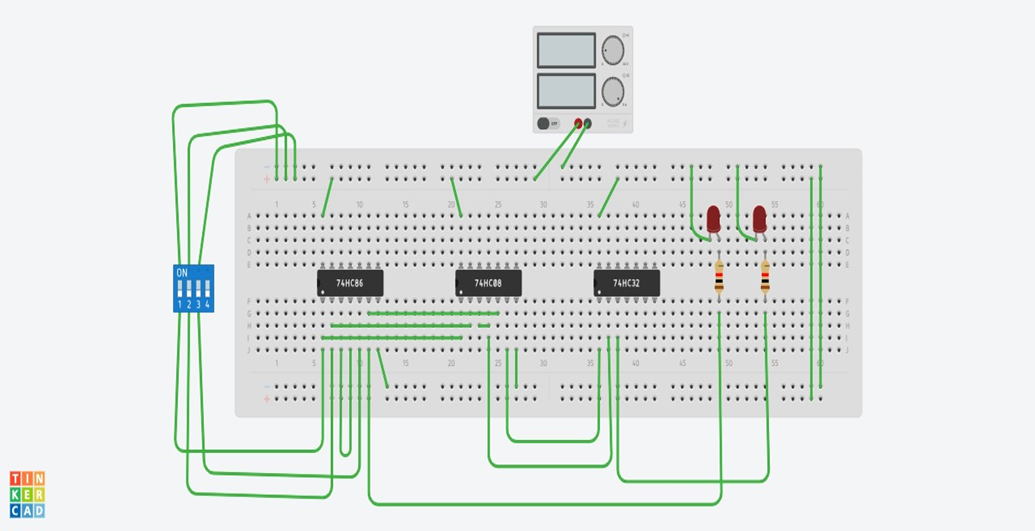
**Implementing Full Adder Using Two Half Adders**

1. **First Half Adder**: The first two 1-bit inputs, let's call them A and B, are added using a half adder. This produces an intermediate sum (S1) and carry (C1). The half adder's sum is given by the XOR of the inputs, and the carry is the AND of the inputs.

2. **Second Half Adder:** The intermediate sum (S1) from the first half adder is then added to the third input, C (carry-in), using the second half adder. This produces the final sum (S) and a new carry (C2). Again, the sum is the XOR of S1 and C, and the carry is the AND of S1 and C.

3. **OR Gate:** The two carry outputs (C1 from the first half adder and C2 from the second half adder) are then combined using an OR gate to produce the final carry-out (C-out)

The implementation of a full adder using two half adders is significant because it allows for the addition of three binary digits and is scalable for constructing multi-bit adders, which are essential in arithmetic operations within digital systems



**Implementing Full Adder in tinker cad**

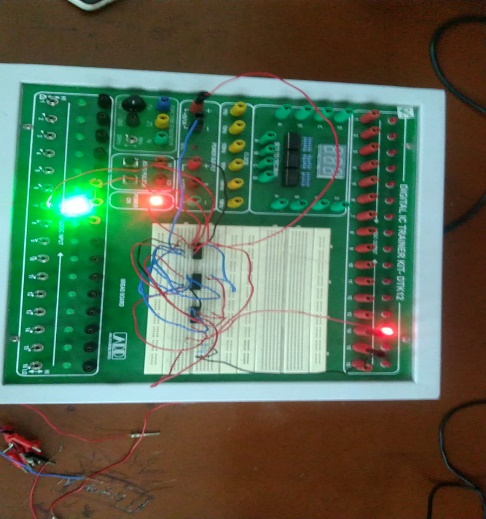
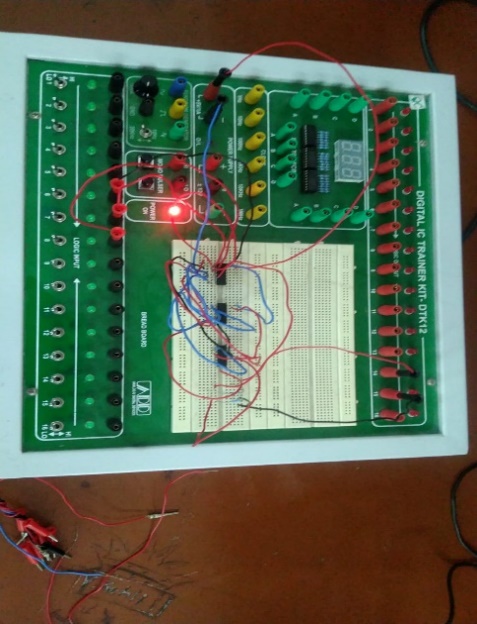
**1. Designing Half Adder in Tinker cad**

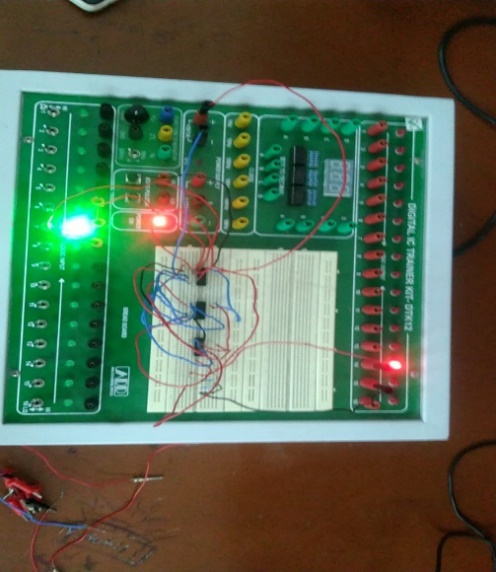
1. Open Tinker cad and create a new circuit.
2. Add two XOR gates and one AND gate from the components library.
3. Connect inputs A and B to the first XOR gate (Sum output).
4. Connect inputs A and B to the first AND gate (Carry output).

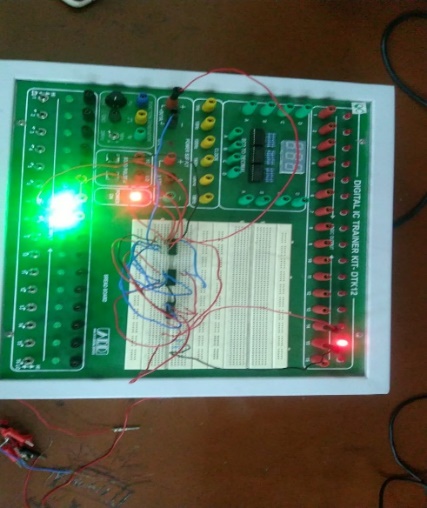
**2. Designing Full Adder Using Two Half Adders**

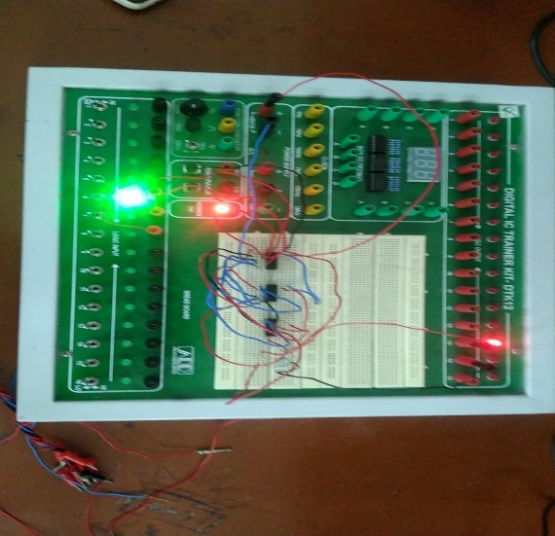
1. Add another XOR gate, another AND gate, and an OR gate to the workspace.
2. Connect the Sum output of the first half adder to one input of the second XOR gate.
3. Connect Cin to the other input of the second XOR gate.
4. Connect the Sum output of the second XOR gate to the final Sum output.
5. Connect the Sum output of the first half adder to one input of the second AND gate.
6. Connect Cin to the other input of the second AND gate.
7. Connect the outputs of both AND gates to the OR gate.
8. Connect the output of the OR gate to the final Carry-out.

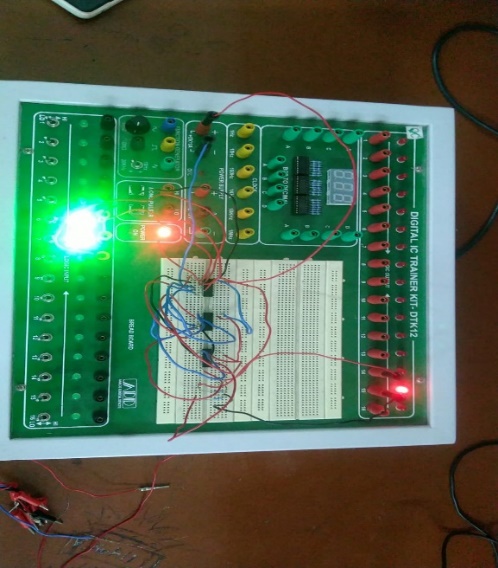
**HARDWARE RESULT**

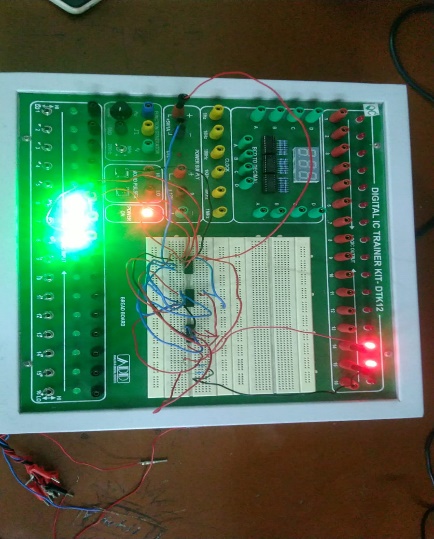




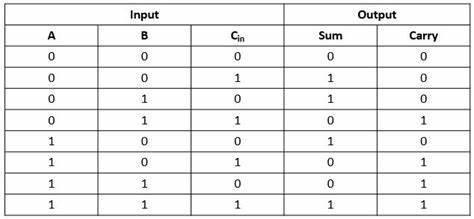








**TRUTH TABLE**



When comparing the simulation and hardware results of a full adder constructed using two half adders, it's important to consider the following aspects:

**Accuracy:** Simulation tools can provide a detailed analysis of the circuit's behavior under various conditions, but actual hardware can exhibit variations due to factors like temperature, manufacturing imperfections, and electrical noise¹.

**Performance Metrics**: Simulations can predict performance metrics such as power consumption, delay, and power-delay product (PDP). However, these metrics can only be verified through hardware testing.

**Design Optimization:** Simulations allow for the exploration of different design options and optimizations before committing to a hardware prototype. This can include testing different logic gate configurations or transistor-level designs.

**Scalability:** Simulations can easily scale to larger systems, whereas hardware implementations may face practical limitations in terms of size, cost, and complexity

In summary, while simulations are essential for initial design and testing, hardware results provide the final validation of the circuit's functionality and performance. It's common to iterate between simulation and hardware testing to refine the design and ensure it meets the required specifications.

**some discrepancies or differences that might be observed are:**

**Propagation Delay:** The simulation might not accurately predict the propagation delay experienced in the hardware. In hardware, the signal must travel through physical gates, which can introduce additional delay.

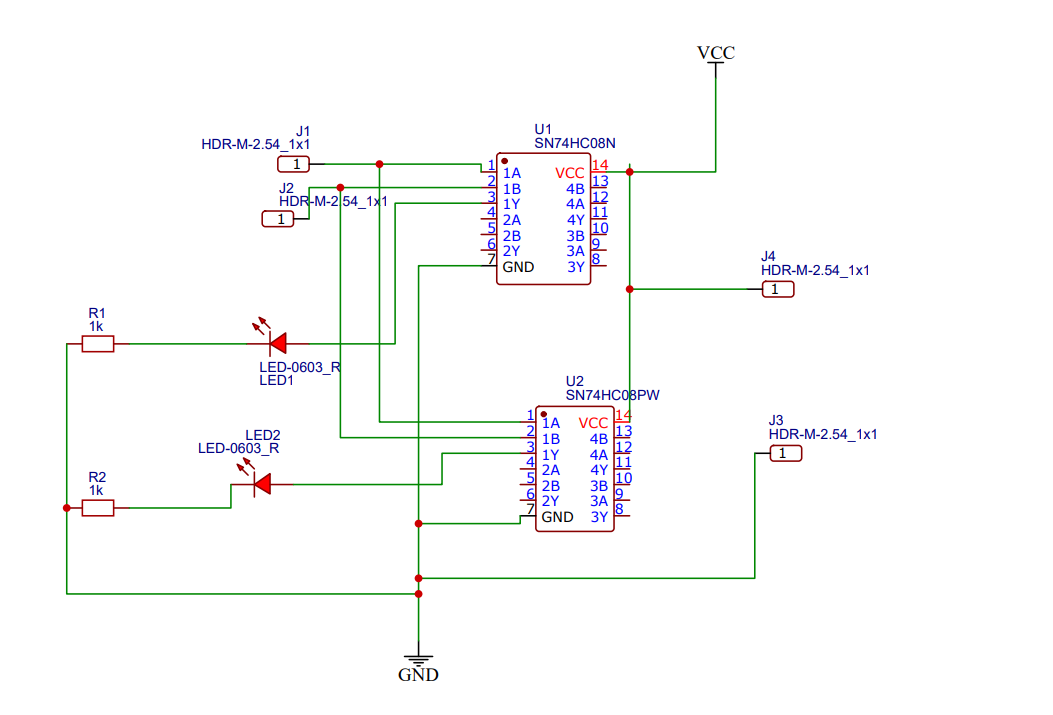
**Power Consumption**: Simulations often estimate power consumption based on ideal conditions. Hardware testing can reveal higher power usage due to factors like leakage current and variations in supply voltage.

**Noise and Interference**: Hardware circuits can be affected by electromagnetic interference and noise, which are not always accounted for in simulations. This can affect the reliability of the output signals.

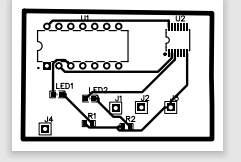
**Manufacturing Variations**: Variations in the manufacturing process can lead to discrepancies in the performance of hardware components, which are not present in the simulation environment.

**Circuit Building on EasyEDA Tools:**

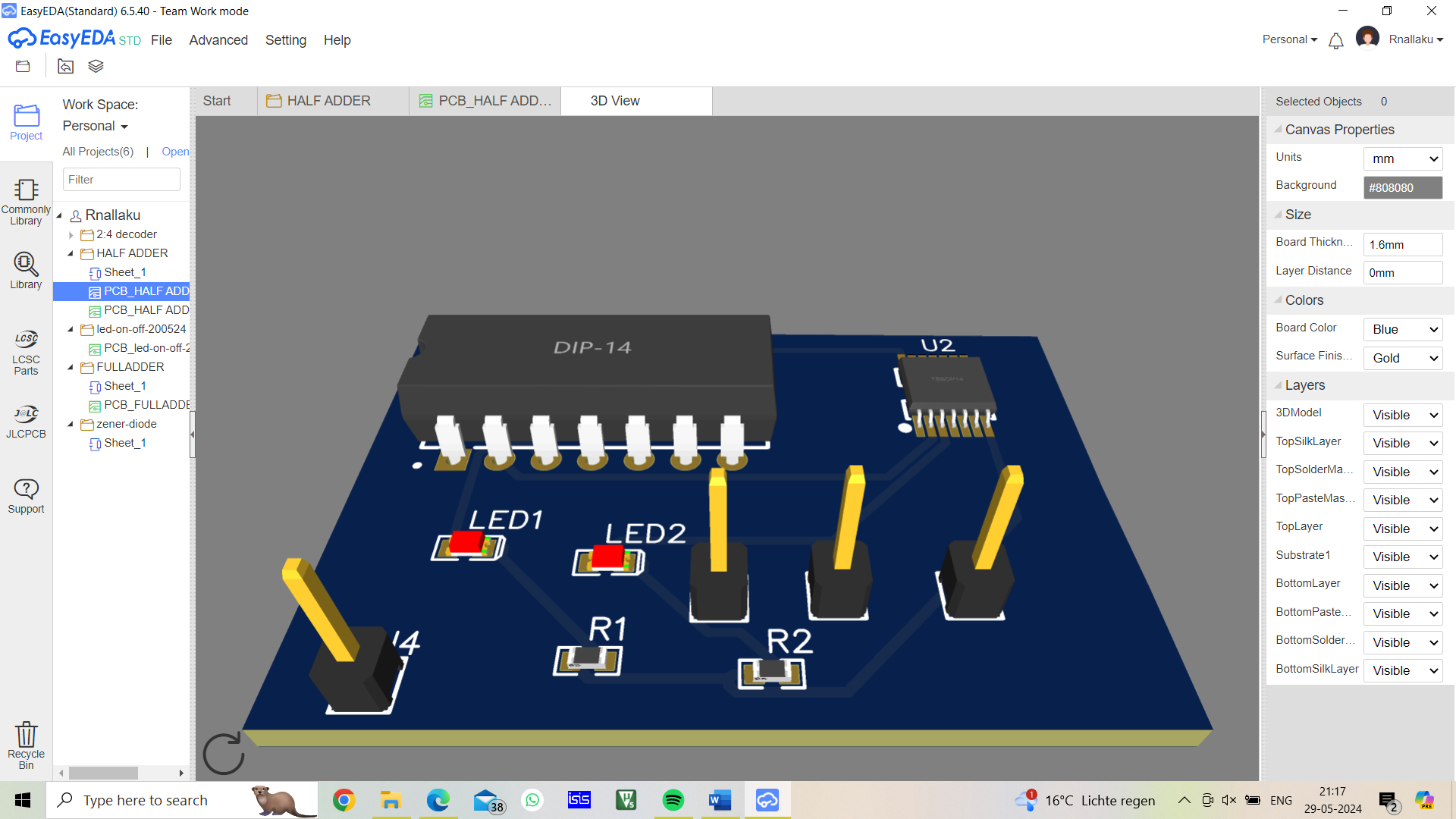
**CRICUIT DIAGRAM**



**PCB CIRCUIT DIAGRAM**



**PCB Designing on EasyEDA Tools:**



**Verification of the Final Design**: **Testing the Circuit**

1. Power on the breadboard.
2. Toggle the input switches and verify that the LED outputs match the expected Sum and Carry-out for all combinations of inputs (A, B, Cin).

**Easy EDA Reference:**

1. **Design on Easy EDA:**
   * Use Easy EDA to draw the schematic of the full adder.
   * Utilize the component library to find XOR, AND, and OR gates.
   * Connect components as per the Tinker cad circuit.
2. **Simulate the Circuit:**
   * Use Easy EDA's simulation feature to test the circuit before hardware implementation

Image of output circuit using in tinker cad and hardware and easy EDA

Appendix

* Learnt how to design circuit in easy eda and tinkercad
* Learnt how to execute code for aurdino
* PCB design
* Learnt about digital in digital out analog in analog out
* Serial communication Group Candidates Are:

Salomi.s-BU22EECE0200006

Karthik.s-BU22EECE0200007

Pavankumar-BU22EECE0200014

Rakshitha-BU22EECE0200019