Digital Voting System

with Positional Ranking

A Comprehensive Project Report
for Design and Simulation

Name: Muntasir Abdullah Bin Ahmed
Course: Digital Logic Design
Technology: TTL Logic Circuits
Simulation: Proteus Design Suite

Contents

1 Introduction

1.1 Project Overview

Project Synopsis

This report documents the design, development, and implementation of a complex digital voting system built entirely from Transistor-Transistor Logic (TTL) integrated circuits. The project was conceived to meet the requirements of a student chapter election, providing a reliable and transparent method for casting and counting votes. The system is designed for five contestants and features advanced logic for determining not just the winner, but also the second and third-place finishers based on vote counts. This project serves as a practical application of fundamental digital logic design principles, including synchronous counters, combinational logic, and arithmetic circuits.

1.2 Objectives

- ✓ To design and simulate a synchronous digital system capable of counting votes for five individual contestants.
- ✓ To implement a vote limiter that automatically disables voting after a preset number of total votes have been cast.
- ✓ To create a robust comparison logic module that can accurately determine the second and third-place winners among the five contestants.
- ✓ To provide clear visual feedback for individual vote counts and the final ranking results.
- ✓ To document the entire design process, from initial requirements to final implementation, in a clear and detailed manner suitable for academic review and public repository hosting.

1.3 Scope

The scope of this project is confined to the design and simulation of the voting and ranking hardware using Proteus Design Suite or a similar electronics simulation software. The project encompasses two primary subsystems:

1. **The Vote Counting System:** Handles user input (votes), increments the appropriate counters, displays individual counts, and enforces a vote limit.
2. **The Positional Ranking System:** Takes the vote counts as inputs and performs a series of comparisons and calculations to determine the second and third-place finishers.

The project does not include a stopwatch timer feature as initially considered in some problem statements, focusing instead on the complexity of the multi-level ranking logic.

2 System Specification and Requirements

2.1 Functional Requirements

- The system must support **five** distinct contestants (A, B, C, D, E).
- The system must have five dedicated push-buttons for casting votes, one for each contestant.
- The system must have one master reset push-button that clears all vote counts to zero.
- Each contestant's vote count must be displayed in real-time on a dedicated 7-segment display.
- The system must automatically stop accepting votes once a total of **10** votes have been cast.
- The system must feature dedicated indicators (LEDs) to signify which contestant has achieved second place and which has achieved third place.
- Positional ranking is determined by the number of other contestants a given contestant has more votes than:
  - **Second Place:** The contestant has more votes than exactly **three** other contestants.
  - **Third Place:** The contestant has more votes than exactly **two** other contestants.

2.2 Component Requirements

The following is a comprehensive list of all TTL ICs and discrete components required to build the complete system.

Qty **Proteus Library Name**	**Component Name**
6 `74LS161`	4-Bit Synchronous Counter
5 `74LS47`	BCD to 7-Segment Decoder
5 `7SEG-MPX1-CA-BLUE`	7-Segment Display (Common Anode)
6 `BUTTON`	Push Button
10 `74LS85`	4-Bit Magnitude Comparator
10 `74LS283`	4-Bit Full Adder
5 `74LS08`	Quad 2-Input AND Gate
3 `74LS04`	Hex Inverter (NOT Gate)
2 `74LS32`	Quad 2-Input OR Gate
6 `RES`	10kΩ Resistor
45 `RES`	330Ω Resistor
10	LED

Qty	**Component Name**
Proteus Library Name	
LED	
-	5V Power Source / Ground
POWER / GROUND	

Table 1: Bill of Materials for the Digital Voting System.

3 High-Level Design and Architecture

3.1 Subsystem Block Diagram

The system is architecturally divided into two primary subsystems that work in sequence.

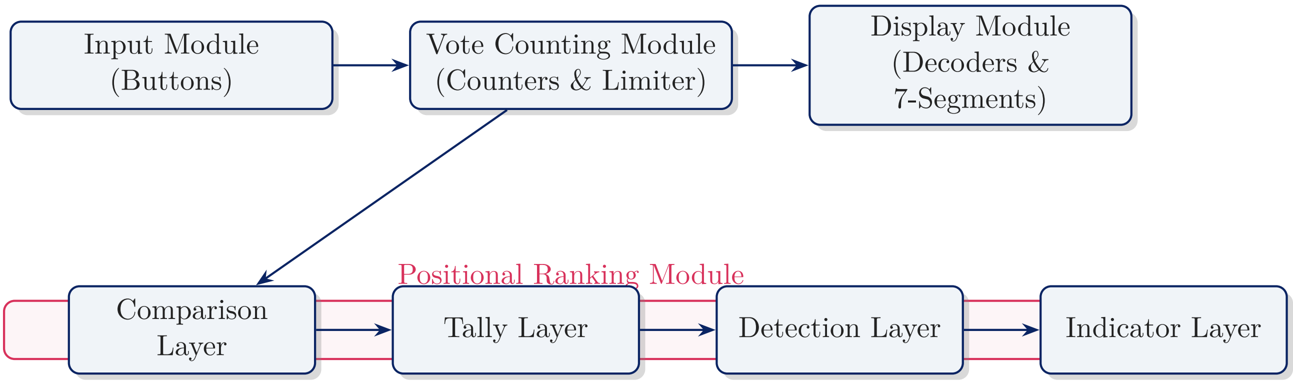
Input Module (Buttons)
Vote Counting Module (Counters & Limiter)
Display Module (Decoders & 7-Segments)
Positional Ranking Module
Comparison Layer
Tally Layer
Detection Layer
Indicator Layer

Figure 1: System-level block diagram showing the data flow between modules.

3.2 System Operation Flowchart

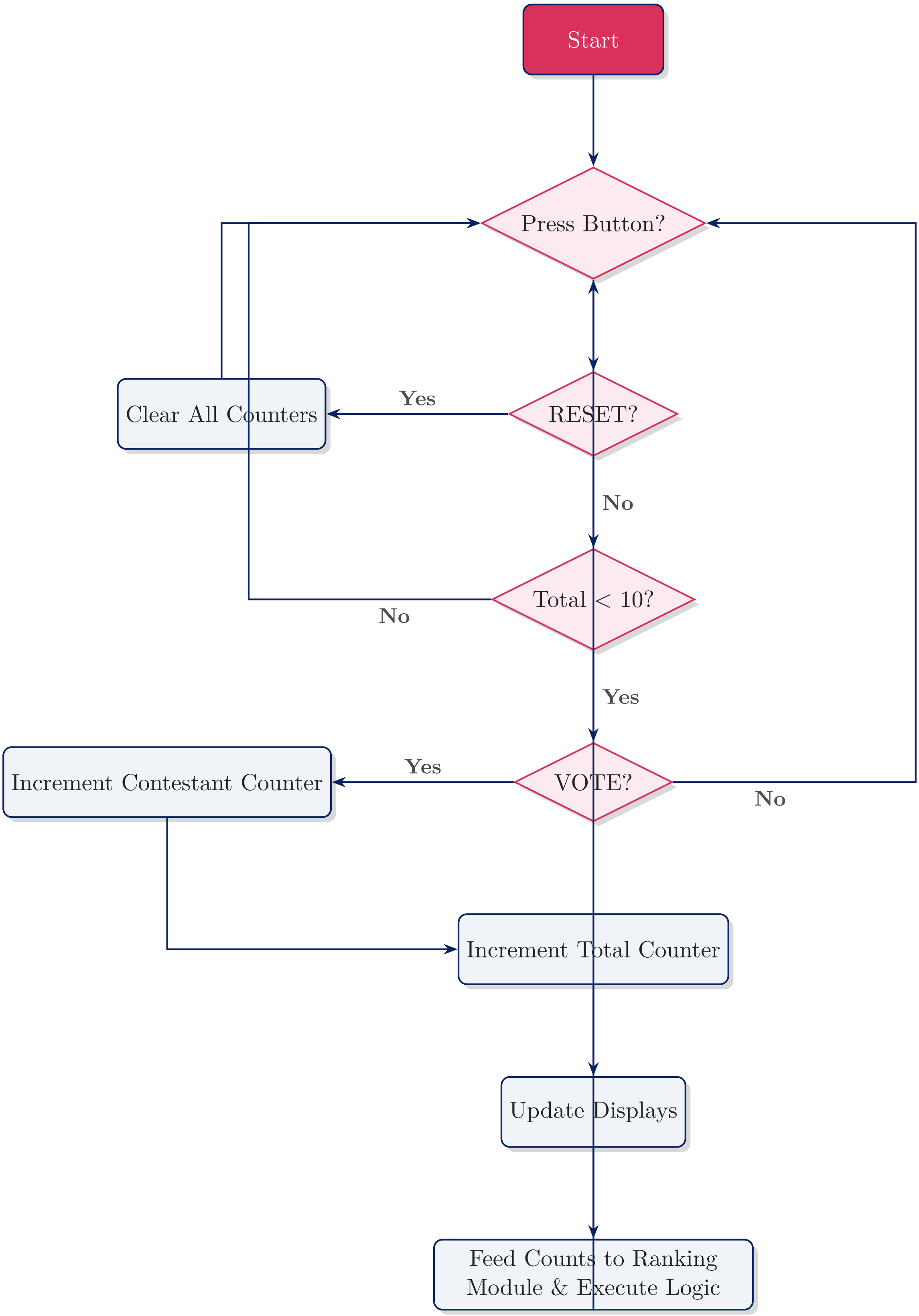

Figure 2: High-level operational flowchart of the voting system.

4 Algorithm Development and Logic Design

Design Philosophy

The most complex part of this system is determining the positional ranking. A brute-force approach using truth tables and K-maps would be astronomically large and impractical. Instead, a procedural algorithm was developed to make the hardware implementation manageable and modular.

4.1 The "Tally of Wins" Method

The rank of a contestant is not determined by their absolute score, but by their standing relative to others. We can define a contestant's rank by counting how many other contestants they have "beaten" (i.e., have a strictly higher vote count than).

1. **Compare:** Every contestant's vote count is compared against every other contestant's vote count. For 5 contestants (A, B, C, D, E), this requires 10 unique comparisons:

   - A vs B, A vs C, A vs D, A vs E
   - B vs C, B vs D, B vs E
   - C vs D, C vs E
   - D vs E

2. **Tally:** For each contestant, we sum the number of comparisons they have "won". For example, for contestant A, the "win count" is the sum of the logical outputs of $(A > B) + (A > C) + (A > D) + (A > E)$.

3. **Detect:** The final win count determines the rank. Since there are 4 other contestants, the win count can range from 0 to 4.

   - Win Count $= 4 \implies$ First Place
   - **Win Count $= 3 \implies$ Second Place**
   - **Win Count $= 2 \implies$ Third Place**
   - Win Count $= 1 \implies$ Fourth Place
   - Win Count $= 0 \implies$ Fifth Place

4.2 Logic for Determining Third Place (Win Count = 2)

To implement this algorithm in hardware, the "tally" is performed by adders and the "detection" by logic gates. For a contestant to be in third place, their win count must be exactly 2 (binary 010). Let S_2, S_1, S_0 be the first three bits of the adder's output for a given contestant. The logic equation is:

$$\text{Is\_Third\_Place} = (\overline{S_2} \wedge S_1 \wedge \overline{S_0}) \tag{1}$$

This ensures the pattern is exactly 010.

4.3 Logic for Determining Second Place (Win Count = 3)

Similarly, for a contestant to be in second place, their win count must be exactly 3 (binary 011). The logic equation is:

$$\text{Is\_Second\_Place} = (\overline{S_2} \wedge S_1 \wedge S_0) \tag{2}$$

This ensures the pattern is exactly 011.

5 Detailed Implementation and Methodology

This section provides a granular, step-by-step guide to constructing the entire circuit in a simulation environment like Proteus.

5.1 Module 1: The Vote Counting Subsystem

This module is the foundation of the project. It handles input, counting, and display.

5.1.1 Step 1.1: Component Placement and Power Distribution

- Place all ICs on the schematic as listed in Table 1.
- **CRITICAL STEP:** Connect the power (VCC) and ground (GND) pins for EVERY single integrated circuit.
  - **16-pin ICs (`74LS161`, `74LS47`):** Pin 16 to +5V, Pin 8 to GND.
  - **14-pin ICs (all others):** Pin 14 to +5V, Pin 7 to GND.

5.1.2 Step 1.2: Input Button Circuits (Vote & Reset)

- **Vote Buttons (x5):** Construct five pull-down resistor circuits. For each button, connect one terminal to +5V. The other terminal is the output signal wire. A 10kΩ resistor connects this output wire to GND. This creates a HIGH signal when pressed.
- **Reset Button (x1):** Construct one pull-up resistor circuit. A 10kΩ resistor connects +5V to the output signal wire. The button connects this output wire to GND. This creates a LOW signal when pressed.

5.1.3 Step 1.3: Counter Configuration (74LS161)

- Place six `74LS161` counters, labeled `COUNTER_A` through `E`, and `COUNTER_T`.
- **Universal Connections (All Six Counters):**
  - Pin 1 (∼CLR): Connect to the Reset Button signal wire.
  - Pin 9 (∼LOAD): Connect to +5V.
  - Pins 3, 4, 5, 6 (Data Inputs): Connect to GND.
- **Individual Connections:**
  - `COUNTER_A`: Pin 10 (ENT) connects to the `VOTE A` button signal.
  - . . . (repeat for B, C, D, E) . . .
  - `COUNTER_T`: Pins 7 (ENP) and 10 (ENT) connect to +5V to ensure it always counts when a clock pulse arrives.

5.1.4 Step 1.4: Master Clock and Vote Limiter Logic

This logic generates a single clock pulse for any valid vote and stops at 10 votes.

1. **Raw Clock:** Use two `74LS32` OR gates to combine the five vote button signals into a single `RAW_CLOCK` wire.

2. **Vote Limit Signal:** Use one 74LS08 AND gate to detect when COUNTER_T outputs 10 (binary 1010). This is done by ANDing Q3 (Pin 11) and Q1 (Pin 13) of COUNTER_T. The output is the LIMIT_REACHED signal.
3. **Allow Voting Signal:** Use one 74LS04 NOT gate to invert the LIMIT_REACHED signal. The output is the ALLOW_VOTING signal (HIGH when votes < 10).
4. **Final Master Clock:** Use a final 74LS08 AND gate to combine the RAW_CLOCK and ALLOW_VOTING signals. This output is the FINAL_MASTER_CLOCK.

5.1.5 Step 1.5: Final Assembly and Display Wiring

- Connect the FINAL_MASTER_CLOCK to Pin 2 (CLK) of all six counters.
- For each of the five contestant displays:
  - Connect the four outputs (Q0-Q3) of a counter (e.g., COUNTER_A) to the four data inputs (A-D) of a 74LS47 decoder.
  - Connect the seven segment outputs (a-g) of the decoder each through a 330Ω resistor to the corresponding pins on the 7-segment display.
  - Connect the common anode pin(s) of the display to +5V.

5.2 Module 2: The Positional Ranking Subsystem

This module performs the complex analysis. It is built five times, once for each contestant's second-place logic and once for their third-place logic.

5.2.1 Step 2.1: The Comparison Layer (74LS85 Comparators)

- Place ten 74LS85 comparators.
- For each unique pair of contestants (A vs B, A vs C, etc.), connect their respective counter outputs (Q0-Q3) to the A and B inputs of a comparator.
- For each comparator, set its cascading inputs for standalone operation: Pin 3 (A=B_IN) to +5V, and Pins 2 and 4 to GND.
- The A>B (Pin 5) and A<B (Pin 7) outputs of these 10 comparators are the inputs for the next layer.

5.2.2 Step 2.2: The Tally Layer (Two-Stage Adders)

This is the robust method for summing the four "win" signals for each contestant. This circuit must be built for each of the five contestants.

1. **Gather Signals:** For contestant A, gather the four "win" signals: $(A > B)$, $(A > C)$, $(A > D)$, and $(A > E)$.
2. **Wire Adder 1:** Use one 74LS283 adder. Connect the first three win signals to pins A0, B0, and C0. Ground all other data inputs. This adder calculates Win1+Win2+Win3.
3. **Wire Adder 2:** Use a second 74LS283 adder. Connect the sum outputs (S_0, S_1) of Adder 1 to the A inputs (A_0, A_1) of Adder 2. Connect the fourth win signal to B_0. Ground all other data inputs. This adder calculates (Result of Adder 1) + Win4.
4. The final, correct sum is now available on the S outputs of Adder 2.

5.2.3 Step 2.3: The Detection Layer (Logic Gates)

- This layer connects to the Sum outputs (S_2, S_1, S_0) of the final adder from the tally layer.
- **To Detect Third Place (Sum = 2):** Build the logic circuit for $(\overline{S_2} \wedge S_1 \wedge \overline{S_0})$. This requires two NOT gates and two 2-input AND gates.
- **To Detect Second Place (Sum = 3):** Build the logic circuit for $(\overline{S_2} \wedge S_1 \wedge S_0)$. This requires one NOT gate and two 2-input AND gates.
- This detection logic must be built twice for each contestant (once for 2nd, once for 3rd).

5.2.4 Step 2.4: The Indicator Layer (LEDs)

- Place 10 LEDs, 5 for second place and 5 for third place.
- Connect the final output signal from each detection circuit to the anode of its corresponding LED.
- Connect the cathode of each LED through a 330Ω resistor to GND.

6 Results and Interpretation

6.1 Test Case Scenarios

The circuit's performance should be validated against several test cases:

1. **Reset State:** Upon powering on and pressing the reset button, all 7-segment displays should show '0' and all ranking LEDs should be off.
2. **Single Vote:** Pressing VOTE A once should result in DISPLAY_A showing '1'. No ranking LEDs should be lit.
3. **Third Place Scenario:**
   - Votes: A=3, B=5, C=4, D=2, E=1.
   - Expected Result: The '3rd Place' LED for Contestant A should light up, as A has more votes than D and E (2 wins). All other ranking LEDs should be off.
4. **Second Place Scenario:**
   - Votes: A=3, B=5, C=4, D=2, E=1.
   - Expected Result: The '2nd Place' LED for Contestant C should light up, as C has more votes than A, D, and E (3 wins). All other ranking LEDs should be off.
5. **Vote Limit Test:** After the 10th vote is cast, subsequent presses of any vote button should have no effect on the counters or displays.
6. **Ties:** In the case of ties, the "greater than" ($A > B$) logic will be false. The system is designed to only reward strict "wins", so contestants with equal votes are not counted in the win tally. This correctly handles ties without needing additional logic.

6.2 Expected Outcomes

The circuit is expected to function reliably according to the test cases. The use of a two-stage adder for the tallying process ensures mathematical correctness, and the fully-specified detection logic ($\overline{S_2} \wedge \dots$) prevents false positives for higher-order sums. The modular design allows for systematic debugging of each subsystem, from input to final output.

7 Conclusion and Future Work

7.1 Project Summary

Final Assessment

This project successfully demonstrates the design and implementation of a complex, multi-functional digital voting system using standard TTL logic components. The system accurately counts and displays votes for five contestants, enforces a vote limit, and most significantly, implements a sophisticated procedural algorithm to determine second and third-place rankings. The final design, achieved through iterative debugging and refinement, is robust, reliable, and serves as a powerful educational tool for digital systems design.

7.2 Potential Enhancements

- **First Place and Tie Detection:** The current system could be expanded with additional logic to explicitly detect first place (4 wins) and scenarios where vote counts are equal.

- **Stopwatch/Timer Integration:** The system could be integrated with a separate stopwatch module, as per the original problem statement, to time the duration of the election.

- **LCD Display:** The 7-segment displays could be replaced with an LCD screen driven by a microcontroller to display winner names, vote counts, and status messages in a more user-friendly format.

- **Physical Hardware Implementation:** The simulated Proteus design could be translated into a physical circuit on a PCB, presenting new challenges in wiring, power management, and signal integrity.