Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Digital Electronics And Computer Organization Laboratory

ENCS2110

Report #1

Experiment No. 2 - Comparators, Adders, and Subtractors

Prepared by :

Mohammed Jamil Saada **Number : 1221972**

Partner :

Laith Nimer **Number : 1213046**

Instructor : **Dr. Abualseoud Hanani**

Teaching assistant: **Eng. Hanan Awawdeh**

Section : **4**

Date : **05 / 03 / 2024**

Abstract

In this experiment, we will review our knowledge of digital systems and combinational logic circuits in general, and understanding the construction of digital adders, subtractors and comparators especially. In this experiment we will construct comparators with basic gates and ICs and its implementation and truth table. then, we will implement half- and full adders using basic logic gates and ICs. After that, we will implement a 4-bit adder and BCD adder using the implementation of the half- and full adders. Then, we should construct half- and full subtractors. After we design these digital circuits practically, we can use their implementation to design more complex circuits such as n-bit adder, subtractor or comparator. Finally, to understand the theory of complements.

At the end of this lab, my theoretical background will be proved practically so, I will be able to construct, implement and design adders, subtractor and comparator digital circuits. In addition to construct more complex circuits using the small one such as construct 4-bit adder using the implementation of a full adder.

Table of Contents

Table of Figures

List of Tables

1. Theory

1- Half- and Full- Adders

- **Half- Adder**

Half-Adder is a digital combinational logic circuit that performs binary addition of two single-bit binary numbers. It has two inputs, A and B, and two outputs, SUM and CARRY. The SUM output represented the result of the summation between A and B (A+B), while the CARRY output indicates whether there was a carry-over from the addition of the two inputs. The half-adder can be implemented using basic gates such as XOR and AND gates. Also, the half-adder is a basic building block for more complex adder circuits such as full adders [1].

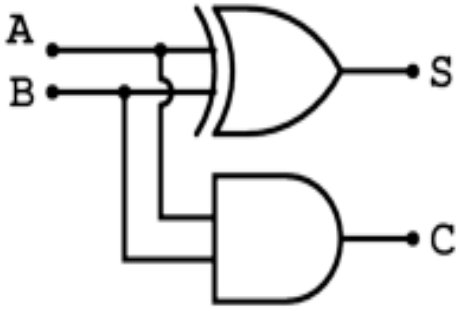

Figure 1 : Half-Adder Circuit

Inputs		Outputs	
A	**B**	**SUM**	**CARRY**
0	**0**	**0**	**0**
0	**1**	**1**	**0**
1	**0**	**1**	**0**
1	**1**	**0**	**1**

Table 1 : Truth table of Half-Adder

- **Full-Adder**

Full-Adder is a combinational logic circuit which is designed to add three binary digits and produces two outputs (sum and carry) is known as a full adder. Thus, a full adder circuit adds three binary digits, where two are the inputs and one is the carry forwarded from the previous addition (carry in) [2].

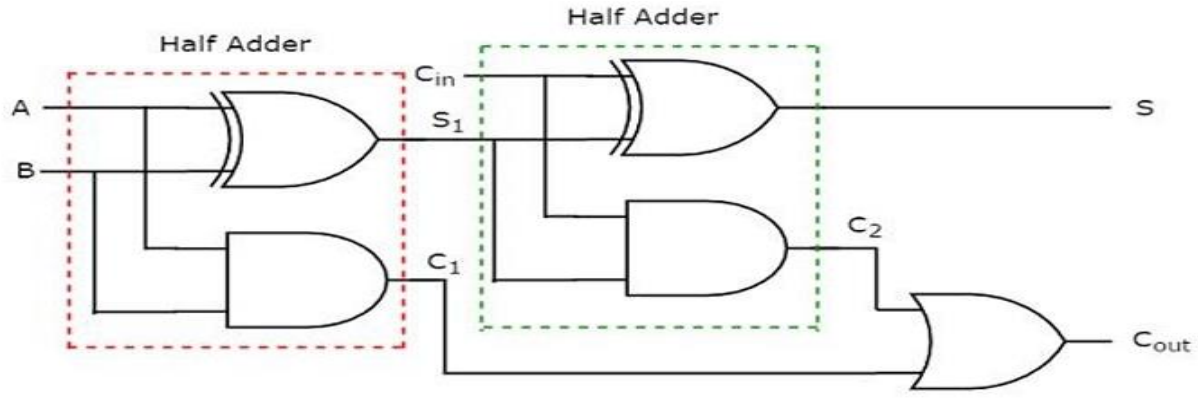

Figure 2 : Full-Adder Circuit

Inputs			Outputs	
A	B	Cin	S	Cout
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

Table 2 : Truth table of Full-Adder

2- Half- and Full- Subtractors

Half-Subtractor

Half-Subtractor is a digital logic circuit that performs binary subtraction of two single-bit binary numbers. It has two inputs, A and B, and two outputs, DF and BW. The DF output is the difference between the two input bits, while the BW output indicates whether borrowing was necessary during the subtraction. The half subtractor can be implemented using basic gates such as XOR and NOT gates. The DF output is the XOR of the two inputs A and B, while the BW output is the NOT of input A then ANDING with input B [3].

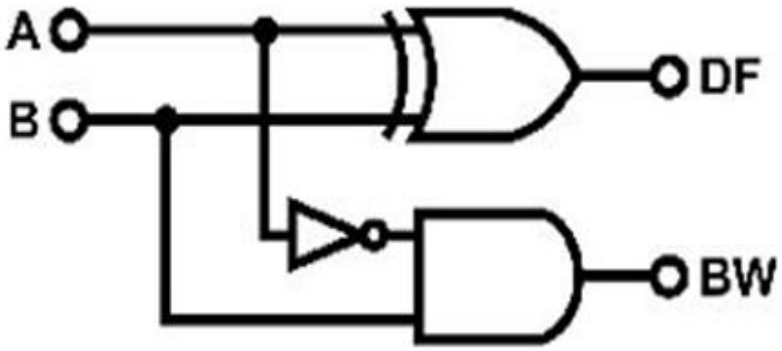

Figure 3 : Half-Subtractor Circuit

Inputs		Outputs	
A	B	DF	BW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Table 3 : Truth table of Half-Subtractor

- **Full-Subtractor**

Full-Subtractor is a combinational circuit that performs subtraction of two bits, one is minuend and other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit. This circuit has three inputs and two outputs. The three inputs A, B and Bin, denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and Bout represent the difference and output borrow, respectively. Although subtraction is usually achieved by adding the complement of subtrahend to the minuend, it is of academic interest to work out the Truth Table and logic realization of a full subtractor; x is the minuend; y is the subtrahend; z is the input borrow; D is the difference; and B denotes the output borrow [4].

Inputs			Outputs	
A	**B**	**C**	**DF**	**BW**
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**

Table 4 : Truth table of Full-Subtractor

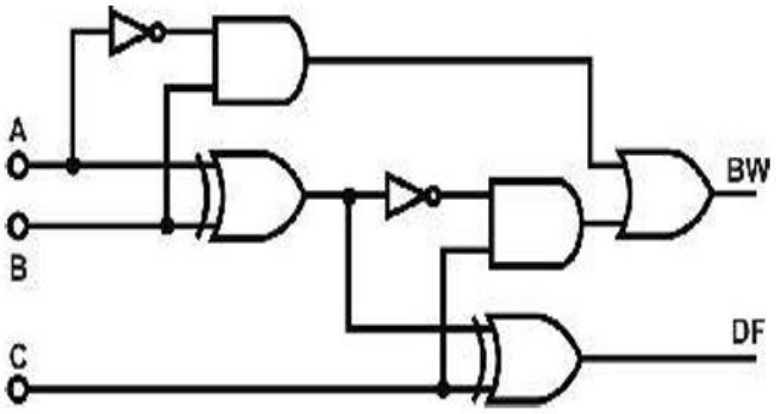

Figure 4 : Full-Subtractor Circuit

3- Comparator

A magnitude digital Comparator is a combinational circuit that compares two digital or binary numbers in order to find out whether one binary number is equal, less than, or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and the other for B and have three output terminals, one for A > B condition, one for A = B condition, and one for A < B condition [5].

Inputs		Outputs		
A	**B**	**A>B**	**A=B**	**A<B**
0	**0**	**0**	**1**	**0**
0	**1**	**0**	**0**	**1**
1	**0**	**1**	**0**	**0**
1	**1**	**0**	**1**	**0**

Table 5 : Truth table of comparator

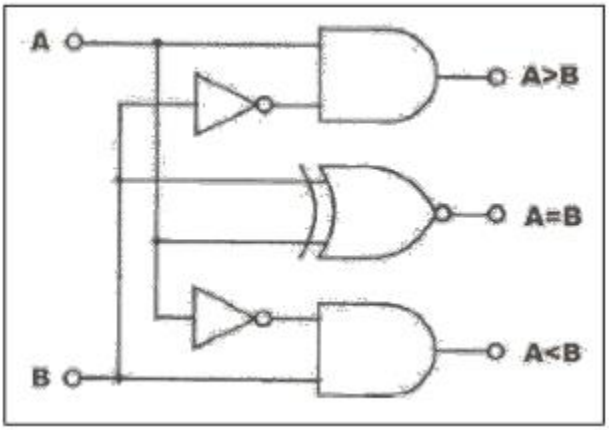

Figure 5 : Comparator Circuit

2. Procedure and Discussion

2.1 Comparator Circuits

2.1.1 Constructing Comparator with Basic Logic Gates

We connected the Circuit as it shown in the Figure 2.1 using IT-3002 block .

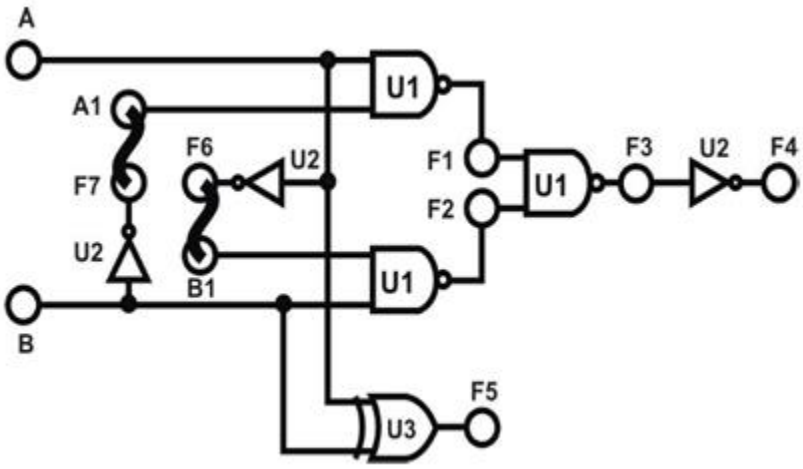

Figure 2. 1 IT-3002 Comparator 1 block (Source: Lab Manual)

a) set module IT-3002 block comparator 1. We will use U1, U2 and U3 to construct the 1-bit Comparator .
b) We connected the +5V of module IT-3002 to the +5V output of the fixed power supply IT-3000 and do the same for GND.
c) Then, We connected the inputs A and B to Data Switches SW1 and SW2 respectively. The inputs are triggered by high-state voltage. After that we connected the outputs to Logic Indicators (LED). The outputs are triggered by low-state voltage.
d) The results are shown in Table 2.1 :

INPUTS			OUTPUTS		
SW2 (B)	SW1(A)		F1	F2	F5
0	0	A=B	1	1	0
0	1	A>B	0	1	1
1	0	A<B	1	0	1
1	1	A=B	1	1	0

Table 2. 1 : Results of 1-bit Comparator

e) The Circuit connected as following Figure 2.2 :

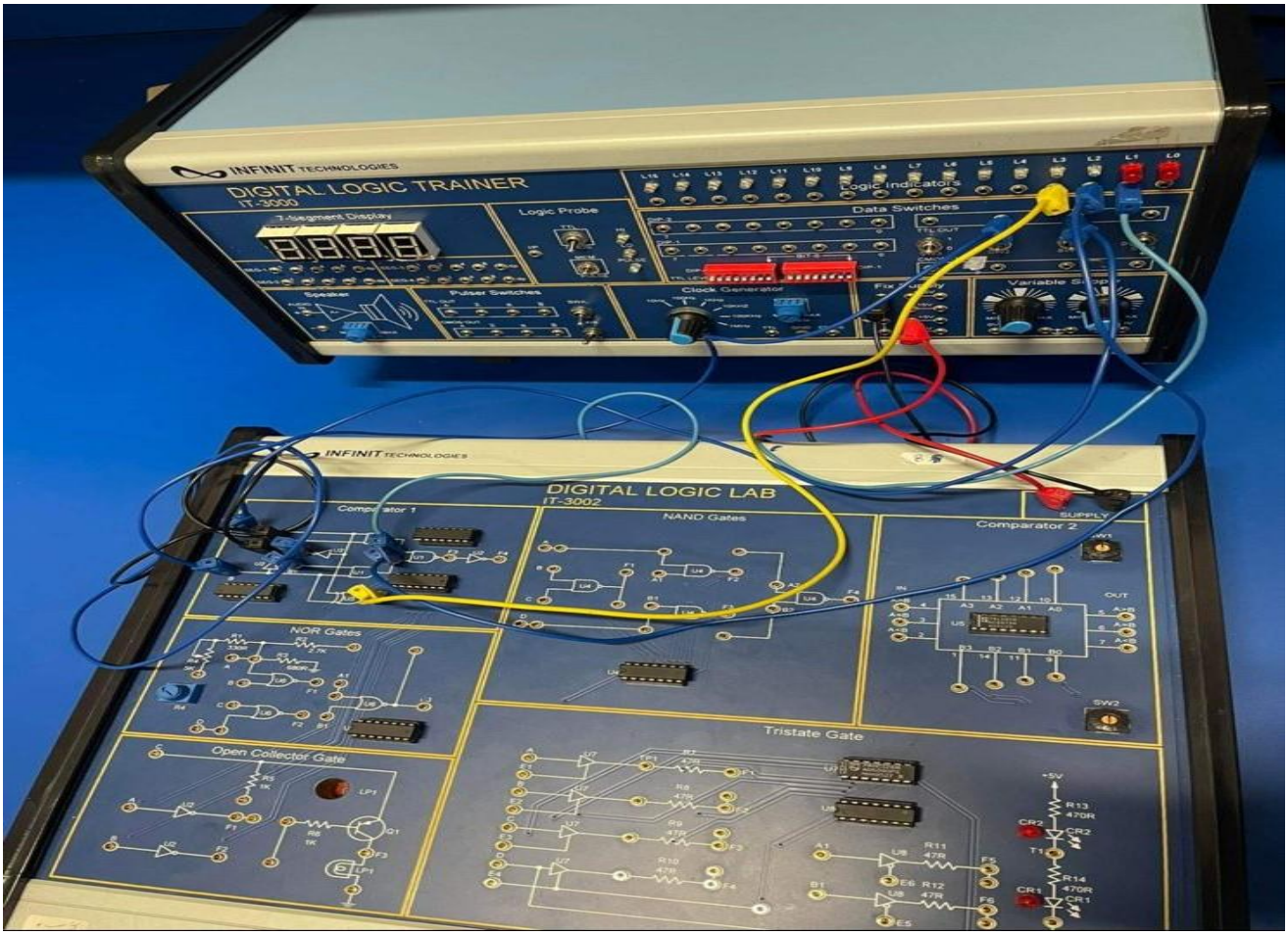

Figure 2. 2 : 1-bit Comparator Circuit Connection

Discussion :

As the result shown above, the 1-bit comparator circuit works properly. There is one LOW output in a time. This output represent the correct case of comparing inputs. Because the outputs are triggered by low-state voltage. Then, the other outputs are HIGH in the same situation, which represent the incorrect cases.

2.1.2 Constructing Comparator with TTL IC

We connected this circuit using block (Comparator 2) of module IT-3002, this block is a 4-bit Comparator IC as shown in the figure 2.3.

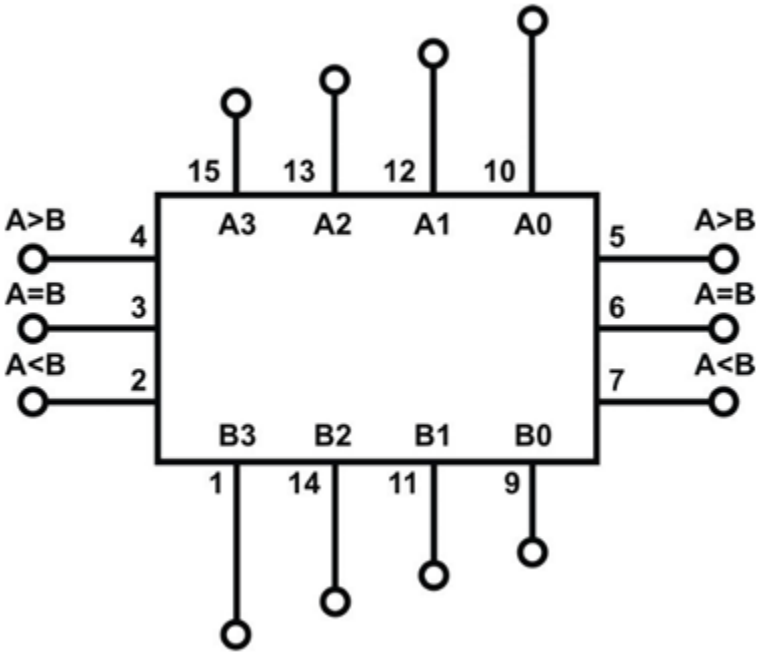

Figure 2. 3 : 4-bit Comparator IC (Source: Lab Manual)

a) We connected in left side of the block to Data Switches , then connected the inputs A0..A3 and B0..B3 to the BCD rotary Switches. Finally, connected the output in right side of the block to LEDs.

b) We set comparing inputs A0..A3 = As, B0..B3 = Bs from the rotary switch.

c) By following the cascading inputs sequence, the results are shown in Table 2.2:

INPUTS			OUTPUTS		
A>B	A=B	A<B	A>B	A=B	A<B
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	1	0	1	0

Table 2. 2 : Results of 4-bit Comparator IC

d) The circuit connected as following Figure 2.4

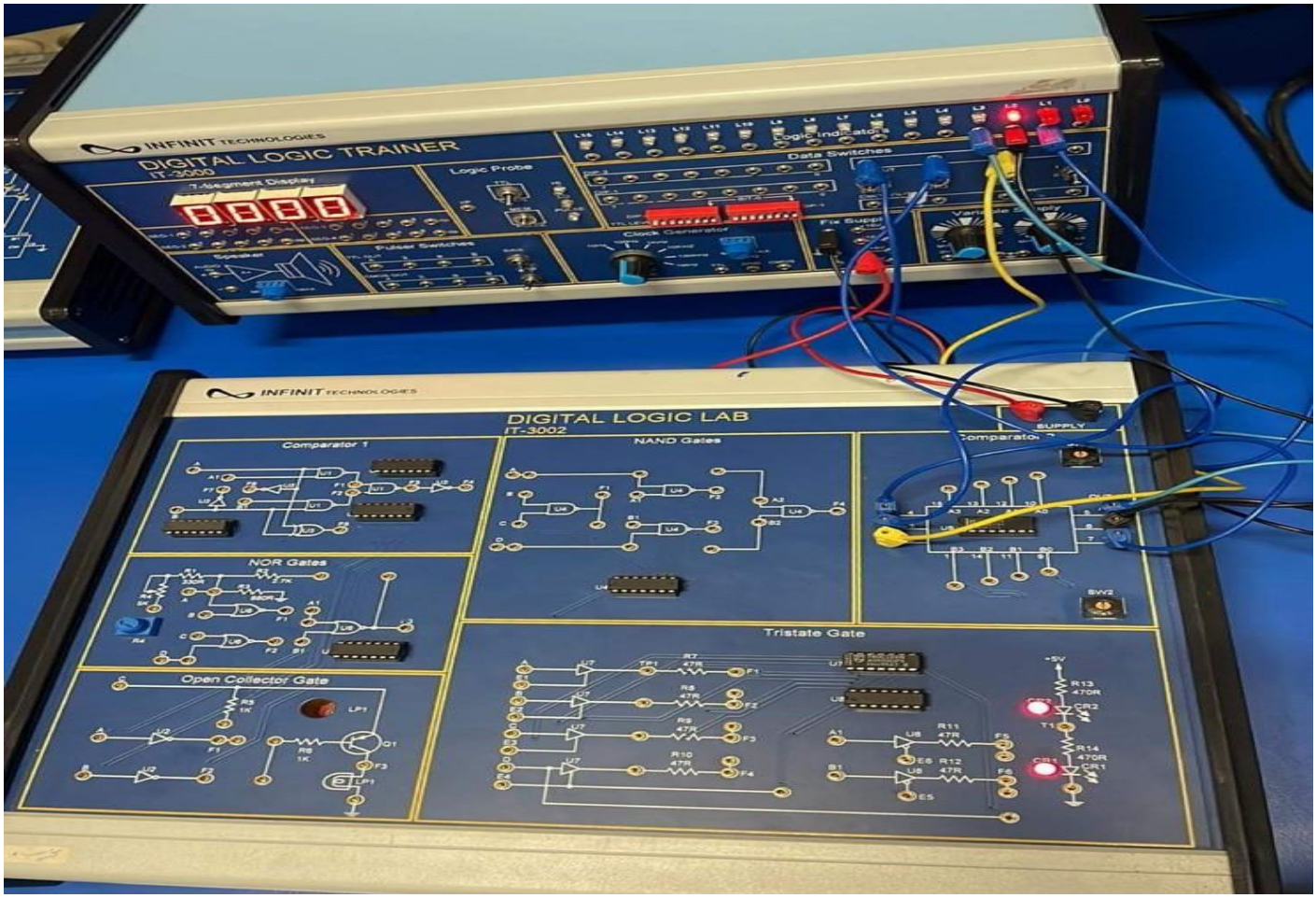

Figure 2. 4 : 4-bit Comparator Circuit Connection

Discussion :

In this section we design a cascade 4-bit comparator (depends on the previous 4-bit comparator inputs). But we control the previous 4-bit comparator outputs by rotary switches. The circuit works properly and the outputs are precise, but when the previous 4-bit comparator has something wrong with its outputs like the third case in Table 2.2 (A=B and A<B) , then there is wrong outputs from this 4-bit Comparator Circuit.

2.2 Half- and Full- Adder Circuits

2.2.1 Half – Adder Circuit

We connected the half – adder circuit using the half – adder block in the module IT-3003 as the Figure 2.5 below :

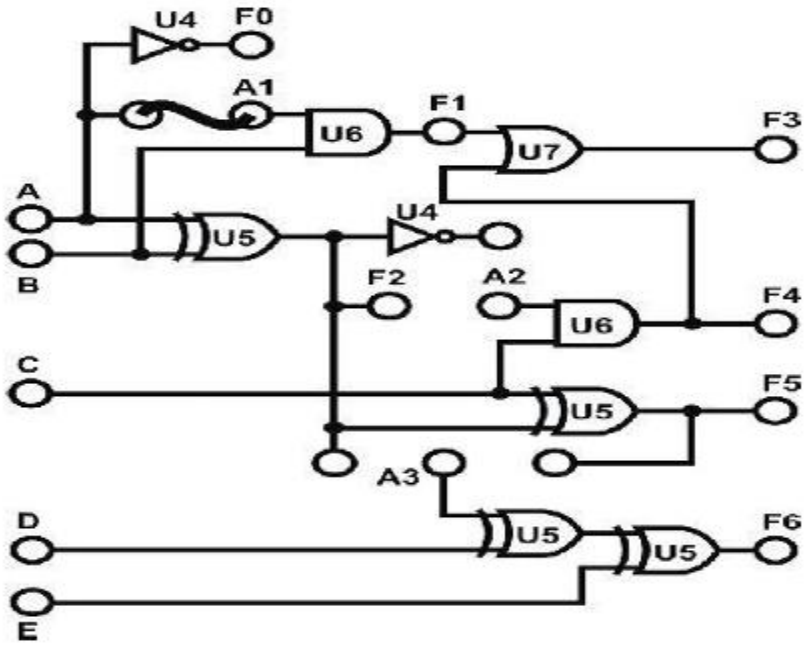

Figure 2. 5 : IT-3003 Half - Adder Block (Source : Lab Manual)

a) We connected the +5V of module IT-3003 to the +5V output of the fixed power supply, and do the same for the Ground (GND).
b) We inserted the connection as the Figure 2.5 above, using U5 and U6 of the module.
c) After that, We connected the inputs A and B to Data Switches and connected the outputs F1 (that represented CARRY) and F2 (that represented SUM) to logic indicators (LEDs)
d) Then, the results are shown in the Table 2.3 below :

INPUTS		OUTPUTS	
B	**A**	**F1 (CARRY)**	**F2 (SUM)**
0	**0**	**0**	**0**
0	**1**	**0**	**1**
1	**0**	**0**	**1**
1	**1**	**1**	**0**

Table 2. 3 : Results of Half - Adder Circuit

e) The circuit connecting as following Figure 2.6

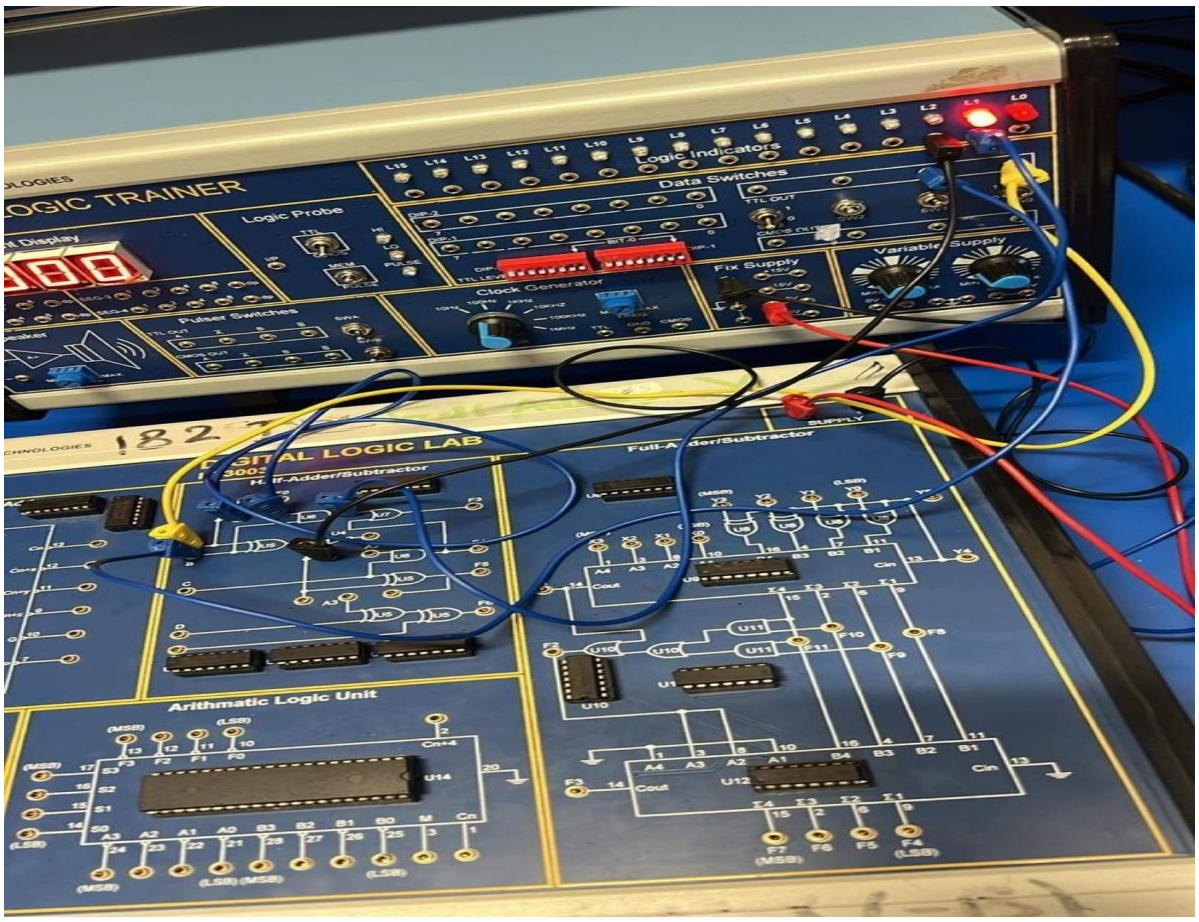

Figure 2. 6 : Half - Adder Circuit Connection

Discussion :

After we connected the Half – Adder Circuit using the half – adder block of the module IT-3003, we show that the results of outputs are as expected and the circuit works properly. So, the output F1 (that represented the CARRY) is High when the two inputs A and B is High, otherwise it is LOW. And the output F2 (that represented SUM) is High when only one of the inputs is High, otherwise it is LOW.

2.2.2 Full – Adder Circuit

To construct the full – adder circuit, we use the full – adder block of the module IT-3003 as the Figure 2.7 that shown below :

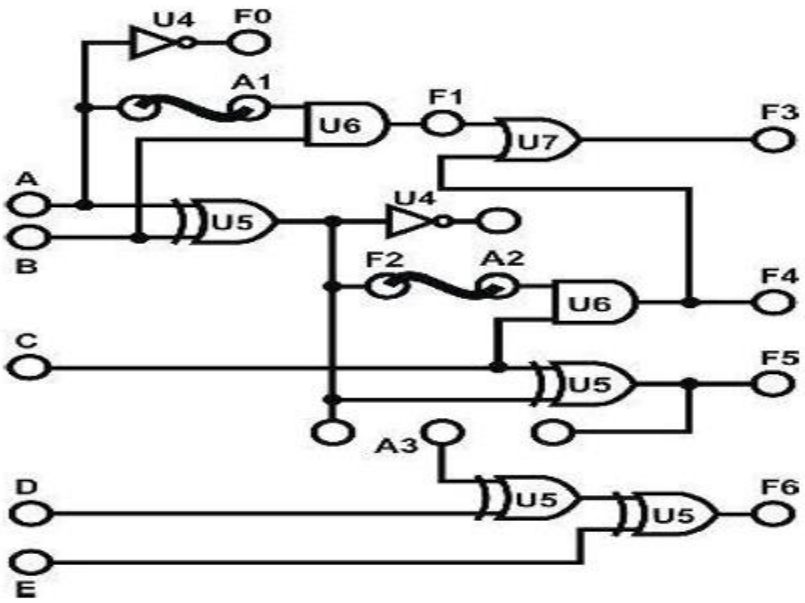

Figure 2. 7 : IT-3003 Full - Adder block (Source : Lab Manual)

a) Firstly, we connected the inputs A, B and C to three Data Switches. A and B are the augends (inputs), while C in the carry in .

b) Secondly, connected the outputs F3 (that represented CARRY) and F5 (that represented SUM) to logic indicators (LEDs).

c) Finaly, the results are shown in the Table 2.4 below :

INPUTS			OUTPUTS	
C	B	A	F3 (CARRY)	F5 (SUM)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table 2. 4 : Results of Full - Adder Circuit

d) The Circuit connecting as following Figure 2.8

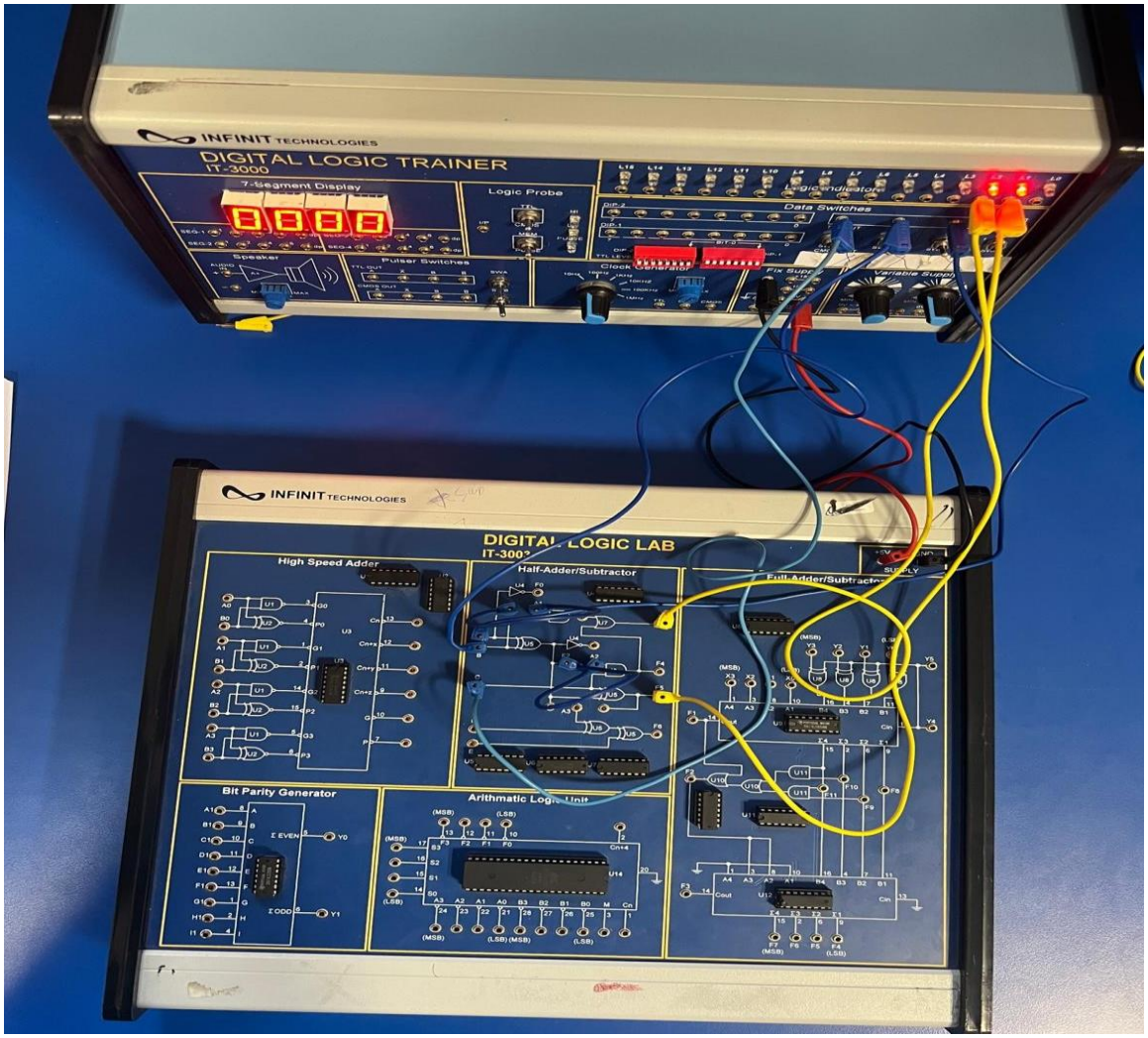

Figure 2. 8 : Full - Adder Circuit Connection

Discussion :

In this section we construct a full – adder circuit, that has two inputs A and B (augends) and the third input C that represent the previous carry (carry in). And this circuit has two outputs one of them represent the summation of the three inputs, while the other represent if there is a carry out. After connecting this circuit, and check the results for all cases, we show that the results of this circuit are similar to the theoretical results, so the circuit works good and as expected.

2.3 Half – and Full Subtractor Circuits

By setting module IT-3003 and locate block Half – Adder, the half – and full – subtractor are connected as the Figure 2.9 below :

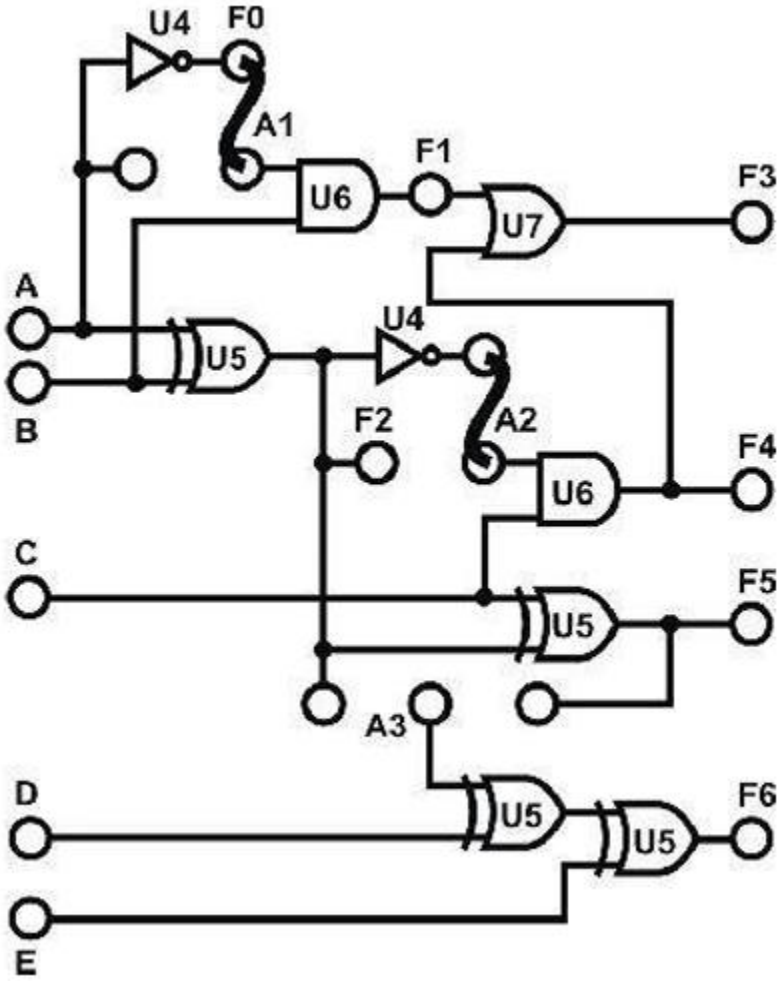

Figure 2. 9 : Half - / Full – Subtractor

a) We connected A, B and C to Data Switches and the outputs F1, F2, F3 and F5 to logic indicators (LEDS) .

b) When (C = 0), the circuit is a half – subtractor . F1 is the Borrow and F2 is the Difference. And the other outputs shown as F5 = F2 ; F3 = F1 .

c) When (C = 1), the circuit is a full – subtractor. F3 is the Borrow and F5 is the Difference.

d) Then, the results are shown in the Table 2.5 bellow :

	INPUTS			OUTPUTS			
	C	A	B	F1	F2	F3	F5
Half – Subtractor	0	0	1	1	1	1	1
	0	0	0	0	0	0	0
	0	1	1	0	0	0	0
	0	1	0	0	1	0	1
Full – Subtractor	1	0	0	0	0	1	1
	1	0	1	1	1	1	0
	1	1	0	0	1	0	0
	1	1	1	0	0	1	1

Table 2. 5 : Results of Half - / Full - Sibtractor Circuit

e) The Circuit connecting as the Figure 2.10 bellow :

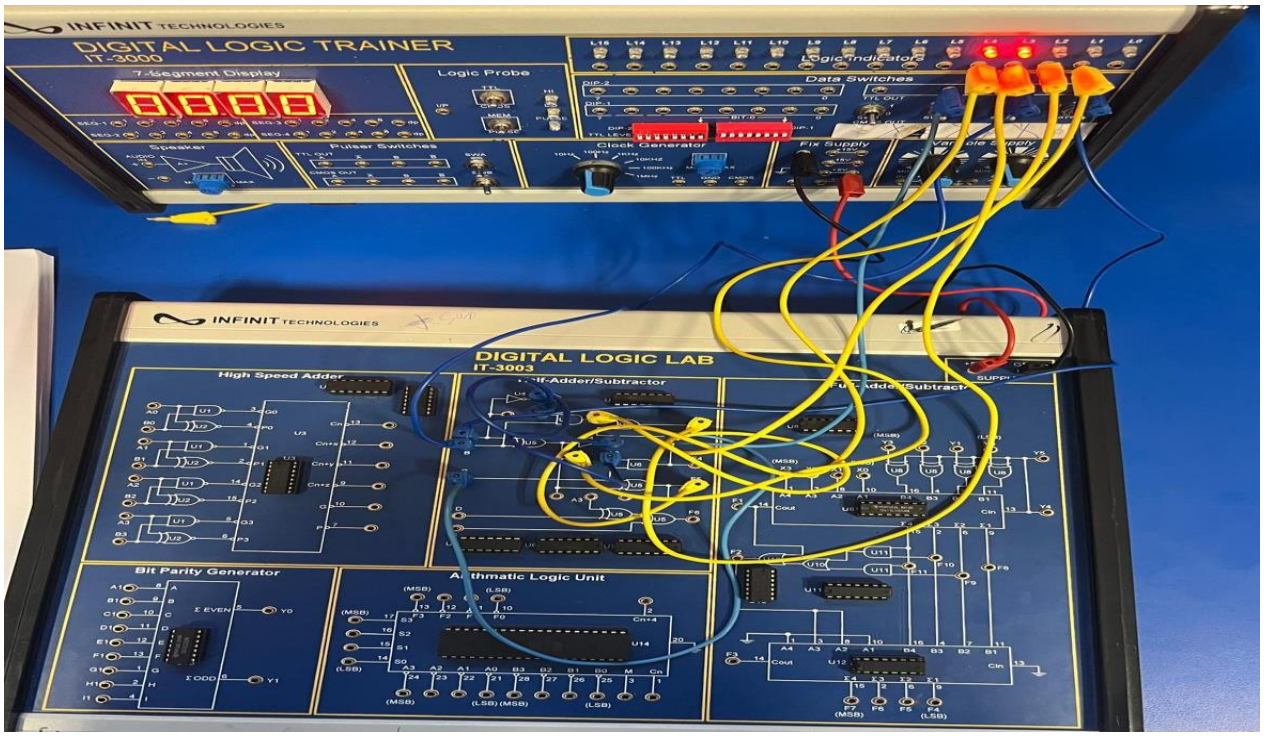

Figure 2. 10 : Half - / Full – Adder Subtractor Circuit Connection

Discussion :

As we show in the Circuit above, we use the basic logic gates to implement the half - / full – subtractor circuit. The input C is like a controller if its value is LOW, then the circuit is a half – subtractor. If its value is HIGH, then the circuit is a full – subtractor. And as we show in the results Table 2.5, the circuit results are as we expected. So, the circuit is work fine.

2.4 4-bit Full – Adder with IC

Using the 4 – bit IC , that shown in Figure 2.11 bellow, we connected a circuit that sum two 4 – bit binary number, this IC have 4 Full – Adders .

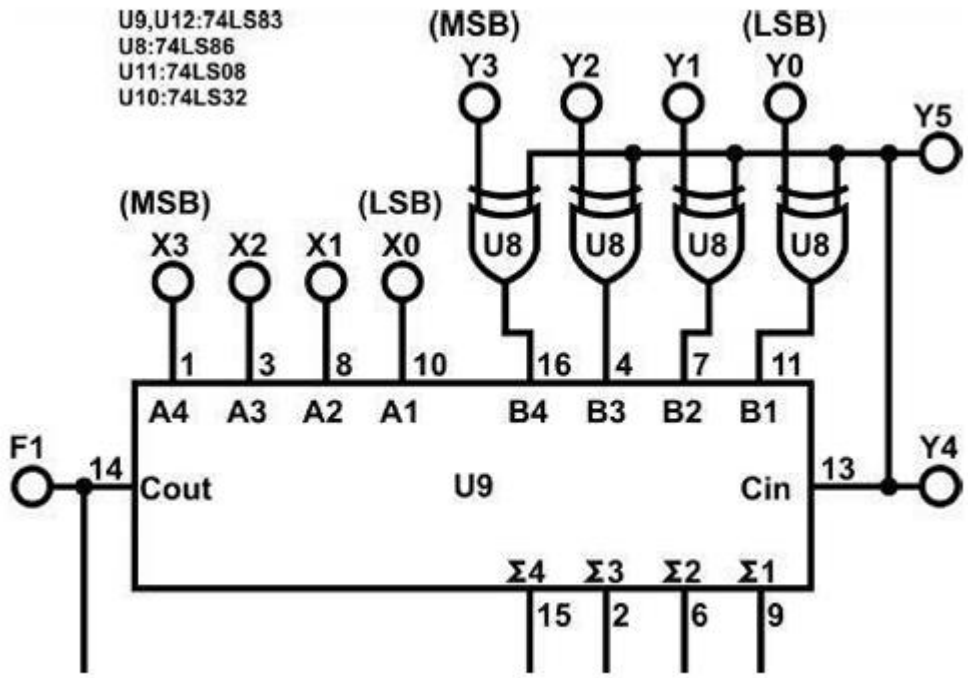

Figure 2. 11 : 4 - bit Full - Adder block (IC)

a) Firstly, connecting X3…X0 to rotary switch DIP1 and connected Y3…Y0 to another rotary switch DIP2 .

b) Secondly, connect the Y5 to Data Switch and give it a LOW value (“0”), so the XOR gates will act as buffer gates , and the Cin has a value of (“0”).

c) Then, connect the outputs ∑4…∑1 to logic indicators (LEDs).

d) Finally, the results are shown in the following Table 2.6 :

INPUTS								OUTPUTS				
Y3	Y2	Y1	Y0	X3	X2	X1	X0	∑4	∑3	∑2	∑1	F1 (CARRY)
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	1	1	0	0	1	1	0	0
0	0	0	0	1	0	0	1	1	0	0	1	0
0	0	0	0	1	1	1	1	1	1	1	1	0

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

Table 2. 6 : Results of 4 - bit Adder

e) The circuit connecting as the following Figure 2.12

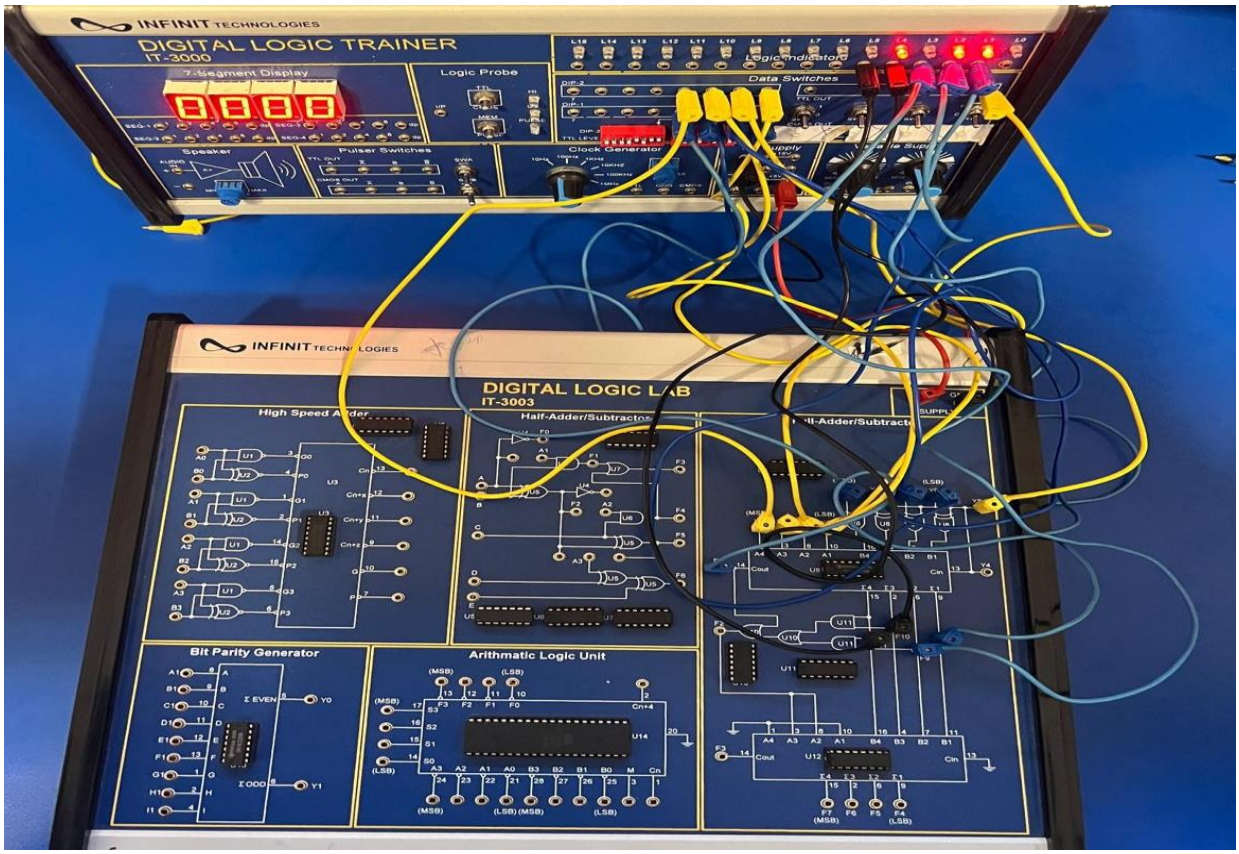

Figure 2. 12 : 4 - bit Adder Circuit Connection

Discussion :

In this section we construct a 4 – bit Adder, by using a IC that has four full – adders. We give Y5 In the Figure 2.11 a LOW value, so the XOR gates will act as a buffer, and the outputs are connected to LEDs to read the results, after read the results we comparing them to the theoretical results and they are similar. So our connection to this circuit is fine.

2.5 4 – bit Full – Subtractor with IC

Using the 4 – bit IC , that shown in Figure 2.13 bellow, we connected a circuit that sum two 4 – bit binary number, this IC have 4 Full – Adders .

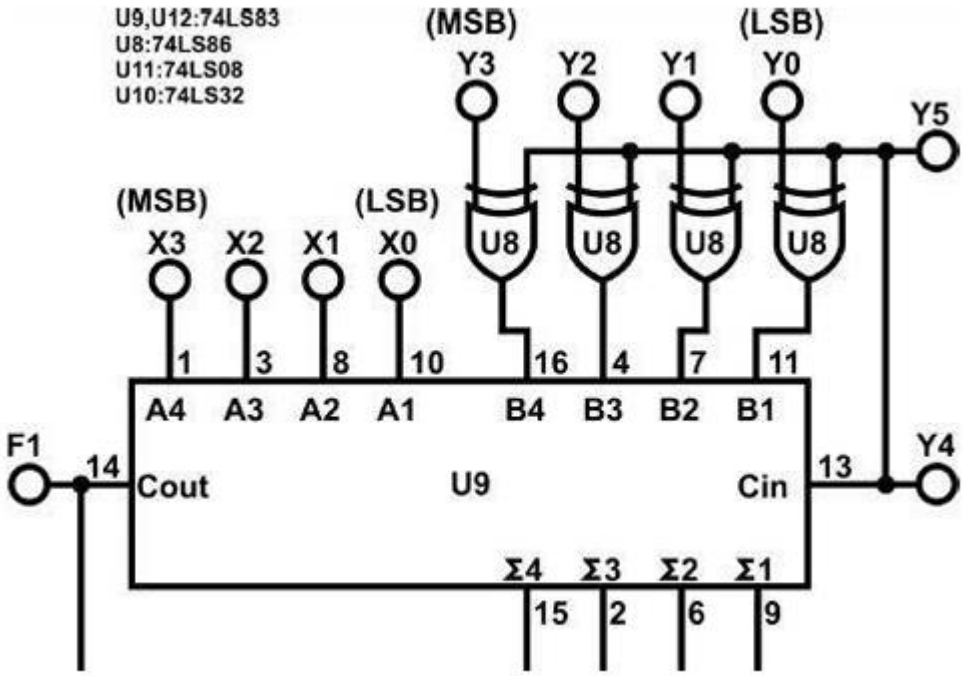

Figure 2. 13 : 4 - bit Full - Adder Block (IC)

a) Firstly, connecting X3…X0 to rotary switch DIP1 and connected Y3…Y0 to another rotary switch DIP2 .

b) Secondly, connect the Y5 to Data Switch and give it a HIGH value (“1”), so the XOR gates will act as NOT gates , and the Cin has a value of (“1”).

c) Then, connect the outputs ∑4…∑1 to logic indicators (LEDs).

d) Finally, the results are shown in the following Table 2.7 :

INPUTS								OUTPUTS				
X3	X2	X1	X0	Y3	Y2	Y1	Y0	F1	F11	F10	F9	F8
0	1	0	0	0	1	0	0	1	0	0	0	0
0	1	0	0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	1	0	1	0	1
1	0	0	0	0	0	0	1	1	0	1	1	1
1	0	0	1	1	0	0	0	1	0	0	0	1

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

Table 2. 7 : Results of 4 - bit Subtractor Circuit

e) The circuit connecting as the following Figure 2.14 :

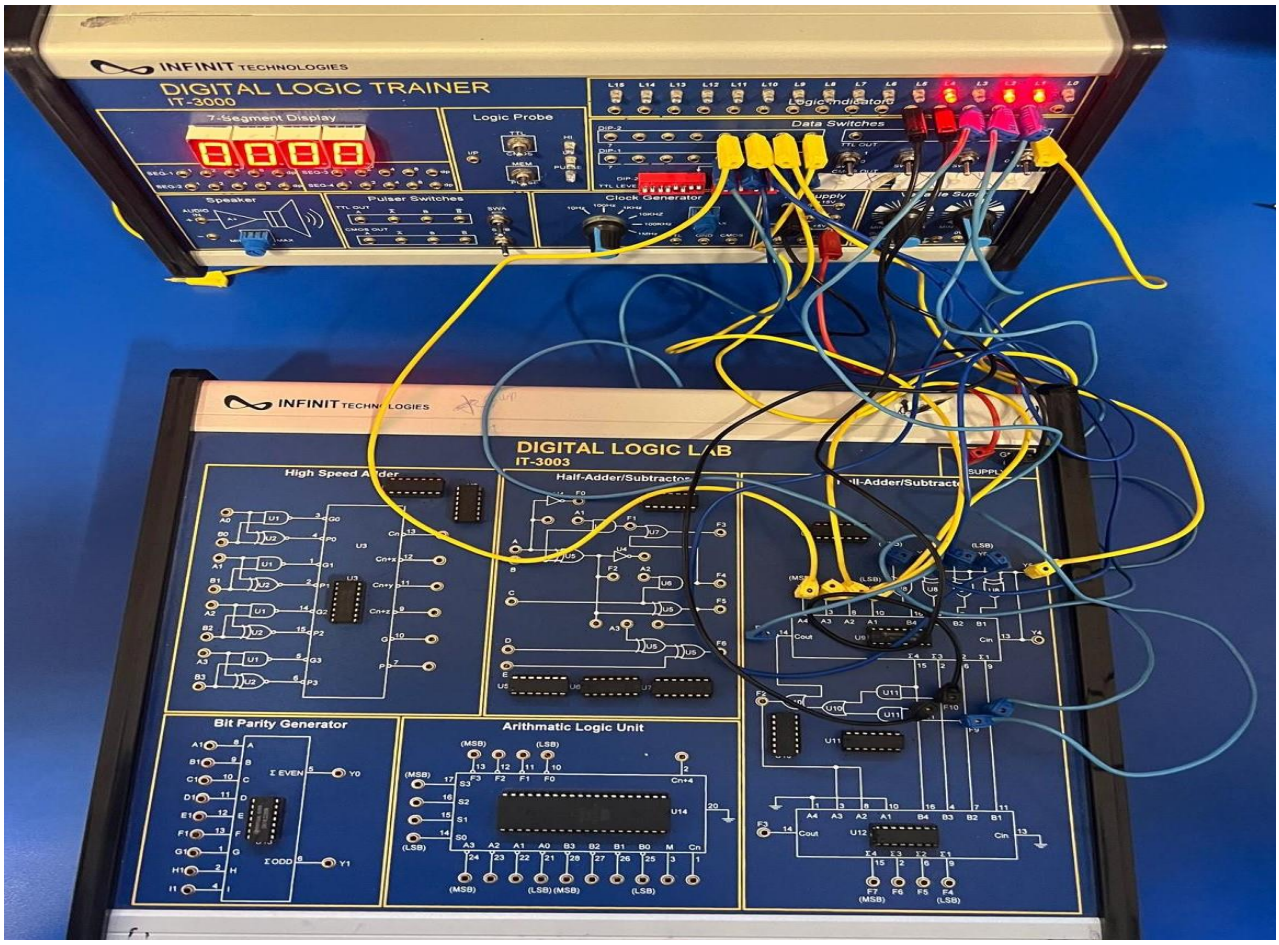

Figure 2. 14 : 4 - bit Subtractor Circuit Connection

Discussion :

To construct a 4 – bit Subtractor we use the same IC that using to construct a 4 – bit adder, but we must give the input that connected to four XOR gates as in Figure 2.13 (Y5) a HIGH voltage (“1”), then the XOR gates will act as NOT gates, so the circuit become as an adder between X3…X0 and 1’s complement of Y3…Y0 , and the Cin has the same value of Y5 (“1”) then the circuit become as X3…X0 + 2’s complement of Y3…Y0 , then the result is the difference between them.

2.6 BCD Adder

To construct BCD Adder, we use a 4-bit Adder IC, then check if the result greater than 9 to use another 4-bit Adder to add 6 to the number then its be a valid BCD.

The followed Figure 2.15 shows a circuit that acts as a BCD Adder :

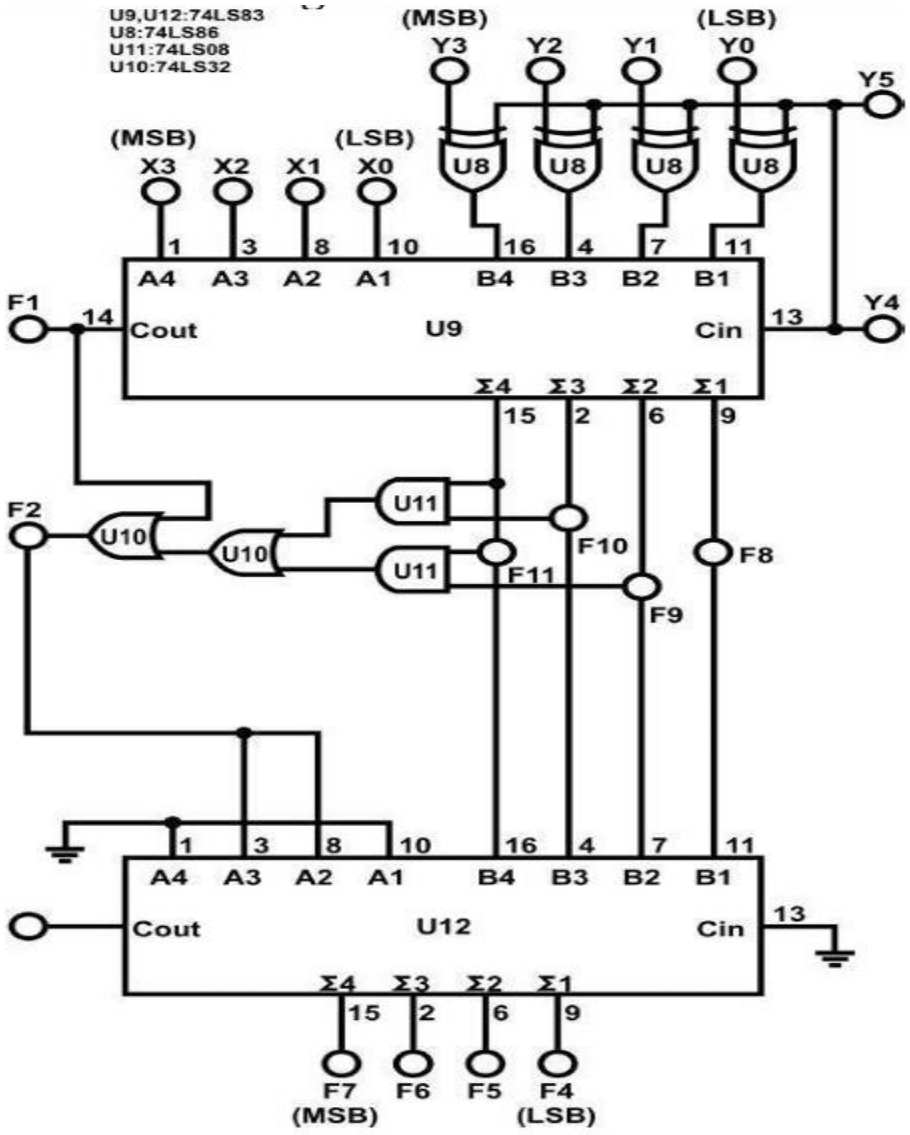

Figure 2. 15 : BCD Adder Circuit (Source : Lab Manual)

a) First, we connected the inputs X3…X0 to rotary switch DIP1, and Y3…Y0 to another rotary switch DIP2.
b) Second, give the LOW voltage value (“0”) to Y5, so U9 acts as a 4-bit adder circuit.
c) Third, connect the outputs F1 and F11…F8 to logic indicators (LEDs). F1 represent the carry out of the first 4-bit adder and F11…F8 represent the summation between X3…X0 and Y3…Y0

d) Then, U10 and U11 gates check if the summation is greater then 9 So we need to add 6 by the second 4-bit adder, else if the summation is less than or equal 9 the second 4-bit adder will add 0 so the output of the second 4-bit adder is the same of the outputs of the first 4-bit adder.

e) The Last Outputs F2 represent if that we need to correction the summation to the BCD representation by adding 6 using the second 4-bit adder, F3 represent the carry out of the circuit and F7…F4 represent the final result of the circuit (BCD summation).

f) Finally, the results of this circuit are shown in Table 2.8 bellow :

INPUTS								OUTPUTS (U9)					LAST (U12)					
X3	X2	X1	X0	Y3	Y2	Y1	Y0	F1	F11	F10	F9	F8	F2	F3	F7	F6	F5	F4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0
0	0	1	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	1
0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0	1	1	0	0	0	0
0	0	1	1	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1
0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1
0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	0	0	0	0
0	1	0	1	0	1	1	0	0	1	0	1	1	1	1	0	0	0	1
0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1
0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	0	1
0	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	0
1	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0	1	0	1	1	0	1	0	1	1	0	0
1	0	1	1	1	1	1	0	1	1	0	0	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	0

Table 2. 8 : Results of BCD Adder

g) The Circuit connecting as the following Figure 2.16 :

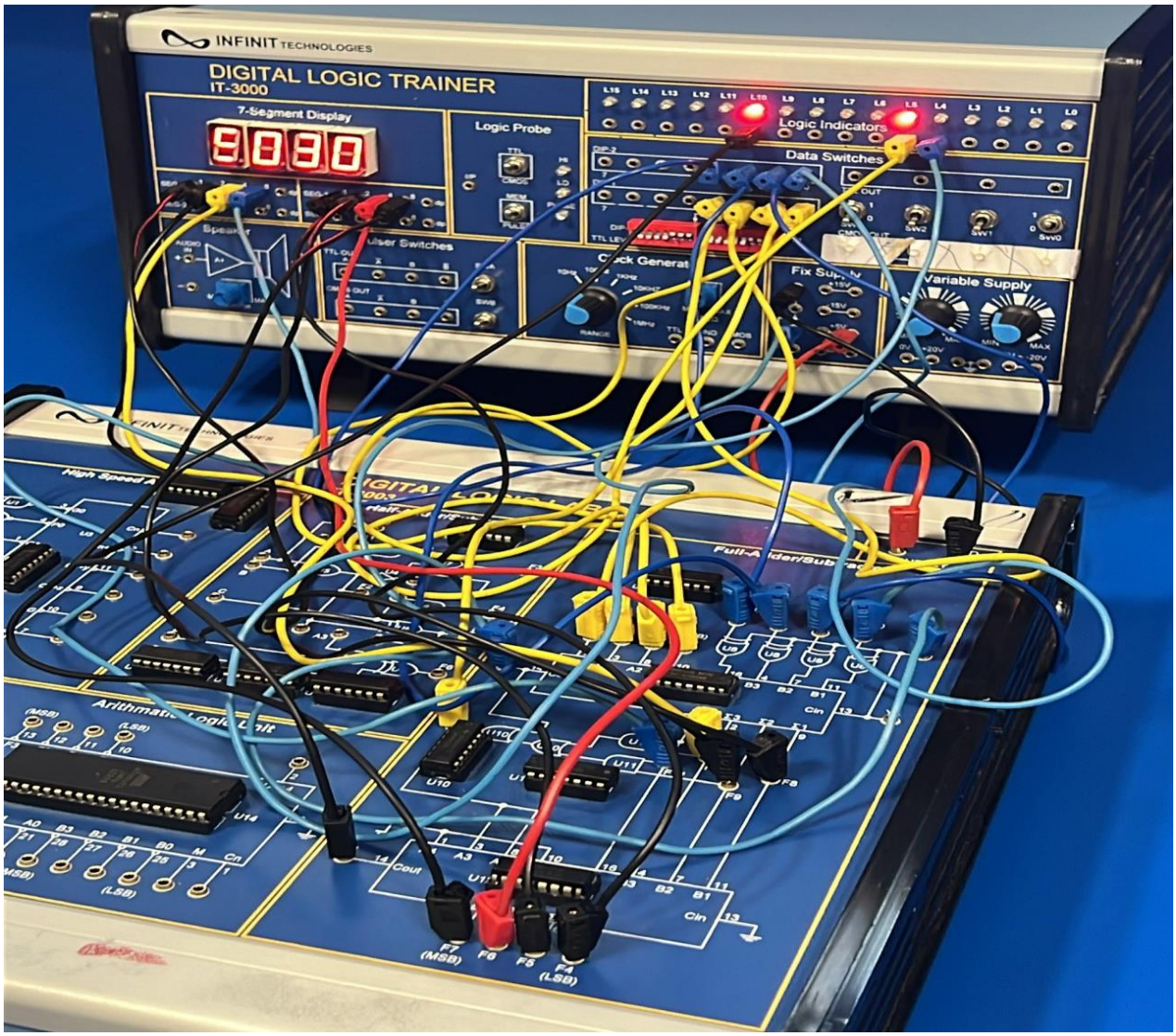

Figure 2. 16 : BCD Adder Circuit Connection

Discussion :

For BCD Adder, we use two 4-bit adder. Firstly, the 4-bit adder sum the inputs A and B, then the result needs to be checked if its larger than 9, so it be an invalid BCD value. If the output of the first 4-bit adder larger than 9 then we need to correct it to a valid BCD value by using another 4-bit adder that add 6 to the result if it is larger than 9, while if the result less than or equal 9, the second adder add 0 to the result.

Finally, the final result is the output of the second 4-bit adder, and this is the valid BCD value that represent the summation of two BCD values.

2.7 High-Speed Adder Carry Generator Circuit

To construct a carry generator circuit, we followed the Figure 2.17 bellow :

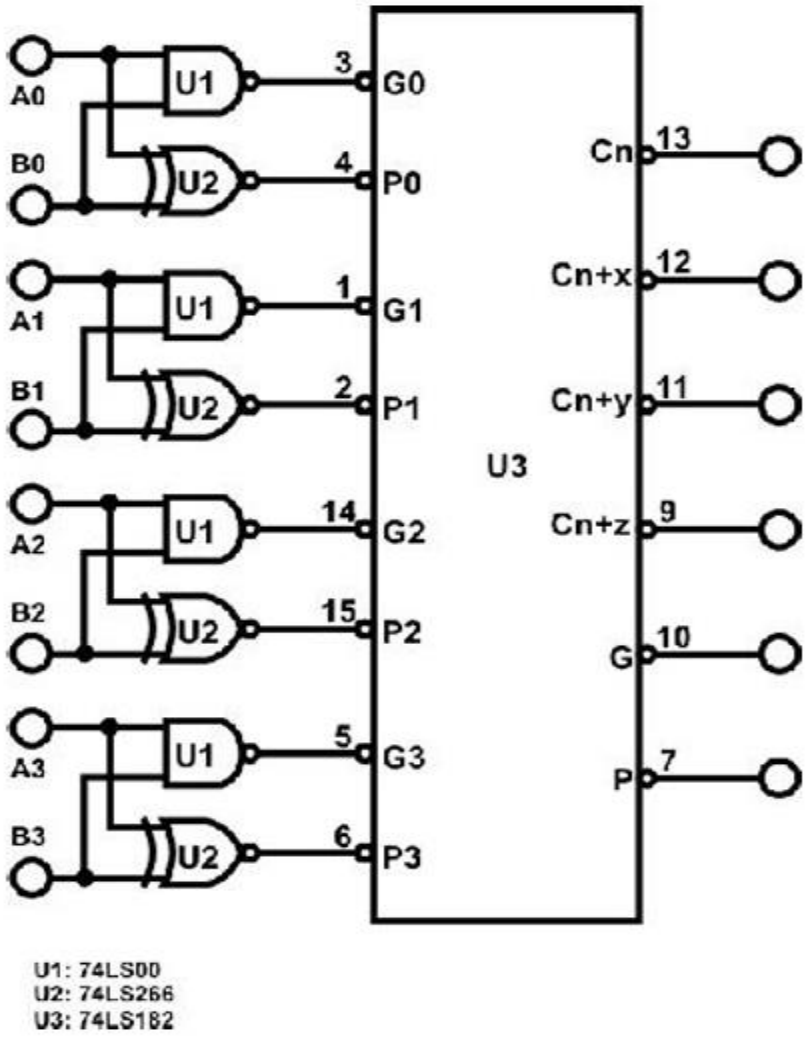

Figure 2. 17 : Carry Generator Circuit

a) At first, we connected inputs A3…A0 to rotary switch DIP1, and B3…B0 to another rotary switch DIP2. Then, we connected Cn to a Data switch, and give it a LOW voltage value (“0”).

b) Finally, the results of this circuit are shown in the Table 2.9 bellow :

INPUTS								OUTPUTS				
B3	**B2**	**B1**	**B0**	**A3**	**A2**	**A1**	**A0**	**Cn+x**	**Cn+y**	**Cn+z**	$\overline{G}$	$\overline{P}$
0	0	0	1	0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	1	1	1	1	0	0	0
1	0	1	0	1	0	0	0	0	0	0	1	0

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

Table 2. 9 : Results of Carry Generator Circuit

c) The Circuit connecting as the following Figure 2.18 :

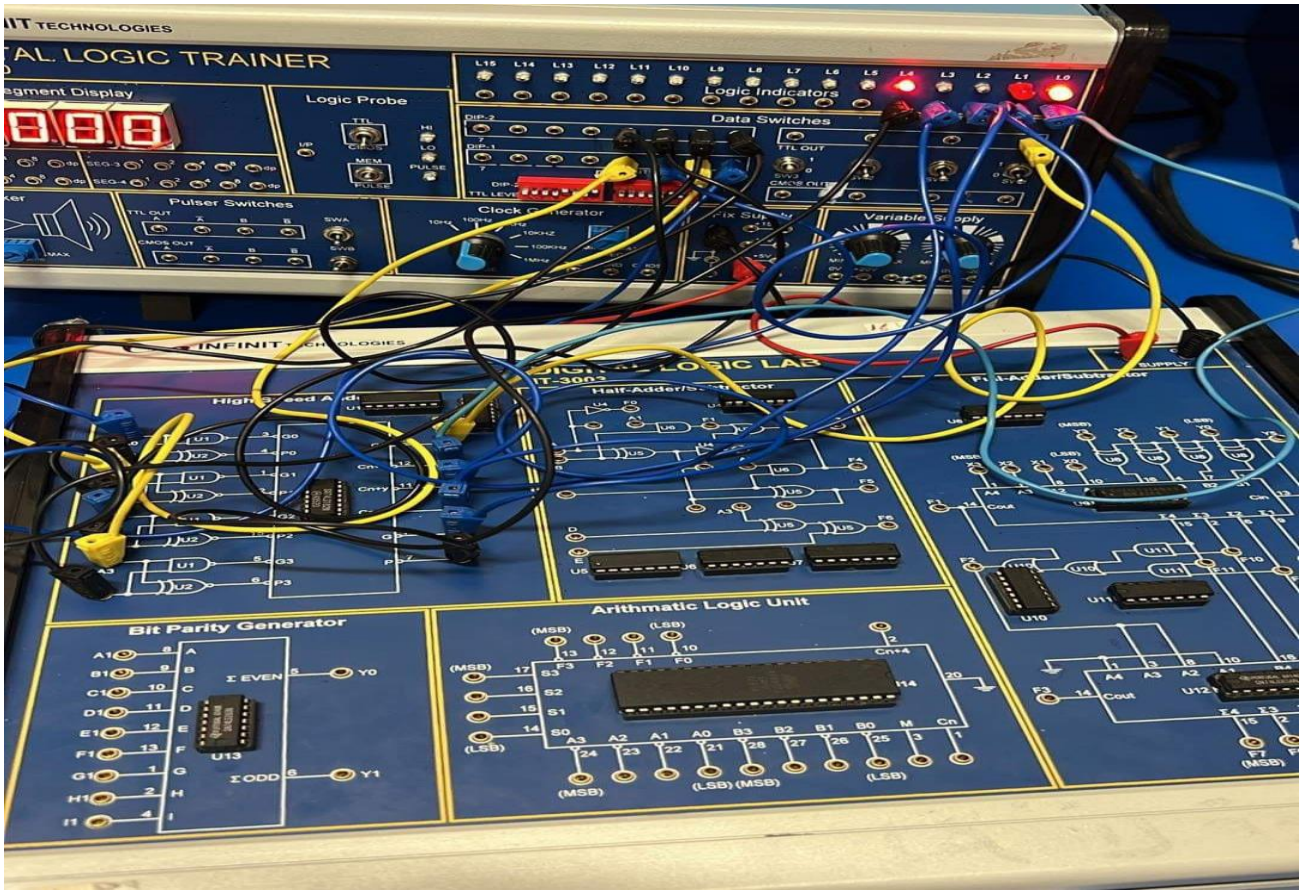

Figure 2. 18 : Carry Generator Circuit Connection

Discussion :

In this section, we construct a High – Speed Adder Carry Generator Circuit. This circuit decreases the delay of the carry signal by generating multiple carry signals in parallel instead of serial form. So, this circuit has a delay that less than the basic adder. This circuit constructed by basic gates and give the output faster than 4-bit adder. So, this circuit is mostly used in arithmetic and logic units (ALU's) of computers or other digital devices.

Conclusion

After completing this experiment all the objectives are obtained. Now, I can construct combinational digital circuits and implement them by using basic gates or ICs. Such as comparator circuit, first we implement 1-bit comparator, then use its implementation to implement more complex circuits like 4-bit comparator. For adder, we construct half-adder circuit, then using it to implement a full-adder that we can use to construct a more complex adder circuits such as 4-bit adder and then use the 4-bit adders to construct BCD adder, So we implement BCD adder using two 4-bit adders, the fist one to sum the inputs, then we check if the result is more than 9, if it is we need to correct the result by adding 6 for it using another 4-bit adder and finally the result become a valid BCD value. To construct a half- or full- subtractor we use basic gates or we can benefit from the implementation of full-adder circuits. Finally, we construct a High-Speed Adder Carry Generator Circuit that decrease the delay of carry signal and gives the outputs faster than the basic adder circuits.

After we construct the circuits and trace the results, we noticed that the results are similar to theoretical results, so we construct them correctly and there is no problem or any issue in constructing the circuits.

Post Lab

1. Design 8-bit BCD adder.

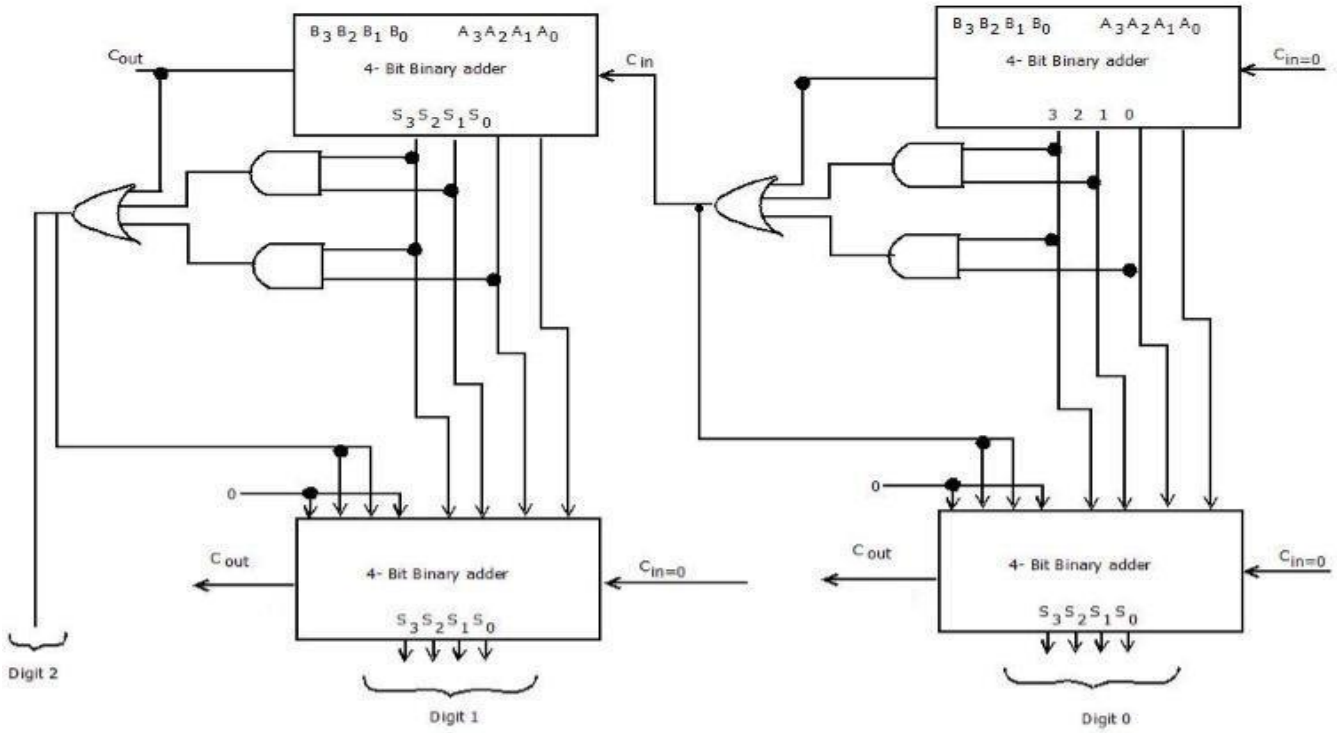

Figure 6 : 8-bit BCD Adder

2. Design 8-bit comparator using 2 of 4-bit comparator.

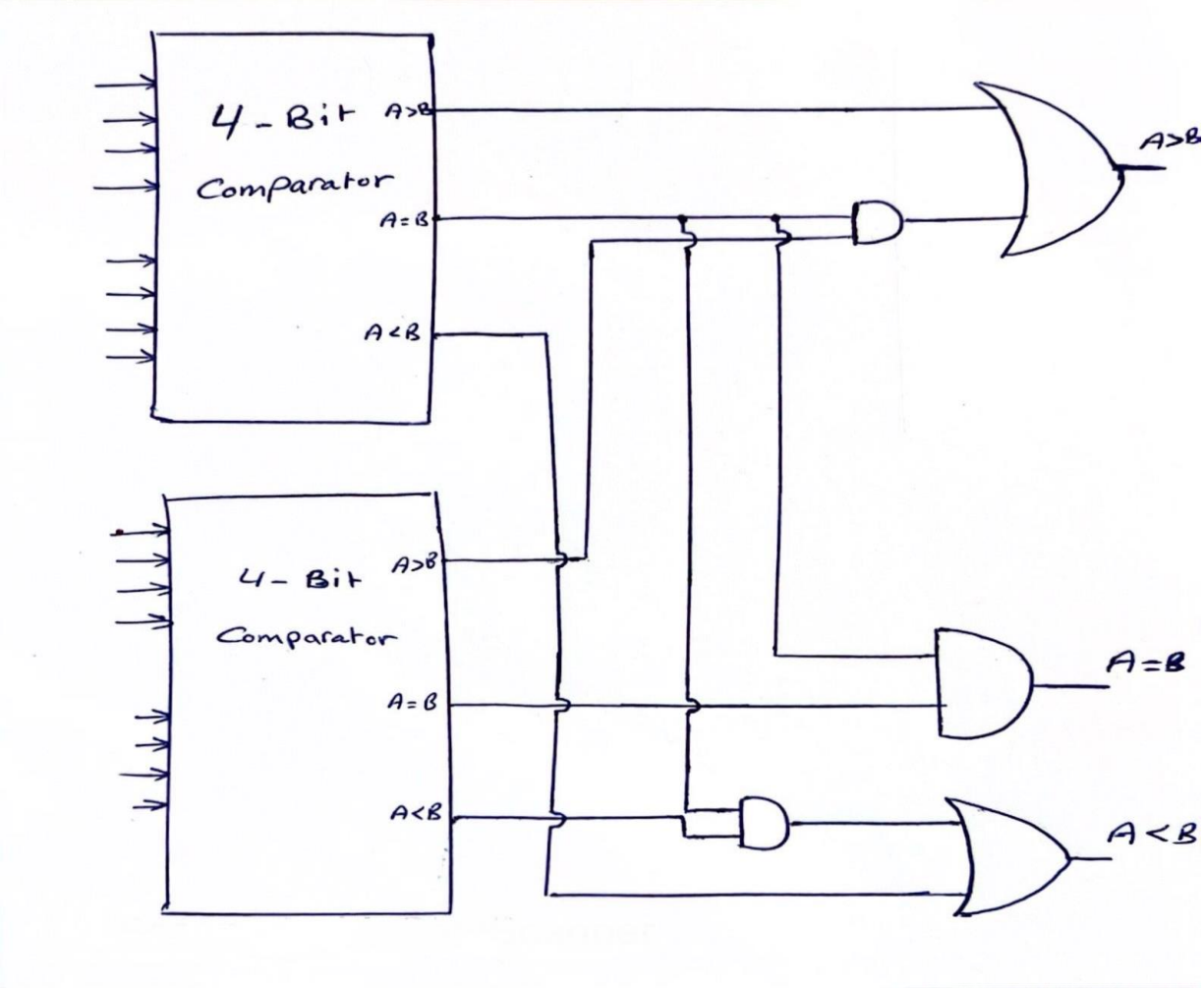

Figure 7 : 8-bit comparator

3. A 4 – inputs , 3 – outputs circuit that compares 2 – bit unsigned numbers and outputs a '1' on one of three output lines according to whether the first number is greater than, equal to or less than the other number. You can only use two 4 × 1 multiplexer.

INPUTS				OUTPUTS			FUNCTIONS	
A1	**A0**	**B1**	**B0**	**GT**	**EQ**	**LT**	**GT**	**LS**
0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**B0 + B1**
0	**0**	**0**	**1**	**0**	**0**	**1**		
0	**0**	**1**	**0**	**0**	**0**	**1**		
0	**0**	**1**	**1**	**0**	**0**	**1**		
0	**1**	**0**	**0**	**1**	**0**	**0**	**(B0+B1)'**	**B1**

0	1	0	1	0	1	0		
0	1	1	0	0	0	1		
0	1	1	1	0	0	1		
1	0	0	0	1	0	0	B1’	B0.B1
1	0	0	1	1	0	0		
1	0	1	0	0	1	0		
1	0	1	1	0	0	1		
1	1	0	0	1	0	0	(B0.B1)’	0
1	1	0	1	1	0	0		
1	1	1	0	1	0	0		
1	1	1	1	0	1	0		

Table 6 : Truth Table og 2-bit comparator

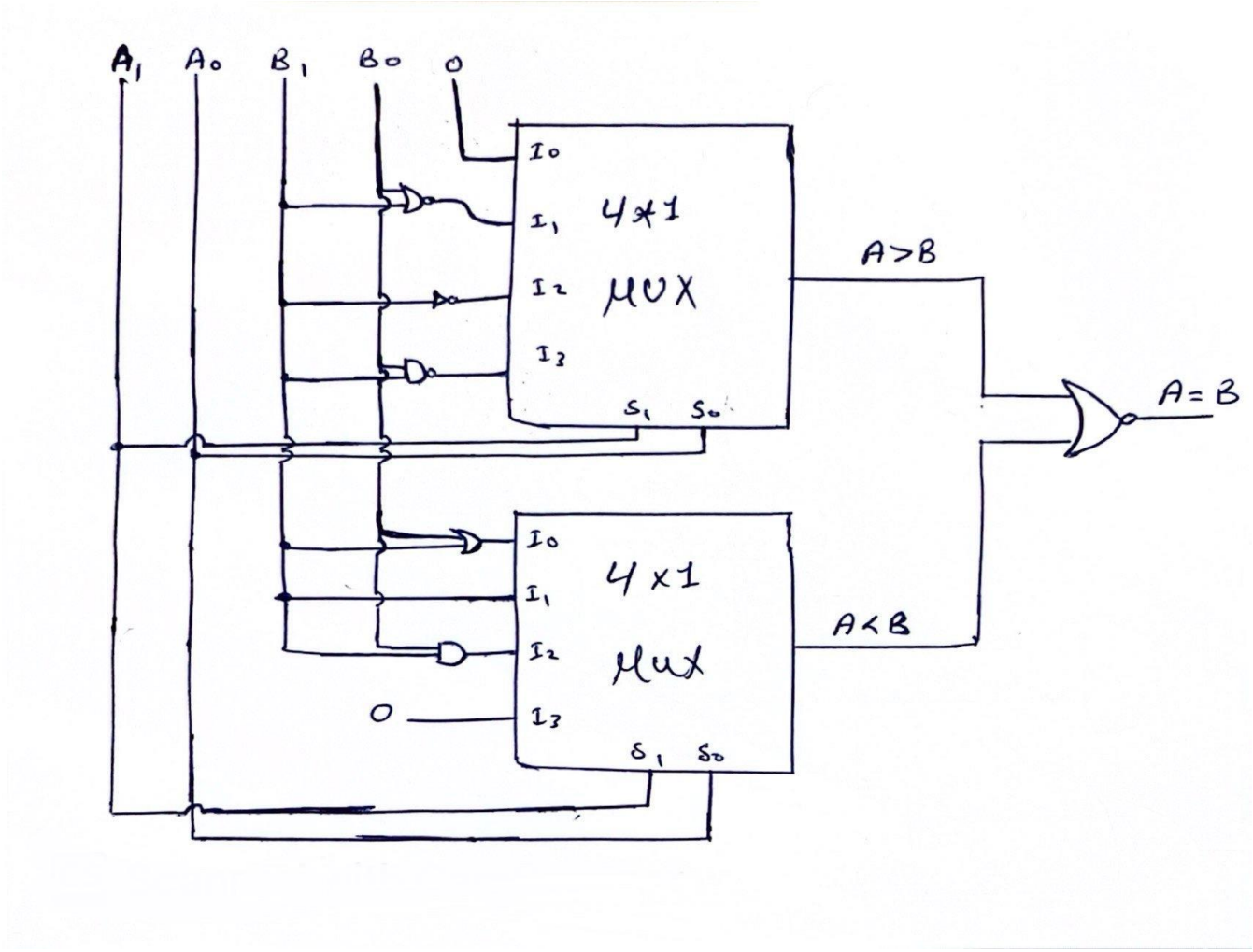

Figure 8 : 2-bit comparator circuit

References

- Manual for Digital Electronics and Computer Organization Lab, 2024, Birzeit University.

[1] : GeeksforGeeks : Half-Adder in digital logic, Retrieved March 9, 2024 from https://www.geeksforgeeks.org/half-adder-in-digital-logic/

[2] : tutorialspoint : Full-Adder in digital logic, Retrieved March 9, 2024 from https://www.tutorialspoint.com/full-adder-in-digital-electronics

[3] : GeeksforGeeks : Half-Subtractor in digital logic, Retrieved March 9, 2024 from https://www.geeksforgeeks.org/half-subtractor-in-digital-logic/

[4] : GeeksforGeeks : Full-Subtractor in digital logic, Retrieved March 9, 2024 from https://www.geeksforgeeks.org/full-subtractor-in-digital-logic/

[5] : GeeksforGeeks : Comparator in digital logic, Retrieved March 10, 2024 from https://www.geeksforgeeks.org/magnitude-comparator-in-digital-logic/