1. **Objectives of the experiment :**

* Understand the concept of binary addition and subtraction.
* Learn about half and full binary adders.
* Perform binary addition and subtraction using IC7483.
* Understand the concept of BCD addition and implement a BCD adder using IC7483

1. **Theory :**

An adder is a [circuit](https://www.computerhope.com/jargon/c/circuit.htm) that sums the amplitudes of two input signals. A half adder is a group of connected logic gates that create a logic circuit, incapable of handling addition for two numbers. A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full-adder adds three one-bit numbers.

The subtractor circuit is a series of arrangements of some logic gates arranged in such a way so that it can perform the subtraction operation of binary numbers.

The half adder circuit has two inputs: A and B, which add two input digits and generate a carry and sum.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Circuit of Half Adder

Truth Table of Half Adder

The full adder circuit has three inputs: A and C, which add the three input numbers and generate a carry and sum.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | y | z | S | C |
| 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 |

Truth Table of Full Adder

BCD adder A 4-bit binary adder that is capable of adding two 4-bit words having a BCD (binary-coded decimal) format. The result of the addition is a BCD-format 4-bit output word, representing the decimal sum of the addend and augend, and a carry that is generated if this sum exceeds a decimal value of 9.

Four binary digits count up to 15 (1111) but in BCD we only use the representations up to 9 (1001). The difference between 15 and 9 is **6**. If you want 9+1 to produce 10, which is 1 0000, you have to add 6 to make 1010 wrap to 1 0000.

Logical circuit with multiple full adders can be used for adding N-bit numbers and each full adder inputs a Cin,which is the Cout of the previous adder.Such kind of adder is known as Ripple Carry Adder, since each carry bit "ripples" to the next full adder.

BCD stands for Binary Coded Decimal. It's use lies in places where alphanumeric characters ranging from (0-9 and A-F) are to be displayed , especially in systems consisting solely of digital logic and not containing a microprocessor.

One such component which uses BCD to display is '[Seven Segment Display](https://en.wikipedia.org/wiki/Seven-segment_display)'.

Other advantages lie in the operations performed by the computer or circuit on these codes :

* Encoding and Decoding into BCD is easy .
* Hardware Algorithms (procedures) are easy to implement on BCD codes.
* Scaling by a factor of 10 (or a power of 10) is simple; this is useful when a decimal scaling factor is needed to represent a non-integer quantity .

The ALU (arithmetic logic circuitry) of a computer uses half adder to compute the binary addition operation on two bits.

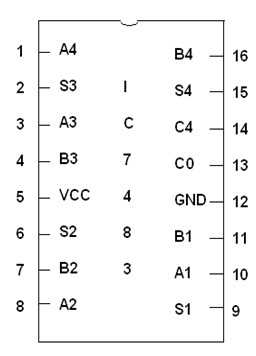
Half adder is used to make full adder as a full adder requires 3 inputs, the third input being an input carry i.e. we will be able to cascade the carry bit from one adder to the other. Ripple carry adder is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C(in), which is the C(out) of the previous adder. This kind of adder is called RIPPLE CARRY ADDER, since each carry bit "ripples" to the next full adder. Note that the first full adder (and only the first) may be replaced by a half adder.

The full adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers. The circuit produces a two-bit output, output carry and sum typically represented by the signals *C*out and *S.*

A full adder can be implemented in many different ways such as with a custom [transistor](https://en.m.wikipedia.org/wiki/Transistor)-level circuit or composed of other gates. One example implementation is with and . In this implementation, the final [OR gate](https://en.m.wikipedia.org/wiki/OR_gate)before the carry-out output may be replaced by an [XOR gate](https://en.m.wikipedia.org/wiki/XOR_gate) without altering the resulting logic. Using only two types of gates is convenient if the circuit is being implemented using simple IC chips which contain only one gate type per chip.

**New Apparatus:**

**IC 7483:** The 16-pin 7483 IC is a 4-bit full adder. That means, it can take two 4-bit binary numbers (A4A3A2A1 and B4B3B2B1) and calculate the sum (S4S3S2S1). The input carry (if any) is connected to C0 and the output carry is obtained from C4. Unlike most other ICs used so far, in the 7483, the 5V VCC needs to be connected to pin 5 and the ground to pin 12.

****Two 7483 ICs can be cascaded to form an 8-bit ripple-through-carry adder. The lower 4 bits of each number is used as input for the first 7483 and the output carry is connected to the input carry of the next 7483. The higher 4 bits of each number is used as input for the second 7483. The first IC provides the lower 4 bits of the sum and the second one provides the upper 4 bits.

Apparatus for experiment 1:

* Trainer board
* 1 x IC 7483 4-bit binary adder
* 1 x IC 7486 quadruple 2-Input XOR gates

Apparatus for experiment 2:

* Trainer board
* 2 x IC 7483 4-bit binary adder

Apparatus for experiment 3:

* Trainer board
* 2 x IC 7483 4-bit binary adder
* 1 x IC 7408 quadruple 2-Input AND gates
* 1 x IC 7432 quadruple 2-Input OR gates

**Figure B1:**

Pinout of IC7483

**Experimental Data and Result**

**F.1 Experimental Data (Binary Adder-Subtractor):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operation** | **M** | **A** | **B** | **C4** | **S4 S3 S2 S1** |
| 7 + 5 | 0 | 0111 | 0101 | 0 | 1100 |
| 4 + 6 | 0 | 0100 | 0110 | 0 | 1010 |
| 9 + 11 | 0 | 1001 | 1011 | 1 | 0100 |
| 15 + 15 | 0 | 1111 | 1111 | 1 | 1110 |
| 7 – 5 | 1 | 0111 | 0101 | 0 | 0010 |
| 4 – 6 | 1 | 0100 | 0110 | 1 | 1110 |
| 11 – 2 | 1 | 1011 | 0010 | 0 | 1001 |
| 15 – 15 | 1 | 1111 | 1111 | 0 | 0000 |

**Table F.1.1**

**F.2 Experimental Data (Ripple-Through-Carry Adder):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **A** | **B** | **Overflow Carry** | **Sum** |
| 7 + 5 | 0000 0111 | 0000 0101 | 0 | 0000 1100 |
| 18 + 19 | 0001 0010 | 0001 0011 | 0 | 0010 0101 |
| 72 + 83 | 0100 1000 | 0101 0011 | 0 | 1001 1011 |
| 129 + 255 | 1000 0001 | 1111 1111 | 1 | 1000 0000 |

**Table F.2.1**

**F.3 Experimental Data (BCD Adder):**

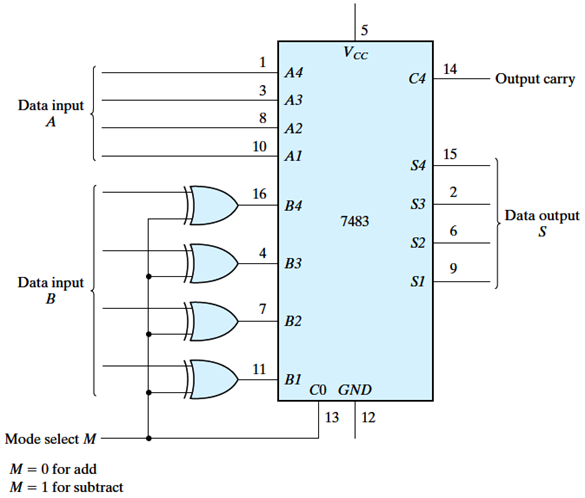
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decimal Value | Binary Sum | | | | | BCD Sum | | | | | |
| K | Z3 | Z2 | Z1 | Z0 | C | S3 | S2 | S1 | S0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 13 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 15 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 16 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 17 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 18 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 19 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

**Table F.3.1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **A** | **B** | **Overflow Carry** | **Sum** |
| 9 + 0 | 1001 | 0000 | 0 | 1001 |
| 9 + 1 | 1001 | 0001 | 1 | 0000 |
| 9 + 2 | 1001 | 0010 | 1 | 0001 |
| 9 + 3 | 1001 | 0011 | 1 | 0010 |
| 9 + 4 | 1001 | 0100 | 1 | 0011 |
| 9 + 5 | 1001 | 0101 | 1 | 0100 |
| 9 + 6 | 1001 | 0110 | 1 | 0101 |
| 9 + 7 | 1001 | 0111 | 1 | 0110 |
| 9 + 8 | 1001 | 1000 | 1 | 0111 |
| 9 + 9 | 1001 | 1001 | 1 | 1000 |

**Table F.3.2**

**Circuit Diagram:**

****

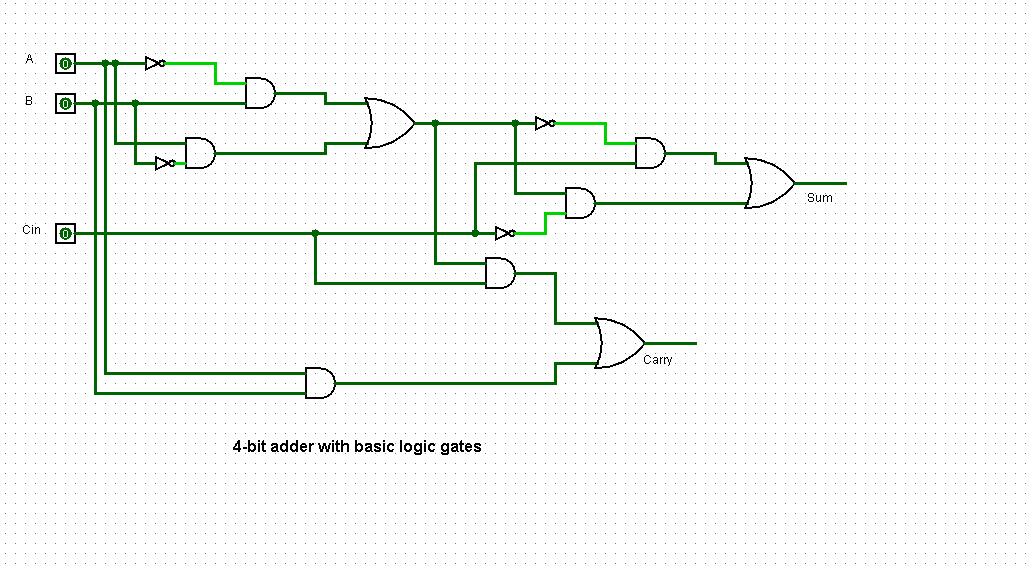
**Figure D.1.1** 4-bit adder-subtractor

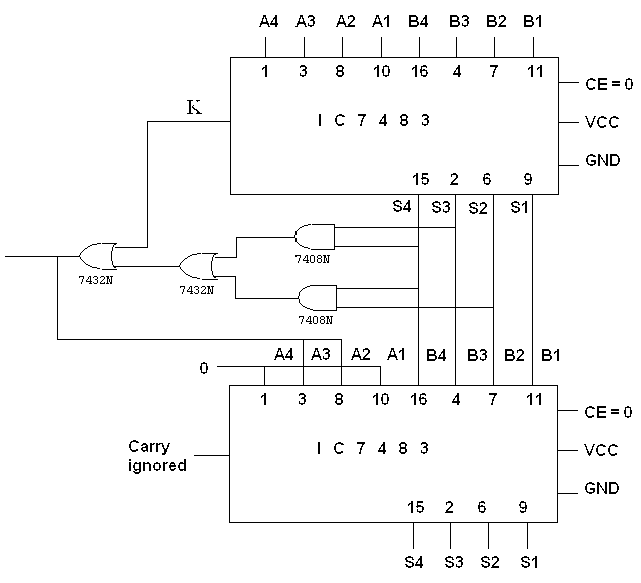
Comments: Here , we have used the XOR gate. And data B (B1,B2,B3,B4) is the input. We have to take 2 inputs for per XOR gate. So we have used “M” for another input to get the right output. In no.13 pin is used for M bit. And M is connected by all the XOR gate .M is known as control bit .It controls the whole output of XOR gate. Here M=0/1.M=0(for add) and M=1(for subtract).

Logisim Circuit schematic and Truth Table:

4-bit adder Truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin | A | B | Carry | 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 |



 **Figure D.3.1**

**Condition:**

* Sum < 9,Final carry=0,no correction.
* Sum < 9,Final carry=1,add+6.
* Sum > 9,Final carry=0,add+6.

**Result Analysis & Discussion**:

From the objectives of the experiment we can say that , in this experiment we have done binary addition and subtraction, half and full binary adders, BCD addition and have implemented a BCD adder using IC7483.

First we have done the binary adder subtractor by using 4 bit binary adder. It was not difficult to implement in the IC 7483. We constructed the circuit of Figure D.1.1. And also done the Logicim circuit schematic by using basic logic gates.

Then we have done our second part which was not very easy to do. It was challenging for us to implement the 8 bit ripple through carry binary adder using two 4-bit adders. We have done the operation in Table F.2.1. This hard work was made easy by our lab instructor. She helped us to implement this.

We finished our experiment by implementing the BCD adder. This was also tough for us. Here we have used 2 x IC 7483 4-bit binary adder, 1 x IC 7408 quadruple 2-Input AND gates, 1 x IC 7432 quadruple 2-Input OR gates. We have done the BCD Sum in Table F.3.1. and also have done some operations in Table F.3.2.This experiment was time consuming. So many wires were used . And it was done in Figure D.3.1.

Only problems were faced using lab equipment. Most of the elements including trainer board and some IC’s were partially damaged. With help of our lab instructor, Experiments were successfully done!

**Conclusion:**

By experimenting this experiment we have learn the binary addition and subtraction, the implementation of half and full adder, BCD addition and its implementation.