|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Digital Electronics (116U40L303)** | **Semester:** | **III** |
| **Date of Performance:** | **17 / 08 / 2020** | **Batch No:** | **A - 3** |
| **Faculty Name:** | **Kirti Aawalani** | **Roll No:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **/ 25** |

**Experiment No: 3**

**Title: Study of Basic Arithmetic Circuits**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| 1. Implementation of FA using HA and four-bit adder using FA 2. Implementation of controlled adder/ subtractor using 7483 |

|  |
| --- |
| **COs to be achieved:** |
| **CO2:** Design combinational logic circuits using MSI devices |

|  |
| --- |
| **Theory:** |
| In digital systems, adder circuits are used to add bits. Half-adder circuits are used to add two bits. If it is three bits addition, we use a full adder. So, to add n-bit binary numbers, n full adder circuits are used.  **Half Adder**    Sum = A XOR B  Carry = A AND B  What is Half Adder and Full Adder Circuit? - Circuit Diagram ...  A half adder is a fundamental digital logic circuit used to perform binary addition of two single-bit numbers. It can add two input bits, A and B, and produces two outputs: the sum (S) and the carry (C). The carry output represents the carry generated when adding the two bits, and the sum output represents the result of the addition.  The truth table for a half adder is as follows:   | **A** | **B** | **Sum** | **Carry** | | --- | --- | --- | --- | | 0 | 0 | 0 | 0 | | 0 | 1 | 1 | 0 | | 1 | 0 | 1 | 0 | | 1 | 1 | 0 | 1 |   The half adder's logic can be expressed using two simple logic gates: an XOR gate (exclusive OR) and an AND gate. The XOR gate produces the sum output, and the AND gate produces the carry output.  The logic equations for the half adder are as follows:  Sum = A XOR B Carry = A AND B  Let's understand how it works:   1. Sum (S): The sum output (S) is generated by the XOR gate, which outputs a 1 when the inputs are different (A = 0, B = 1, or A = 1, B = 0). It produces a 0 when both inputs are the same (A = 0, B = 0, or A = 1, B = 1). 2. Carry (C): The carry output (C) is generated by the AND gate, which outputs a 1 only when both inputs are 1 (A = 1, B = 1). In this case, the AND gate produces a carry because there is a carry generated when adding two 1s.   The half adder is the simplest form of adder and can only handle two input bits. To add multi-bit binary numbers, a full adder is used, which takes into account both the input bits and the carry from the previous stage of addition.  In summary, a half adder is a basic building block for binary addition, and it is used to add two single-bit binary numbers, producing the sum and carry outputs. Its simple logic is implemented using an XOR gate for the sum and an AND gate for the carry.  **Full Adder**  Full adder gives addition of 3 bits    Sum = A XOR B XOR C  Carry = (A AND B) OR ( B AND C) OR ( C OR A)  Full Adder - an overview | ScienceDirect Topics    A full adder is a digital logic circuit used to perform binary addition of three single-bit numbers: A, B, and a carry-in (Cin). It is an extension of the half adder, which can only add two single-bit numbers. The full adder generates two outputs: the sum (S) and the carry-out (Cout).  The truth table for a full adder is as follows:   | **A** | **B** | **Cin** | **Sum** | **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 |   The full adder's logic is more complex than the half adder because it has to consider the carry-in (Cin) from the previous stage of addition. It uses three logic gates: two XOR gates (exclusive OR) and one AND gate.  The logic equations for the full adder are as follows:  Sum = A XOR B XOR Cin Cout = (A AND B) OR (Cin AND (A XOR B))  Let's understand how it works:   1. Sum (S): The sum output (S) is generated by the first XOR gate, which performs an exclusive OR operation on the inputs A, B, and the carry-in (Cin). It produces a 1 when an odd number of inputs are 1. 2. Carry-out (Cout): The carry-out (Cout) is generated by the AND and OR gates. The AND gate checks if both inputs A and B are 1 and produces a carry when they are. The OR gate combines the carry from the AND gate with the carry-in (Cin) to produce the final carry-out (Cout).   The full adder is used in multi-bit binary addition, where it adds the corresponding bits of two binary numbers and takes into account the carry generated from the previous stage. By chaining multiple full adders together, you can add n-bit binary numbers efficiently.  In summary, a full adder is a more complex adder circuit compared to a half adder. It adds three single-bit binary numbers: A, B, and Cin, producing the sum and carry-out as outputs. It is a crucial building block for multi-bit binary addition in digital systems.  A full adder can be implemented using two half adders and some additional gates  HALF-ADDER-FULL-ADDER-FIG-2  IC 7483 is a 4-bit full adder circuit  Four Bit Adder or Subtractor using 7483 - EEES.IN  This circuit uses 2’s complement subtraction. When the control sub=0, one of the inputs to the XOR gates is 0 and hence the output of each XOR will be the B input. Hence the circuit works as a 4-bit full adder. When sub = 1, the output of XOR will be the complement of input, i.e., 1’s complement. Now the circuit will perform the addition of A+ 1’s complement of B + sub( Now sub= 1). This means the circuit will do 4-bit subtraction in 2’s complement mode  The 7483 is a 4-bit full adder integrated circuit (IC) that can be used to implement both addition and subtraction operations. It is a versatile component commonly used in digital systems for arithmetic computations.   1. Adder using 7483: To implement an adder using the 7483 IC, you simply connect the A and B inputs to the corresponding inputs of the IC and set the control input (sub) to 0. This configuration turns the 7483 IC into a 4-bit full adder, which will produce the correct sum and carry-out results for the addition of the two 4-bit binary numbers A and B. 2. Subtractor using 7483: To perform subtraction using the 7483 IC, you set the control input (sub) to 1. When sub = 1, the output of the XOR gates within the 7483 IC will be the complement (1's complement) of the B input. The circuit will then perform the operation: A + 1's complement of B + 1 (sub = 1).   Subtraction using 1's complement: To subtract two 4-bit binary numbers, A and B, using the 7483 IC, you follow these steps:   1. Obtain the 1's complement of B: This can be achieved by taking the bitwise complement of B, where all 0s are changed to 1s, and all 1s are changed to 0s. 2. Set the control input (sub) of the 7483 IC to 1. 3. Connect A to the A inputs and the 1's complement of B to the B inputs of the 7483 IC. 4. Set the Cin (carry-in) input to 1. 5. The 7483 IC will now perform A + 1's complement of B + 1 (sub = 1), which effectively subtracts B from A.   Note: The result obtained from the 7483 IC will be the 2's complement representation of the subtraction result. To convert it back to the standard binary representation, you can perform a 2's complement operation by taking the 1's complement of the result and adding 1.  In summary, the 7483 IC can be used as a 4-bit full adder for addition by setting the control input (sub) to 0. By setting the control input (sub) to 1 and using the 1's complement of the B input, the IC can be used to perform subtraction in 2's complement form. The 7483 is a versatile and widely used component for arithmetic operations in digital circuits. |

|  |
| --- |
| **Stepwise-Procedure:** |
| 1. Implement half adder, full adder, and full adder using two half adders. 2. Verify the Truth Table of each of them. 3. Implement full subtractor using half subtractor. 4. Implement a controlled adder/subtractor circuit using ant 4-bit full adder IC. |

|  |
| --- |
| **Truth Table:** |
| **Half Adder**   |  |  |  |  | | --- | --- | --- | --- | | **A** | **B** | **Sum** | **Carry** | | **0** | **0** | **0** | **0** | | **0** | **1** | **1** | **0** | | **1** | **0** | **1** | **0** | | **1** | **1** | **0** | **1** |   **Full Adder**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **A** | **B** | **Cin** | **Sum** | **Carry** | | **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** |   **Make truth Table for 4 Bit full adder and subtractor and verify it using IC 7483.**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ctrl** | **A3** | **A2** | **A1** | **A0** | **B3** | **B2** | **B1** | **B0** | **S3** | **S2** | **S1** | **S0** | **Cout** | | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **0** | | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **0** | | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | **1** | **0** | **0** | | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | | **1** | **1** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **1** | | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **1** |   **1-3 = -2**  **4+3 = 7** |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions: (Must be handwritten and then upload screenshot)** |
| 1. Explain the working principle of a half adder with the help of a circuit diagram. 2. How does a full adder differ from a half adder in terms of the number of inputs and outputs? 3. Implement a 3-bit binary addition using full adders and draw the circuit diagram. 4. Implement a 3-bit binary subtraction using full adders and draw the circuit diagram. 5. How can a full subtractor be implemented using a half subtractor? 6. Discuss the differences between 1's complement and 2's complement subtraction methods. 7. How does the controlled adder/subtractor circuit using the 4-bit full adder IC work? Provide a detailed explanation and circuit diagram. 8. Compare and contrast the advantages and disadvantages of using half adders and full adders in digital systems. 9. How can you extend the circuit to perform addition and subtraction for n-bit binary numbers? |

|  |
| --- |
| **Conclusion:** |
| In this exploration, we learned about binary arithmetic using half adders and full adders. We compared 1's and 2's complement subtraction methods, highlighting their similarities and differences. This study emphasizes their significance in digital systems for accurate arithmetic computations. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |