A picture containing calendar

Description automatically generated

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DIGITAL LOGIC AND CIRCUITS LAB**

**Summer 2022-2023**

Section: F

Group Number: 02

**EXPERIMENT NO. 3**

**NAME OF THE EXPERIMENT**

*Design of adder, subtractor and comparator circuits.*

**Supervised By**

SHAHRIYAR MASUD RIZVI

Faculty of Engineering, AIUB

**Submitted By**:

|  |  |
| --- | --- |
| **Name of the Student** | **ID Number** |
| 1. NOKIBUL ARFIN SIAM | 21-44793-1 |
| 1. AHNAF AHMED | 20-42173-1 |
| 1. SAIFUL ISLAM | 20-42585-1 |
| 1. MAHADI HASAN | 19-39711-1 |
| 1. SHEIKH FAHIM FUAD | 21-44721-1 |

**Abstract:**

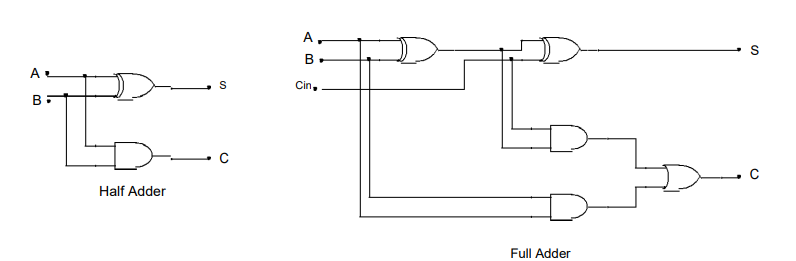
The purpose of this experiment is to learn the design and behavior of adder, subtractor and comparator logic circuits. Adders and subtractors are the most basic and most important part of digital electronics.

**Part I (Adder):**

Adders and are digital circuits which are capable of adding binary digits. They are the most important part in the design of Arithmetic Logic Unit (ALU). In this experiment different types of adders 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.



**Fig.1.1:** Schematics of Half Adder and Full Adder

Half adder–

S=A⊕B

Cout = AB

Full adder –

S = A ⊕ B ⊕ Cin

Cout = Cin (A ⊕ B) + AB

Truth table for half adder –

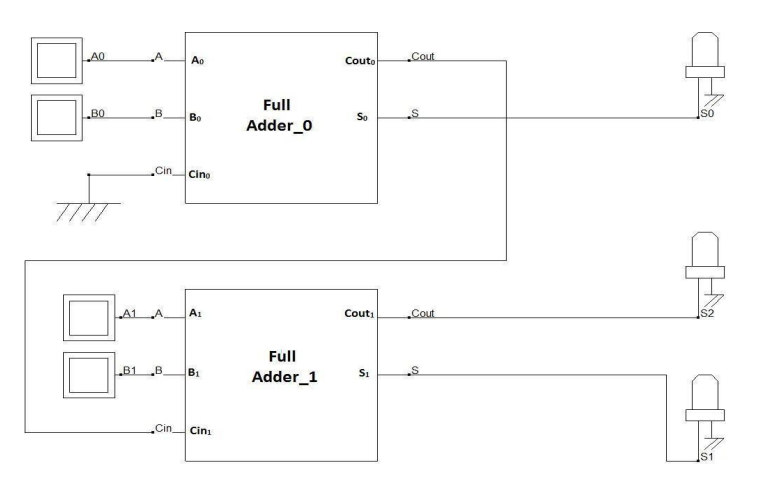
|  |  |  |  |
| --- | --- | --- | --- |
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

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 |

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

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

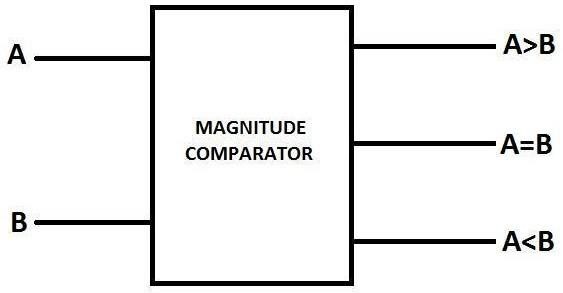


**Fig. 1.3:** 2-bit Full adder design using 1-bit full adder blocks

Here, the LSB of both words 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 words 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 following stage output (S1 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, a 1-bit comparator will be designed first and using the 1-bit comparator block, a 2-bit comparator will be designed.

**Theory and Methodology:** Magnitude Comparators are combinational logic circuits that take two sets of data as their inputs and test whether the value represented by one binary word is more significant than, less than, or equal to the value represented by another word.



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

Depending on the input combination for a 1-bit magnitude comparator, the 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;**

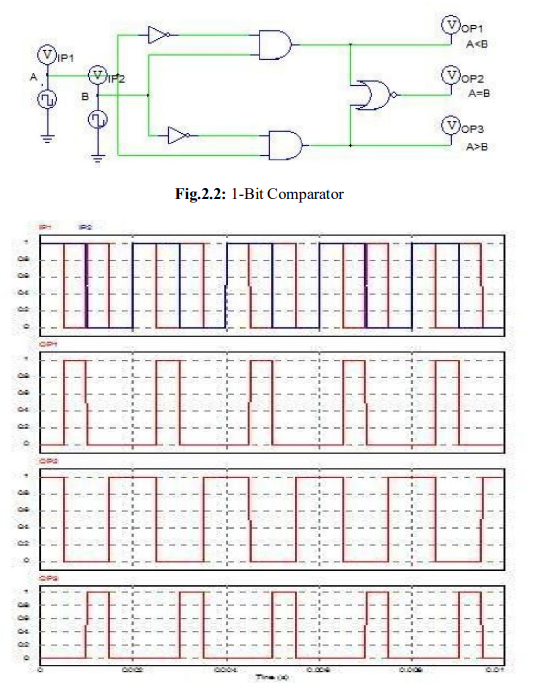
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** |  | **B** |  | **A=B** |  |  | **A>B** |  |  | **A<B** |  |
| **0** | | **0** | | **1** | | | **0** | | | **0** | | |
| **0** | | **1** | | **0** | | | **0** | | | **1**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

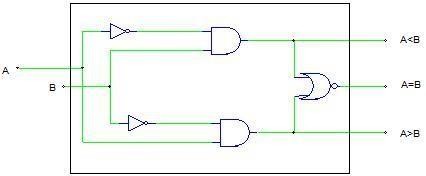
**(A>B) = A’B;**

****

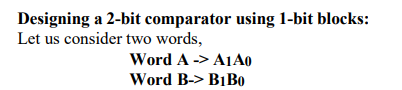
**Fig.2.3:** Timing Diagram for 1-BitComparator

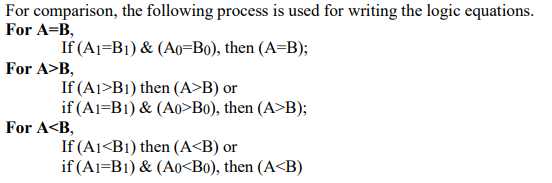
**2-Bit Comparator design using bit block:**

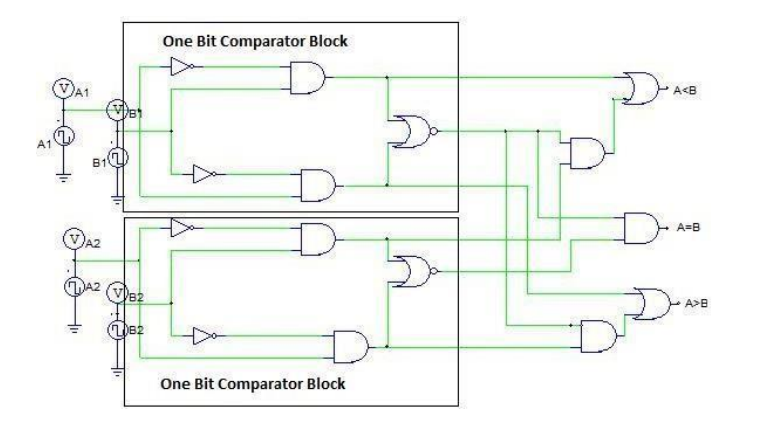
Using 1-bit blocks, an n-bit Magnitude comparator can be designed.



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



****

****

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

For designing a 2-bit comparator using a 1-bit comparator block, two 1-bit comparator blocks, 3 AND gates, and 2 OR entrances are needed, as shown in Fig.4.

**Apparatus:**

1. Digital trainer board

2. IC 7408:2pcs

3. IC 7404:2pcs

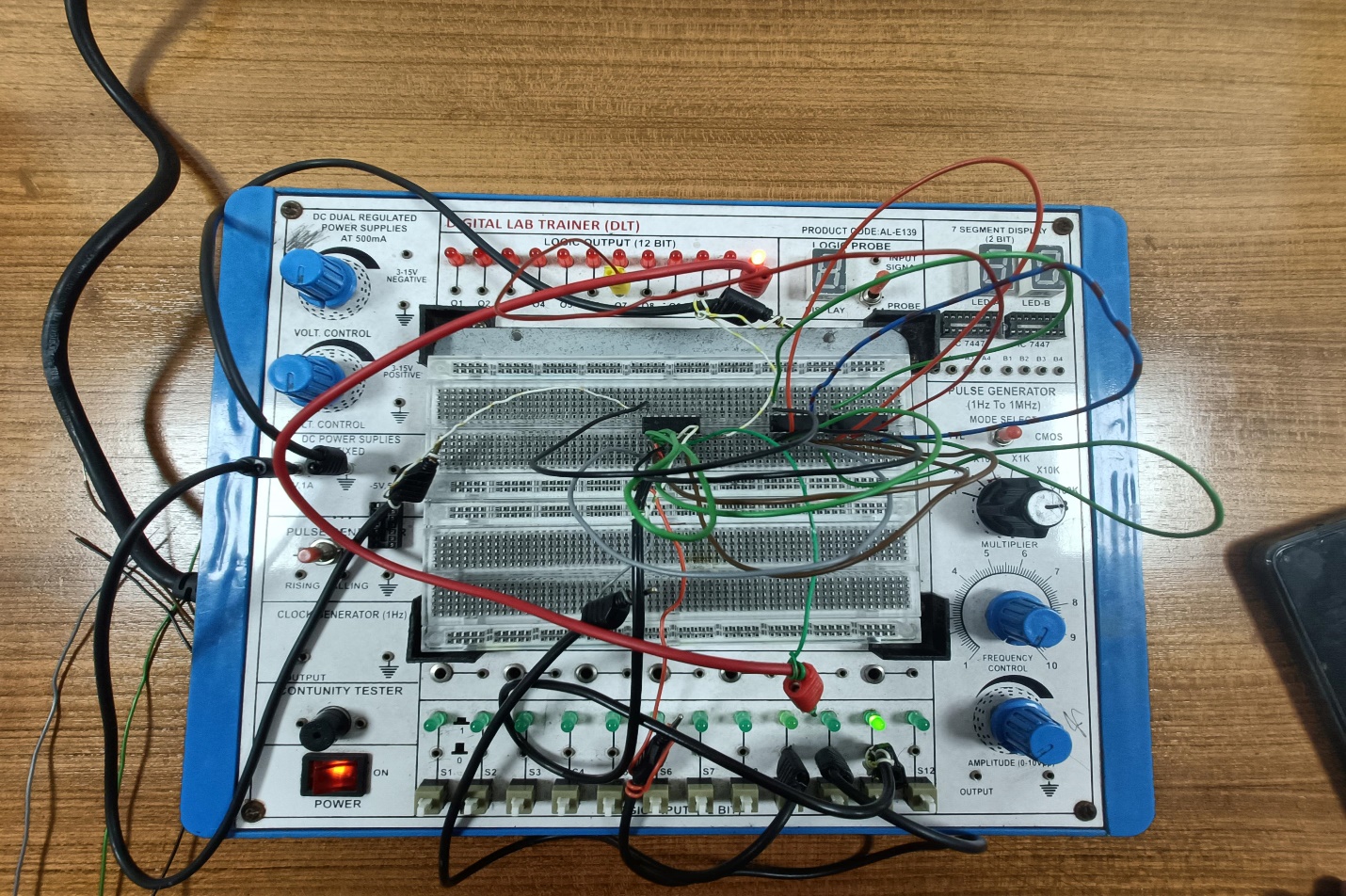
4. IC 7486:2pcs

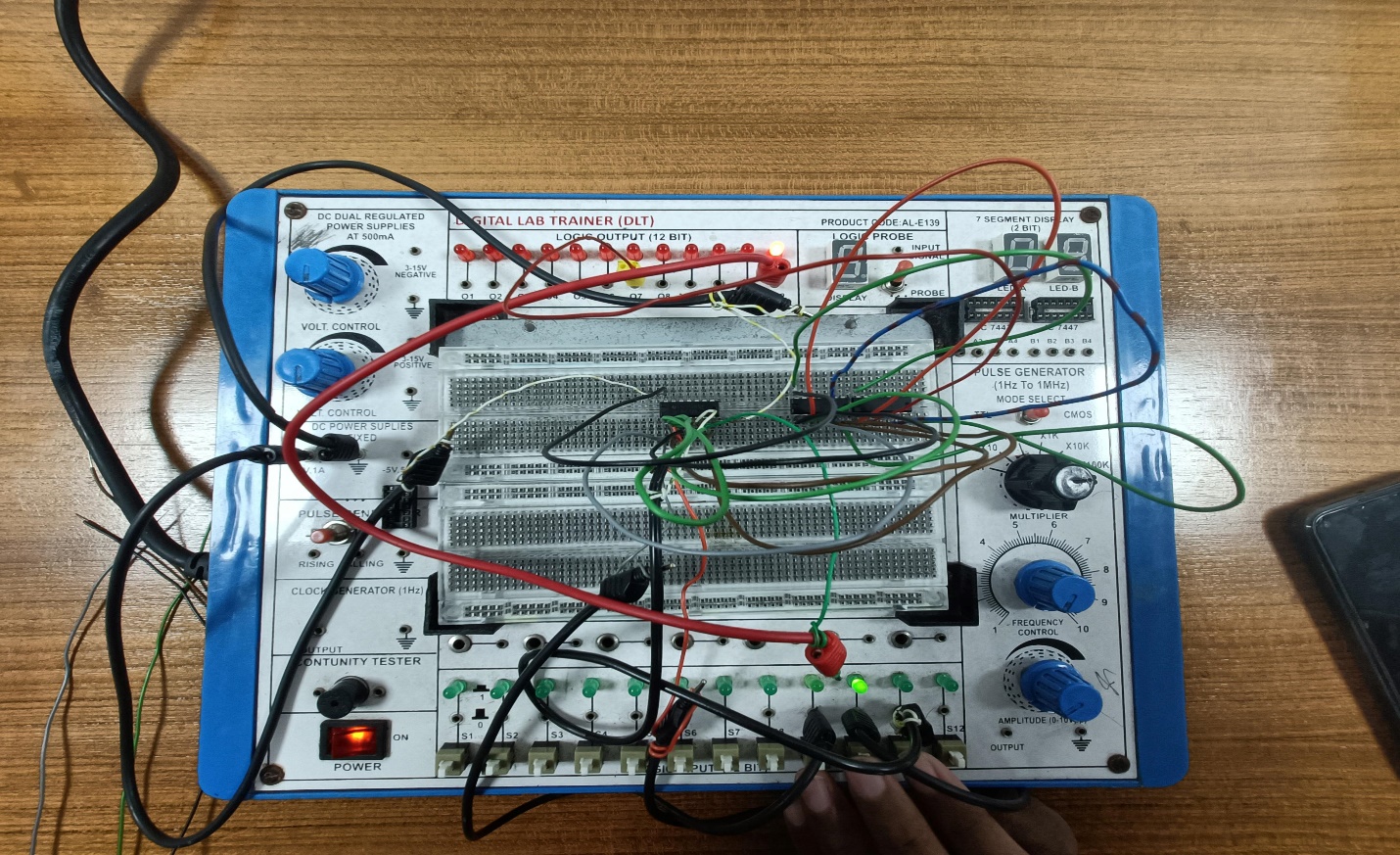
5. IC 7431:2pcs

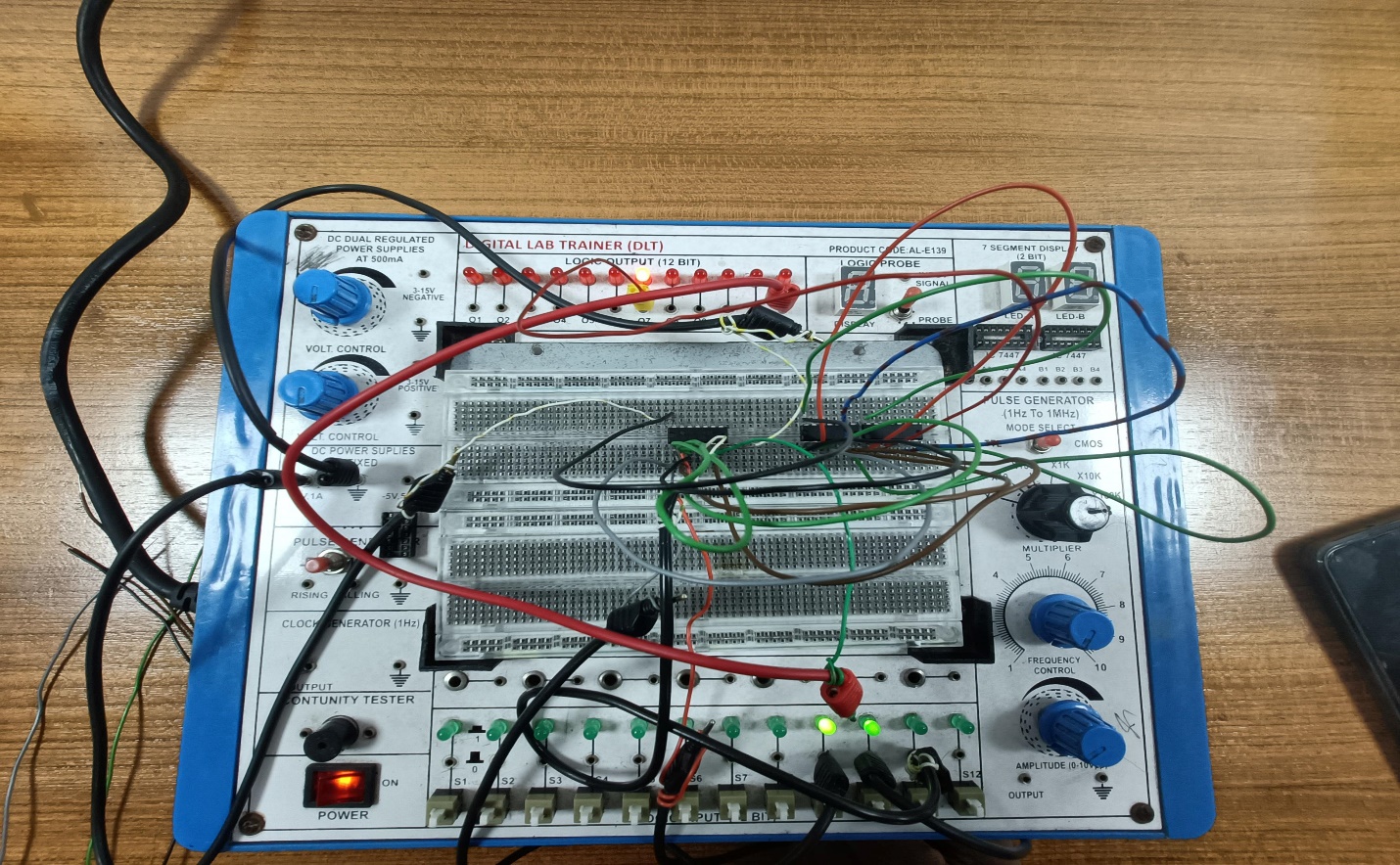
6. IC 7483:1pcs

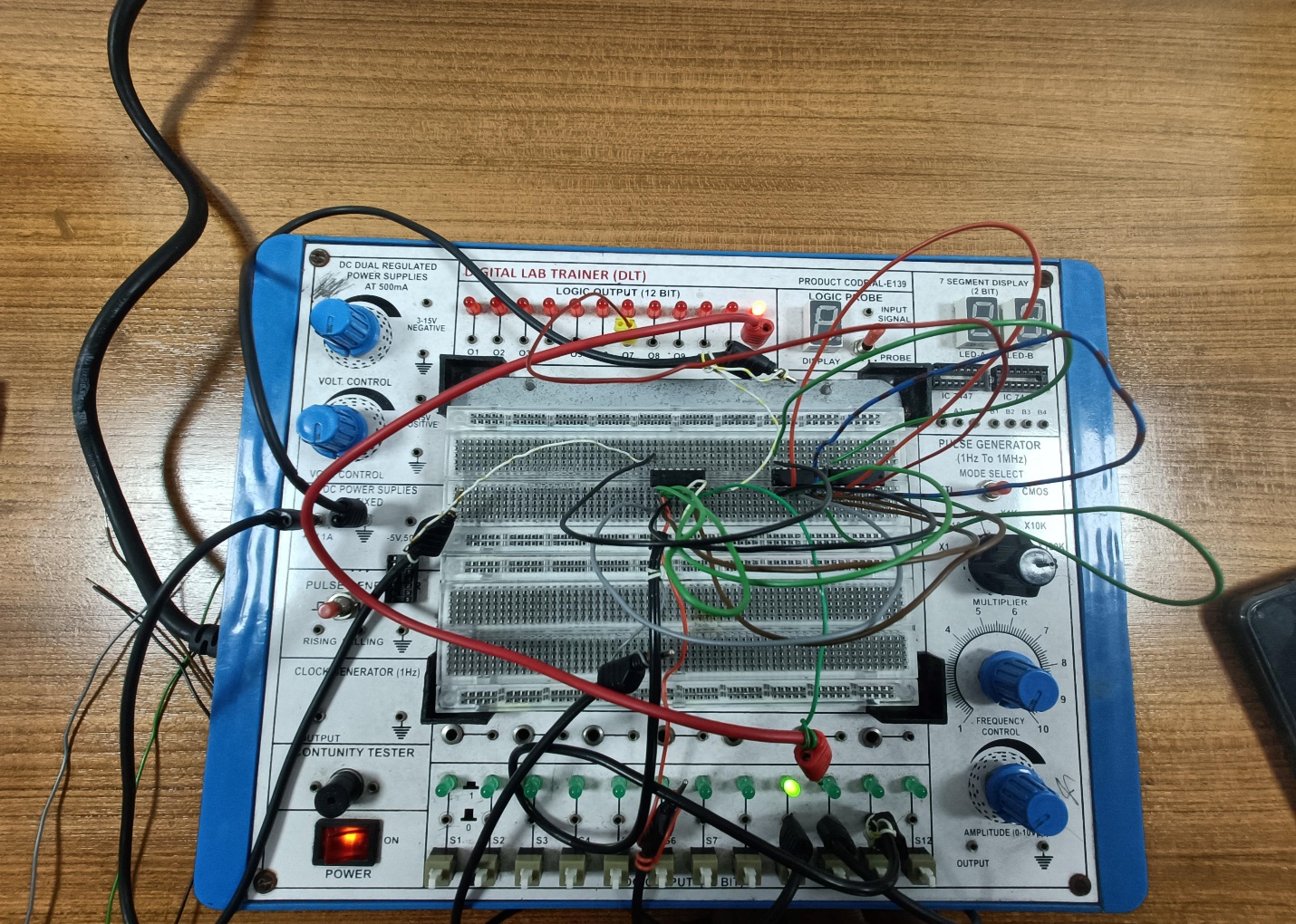
7. Connecting wires

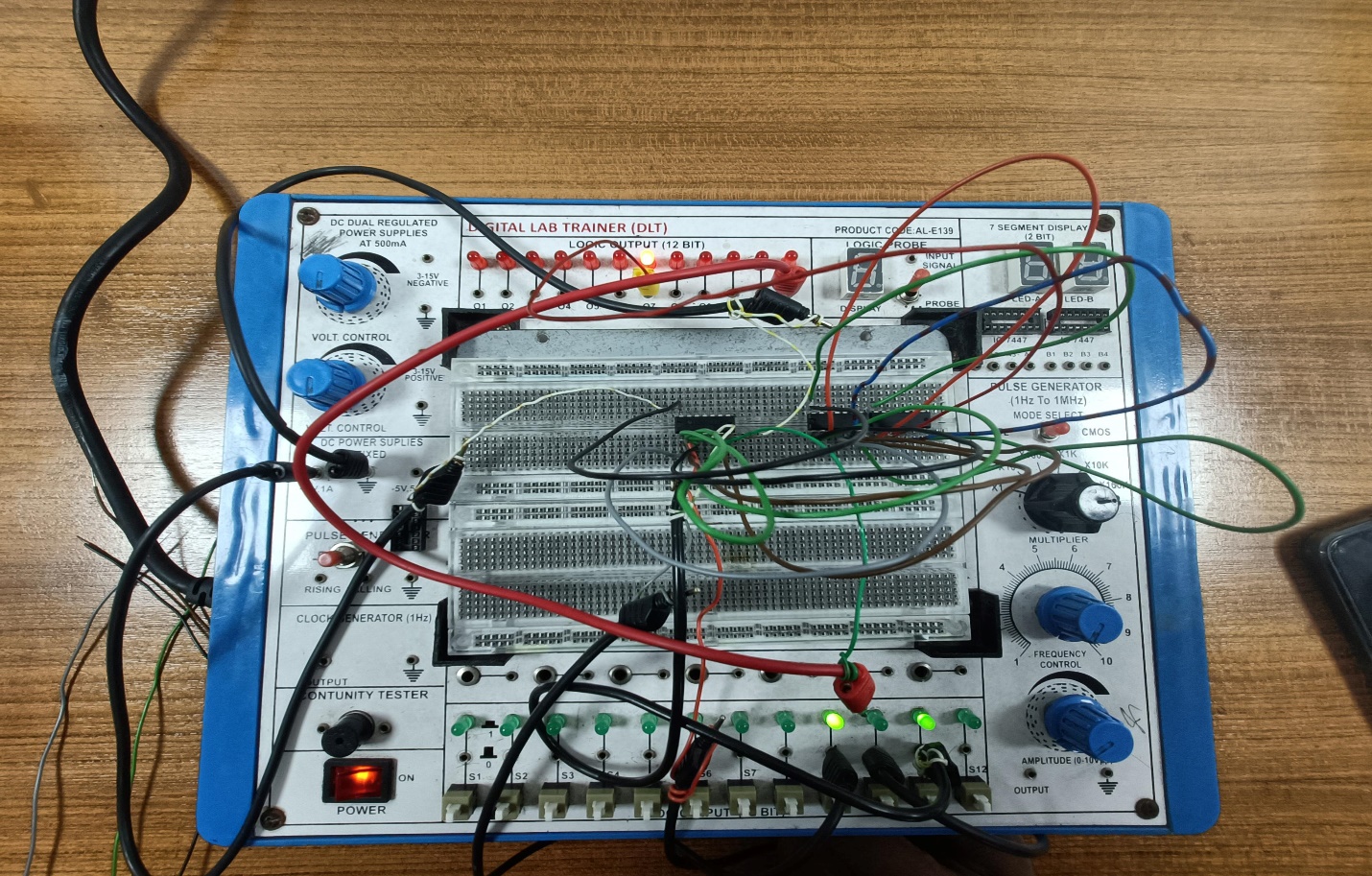
**Hardware Implementation:**

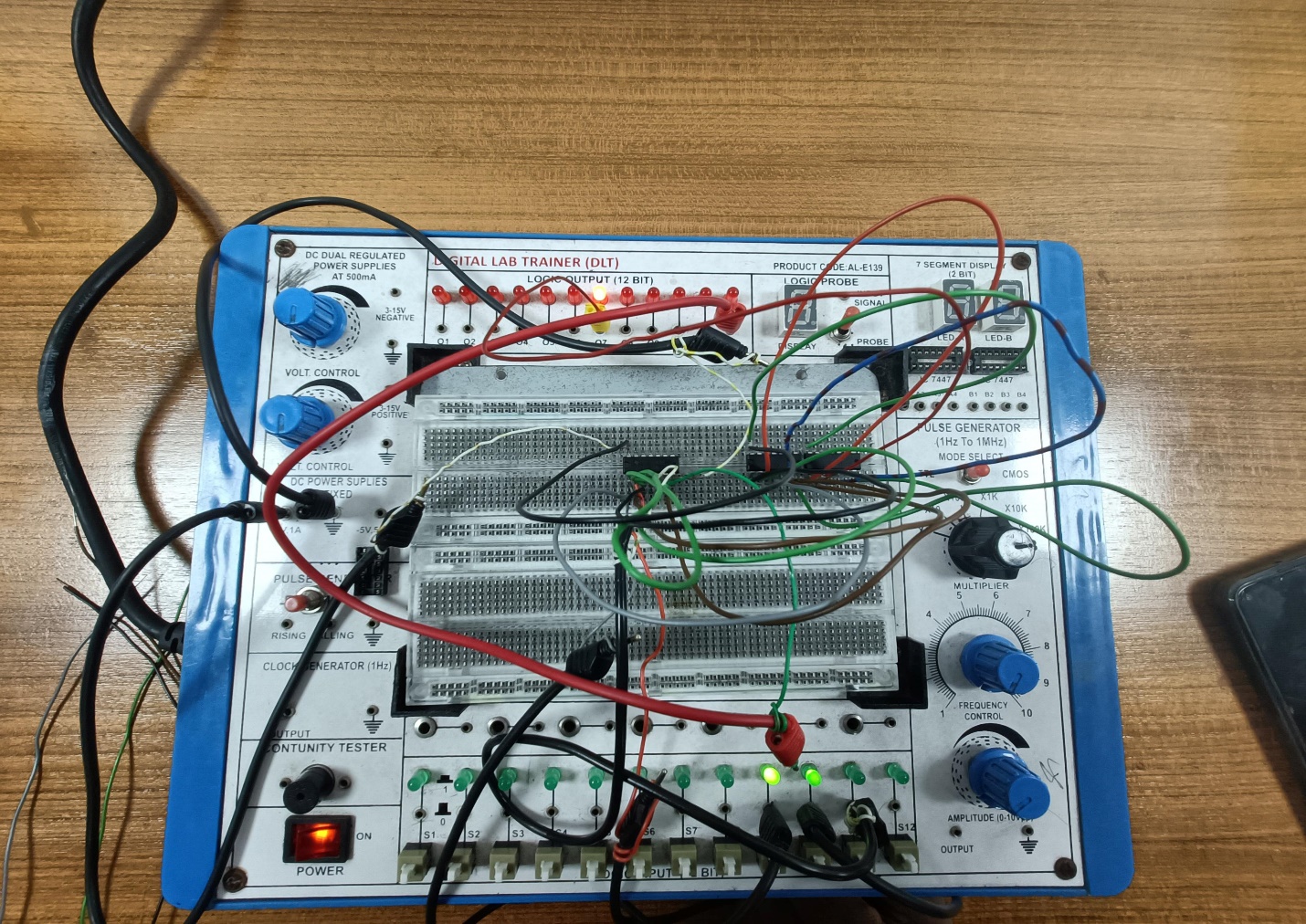
****

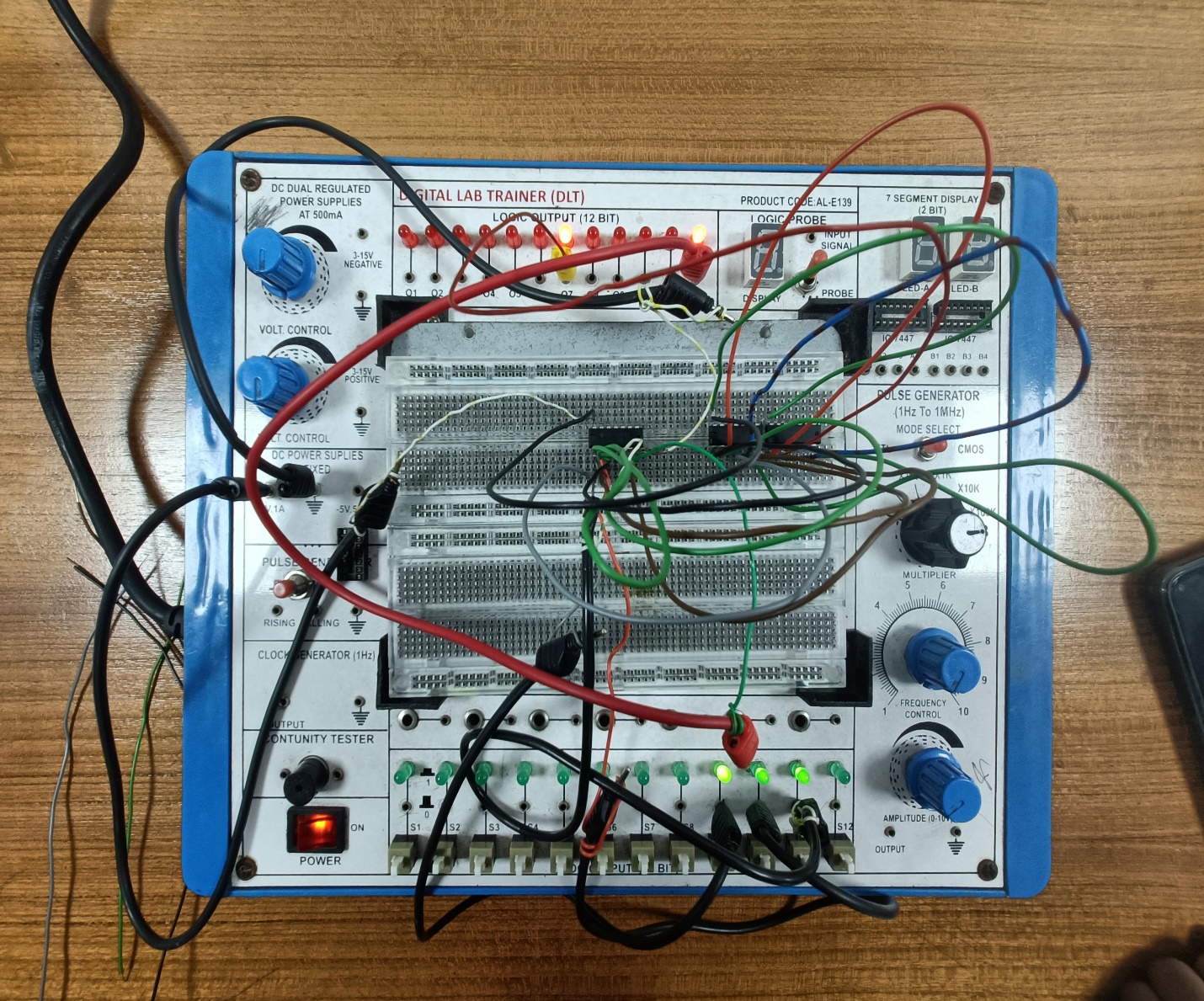
****









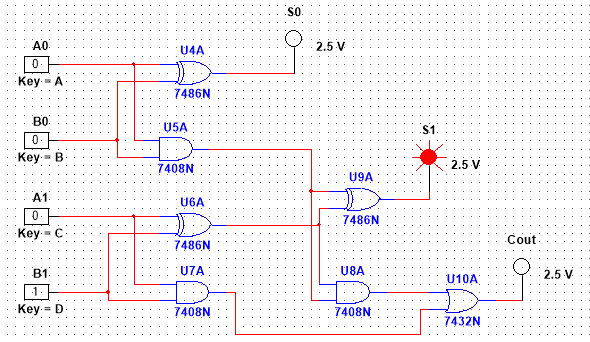


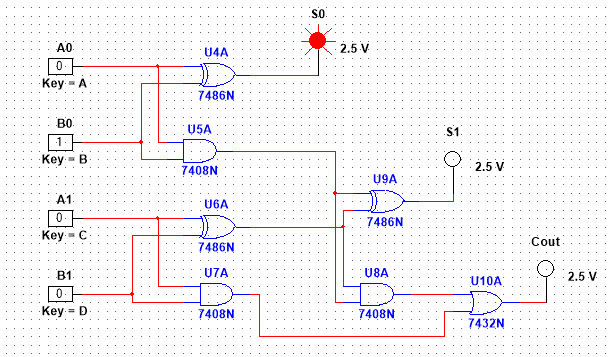
**Simulation:**

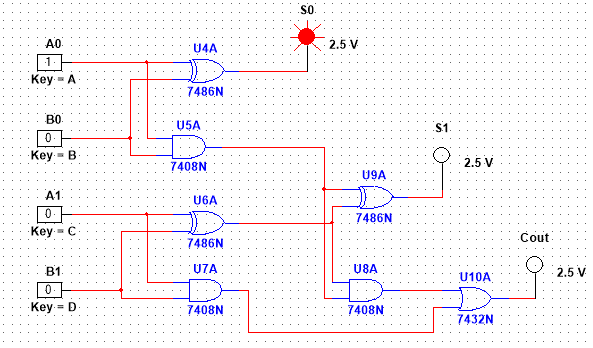
2-Bit Full Adder using 1-Bit Full Adder:

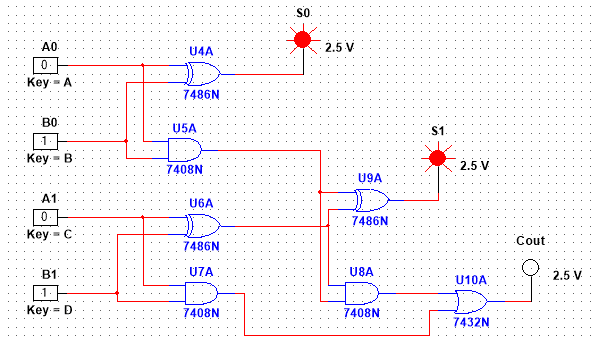
**Truth Table**

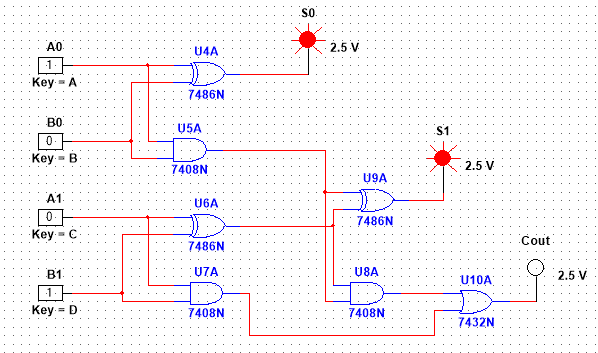
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A1 | A0 | B1 | B0 | Cout | S1 | S0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

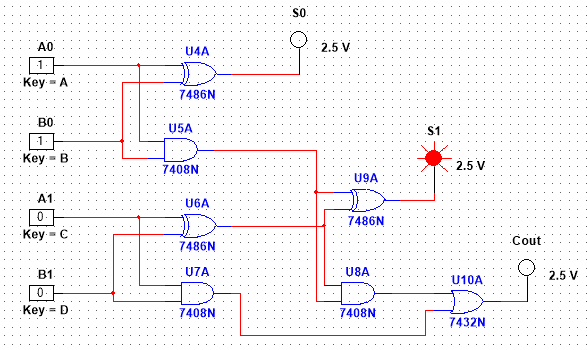


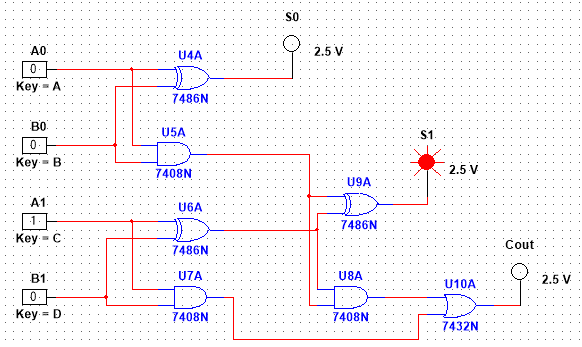


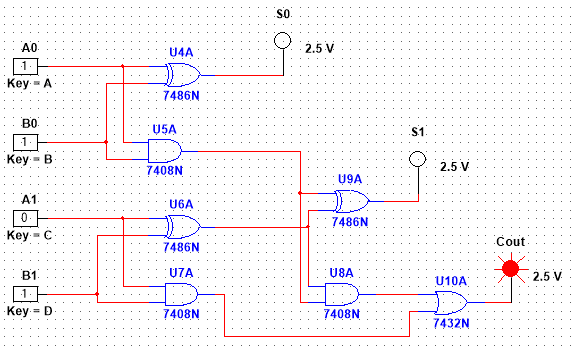


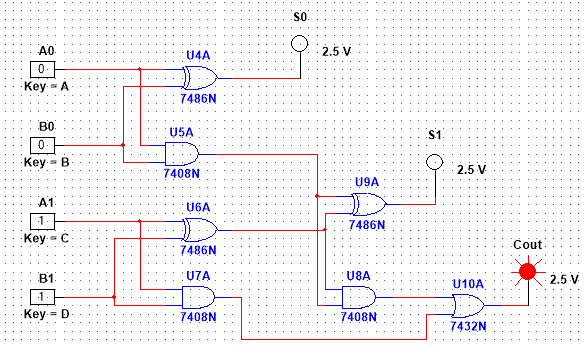


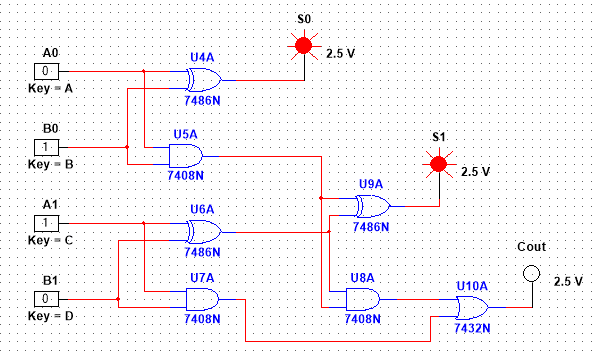


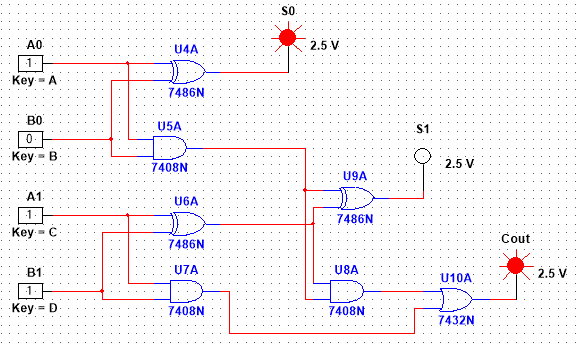




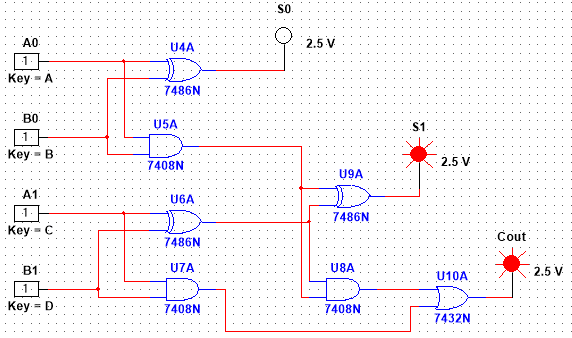








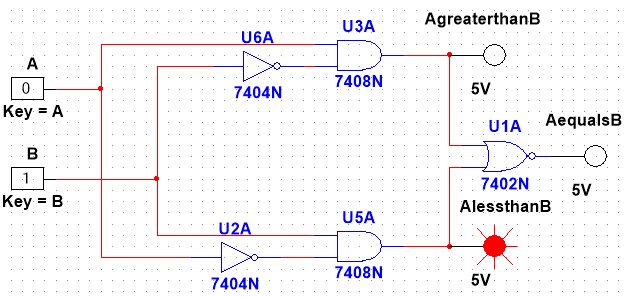


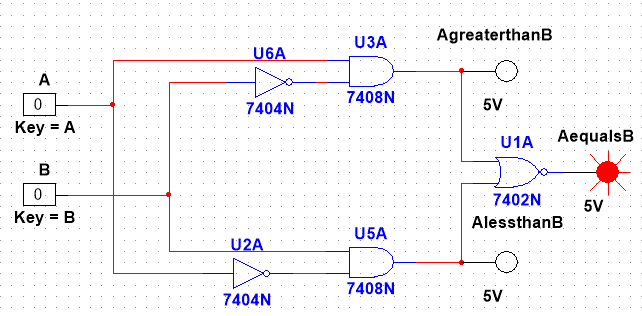


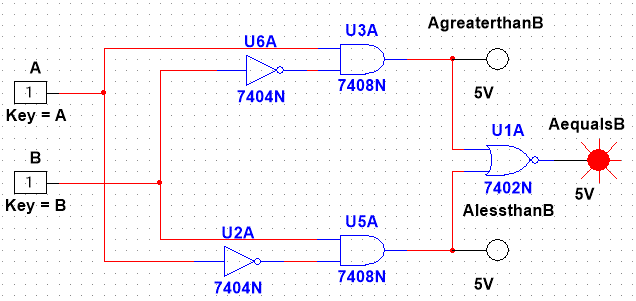
1-Bit Comparator:

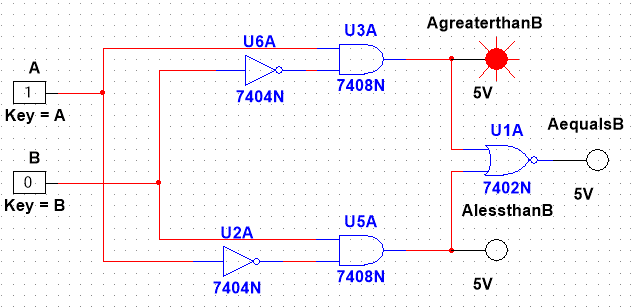
**Truth Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |





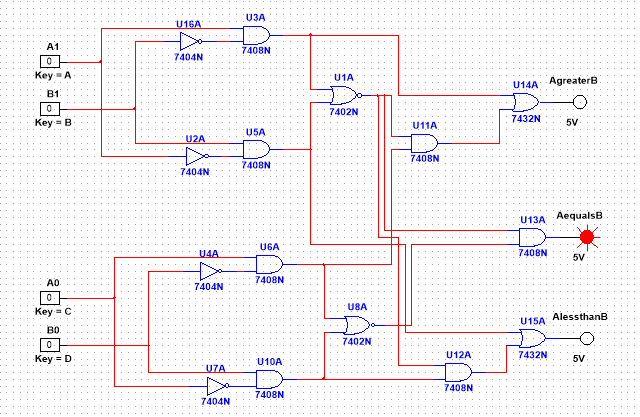


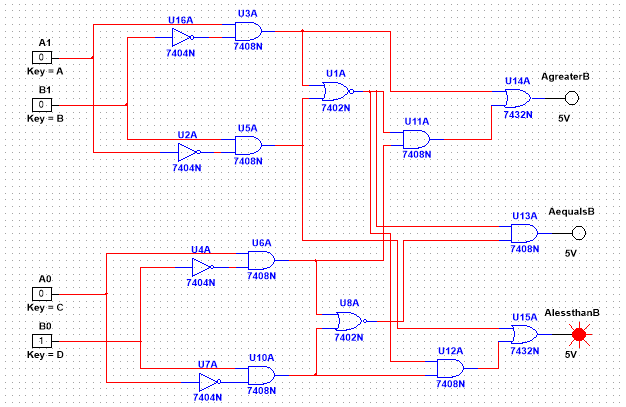


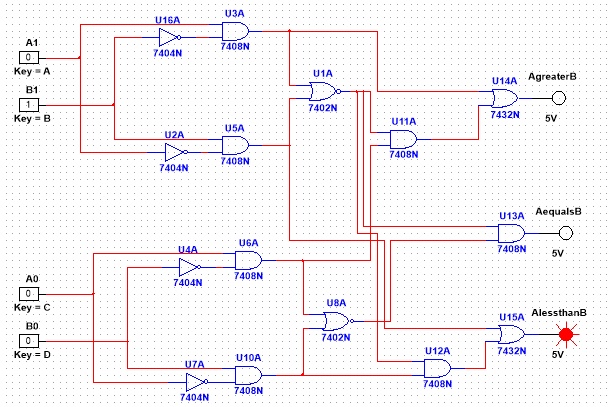
2-Bit Comparator Using 1-Bit Comparator:

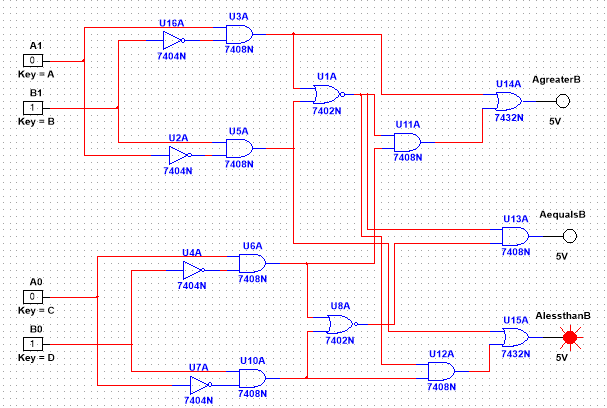
**Truth Table**

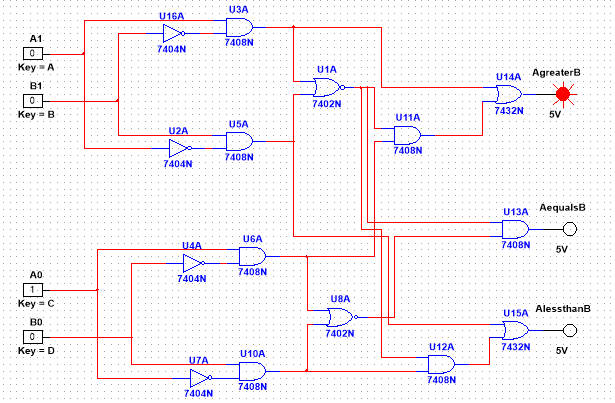
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **INPUTS** |  |  |  | **OUTPUTS** | |  |
| **A1** | **A0** | **B1** | **B0** | **A=B** | | **A>B** | **A<B** |
| 0 | 0 | 0 | 0 | 1 | | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | | 0 | 0 |

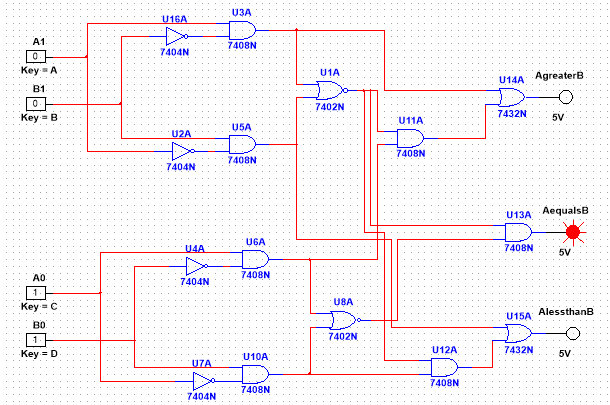
****

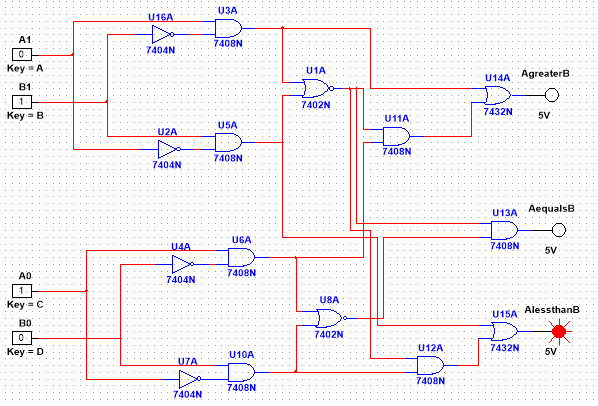
****

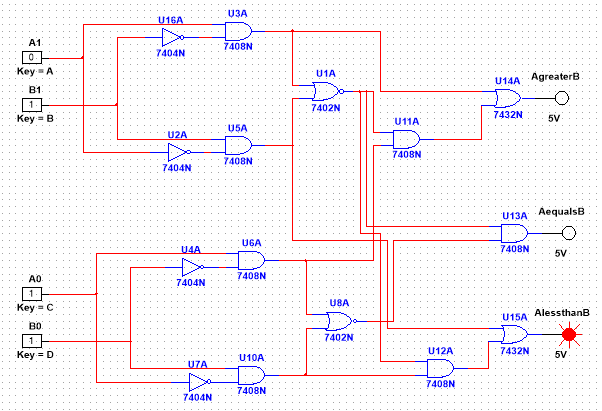
****

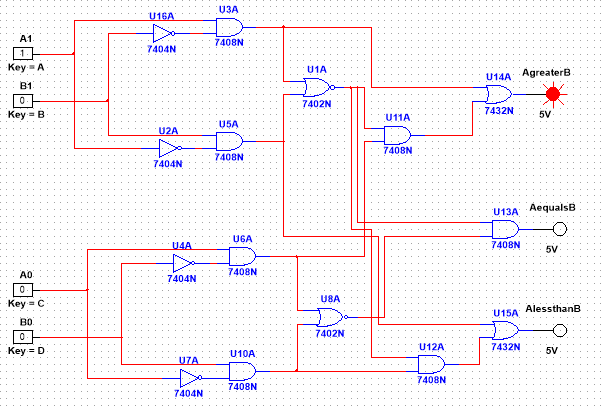
****

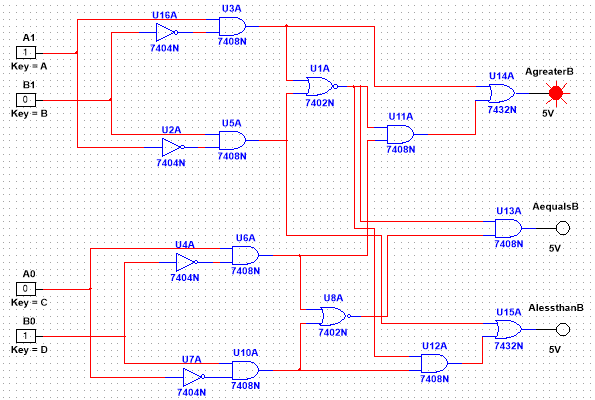
****

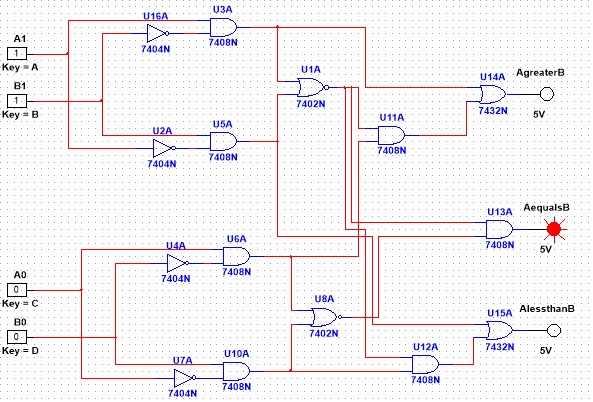
****

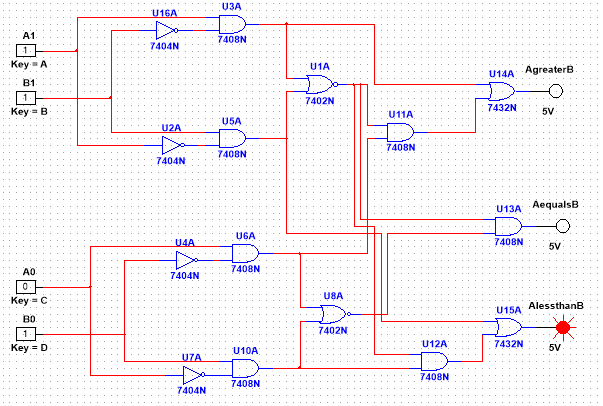
****

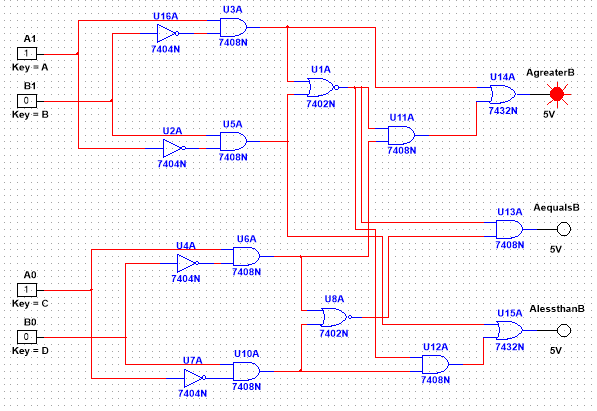
****

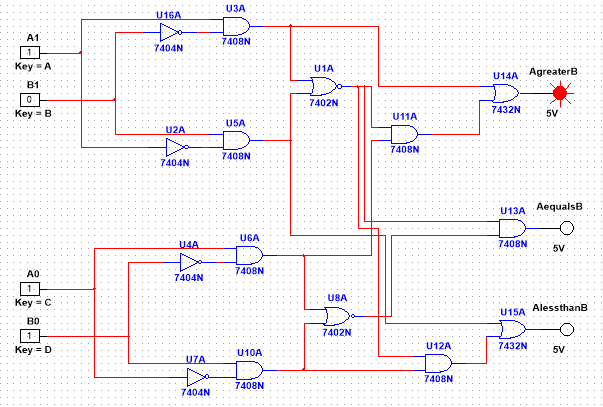
****

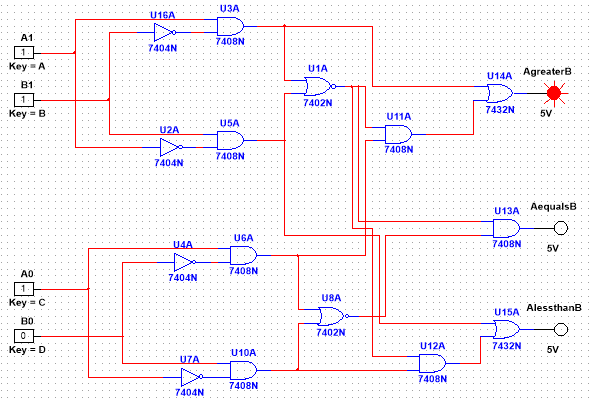
****

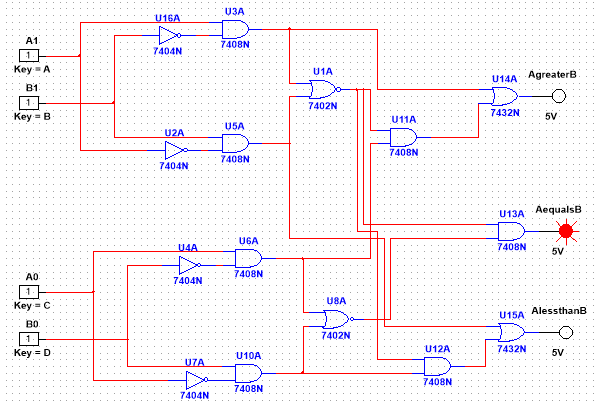
****

****

****

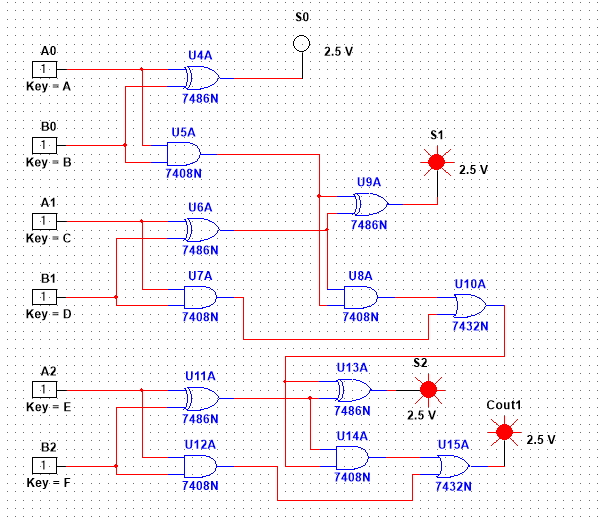
****

****

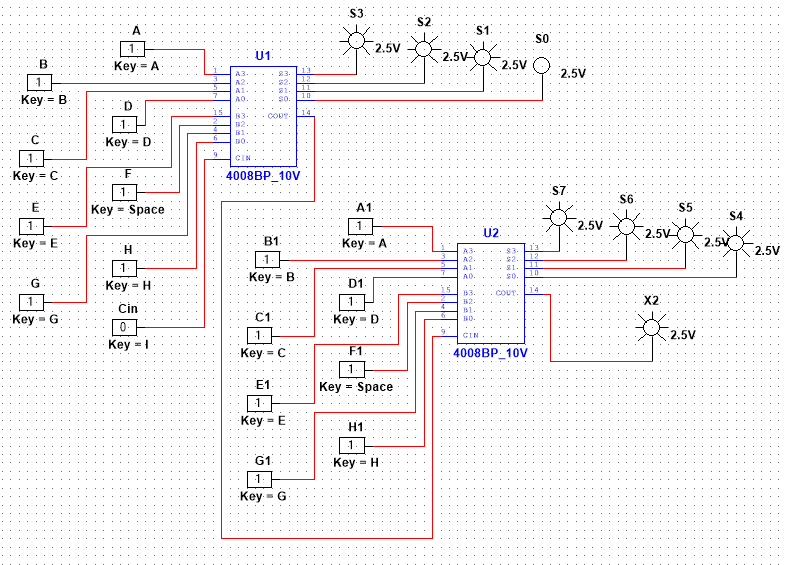
****

**Report Question:**

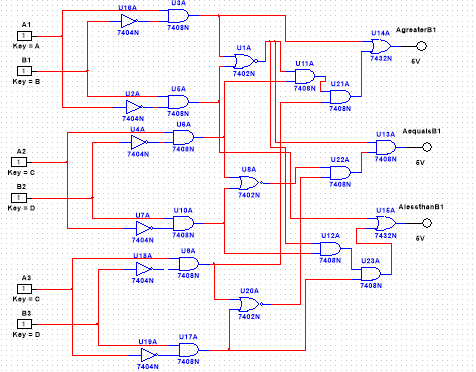
1. Design a full adder circuit for performing 3-bit binary addition.

****

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

****

1. Design a comparator circuit for comparing two words, each of 3 bits of input using 1 bit block.

****

**Discussion and Conclusion:**

In this experiment, we acquired fundamental understanding of learning and implementing an Adder and Comparator. The performance of our system was flawless throughout the simulation phase, as confirmed by the truth table. Ultimately, we conducted a thorough comparison between the simulation results and our truth table, and they were found to be completely identical.

**Reference:**

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