Course Name:	**Digital Electronics (116U40L303)**	**Semester:**	**III**
Date of Performance:	**24 / 08 / 2023**	**Batch No:**	**A - 3**
Faculty Name:	**Kirti Sawalani**	**Roll No:**	**16014022050**
Faculty Sign & Date:		**Grade/Marks:**	**___ / 25**

Experiment No.: 4
Title: BCD Adder

Aim and Objective of the Experiment:

To study BCD Adder Using IC 7483

COs to be achieved:

CO2: Design combinational circuits using MSI devices.

Theory:

BCD (Binary Coded Decimal) addition and binary addition are both arithmetic operations performed on binary numbers, but they are designed for different purposes and have distinct characteristics. Let's break down how BCD addition is different from binary addition:

1. Representation:
   - BCD: Binary Coded Decimal represents each decimal digit (0 to 9) using a 4-bit binary code. It is often used in applications where decimal arithmetic is essential, such as in financial calculations, displays, and input/output systems.
   - Binary: Binary numbers represent values in base-2 using only two digits, 0 and 1. It is the fundamental numeral system used in digital electronics and computing.

2. Number Range:
   - BCD: BCD is restricted to representing decimal digits, so each BCD digit ranges from 0000 to 1001. BCD does not go beyond these values.
   - Binary: Binary numbers can represent any integer value, positive or negative, with a varying number of bits. The range is determined by the number of bits used in the representation.

3. Addition Rules:
   - BCD: BCD addition involves adding two BCD digits, similar to binary addition, but with some specific rules to ensure the result stays within the valid BCD range (0 to 9). If the sum of two BCD digits is greater than 9, or if a carry occurs from a lower-order digit, a correction is needed to maintain BCD validity. Adding 6 (0110 in BCD) corrects the result while accounting for the carry.
   - Binary: Binary addition follows the same rules as BCD addition when it comes to carrying over digits, but there's no specific restriction on the range of values. The result of binary addition can be any binary value, and there's no need for correction to ensure validity.

4. Purpose:
   - BCD: BCD addition is mainly used in applications where decimal calculations and accurate representation of decimal values are crucial, such as in calculators, digital displays, and financial systems.
   - Binary: Binary addition is the basis for all digital arithmetic and computation, including calculations in computer processors, memory operations, and general-purpose computation.

In summary, BCD addition and binary addition both involve the manipulation of binary numbers, but they serve different purposes. BCD addition is tailored for accurate decimal calculations and representation, while binary addition forms the foundation for all digital computation. BCD addition incorporates correction mechanisms to ensure valid decimal results within the limited range of decimal digits, whereas binary addition has a broader range of possible outcomes without specific restrictions.

Studying the BCD adder using the IC 7483 offers a practical insight into the world of digital arithmetic circuits and binary-coded decimal representations. The IC 7483 is a 4-bit binary adder that holds a unique significance in understanding how BCD numbers are processed and manipulated within digital systems. This exploration sheds light on how complex operations involving decimal digits are performed through binary manipulation.

The primary objective of this study is to comprehend the process of BCD addition using the IC 7483. Binary Coded Decimal (BCD) is a coding system that represents decimal digits using a 4-bit binary code. This code allows for the representation of decimal digits 0 to 9, with the first 10 4-bit combinations considered valid BCD representations.

The IC 7483 acts as a fundamental building block in this study. It consists of parallel binary adder circuits that are essential for performing the addition of BCD digits. The study involves connecting the BCD digits to the A and B inputs of the IC 7483 and analyzing the sum outputs (S) and carry outputs (Cout). The first adder circuit produces the initial binary sum output, which is then checked for invalid BCD values exceeding 9.

The study introduces an additional layer of complexity in the form of a combinational circuit that evaluates the sum outputs to detect invalid BCD values. When an invalid BCD value is detected, the combinational circuit triggers the need for correction. This correction process involves adding 6 (0110 in binary) to the original sum. This clever technique aligns with the binary manipulation required for BCD correction.

By engaging with the IC 7483 and the associated combinational circuit, learners gain hands-on experience in digital circuitry, binary arithmetic, error detection, and correction. This study bridges the gap between theoretical knowledge and practical application, fostering a deeper understanding of how digital systems operate on a fundamental level. The BCD adder using IC 7483 exemplifies the elegance and efficiency of digital design, where a simple circuit can orchestrate complex operations, ultimately contributing to the foundational knowledge of digital electronics and computation.

Truth Table & K-Map for BCD Addition:

- Truth table for BCD adder :

INPUTS				OUTPUT
S_3	S_2	S_1	S_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

- Kmap for BCD addition:

Ketaki mahajan
16014022050

S_3S_2 \ S_1S_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$$\therefore Y = S_3S_2 + S_3S_1$$

Logic Diagram of BCD Adder:

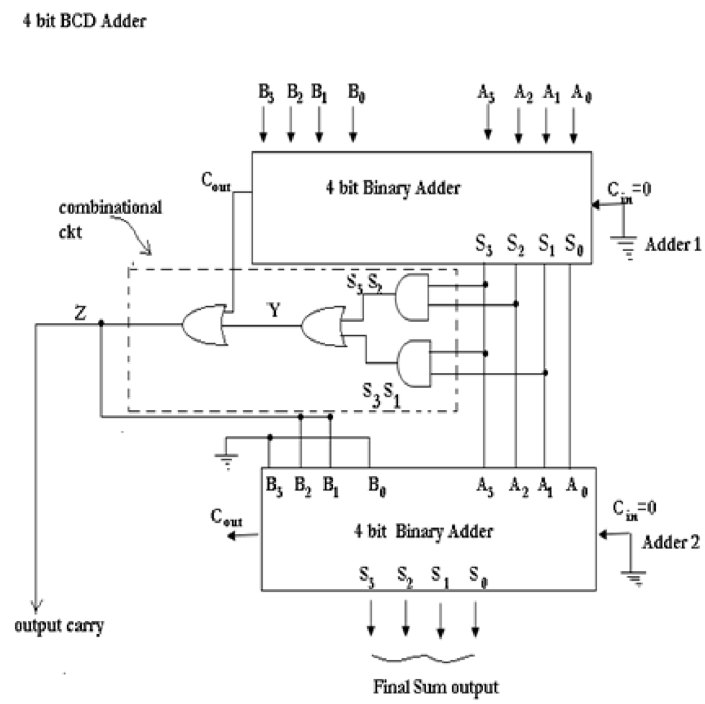
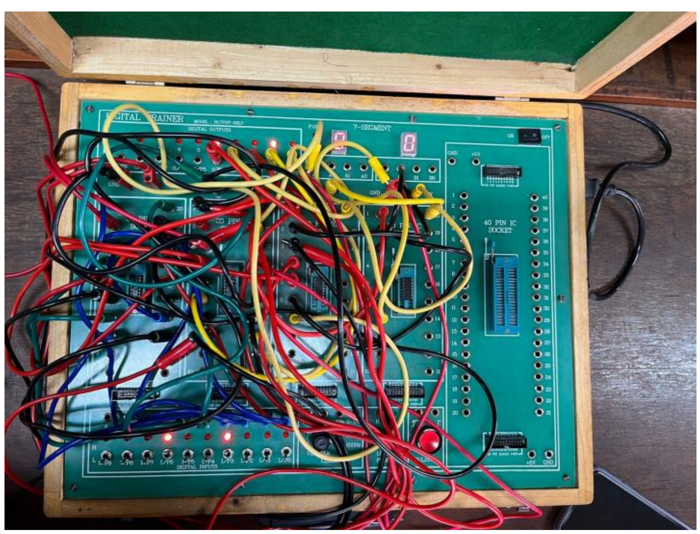


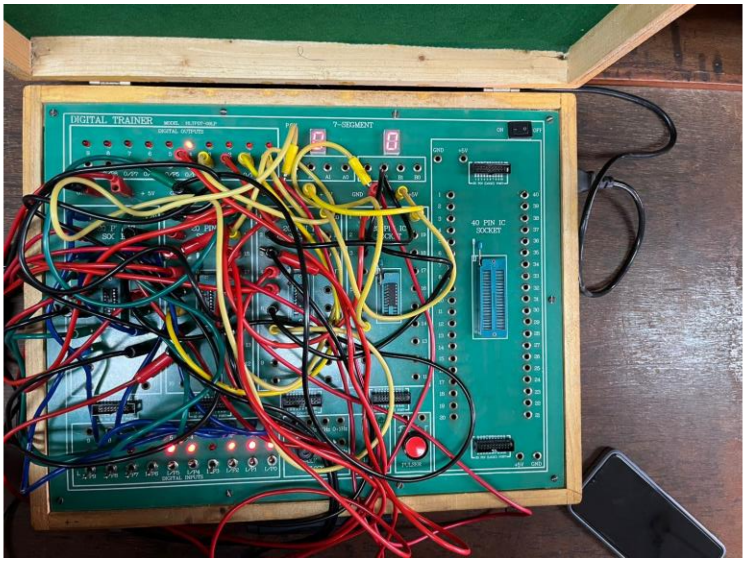
Fig5: 4-bit BCD adder

Circuit Cases:

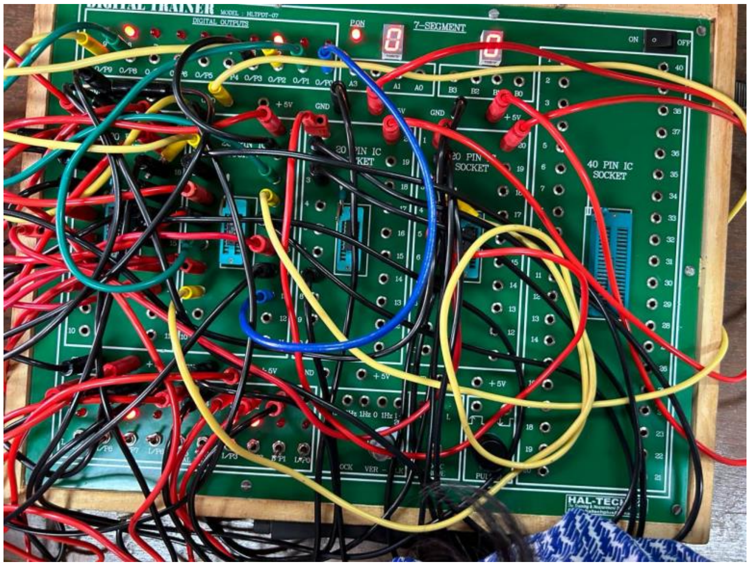
- **CASE 1:** 8 + 4 = 12



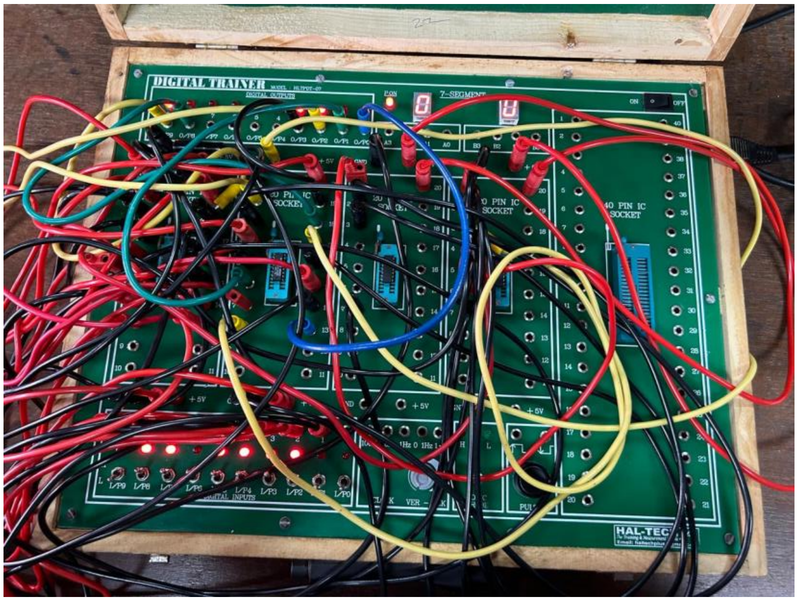
- **CASE 2:** 7 + 3 = 10



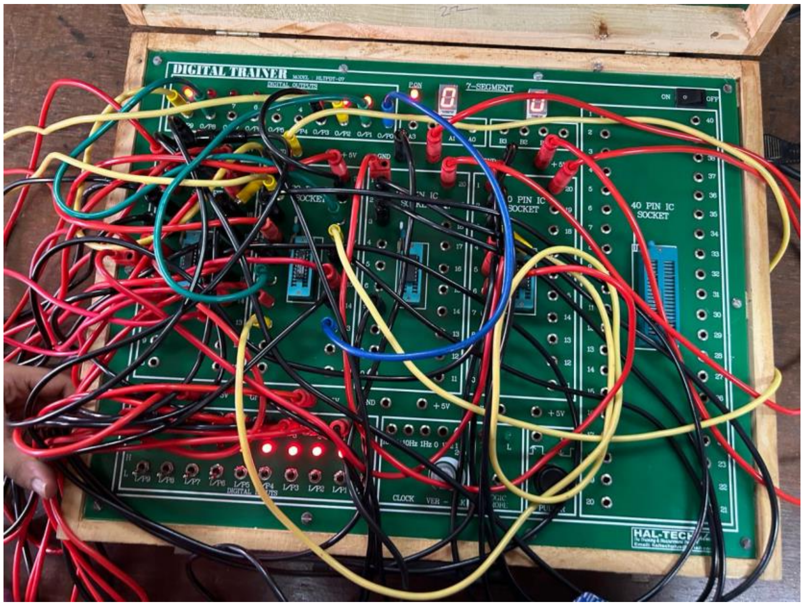
- **CASE 3:** 12 + 8 = 20



- **CASE 4:** 13 + 14 = 27



- **CASE 5:** 15 + 1 = 16



Stepwise-Procedure:

1. Make a truth table of BCD adder.
2. Obtain the expressions for sum and carry bit by solving K Map.
3. Implement the obtained expressions on a trainer kit.
4. Check for at least 5 cases using obtained expression to justify BCD adder.

Post Lab Subjective/Objective type Questions: (Must be handwritten)

1. **How does a BCD adder ensure that the result of adding two BCD digits remains within the valid range of decimal digits (0 to 9)?**

2. **Explain the purpose of the combinational circuit in a BCD adder. How does it detect and correct invalid BCD values?**

3. **If you're adding the BCD digits 3 (0011 in BCD) and 8 (1000 in BCD), what will be the output of the BCD adder, and will any correction be needed?**

4. **Contrast the rules and correction mechanisms used in binary addition and BCD addition when a carry occurs during addition.**

5. **In a BCD adder, why is the value 6 (0110 in BCD) added to the sum output when invalid BCD values are detected? How does this correction process work to yield a valid BCD result?**

6. **Implement a one-digit BCD Subtractor.**

Date: 24 /08 / 23

PAGE 1

• Post lab Questions :

1). How does a BCD adder ensure that the result of adding two BCD digits remains with the valid range of decimal digits (0 to 9) ?

A BCD (Binary coded decimal) adder works with BCD digits, where each digit (0 to 9) is represented by 4-bit binary code. When adding two binary digits, the adder ensures that the result remains within the ~~#~~ valid range of decimal digits by incorporating a correction mechanism.

If the sum of 2 BCD digits results in a value greater than 9 (1001 in BCD), it is an invalid BCD value. In this case, the adder detects the carry-out from the 4th bit & performs a correction by adding 6 (0110 in BCD) to the sum to get valid BCD result.

Ketaki Mahajan
16014022050

e.g.
We want to add 5 (0101) & 7 (0111),

- 0101 + 0111
- sum of these BCD digits is 12, 1100, greater than 9.
  ∴ INVALID sum value
- BCD adder checks if sum > 9
- To correct this, BCD adder adds 6 (0110) to sum,
  ∴ 1100 + 0110 = 10010
- Final result after correction is 0010 in BCD, which is decimal digit 2.

Date: 24 / 08 / 23
PAGE 2

2). Explain the purpose of the combinational circuit in BCD adder. How does it detect & correct invalid BCD values.

The combination circuit in a BCD adder performs the addition of 2 BCD digits and includes the carry logic and correction mechanism. It detects invalid BCD values by checking if the sum of BCD digits is greater than 9 (1001 in BCD).
If the condition is met, it generates a carry-out from the 4th bit and triggers the correction process by adding 6 (0110 in BCD) to the sum.
This ensures that the result remains within the valid range of BCD digits (0-9).

3). If you're adding the BCD digits 3 (0011 in BCD) & 8 (1000 in BCD), what will be the output of the BCD adder, and will any correction be needed.

- Adding 3 (0011 in BCD) & 8 (1000 in BCD) gives 11 (1011 in BCD).
- Sum is greater than 9 (1001 in BCD), correction is needed.
- Correction involves adding 6 (0110 in BCD) to the sum,

```
3:   0011
8: + 1000
    1011
6: + 0110
   11001
```

Ketaki Mahajan
16014022050

- Resulting in 17,

$\therefore \underset{\text{CARRY}}{1} \quad \underbrace{1001}_{7} \Rightarrow 17$ //

Date: 24 / 08 / 23

PAGE 3

4) Contrast the rules and correction mechanisms used in binary addition and BCD addition when a carry occurs during addition.

In binary addition, when a carry occurs, it simply propagates to the next higher-order bit. There is no need for correction as the binary system is based on powers of 2 and doesn't have the same decimal digits contraints as BCD.
In BCD addition, when a carry occurs and the sum of two BCD digits exceeds 9, it is an ivalid BCD value. The correction involves adding 6 (0110 in BCD) to the sum to bring it back within valid BCD range.

5). In a BCD adder, why is the value 6 (0110 in BCD) added to the sum output when invalid BCD values are detected? How does this correction process work to yield a value that is valid?

The values 6 (0110 in BCD) is added to the sum output to correct the invalid BCD value & bring it within the valid range. Adding 6 offsets the sum by a value equivalent to 6 in decimal.
This correction value ensures that the resulting BCD value after adding 6 will be a valid BCD digit between 0 & 9.

Ketaki Mahajan
16014022050

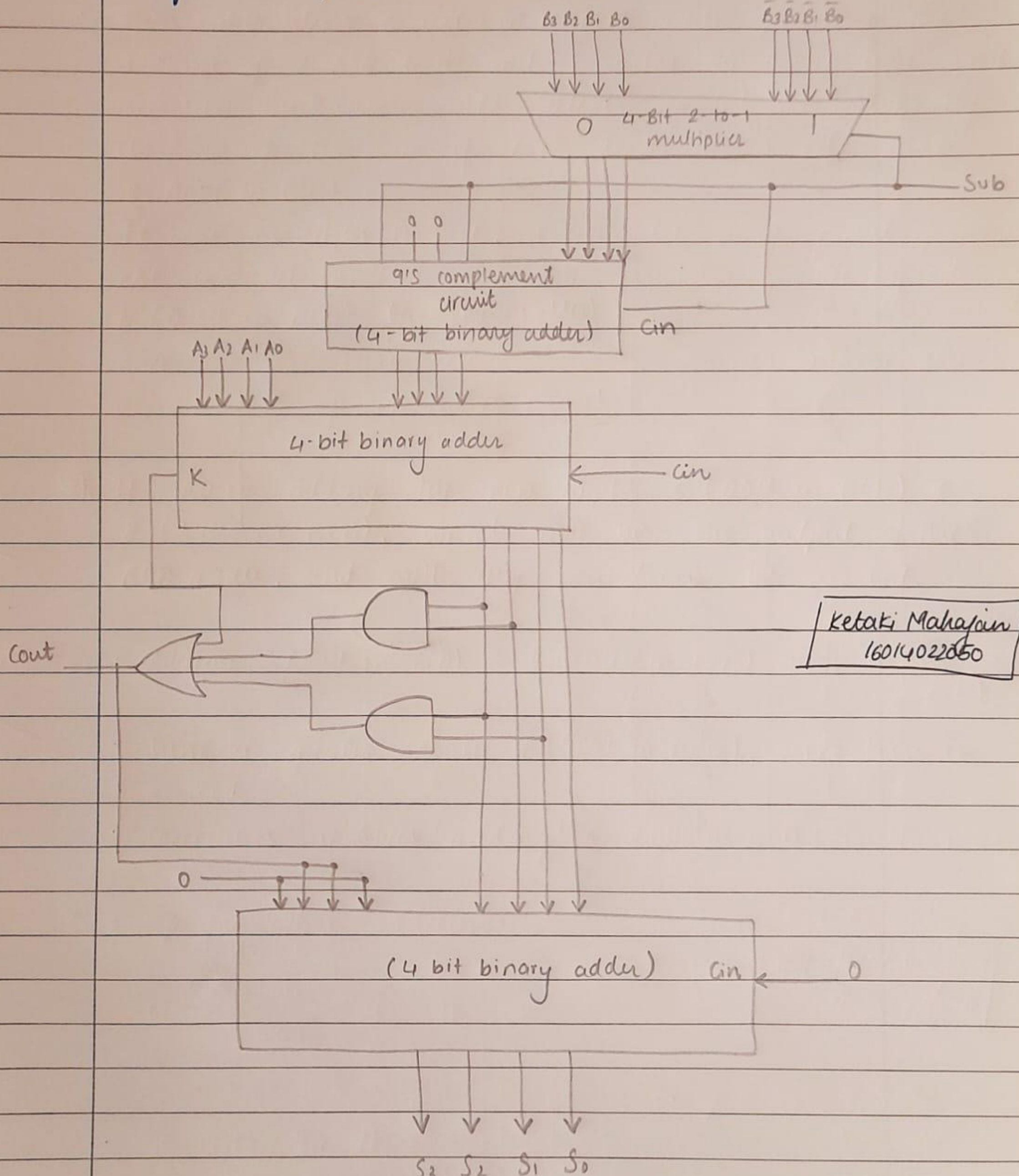
6). Implement a one-digit BCD subtractor.

To implement one-digit BCD subtractor, we can use IC-7483 which is 4-bit binary adder with BCD correction. To

Date: 24 / 08 / 23

PAGE 4

perform subtraction, you would use 2's complement method. ~~Since~~ Subtracting B from A is equivalent to adding 9's complement of B to A.



Conclusion:
The experiment focused on utilizing IC 7483 to perform BCD addition. This process maintains valid decimal results by detecting and rectifying invalid BCD outcomes, reaffirming the adherence of the sum to the decimal range of 0 to 9.

Signature of faculty in-charge with Date: