BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY (ENCS2110)

"Report 1"

Comparators, Adders, and Subtractors

Prepared by:

Name: Eman Asfour **Number**: 1200206

Partners:

Name: Worood Fathi **Number**: 1210412

Name: Rand Awadallah **Number**: 1211963

Instructor: Dr. Jamal Seyam

Assistant: Eng.Haleema Hmedan

Section: 4

Date: 8 / 14 / 2023 **Place**: Masri 107

1. Abstract:

In order to gain a thorough understanding of the complex design and management principles governing digital comparators, this experiment conducts a thorough investigation. It broadens its scope to include related logical circuits and investigates the underlying principles of half- and full-subtractor circuits, as well as half- and full-adders built with simple logic gates and integrated circuits (ICs). Additionally, the experiment concludes with the use of inexpensive logic gates and integrated circuits to implement these components practically.

2. Table of contents

3. Table for Figures

4. Table of Tables

5. Theory:

- **Half Adder**

"A half-adder is a digital 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 is the least significant bit (LSB) of the result, while the CARRY output is the most significant bit (MSB) of the result, indicating 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.

The half adder is a basic building block for more complex adder circuits such as full adders and multiple-bit adders. It performs binary addition of two single-bit inputs, A and B, and provides two outputs, SUM and CARRY.

The SUM output is the least significant bit (LSB) of the result, which is the XOR of the two inputs A and B. The XOR gate implements the addition operation for binary digits, where a "1" is generated in the SUM output only when one of the inputs is "1".

The CARRY output is the most significant bit (MSB) of the result, indicating whether there was a carry-over from the addition of the two inputs. The CARRY output is the AND of the two inputs A and B. The AND gate generates a "1" in the CARRY output only when both inputs are "1" ". [1]

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 1: The Truth Table of The Half-Adder [2]

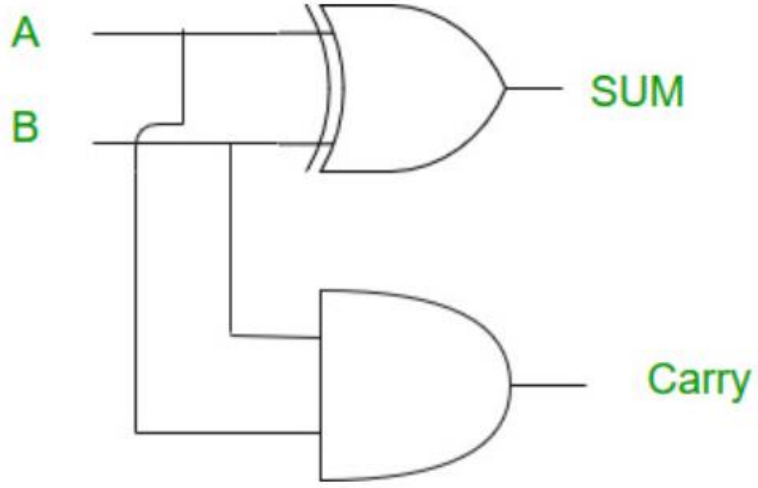


Figure 2: Half Adder

- **Full Adder :**

"Full Adder is the adder that adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. The C-OUT is also known as the majority 1's detector, whose output goes high when more than one input is high. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to another. We use a full adder because when a carry-in bit is available, another 1-bit adder must be used since a 1-bit half-adder does not take a carry-in bit. A 1-bit full adder adds three operands and generates 2-bit results." [3]

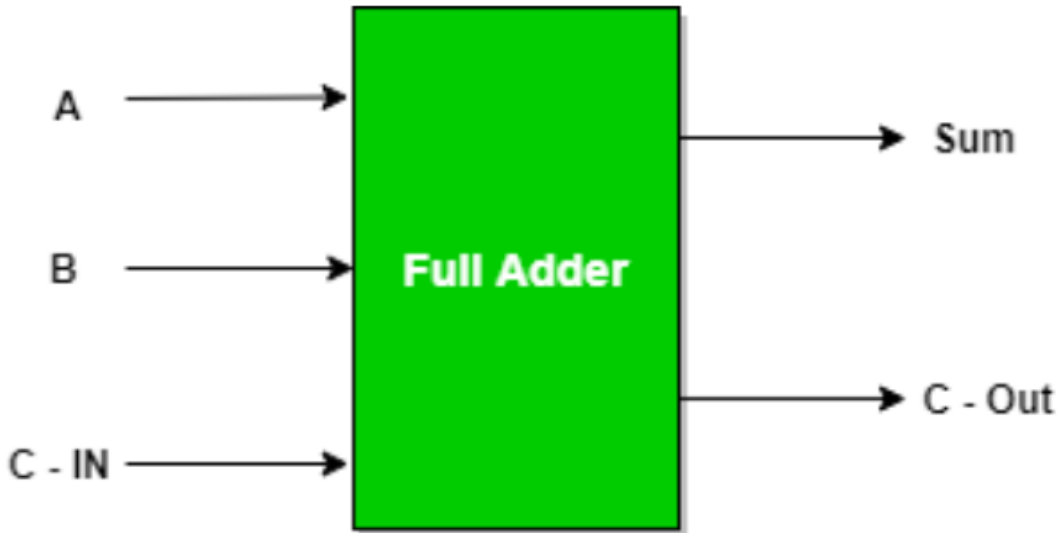


Figure 3: Full Adder

Adder Truth Table:

Inputs			Outputs	
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

Logical Expression for SUM: = A' B' C-IN + A' B C-IN' + A B' C-IN' + A B C-IN = C-IN (A' B' + A B) + C-IN' (A' B + A B') = C-IN XOR (A XOR B) = (1,2,4,7)

Logical Expression for C-OUT: = A' B C-IN + A B' C-IN + A B C-IN' + A B C-IN = A B + B C-IN + A C-IN = (3,5,6,7)

Figure 4: The Truth Table of Full Adder

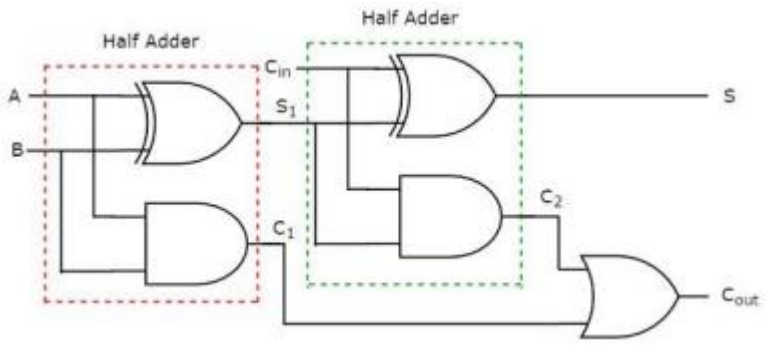


Figure 5: Full Adder Circuit. [4]

"To perform additions of numbers greater than 2-bits in length, the connection shown in Figure 2.5, or "Parallel Input" should be used to generate sums simultaneously. However, the sum of the next adder will be stable only after the previous adder's carry has stabilized. For example, in Figure 2.5, the sum of FA2 will not be stable unless the carry of FA1 is stable."[4]

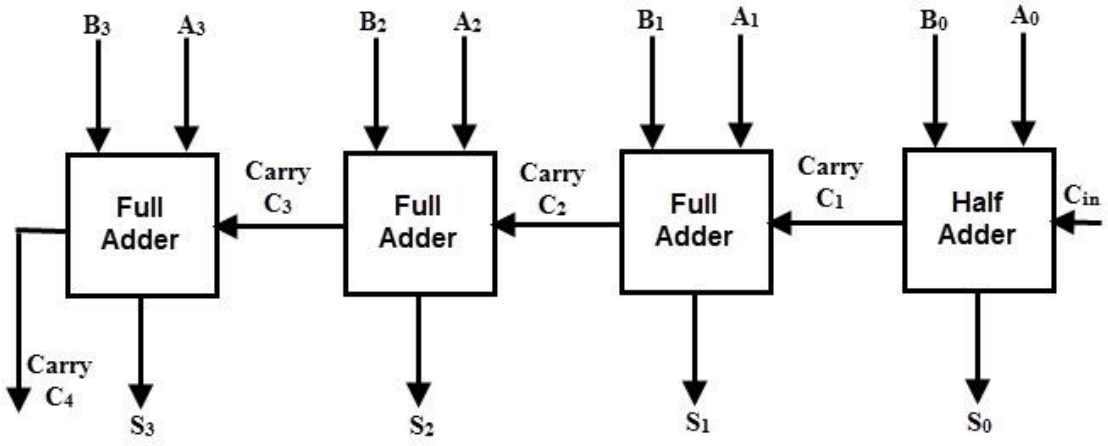


Figure 6: 4 bit Adder [5]

"When FA1 adds A1 and B1, a sum S1 and a carry C1 is generated. C1 will be added to A2 and B2 by FA2, generating another sum S2 and another carry C2. In the case of Figure 2.5, the sum of the four adders do not stabilize at the same time, delaying the adding process. This delay can be eliminated by using the "Look-Ahead" adder. Binary adders can be converted into BCD adders. Since BCD has 4 bits with the largest number being 9, and the largest 4-bit binary number is equivalent to 15, there is a difference of 6 between the binary and the BCD adder: Under the following conditions, 6 must be

added when binary adders are used to add BCD codes: 1. When there is any carry. 2. When the sum is larger than 9. If the order of priority is S8, S4, S2, S1, and the sum is larger than 9 then S8 x S4 + S8u x S2. If any carry is involved, assuming the carry is CY, under this term, 6 must be added: CY + S8 X S4 + S8 X S2". [4]

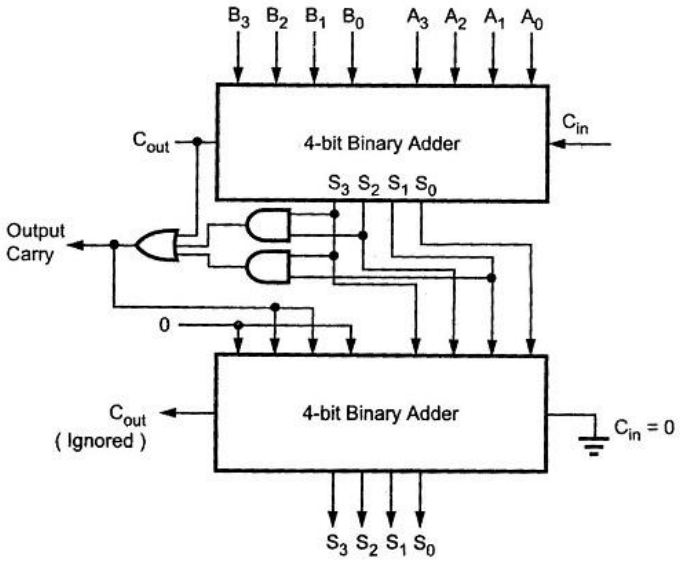


Fig. 3.32 Block diagram of BCD adder

Figure 7: BCD Adder [6]

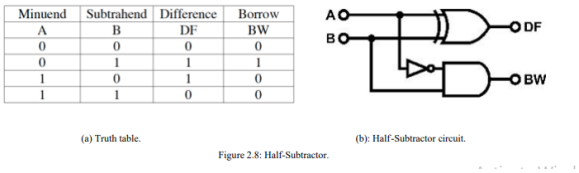
"A half-subtractor circuit is a digital electronic circuit that performs subtraction of two single-bit binary numbers (operands) and produces the difference and borrow outputs. It has two inputs, the minuend (A) and subtrahend (B), and generates two outputs, the difference (D) and the borrow (Borrow Out) signal. The full-subtractor circuit extends this concept to handle three inputs, including a borrow input (Bin), allowing it to subtract two single-bit binary numbers and a borrow input, producing the difference output and a borrow output. These subtractor circuits form fundamental building blocks for more complex digital arithmetic and logic operations in computers and other digital systems." [7]

Subtractors are classified into two types:

- **A half subtractor**

A half subtractor is a combinational logic circuit. It has two inputs (one bit each) termed A and B that generates difference (D) and borrow (DF) as output. Half subtractor is designed using three logic gates that are AND gate, NOT gate, and XOR gate. The

output of difference is obtained from the XOR gate and the output of borrow is obtained from the AND gate.



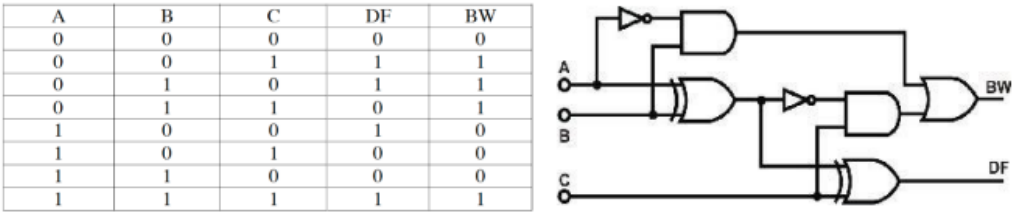
Minuend	Subtrahend	Difference	Borrow
A	B	DF	BW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(a) Truth table. (b): Half-Subtractor circuit.

Figure 2.8: Half-Subtractor.

Figure 8: Half Subtractor[4]

A full subtractor is a combinational logic circuit. It has three inputs (one bit each) termed A, B, and C that generate difference (D) and borrow (DF) in output. A full subtractor is designed using two XOR gates, two NOT gates, two AND gates, and one OR gate. The output of Difference (D) is obtained from the XOR gate and the output of borrow (DF) is obtained from the OR gate. As shown in the circuit below.



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

(a) Truth table. (b): Full-Subtractor circuit.

Figure 2.9: Full-Subtractor.

Figure 9: Full Subtractor

- **Comparator Circuit**

In an electronic circuit known as a comparator, two input voltages are compared, and the output signal that results shows which input is greater. In essence, it establishes the relative magnitudes of the input voltages and produces a logical output, frequently denoted as high or low (1 or 0), to denote the comparison's outcome. Many different applications, including analog-to-digital converters, voltage level detection, waveform analysis, and decision-making in digital systems, make extensive use of comparators. They are essential parts of electronics that allow decision-making based on signal and voltage comparisons.

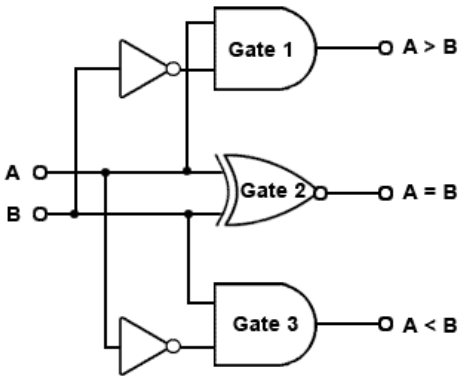


Figure 10: Comparator circuit [8]

"At least two numbers are required to perform any comparison. The simplest form of the comparator has two inputs. If the two inputs are called A and B, then there are three possible outputs: A>B, A=B, and A<B. In actual applications, 4-bit comparators are used most often. In a 4-bit comparator, each bit represents 20 , 21 , 22 , and 23 . Comparison will start from the most significant bit (23), if input A is greater than input B at the 23 bits, the "A>B" output will be in the high state. If A and B are equal at the 23 bits, the comparison will be carried out at the next highest bit (22). If there is still no result at this bit, the process is repeated again at the next bit. At the lowest bit (20), if 30 the inputs are still equal then the "A=B" output will be in the high state. Figure 2.1 shows the schematic and symbol of a 4-bit comparator." [4]

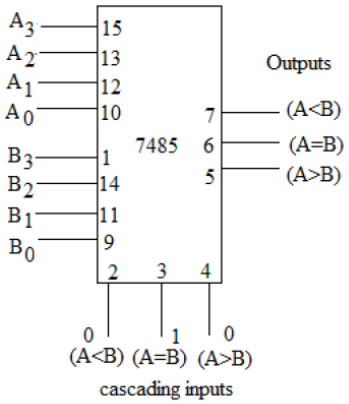


Figure 11: 4-bit Comparator [9]

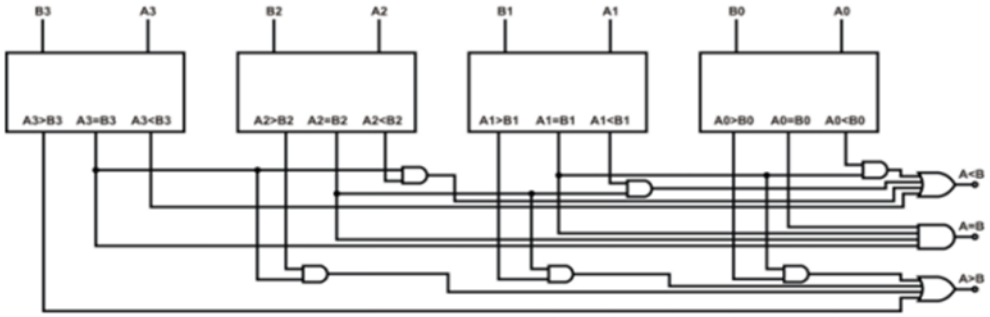


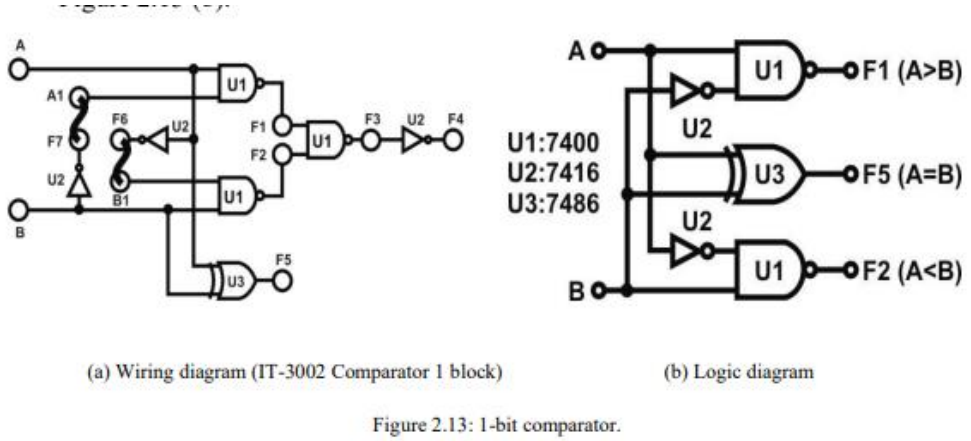
Figure 12: A 4-bit comparator constructed with four 1-bit comparators [4]

3. Procedure

3.1 Comparator Circuits:

A. Constructing Comparator with Basic Logic Gates

The circuit shown in figure 13 was connected.



(a) Wiring diagram (IT-3002 Comparator 1 block) (b) Logic diagram

Figure 2.13: 1-bit comparator.

Figure 13 : IT-3002 block Comparator [4]

A high-state voltage was used to activate the inputs, and Data Switches SW1 and SW2 were connected to inputs A and B, respectively. Logic Indicators L1, L2, and L3 were connected to outputs F1, F2, and F5, respectively, and the outputs were triggered by a low-state voltage.

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 1: Constructing Comparator with Basic Logic Gates

Let's deconstruct the output values' meanings in light of the given data. When both SW2 and SW1 are 0 (B = 0, A = 0):F1 = A = B = 0 F2 = 1 F5 = 0. When SW2 is 0 and SW1 is 1 (B = 0, A =1):F1 = A > B, so F1 = 1F2 = 1 F5 = 1. In conclusion, the data depicts how a comparator or decision-making circuit would behave. It generates three output signals (F1, F2, and F5) from two binary inputs (A and B). The outputs show whether or not A is greater than B, whether or not the inputs are equal, and a combination of A and B-related conditions (F5). Different digital systems that require logical comparisons and decision-making can benefit from using this type of circuit.

B. Constructing Comparator with TTL IC

In the experiment, a 74LS85 4-bit Comparator IC called IT-3002 U5 was used. On the module's left, where SW1 and AB were connected, connections were made. The input pins A0–A3 and B0–B3 of the 74LS85 were connected to the BCD rotary switch. The outputs AB were connected to L1, L2, and L3, respectively, on the right side of the module. The experimental setup involved choosing A0˜A3=As, B0˜B3=Bs, and As=Bs from the rotary switch to configure the comparator. Following the cascading input sequences listed in Table 2, the resulting outputs were recorded.

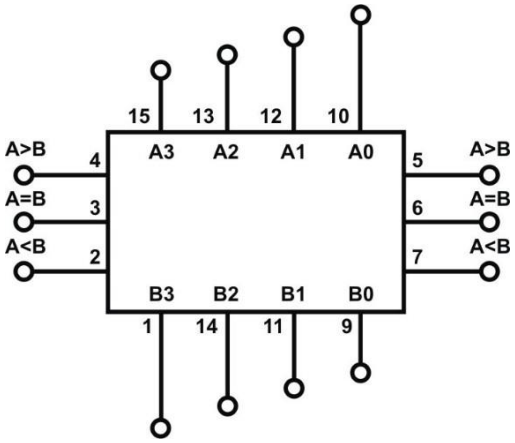


Figure 14: 4-bit Comparator IC (IT-3002 block Comparator 2).

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	0	0

Table 2 Constructing Comparator with TTL IC:

Input-output relationships for a comparator circuit are shown in the data. The output A>B is active when A is greater than B, and the output A=B is active when A is equal to B. On the other hand, the output AB is active when A is less than B. The table shows how these outputs react to various input conditions.

3.2 Half- and Full-Adder Circuits

The Half-Adder block was positioned, and the setup of the IT-3003 module was carried out. The assembly of the half-adder circuit employed U5 and U6. The +5V output from the fixed power supply was connected to module IT-3003's +5V input. Inputs A and B were linked to

Data Switches SW0 and SW1, respectively, while outputs F1 and F2 were connected to Logic Indicators L1 and L2. By adhering to the input sequences detailed in Table 2.3, the corresponding outcomes were duly recorded.

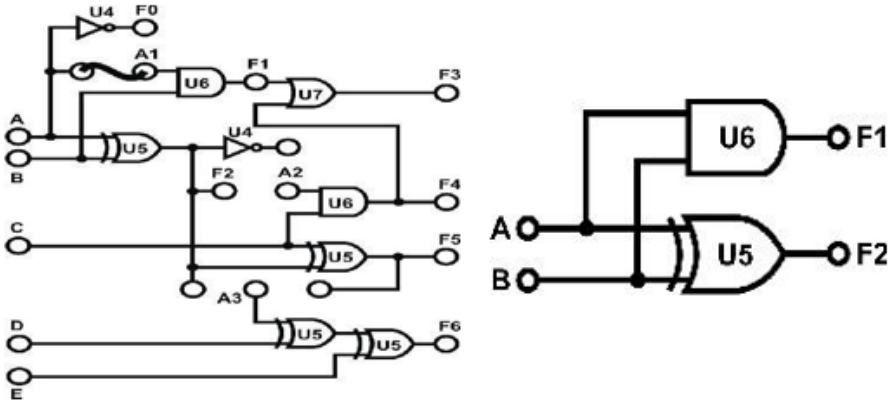


Figure 15: Half Adder

INPUTS		OUTPUTS	
SW1 (B)	SW0 (A)	F1 (CARRY)	F2 (SUM)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 3 : Half Adder

A half-adder circuit's input-output relationships are demonstrated by the given data. F1 (CARRY) and F2 (SUM) produce the appropriate outputs from the inputs SW1 (B) and SW0 (A). The table demonstrates how these outputs react to various input combinations.

The connections to SW1, SW2, and SW3 were made by A, B, and C, respectively. The preceding carry was represented by C, while A and B served as augends. L1 and L2 were connected by F3 and F5, respectively. The input sequences listed in Table 4 were followed in order to record the output states.

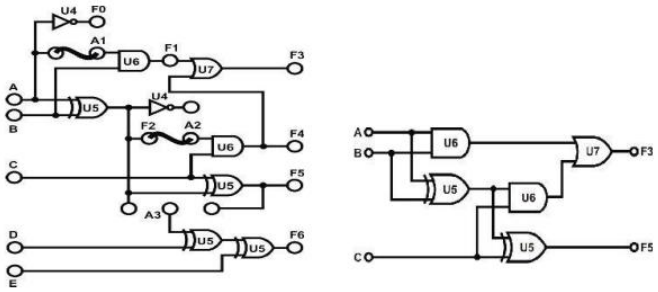


Figure 16 : Full Adder

OUTPUTS			OUTPUTS	
SW3 (C)	**SW2 (B)**	**SW1 (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**	0	1
1	**1**	**1**	1	1

Table 4: Full Adder

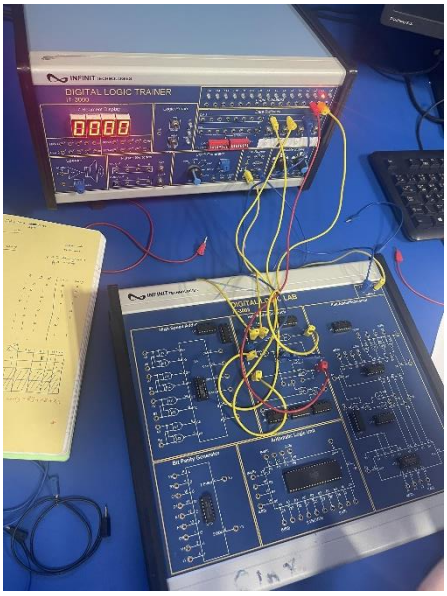


Figure 17: Circuit Connection for Full Adder

This data represents the operational states of a full adder circuit, demonstrating the relationship between input combinations and resulting carry and sum outputs.

3.3 Half- and Full Subtractor Circuits

The Module IT-3003 was configured, and the Half-Adder block was positioned. Inputs A and C were connected to Data Switches SW0 and SW2, respectively. The outputs were directed as follows: F2 to Logic Indicator L1, F1 to L2, F3 to L3, and F5 to L4. In the case where C=0, the circuit functions as a half-subtractor, with F1 functioning as the borrow output, F2 as the difference, F5 equal to F2, F4 equal to 0, and F3 equal to F1. Conversely, when C=1, the circuit operates as a full subtractor, with F3 serving as the borrow output and F5 as the difference.

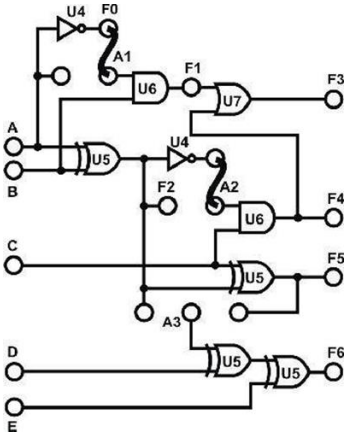


Figure 19: Half- and Full Subtractor Circuits

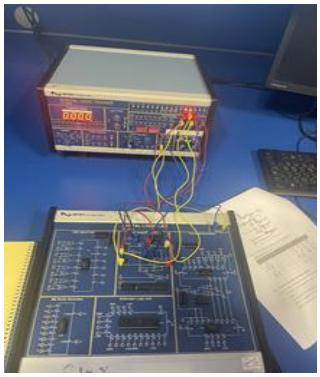


Figure 18: Circuit Connection

	INPUTS			OUTPUTS			
	C	A	B	F1	F2	F3	F5
Half-subtractor Half-adder	0	0	1	1	1	1	0
	0	0	0	0	0	0	0
	0	1	1	0	0	0	0
	0	1	0	1	0	0	1
Full-subtractor Full-adder	1	0	0	0	0	1	1
	1	0	1	1	1	1	0
	1	1	0	1	0	0	0
	1	1	1	0	0	1	1

Table 5: Half- and Full Subtractor Circuits

Table 5 shows the behaviors of half-subtractor and half-adder circuits and their relationships between input and output. In the Half-Subtractor, when the inputs are 0 or 1, the outputs are 1 or 0, when the inputs are 1 or 0, and when the inputs are 1 or 1,

the outputs are 0 or 1. On the other hand, the Full-Subtractor operates as follows: when inputs are 1 0 0, outputs are 0 0 1; when inputs are 1 0 1, outputs are 1 1 0; when inputs are 1 1 0, outputs are 0 0 0; and when inputs are 1 1 1, outputs are 0 1 1.

3.4 Constructing 4-Bit Full-Adder with IC

4-Bit Adder Configuration using IT-3003:

Inputs:

- **Addends (X0~X3):** Connected to DIP switches DIP2.0~2.3.
- **Augends (Y0~Y3):** Connected to DIP switches DIP1.0~1.3.

Internal Connections:

- **Y5 (output):** Connected to SW0 to provide the carry-in input.
- **XOR Gates U8 (Y0~Y3 outputs):** Used as buffers for Y0~Y3.

Outputs: Sum Outputs (F1, F8, F9, F10, F11): Connected to LEDs L1~L5 for displaying results.

This setup transforms the IT-3003 into a functional 4-bit adder. The addends (X0~X3) and augends (Y0~Y3) are provided via DIP switches, and the output sums are displayed on LEDs (L1~L5). Y5 serves as the carry-in input, and XOR gates U8 act as buffers for the augend inputs. This concise configuration demonstrates the 4-bit addition process effectively.

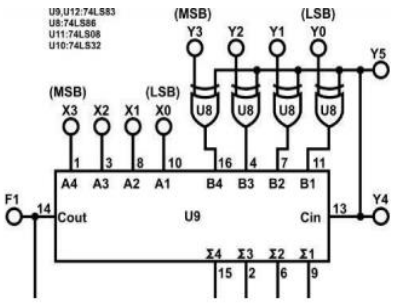


Figure 21: Wiring Diagram (IT-3003 4-bit Full-Adder block).

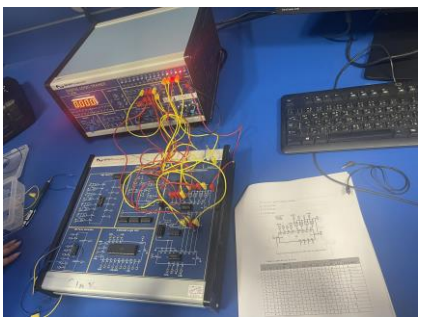


Figure 20: Circuit Connection

INPUTS								OUTPUTS				
Y3	Y2	Y1	Y0	X3	X2	X1	X0	Σ4	Σ3	Σ2	Σ1	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 6 : Constructing 4-Bit Full-Adder with IC

The outputs for each set of inputs are listed in the table. The columns from ∑4 to F1 carry the outputs from the 4-bit adder circuit based on the supplied inputs and represent the binary representation of the sum.

3.5 Constructing 4-Bit Full-Subtractor with IC

The Full Adder module of IT-3003 was employed as the foundational component. Input connections included X3~X0 (minuend) linked to DIP Switch 1.31.0, Y3Y0 (subtrahend) to DIP 2.32.0, and Y5 connected to SW0. Outputs were established as follows: F1 was linked to L4, while F1~1F8 was connected on, it is advised to set SW0 to "1" (or Cin of U9=1).

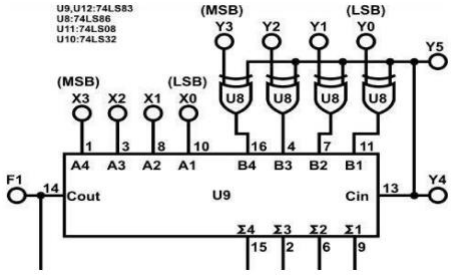


Figure 23 : Wiring Diagram for full subtractor (IT-3003 4-bit Full-Adder block)

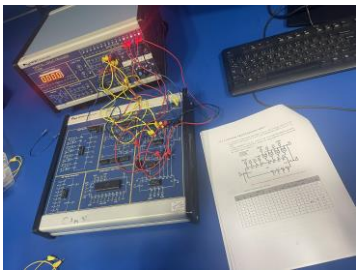


Figure 22: Circuit Connection

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 7 Constructing 4-Bit Full-Subtractor with IC

3.6 Constructing BCD Adder

The circuit was established, with X0~X3 inputs connected to DIP 1.01.3, Y0Y3 inputs connected to DIP 2.02.3, and Y5 set to "0". Utilizing 74LS83 look-ahead 4-bit BCD adders U9 and U12, the outputs F8~F11 were directed to the inputs of the 7-Segment display SEG-1. F1 and F2 were linked to Logic Indicators L4 and L5, while F4~F7 of U12 were routed to another 7-Segment display SEG-3, and F3 was connected to L10.

The calculation involved summing X0~X3 with Y0~Y3, with F8F11 representing the sum and F1 being the carry. The input sequences for X0~X3 and Y0~Y3 were sourced from Table 8, and the resulting outputs were recorded.

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

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

Table 8: Constructing BCD Adder

This table appears to be demonstrating binary addition with different inputs and their corresponding outputs. It's not clear if U9 and U12 are part of a larger system, but the analysis focuses on the basic addition process and carry propagation.

3.7 High-Speed Adder Carry Generator Circuit

The block High-Speed Adder in the IT-3003 module incorporated the use of U3 (74182) to build a carry generator circuit. Inputs A0~A3 (addends) were linked to DIP Switches 1.01.3, while B0~B3 were connected to DIP2.02.3. Additionally, Cn was directed to SW0, with SW0 being configured to "0".

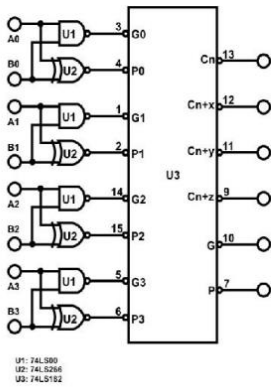


Figure 25: Carry generator circuit [4]

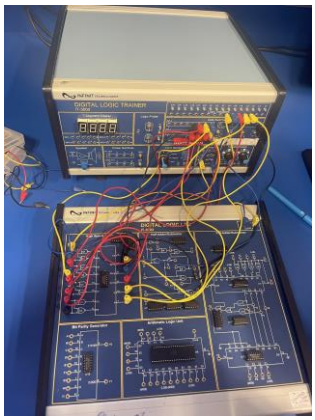


Figure 24: Carry generator circuit

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

1	**1**	**1**	**1**	**1**	**1**	**0**	**1**	1	1	1	0	1
0	**1**	**1**	**1**	**0**	**1**	**1**	**0**	0	1	1	1	1
1	**0**	**0**	**1**	**0**	**1**	**0**	**1**					

Figure 26: High-Speed Adder Carry Generator Circuit

This table appears to showcase binary addition scenarios with their corresponding generate (G) and propagate (P) outputs, which are typical components in carry generator circuits. The outputs indicate whether a carry should be generated or propagated based on the input values.

4. Conclusion

Through comprehensive hands-on engagement, encompassing pre-lab preparation, experiment execution, and post-lab analysis, we have achieved a deep understanding of the construction and operational principles underlying digital comparators. Proficiently employing fundamental logic gates and integrated circuits (ICs), we successfully implemented both half- and full-adders. This extended to mastering the intricate construction and operational intricacies of a 4-bit adder unit/IC, facilitating seamless addition of 4-bit numbers. Collaborative teamwork, expertly coordinated and verified by the Assistant, ensured precise data collection and analysis. Our grasp of theoretical complements translated into the construction of precise half and full subtractor circuits. In summation, the experiment was executed with an exemplary level of professionalism, yielding a profound comprehension of the subject matter.

5. References

[1] "Half Adder in Digital Logic," [Online]. Available: https://www.geeksforgeeks.org/half-adder-in-digital-logic/. [Accessed Aug /15/20203 7:00PM]

[2] [Online]. Available: 2 https://www.javatpoint.com/half-adder-in-digital-electronics. [Accessed Aug /15/20203 9:30PM]

[3] [Online]. Available: [3] https://ritaj.birzeit.edu/bzu-msgs/attach/2357758/Digital+lab+manual+2023.pdf. [Accessed Aug /15/20203 6:60PM]

[4] [Online]. Available: [4] https://ritaj.birzeit.edu/bzu-msgs/attach/2357758/Digital+lab+manual+2023.pdf.

[5] [Online]. Available: [5] https://www.electronicshub.org/binary-adder-and-subtractor/. [Accessed Aug /15/2023 8:40 PM]

[6] [Online]. Available: [6] https://www.eeeguide.com/bcd-adder-circuit/.[Accessed Aug /15/20203 10:40 PM]

[7] [Online]. Available: [7] https://freakengineer.com/half-subtractor-and-full-subtractor/. [Accessed Aug /15/20203 11:50]

[8] [Online]. Available: [8] https://learnabout-electronics.org/Digital/dig43.php. [Accessed Aug /15/20203 11:57]

[9] [Online]. Available: [9] https://www.deldsim.com/study/material/15/implementation-of-4-bit-magnitude-comparator-using-ic-74ls85/.[Accessed Aug /15/20203 7:09]