LAB REPORT – 4

Name: Vanshita

Roll Number: 2021101102

Group: G6

Aim of the experiment:

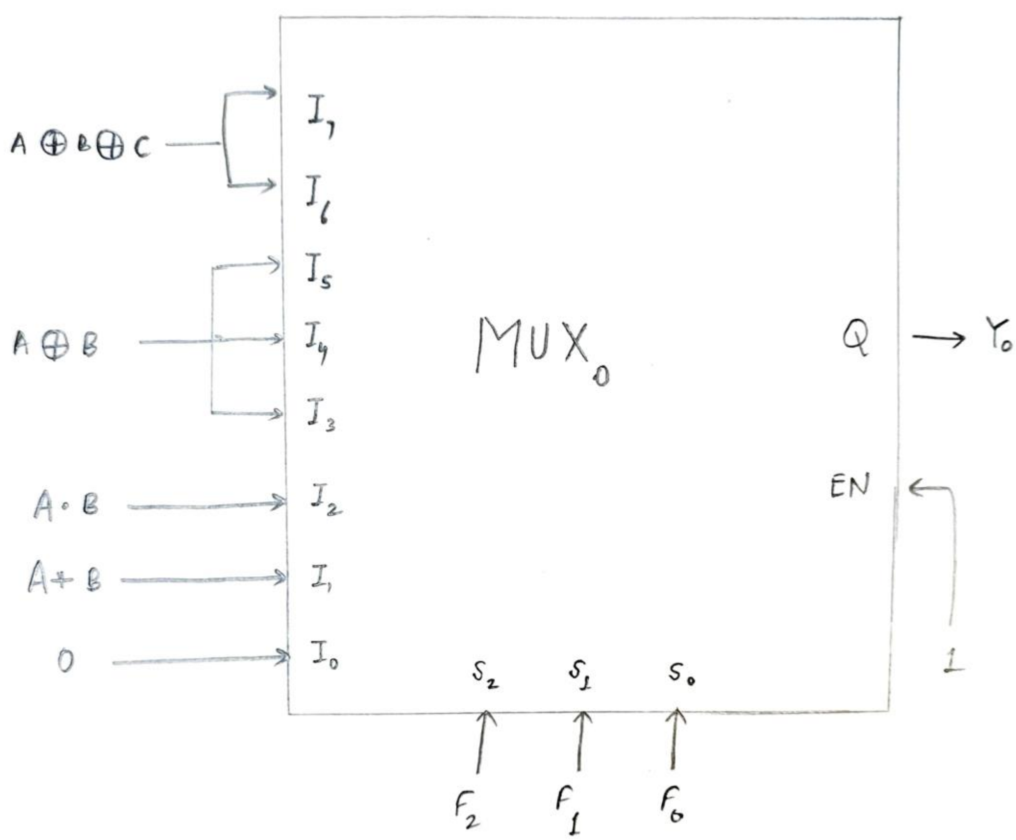
To design an Arithmetic and Logic Unit (ALU) capable of performing 8 Arithmetic/Logic functions on 1-bit operands as listed:

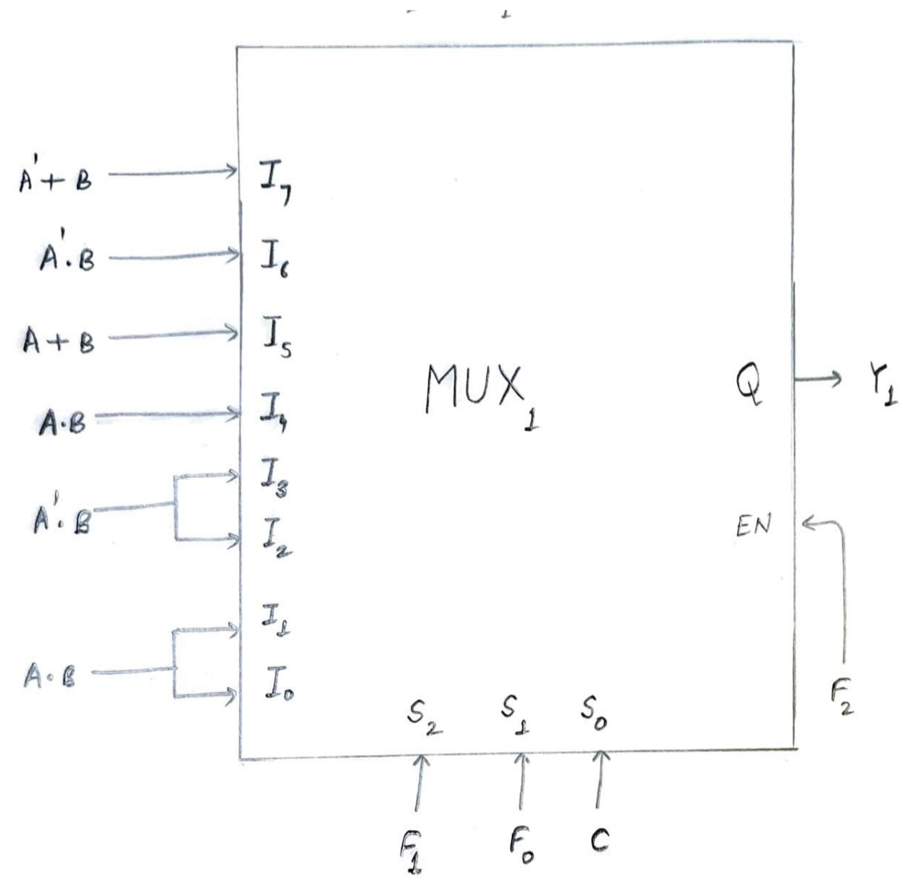
$F_2F_1F_0$	ALU Function	Y1	Y0
000	0(ZERO)	-	0
001	A OR B	-	A + B
010	A AND B	-	A.B
011	A EXOR B	-	$A \oplus B$
100	A PLUS B	Carry	Sum
101	A MINUS B	Borrow	Difference
110	A PLUS B PLUS C	Carry	Sum
111	A MINUS B MINUS C	Borrow	Difference

Electronic components used:

1. Breadboards (2)
2. Small Breadboards (2)
3. Mini Breadboards (4)
4. Connecting Wires
5. LEDs (2)
6. Resistors (2)
7. Arduino UNO

Reference Diagram:





Procedure:

1. Drag all the electronic components to the Tinkercad working area and set up the MUX circuit.
2. The final ALU output bits i.e., Y_0 and Y_1 will be generated by the two 8-input multiplexers – referred to as MUX 0 and MUX 1 respectively.
3. Note that MUX 0 is always enabled, while MUX 1 is enabled only when F_2= 1 (for Arithmetic functions only).

This is because Y_1 is required only to provide the CARRY/BORROW output for Arithmetic functions.

4. Verify that MUX0 and MUX1 generate the expected outputs Y0 and Y1 for different input lines.

Code:

```
int pin1 = 2;
int pin2 = 3;
int pin3 = 4;
int pin4 = 11;
int pin5 = 12;
int pin6 = 13;
int F2, F1,F0, A, B, C, k;

void setup()
{
pinMode(pin1, OUTPUT);
pinMode(pin2, OUTPUT);
pinMode(pin3, OUTPUT);
pinMode(pin4, OUTPUT);
pinMode(pin5, OUTPUT);
pinMode(pin6, OUTPUT);
Serial.begin(9600);
}
```

```
void loop()
{
Serial.print("\nF2=");
while(Serial.available()==0){}
F2= Serial.read();
F2= F2-'0';
Serial.println(F2);

_Serial.print("F1=");
while(Serial.available()==0){}
F1= Serial.read();
F1 = F1- '0';
Serial.println(F1);

Serial.print("F0=");
while(Serial.available()==0){}
F0= Serial.read();
F0 = F0 - '0';
Serial.println(F0);

Serial.print("A=");
while(Serial.available()==0){}
A= Serial.read();
```

```
A = A - '0';
Serial.println(A);

Serial.print("B=");
while(Serial.available()==0){}
B= Serial.read();
B = B- '0';
Serial.println(B);

Serial.print("C=");
while(Serial.available()==0){}
C= Serial.read();
C = C - '0';
Serial.println(C);

digitalWrite(pin1, OUTPUT);
digitalWrite(pin2, OUTPUT);
digitalWrite(pin3, OUTPUT);
digitalWrite(pin4, OUTPUT);
digitalWrite(pin5, OUTPUT);
digitalWrite(pin6, OUTPUT);

Serial.print("Enter anything to go to read again");
while(Serial.available()==0){}
```

```
k=Serial.read();
}
```

Observation:

MUX$_0$

$F_2F_1F_0$	Output	Operation	ABC							
			000	001	010	011	100	101	110	111
000	I_0	0	0	0	0	0	0	0	0	0
001	I_1	A + B	0	0	1	1	1	1	1	1
010	I_2	A . B	0	0	0	0	0	0	1	1
011	I_3	$A \oplus B$	0	0	1	1	1	1	0	0
100	I_4	$A \oplus B$	0	0	1	1	1	1	0	0
101	I_5	$A \oplus B$	0	0	1	1	1	1	0	0
110	I_6	$A \oplus B \oplus C$	0	1	1	0	1	0	0	1
111	I_7	$A \oplus B \oplus C$	0	1	1	0	1	0	0	1

MUX$_1$

$F_2F_1F_0$	Output	ABC							
		000	001	010	011	100	101	110	111
000	I_0	-	-	-	-	-	-	-	-

001	I_1	-	-	-	-	-	-	-	-
010	I_2	-	-	-	-	-	-	-	-
011	I_3	-	-	-	-	-	-	-	-
100	I_4	0	0	0	0	0	0	1	0
101	I_5	0	0	1	1	0	0	0	0
110	I_6	0	0	0	1	0	1	1	1
111	I_7	0	1	1	1	0	0	0	1

Conclusion:

Upon giving different inputs of $F_2F_1F_0$ and ABC in the Tinkercad circuit, we see that the outputs match with those depicted in the truth table. Hence, the ALU is verified.

Link for Tinkercad Simulation Circuit