**Title:** Design of adder, subtractor and comparator circuits.

## Abstract:

The purpose of this experiment is to learn the design and behaviour of adder, subtractor and comparator logic circuits. Adders and subtractors are the most basic and most important part of digital electronics.

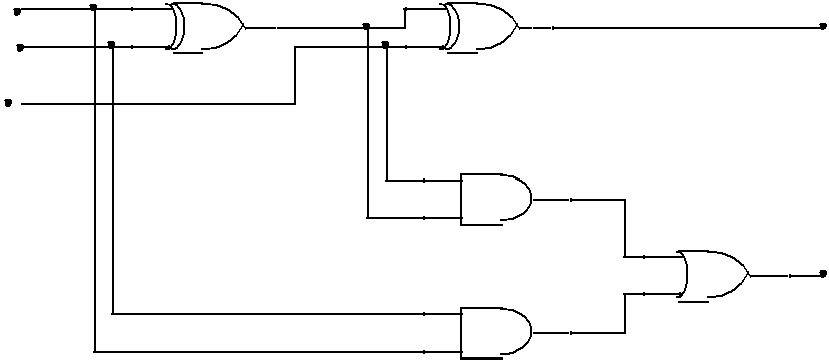
## Part I (Adder and Subtractor):

Adders and subtractors are digital circuits which are capable of adding and subtracting binary digits. They are the most important part in the design of Arithmetic Logic Unit (ALU). In this experiment different types of adders and subtractors will be designed and their behavior will be observed.

## Theory and Methodology:

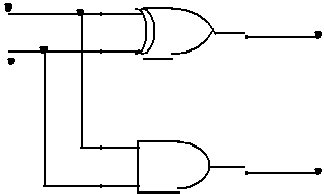
An adder or summer is a combinational circuit that adds binary numbers. There are mainly two kinds of adders, half adder and full adder. The half adder can add only two single bits of binary digit and outputs the sum of the bits and a carry which is the overflow of the sum. A full adder can add two single bit digits and one carry bit which is the overflow of the sum of the previous stage of addition and outputs the sum and the carry.

S



A B

in

A

S C

B

C

C

Half Adder

Full Adder

**Fig.1.1**: Schematics of Half Adder and Full Adder The Boolean expression for half and full adder is given below –

S = A⊕ B

Half adder – C

= AB

Full adder –

S = A ⊕ B ⊕ Cin

Cout = Cin (A ⊕ B) + AB

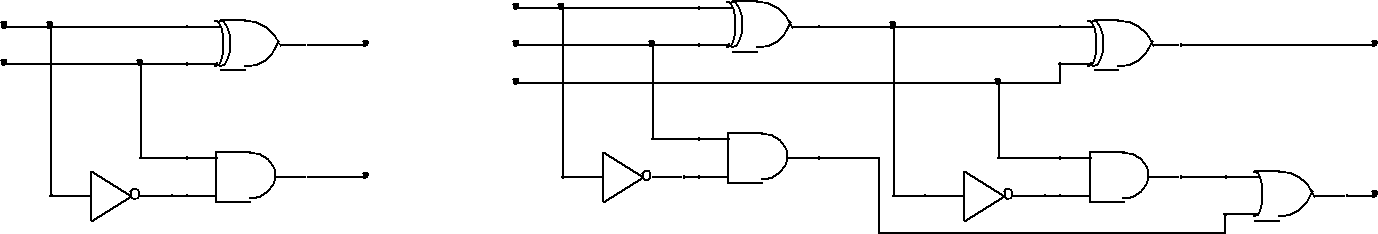
Truth table for half adder –

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Truth table for full adder –

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | S | C |
| 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 |

A subtractor is also a combinational circuit that calculates the difference of two binary digits. This is done by taking the two’s complement of the subtrahend and then adding it with the minuend. So the subtractor circuit can be designed with the help of adder circuits. Like adders, there are two types of subtractor circuits, half subtractor and full subtractor.

A half subtractor performs a subtraction between two single bits and produces their difference and another output called borrow. A full subtractor performs a subtraction two single bits, taking into account a borrow bit. It outputs the difference of the subtraction and a borrow bit.

|  |  |  |  |
| --- | --- | --- | --- |
| X |  | X |  |
| D | Y |  |
| Y | D |
|  | Bin |  |
|  | B |  | B |
| Half Subtractor | | | |
| Full Subtractor | | | |

**Fig.1.2:** Schematics of Half Subtractor and Full Subtractor

The Boolean expressions for half and full subtractor are given below – Half subtractor –

D = X ⊕ Y B = X’Y

Full subtractor –

D = X ⊕ Y ⊕ Bin

Bout = X’Y + X’Bin + YBin

Truth table for half subtractor –

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | D | B |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

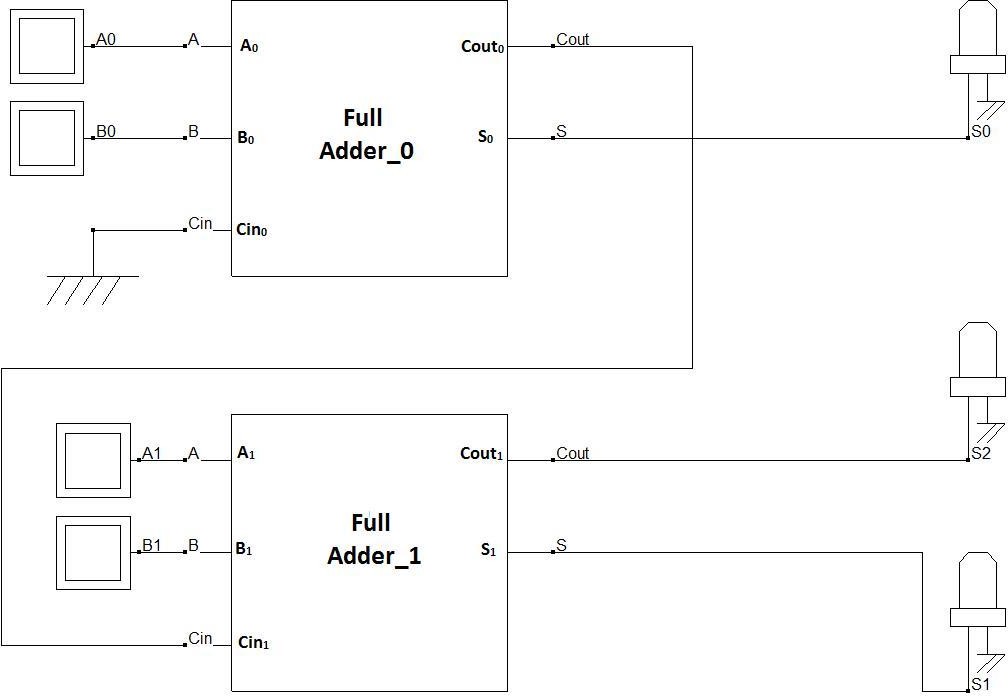
Truth table for full subtractor –

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Bin | D | B |
| 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 |

## Using Full Adder blocks for addition of n- bit systems:

Full adder blocks can be connected for summation of n-bit systems. To design a 2 bit full adder, two 1 bit full adders are connected in parallel connection as shown in the figure below. The same process can be used for designing n-bit Full Adder for addition of words having a length of n-bits.

|  |
| --- |
|  |

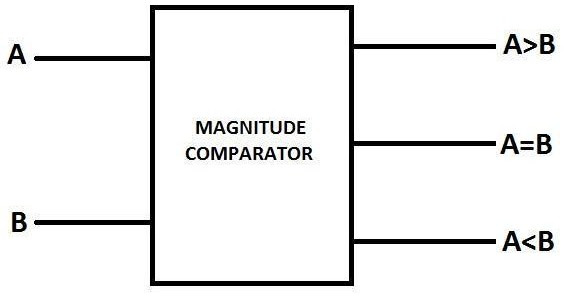


**Fig. 1.3:** 2-bit Full adder design using 1 bit full adder blocks

Here, the LSB of both word A and B (A0 and B0) are connected in the first stage full adder block and Cin of this block (Cin0) is connected to ground (as there is no carry in available at the initial stage). The MSB of both word A and B (A1 and B1) are connected in the first stage full adder block and Cin of this block (Cin1) is connected to the previous stage Cout (Cout0). Summation output for the LSB is available from the first stage Sum (S0). The next stage block outputs Sum (S1) and Carry out (Cout1) provide the MSBs for the next stage output (S1 and and S2).

**Part II (Comparator):** A magnitude comparator is a device that takes in two sets of inputs in its input and compares them to provide an output, if they are equal, greater than or less than the other. In this experiment 1-bit comparator will be designed at first and using the 1-bit comparator block, 2-bit comparator will be designed.

**Theory and Methodology:** Magnitude Comparators are combinational logic circuits that take 2 sets of data as its inputs and tests whether the value represented by one binary word is greater than, less than, or equal to the value represented by another binary word.



**Fig.2.1**: Block Diagram of 1 Bit Magnitude Comparator

Depending on the input combination for a 1-bit magnitude comparator, following behavior table can be developed using the logic expressions.

# A=B if, A=B=0 or A=B=1;

**A>B if A=1 and B=0; A<B if A=0 and B=1;**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** |  | **B** |  |  | **A=B** |  |  | **A>B** |  |  | **A<B** |  |
| **0** | | **0** | | **1** | | | **0** | | | **0** | | |
| **0** | | **1** | | **0** | | | **0** | | | **1** | | |
| **1** | | **0** | | **0** | | | **1** | | | **0** | | |
| **1** | | **1** | | **1** | | | **0** | | | **0** | | |

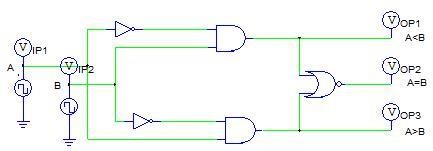
The SOP expressions for the output lines can be written as

# (A=B)= A’B’+AB;

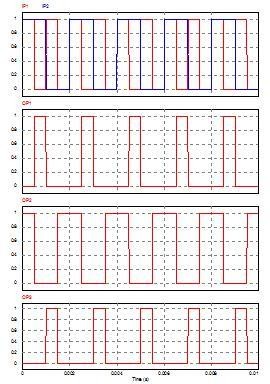
**(A<B)= AB’;**

**(A>B)=A’B;**

|  |  |
| --- | --- |
|  |  |

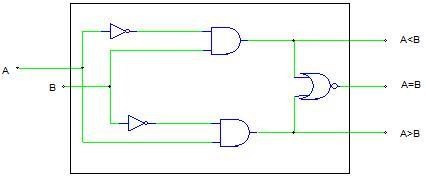


**Fig.2.2:** 1-Bit Comparator



**Fig.2.3**: Timing Diagram for 1-Bit Comparator **2 Bit Comparator design using 1 bit block:**

Using 1-bit blocks, n-bit Magnitude comparator can be designed.



**Fig.2.4:** 1-Bit Comparator Block

|  |
| --- |
|  |

**Designing a 2-bit comparator using 1-bit blocks:**

Let us consider 2 words,

**Word A -> A1A0 Word B-> B1B0**

For comparing, the following process is used as writing the logic equations.

**For A=B**,

If (A1=B1) & (A0=B0), then (A=B);

**For A>B**,

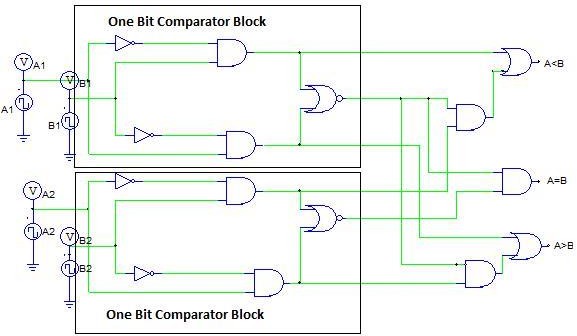
If (A1>B1) then (A>B) or

if (A1=B1) & (A0>B0), then (A>B);

**For A<B**,

If (A1<B1) then (A<B) or

if (A1=B1) & (A0<B0), then (A<B)



**Fig.2.5:** 2-Bit Comparator using 1\_bit Comparator Block

For designing a 2-bit comparator using 1-bit comparator block, 2 1-bit comparator block, 3 AND gate and 2 OR gate is needed as shown in Fig.4.

## Pre-Lab Homework:

Read about the characteristics of adder and subtractor circuits from any book or websites and use PSIM to generate the output of the circuits provided in this lab sheet. Save the simulation results and bring it to the lab.

## Apparatus:

1. Digital trainer board 2. IC 7408:2 pcs

3. IC 7404:2 pcs

4. IC 7486:2 pcs

5. IC 7431:2 pcs

6. IC 7483:1 pcs

7. Connecting wires

## Precautions:

1. Make sure that all the LEDs and the toggle switches of the trainer board are working properly.
2. Do not short any connections. Short connection can produce heat (due to high current flow) which is harmful for the components.

## Experimental Procedure:

* 1. Determine the output and the truth tables of the logic circuits for full adder and half subtractor given in the theory and methodology part.
  2. Determine which gates and how many of them are required, check and detect all the IC numbers.
  3. Carefully place the ICs on the Trainer Board and bias them by connecting them to the +5 volt DC supply and ground.
  4. Connect those using wires according to the logic diagram; connect the outputs to the LEDs.
  5. Check and note down the outputs by giving different inputs according to the derived truth table.
  6. Design a 4 bit full adder using IC 7483 and verify its operation.

**Simulation and Measurement:**

**Half Adder:**

|  |  |
| --- | --- |
| **Simulations** | **Simulations** |
| A=0, B=0 | A=0, B=1 |
| A=1, B=0 | A=1, B=1 |
|  |  |

**Full Adder:**

|  |  |
| --- | --- |
| **Simulations** | **Hardware** |
| A=0, B=0, C=0 | A=0, B=0, C=0 |
| A=0, B=0, C=1 | A=0, B=0, C=1 |
| A=0, B=1, C=0 | A=0, B=1, C=0 |
| A=0, B=1, C=1 | A=0, B=1, C=1 |
| A=1, B=0, C=0 | A=1, B=0, C=0 |
| A=1, B=0, C=1 | A=1, B=0, C=1 |
| A=1, B=1, C=0 | A=1, B=1, C=0 |
| A=1, B=1, C=1 | A=1, B=1, C=1 |
|  |  |

**Half Subtractor:**

|  |  |
| --- | --- |
| **Simulations** | **Simulations** |
| A=0, B=0 | A=1, B=0 |
| A=0, B=1 | A=1, B=1 |
|  |  |

**Full Subtractor:**

|  |  |
| --- | --- |
| **Simulations** | **Hardware** |
| A=0, B=0, C=0 | A=0, B=0, C=0 |
| A=0, B=0, C=1 | A=0, B=0, C=1 |
| A=0, B=1, C=0 | A=0, B=1, C=0 |
| A=0, B=1, C=1 | A=0, B=1, C=1 |
| A=1, B=0, C=0 | A=1, B=0, C=0 |
| A=1, B=0, C=1 | A=1, B=0, C=1 |
| A=1, B=1, C=0 | A=1, B=1, C=0 |
| A=1, B=1, C=1 | A=1, B=1, C=1 |
|  |  |

**1-bit Comparator:**

|  |  |
| --- | --- |
| **Simulations** | **Simulations** |
| A=0, B=0 | A=0, B=1 |
| A=1, B=0 | A=1, B=1 |
|  |  |

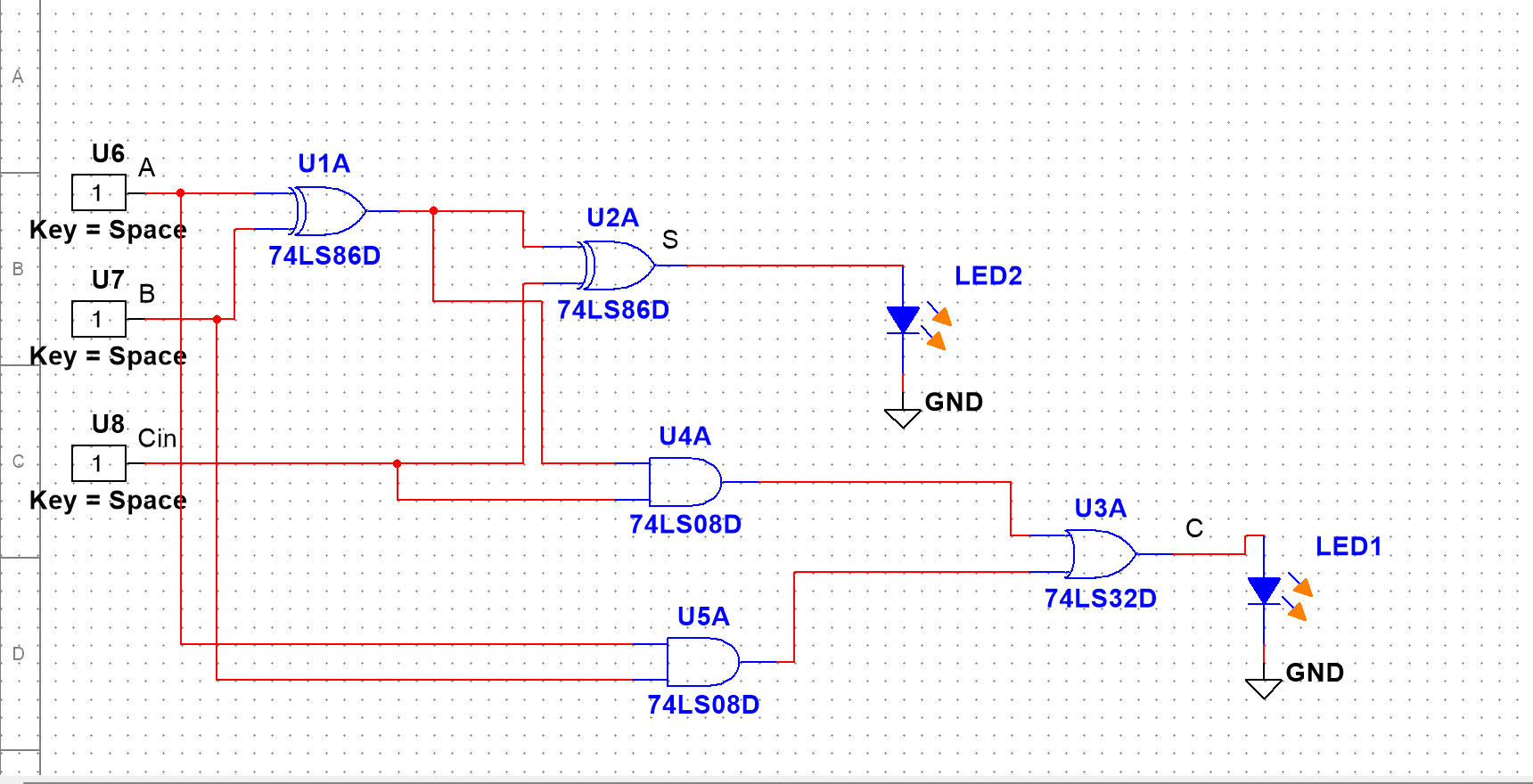
**2-bit Comparator:**

|  |  |
| --- | --- |
| **Simulations** | **Hardware** |
| A=0, B=0, C=0, D=0 | A=0, B=0, C=0, D=0 |
| A=0, B=0, C=0, D=1 | A=0, B=0, C=0, D=1 |
| A=0, B=0, C=1, D=0 | A=0, B=0, C=1, D=0 |
| A=0, B=0, C=1, D=1 | A=0, B=0, C=1, D=1 |
| A=0, B=1, C=0, D=0 | A=0, B=1, C=0, D=0 |
| A=0, B=1, C=0, D=1 | A=0, B=1, C=0, D=1 |
| A=0, B=1, C=1, D=0 | A=0, B=1, C=1, D=0 |
| A=0, B=1, C=1, D=1 | A=0, B=1, C=1, D=1 |
| A=1, B=0, C=0, D=0 | A=1, B=0, C=0, D=0 |
| A=1, B=0, C=0, D=1 | A=1, B=0, C=0, D=1 |
| A=1, B=0, C=1, D=0 | A=1, B=0, C=1, D=0 |
| A=1, B=0, C=1, D=1 | A=1, B=0, C=1, D=1 |
| A=1, B=1, C=0, D=0 | A=1, B=1, C=0, D=0 |
| A=1, B=1, C=0, D=1 | A=1, B=1, C=0, D=1 |

|  |  |
| --- | --- |
| A=1, B=1, C=1, D=0 | A=1, B=1, C=1, D=0 |
| A=1, B=1, C=1, D=1 | A=1, B=1, C=1, D=1 |

## Report:

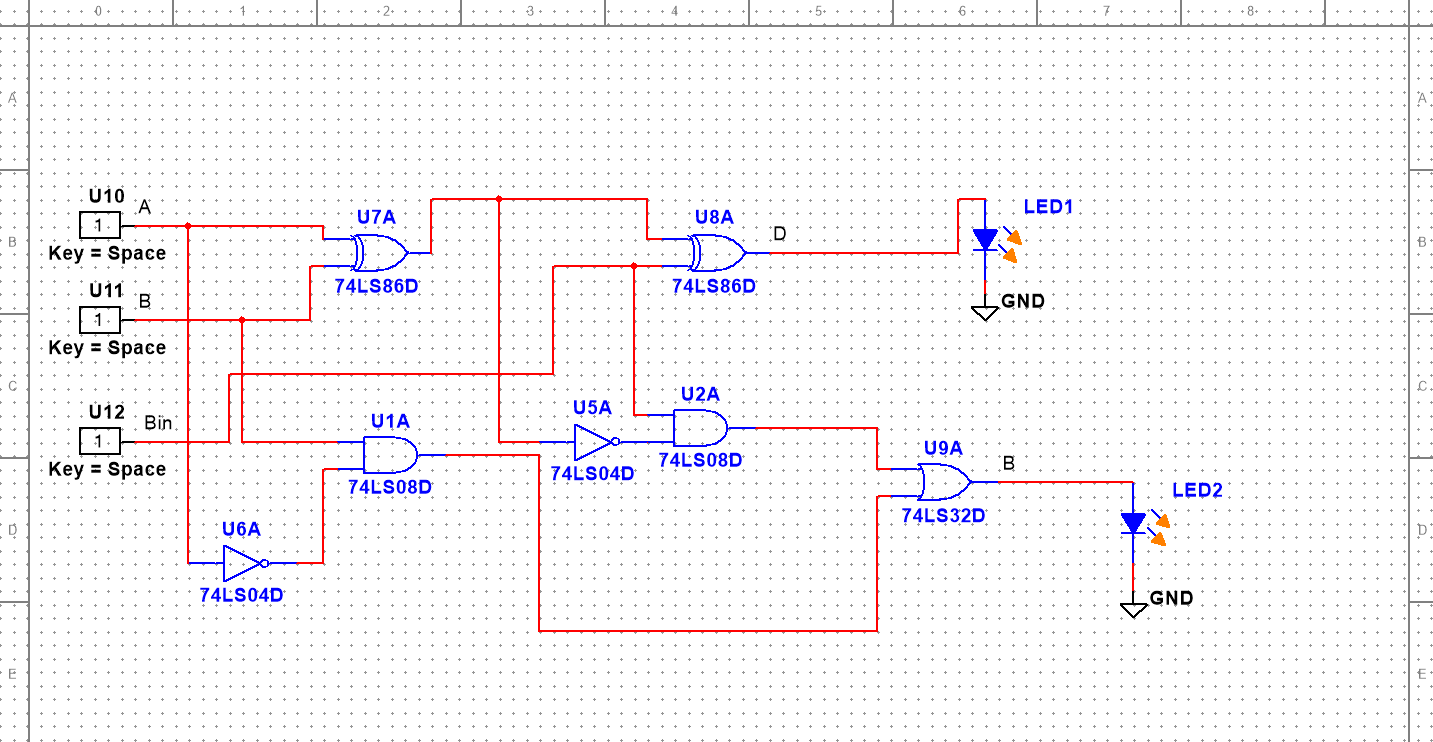
1. Design a full adder circuit for performing 3 bit binary addition.



Truth table for 3-bit full adder –

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | S | C |
| 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 |

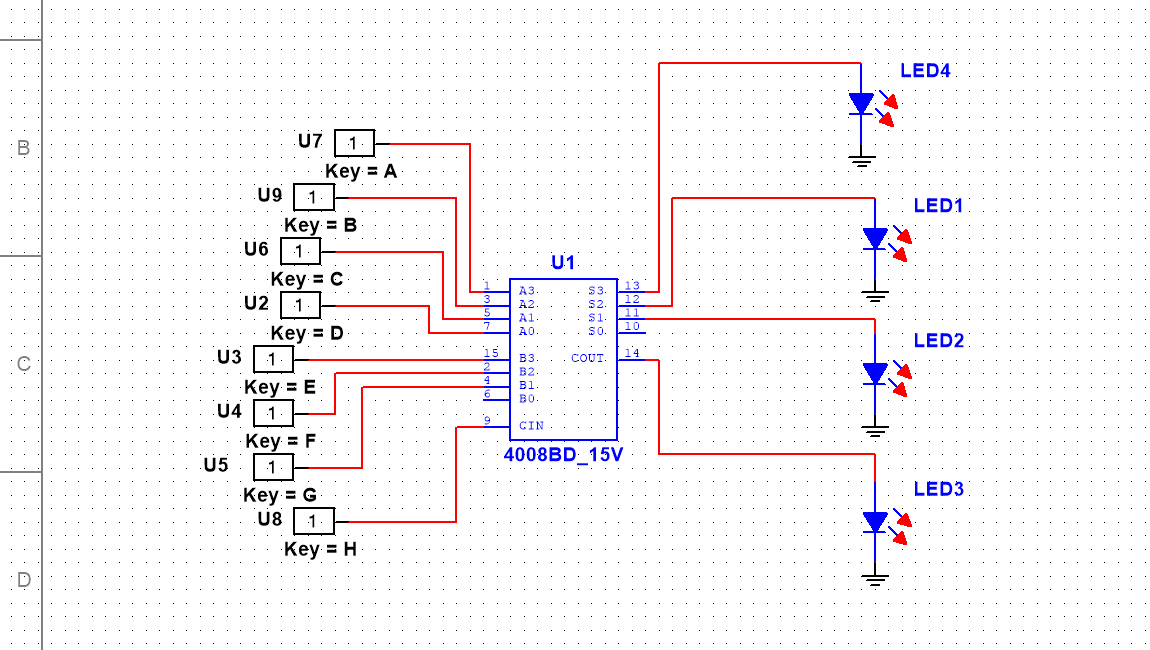
1. Design a full subtractor circuit for performing 3 bit binary subtractor.



Truth table for 3-bit full subtractor –

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Bin | D | B |
| 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 |

1. Design an 8 bit full adder using 4 bit full adder IC 4008 from PSIM.



## Discussion and Conclusion:

In this experiment, circuits were designed, and simulated, verified with truth tables, focusing on Adder, Subtractor, and Comparator functions. Full and Half adder circuits were explored, with Full adders physically implemented and simulated. Similarly, Full and half subtractor circuits were explored and Full subtractor circuits were implemented and simulated. The experiment also included one-bit and two-bit Comparator circuits, all of which were verified using truth tables. Overall, the experiment offered a comprehensive understanding of adder, subtractor, and comparator devices functionalities.

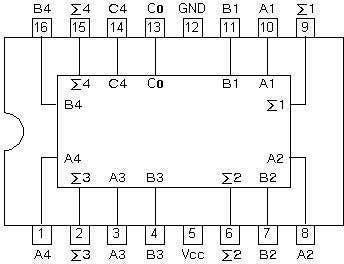
## Conclusion:

We've successfully completed the experiment and got all the expected results. Designing half and full adders, half and full subtractors, and 1-bit and 2-bit magnitude comparators is an important topic in digital logic design. We must be careful when entering inputs to these circuits.

## Reference:

<http://www.circuitstoday.com/half-adder-and-full-adder>

## Appendix:

**Pin configuration of IC 74LS83**

**Fig. :** 4-bit Full Adder IC pin configuration

## Pin configuration for IC-74LS85