the output is observed on the LEDs.
7. The block diagram and Tinkercad circuit diagram are used for reference during circuit assembly and testing.

Code



```
int apin = 13;
int bpin = 12;
int cpin = 11;
int dpin = 10;
int slpin = 9;
int s2pin = 8;
int a = 0;
int b = 0;
int c = 0;
int d = 0;
int s1 = 0;
int s2 = 0;

void setup() {
  pinMode(apin, OUTPUT);
  pinMode(bpin, OUTPUT);
  pinMode(cpin, OUTPUT);
  pinMode(dpin, OUTPUT);
  pinMode(slpin, OUTPUT);
  pinMode(s2pin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() >= 7) {
    a = Serial.read() - '0';
    b = Serial.read() - '0';
    c = Serial.read() - '0';
    d = Serial.read() - '0';
    Serial.read(); // Read and discard the separator
    s1 = Serial.read() - '0';
    s2 = Serial.read() - '0';
```



```
    digitalWrite(apin, a);
    digitalWrite(bpin, b);
    digitalWrite(cpin, c);
    digitalWrite(dpin, d);
    digitalWrite(slpin, s1);
    digitalWrite(s2pin, s2);

    Serial.print("A: ");
    Serial.print(a);
    Serial.print("; B: ");
    Serial.print(b);
    Serial.print("; C: ");
    Serial.print(c);
    Serial.print("; D: ");
    Serial.println(d);

    if (s1 == 0 && s2 == 0)
      Serial.println("The bits 00 are selected. A is chosen");
    else if (s1 == 0 && s2 == 1)
      Serial.println("The bits 01 are selected. B is chosen");
    else if (s1 == 1 && s2 == 0)
      Serial.println("The bits 10 are selected. C is chosen");
    else
      Serial.println("The bits 11 are selected. D is chosen");
  }

  delay(100);
}
```

Tinkercad

https://www.tinkercad.com/things/3mO7ngHCUNa/editel?returnTo=%2Fclassrooms%2F65dBoBuNJYW%2Factivities%2FhdfznnLdskP&sharecode=WOKHNuemf1MeSqGrTpNDay1j_4SaTaJgPTG27jNXTsQ

Truth Table

S1	S2	Y0	Y1	Y2	Y3	Y4
0	0	I0	0	0	0	0
0	1	I1	0	I1	0	0
1	0	I2	0	0	I2	0
1	1	I3	0	0	0	I3

Conclusion

In this segment of the experiment, both the 4:1 Multiplexer (MUX) and the 1:4 Demultiplexer (DEMUX) circuits were assembled and tested together. An Arduino program was used to control the input combinations of the MUX and observe the output of the DEMUX via LEDs. The successful integration and testing of both circuits demonstrated their functionality as a combined system, showcasing the practical application of multiplexing and demultiplexing in digital systems.