**EXPERIMENT NO. 02**

|  |  |
| --- | --- |
| **DATE OF PERFORMANCE:** | **GRADE:** |
| **DATE OF ASSESSMENT:** | **SIGNATURE OF LECTURER/ TTA:** |

**AIM:**  **Implementation of HALF ADDER, FULL ADDER using basic logic gates.**

**THEORY:**

**An adder is a digital circuit that performs addition of numbers. The half adder adds two binary digits called as augend and addend and produces two outputs as sum and carry; XOR is applied to both inputs to produce sum and AND gate is applied to both inputs to produce carry. The full adder adds 3 one bit numbers, where two can be referred to as operands and one can be referred to as bit carried in. And produces 2-bit output, and these can be referred to as output carry and sum. Half Adder By using half adder, you can design simple addition with the help of logic gates.**

**Half Adder**

**0+0 = 0**

**0+1 = 1**

**1+0 = 1**

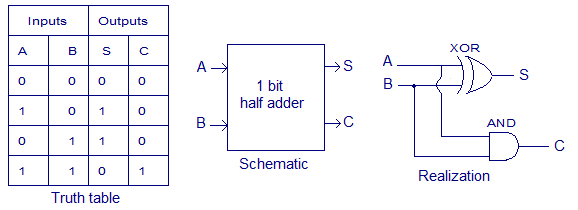
**1+1 = 10**

**adder can be easily implemented with the help of the XOR Gate for the output ‘SUM’ and an AND Gate for the ‘Carry’. When we need to add, two 8-bit bytes together, we can be done with the help of a full-adder logic. The half-adder is useful when you want to add one binary digit quantities. A way to develop a two-binary digit adders would be to make a truth table and reduce it. When you want to make a three binary digit adder, do it again. When you decide to make a four digit adder, do it again. The circuits would be fast, but development time is slow.**

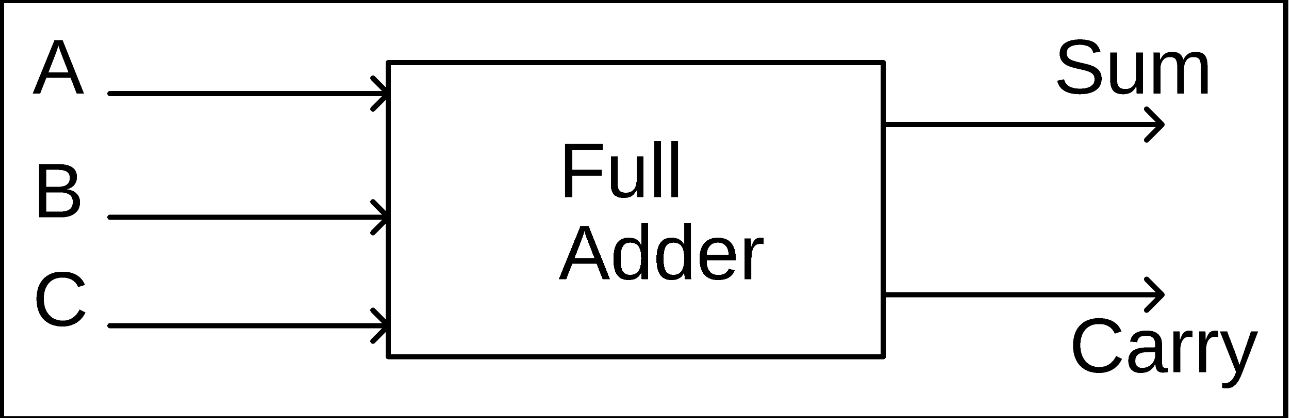
**A+B**

**A.B**

**( A ⨁ B)**



**FULL ADDER**

****

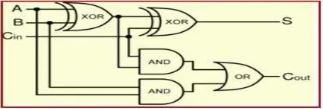
**The output carry is designated as C-OUT and the normal output is designated as S.**

**FULL ADDER Truth Table:**

****

**With the truth-table, the full adder logic can be implemented. You can see that the output S is an XOR between the input A and the half-adder, SUM output with B and C-IN inputs. We take C-OUT will only be true if any of the two inputs out of the three are HIGH.**

**So, we can implement a full adder circuit with the help of two half adder circuits. At first, half adder will be used to add A and B to produce a partial Sum and a second half adder logic can be used to add C-IN to the Sum produced by the first half adder to get the final S output.**

****

**( A ⨁ B)**

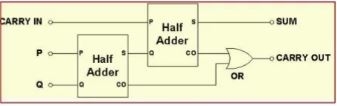
**( A ⨁ B) ⨁ Cin ( A ⨁ B) ⨁ Cin**

**A\*B**

**( A ⨁ B) \* Cin**

**[( A ⨁ B) \* Cin + (A\*B)]**

**The implementation of larger logic diagrams is possible with the above full adder logic a simpler symbol is mostly used to represent the operation. Given below is a simpler schematic representation of a one-bit full adder.**

****

**Full Adder Design Using Half Adders With this type of symbol, we can add two bits together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude.**

**In a computer, for a multi-bit operation, each bit must be represented by a full adder and must be added simultaneously. Thus, to add two 8-bit numbers, you will need 8 full adders which can be formed by cascading two of the 4-bit blocks.**

**Full-Adder is of two Half-Adders, the Full-Adder is the actual block that we use to create the arithmetic circuits.**