Lab Report:4

Programmable ALU

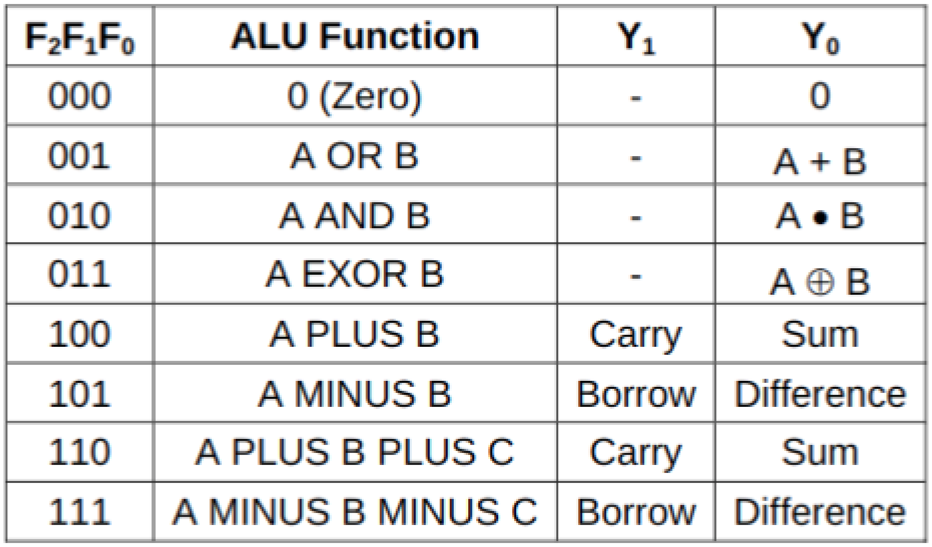
Name: Macharla Harish

Roll Number: 2020102062

Group Number: G4

Objective:

To design an Arithmetic and Logic Unit (ALU) capable of performing 8 Arithmetic/Logic functions on 1-bit operands.



$F_2F_1F_0$	ALU Function	Y_1	Y_0
000	0 (Zero)	-	0
001	A OR B	-	A + B
010	A AND B	-	A • B
011	A EXOR B	-	$A \oplus B$
100	A PLUS B	Carry	Sum
101	A MINUS B	Borrow	Difference
110	A PLUS B PLUS C	Carry	Sum
111	A MINUS B MINUS C	Borrow	Difference

Figure 1: ALU Function Table

Note that the first 4 functions are Logic functions generate 1-bit output Y0,

while the last four are Arithmetic functions generate 2-bit output Y1 Y0.

Equipment Required:

1) Logic AND gate

2) Logic OR gate

3) Logic XOR gate

4) logic Hex Inverter

5) Bread board

6) Arduino

7) Connecting wires, Resistors (1 kΩ)

8) Led’s

Reference Circuit:

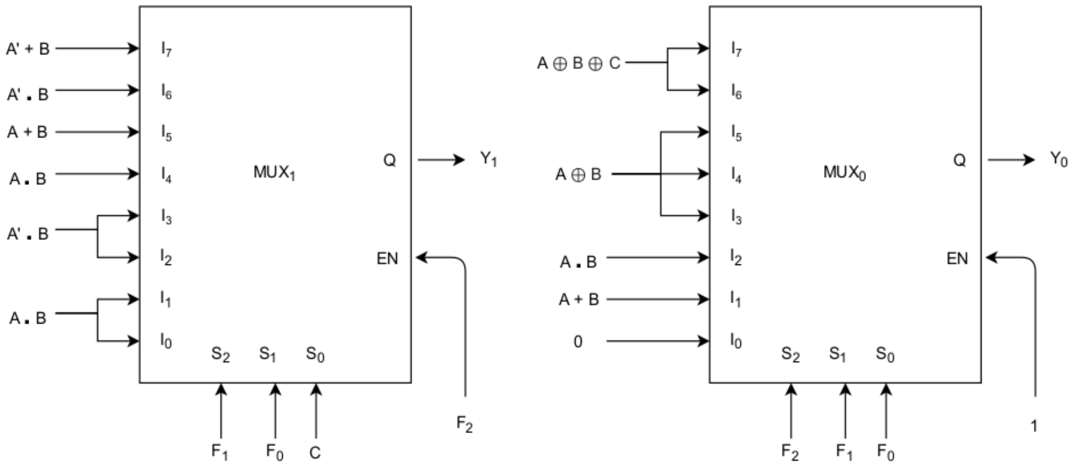


Figure 2: ALU Function Table

Procedure:

1.We need to get the bread board and Arduino and place them

2.Place the ICs needed on the Breadboard. Connect vcc and gnd .

3.Connect through the wires to get the Boolean expression we need

4. The final ALU output bits Y 0 and Y 1 will be generated by the two 8-input multiplexers – referred to as MUX 0 and MUX 1 respectively. The required data, select and output enable inputs of MUX 0 and MUX 1 are shown in Fig. 2.

5. Note that MUX 0 is always enabled, while MUX 1 is enabled only when F2= 1, i.e., for Arithmetic functions only. This is because Y1 is required only to provide the CARRY/BORROW output for Arithmetic functions.

6. Verify theoretically that MUX 0 and MUX 1 do generate the outputs Y0 and Y1 as required by Fig. 1.

7. Given a circuit with two 8:1 MUX, design the ALU according to the circuit diag. given in fig. 2.

8. Give F0, F1, F2, A, B, and C as input from an Arduino.

9. Apply all the combinations of the Function select inputs F2 F1 F0 one by one and tabulate the observed outputs Y 0 and Y 1 for as many combinations of the data inputs A, B, C as possible. Verify that the tabulated results conform to the ALU functions given in Fig. 1.

10) In order to work control the input of the mux Enable should be taken.

Code:

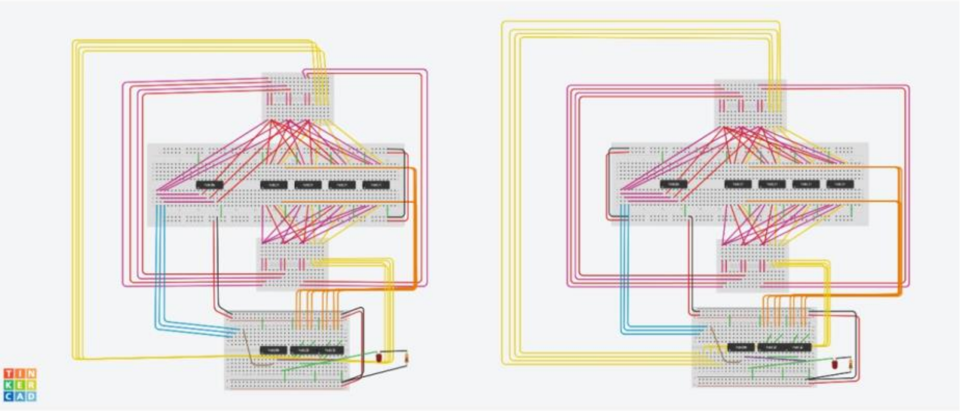
```
int A, B, C, F0, F1, F2;
void setup() {
  // put your setup code here, to run once:
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  Serial.print("----------------------");
  Serial.println();
  // put your main code here, to run repeatedly:
  if (Serial.available() > 0)
  {
    A = Serial.read();
    A = A - '0';
    B = Serial.read();
    B = B - '0';
    C = Serial.read();
    C = C - '0';
    F2 = Serial.read();
    F2 = F2 - '0';
    F1 = Serial.read();
    F1 = F1 - '0';
    F0 = Serial.read();
    F0 = F0 - '0';
  }
  digitalWrite(2, A);
  digitalWrite(3, B);
  digitalWrite(4, C);
  digitalWrite(5, F0);
  digitalWrite(6, F1);
  digitalWrite(7, F2);
  Serial.print("A = ");
  Serial.println(A);
  Serial.print("B = ");
  Serial.println(B);
  Serial.print("C = ");
  Serial.println(C);
  Serial.print("F0 = ");
  Serial.println(F0);
  Serial.print("F1 = ");
  Serial.println(F1);
  Serial.print("F2 = ");
  Serial.println(F2);
```

```
  Serial.println();

  delay(2000);

}
```

Given Circuit:



Conclusion:

This project helps us to know how to implement ALU with the help of Arduino. After doing programmed ALU we have to check and result match with correct one, which indicate that ALU is successfully built.

Link for the circuit:

https://www.tinkercad.com/things/lX2vaiuIdTL-copy-of-2-81-mux/editel?sharecode=v0ugNIp2ymtxQGza6PfyikvASvTH3vpgn2o-558aul