| **Course Name:** | **Digital Design Laboratory** | **Semester:** | **III** |
| --- | --- | --- | --- |
| **Date of Performance:** | **\_\_24\_ / \_\_07\_ / \_\_2023\_\_\_\_** | **Batch No:** | **B-1** |
| **Faculty Name:** |  | **Roll No:** | **16010122104** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_/25** |

**Experiment No: 2**

**Title: Binary Adders and Subtractors**

| **Aim and Objective of the Experiment:** |
| --- |
| To implement half and full adder–subtractor using gates and IC 7483 |

| **COs to be achieved:** |
| --- |
| **CO2**: Use different minimization technique and solve combinational circuits. |

| **Tools used:** |
| --- |
| Trainer kits |

| **Theory:** |
| --- |
| **Adder:** The addition of two binary digits is the most basic operation performed by the digital computer. There are two types of adder:   * Half adder * Full adder   **Half Adder:** Half adder is a combinational logic circuit with two inputs and two outputs. It is the basic building block for the addition of two single-bit numbers.  **Full adder:** A half adder has a provision not to add a carry coming from the lower order bits when multi-bit addition is performed. for this purpose, a third input terminal is added and this circuit is to add A, B, and C where A and B are the nth order bits of the number A and B respectively and C is the carry generated from the addition of (n-1) order bits. This circuit is referred to as full adder.  **Subtractor:** Subtraction of two binary digits is one of the most basic operations performed by digital computer .there are two types of subtractors:   * Half subtractor * Full subtractor   **Half subtractor:** Logic circuit for the subtraction of B from A where A,B are 1 bit numbers is referred to as half subtract or .the subtract or process has two input and difference and borrow are the two outputs.  **Full subtractor:** As in the case of the addition using logic gates, a full subtractor is made by combining two half-sub tractors and an additional OR-gate. A full subtractor has the borrow in capability (denoted as BORIN) and so allows cascading which results in the possibility of multi-bit subtraction.  **IC 7483**  For subtraction of one binary number from another, we do so by adding 2’s complement of the former to the latter number using a full adder circuit.  IC 7483 is a 16 pin, 4-bit full adder. This IC has a provision to add the carry output to transfer and end around carry output using Co and C4 respectively.  **2’s complement:** 2’s complement of any binary no. can be obtained by adding 1 in 1’scomplement of that no.  e.g. 2’s complement of +(10)10 =1010is   | 1C of 1010 |  | | 0101 | | --- | --- | --- | --- | |  |  | + | 1 | | -(10)10 |  | | 0110 |   In 2’s complement subtraction using IC 7483, we are representing negative number in 2’s complement form and then adding it with 1st number.  **Implementation Details:**  **Half Adder Block Diagram**  hadder.png  **Half Adder Circuit**  images.png  **Truth Table for Half Adder**   | **Inputs** | | **Outputs** | | | --- | --- | --- | --- | | **A** | **B** | **S** | **C** | | 0 | 0 | 0 | 0 | | 0 | 1 | 1 | 0 | | 1 | 0 | 1 | 0 | | 1 | 1 | 0 | 1 |   **From the truth table (with steps):**  S=AB’+A’B  C=A.B   | **Inputs** | |  | | **Outputs** | | | --- | --- | --- | --- | --- | --- | | **A** | **B** | **A** ⊕ **B** | **A . B** | **S** | **C** | | **0** | **0** | **0** | **0** | **0** | **0** | | **0** | **1** | **1** | **0** | **1** | **0** | | **1** | **0** | **1** | **0** | **1** | **0** | | **1** | **1** | **0** | **1** | **0** | **1** |   **Full Adder Block Diagram**  download (1).png  **Full Adder Circuit**  download.png  **Truth Table for Full Adder**   | **Sr. No** | **A** | **B** | **Cin** | **S** | **Cout** | | --- | --- | --- | --- | --- | --- | | 1. | 0 | 0 | 0 | 0 | 0 | | 2. | 0 | 0 | 1 | 1 | 0 | | 3. | 0 | 1 | 0 | 1 | 0 | | 4. | 0 | 1 | 1 | 0 | 1 | | 5. | 1 | 0 | 0 | 1 | 0 | | 6. | 1 | 0 | 1 | 0 | 1 | | 7. | 1 | 1 | 0 | 0 | 1 | | 8. | 1 | 1 | 1 | 1 | 1 |   **From the truth table (with steps):**  S = A’B’Cin + A’BCin’ + AB’Cin’ + ABCin  Cout = ACin + AB+ BCin   | Inputs | | | Operations | | | | | Ouputs | | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | A | B | Cin | A ⊕ B  = D | D ⊕ Cin  = E | D . Cin  = F | A . B = G | F + G = H | S | C | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |   **Half Subtractor Block Diagram**  Capture.PNG  **Half Subtractor Circuit**    **Truth Table for Half Subtractor**   | **A** | **B** | **DIFFERENCE(D)** | **BORROW(Bo)** | | --- | --- | --- | --- | | 0 | 0 | 0 | 0 | | 1 | 0 | 1 | 0 | | 0 | 1 | 1 | 1 | | 1 | 1 | 0 | 0 |  |  |  |  |  |  | | --- | --- | --- | --- | --- |   **From the truth table (with steps) :**  Difference (D) = A’B + AB’  Borrow(B) = A’B   | Inputs | |  |  |  | Outputs | | | --- | --- | --- | --- | --- | --- | --- | | A | B | ~A = C | A ⊕ B = E | B . C = F | D | Borrow | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 0 | 0 | 1 | 0 | 1 | 0 | | 1 | 1 | 0 | 0 | 0 | 0 | 0 |   **Full Subtractor Block Diagram**  Capture.PNG  **Full Subtractor Circuit**  Capture.PNG  **Truth Table for Full subtractor**   | **A** | **B** | **BIN** | **D** | **BOROUT** | | --- | --- | --- | --- | --- | | 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 |     **From the truth table (with steps):**  Difference = A’B’Bin + A’BBin’ + AB’Bin’ + ABBin  Borrow out= A’B + A’Bin + BBin   | **Input** | | |  |  |  |  |  |  |  | **Output** | | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **A** | **B** | **Bin** | **A** ⊕ B =  C | **~A = E** | **E.B = F** | **Bin**⊕ C  = G | **~C = F** | **F.Bin = H** | **F + H = I** | **D** | **Borrout** | | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | | **0** | **1** | **0** | **1** | **1** | **1** | **1** | **0** | **0** | **0** | **1** | **1** | | **0** | **1** | **1** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | | **1** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | | **1** | **1** | **1** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** |   **Example:**   | 1) 710 -210 = 510 | |  | | --- | --- | --- | | 7 |  | 0111 | | 2 |  | 0010 | | 1’C of 2 | | 1101 | |  |  | + 1 | | 2’C of 2 | | 1110 |   0111 + 1110 1 0101  **Pin Diagram IC7483**  IC7483 (1).gif  **Adder**  71.png  **Subtractor**  comb35.gif |

| **Implementation Details** |
| --- |
| **Procedure:**   1. Locate the IC 7483 and 4-not gates block on trainer kit. 2. Connect 1st input no. to A4-A1 input slot and 2nd (negative) no. to B4-B1 through 4-not gates (1C of 2nd no.) 3. Connect high input to Co so that it will get added with 1C of 2nd no. to get 2C. 4. Connect 4-bit output to the output indicators. 5. Switch ON the power supply and monitor the output for various input combinations. |
| **Post Lab Subjective/Objective type Questions:** |
| 1. Design a full adder using two half adders.      1. Perform the following Binary subtraction with the help of appropriate ICs:    1. 6-4 |

| **Conclusion:** |
| --- |
| Circuits of binary adder and sub tractors were studied on the IC kit using . .Connectors and tested using sample values. |

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