# BCD Counter with T Flip-Flops

**EE24BTECH11002 EE24BTECH11024**

April 6, 2025

**Abstract**

This document details the design and implementation of a bidirectional BCD counter using T flip-flops. The counter, which counts from 0 to 9 (and vice versa), employs additional logic to skip invalid BCD states. The design includes detailed state tables, Karnaugh maps, and circuit diagrams.

## Aim

To design and implement a BCD (Binary Coded Decimal) counter using T flip-flops that can count both up and down based on input conditions, with a maximum count of 9 (0–9).

## Apparatus

- JK Flip-Flops (IC 7476) configured as T flip-flops (by tying J and K high) – 4 units
- Logic gates (AND, OR, XOR, NOT gates)
- Push buttons (2 units – one for up counting, one for down counting)
- 7-segment display with BCD to 7-segment decoder
- Power supply (5V DC)
- Breadboard and jumper wires
- Resistors (1kΩ, 330Ω)
- Arduino board (for clock generation and power)
- Oscilloscope (to verify clock signals)
- Logic analyzer (to verify state transitions)

## Theory

A BCD counter is a digital counter that counts from 0 (0000) to 9 (1001) in binary and then resets back to 0. In this experiment, we implement a bidirectional BCD counter using T flip-flops that can count both up and down based on input signals.

## T Flip-Flop

T flip-flops (Toggle flip-flops) change their output state when triggered by a clock pulse if their T input is 1. If $T = 0$, the output remains unchanged. This makes them ideal for counter applications.

The characteristic equation of a T flip-flop is:

$$Q^+ = Q \oplus T$$

where:

- $Q^+$ is the next state,
- $Q$ is the current state,
- $T$ is the toggle input,
- $\oplus$ represents the XOR operation.

A T flip-flop can be constructed from a JK flip-flop by connecting both J and K inputs to the same signal $T$. When $T = 1$, the flip-flop toggles on each clock pulse; when $T = 0$, it maintains its state.

## BCD Counter

A BCD counter counts from 0 to 9 in binary (0000 to 1001) and then resets to 0. Since a 4-bit binary counter can represent 16 states (0–15), additional logic is required to skip the invalid states (10–15). This counter is known as a modulo-10 counter.

## 7-Segment Display

The 7-segment display consists of seven LEDs arranged in a pattern to display decimal digits. A BCD-to-7-segment decoder converts the 4-bit BCD output from the counter into signals that illuminate the appropriate segments (labeled 'a' through 'g') to display digits 0–9.

# Derivation of Logic

Below are the state tables and Karnaugh map derivations for both the up and down counter modes.

## State Tables

**Up Counter State Table:**

| Present State | | | | T Flip-Flops | | | | Next State | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | $T_3$ | $T_2$ | $T_1$ | $T_0$ | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X | X | X | X | X |

Table 1: State table for the single-digit BCD Up-counter. (X denotes don't-care conditions.)

**Down Counter State Table:**

| Present State | | | | T Flip-Flops | | | | Next State | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | $T_3$ | $T_2$ | $T_1$ | $T_0$ | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X | X | X | X | X |

Table 2: State table for the single-digit BCD Down-counter. (X denotes don't-care conditions.)

# Karnaugh Maps for T Flip-Flops

## Up Counter T Flip-Flops

**For $T_0$:**
Since $T_0 = 1$ for all cases, no Karnaugh map is required.

**For $T_1$:**

| $Q_3Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 1 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | X | X | X | X |
| 10 | 0 | 0 | X | X |

Thus,

$$T_1 = \overline{Q_3}\, Q_0.$$

**For $T_2$:**

| $Q_3Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | X | X | X | X |
| 10 | 0 | 0 | X | X |

Thus,

$$T_2 = Q_1\, Q_0.$$

**For $T_3$:**

| $Q_3Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | X | X | X | X |
| 10 | 0 | 1 | X | X |

Thus,

$$T_3 = Q_2\,Q_1\,Q_0 + Q_3\,Q_0.$$

**Down Counter T Flip-Flops**

**For $T_0$:**
Again, $T_0 = 1$ for all cases.

**For $T_1$:**

| $Q_3Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 |
| 01 | 1 | 0 | 0 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 0 | X | X |

Thus,

$$T_1 = Q_1\,\overline{Q_0} + Q_2\,\overline{Q_1}\,\overline{Q_0} + Q_3\,\overline{Q_1}\,\overline{Q_0}.$$

**For $T_2$:**

| $Q_3Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | X | X | X | X |
| 10 | 1 | 0 | X | X |

Thus,

$$T_2 = Q_2\,\overline{Q_1}\,\overline{Q_0} + Q_3\,\overline{Q_1}\,\overline{Q_0}.$$

**For $T_3$:**

| $Q_3Q_2$ \ $Q_1Q_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | X | X | X | X |
| 10 | 1 | 0 | X | X |

Thus,

$$T_3 = \overline{Q_2}\,\overline{Q_1}\,\overline{Q_0}.$$

The Karnaugh maps for Down counter can be simplified further but the reason to leave them as give is, take $T = \overline{Q_1}\,\overline{Q_0}$ then:

1. $T_0 = 1$
2. $T_1 = Q_1\,\overline{Q_0} = T_2$
3. $T_2 = (Q_1 + Q_2)T$
4. $T_3 = \overline{Q_2}\,T$

Which are easier to make in Circuit

# Circuit Diagram

## Circuit Diagram for Incrementing

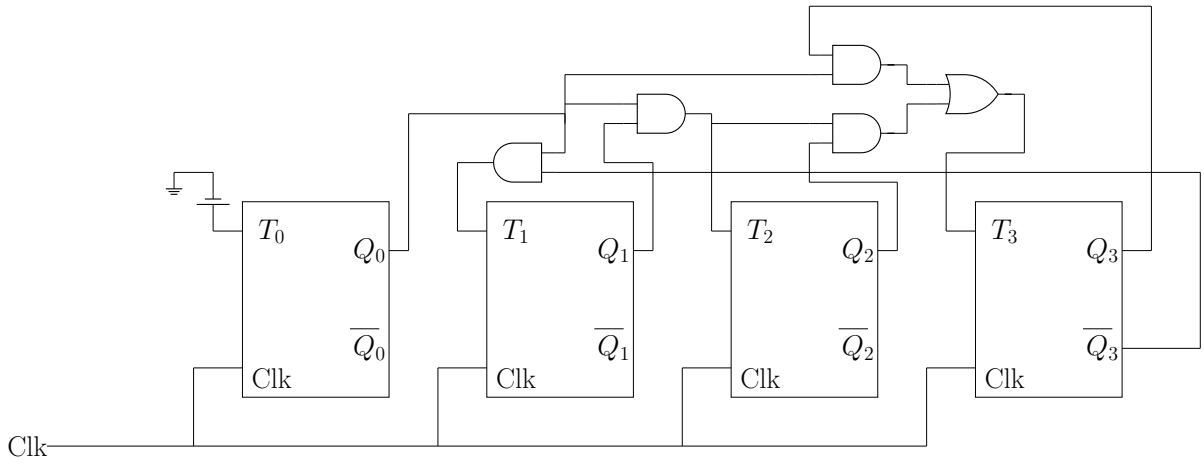



Figure 1: Circuit diagram representing the logic for incrementing.

Let the circuit be represented as:

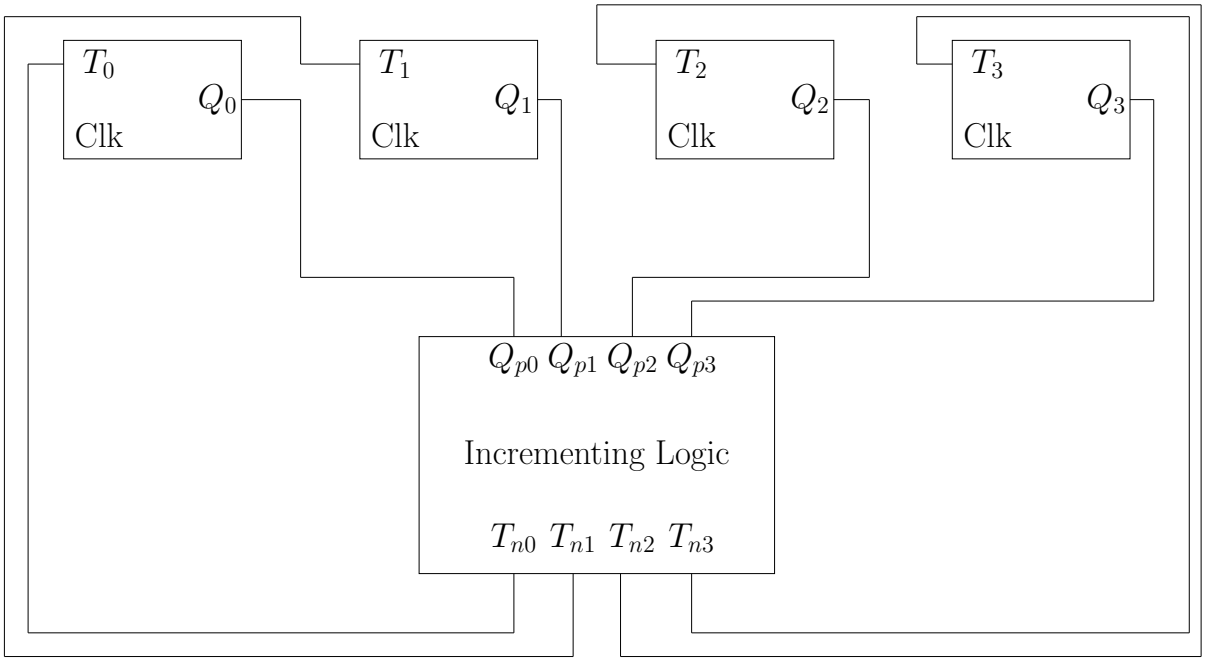



Figure 2: Simplified version of Circuit diagram

## Circuit Diagram for Decrementing

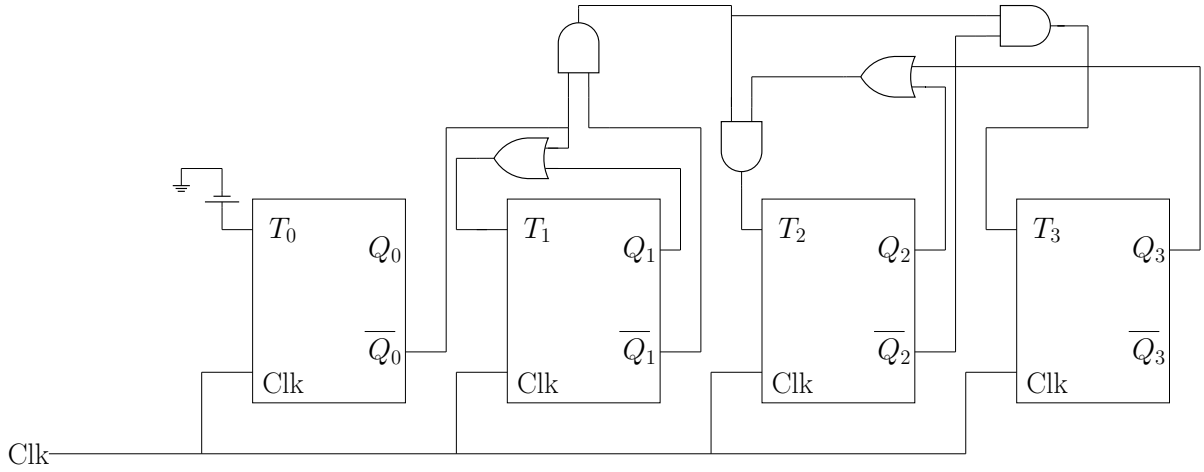



Figure 3: Circuit diagram representing the logic for decrementing.

Let the circuit be represented as:

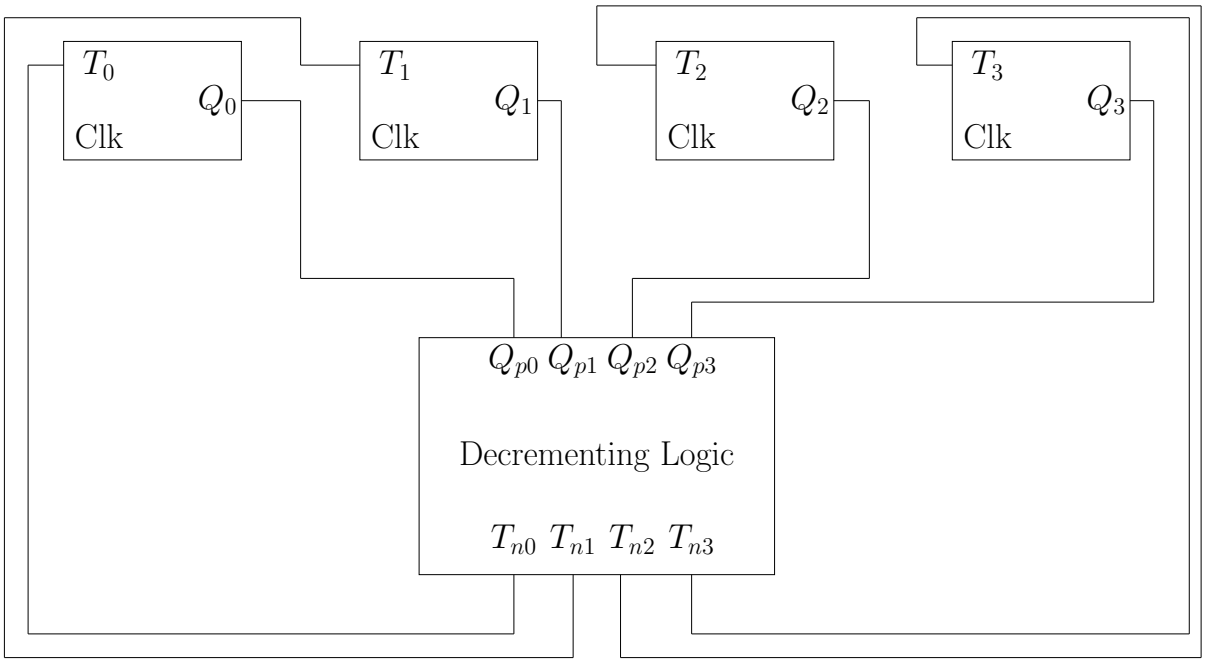



Figure 4: Simplified version of Circuit diagram

## Connecting Incrementing and Decrementing Logic

Since we need to activate the Incrementing logic only when the incrementing button $B_I$ is pressed, and activate the Decrementing logic only when the decrementing button $B_D$ is pressed, we pass the Incrementing logic and input of $B_I$ to an AND gate, similarly for Decrementing. Then we will pass outputs of both the AND gates to an OR gate so that whenever $B_I$ is pressed incrementing logic will go through, if $B_D$ is pressed decrementing logic will go through. We will connect clocks of the flip flops to OR of $B_I$ and $B_D$ so that whenever we press a button a pulse will pass activating the flip flops. The Circuit Diagram for this implementation will be:

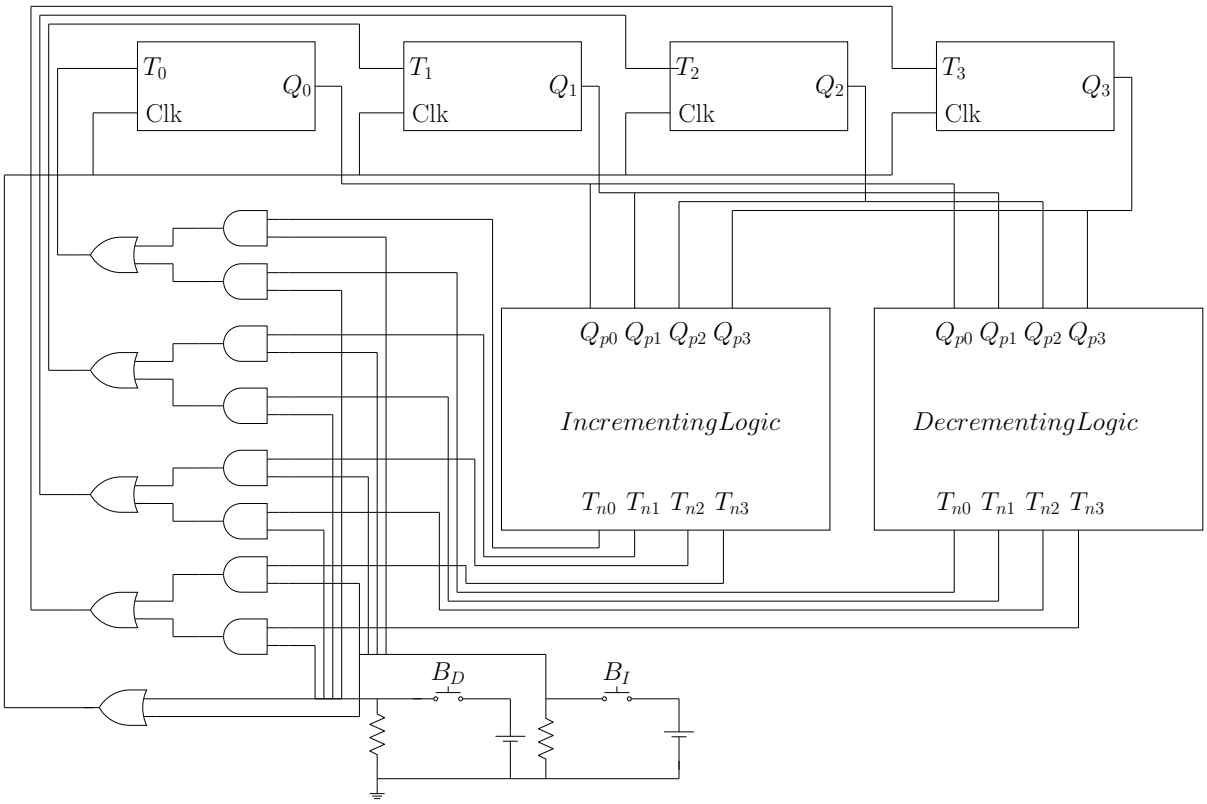



Figure 5: Combining Incrementing and Decrementing for one digit

Let us represent this circuit as:

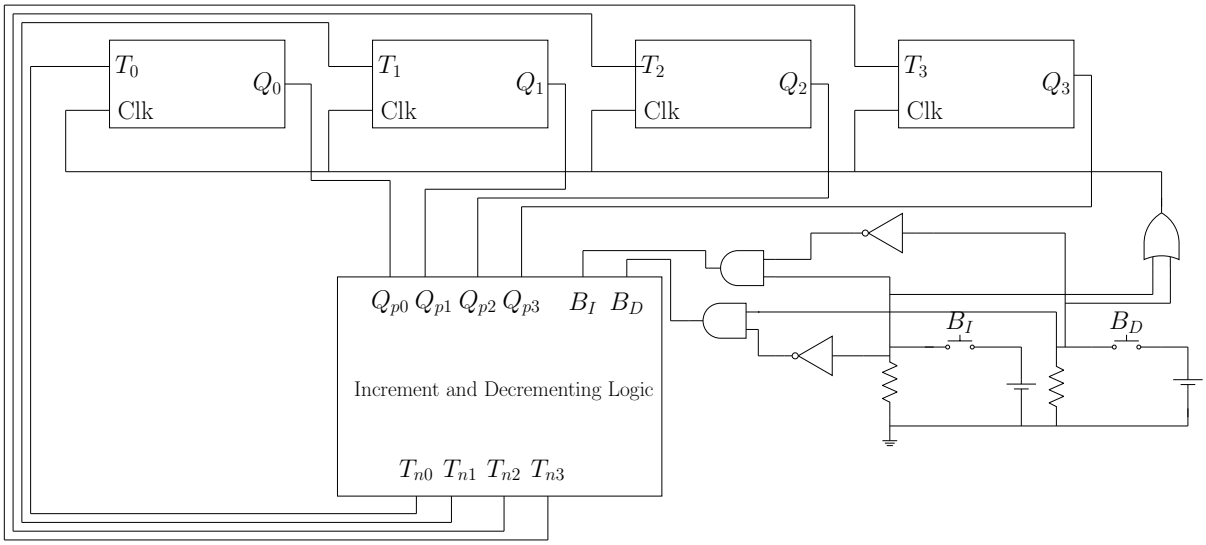
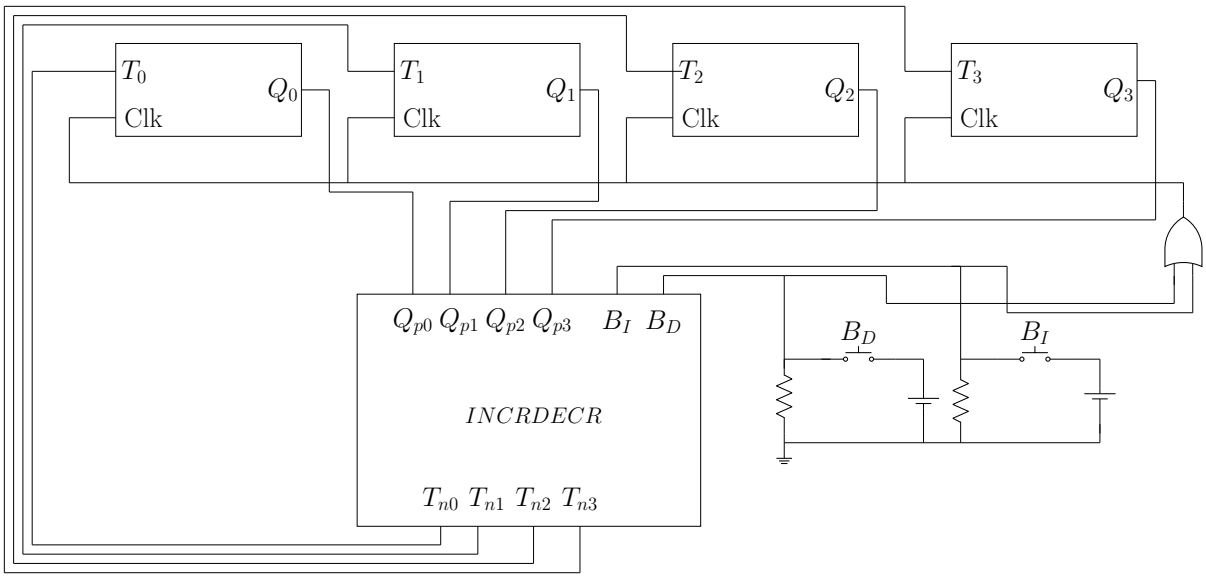



Figure 6: Simplified circuit without Button Logic

The AND and NOT gates which take input as $B_I$ and $B_D$ is a logic so that the circuit increments only when only $B_I$ is pressed, and decrements only when only $B_D$ is pressed. In rest all cases it does nothing. Let us represent the circuit in the following way:




## The Entire Circuit

Now that we have the increment and decrement logic for the first digit, For the second digit we will create the exact same circuit but AND the clock with $Q0$, $\overline{Q1}$, $\overline{Q2}$, $Q3$ of the first digit so that it will only increment when the first digit is 9. The Circuit for that will be as follows:

In the above circuit $Q_{ij}$ represents $j^{th}$ bit of $i^{th}$ digit. That is the BCD number is in the form:

$$Q_{13}Q_{12}Q_{11}Q_{10} \quad Q_{03}Q_{02}Q_{01}Q_{00}$$

Connecting the output to 7447 Decoder and then Connecting the output of that to Seven segment Displays helps us to visualize the output

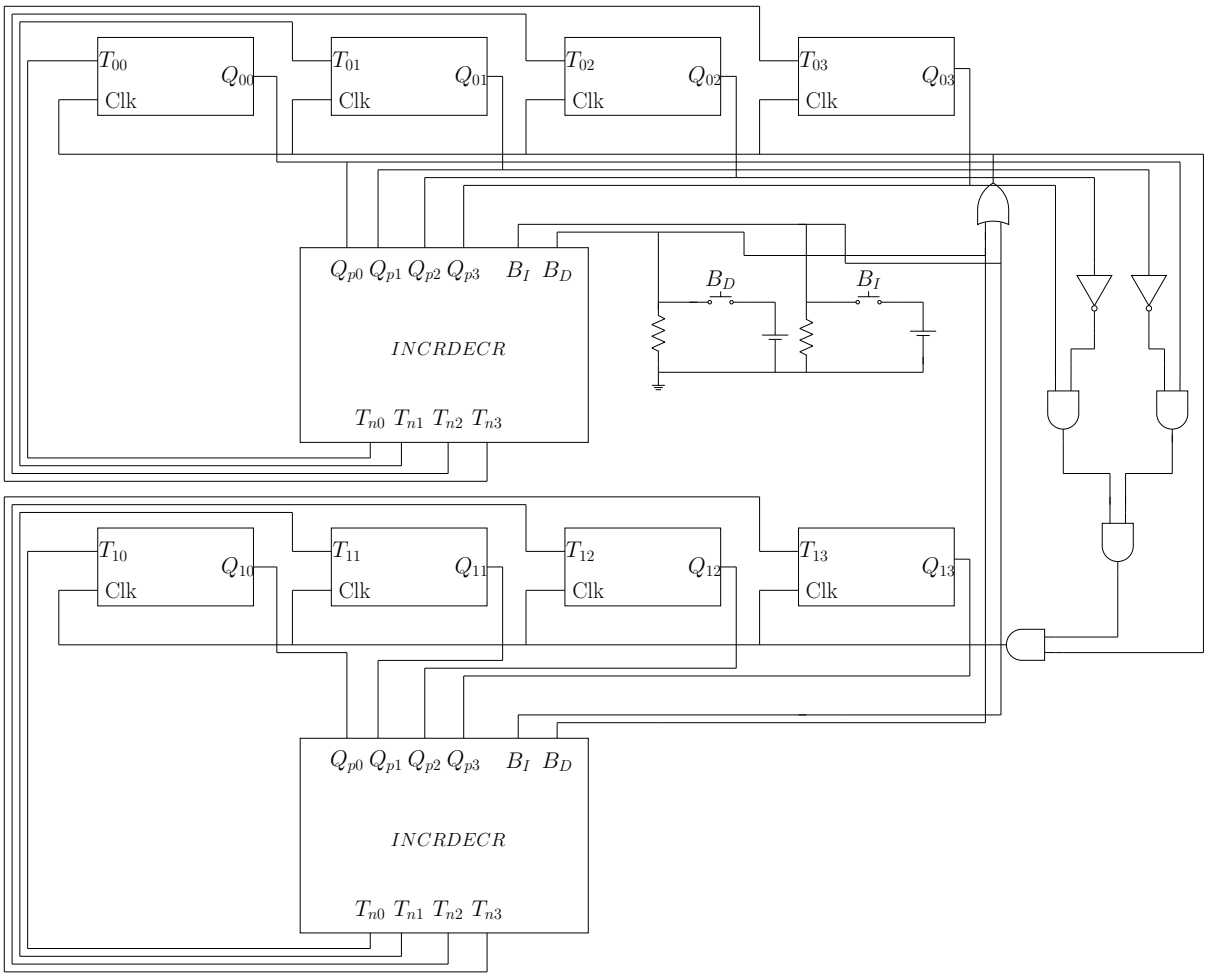



Figure 7: Complete Circuit

# Procedure

## 1. Circuit Assembly

- Configure four JK flip-flops as T flip-flops by connecting both J and K inputs to the same control signal.
- Implement T input logic for each flip-flop:
  - **Up Counter:** $T_0 = 1$, $T_1 = \overline{Q_3}Q_0$, $T_2 = Q_1Q_0$, $T_3 = Q_2Q_1Q_0 + Q_3Q_0$
  - **Down Counter:** $T_0 = 1$, $T_1 = (Q_1 + Q_2 + Q_3)\overline{Q_0}$, $T_2 = (Q_2 + Q_3)\overline{Q_1Q_0}$, $T_3 = \overline{Q_2Q_1Q_0}$
- Design the mode selection logic using basic gates as shown in Figure 5.
- Connect push buttons $B_1$ (for increment) and $B_2$ (for decrement) to provide manual clock pulses.
- Attach the Q outputs to a BCD-to-7-segment decoder (7447) and then connect it to a 7-segment display.

### 2. Power and Clock Setup

- Power the circuit using a 5V DC supply.
- Optionally use an Arduino board to provide precise and stable clock signals.
- Use a multimeter to verify all wiring and component connectivity.

### 3. Up Counter Testing

- Activate $B_1$ to select up-count mode.
- Press the increment button to provide clock pulses.
- Observe the 7-segment display for the following sequence: $0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \rightarrow 9 \rightarrow 0$.

### 4. Down Counter Testing

- Activate $B_2$ to select down-count mode.
- Press the decrement button to provide clock pulses.
- Observe the 7-segment display for the following sequence: $9 \rightarrow 8 \rightarrow \ldots \rightarrow 0 \rightarrow 9$.

### 5. Verification and Troubleshooting

- Use an oscilloscope to check clock waveform integrity.
- Monitor the output states using a logic analyzer to ensure correctness.
- Confirm proper reset behavior at $0 \leftrightarrow 9$ transition points.
- Debug logic gate outputs and T-input conditions in case of errors.

## Observation

### Up Counter

- The display starts at 0 and increments with each press of the increment button.
- Valid BCD outputs from 0000 to 1001 (0 to 9 in decimal) are shown.
- After reaching 9, the next pulse resets the counter to 0.

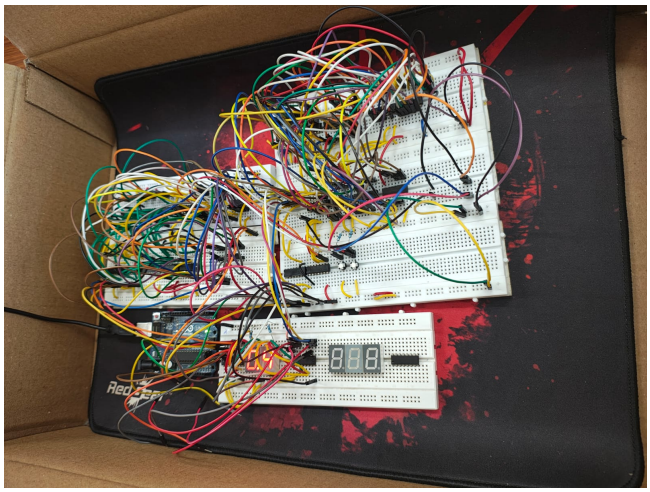
### Down Counter

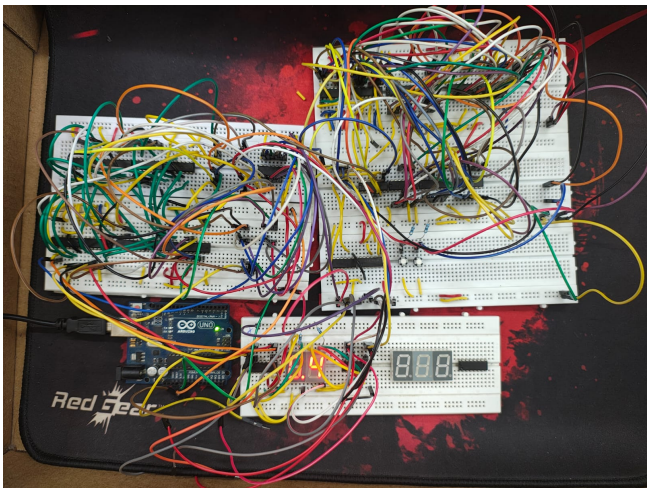
- The display starts at 9 and decrements with each press of the decrement button.
- Valid BCD outputs from 1001 to 0000 (9 to 0) are shown.
- After reaching 0, the next pulse resets the counter to 9.

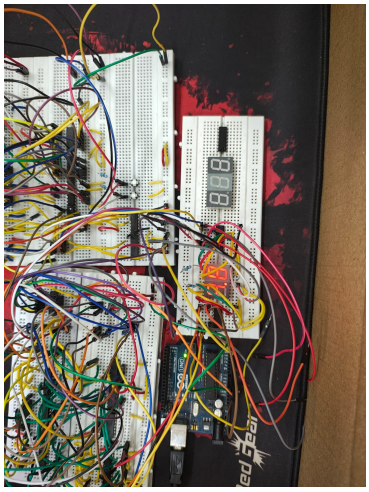
### Mode Switching

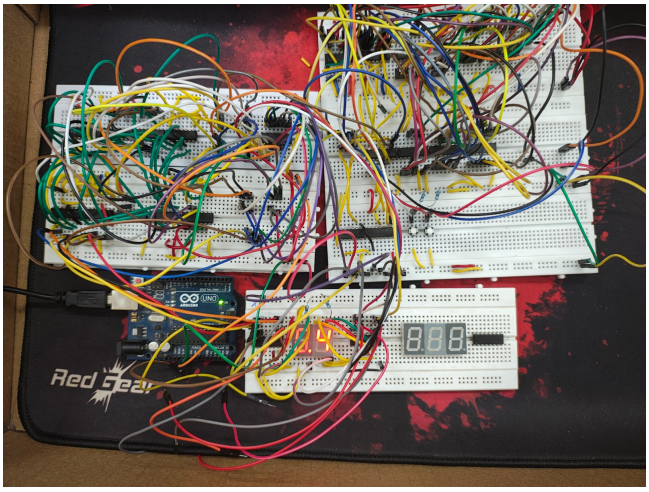
- Switching between modes does not reset the counter value.
- The counter continues in the selected direction from its current state.
- No invalid BCD states (e.g., 1010 to 1111) are observed.

# Results









# Conclusion

The BCD counter using T flip-flops was successfully designed and implemented with bidirectional counting capability. The counter correctly counts from 0 to 9 in up mode and from 9 to 0 in down mode, resetting appropriately at the boundaries. The use of push buttons to select the counting direction, along with a BCD-to-7-segment decoder for display, provides a clear demonstration of sequential logic in digital systems. This design can be extended to applications such as digital clocks, frequency counters, and event counters where bidirectional counting is required.