GITAM

(DEEMED TO BE UNIVERSITY)

TITLE:PCB REPORT

SUBTITLE:DIGITAL CIRCUIT

Name: SINCHANA R

Program:ECE-AIML

Regis. No. :BU22EECE0100441

Academic Year:2022-2026

PROTOCOL TITLE: 3*8 Decoder

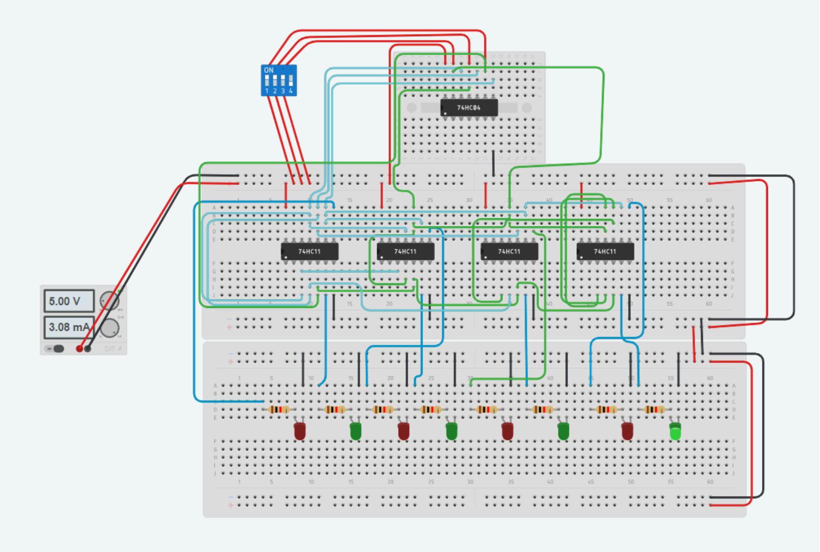
SIMULATION RESULTS:

1. Detailed description of the simulation results:

In Tinkercad and EasyEDA simulations of a 3-to-8 decoder, each 3-bit input combination (000 to 111) correctly activates its corresponding output (Y0 to Y7), with only one output high at a time. The timing diagrams confirm accurate transitions without glitches, ensuring stable operation. Both platforms demonstrate reliable decoder functionality, with EasyEDA offering detailed timing analysis and Tinkercad providing intuitive visualisation, making it easy to verify the decoder's correctness and performance.

2. Include screenshots:

   Note:Screenshots of the simulation results will be included here.These should capture the significant outputs, such as waveforms and plots, which provide insight into the circuit's performance.



3. Illustrate key aspects of the simulation:

- Input/Output Waveforms: Verify that for each 3-bit input combination, the corresponding output is high, and the rest are low.
  INPUT WAVEFORMS:The input waveforms for the 3-to-8 decoder consist of the three input lines: A2, A1, and A0. Each input line can be either high (1) or low (0), and they are varied to cover all 8 possible combinations (000 to 111).

```
     A2:  _|‾|_______|‾|_______|‾|_______|‾|_____
     A1:  ____|‾|_______|‾|_______|‾|_______|‾|__
    A0:  _______|‾|_______|‾|_______|‾|_______|‾|
```

OUTPUT WAVEFORMS:For each combination of inputs, one of the 8 output lines Y0 to Y7 is high while the others remain low.

```
Y0: ‾|______________________
 Y1:  _|‾|___________________
 Y2:  __|‾|__________________
 Y3:  ___|‾|_________________
 Y4:  ____|‾|________________
 Y5:  _____|‾|_______________
 Y6:  ______|‾|______________
 Y7:  _______|‾|_____________
```

- Frequency Response Plots: A frequency response plot is not typically relevant for a digital 3-to-8 decoder since it operates based on discrete input values rather than continuous frequencies. Ensure the decoder can handle the desired switching frequency without timing issues.
- Transient Responses: Ensure clean transitions on the outputs without glitches or unexpected pulses.

HARDWARE RESULTS:

Present hardware implementation of the prototype on bread-board:

Components Required:

- ❖ 1x 3-to-8 Decoder IC (e.g., 74LS138)
    - ❖ Breadboard
    - ❖ Jumper wires.
    - ❖ Power supply (5V for 74LS138)
- Include measurements, observations:
    - ❖ Voltage Levels: (0V for logic low and 5V for logic high).
    - ❖ Current through LEDs:(typically around 10-20mA).
- Deviations from the expected behaviour :

  Observations should match the expected single active output for each input combination, with potential deviations typically due to wiring errors or switch bounce.

COMPARISON OF SIMULATION AND HARDWARE RESULTS:

Analyse and compare the simulation results with the hardware results:

Simulation Results:

- Tinkercad and EasyEDA: In the simulations, each 3-bit input combination (000 to 111) correctly activates its corresponding output (Y0 to Y7), with only one output high at a time. Timing diagrams confirm accurate transitions without glitches, ensuring stable operation.

Hardware Results:

- Breadboard Implementation: On the breadboard, when the 3-bit input (A2, A1, A0) is toggled, the corresponding output LED lights up, indicating the correct decoding. Each input combination results in exactly one corresponding LED lighting up, mirroring the simulation results.

Both simulation and hardware results show consistent behaviour, with one and only one output high for each input combination.

Identify any discrepancies or differences:

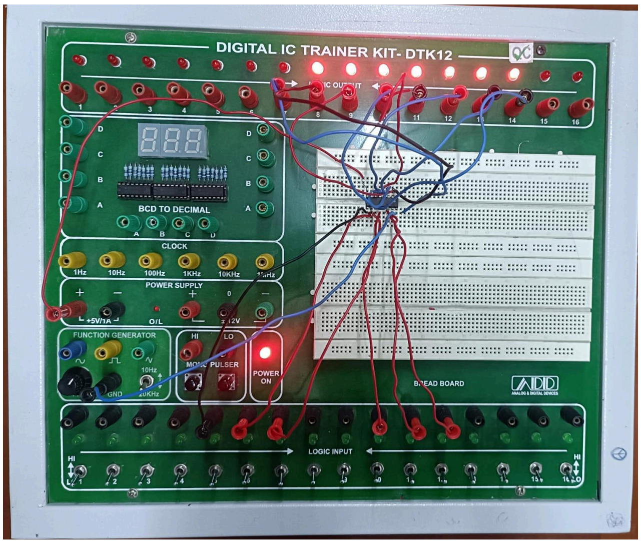
- Potential Discrepancies
- Noise and Stability

- Component Variations

Discuss possible reasons for variations:

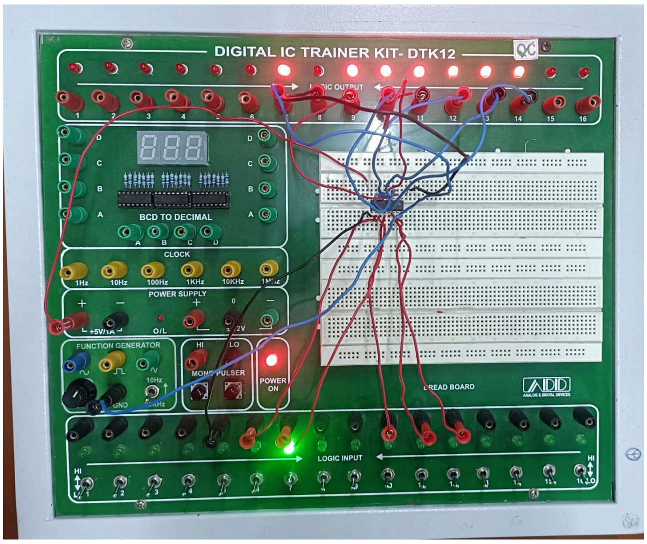
Possible Reasons for Variations:

- Propagation Delay
- Debounce Issues
- Power Supply Noise
- Component Tolerances
- Contact Resistance



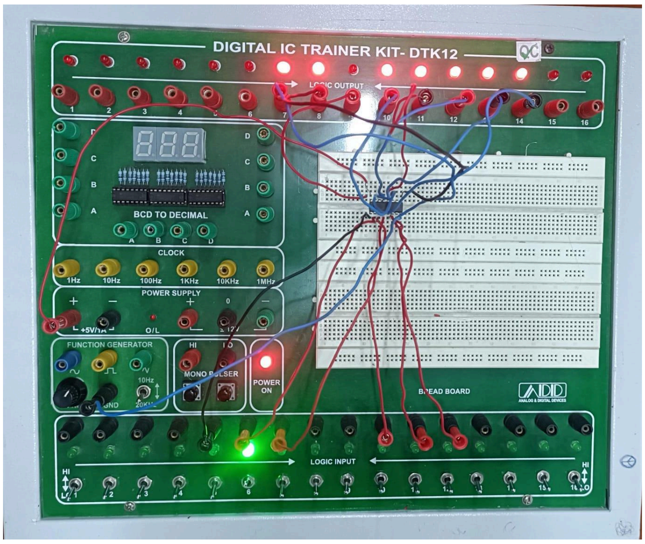
INPUT-000

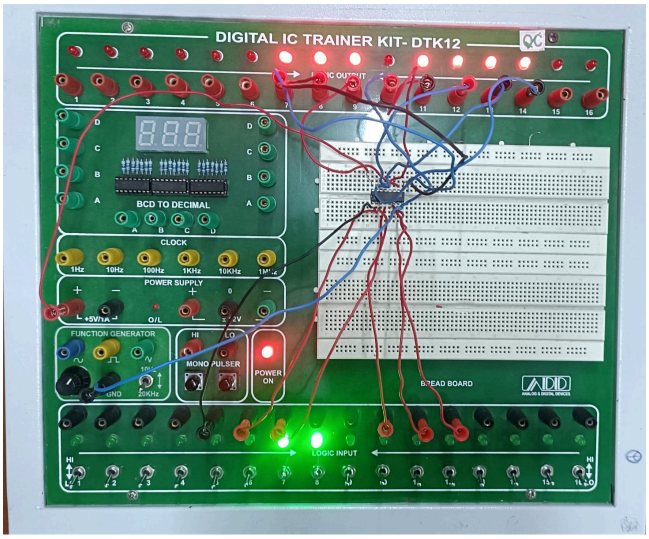
OUTPUT-01111111



INPUT-001

OUTPUT-10111111



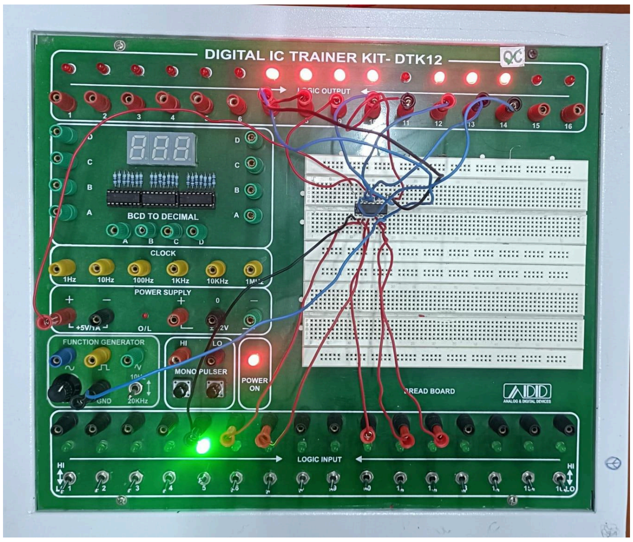


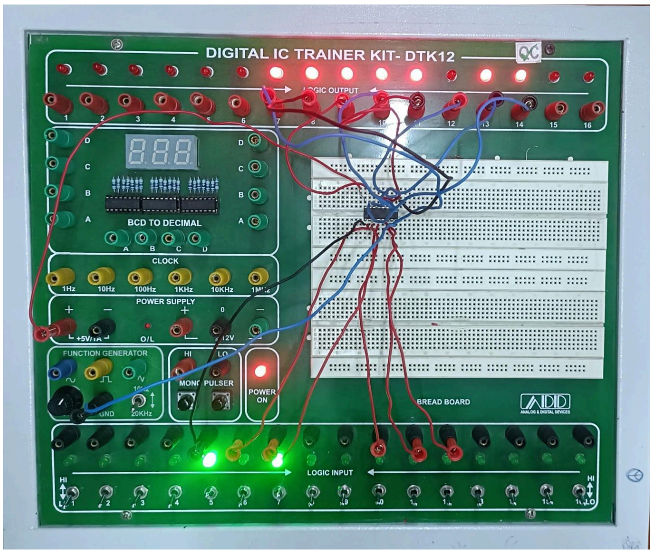
INPUT-010

OUTPUT-11011111

INPUT-011

OUTPUT-11101111



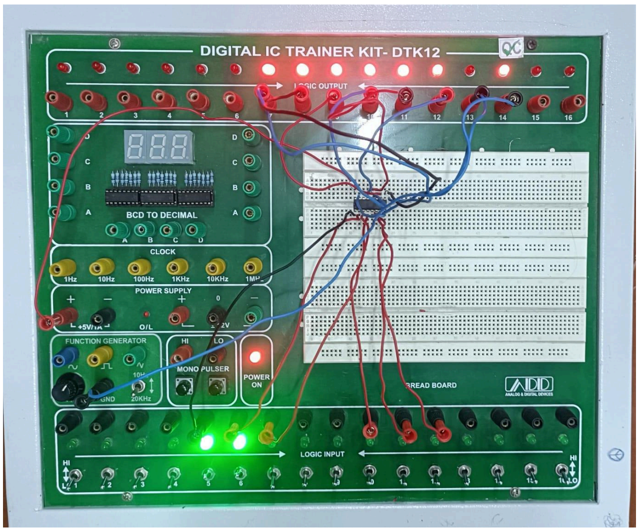


INPUT-100

OUTPUT-11110111

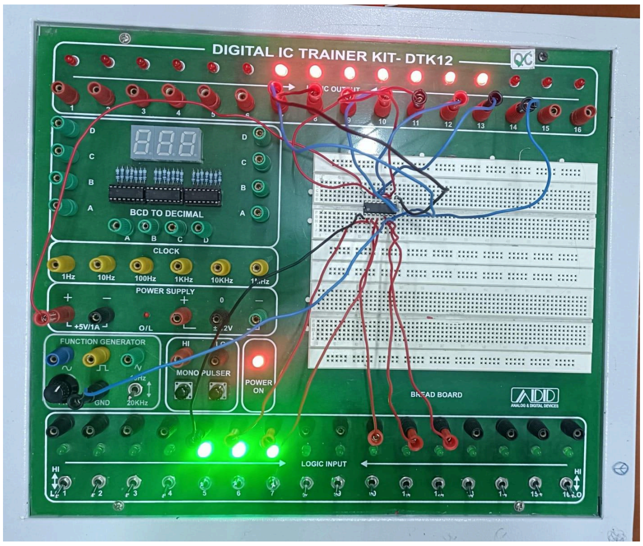
INPUT-101

OUTPUT-111110111



INPUT-110

OUTPUT-11111101



INPUT-111

OUTPUT-11111110

DESIGN FINALISATION:

Document the final design parameters

a. Final Design Parameters

1. Decoder IC: 74LS138 (3-to-8 line decoder)
2. Input Lines: 3 (A2, A1, A0)
3. Output Lines: 8 (Y0 to Y7)
4. Power Supply: 5V DC

b. Fix Specifications Based on Simulation and Hardware Results

1. Operating Voltage: 5V DC
2. Input Logic Levels:
   a. Logic Low (0): 0V to 0.8V
   b. Logic High (1): 2V to 5V

c. Modifications or Optimizations Made to Improve Performance

1. Debouncing Circuit
2. Power Supply Filtering
3. LED Indicators
4. Breadboard Layout

d. Issues Encountered During Testing

1. Switch Bounce
2. Power Supply Noise
3. Poor Connections
4. LED Brightness Variability

CIRCUIT BUILDING ON EasyEDA TOOLS:

- a. Outline the process of translating the finalised design into a circuit layout

  i. Provide step-by-step instructions, Screenshots

  Step 1: Create a New Project

  - ➢ Log in to EasyEDA
  - ➢ Start a New Project

  Step 2: Draw the Schematic

  - ➢ Open Schematic Editor
  - ➢ Add Components:Decoder IC-74LS138" ,Switches, LEDs, Resistors,Capacitors.
  - ➢ Connect Components
  - ➢ Label and Annotate

Step 3: Convert to PCB Layout

- ➢ Switch to PCB Editor
- ➢ Define Board Outline
- ➢ Place Components
- ➢ Route Traces

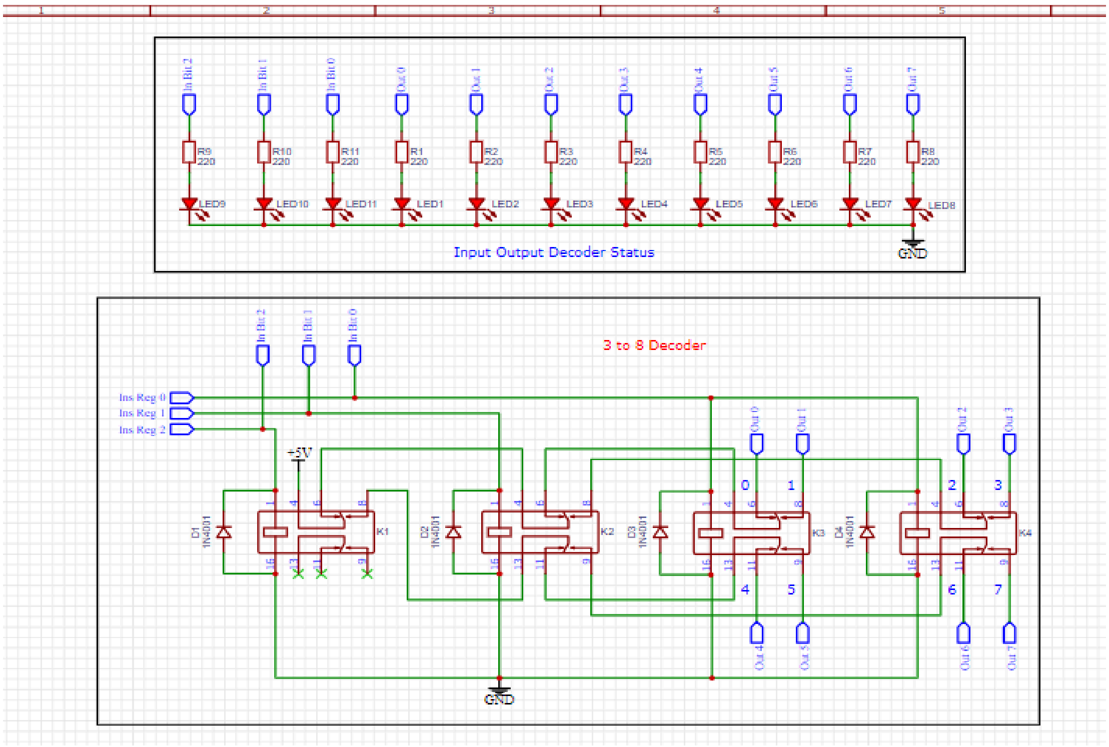
Step 4: Add Silkscreen and Final Touches

- ➢ Labels and Text
- ➢ Final Review
- ➢ DRC Check

Step 5: Generate Gerber Files

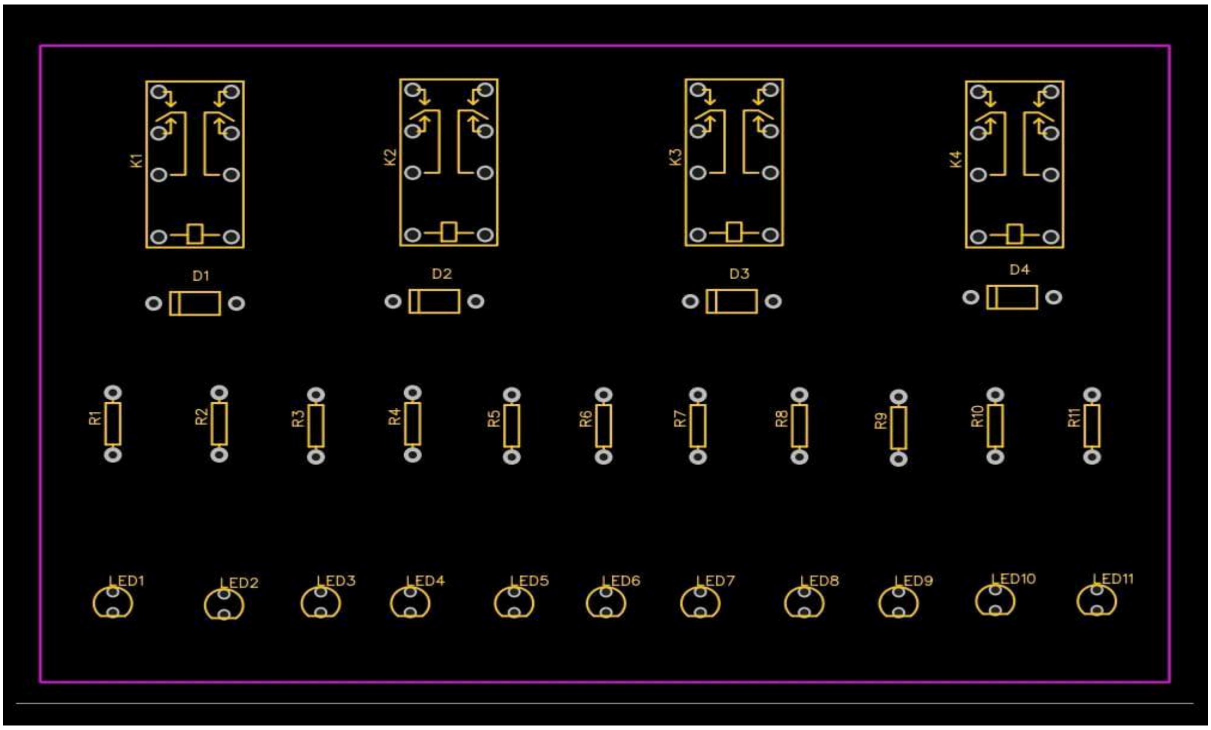
- ➢ Generate Gerber Files
- ➢ Download File

1. Schematic Design in EasyEDA

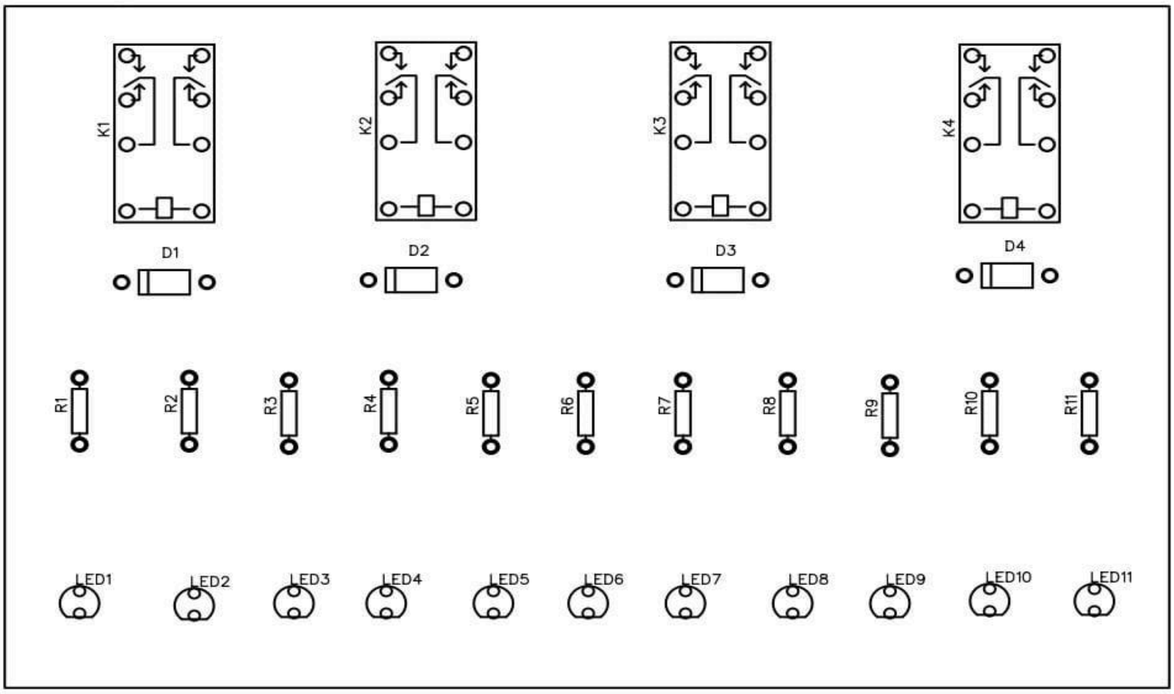


2. PCB Layout in EasyEDA

Component Placement:

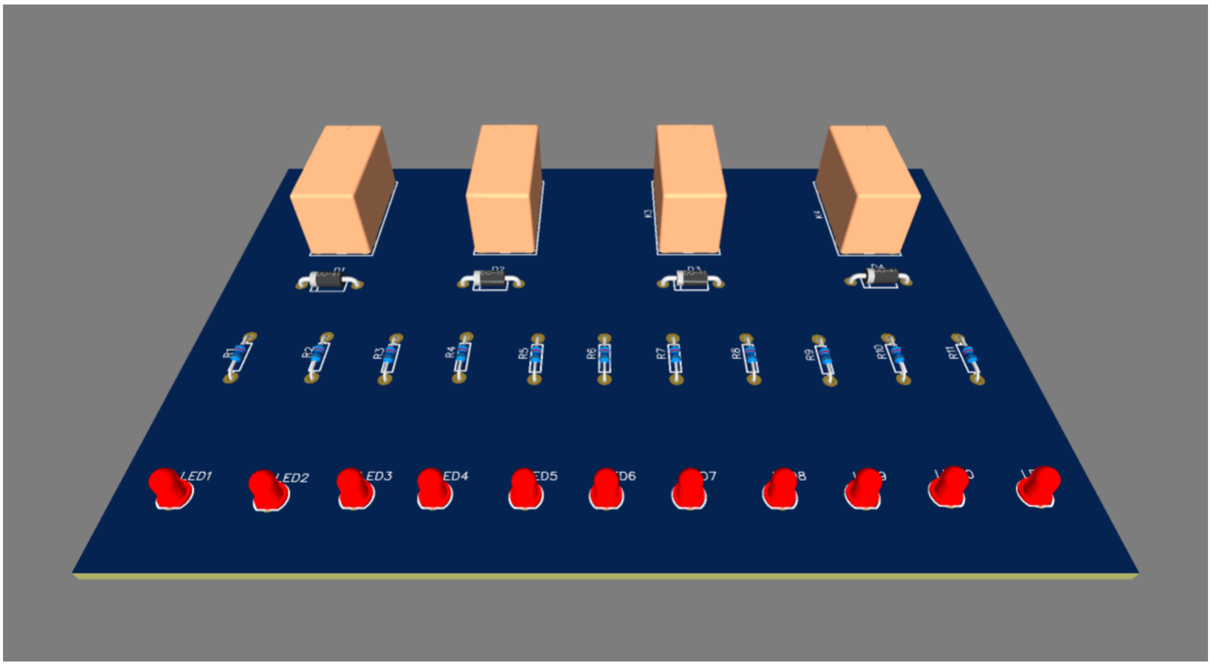


Routing Traces:



Final Review and Gerber Generation:

Final PCB Layout:



PCB DESIGNING ON EasyEDA TOOLS:

a. Describe the Process of Designing the PCB Layout Based on the Circuit Layout

Step 1: Convert Schematic to PCB

- ➔ Complete the Schematic
- ➔ Convert to PCB

Step 2: Define the Board Outline

- ➔ Draw the Board Outline

Step 3: Place Components

- ➔ Initial Placement
- ➔ Optimise Placement
    - IC Positioning
    - Input and Output Placement

- Power and Ground

Step 4: Route Traces

- ➔ Manual Routing
- ➔ Ground Plane
- ➔ Optimise Traces

Step 5: Add Silkscreen and Final Touches

- ➔ Component Labels
- ➔ Annotations
- ➔ Design Rule Check (DRC)

Step 6: Generate Gerber Files

- ➔ Generate Gerber Files
- ➔ Review Gerber Files

b. Placing Components, Routing Traces, and Optimising the PCB Layout

- ➔ Placing Components:
    - Logical Grouping
    - Accessibility
    - Spacing
- ➔ Routing Traces:
    - Manual Routing
    - Shortest Path
    - Avoid Crosstalk

VERIFICATION OF THE FINAL DESIGN

a. Steps Taken to Verify the Final PCB Design

- Electrical Connectivity Check
- Design Rule Check (DRC)

- Additional Verification Procedures

b. Checks for Verification

- Electrical Connectivity
- Design Rule Compliance
- Additional Verification Procedures

DOWNLOAD THE GERBER FILE:

Step 1: Complete the PCB Layout

- ➔ Open Your Project

Step 2: Generate Gerber Files

- ➔ Open Gerber Generation Tool
- ➔ Configure Gerber Settings
- ➔ Generate Gerber Files

Step 3: Review Gerber Files

- ➔ View Gerber Files

Step 4: Download Gerber Files

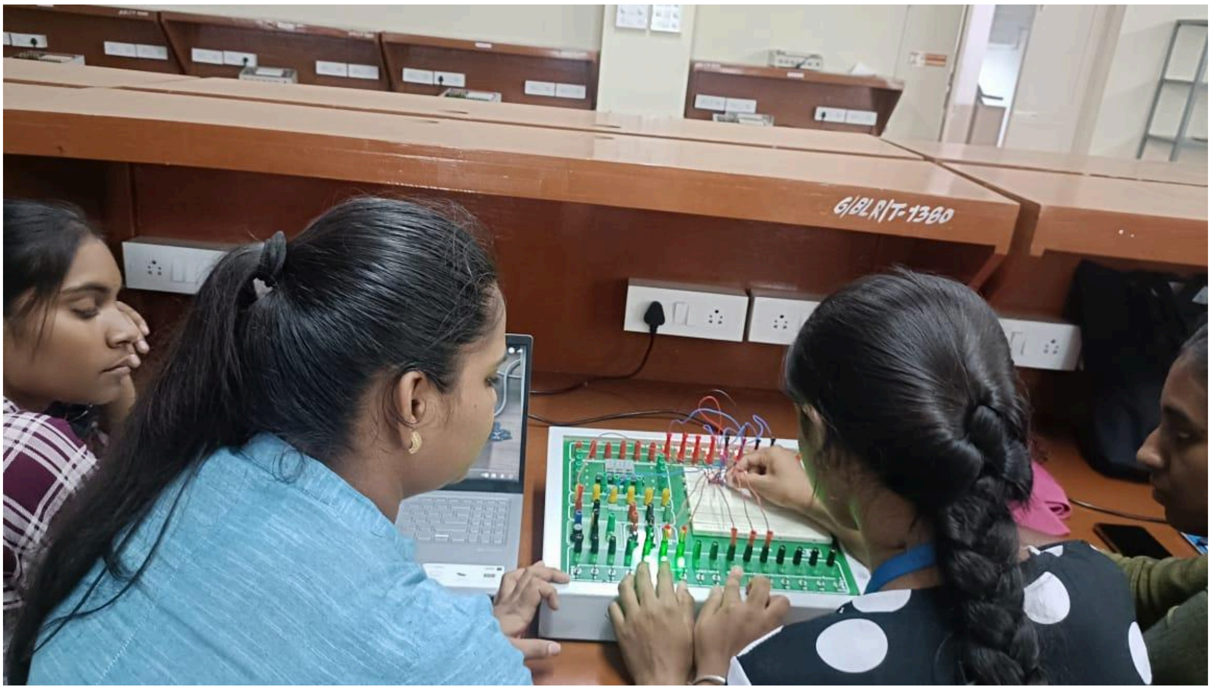
- ➔ Download Gerber Files
- ➔ Save the ZIP File

Step 5: Submit Gerber Files to a PCB Fabrication Service

- ➔ Select a PCB Fabrication Service
- ➔ Upload Gerber Files
- ➔ Review and Place Order

APPENDIX:

This project involved the design, simulation, and verification of a 3-to-8 decoder using Tinkercad and EasyEDA. Through careful planning, verification, and adherence to best practices, the final PCB design is ready for manufacturing. Following the guidelines and steps provided ensures a reliable and functional PCB that meets design specifications.



THANK YOU