

Heaven's light is our guide.

Rajshahi University of Engineering and Technology (RUET)

Department of Electrical & Electronic Engineering

Course no. EEE2214

Course title: Digital Electronics I Sessional

Experiment no. 01

Experiment name: Experimental study of a Full-adder circuit.

Submitted to:

Prof. Dr. Md. Shahidul Islam

Professor

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Submitted by:

Ashraf Al- Khalique

Roll: 1801171; Session: 2018-2019
Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Date of experiment: July 04, 2021.

Date of submission: August 31, 2021.

Experiment No. 01

Experiment name: Experimental study of a Full-adder circuit.

Objectives: Followings are the primary objective of this experiment,

1. To learn about the logic operation of a full-adder circuit.
2. To design a full-adder circuit using its basic expression.
3. To simulate logic opeartion of a full-adder circuit

Required Apparatus:

1. Power supply (1 piece);
2. Quad XOR gate (IC no.7486, 1 piece);
3. Quad AND gate (IC no.7408, 1 piece);
4. Quad OR gate (IC no.7432, 1 piece);
5. Dip switch SPST x4 (1 piece);
6. LED (2 pieces);
7. Resistor ($1k\Omega$, 2 pieces);
8. Simulator (Tinker cad).

Theory:

Binary Adders are arithmetic circuits in the form of half-adders and full-adders used to add together two binary digits. A basic Binary “Half Adder” circuit can be made from standard AND and Ex-OR gates allowing us to “add” together two single bit binary numbers, A and B.

The addition of these two digits produces an output called the SUM of the addition and a second output called the CARRY or Carry-out, (C_{OUT}) bit according to the rules for binary addition.

The main difference between the Full Adder and the previous Half Adder is that a full adder has three inputs. The same two single bit data inputs A and B as before plus an additional Carry-in (C_{IN}) input to receive the carry from a previous stage as shown below. Apart from this, Logic gates are all same.

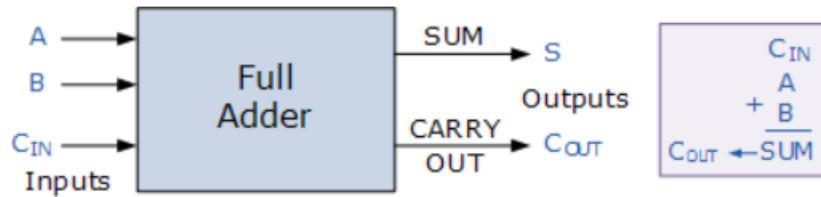


Fig 1.1: Block Diagram of Full-Adder circuit.

Then the full adder is a logical circuit that performs an addition operation on three binary digits and just like the half adder, it also generates a carry out to the next addition column. Then a Carry-in is a possible carry from a less significant digit, while a Carry-out represents a carry to a more significant digit. In many ways, the full adder can be thought of as two half adders connected together, with the first half adder passing its carry to the second half adder as shown.

Then the Boolean expression for a full adder is as follows.

For the SUM(S) bit:

$$\text{SUM} = (\text{A XOR B}) \text{ XOR } \text{C}_{\text{in}} = (\text{A} \oplus \text{B}) \oplus \text{C}_{\text{in}}$$

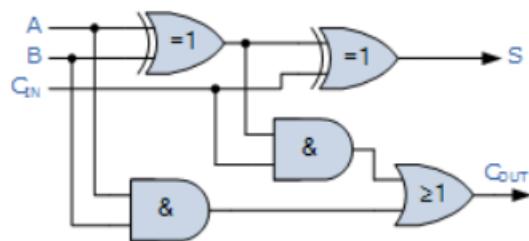
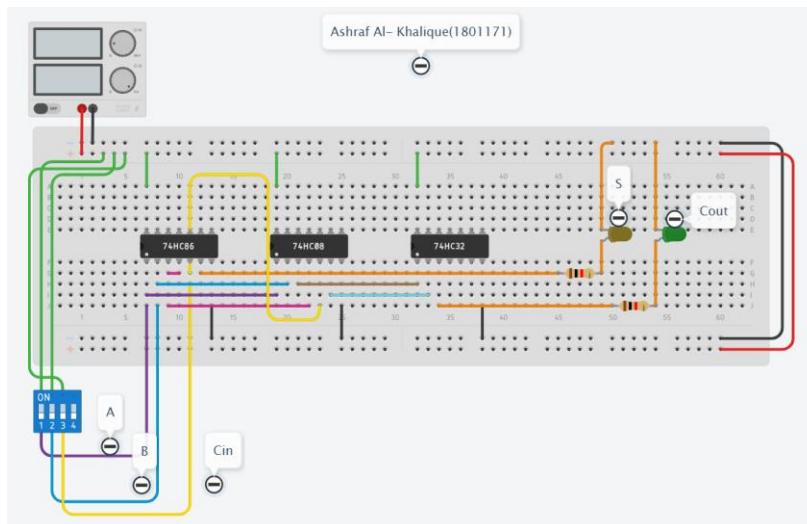
For the CARRY-OUT(C_{out}) bit:

$$\text{CARRY-OUT} = \text{A AND B OR } \text{C}_{\text{in}} (\text{A XOR B}) = \text{A.B} + \text{C}_{\text{in}} (\text{A} \oplus \text{B})$$

Truth Table:

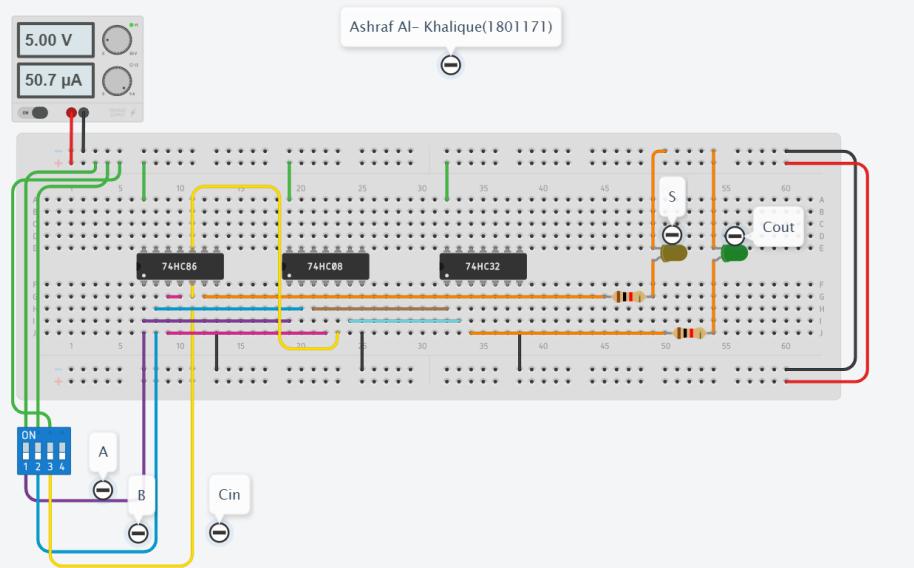
Input			Output	
A	B	C _{in}	Sum	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0

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

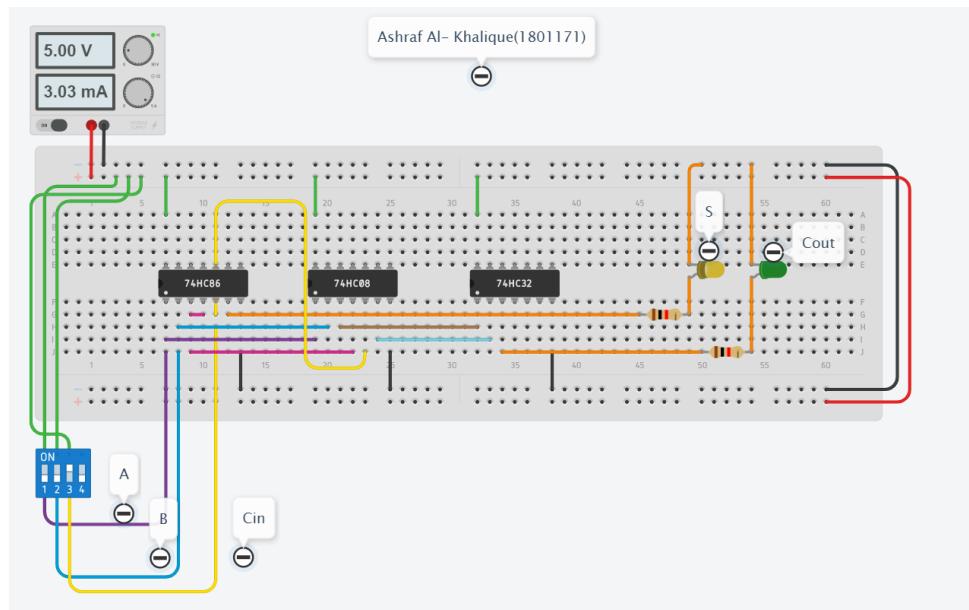
Circuit Diagram:***Fig 1.2: Logic Diagram of Full-Adder circuit.******Fig 1.3: Circuit Diagram of Full-Adder circuit.***

Output:

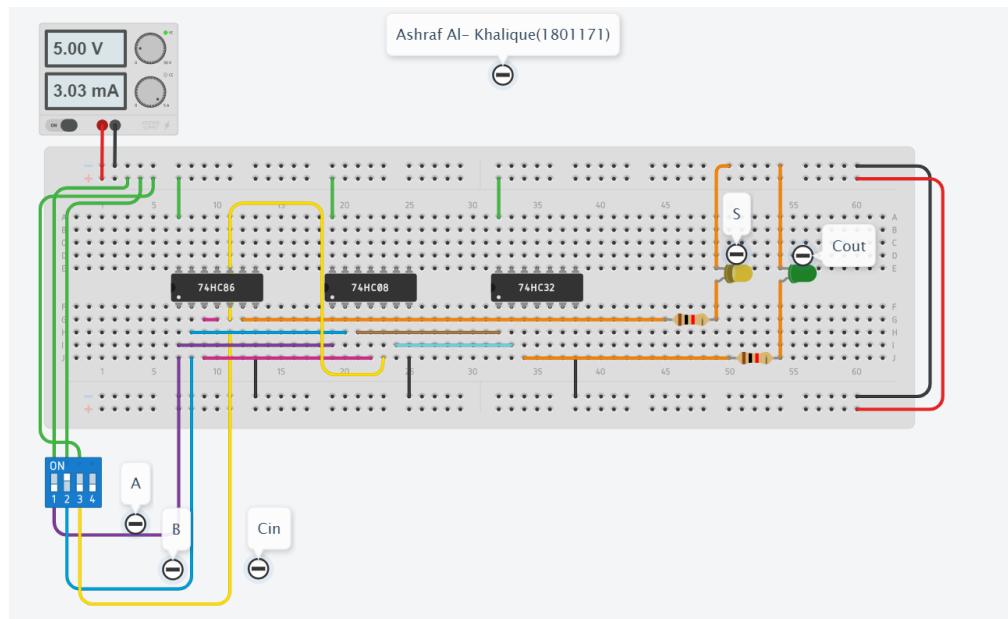
1. When $A=0, B=0, C_{in}=0$; then $S=0, C_{out}=0$



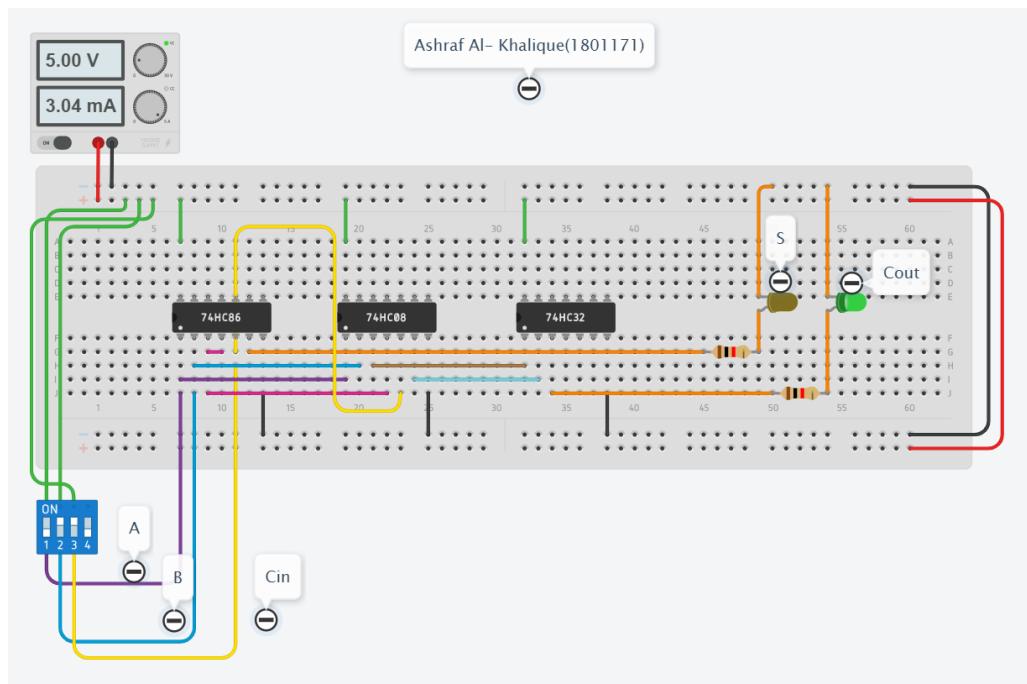
2. When $A=0, B=0, C_{in}=1$; then $S=1, C_{out}=0$



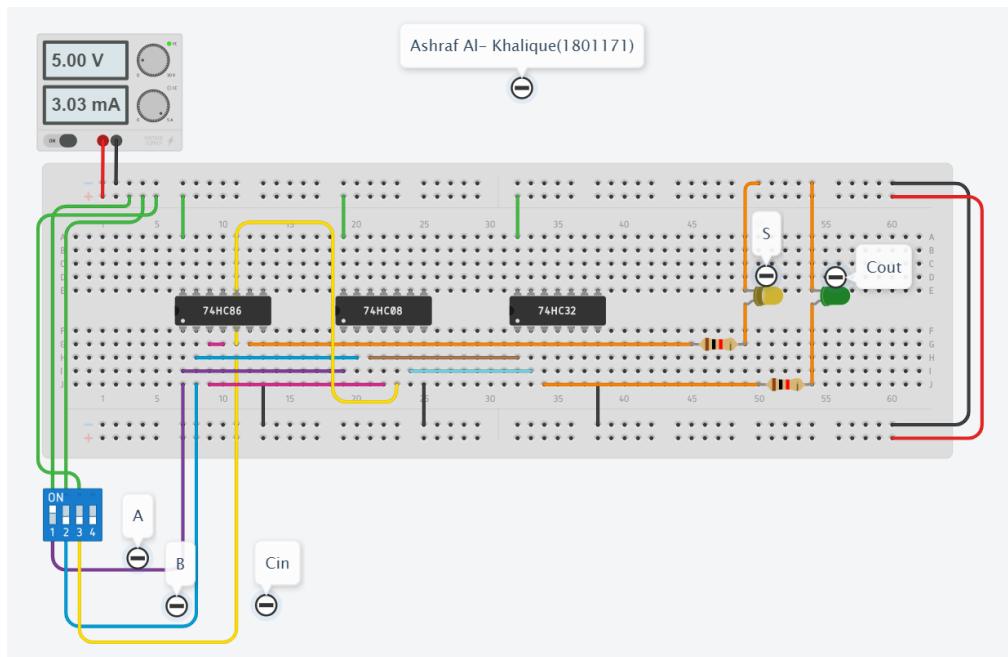
3. When $A=0, B=1, C_{in}=0$; then $S=1, C_{out}=0$



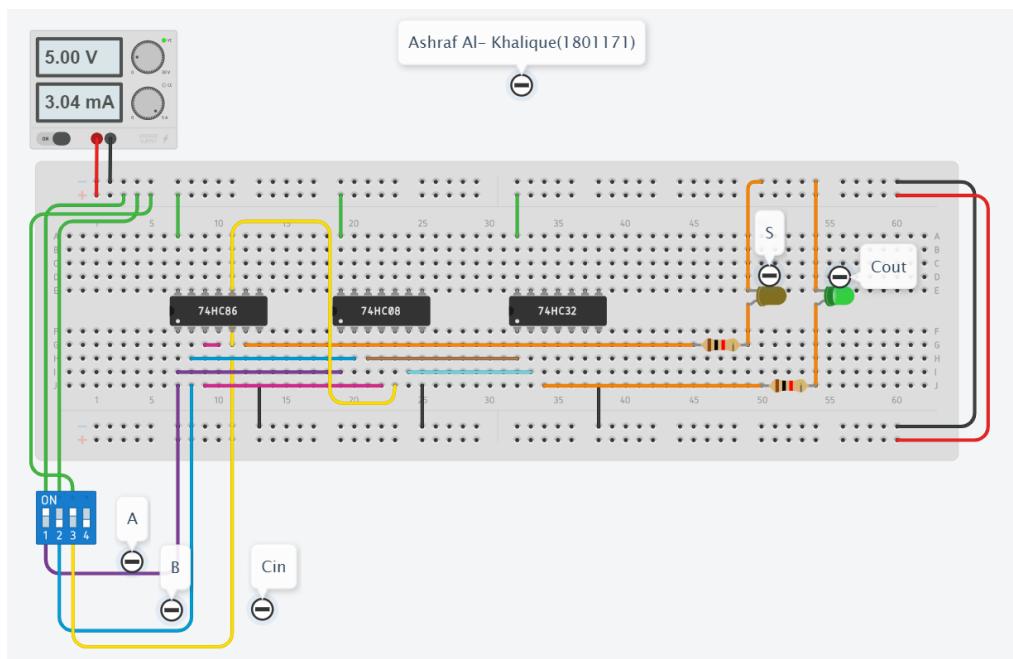
4. When $A=0, B=1, C_{in}=1$; then $S=0, C_{out}=1$



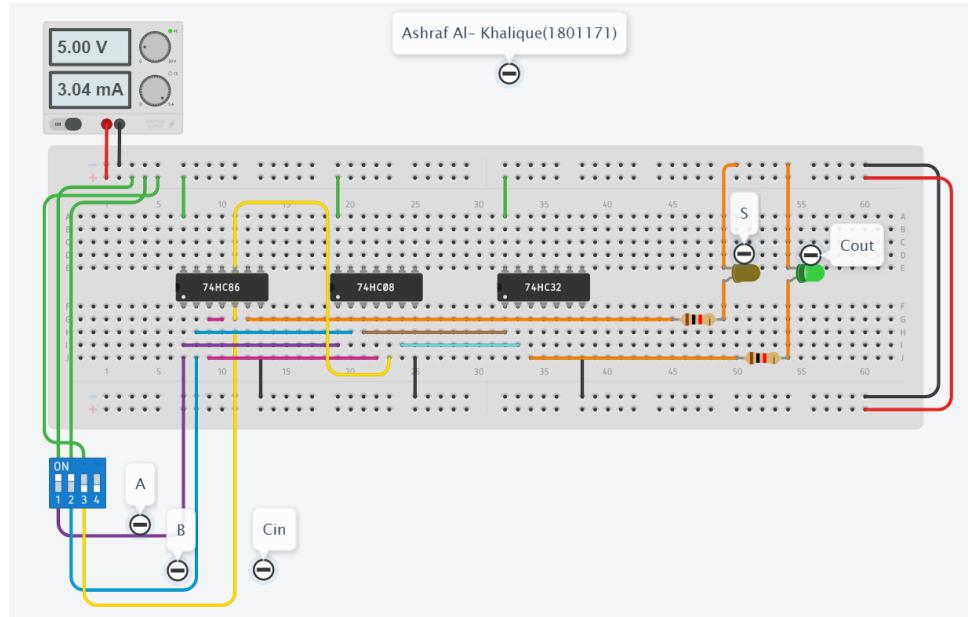
5. When A= 1, B= 0, C_{in}= 0; then S= 1, C_{out}= 0



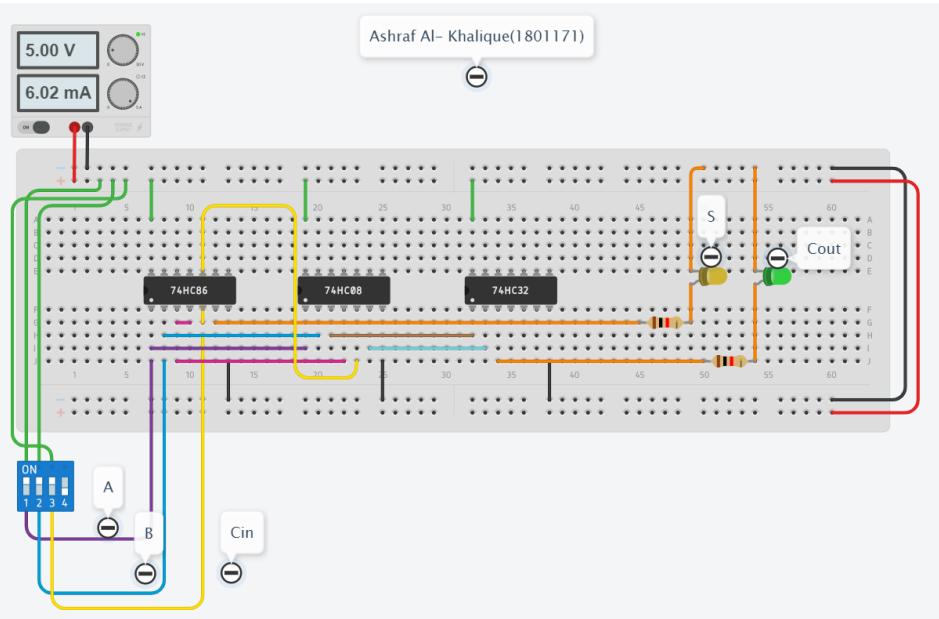
6. When A= 1, B= 0, C_{in}= 1; then S= 0, C_{out}= 1



7. When A= 1, B= 1, C_{in}= 0; then S= 0, C_{out}= 1



8. When A= 1, B= 1, C_{in}= 1; then S= 1, C_{out}= 1



Discussion & Conclusion:

The circuit was connected according to fig 1.2 in tinkercad. After that, X-OR operation was done on A and B which gave SUM output. Again, AND operation was done on A, B, C_{in}, and all of their output was given X-OR input, giving us CARRY-OUT output. When definite binary input was given at the input, their

designate logic operations occurred and SUM and C_{out} LED lit up according to truth table. Thus, it was assured that the experiment was done correctly.

Heaven's light is our guide.

Rajshahi University of Engineering and Technology (RUET)

Department of Electrical & Electronic Engineering

Course no. EEE2214

Course title: Digital Electronics I Sessional

Experiment no. 02

Experiment name: Experimental study of a BCD to Excess-3 circuit.

Submitted to:

Prof. Dr. Md. Shahidul Islam

Professor

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Submitted by:

Ashraf Al- Khalique

Roll: 1801171; Session: 2018-2019
Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Date of experiment: August 03, 2021.

Date of submission: August 31, 2021.

Experiment No. 02

Experiment name: Experimental study of BCD to Excess-3 circuit.

Objectives: Followings are the primary objective of this experiment,

1. To learn about the BCD code and Excess-3 code.
2. To understand the conversion of BCD to Excess-3 circuit.
3. To design BCD to Excess-3 circuit using its basic expression.

Required Apparatus:

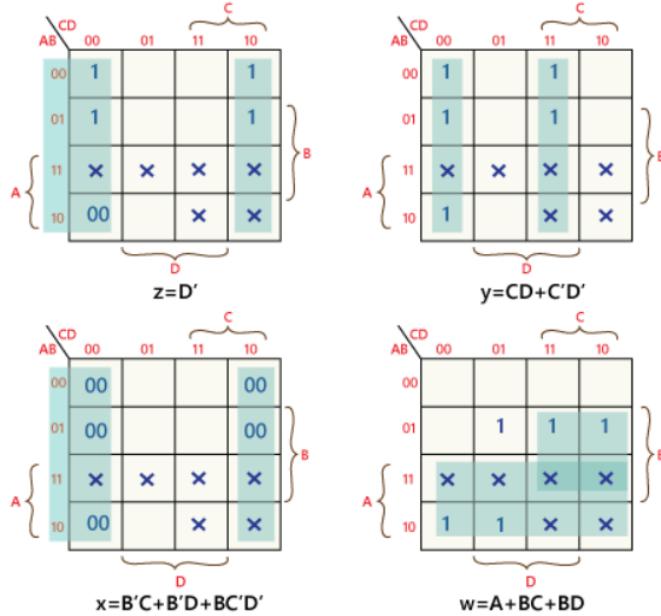
1. Power supply (1 piece);
2. Quad AND gate (IC no.7408, 1 piece);
3. Quad OR gate (IC no.7432, 1 piece);
4. Hex Inverter (IC no. 7404, 1 piece);
5. Dip switch SPST x4 (1 piece);
6. LED (2 pieces);
7. Resistor ($1k\Omega$, 2 pieces);
8. Simulator (Tinker cad).

Theory:

The Excess-3 binary code is an example of a self-complementary BCD code. A self-complementary binary code is a code which is always complimented in itself. By replacing the bit 0 to 1 and 1 to 0 of a number, we find the 1's complement of the number. The sum of the 1'st complement and the binary number of a decimal is equal to the binary number of decimals 9.

The process of converting BCD to Excess-3 is quite simple from other conversions. The Excess-3 code can be calculated by adding 3, i.e., 0011 to each four-digit BCD code. Below is the truth table for the conversion of BCD to Excess-3 code. In the below table, the variables A, B, C, and D represent the bits of the binary numbers. The variable 'D' represents the LSB, and the variable 'A' represents the MSB. In the same way, the variables w, x, y, and z represent the bits of the Excess-3 code. The variable 'z' represents the LSB, and the variable 'w' represents the MSB. The 'don't care conditions' is expressed by the variable 'X'.

The K mapping will look like this:



The minimize expression from each output obtained from K map are given below:

$$W = A + BC + BD; \quad X = B'C + B'D + BC'D'; \quad Y = CD + C'D'; \quad Z = D'$$

Truth Table:

BCD					Excess-3				
A	B	C	D	Decimal	W	X	Y	Z	Decimal
0	0	0	0	0	0	0	1	1	3
0	0	0	1	1	0	1	0	0	4
0	0	1	0	2	0	1	0	1	5
0	0	1	1	3	0	1	1	0	6
0	1	0	0	4	0	1	1	1	7
0	1	0	1	5	1	0	0	0	8
0	1	1	0	6	1	0	0	1	9
0	1	1	1	7	1	0	1	0	10
1	0	0	0	8	1	0	1	1	11
1	0	0	1	9	1	1	0	0	12

Circuit Diagram:

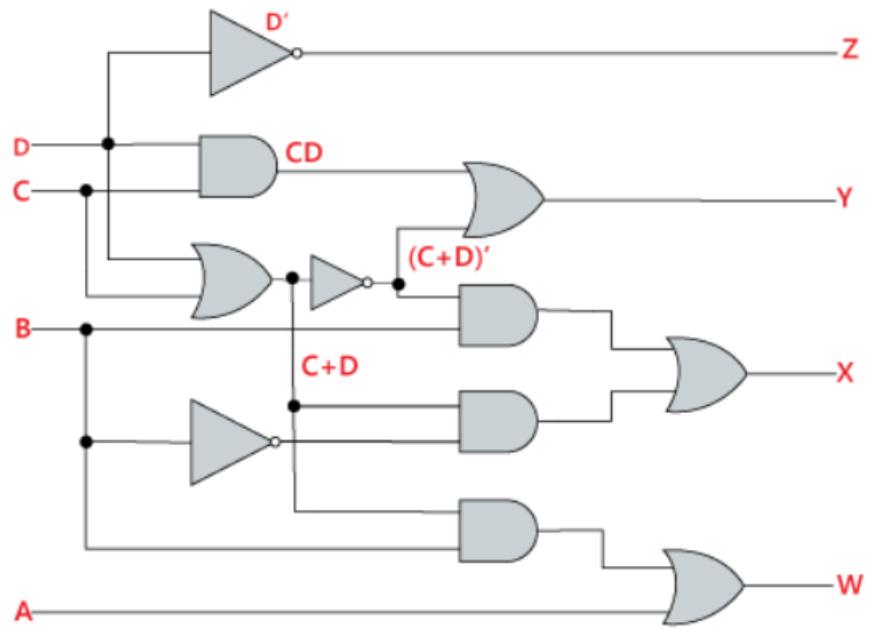


Fig 2.1: Logic Diagram of BCD to Excess-3 circuit.

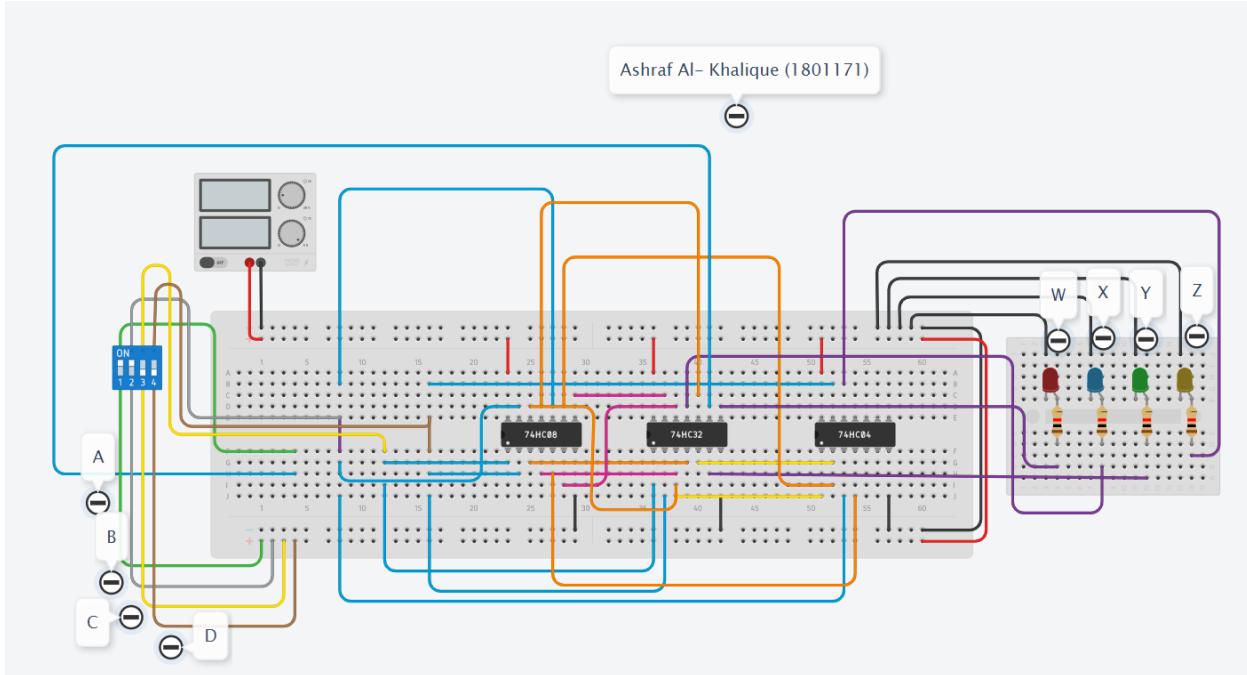
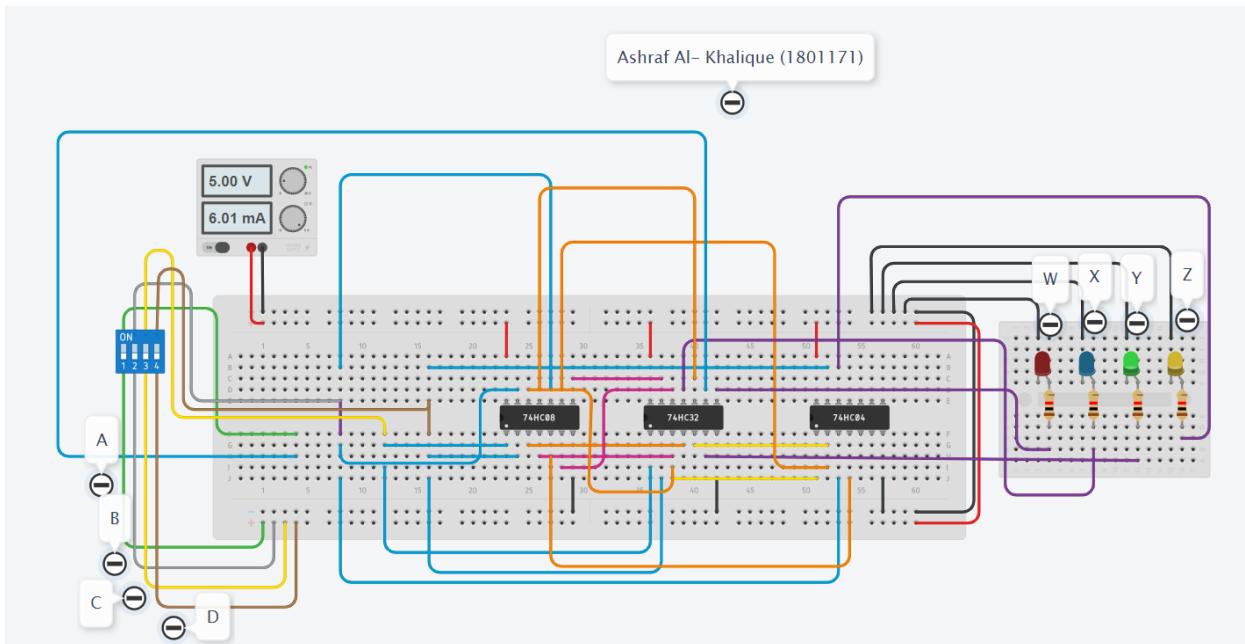


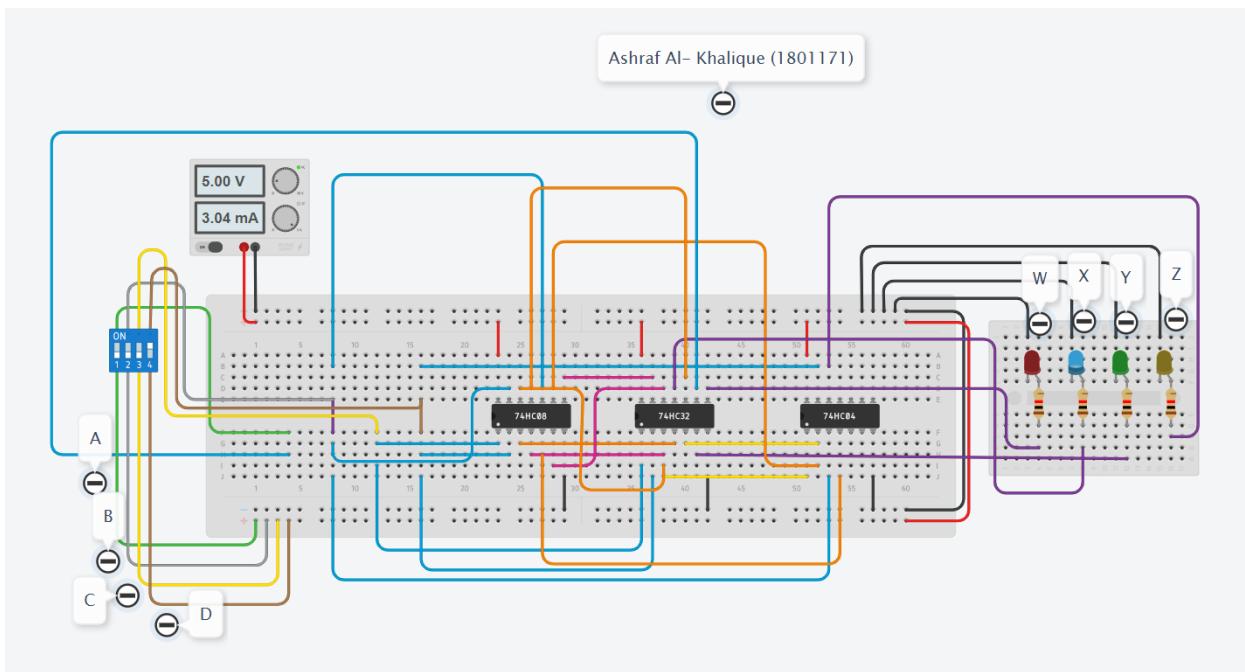
Fig 2.1: Circuit Diagram of BCD to Excess-3 circuit.

Output:

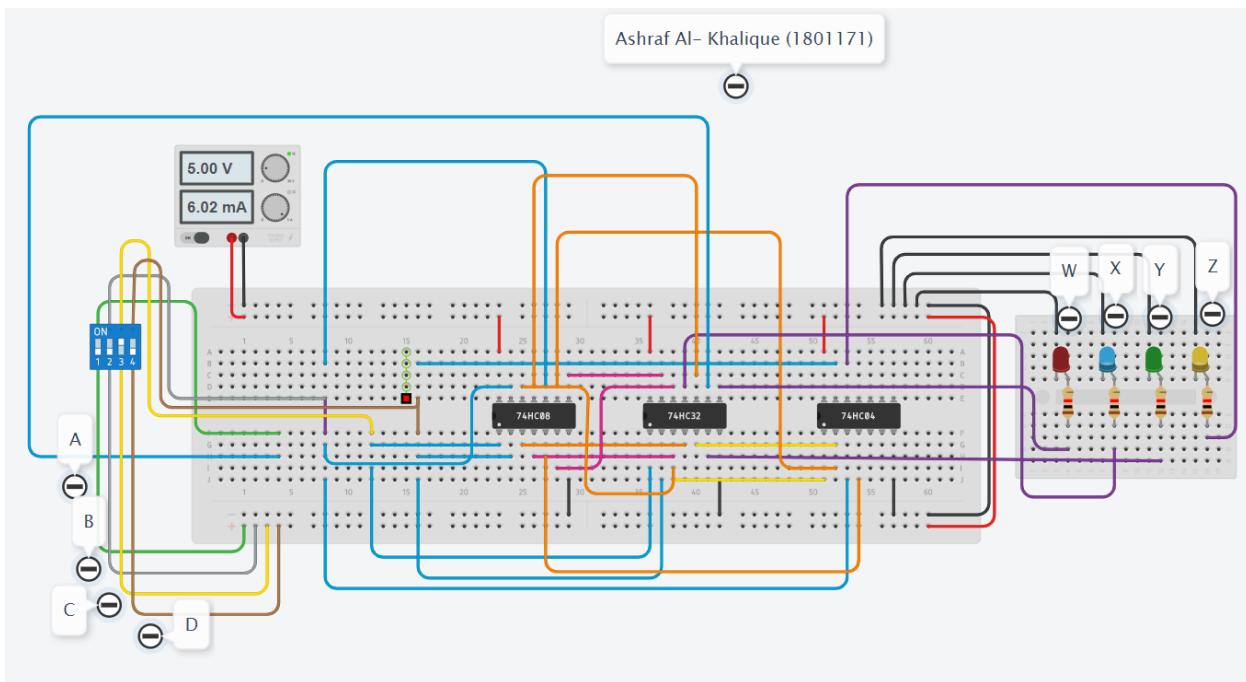
1. When $A=B=C=D=0$, then $W=0, X=0, Y=1, Z=1$



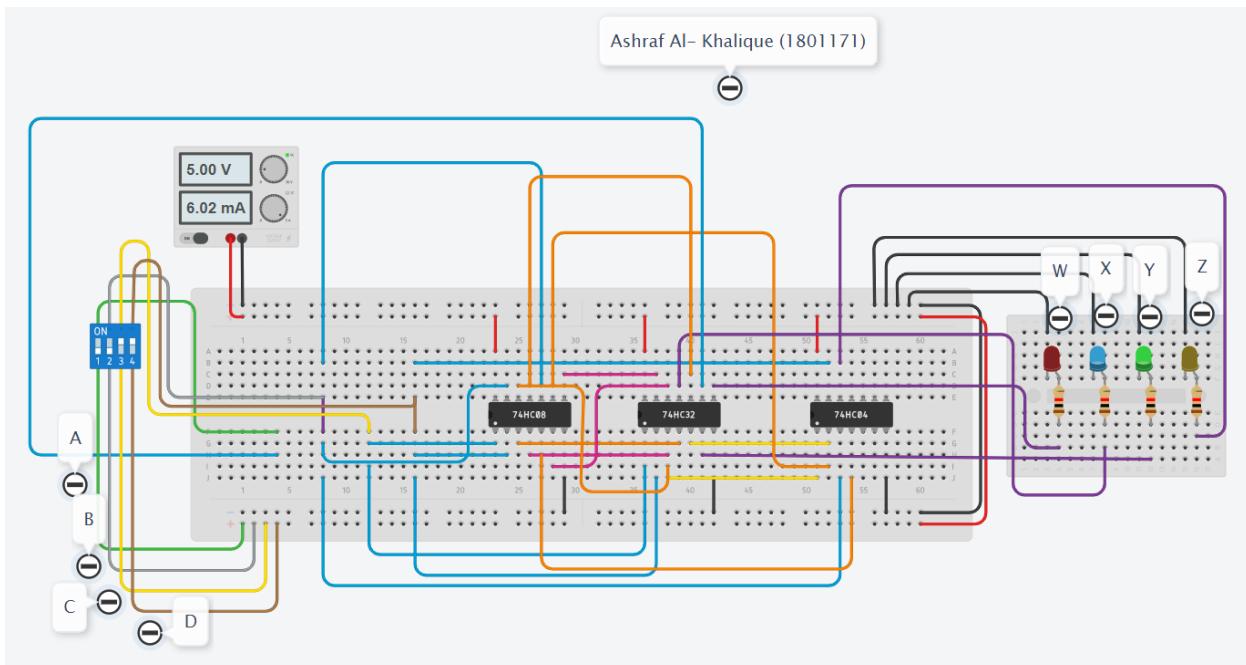
2. When $A=B=C=0, D=1$, then $W=0, X=1, Y=0, Z=0$



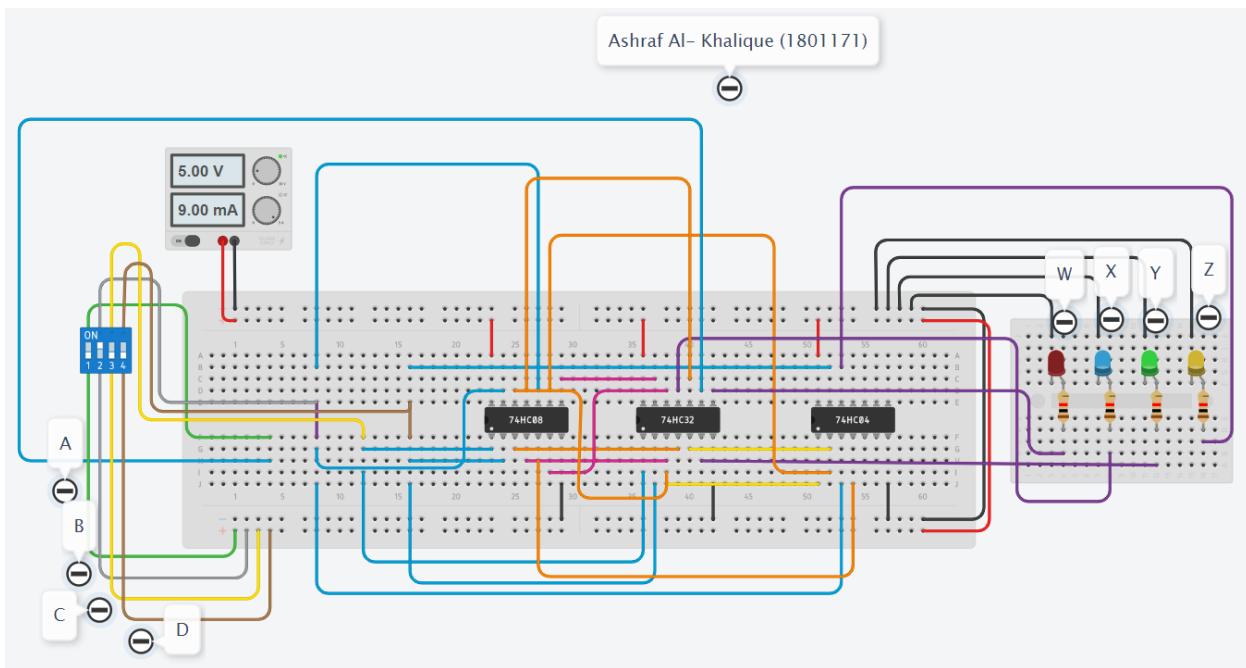
3. When $A=B=0, C=1, D=0$, then $W=0, X=1, Y=0, Z=1$



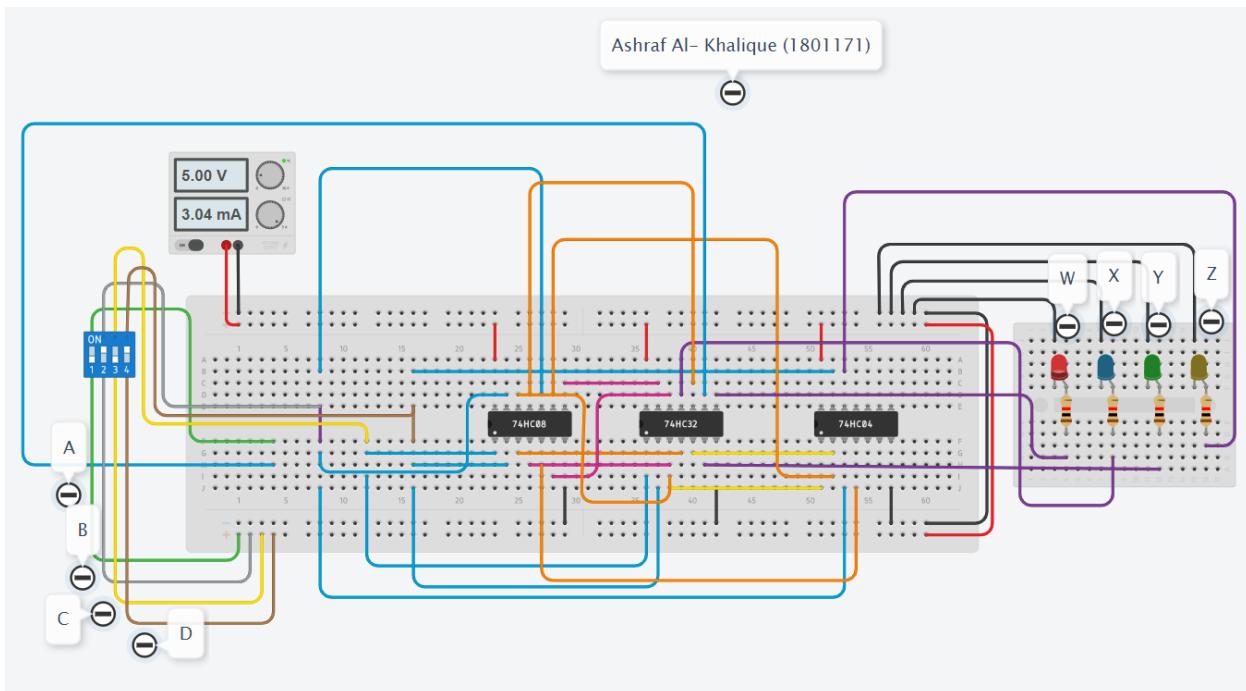
4. When $A=B=0$, $C=D=1$, then $W=0$, $X=1$, $Y=1$, $Z=0$



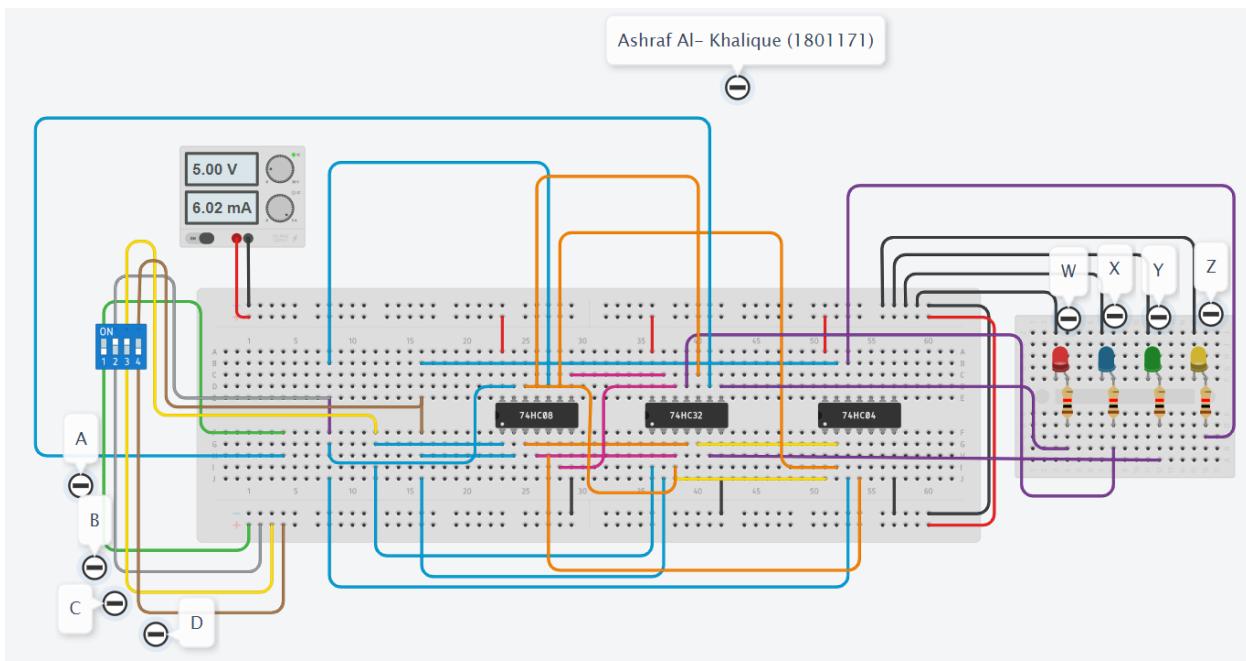
5. When $A=0$, $B=1$, $C=D=0$, then $W=0$, $X=1$, $Y=1$, $Z=1$



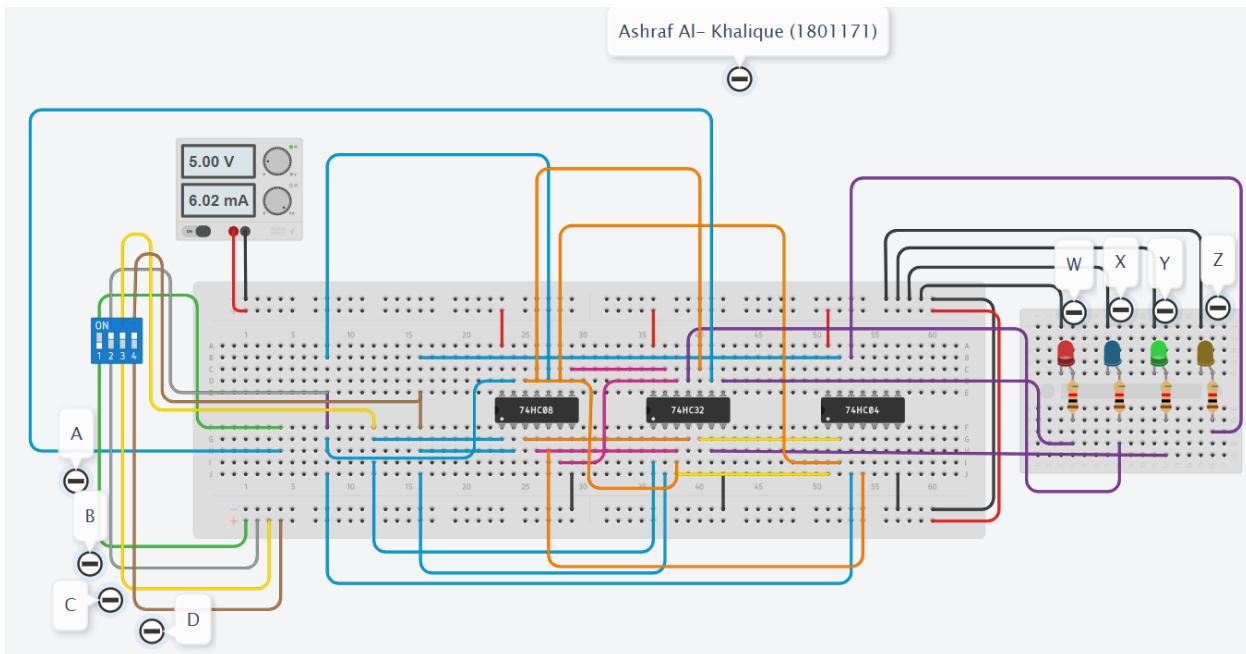
6. When $A=0, B=1, C=0, D=1$, then $W=1, X=0, Y=0, Z=0$



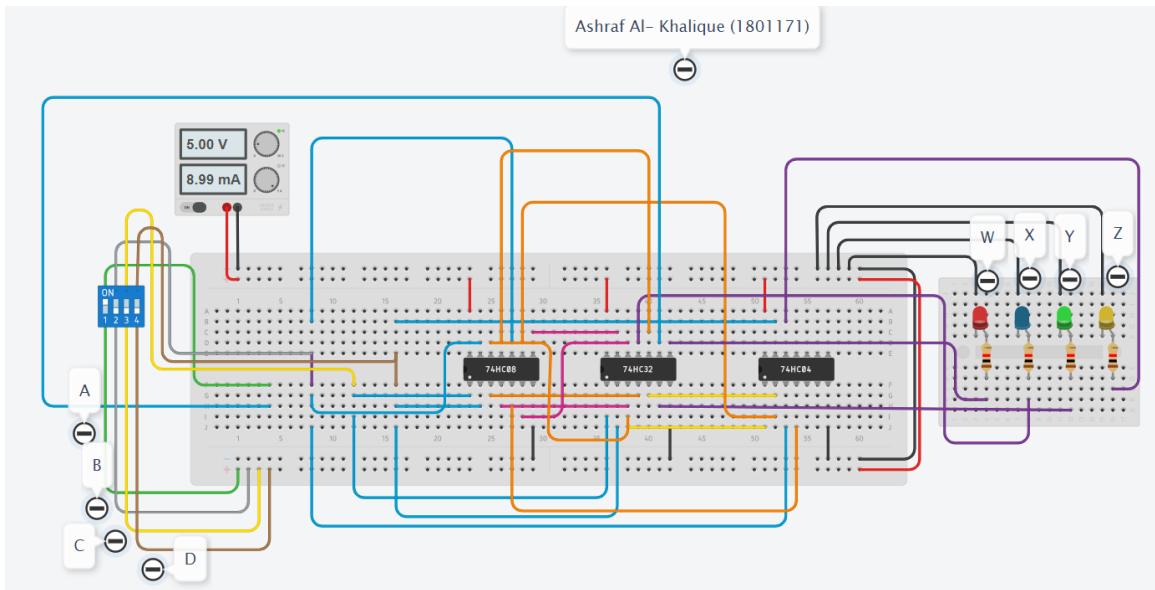
7. When $A=0, B=C=1, D=0$, then $W=1, X=0, Y=0, Z=1$



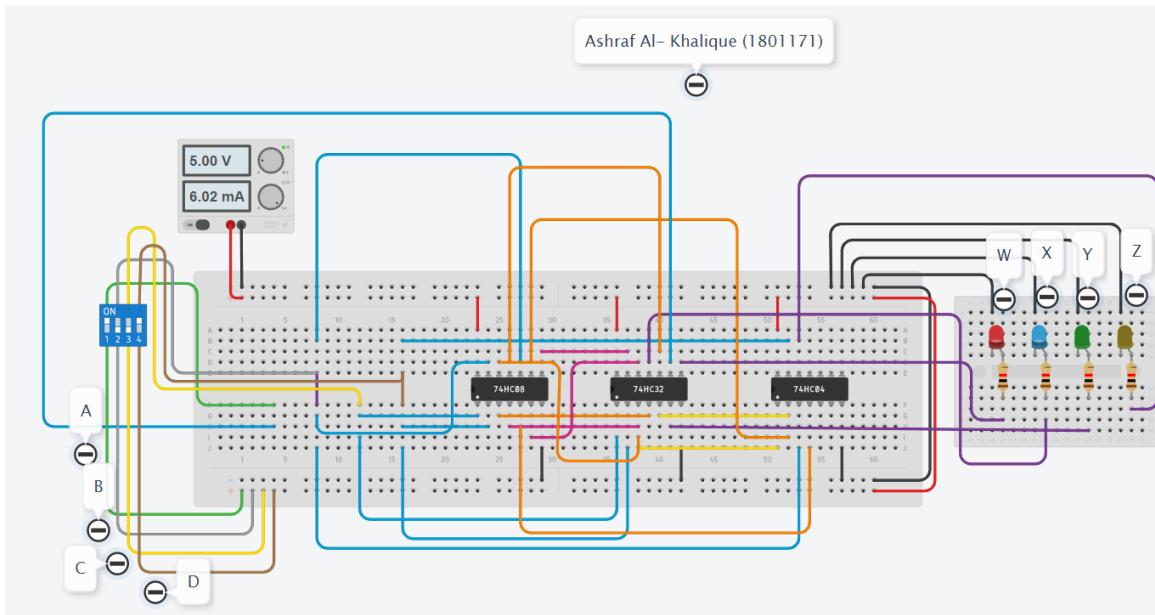
8. When $A=0$, $B=C=D=1$, then $W=1$, $X=0$, $Y=1$, $Z=0$



9. When $A=1$, $B=C=D=0$, then $W=1$, $X=0$, $Y=1$, $Z=1$



10. When $A = 1$, $B=C= 0$, $D= 1$, then $W= 1$, $X= 1$, $Y= 0$, $Z= 0$



Discussion & Conclusion:

In this experiment, the main purpose was to add 3 to any 4-digit BCD code. So, when, we gave decimal 0 in the input which means binary 0000, the output was 0011 which is decimal 3. This process was repeated for all the other values till decimal 9. We also gave input from 0-9. But whenever we crossed decimal 9 input, the output didn't add 3, instead added 6. Here, outputs follow the data table we found using the expression of BCD to Excess-3 converter.

Heaven's light is our guide.

Rajshahi University of Engineering and Technology (RUET)

Department of Electrical & Electronic Engineering

Course no. EEE2214

Course title: Digital Electronics I Sessional

Experiment no. 03

Experiment name: Experimental study of Multiplexer circuit.

Submitted to:

Prof. Dr. Md. Shahidul Islam

Professor

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Submitted by:

Ashraf Al- Khalique

Roll: 1801171; Session: 2018-2019
Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Date of experiment: August 03, 2021.

Date of submission: August 31, 2021.

Experiment No. 03

Experiment name: Experimental study of Multiplexer circuit.

Objectives: Followings are the primary objective of this experiment,

1. To learn about the Multiplexer circuit.
2. To simulate the logic operation Multiplexer circuit.
3. To design Multiplexer circuit from its basic expression.

Required Apparatus:

1. Power supply (1 piece);
2. Quad AND gate (IC no.7408, 1 piece);
3. Quad OR gate (IC no.7432, 1 piece);
4. Hex Inverter (IC no. 7404, 1 piece);
5. Dip switch SPST x4 (1 piece);
6. LED (1 piece);
7. Resistor ($1k\Omega$, 1 piece);
8. Simulator (Tinker cad).

Theory:

The multiplexer is a combinational logic circuit designed to switch one of several input lines to a single common output line. Multiplexers operate like very fast acting multiple position rotary switches connecting or controlling multiple input lines called “channels” one at a time to the output.

Multiplexers, or MUX's, can be either digital circuits made from high-speed logic gates used to switch digital or binary data or they can be analogue types using transistors, MOSFET's or relays to switch one of the voltage or current inputs through to a single output.

A 2-to-1 multiplexer consists of two inputs I_0 and I_1 , one selects input S and one output Y . Depending on the select signal, the output is connected to either of the inputs. Since there are two input signals, only two ways are possible to connect the inputs to the outputs, so one selects is needed to do these operations.

$$Y = S' I_0 + S I_1$$

Truth Table:

Inputs			MUX
Selector(s)	I ₀	I ₁	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Circuit Diagram:

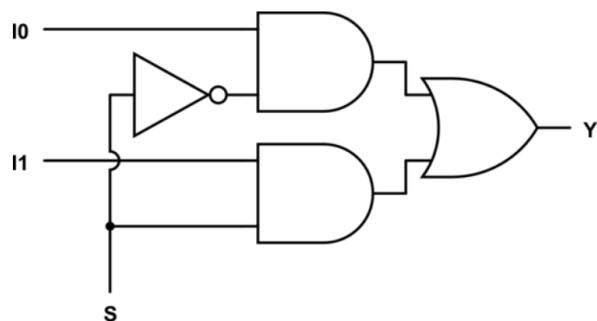


Fig 3.1: Logic Diagram of 2-1 Multiplexer circuit.

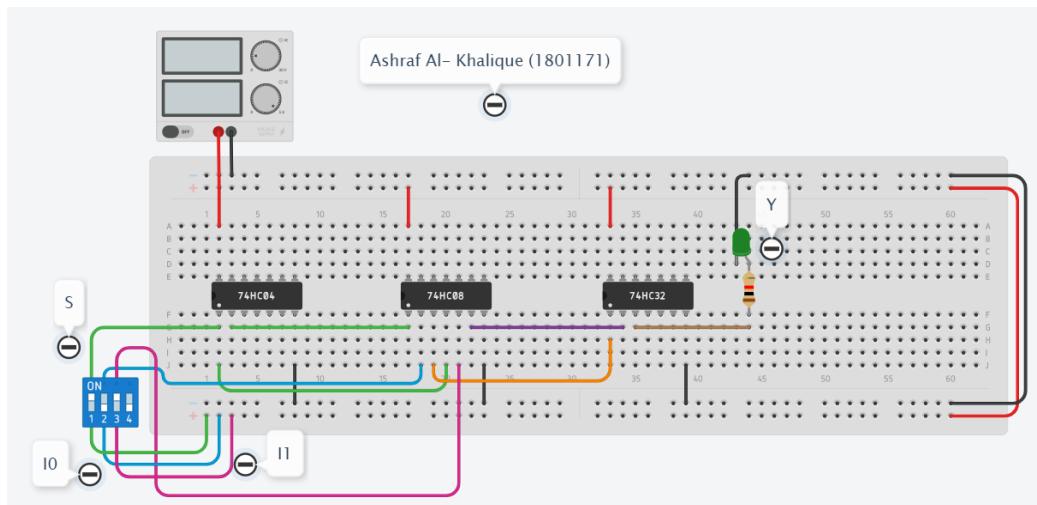
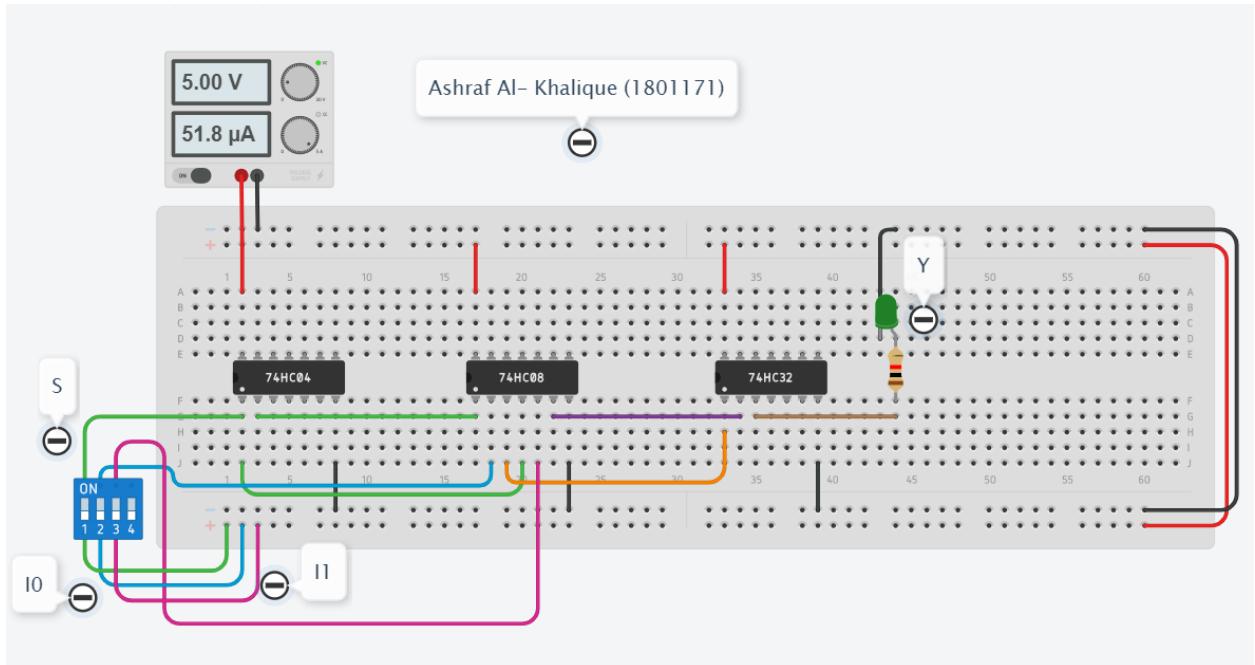


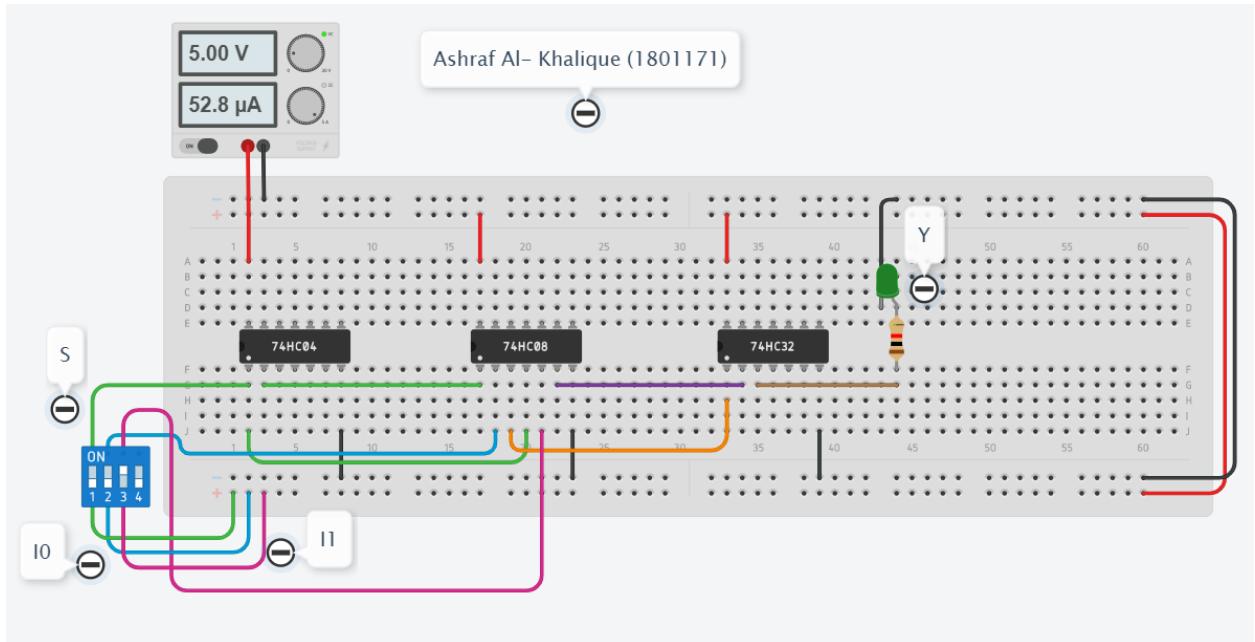
Fig 3.2: Circuit Diagram of 2-1 Multiplexer circuit.

Output:

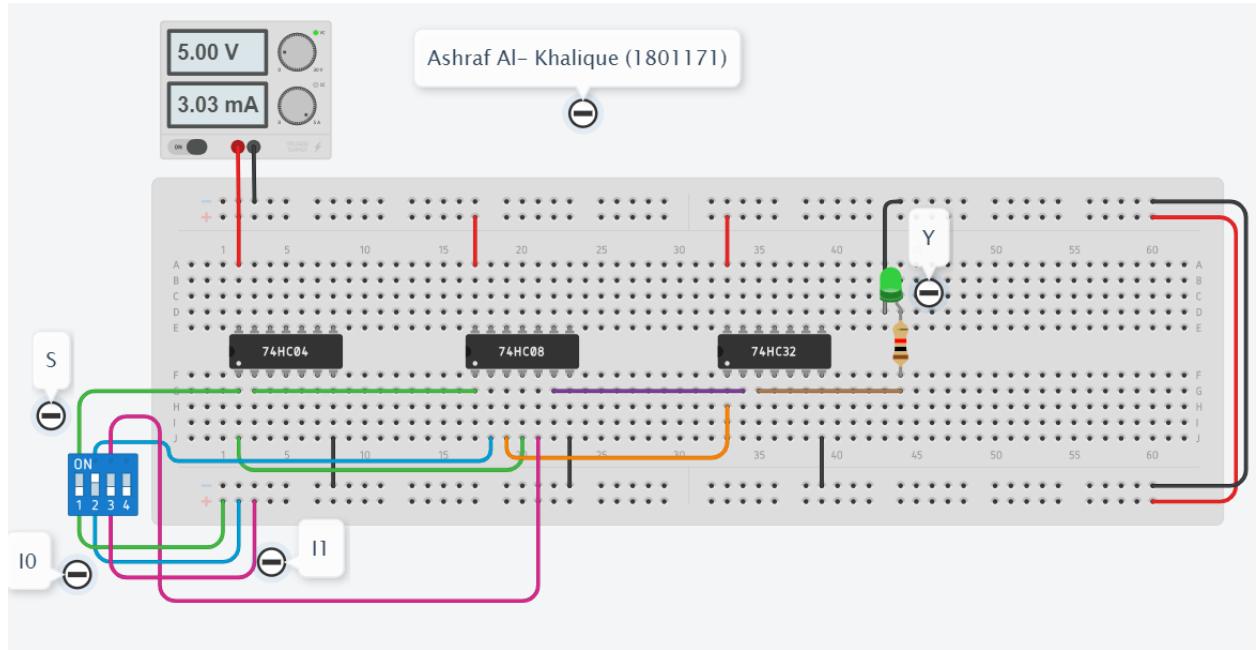
1. When $S=0, I_0=0, I_1=0$, then $Y=0$



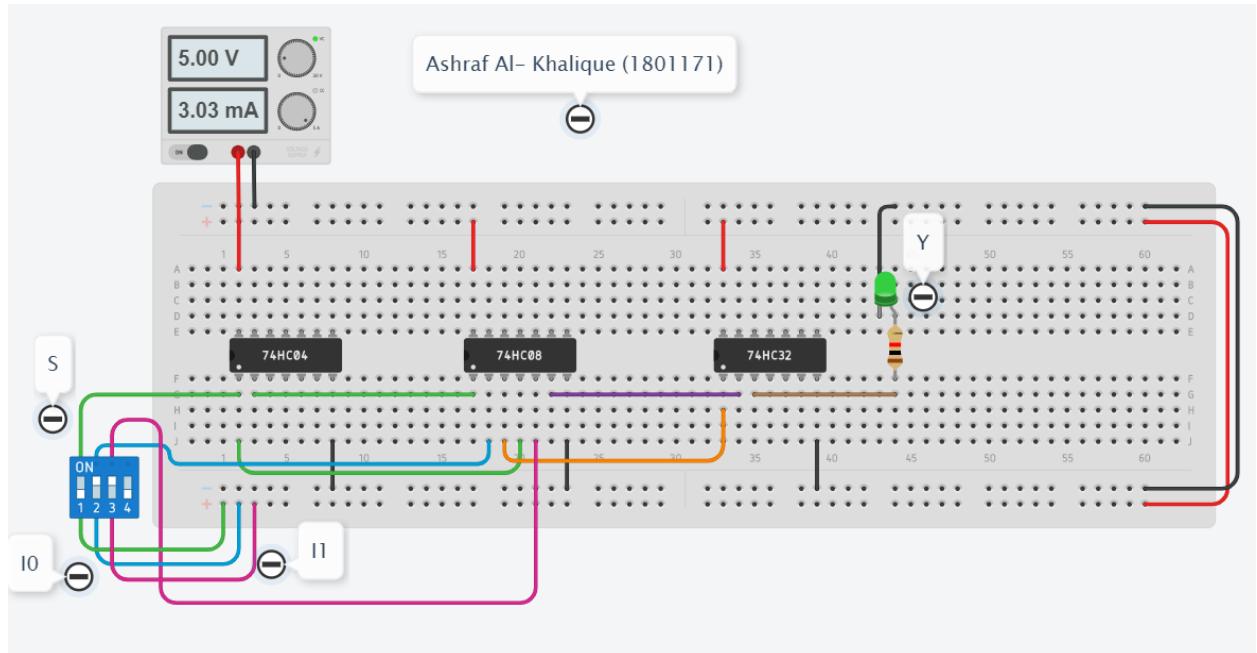
2. When $S=0, I_0=0, I_1=1$, then $Y=0$



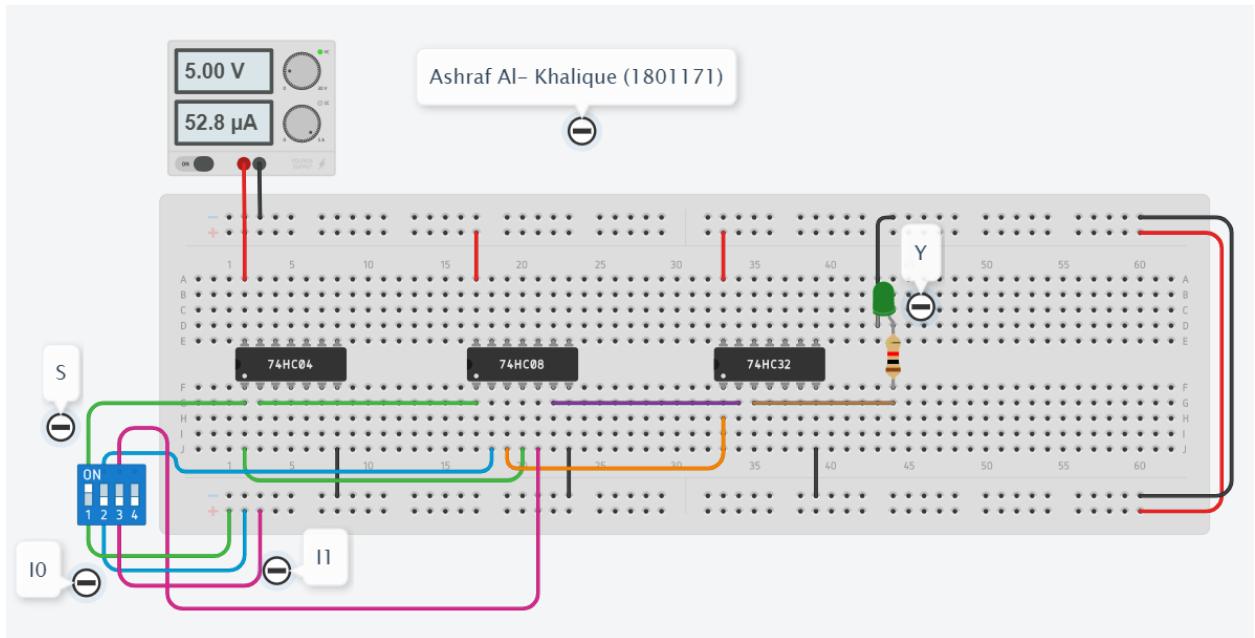
3. When $S=0$, $I_0=1$, $I_1=0$, then $Y=1$



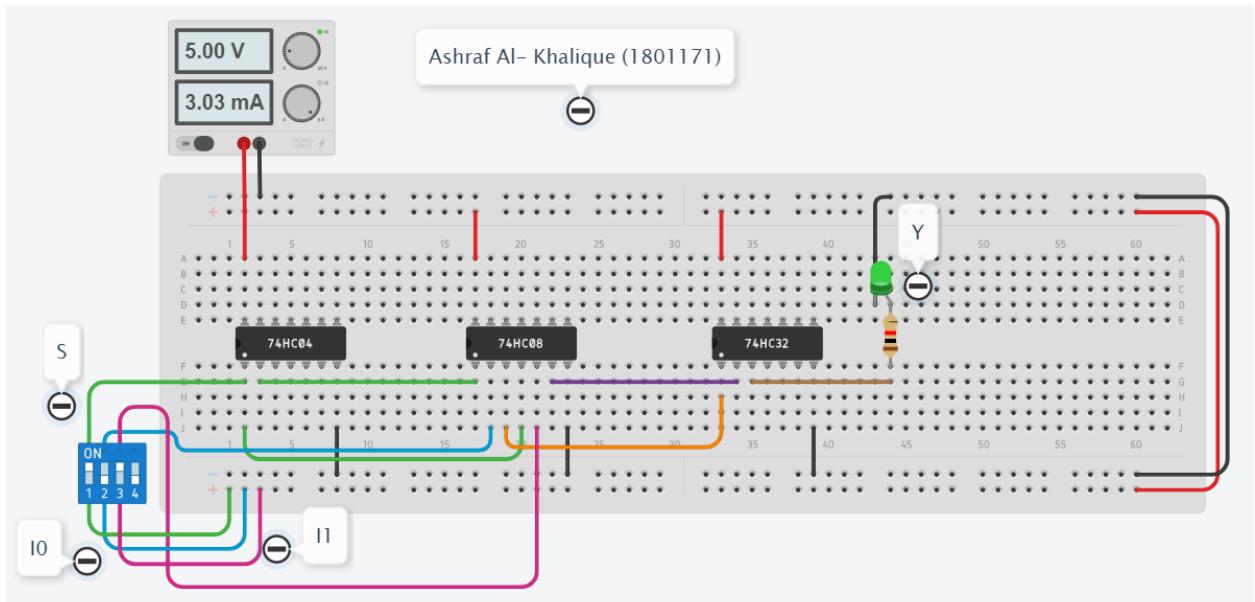
4. When $S=0$, $I_0=1$, $I_1=1$, then $Y=1$



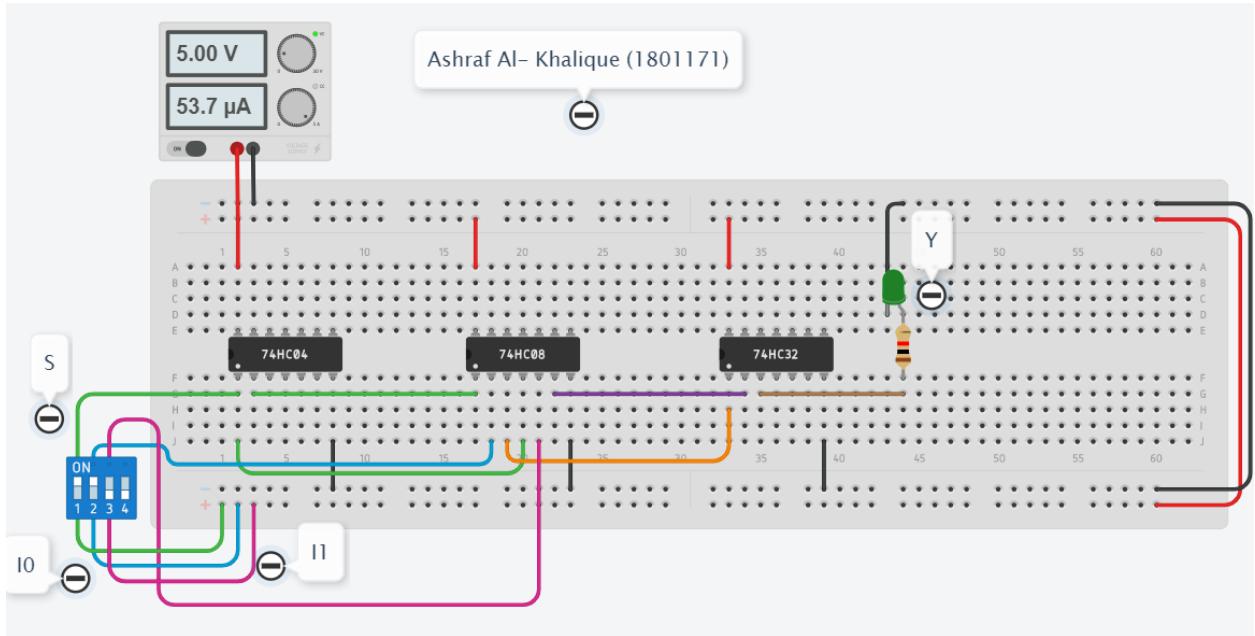
5. When $S=1$, $I_0=0$, $I_1=0$, then $Y=0$



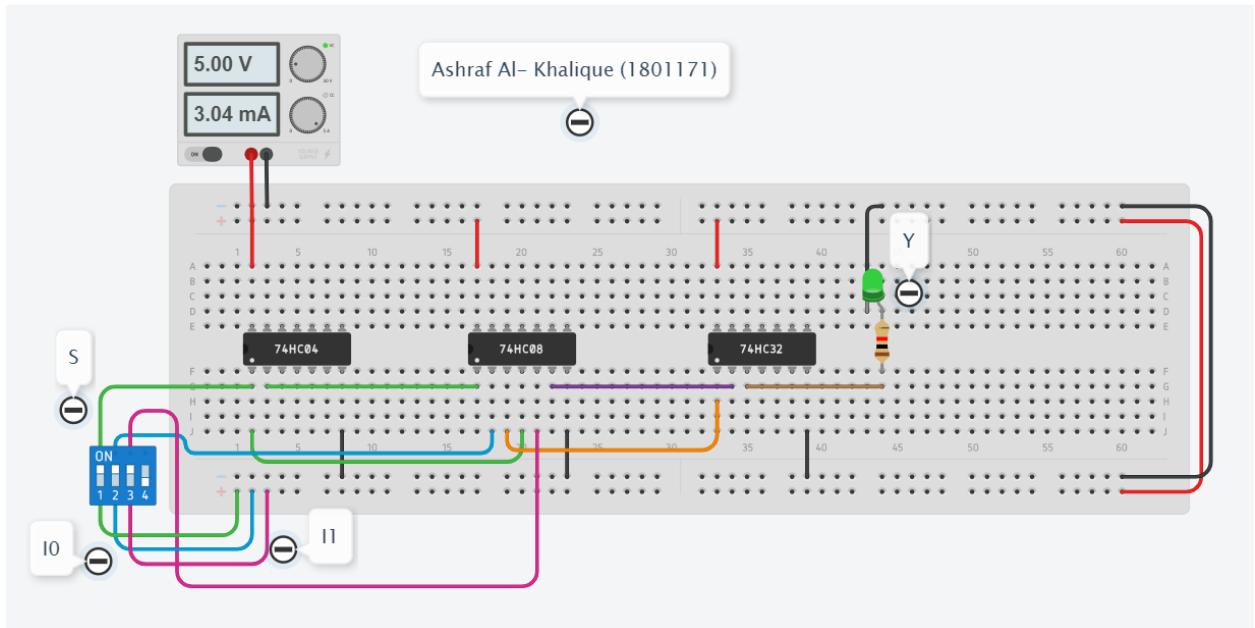
6. When $S=1$, $I_0=0$, $I_1=1$, then $Y=1$



7. When $S= 1, I_0= 1, I_1= 0$, then $Y= 0$



8. When $S= 1, I_0= 1, I_1= 1$, then $Y= 1$



Discussion & Conclusion:

In this experiment when we applied $S= 0$, I_0 LED lit up. Again, when we applied $S= 1$, I_1 LED lit up. Output Y was observed at the same time. As multiplexer is a circuit where the output is connected to either of the inputs depending on the select signal, here are only two ways possible to connect the inputs to the outputs, so one selects is needed to do these operations.

Heaven's light is our guide.

Rajshahi University of Engineering and Technology (RUET)

Department of Electrical & Electronic Engineering

Course no.

EEE2214

Course title:

Digital Electronics I Sessional

Experiment no.

04

Experiment name: Experimental study of a 3-bits comparator circuit.

Submitted to:

Prof. Dr. Md. Shahidul Islam

Professor

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Submitted by:

Ashraf Al- Khalique

Roll: 1801171; Session: 2018-2019

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Date of experiment: August 13, 2021.

Date of submission: August 31, 2021.

Experiment No. 04

Experiment name: Experimental study of a 3-bits comparator circuit.

Objectives: Followings are the primary objective of this experiment,

1. To learn about a 3-bits comparator circuit.
2. To simulate the logic operation of a 3-bits comparator circuit.
3. To design a 3-bits comparator circuit from its basic expression and see it's output.

Required Apparatus:

1. Power supply (1 piece);
2. Quad AND gate (IC no.7408, 3 pieces);
3. Triple- input AND gate (IC no. 7411, 1 piece)
4. Quad OR gate (IC no.7432, 1 piece);
5. Quad NOR gate (IC no.7402, 1 piece);
6. Hex Inverter (IC no. 7404, 1 piece);
7. Slide switch SPST x1 (6 pieces);
8. LED (9 pieces);
9. Resistor ($1k\Omega$, 6 pieces);
10. Simulator (Tinker cad).

Theory:

A comparator is a decision-making tool and it holds the ability to be executed in numerous control devices. Accepting two binary numbers as input (A and B), data comparison through magnitude comparators produces the output to indicate equality ($A=B$), logic 1 in two conditions when ($A>B$ or $A<B$).

Magnitude comparators are mostly utilized in microcontrollers and CPUs to address data comparison, register and perform all other arithmetic operations.

A comparator that compares two binary numbers (each number having 3 bits) and produces three outputs based on the relative magnitudes of given binary bits is called a 3-bit magnitude comparator.

Expressions can be derived as,

$$x_3 = \bar{A}_3 \bar{B}_3 + A_3 B_3$$

$$x_2 = \bar{A}_2 \bar{B}_2 + A_2 B_2$$

$$x_1 = \bar{A}_1 \bar{B}_1 + A_1 B_1$$

$$x_0 = \bar{A}_0 \bar{B}_0 + A_0 B_0$$

$$(A = B) = x_3 x_2 x_1 x_0$$

$$(A > B) = \bar{A}_3 \bar{B}_3 + x_3 \bar{A}_2 \bar{B}_2 + x_3 x_2 \bar{A}_1 \bar{B}_1 + x_3 x_2 x_1 \bar{A}_0 \bar{B}_0$$

$$(A < B) = \bar{A}_3 B_3 + x_3 \bar{A}_2 B_2 + x_3 x_2 \bar{A}_1 B_1 + x_3 x_2 x_1 \bar{A}_0 B_0$$

$A_3 A_2 A_1 A_0 \ B_3 B_2 B_1 B_0$

↓ ↓ ↓ ↓ ↓ ↓

Magnitude
Comparator

↓ ↓ ↓

$A < B \ A = B \ A > B$

Truth Table:

A	Decimal	B	Decimal	A=B	A>B	A<B
000	0	111	7	0	0	1
001	1	110	6	0	0	1
100	4	011	3	0	1	0
101	5	010	2	0	1	0
011	3	011	3	1	0	0
010	2	010	2	1	0	0

Circuit Diagram:

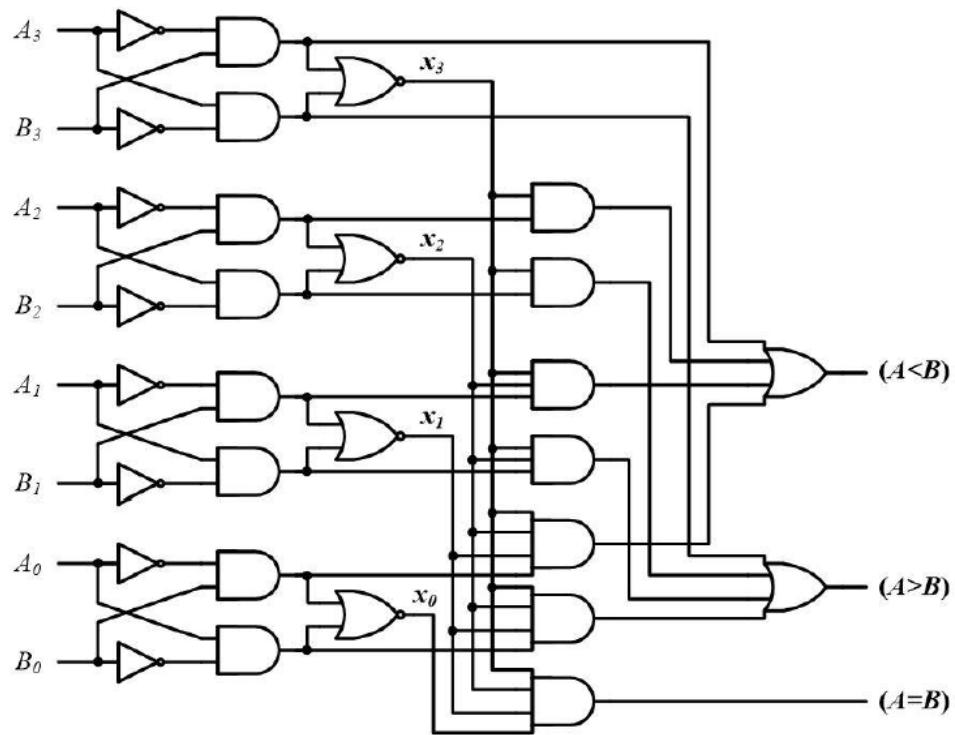


Fig 4.1: Logic Diagram of a 3-bits comparator circuit.

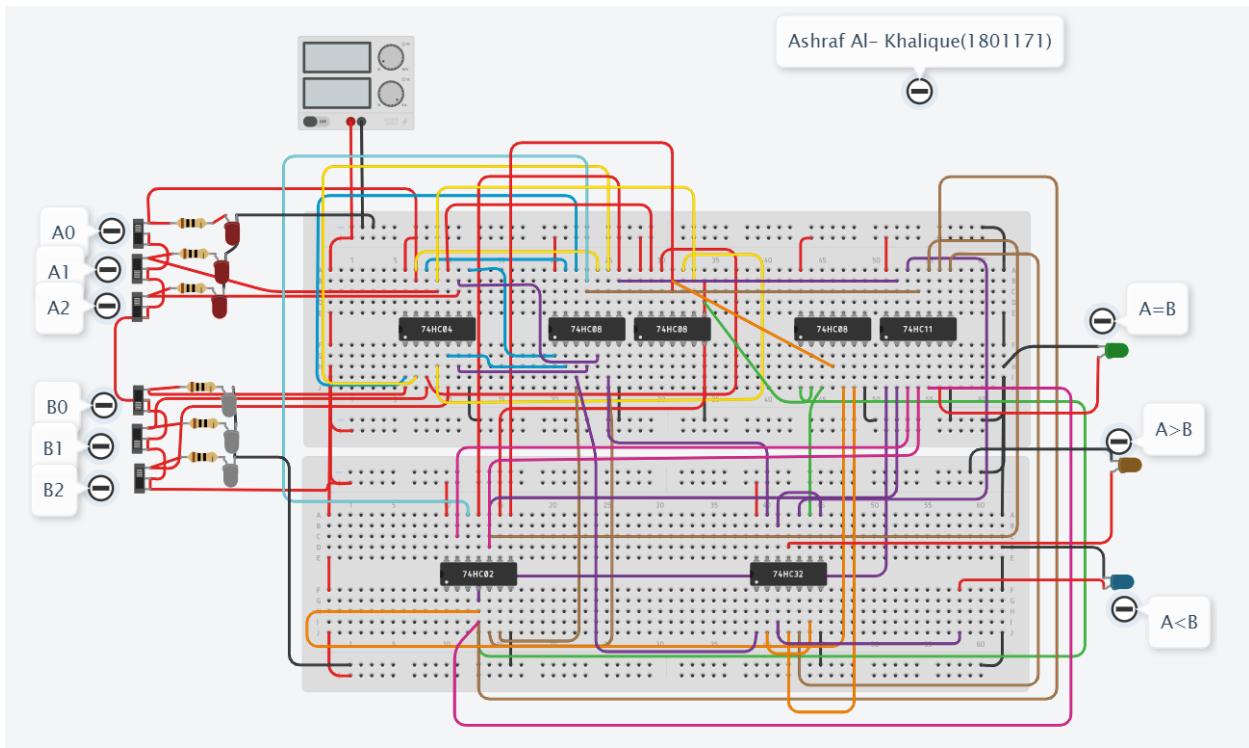
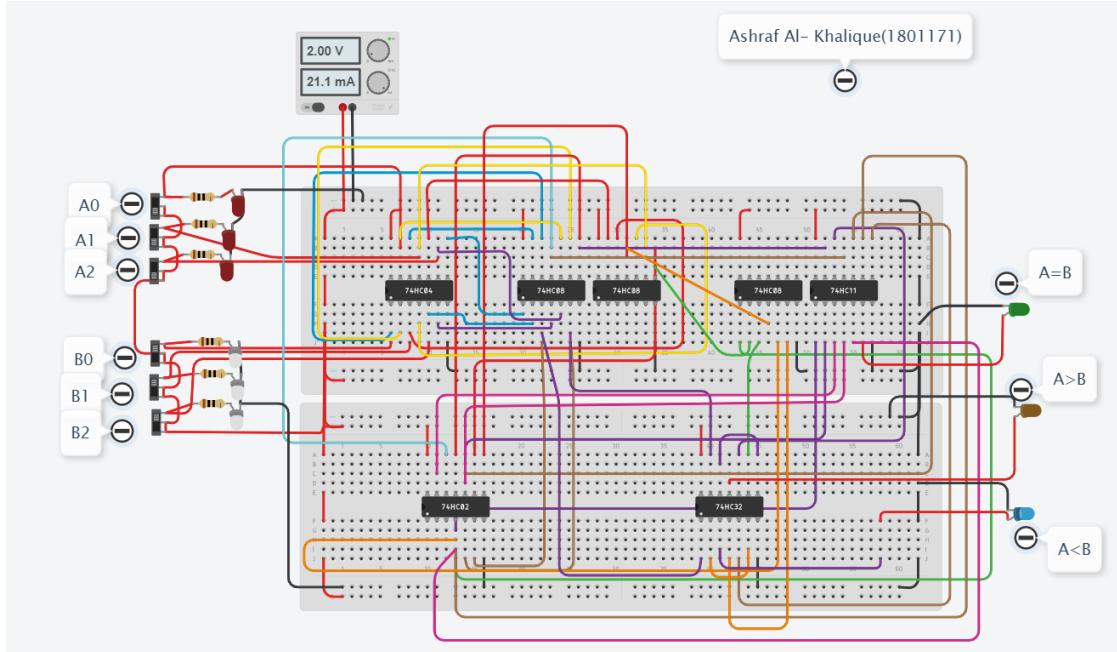


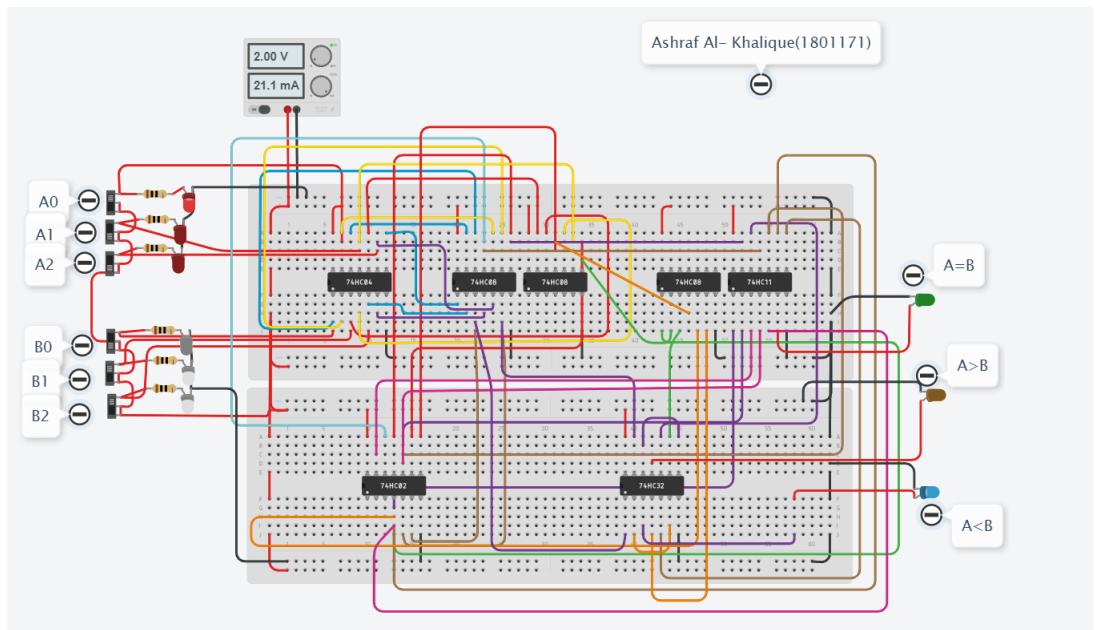
Fig 4.2: Circuit Diagram of a 3-bits comparator circuit.

Output:

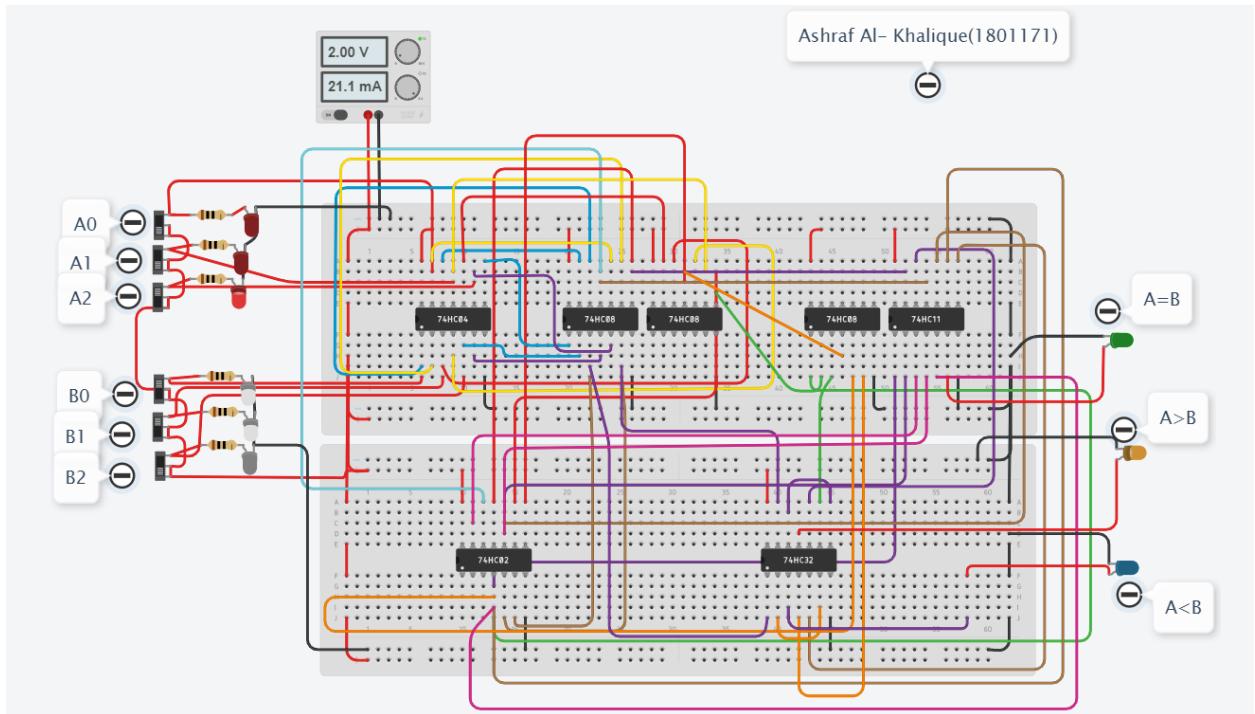
1. When A (A₂, A₁, A₀) = 000, B (B₂, B₁, B₀) = 111, then A < B



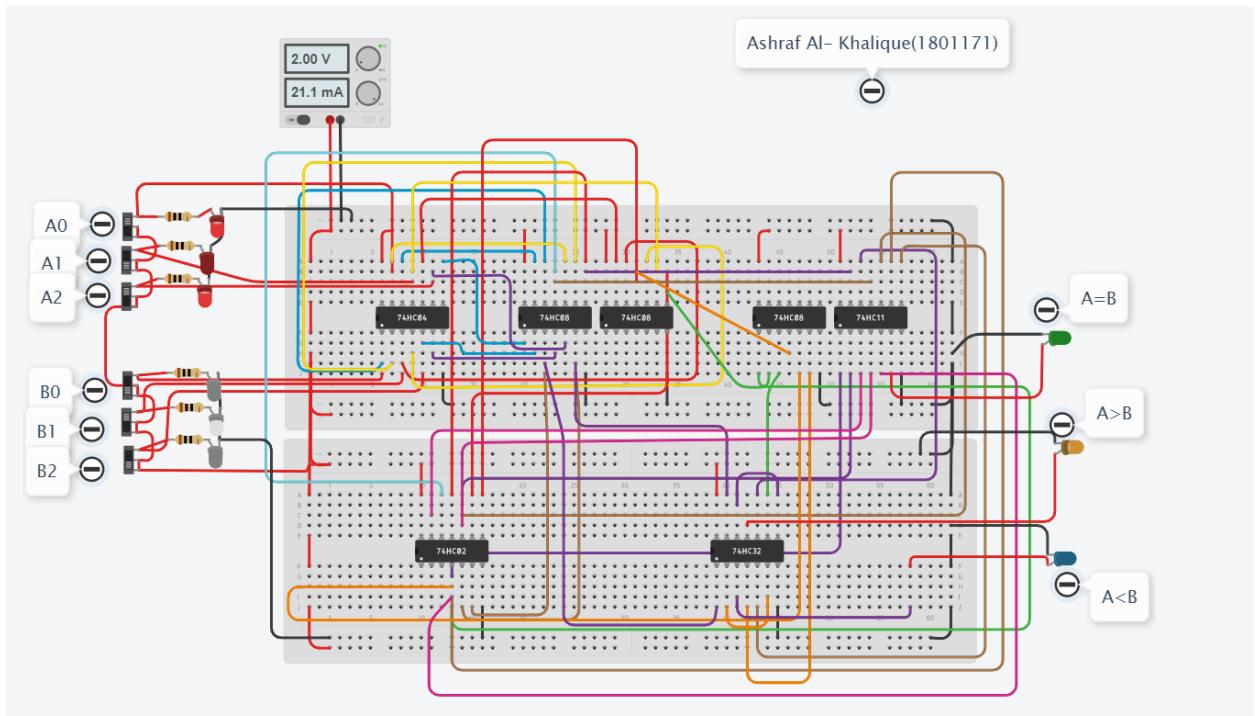
2. When A (A₂, A₁, A₀) = 001, B (B₂, B₁, B₀) = 110, then A < B



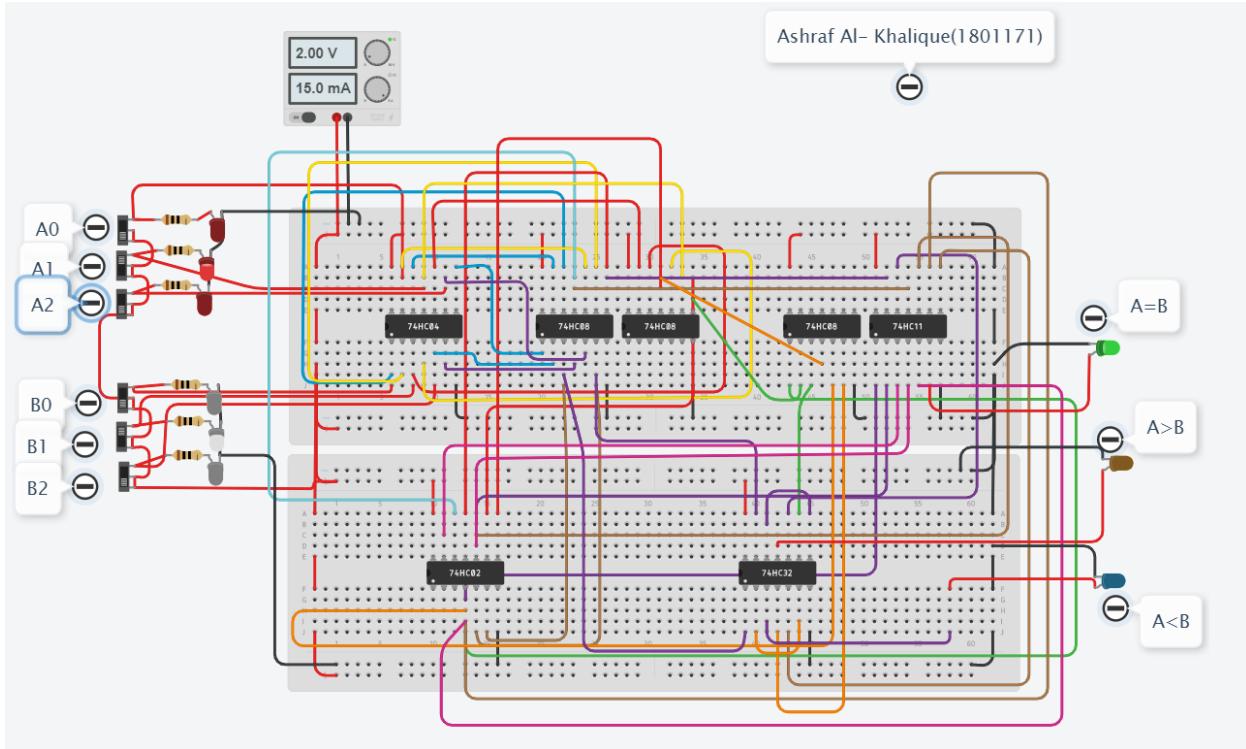
3. When A (A₂, A₁, A₀) = 100, B (B₂, B₁, B₀) = 011, then A>B



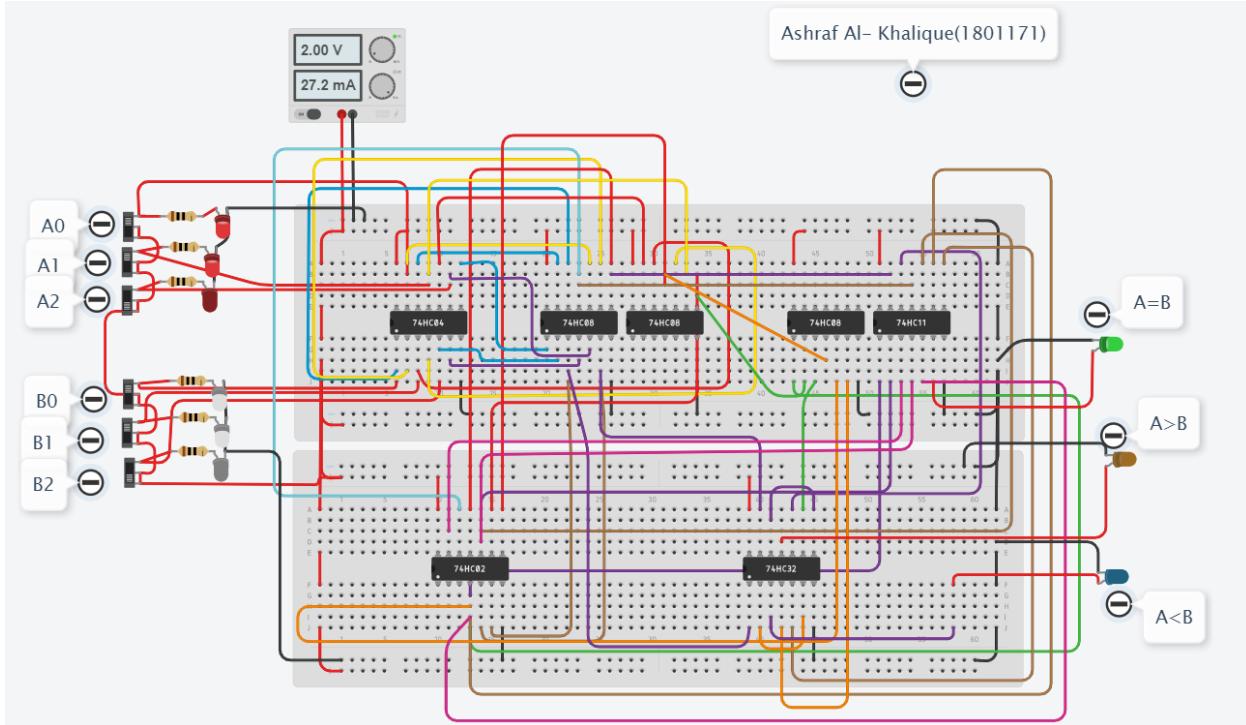
4. When A (A₂, A₁, A₀) = 101, B (B₂, B₁, B₀) = 010, then A>B



5. When A (A₂, A₁, A₀) = 010, B (B₂, B₁, B₀) = 010, then A=B



6. When A (A₂, A₁, A₀) = 011, B (B₂, B₁, B₀) = 011, then A=B



Discussion & Conclusion:

In this circuit, the designed logic circuit compares between 3- bits number. Here we gave multiple 3- bits input for both A and B, take 2 of each input and compared them. As a result, when we gave input 000 in A and 111 in B, the output LED A<B lit up. We took another data input for similar output as well.

Again, when we gave input 100 in A and 011 in B, the output LED A>B lit up. We took another data input for similar output as well.

Lastly, when we gave input 010 in A and 010 in B, the output LED A=B lit up. We took another data input for similar output as well. Thus, we can conclude that the experiment was done correctly.

Heaven's light is our guide.

Rajshahi University of Engineering and Technology (RUET)

Department of Electrical & Electronic Engineering

Course no.

EEE2214

Course title: Digital Electronics I Sessional

Experiment no.

05

Experiment name: Experimental study of adder & sub tractor circuit using 4-bits adder IC.

Submitted to:

Prof. Dr. Md. Shahidul Islam

Professor

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Submitted by:

Ashraf Al- Khalique

Roll: 1801171; Session: 2018-2019

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Date of experiment: August 13, 2021.

Date of submission: August 31, 2021.

Experiment No. 05

Experiment name: Experimental study of adder & sub tractor circuit using 4-bits adder IC.

Objectives: Followings are the primary objective of this experiment,

1. To learn about adder & sub tractor circuit using 4-bits adder IC.
2. To simulate the logic operation of adder & sub tractor circuit using 4-bits adder IC.
3. To design adder & sub tractor circuit using 4-bits adder IC from its basic expression and see it's output.

Required Apparatus:

1. Power supply (1 piece);
2. 4-bits adder (IC no.74283, 1 piece);
3. Quad XOR gate (IC no.7486, 1 piece);
4. Dip switch SPST x4 (2 pieces);
5. Dip switch DPST x1 (1 piece);
6. LED (9 pieces);
7. Resistor (1kΩ, 23 pieces);
8. Simulator (Tinker cad).

Theory:

In Digital Circuits, A Binary Adder-Subtractor is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value the control signal holds. It is one of the components of the ALU (Arithmetic Logic Unit).

This Circuit Requires prerequisite knowledge of Exor Gate, Binary Addition and Subtraction, Full Adder.

An adder & sub tractor circuit consists of 4 full adders since we are performing operation on 4-bit numbers. There is a control line M that holds a binary value of either 0 or 1 which determines that the operation being carried out is addition or subtraction.

$$M=0 \rightarrow F = x + y$$

$$M=1 \rightarrow F = x - y$$

Truth Table:

A	B	Carry	Summation	Borrow	Difference
A_3, A_2, A_1, A_0	B_3, B_2, B_1, B_0				
0000	0000	0	0000	1	0000
0001	0111	0	1000	0	1010
0011	1010	0	1001	0	1011
0110	1000	0	1110	0	1110
1011	1111	1	1010	0	1100
1100	1001	1	0101	1	0011
1111	1111	1	1110	1	0000

Circuit Diagram:

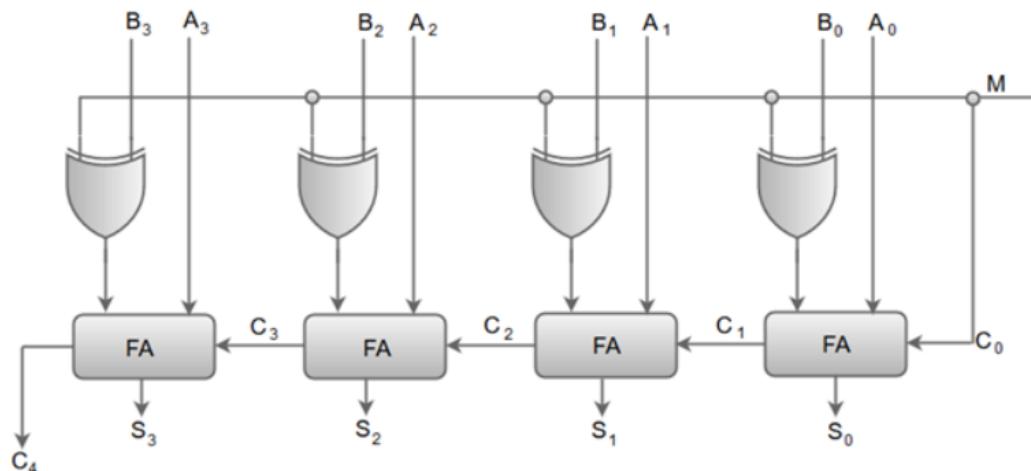


Fig 5.1: Logic Diagram of adder & sub tractor circuit using 4-bits adder IC.

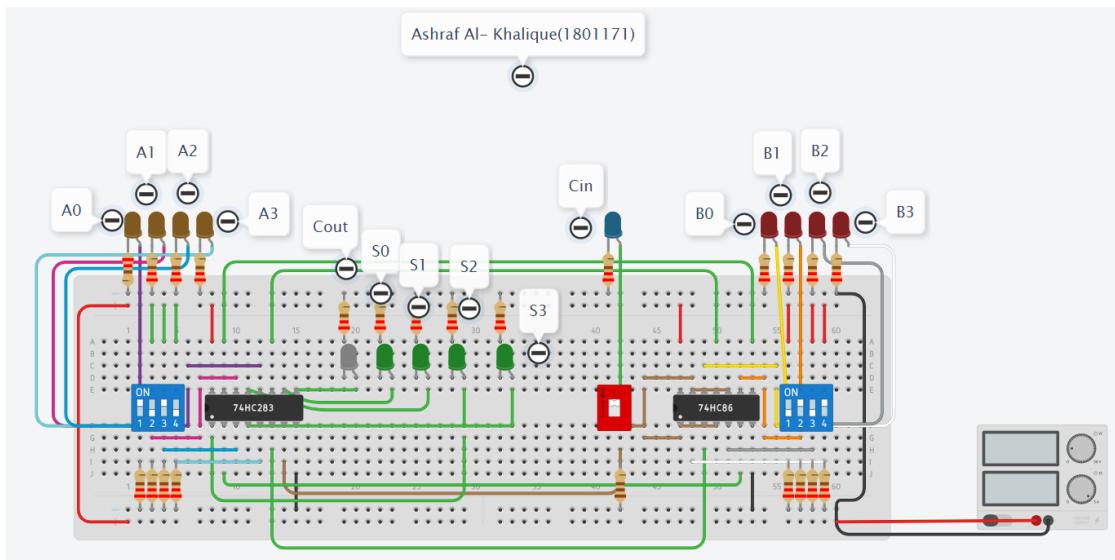
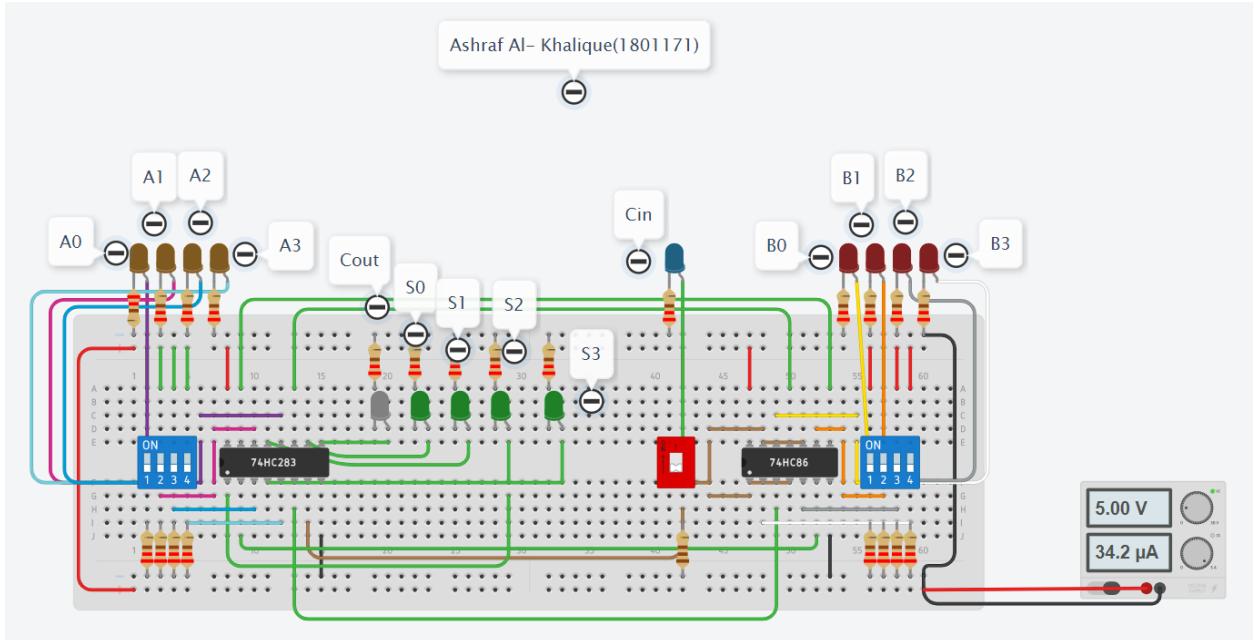
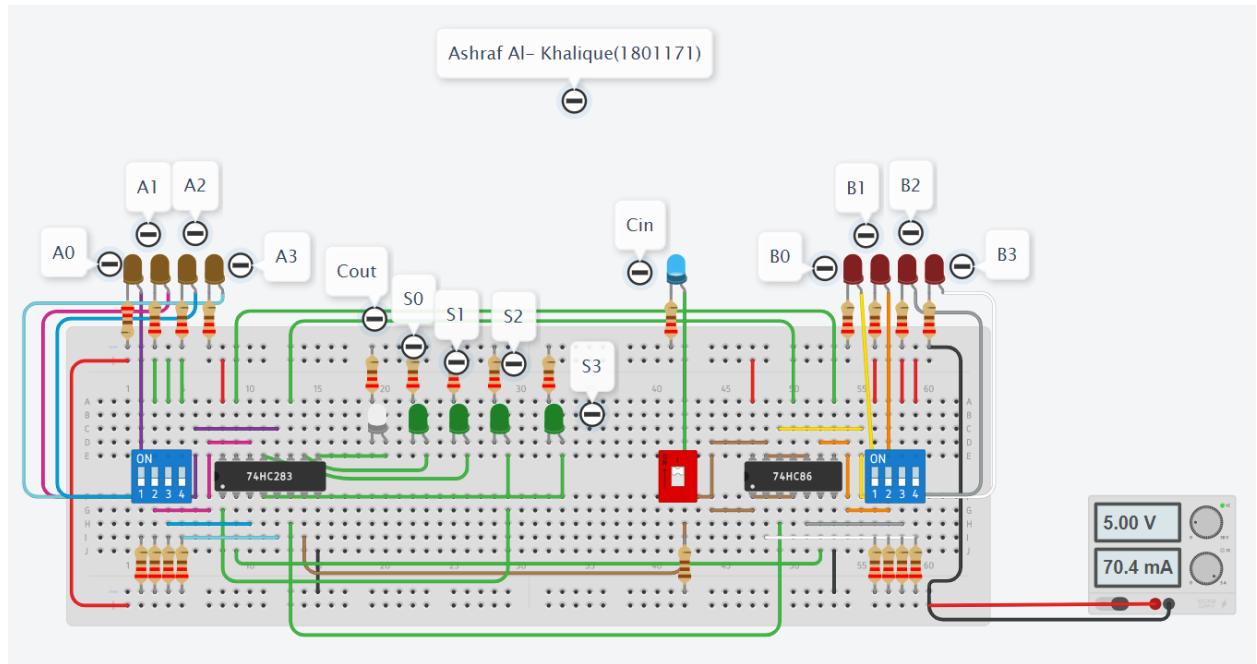


Fig 5.2: Logic Diagram of adder & sub tractor circuit using 4-bits adder IC.

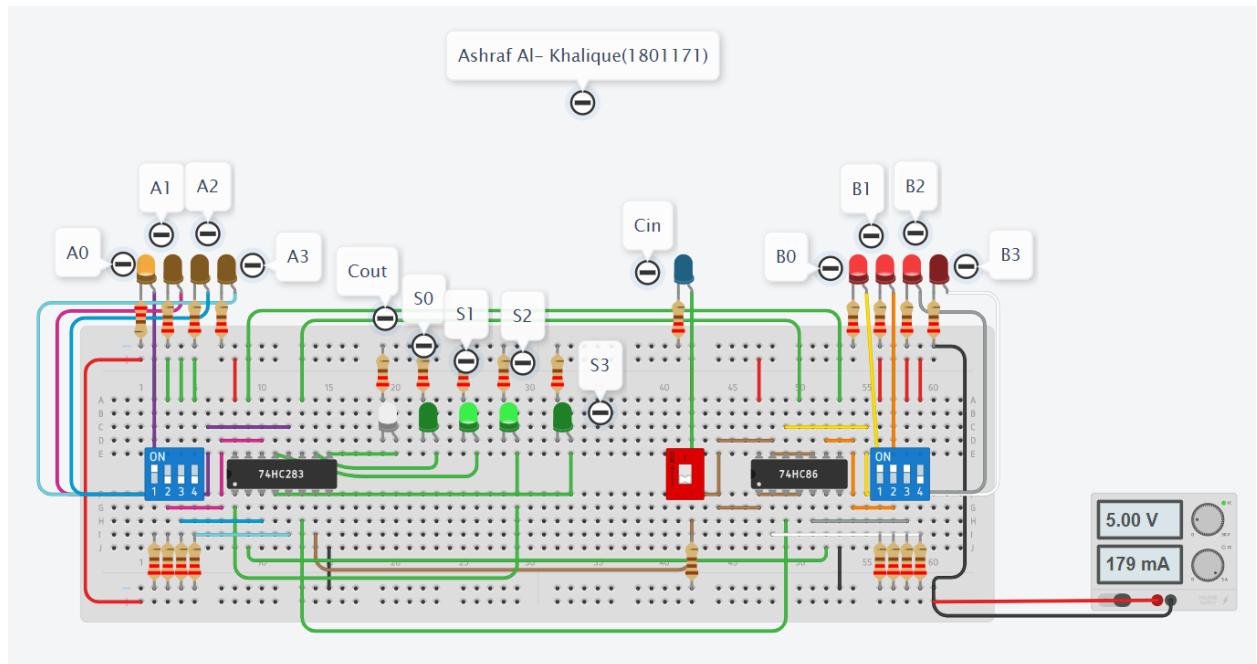
Output:

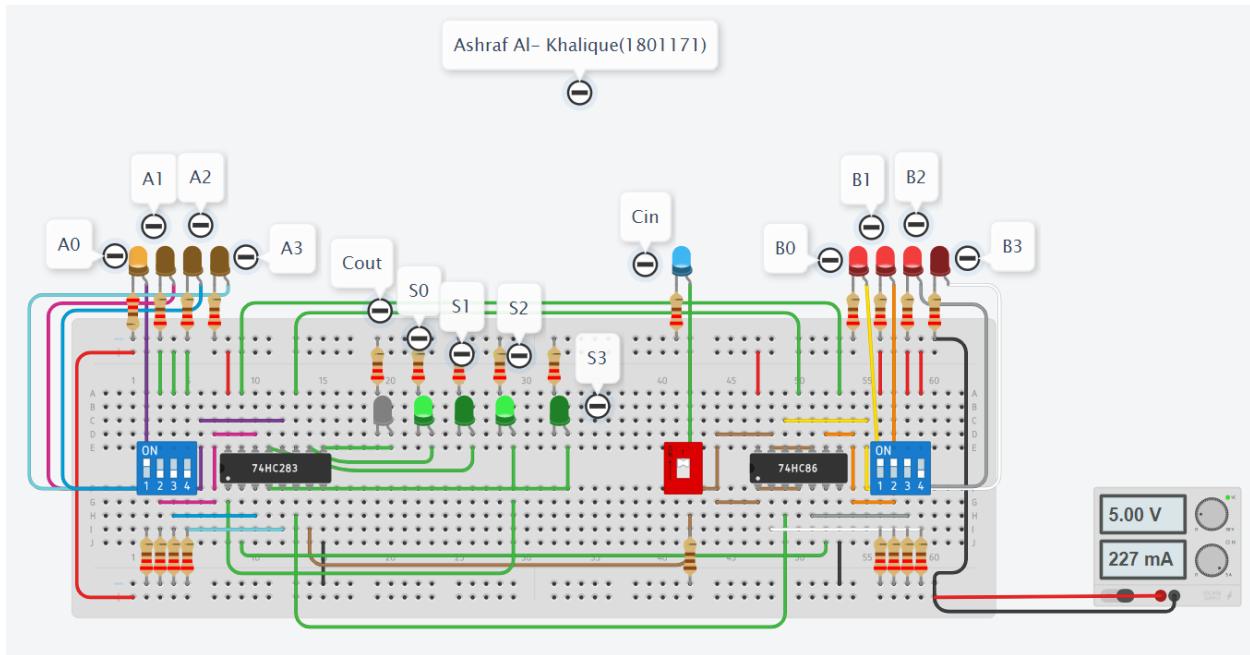
- When A (A₃, A₂, A₁, A₀) = 0000, B (B₃, B₂, B₁, B₀) = 0000, Carry= 0 then
Summation= 0000, Borrow= 1 then difference= 0000



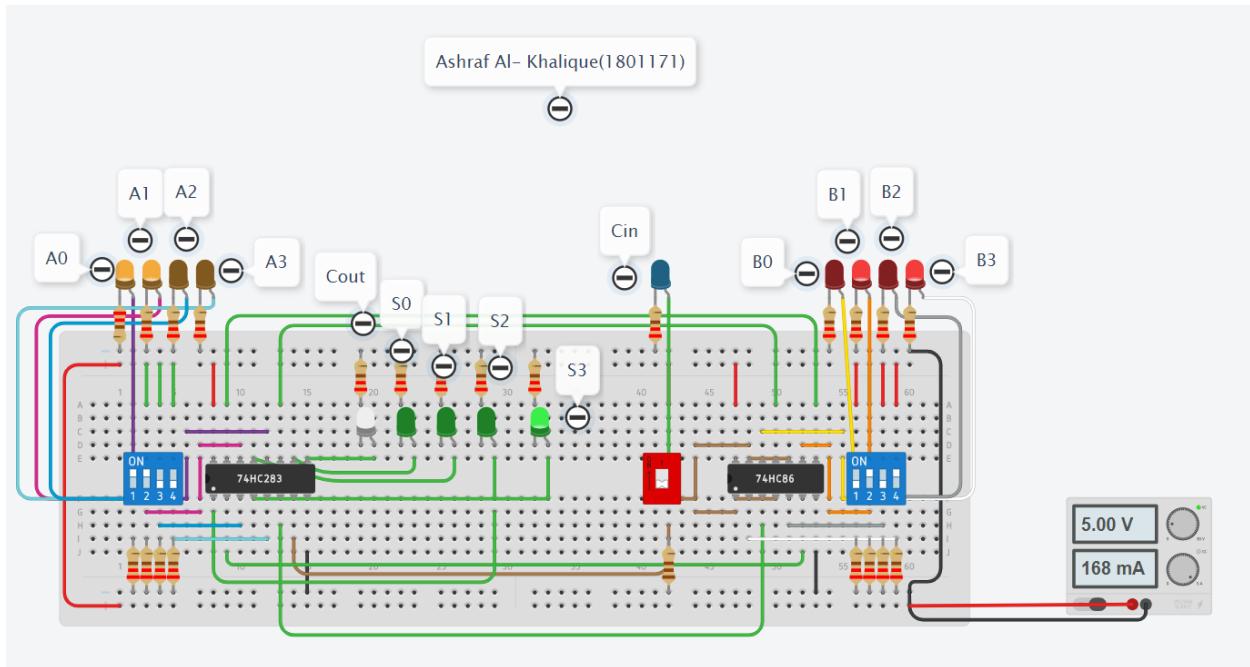


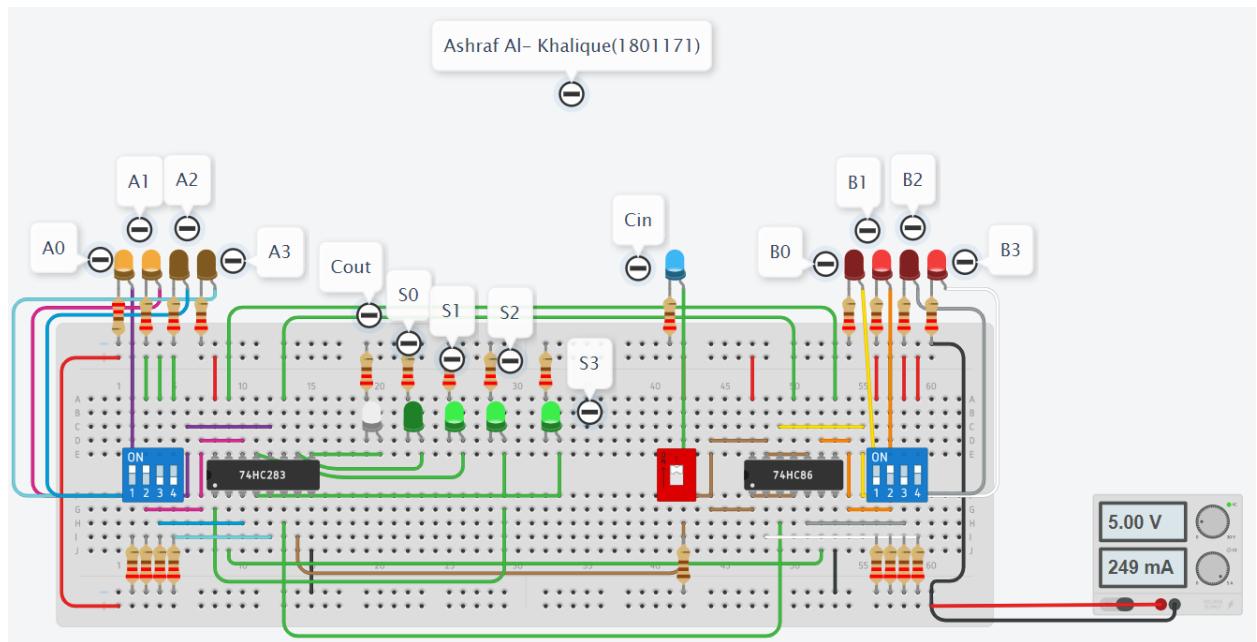
2. When A (A_3, A_2, A_1, A_0) = 0001, B (B_3, B_2, B_1, B_0) = 0111, Carry= 0 then
 Summation= 1000, Borrow= 0 then difference= 1010



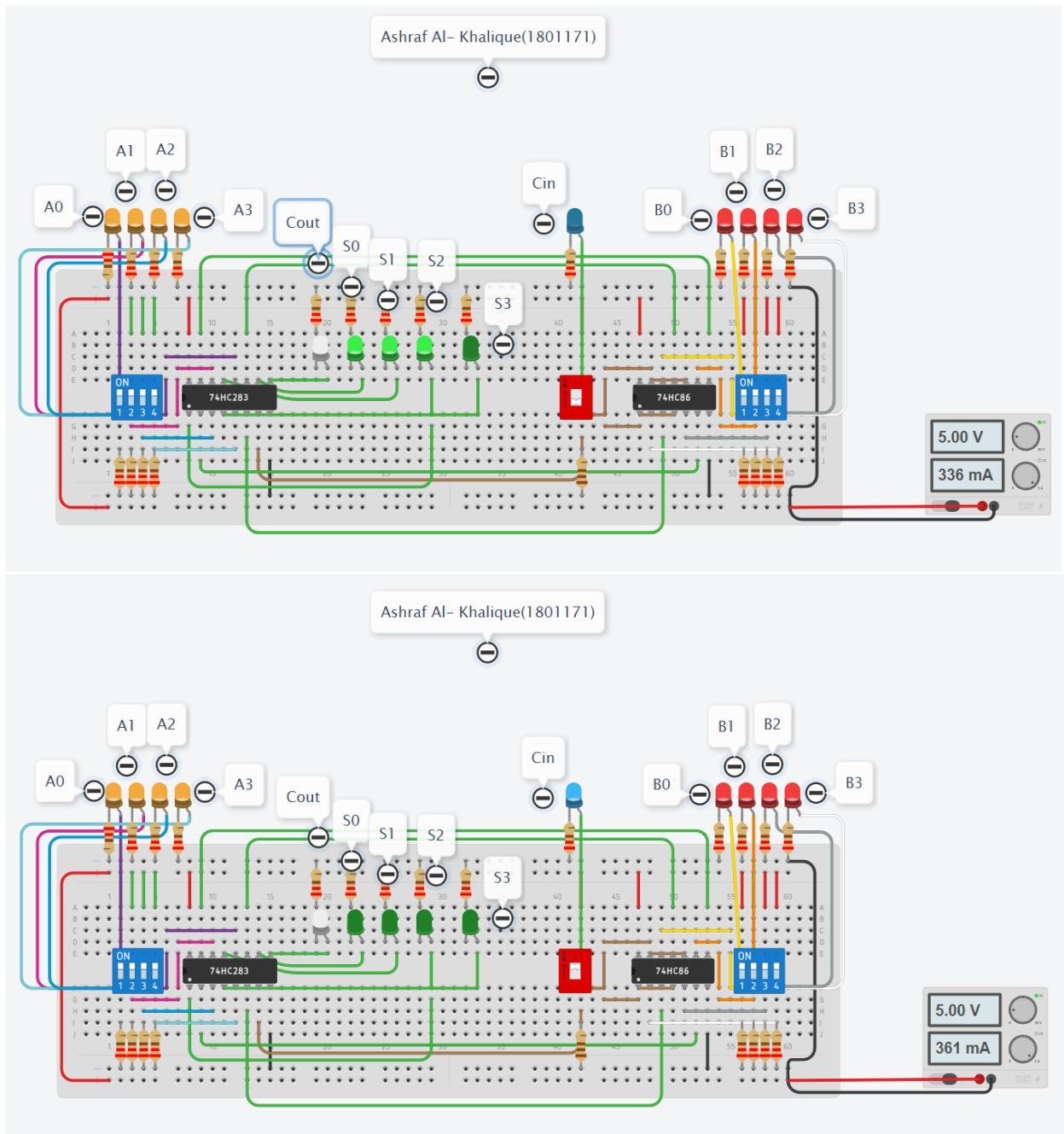


3. When A (A_3, A_2, A_1, A_0) = 0011, B (B_3, B_2, B_1, B_0) = 1010, Carry= 0 then
 Summation= 1001, Borrow= 0 then difference= 1011





4. When A (A₃, A₂, A₁, A₀) = 1111, B (B₃, B₂, B₁, B₀) = 1111, Carry= 1 then
Summation= 1110, Borrow= 1 then difference= 0000



Discussion & Conclusion:

In this experiment, we used 4 –bits adder IC which adds 4-bits numbers.

When it is in M=0 mode, it will add the input 4-bits. When the circuit is in M=1 mode, it will work as a sub tractor circuit. When we gave input of 0000 and 0111, at M=0, it gave us output of 1000 and carry was 0. When M=1, it subtracted and gave us output of 1010 and borrow bit was 0. Thus, it was concluded, the circuit was running correctly.

Heaven's light is our guide.

Rajshahi University of Engineering and Technology (RUET)

Department of Electrical & Electronic Engineering

Course no.

EEE2214

Course title: Digital Electronics I Sessional

Experiment no.

06

Experiment name: Experimental study of MOD 8 asynchronous up/down counter.

Submitted to:

Prof. Dr. Md. Shahidul Islam

Professor

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Submitted by:

Ashraf Al- Khalique

Roll: 1801171; Session: 2018-2019

Dept. of Electrical & Electronic Engineering,
Rajshahi University of Engineering and Technology.

Date of experiment: August 13, 2021.

Date of submission: August 31, 2021.

Experiment No. 06

Experiment name: Experimental study of MOD 8 asynchronous up/down counter.

Objectives: Followings are the primary objective of this experiment,

1. To learn about MOD 8 asynchronous up/down counter.
2. To simulate the logic operation of MOD 8 asynchronous up/down counter.
3. To design MOD 8 asynchronous up/down counter from its basic expression and see it's output.

Required Apparatus:

1. Power supply (1 piece);
2. Quad AND gate (IC no.7408, 1 piece);
3. Dual JK Flip flop gate (IC no. 7473, 2 pieces)
4. Quad OR gate (IC no.7432, 1 piece);
5. Hex Inverter (IC no. 7404, 1 piece);
6. Dip switch SPST x4 (1 piece);
7. LED (4 pieces);
8. Resistor ($1k\Omega$, 3 pieces);
9. Simulator (Tinker cad).

Theory:

Counters are sequential logic devices that are activated or triggered by an external timing pulse or clock signal.

Modulus Counters, or simply MOD counters, are defined based on the number of states that the counter will sequence through before returning back to its original value. counters can be designed to count to any number of 2^n states in their sequence by cascading together multiple counting stages to produce a single modulus or MOD-N counter.

Therefore, a “Mod-N” counter will require “N” number of flip-flops connected together to count a single data bit while providing 2^n different output states, (n is the number of bits). Note that N is always a whole integer value.

We can therefore construct mod counters to have a natural count of $2n$ states giving counters with mod counts of 2, 4, 8, 16, and so on, before repeating itself.

Truth Table:**1. Up Counter:**

S	Q_3	Q_2	Q_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

2. Down Counter:

S	Q_3	Q_2	Q_1
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0

Circuit Diagram:

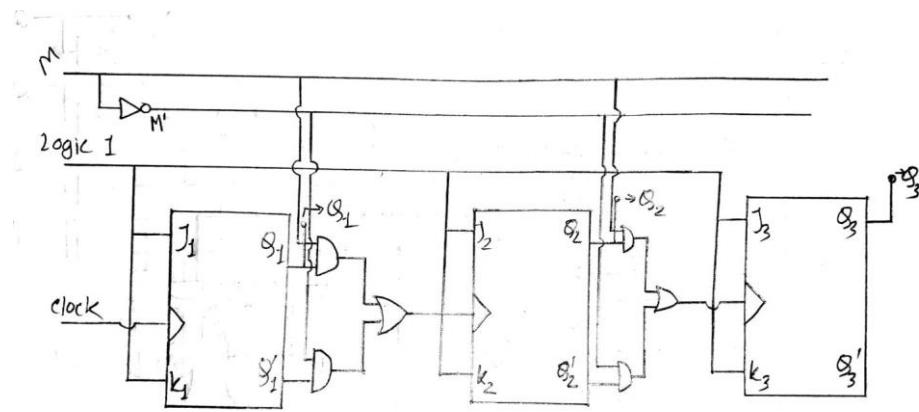


Fig 5.1: Logic Diagram of adder & sub tractor circuit using 4-bits adder IC.

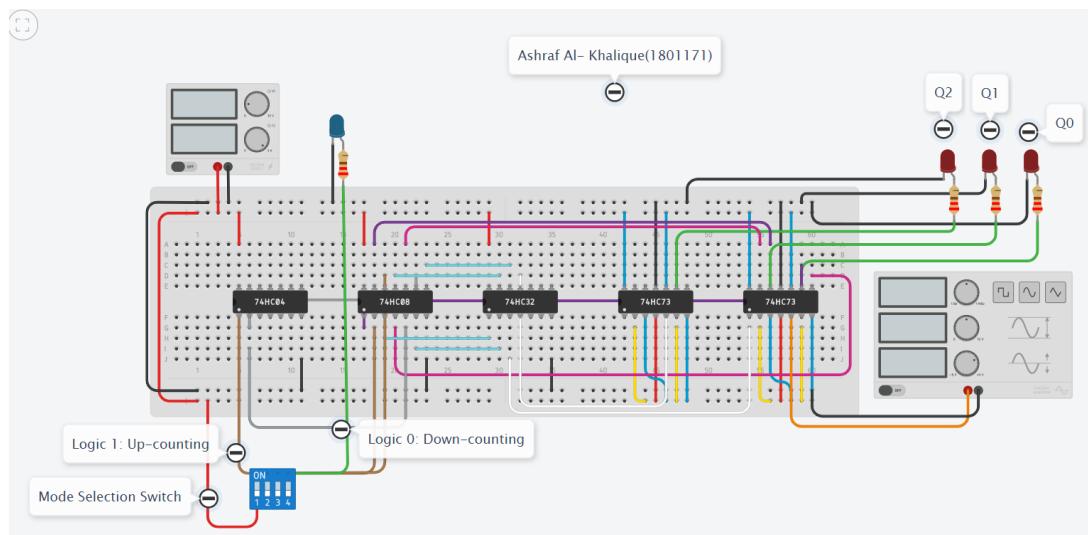
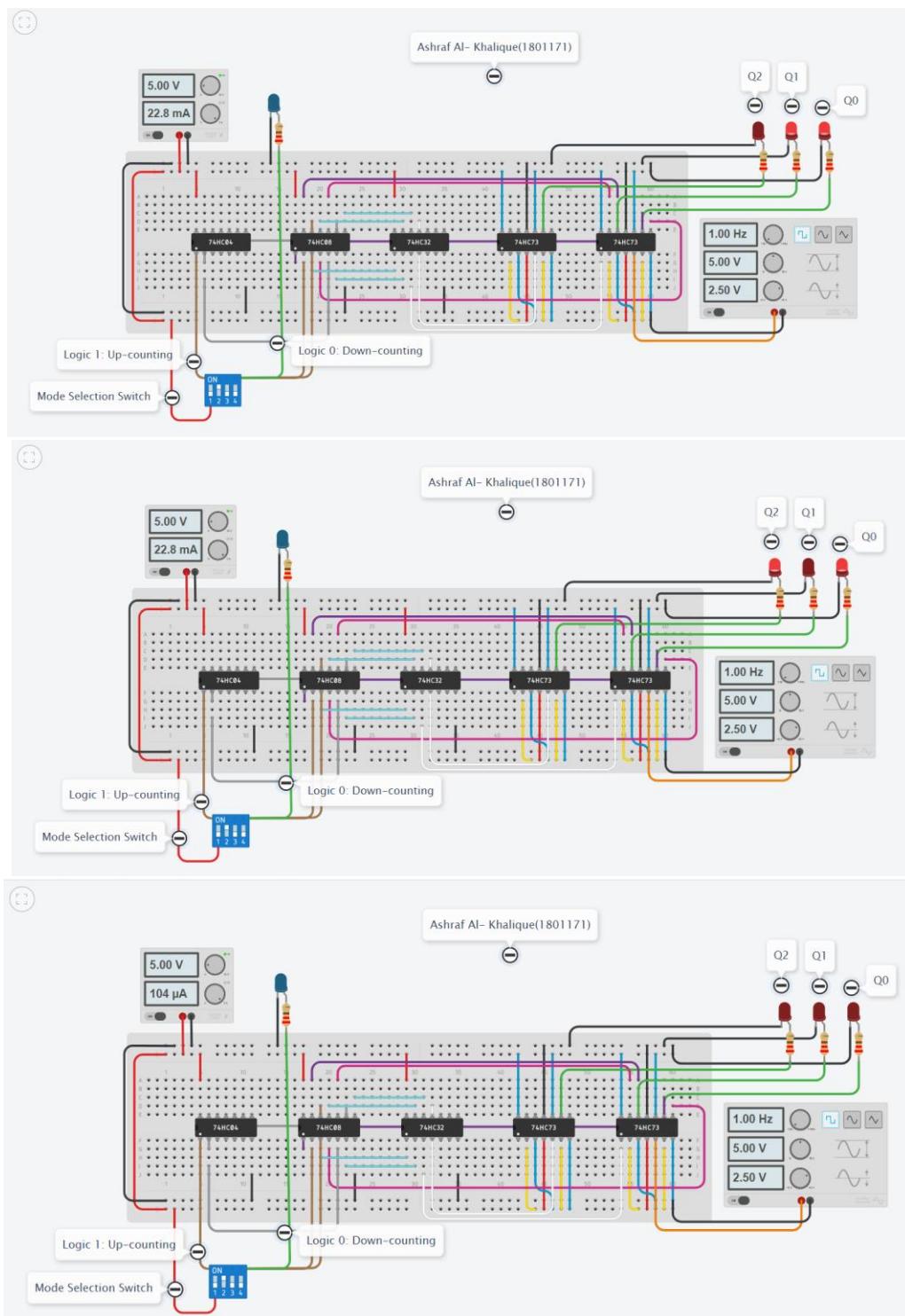
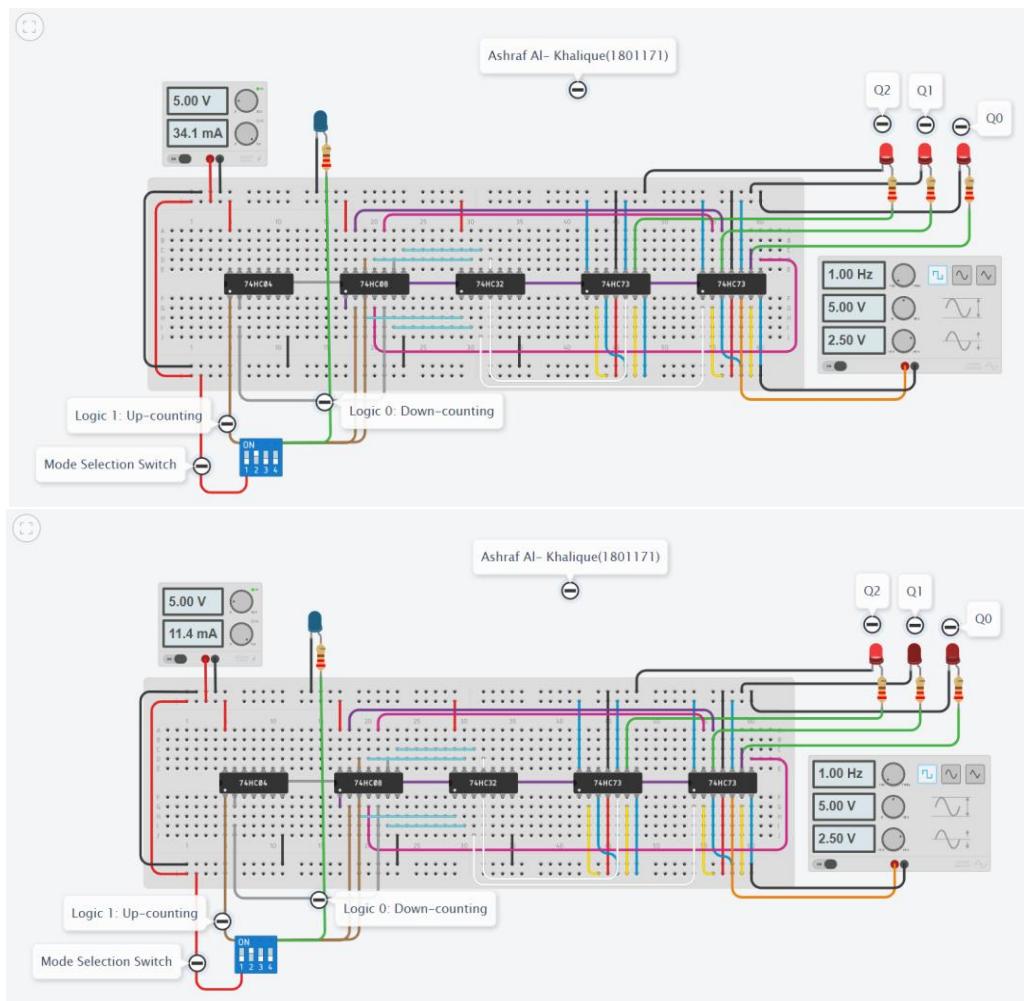
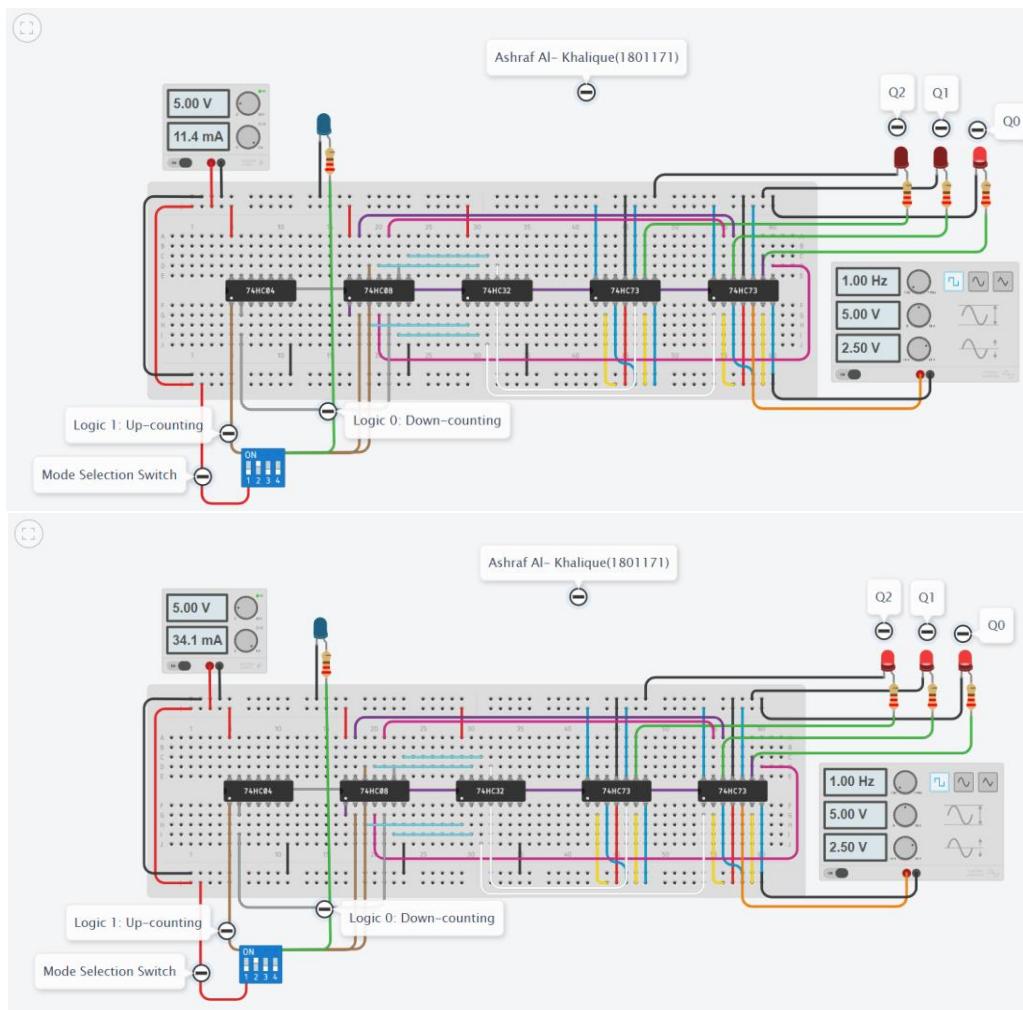


Fig 5.2: Logic Diagram of adder & sub tractor circuit using 4-bits adder IC.

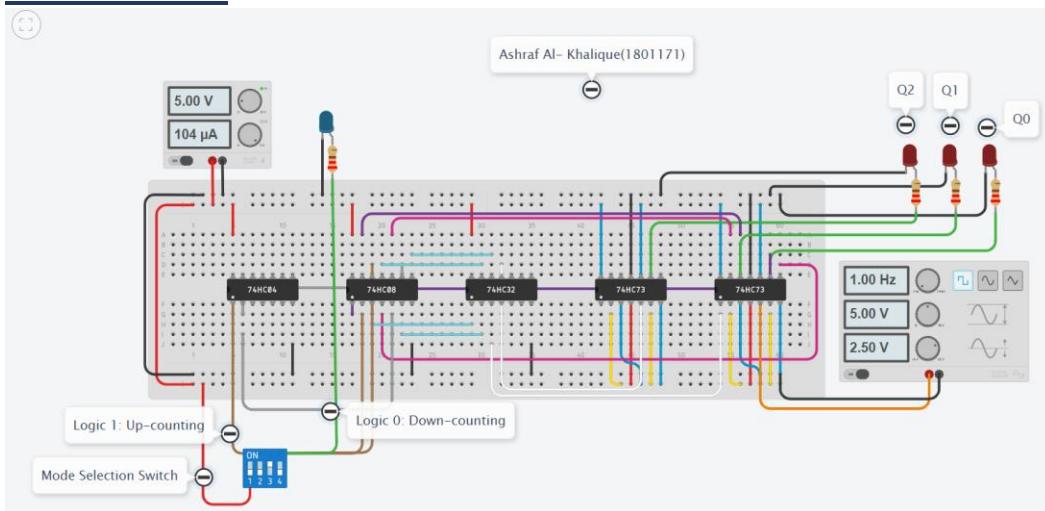
Output: Up Counter:

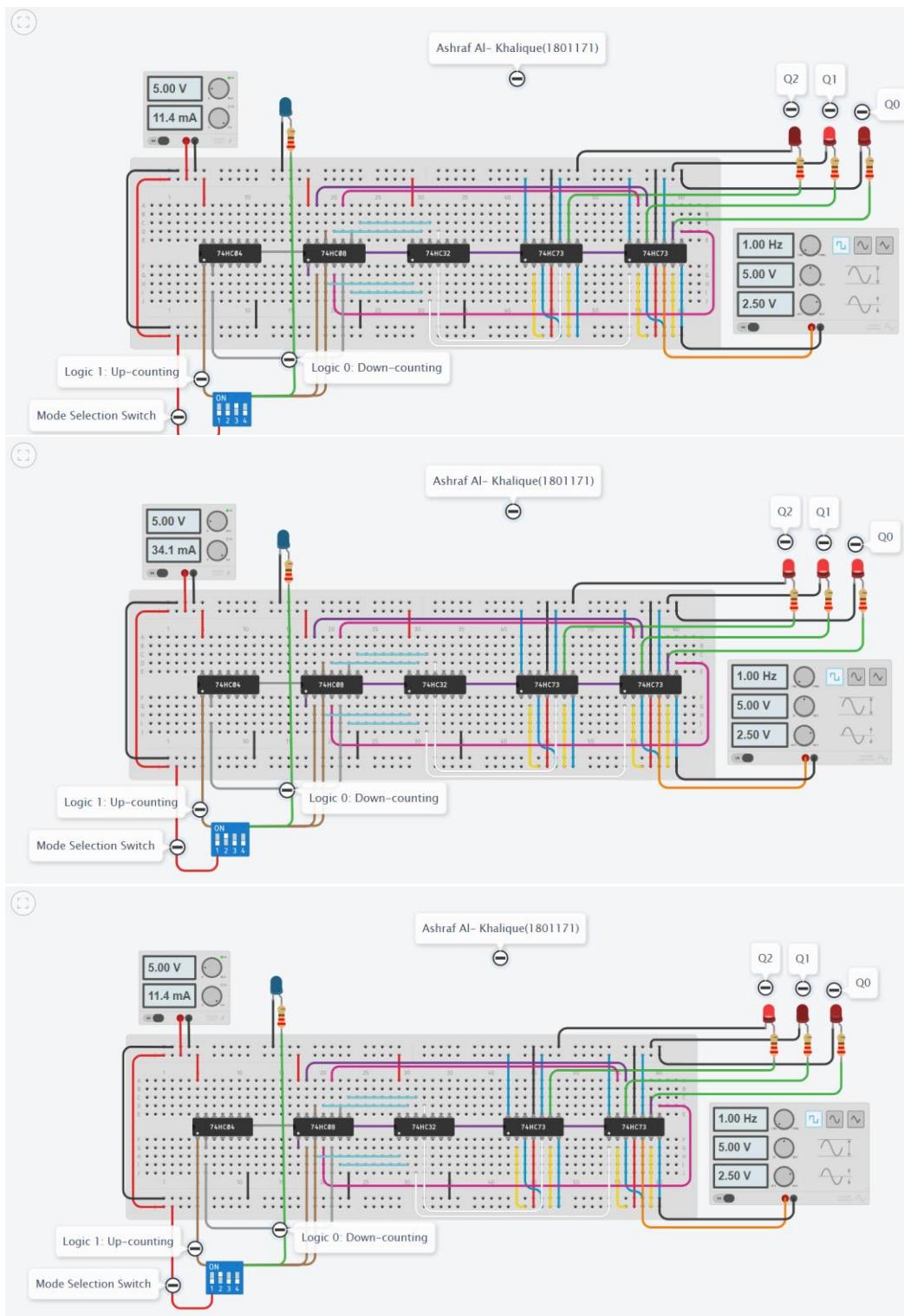






Down Counter:





Discussion & Conclusion:

In this experiment, we wanted an asynchronous counter that counts according to the clock signal given as input. A 3-bit up counter counts 0-7 and a 3-bit down

counter counts 7-0 asynchronously. In this experiment, we used J-K flip flop to make such counter.

In this circuit, when $M=1$, it counted from 0-7 asynchronously which is the output of an up counter. When $M=0$, it counted from 7-0 asynchronously which is the output of a down counter. Thus, it can be said that the circuit was working correctly.