|  |  |  |
| --- | --- | --- |
| G:\nsu-logo.png  **North South University**  Department of Electrical & Computer Engineering    **LAB REPORT**  Course Name: **CSE332**  Experiment Number: 05     |  | | --- | | Experiment Name: **Binary Arithmetic** |     Experiment Date: 6/12/2020  Report Submission Date: 13/12/2020  Section: 03 | |
| Student Name: **Koushik Banerjee** | Score |
| Student ID: **1812171642** |  |
| Remarks: |

**LAB-05: Binary Arithmetic**

**Objectives:**

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

**Apparatus:**

• Trainer board

• 1 x IC 7483 4-bit binary adder

• 1 x IC 7486 quadruple 2-Input XOR gates

**Theory:**

Binary arithmetic is used in digital systems mainly because the numbers (decimal and floating-point numbers) are stored in binary format in most computer systems. All arithmetic operations such as addition, subtraction, multiplication, and division are done in binary representation of numbers. n mathematics, the four basic arithmetic operations applied on numbers are addition, subtraction, multiplications and division. ... It performs binary subtractions using a process known as for multiplication and division, the arithmetic and logic unit uses a method called shifting before adding the bits. The binary number system is an alternative to the decimal (10-base) number system that we use every day. Binary numbers are important because using them instead of the decimal system simplifies the design of computers and related technologies. ... But if the second digit is 1, then it represents the number 2. Some PLCs are equipped to carry out just the arithmetic operations of addition and subtraction, others the four basic arithmetic operations of addition, subtraction, multiplication, and division, and still others can carry out these and various other functions such as the exponential.

**Circuit Diagram:**

Attach the following Circuit Diagrams Screenshots.

(*After Discussion*)

1. E.1 Report Simulate a 4-bit adder in Logisim using basic logic gates. Provide a screenshot of the Logisim circuit schematic and truth table with your report
2. E.1 Report Construct the 4-bit adder-subtractor circuit of Figure D.1.1. Using logic IC’s
3. E.2 Report Draw the IC logic diagram for the 8-bit ripple-through-carry adder using logic ICs.
4. E.3 Report - Derive the circuit for the BCD adder

**Data Tables:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operation** | **M** | **A** | **B** | **C4** | **S4 S3 S2 S1** |
| 7 + 5 | 0 | 0111 | 0101 | 0 | 0010 |
| 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 | 0010 |
| 11 – 2 | 1 | 1100 | 0010 | 0 | 1010 |
| 15 – 15 | 0 | 1111 | 1111 | 0 | 0000 |

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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **A** | **B** | **Overflow Carry** | **Sum** |
| 7 + 5 | 00000111 | 00000101 | 0 | 00001100 |
| 18 + 19 | 00010010 | 00010011 | 0 | 00100101 |
| 72 + 83 | 01001000 | 01010011 | 0 | 10011011 |
| 129 + 255 | 10000001 | 11100001 | 1 | 10000000 |

**Table F.2.1**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decimal Value |  | B | inary 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 | 1 | 0 | 1 |
| 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 |

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

**Table F.3.1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operation** | **A**  (In Binary) | **B**  (In Binary) | **Overflow Carry** | **Sum**  (In Binary) | **SUM**  **(**Decimal**)** |
| 9 + 0 | 1001 | 0000 | 0 | 1001 | 9 |
| 9 + 1 | 1001 | 0001 | 0 | 01010 | 10 |
| 9 + 2 | 1001 | 0010 | 0 | 01011 | 11 |
| 9 + 3 | 1001 | 0011 | 0 | 01100 | 12 |
| 9 + 4 | 1001 | 0100 | 0 | 01101 | 13 |
| 9 + 5 | 1001 | 0101 | 0 | 01110 | 14 |
| 9 + 6 | 1001 | 0110 | 0 | 01111 | 15 |
| 9 + 7 | 1001 | 0111 | 1 | 10000 | 16 |
| 9 + 8 | 1001 | 1000 | 1 | 10001 | 17 |
| 9 + 9 | 1001 | 1001 | 1 | 11000 | 18 |

**Table F.3.2**

**Question Answering:**

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).

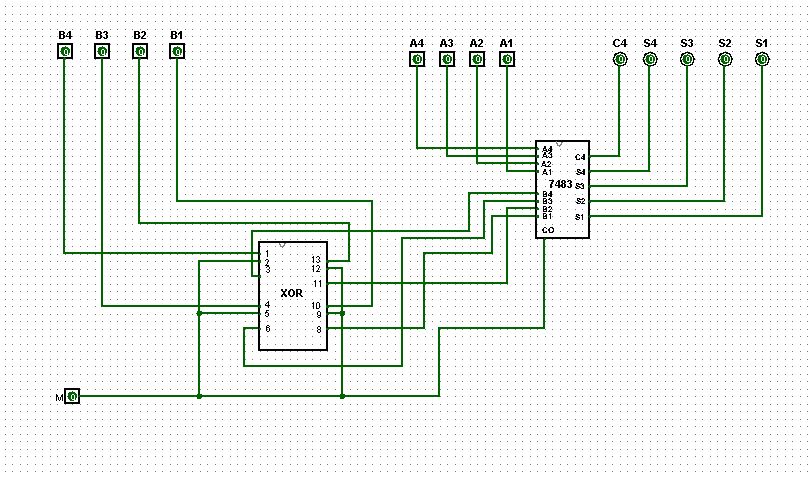
This Circuit Requires prerequisite knowledge of Exor Gate, Binary Addition and Subtraction, Full Adder.

Lets consider two 4-bit binary numbers A and B as inputs to the Digital Circuit for the operation with digits

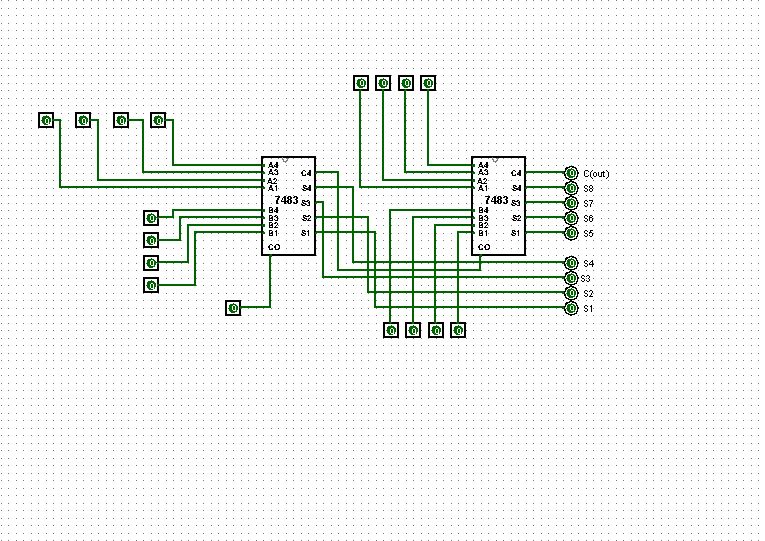
**Discussion:**

In lab 5 and in the lab class I face couple of problem doing the IC circuit, I had done mistake in min terms but I solve that out by following the equation. But this lab was pretty simple. It took some time but finally I found out where the problem was and fix IC circuit and then solved it properly. By the help of our class lab instructor I fix that problem also. That was all human e error problem. After understanding all the problem and practicing that problem, I answered all the questions.

**EXP-1:**

****

**EXP-2:**



**EXP-3:**

