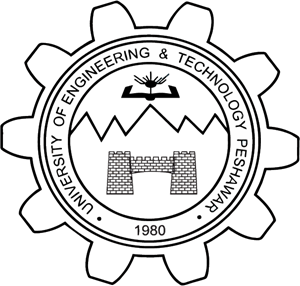
Lab report 05

**Adder and Subtractor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Group No 05**



**202L-Digital Logic Design Lab**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Engr. Rehmat Ullah**

Submitted by

**Muhammad Zaid (1991)**

**Muhammad Saad (1997)**

**Muhsin Shah (2018)**

**DCSE, Batch 23, Section “B”**

**Adder and Subtractor**

**Objective:**

* To design half adder, half subtractor, full adder and full subtractor and verify their truth table

**Apparatus:**

1. **Logic gates**

* AND gate (IC 7408)
* OR gate (IC 7432)
* XOR gate (IC 7486)
* NOT gate (IC 7404)

**2.** **LEDs**

**3. DIP Switch**

**4. 520 Ω / 1k Ω resistors**

**Theory:**

An adder or subtractor adds or subtracts two binary inputs of one bit. Half adder and half subtractor can only add two one bit binary numbers and in output it has also two numbers one is the sum and other is the carry or borrow. But as it can only operate on two inputs and there room for carry or borrow from the previous stage makes it limiting and that is why it is called half adder and subtractor. A full adder on the other hand can also take another input of carry or borrowfrom previous stage and therefore can add any two numbers. We can also chain multiple full adders which is called ripple carry adder.

**Circuit Diagrams and truth tables**

**Half Adder:**

It is a one bit adder which add two one bit numbers and then gives the result as sum and carry.

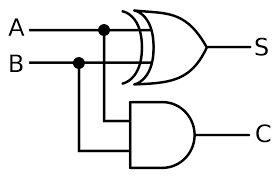
****

Fig. 01 Half adder circuit diagram

**S(A, B)= A**⊕**B**

**C(A, B) = A.B**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | S(A, B) | C(A, B) |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Half Adder

**Half Subtractor:**

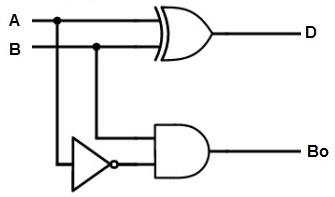
 It also adds two one bit numbers and gives the result as difference and borrow. The connections are according to circuit diagrams.

Fig 02. Half subtractor

**D(A, B)= A**⊕**B**

**Bo(A, B) = A’ . B**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | D(A, B) | Bo(A, B) |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Half Subtractor

**Full Adder:**

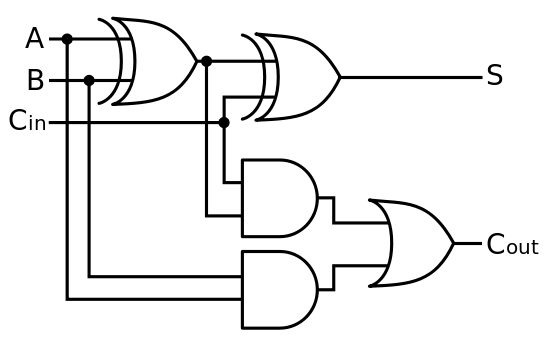
Full adder takes two inputs and one comes Cin comes from the previous stage.

Fig 03. Full Adder

**S(A, B, Cin) = A⊕B⊕Cin**

**Cout = A⊕B.Cin + A.B**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | S(A, B, Cin) | Cout(A, B. Cin) |
| 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 |

Full Adder

**Full Subtractor:**

Full subtractor like full adder has two input and third input called Borrow (Bin ). The result is a difference and borrow.

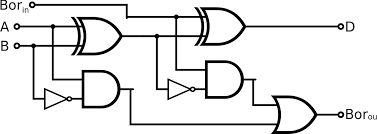
****

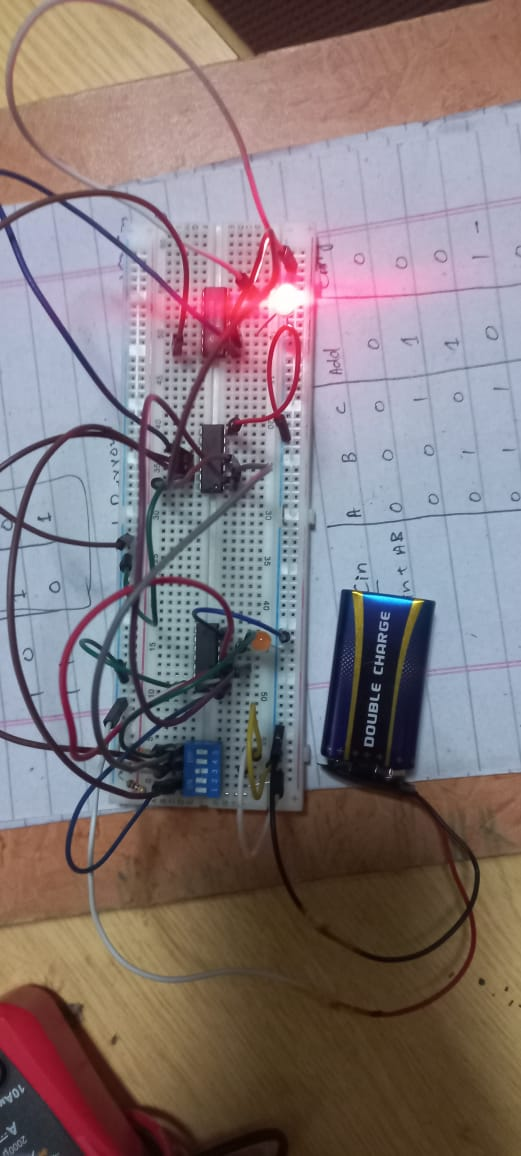
Fig 04. Full Subtractor

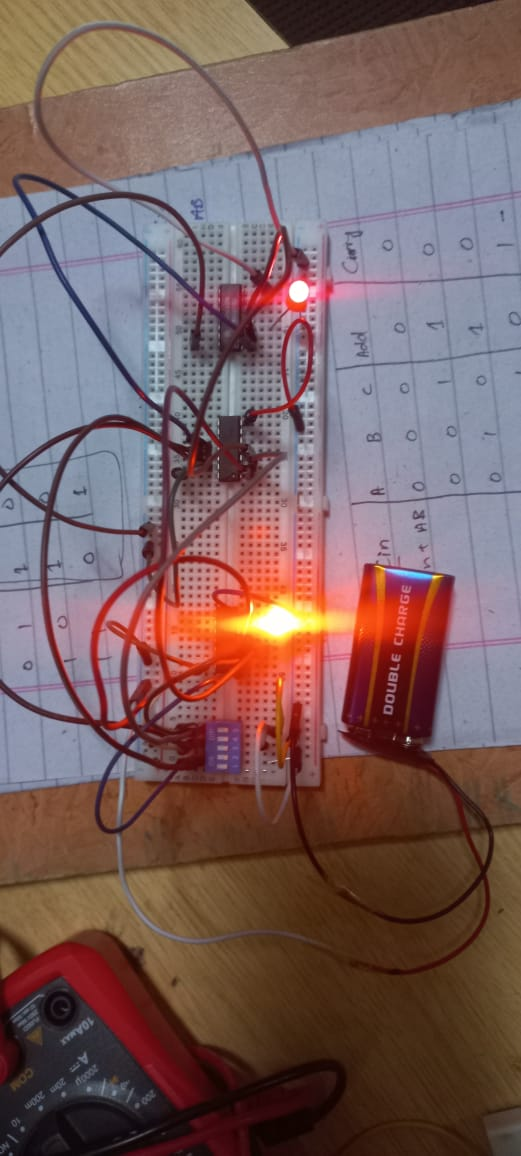
**S(A, B, Cin) = A⊕B⊕Bin**

**Bout = (A⊕B)’ .Bin + A’ . B**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Bin | D(A, B, Bin) | Bout(A, B. Bin) |
| 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 |

Full Subtractor

**Experiment Pictures**

****

