Lab Report: 4

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

Experiment 4: Programmable ALU

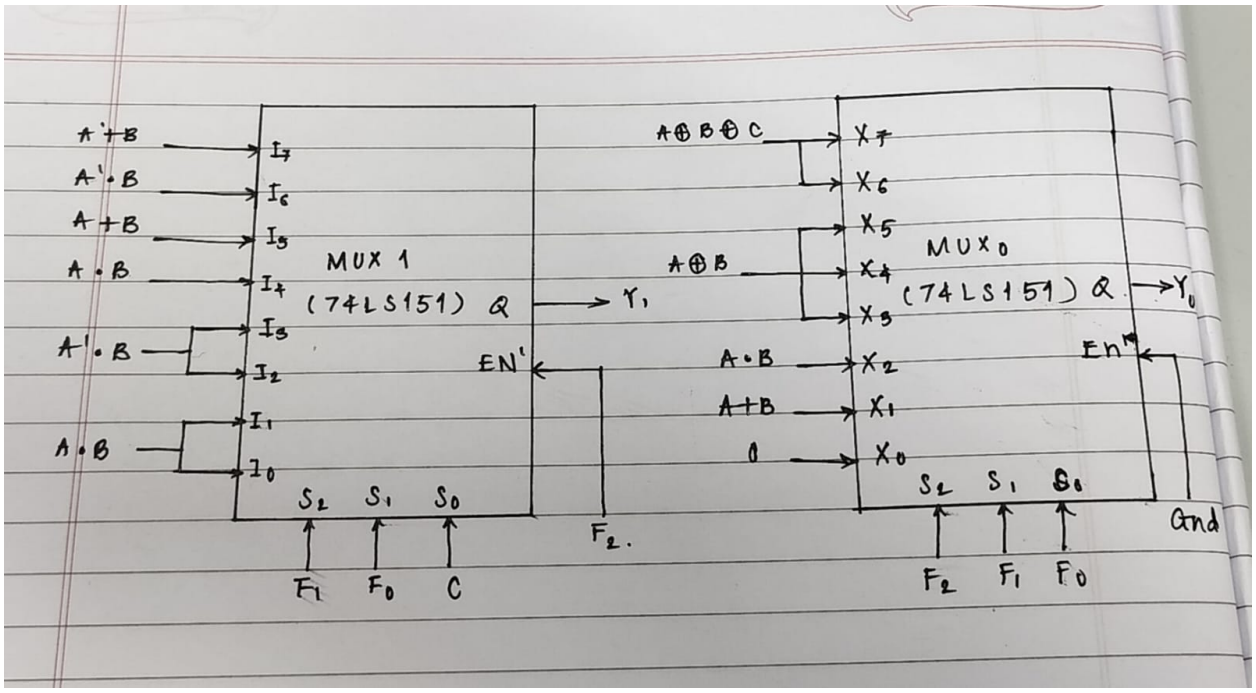
Aim/Objective: The objective of this experiment is to design and implement a programmable arithmetic logic unit capable of performing 8 arithmetic or Logical operations, which are:

1) 0 (Zero)
2) A+B (OR)
3) A • B (AND)
4) A ⊕ B (XOR)
5) A+B (SUM)
6) A-B (DIFFERENCE)
7) A+B+C (SUM)
8) A-B-C (DIFFERENCE)

Electronic Components Used:

1. Breadboard
2. quad 2 input multiplexers 74ls157
3. 1 8 input multiplexer 74LS151)
4. 1 quad 2 input XOR gates(74LS86)
5. Arduino UNO
6. LED Displays
7. Jumper Wires

Reference Circuit



Procedure

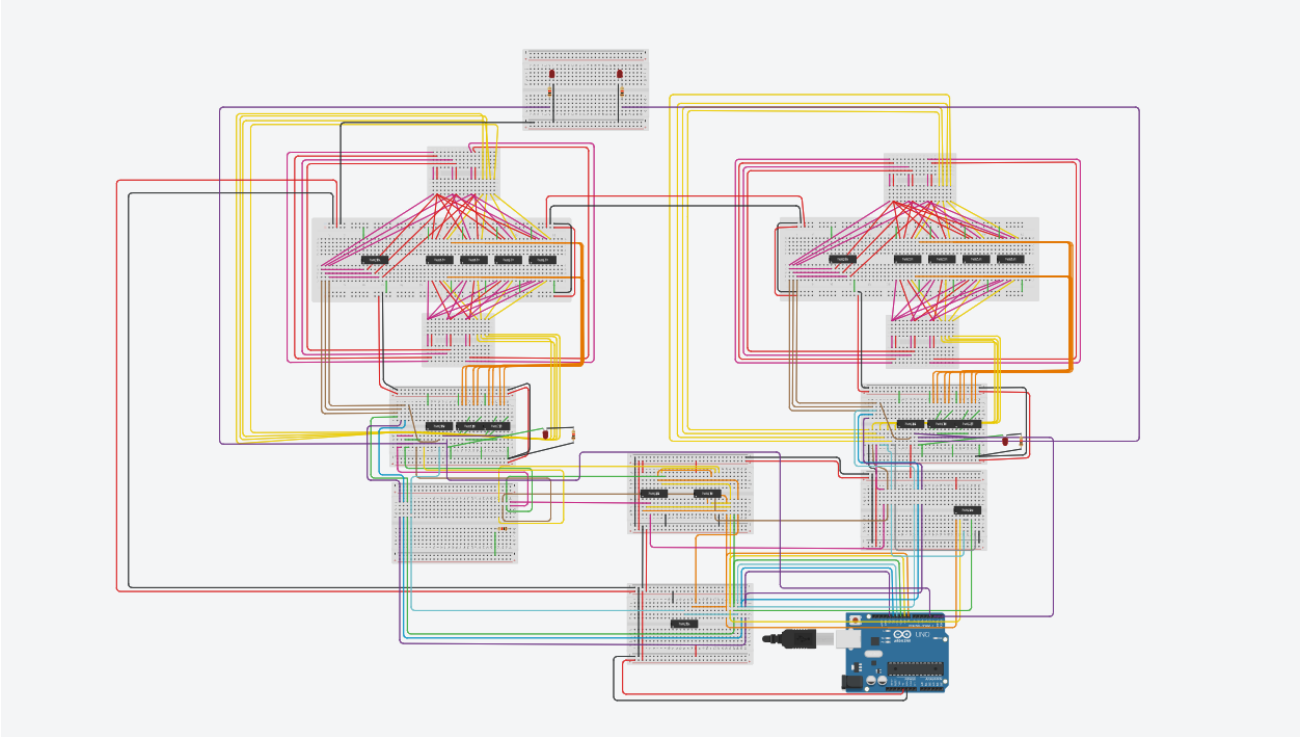
1. Power the breadboard using 5V and ground pins of Arduino.
2. Test all the given ICs using the Gate verification experiment procedure.
3. Provide all the inputs from Arduino: A, B, C, F0, F1, and F2.
4. You can make A' and B' using XOR Gate as A' = A ⊕ 1 and B' = B ⊕ 1.
5. Make the following connections using the two input mux. Connect the enable to ground.
   ➔ A • B
   ➔ A' • B
   ➔ A+B
   ➔ A'+B
6. Use the XOR Gate to make A ⊕ B and A ⊕ B ⊕ C.
7. Verify whether the above-made combinations are correct using Truth tables for respective operations and correct them if not.
8. Label the 8 input MUXes as MUX0 and MUX1 to avoid confusion while working.
9. Connect the enable of MUX0 to ground and the enable of MUX1 to F2'.
   *F2' = F2 ⊕ 1.
10. Make all the connections to MUX0 and MUX1 as shown in the reference circuit.
11. Code the Arduino, check for errors, and upload it to the board to give the inputs.
12. Try out all possible combinations of inputs and tabulate your observations.

Code

```
int a = 8;
int b = 9;
int c = 10;
int f0 = 13;
int f1 = 12;
int f2 = 11;
int out0 = 2;
int out1 = 3;
void setup()
{


  pinMode(out0, INPUT);
  pinMode(out1, INPUT);
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(f0, OUTPUT);
  pinMode(f1, OUTPUT);
  pinMode(f2, OUTPUT);
  Serial.begin(9600);

    Serial.print(" f2 | f1 | f0 | a | b | c |    function      | out1 | out0 |\n");
  Serial.print("-------------------------------------------------------------------\n");

  int s_f2,s_f1,s_f0,s_a,s_b,s_c,s_o0,s_o1;
  char buffer[60];
  for(int i = 0;i<64;i++){
    s_f2 = i/32;
    s_f1 = ((i-(s_f2*32))/16)%2;
    s_f0 = ((i-(s_f1*16))/8)%2;
    s_a = ((i-(s_f0*8))/4)%2;
    s_b = ((i-(s_a*4))/2)%2;
    s_c = i%2;

    digitalWrite(f2,s_f2);
    digitalWrite(f1,s_f1);
    digitalWrite(f0,s_f0);
    digitalWrite(a,s_a);
    digitalWrite(b,s_b);
    digitalWrite(c,s_c);

    s_o0 = digitalRead(out0);
    s_o1 = digitalRead(out1);
```

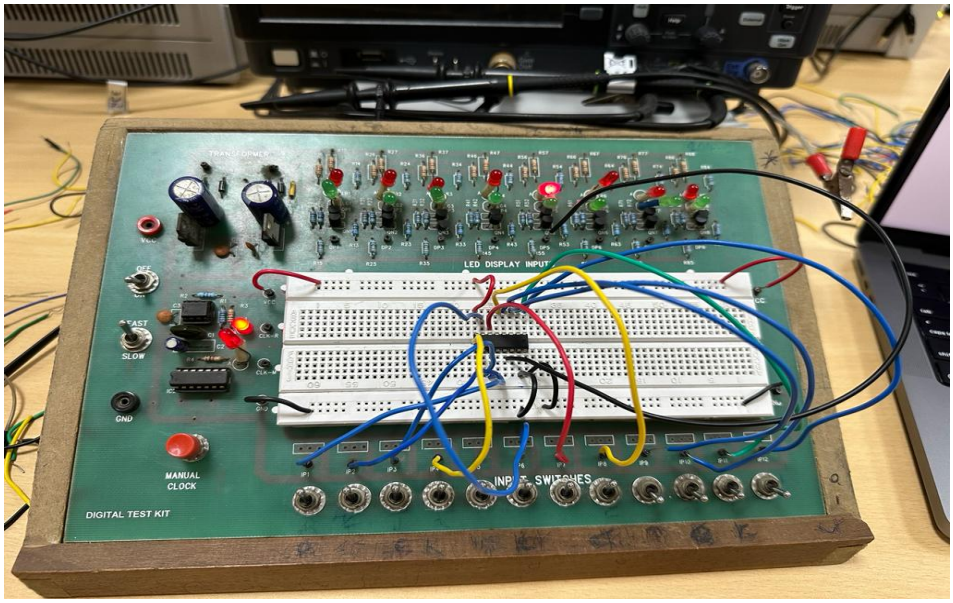
```
  int num = (1*s_f0)+(2*s_f1)+(4*s_f2);
  String func;
  switch (num)
  {
  case 0:
      func = "       ZERO      ";
      break;
  case 1:
      func = "      A OR B     ";
      break;
  case 2:
      func = "     A AND B     ";
      break;
  case 3:
      func = "     A EXOR B    ";
      break;
  case 4:
      func = "     A PLUS B    ";
      break;
  case 5:
      func = "    A MINUS B    ";
      break;
  case 6:
      func = " A PLUS B PLUS C ";
      break;
  case 7:
      func = "A MINUS B MINUS C";
      break;

  }

  snprintf(buffer,sizeof(buffer),"  %d |  %d |  %d | %d | %d | %d |%s|   %d  |   %d  |\n",s_f2,s_f1,s_f0,s_a,s_b,s_c,func.c_str(),s_o1,s_o0);
  Serial.println(buffer);
  delay(100);
 }

}

void loop()
{

}
```

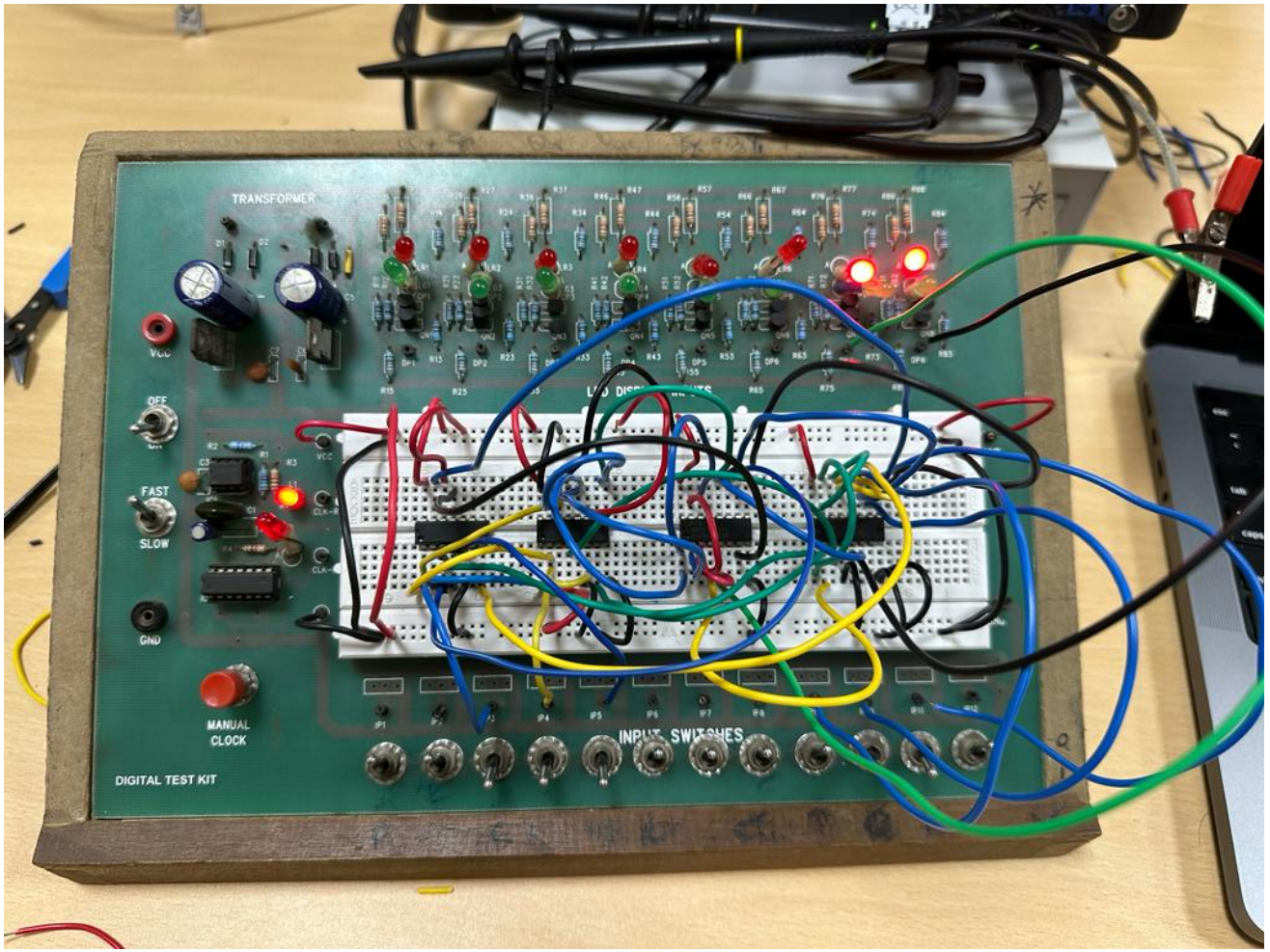
Tinkercad

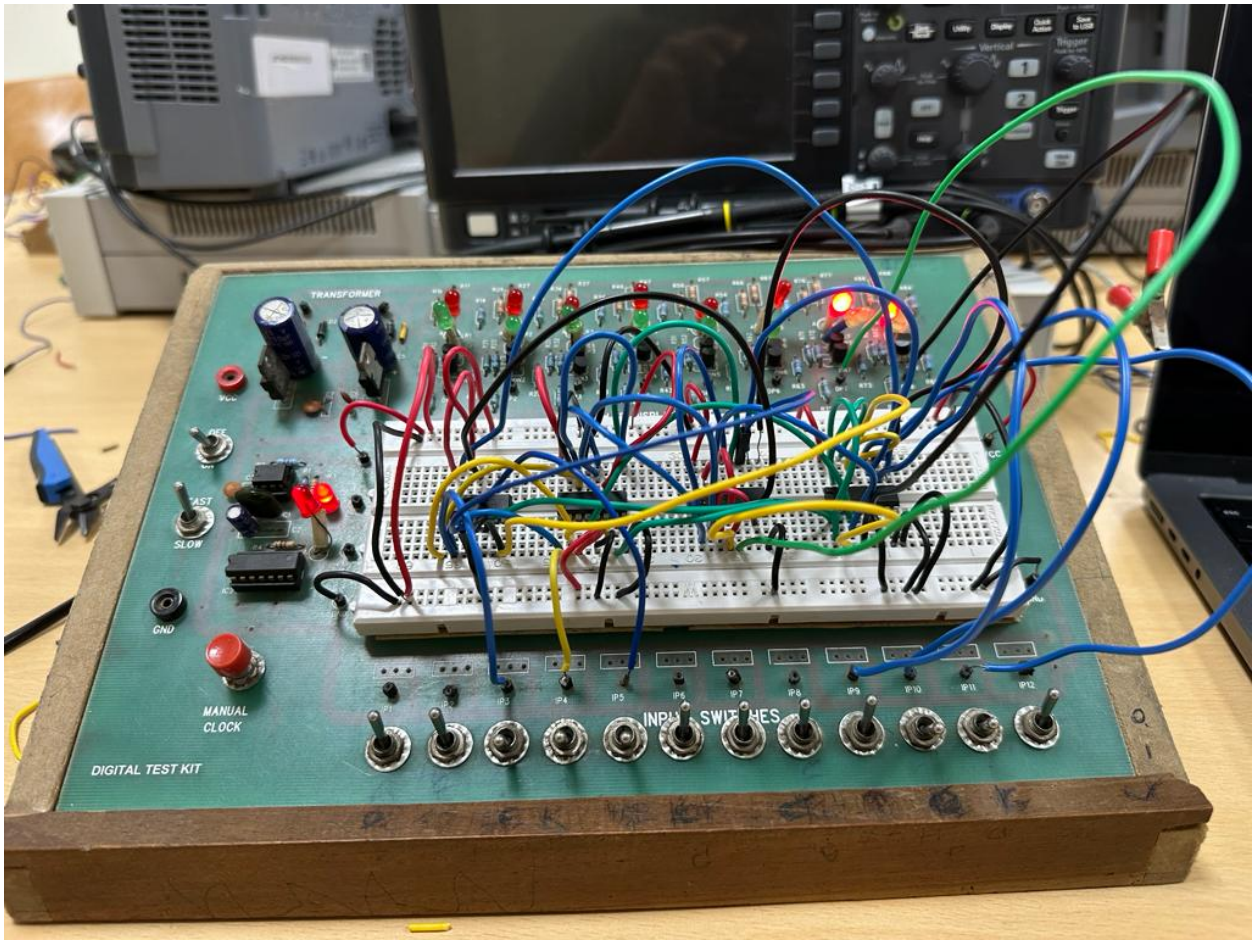


https://www.tinkercad.com/things/jefCyiQZAVq-lab-4-alu/editel?sharecode=o3ok6oxqhFtIrQmSFdIx4NnvyMyy7knlhebSQDtG3KU

Images




TRANSFORMER
VCC
OFF
FAST
SLOW
GND
MANUAL
CLOCK
DIGITAL TEST KIT
INPUT SWITCHES



Truth Tables:

When F2 = 0, F1 = 0, and F0 = 0

A	B	C	Y0 (0)
0	0	X	0
0	1	X	0
1	0	X	0
1	1	X	0

When F2 = 0, F1 = 0, and F0 = 1

A	**B**	**C**	**Y0 (A+B)**
0	0	X	0
0	1	X	1
1	0	X	1
1	1	X	1

When F2 = 0, F1 = 1, and F2 = 0

A	**B**	**C**	**Y0 (A . B)**
0	0	X	0
0	1	X	0
1	0	X	0
1	1	X	1

When F2 = 0, F1 = 1, and F0 = 1

A	**B**	**C**	**Y0 (A ⊕ B)**
0	0	X	0

0	1	X	1
1	0	X	1
1	1	X	0

When F2 = 1, F1 = 0, and F0 = 0

A	**B**	**C**	**Y0 (A PLUS B)**
0	0	X	0
0	1	X	1
1	0	X	1
1	1	X	0

When F2 = 1, F1 = 0, and F0 = 1

A	**B**	**C**	**A MINUS B**
0	0	X	0
1	0	X	1
0	1	X	1
1	1	X	0

When F2 = 1, F1 = 1, and F0 = 0

A	**B**	**C**	**Y0 (A PLUS B PLUS C) (SUM)**	**Y1 (CARRY)**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

When F2 = 1, F1 = 1, and F0 = 1,

A	**B**	**C**	**Y0 (A MINUS B MINUS C) (DIFFERENCE)**	**Y1 (BORROW)**
0	0	0	0	0
0	0	1	1	1

0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Conclusion

The final ALU outputs Y0 and Y1 were obtained from (8:1) Multiplexers MUX0 and MUX1 respectively. The output Y1 was only observed when F2 = 1. The outputs observed for different values of A, B, C, F0, F1, and F2 are given in the truth table of observations.