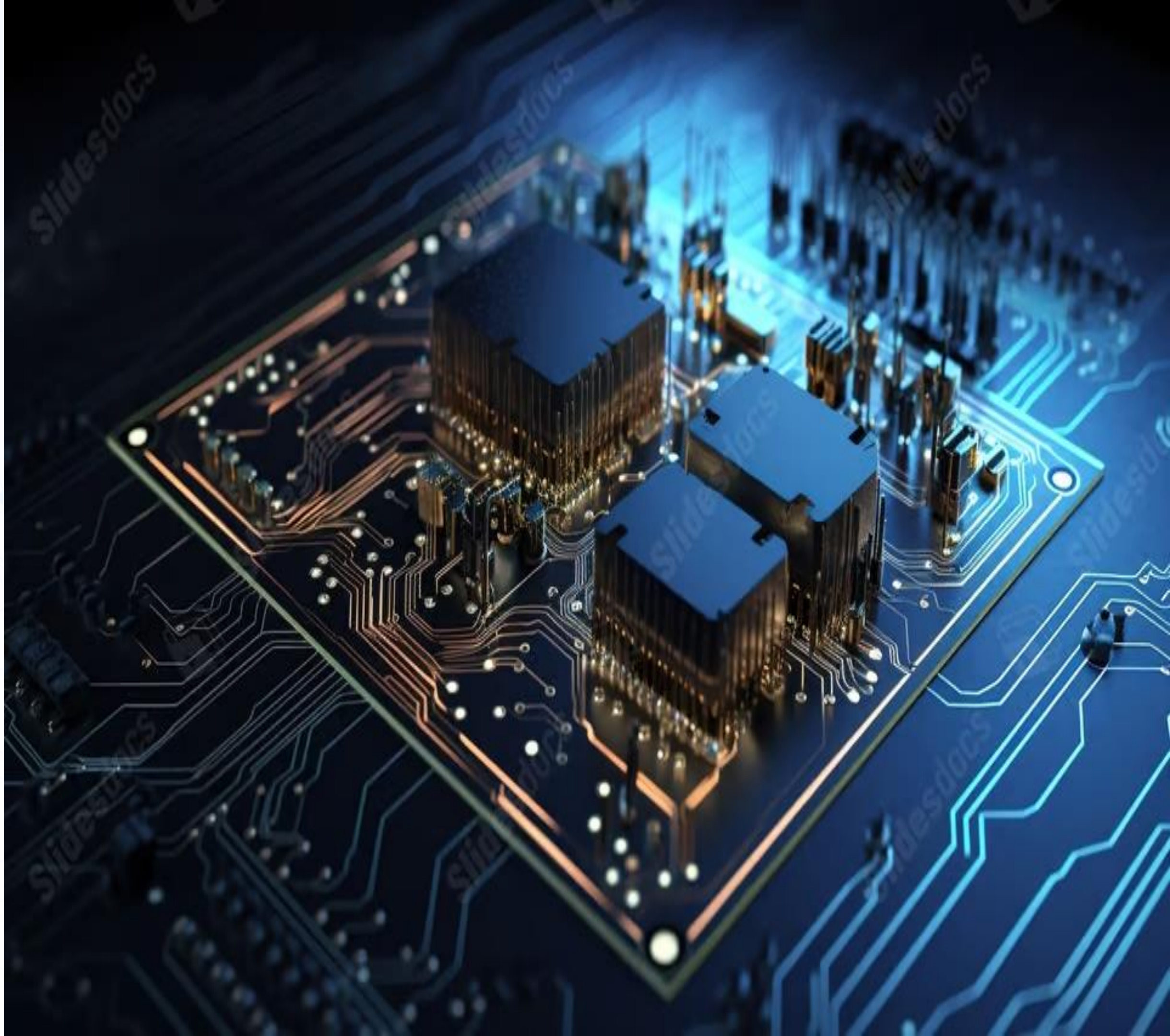
# R's Complement Calculator

R. Apoorva
K. Sravya
V. Srinath
F. Rahman



# Introduction

In digital systems, complements like 1's, 2's, 9's, and 10's are fundamental for arithmetic operations, error detection, and data representation.

Complement systems simplify the process of binary subtraction and are crucial in handling negative numbers in computing.

## 1’s Complement

**Definition:** The 1's complement of a binary number is obtained by inverting all the bits (i.e., changing all 0s to 1s and all 1s to 0s).

**Use Case:** Commonly used in digital systems to simplify the subtraction of binary numbers.

## 2’s Complement

**Definition:** The 2's complement is obtained by adding 1 to the 1's complement of a binary number.

**Use Case:** Widely used in computer systems to represent negative numbers and perform binary subtraction.

## 9’s Complement

**Definition:** For a decimal number, the 9's complement is found by subtracting each digit from 9.

**Use Case:** Used in arithmetic operations like subtraction and in certain error detection methods.

## 10’s Complement

**Definition:** The 10's complement is obtained by adding 1 to the 9's complement of a decimal number.

**Use Case:** Used in similar contexts as the 9's complement but provides a method to directly subtract numbers by adding complements.

# Objective

Goals

### Design and Implementation:

Develop a real-time calculator for generating 1's, 2's, 9's, and 10's complements for a four-bit binary input.

### User Interaction:

Enable users to input a decimal number through a keypad and select the desired complement type.

### Output Display:

Display the calculated complement using LEDs for easy and immediate feedback.

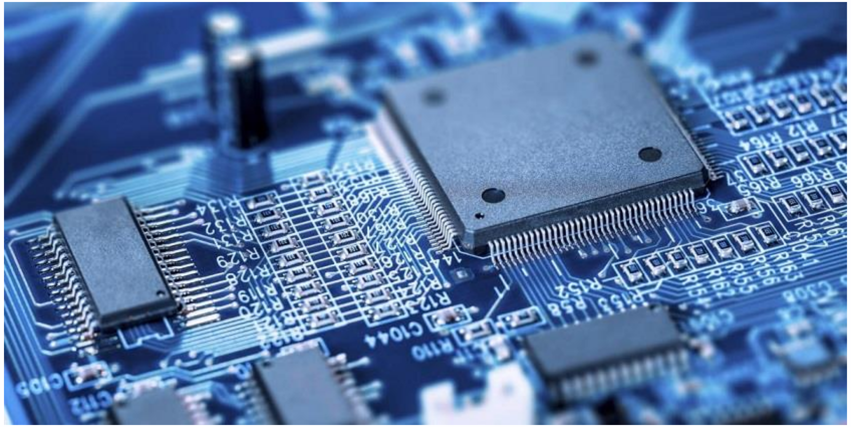
Vision

### Efficiency in Digital Systems:

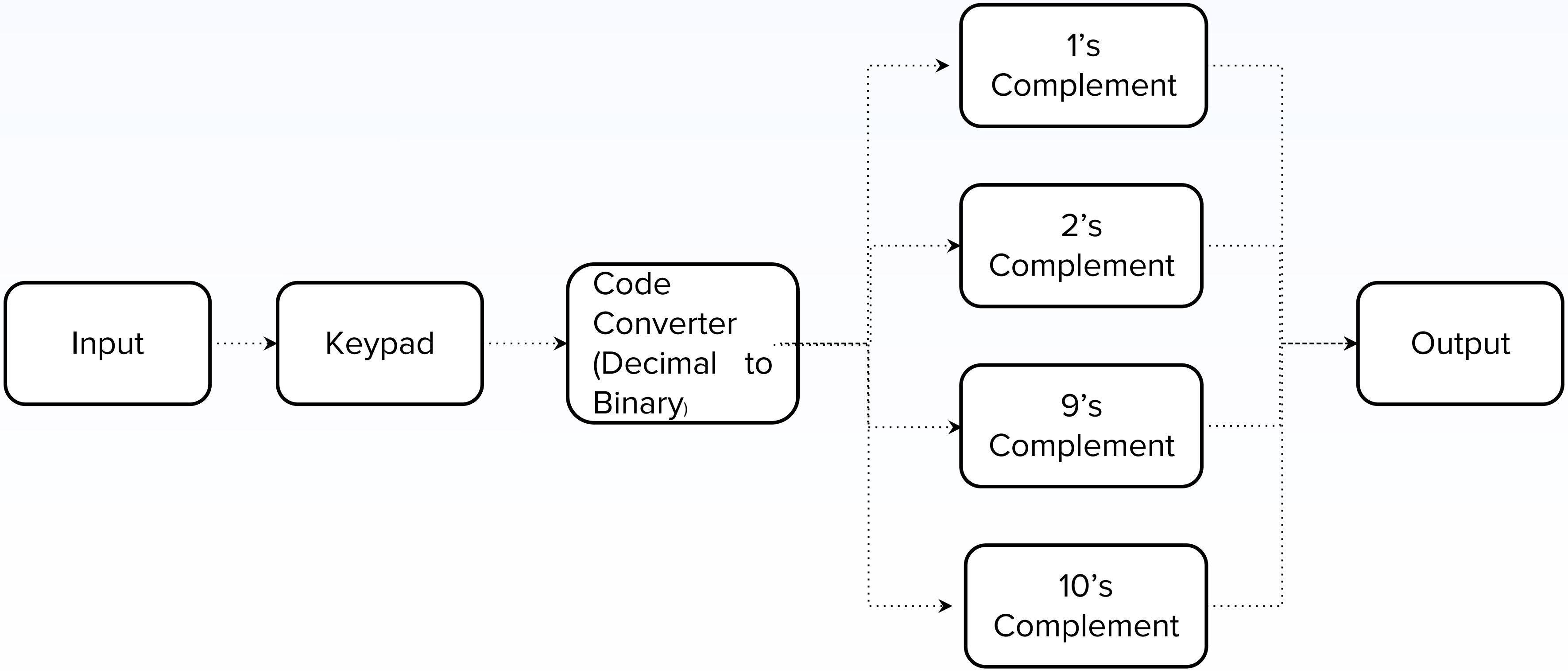
Simplify and automate the process of complement calculation, reducing manual effort and errors in digital arithmetic operations.

### Educational Tool:

Provide a practical and accessible tool for students and professionals working with digital logic, computer architecture, and binary arithmetic.



# Block Diagram



# Component Details

## Keypad (4x4)

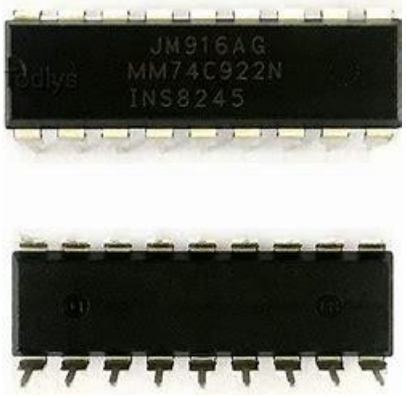
**Function:** It takes input from the user in decimal form and send it to the IC74C922

**Role:** Sends the decimal data input to the code converter

## IC74C922 - Code Converter

**Function:** Converts the decimal input from the keypad into a 4-bit binary code.

**Role:** Acts as the intermediary between user input and the complement circuits.

## IC7411 - 3-input AND Gate (9's Complement)

**Function:** Part of the logic that generates the most significant bit (MSB) of the 9's complement.

**Role:** Contributes to the combination of outputs needed to form the 9's complement.

## IC7404 - NOT Gate (1's Complement)

**Function:** Inverts each bit of the 4-bit binary input to generate the 1's complement.

**Role:** Simple bit inversion for the first step in complement generation.

## IC7486 - XOR Gate (9's Complement)

**Function:** Combines with other gates to produce the 9's complement of the input binary number.

**Role:** Essential for the accurate generation of the 9's complement.

## IC7483 - 4-bit Binary Adder (2's and 10's Complement)

**Function:** Adds 1 to the 1's complement to form the 2's complement.

- Adds 1 to the 9's complement to form the 10's complement.

**Role:** Handles binary addition essential for generating 2's and 10's complements.

# Working Principle

Step 1: User Input

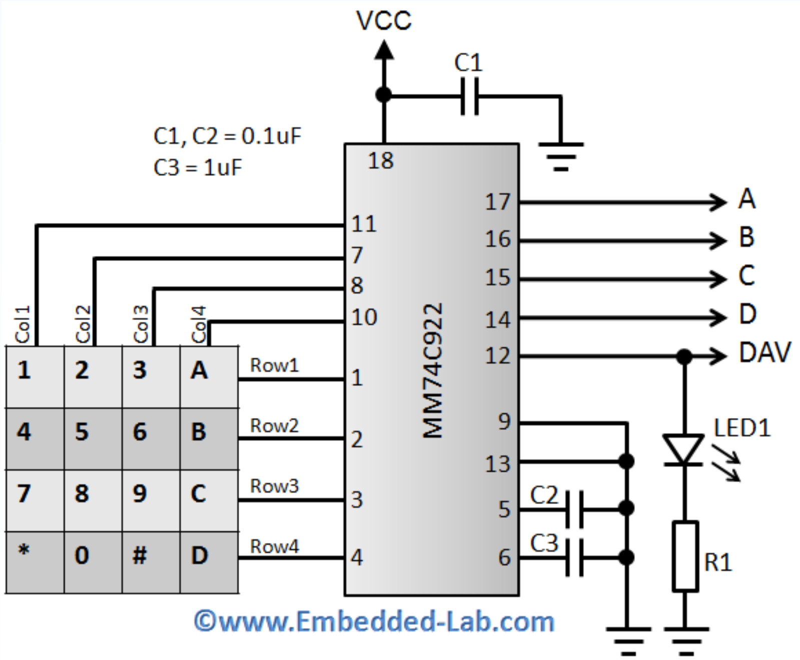
Input Method: The user enters a decimal number (0 to 9) using the 4x4 Keypad.

Processing: The input is received by the system for conversion.

Step 2: Decimal to Binary Conversion

Code Converter: Converts the user-inputted decimal number into a 4-bit binary number.

Binary Output: This binary number is fed simultaneously into the different complement circuits.



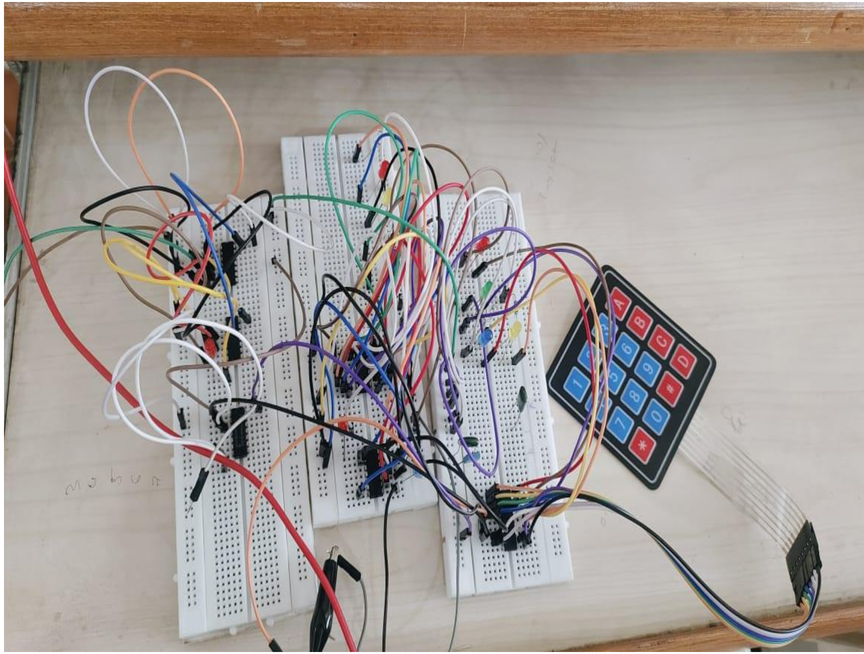
Step 3: Complement Calculation

**1’s Complement Circuit:** Inverts the binary bits to generate the 1’s complement.

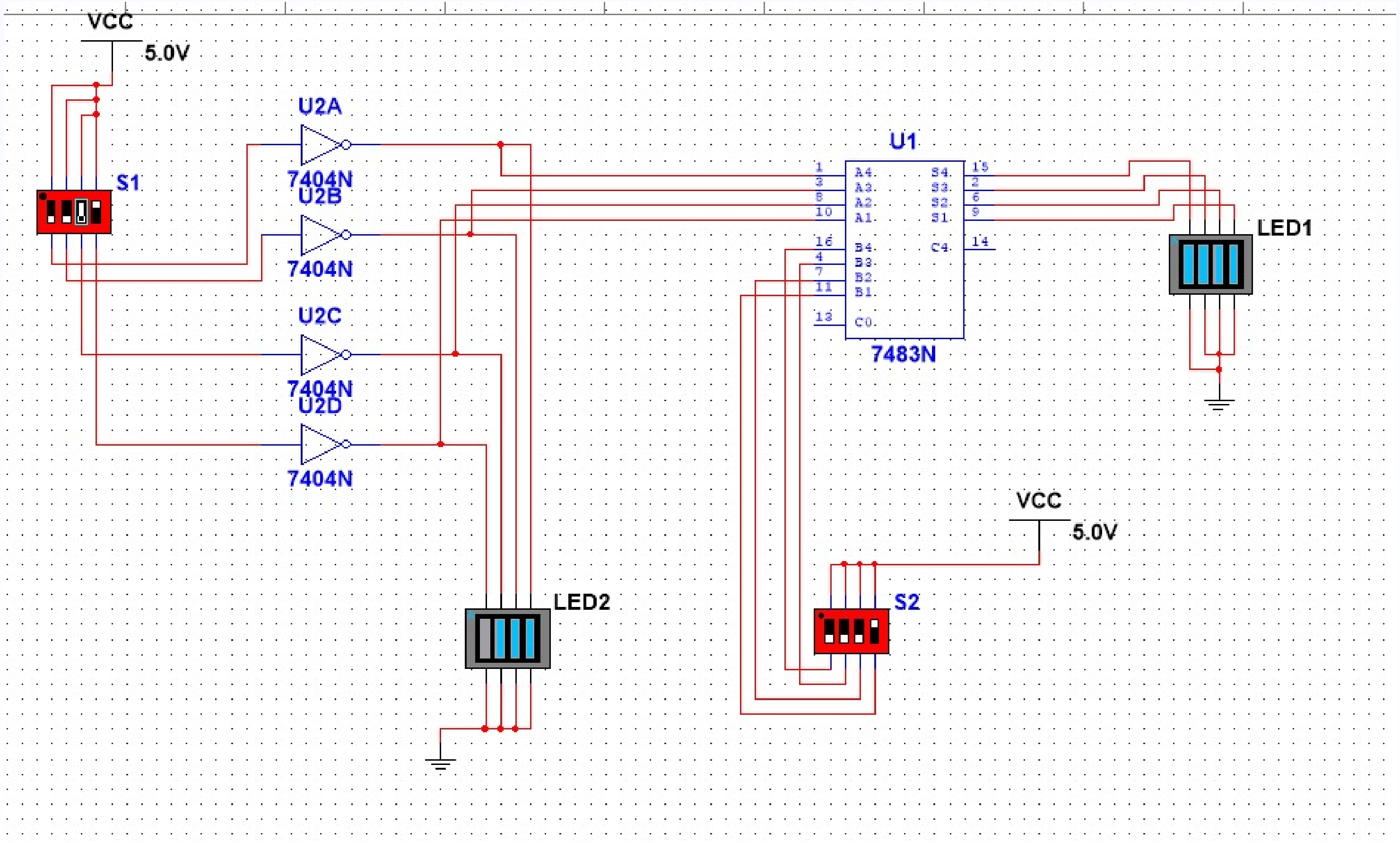
**2’s Complement Circuit:** Adds 1 to the 1’s complement using a binary adder to generate the 2’s complement.

**9’s Complement Circuit:** Generates the 9’s complement by specific logic operations using NOT, AND, and XOR gates.

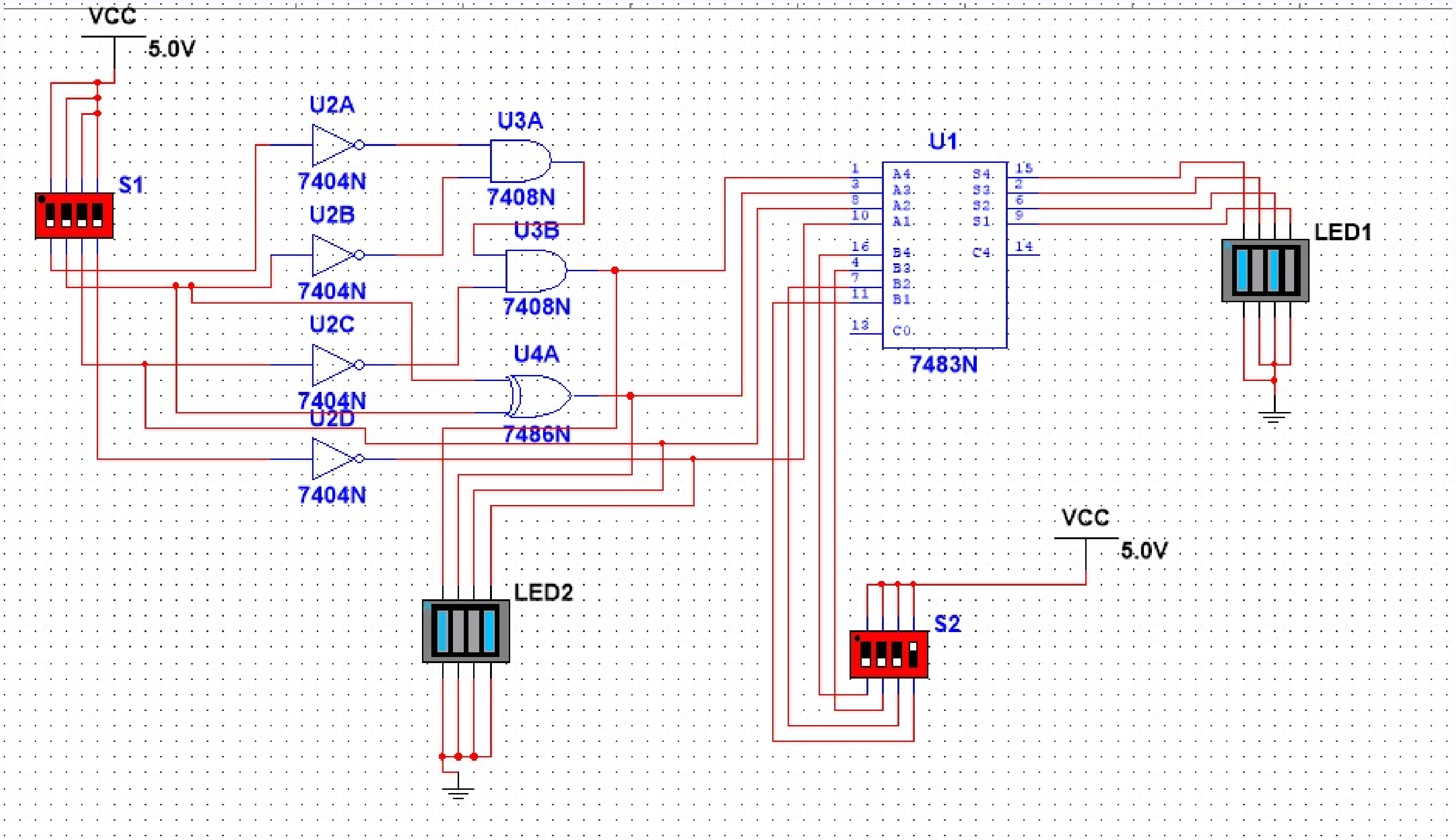
**10’s Complement Circuit:** Adds 1 to the 9’s complement to generate the 10’s complement.



# 1’s and 2’s Complement Calculator Diagram



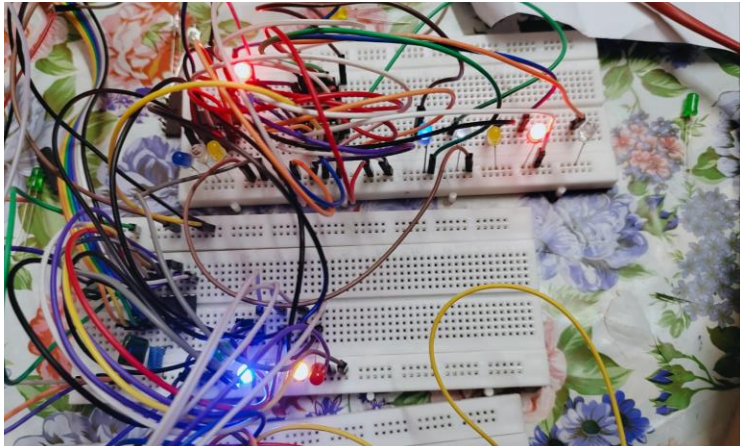
# 9’s and 10’s Complement Calculator Diagram

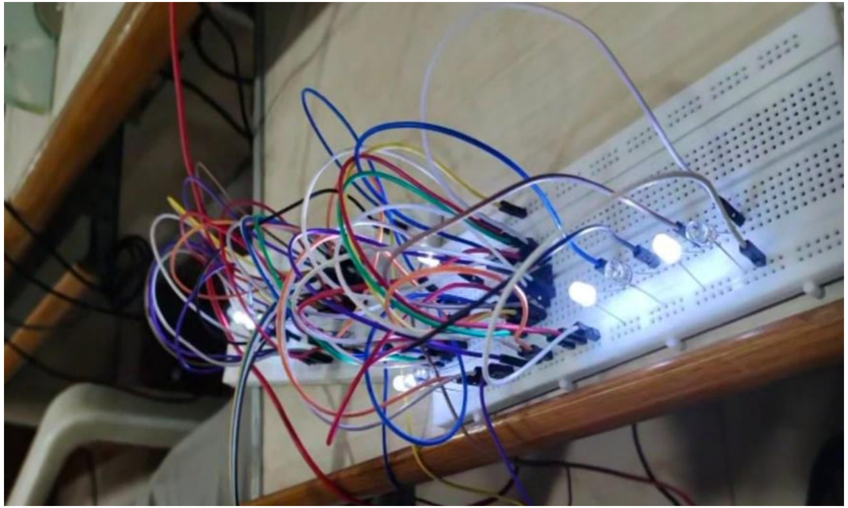


## Step 4: Output Display

LEDs: The calculated complement (1’s, 2’s, 9’s, or 10’s) is displayed via LEDs.

User Feedback: The corresponding LEDs pattern gives a visual indication of the calculated complement.





# Inputs and Outputs of Circuit:

## Practical and Theoretical Observation Table

### OBSERVATION TABLE

| Keypad I/P | Keypad O/P | 1's Complement | 2's Complement | 9's Complement | 10's Complement |
|---|---|---|---|---|---|
| 0 | 0000 | 1111 | 1000 | 1001 | 0000 |
| 1 | 0001 | 1110 | 1111 | 1000 | 1001 |
| 2 | 0010 | 1101 | 1110 | 0111 | 1000 |
| 3 | 0011 | 1100 | 1101 | 0110 | 0111 |
| 4 | 0100 | 1011 | 1100 | 0101 | 0110 |
| 5 | 0101 | 1010 | 1011 | 0100 | 0101 |
| 6 | 0110 | 1001 | 1010 | 0011 | 0100 |
| 7 | 0111 | 1000 | 1001 | 0010 | 0011 |
| 8 | 1000 | 0111 | 1000 | 0001 | 0010 |
| 9 | 1001 | 0110 | 0111 | 0000 | 0001 |

# Application & Limitations

### Applications

**Binary Arithmetic:** Simplifies subtraction and other arithmetic operations in digital systems using complement methods.

**Error Detection:** Assists in error detection and correction algorithms by leveraging complement operations to verify data integrity.

**Digital Circuit Design:** Valuable in designing and simulating digital circuits, particularly in arithmetic units of processors.

**Programming and System Design:** Supports low-level programming tasks, especially where efficient arithmetic operations are needed in embedded systems.

**Education:** Useful as a teaching tool for students learning about binary arithmetic, computer architecture, and digital logic design.

## Limitations

**Input Range:** Limited to decimal inputs from 0 to 9, restricting the scope of use for larger or more complex numbers.

**Complement Types:** Supports only 1’s, 2’s, 9’s, and 10’s complements, not other types of complements that might be needed in certain applications.

**Binary Length:** Restricted to 4-bit binary numbers, which may not suffice for all digital arithmetic tasks.

**Circuit Complexity:** The project requires multiple ICs and connections, leading to a relatively complex circuit design.

**Power Requirements:** Needs multiple DC supply voltages, adding to the operational complexity.

# Conclusion

## Summary

**Successful Implementation:** Developed a real-time R's complement calculator that accurately computes 1's, 2's, 9's, and 10's complements for 4-bit binary numbers.

**Educational Value:** The project demonstrates key digital logic principles and provides a practical tool for learning and teaching binary arithmetic.

## Importance of R's Complement

**Digital Arithmetic:** R's complement methods are essential in simplifying subtraction and handling negative numbers in digital systems, highlighting their significance in modern computing.

## Future Enhancement

**Expanded Input Range:** Potential to support a wider range of decimal inputs and larger binary numbers.

**Additional Features:** Could include support for other types of complements or more complex arithmetic operations.

**User Interface:** Improving the user interface for more accessible operation, such as adding a digital display or expanding input methods.