LAB REPORT : 4

NAME: ROMICA RAISINGHANI

ROLL NUMBER: 2021101053

GROUP: 1

AIM OF THE EXPERIMENT:

To design an Arithmetic and Logic Unit (ALU) capable of performing 8 Arith-metic/Logic functions on 1-bit operands as shown in Figure 1: ALU Function Table with the help of the already provided (8*1) Muxes and then setting it up using Arduino and input pins.

$F_2F_1F_0$	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 ⊕ 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

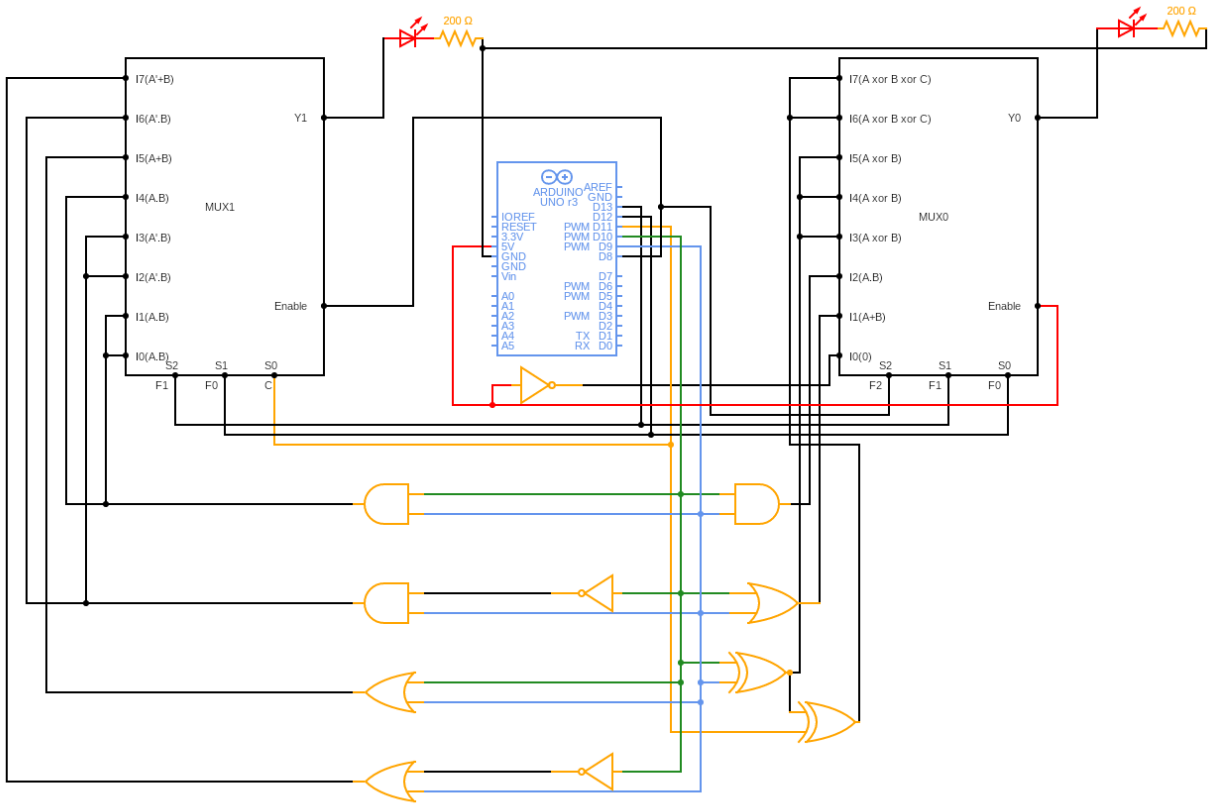
Figure 1: ALU Function Table

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 .

ELECTRONIC COMPONENTS USED:

Two already provided (8 * 1) MUX, Arduino Uno R3, breadboard, 74HC32 Quad OR Gate, 74HC04 Hex Inverter(NOT Gate), 74HC08 Quad AND Gate, 74HC06 XOR Gate and some connecting wires.

REFERENCE CIRCUIT:



PROCEDURE:

1. The circuit is copied from the already provided template containing the basic structure of the Arithmetic and Logic Unit(ALU) that is formed from two (8*1) muxes as shown.The already provided circuits also contains two RED LEDs and two resistances each 200 ohms depicting the outputs Y0 and Y1 of MUX 0 and MUX 1 respectively.
2. The breadboards are appropriately connected to the live (5V pin) and the ground pins of the Arduino and also ensuring proper connections through each gated as well.
3. The Integrated Circuit chips are placed on the breadboards – 2 Quad AND gates , 2 QUAD OR gates and one QUAD XOR gate to facilitate our logic inputs through these mentioned gates.
4. The IC's are connected to ground and live appropriately for proper supply voltages.
5. The inputs are taken by the Arduino, A(orange), B(grey), C(purple) and F0(turquoise), F1(blue), F2(white) and then connected to the appropriate gates for the proper functioning of the ALU as mentioned in Figure 1: ALU Function Table above.
6. The 8 inputs for both the Muxes are connected as shown in the above reference circuit and the correct enable inputs are also given.
7. The inputs are varied and the output tables were noted by considering the glow of the leds for Y0 and Y1.
8. An appropriate code is given for the Arduino to function properly.
9. The truth tables are constructed from the observation tables and later used for verification of the outputs Y0 and Y1 through MUX0 and MUX1 respectively

NOTE: THE INPUT FORMAT FOR THE SERIAL MONITOR IS AS FOLLOWS : ABC F2F1F0 (for example the input 010 001 leads to a glow in Y0 whereas Y1 remains off)

CONCLUSION:

We have successfully made an Arithmetic and Logic Unit (ALU) using basic logic gates and simulated it with the help of an Arduino.

The final ALU output bits Y0 and Y1 are generated by the two 8-input multiplexers – referred to as MUX 0 and MUX 1 respectively. We also conclude 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. Further we verify our theory and output with the truth tables as shown below.

The Theorotical Truth Table for the Arithmetic and Logic Unit (ALU) is:

S.NO.	A	B	C	A.B	A+B	A'. B	A'+B	A ⊕ B	A ⊕B ⊕C
1	0	0	0	0	0	0	1	0	0
2	0	0	1	0	0	0	1	0	1
3	0	1	0	0	1	1	1	1	1
4	0	1	1	0	1	1	1	1	0
5	1	0	0	0	1	0	0	1	1
6	1	0	1	0	1	0	0	1	0
7	1	1	0	1	1	0	1	0	0
8	1	1	1	1	1	0	1	0	1

The Observation Truth Table for the Arithmetic and Logic Unit (ALU) is:

NOTE : MUX 0 is always enabled whereas MUX 1 is enabled in Arithmetic Functions only when F2 = 1.

FUNCTION (000) WHERE F2=0 ; F1=0 ; F0=0 (NULL FUNCTION)

S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	1	0	0
5	0	0	0	1	0	0	0	0
6	0	0	0	1	0	1	0	0
7	0	0	0	1	1	0	0	0
8	0	0	0	1	1	1	0	0

FUNCTION (001) WHERE F2=0 ; F1=0 ; F0=1 (A OR B)

S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	0	0	1	0	0	0	0	0
2	0	0	1	0	0	1	0	0
3	0	0	1	0	1	0	1	0
4	0	0	1	0	1	1	1	0
5	0	0	1	1	0	0	1	0
6	0	0	1	1	0	1	1	0
7	0	0	1	1	1	0	1	0
8	0	0	1	1	1	1	1	0

FUNCTION (010) WHERE F2=0 ; F1=1; F0=0 (A AND B)

S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0
3	0	1	0	0	1	0	0	0
4	0	1	0	0	1	1	0	0
5	0	1	0	1	0	0	0	0
6	0	1	0	1	0	1	0	0
7	0	1	0	1	1	0	1	0
8	0	1	0	1	1	1	1	0

FUNCTION (011) WHERE F2=0 ; F1=1 ; F0=1 (A XOR B)

S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	0	1	1	0	0	0	0	0
2	0	1	1	0	0	1	0	0
3	0	1	1	0	1	0	1	0
4	0	1	1	0	1	1	1	0
5	0	1	1	1	0	0	1	0
6	0	1	1	1	0	1	1	0
7	0	1	1	1	1	0	0	0
8	0	1	1	1	1	1	0	0

FUNCTION (100) WHERE F2=1; F1=0 ; F0=0 (A PLUS B)

S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	1	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0
3	1	0	0	0	1	0	1	0
4	1	0	0	0	1	1	1	0
5	1	0	0	1	0	0	1	0
6	1	0	0	1	0	1	1	0
7	1	0	0	1	1	0	0	1
8	1	0	0	1	1	1	0	1

FUNCTION (101) WHERE F2=1 ; F1=0 ; F0=1 (A DIFFERENCE B)

S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	1	0	1	0	0	0	0	0
2	1	0	1	0	0	1	0	0
3	1	0	1	0	1	0	1	1
4	1	0	1	0	1	1	1	1
5	1	0	1	1	0	0	1	0
6	1	0	1	1	0	1	1	0
7	1	0	1	1	1	0	0	0
8	1	0	1	1	1	1	0	0

FUNCTION (110) WHERE F2=1 ; F1=1 ; F0=0 (A PLUS B PLUS C)

S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	1	1	0
3	1	1	0	0	1	0	1	0
4	1	1	0	0	1	1	0	1
5	1	1	0	1	0	0	1	0
6	1	1	0	1	0	1	0	1
7	1	1	0	1	1	0	0	1
8	1	1	0	1	1	1	1	1

FUNCTION (111) WHERE F2=1 ; F1=1 ; F0=1 (A DIFFERENCE B DIFFERENCE C)

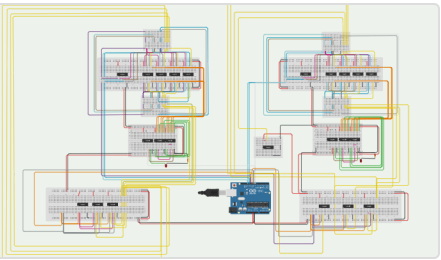
S.NO.	F2	F1	F0	A	B	C	OUTPUT PIN Y0	OUTPUT PIN Y1
1	1	1	1	0	0	0	0	0
2	1	1	1	0	0	1	1	1
3	1	1	1	0	1	0	1	1
4	1	1	1	0	1	1	0	1
5	1	1	1	1	0	0	1	0
6	1	1	1	1	0	1	0	0
7	1	1	1	1	1	0	0	0
8	1	1	1	1	1	1	1	1

LINK FOR TINKERCAD SIMULATION:

Circuit design LAB 4 ALU | Tinkercad

Circuit design LAB 4 ALU created by romicar@11 with Tinkercad

https://www.tinkercad.com/things/elxPxTIanMp



THANK YOU!