

# Experiment:0 to 99 Synchronous Up/Down Counter using JK Flip-Flops

EE24BTECH11001 - Aditya Tripathy, EE24BTECH11018 - Durgi Swaraj Sharma

## Aim

To design and implement a 0 to 99 synchronous up/down counter using JK flip-flops configured as T flip-flops.

## Apparatus

- IC 7476 (JK Flip-Flop)
- IC 7411 (AND gates)
- IC 7432 (OR gates)
- IC 7447 (BCD Decoder)
- 7 Segment Displays
- Push buttons for UP and DOWN counting
- Breadboard and connecting wires
- Arduino UNO R3

## Theory

A MOD-10 counter goes through 10 states (0000 to 1001) before resetting to 0000. JK flip-flops can be configured as T flip-flops by connecting both J and K inputs together.

Each T flip-flop toggles on receiving a high (1) on its T input. The T inputs are generated by combinational logic based on the current state and the count direction (UP or DOWN).

The UP and DOWN signals are ANDed with the toggle logic of each flip-flop, and then both outputs are ORed to produce the final T input for each flip-flop.

## Circuit Picture

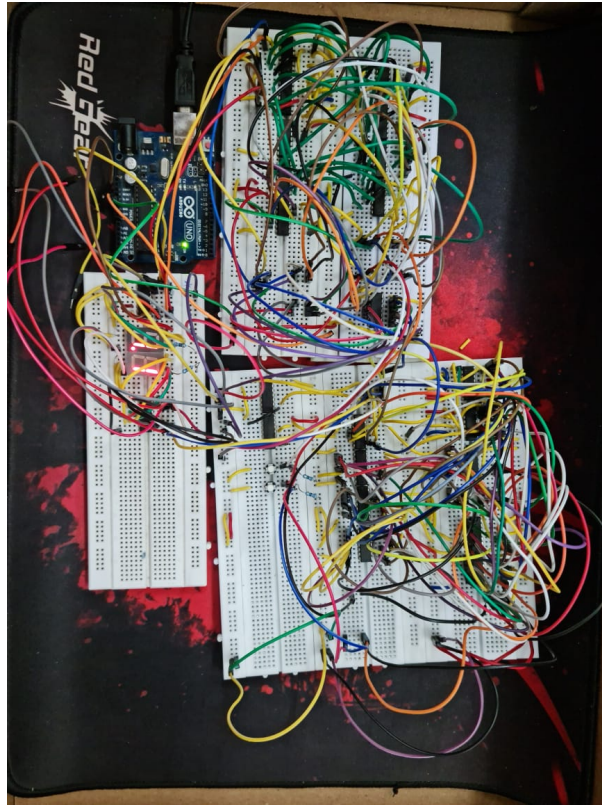
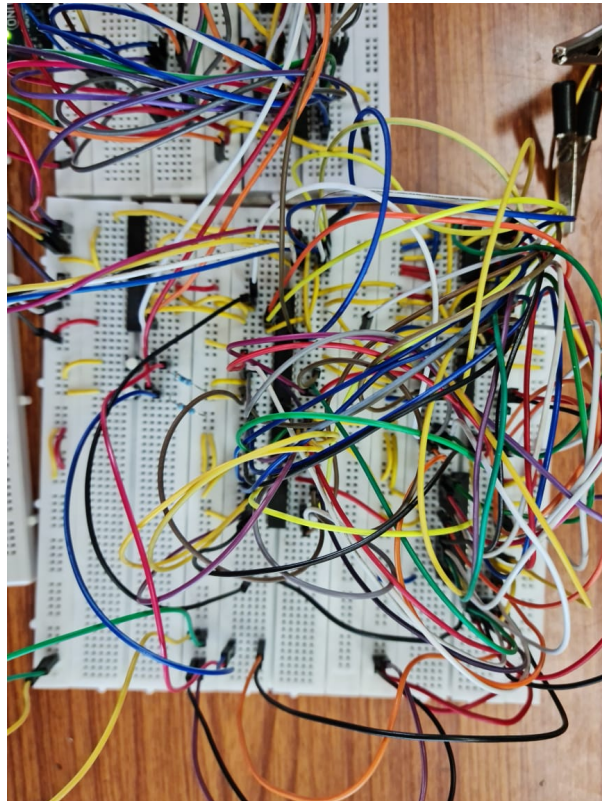
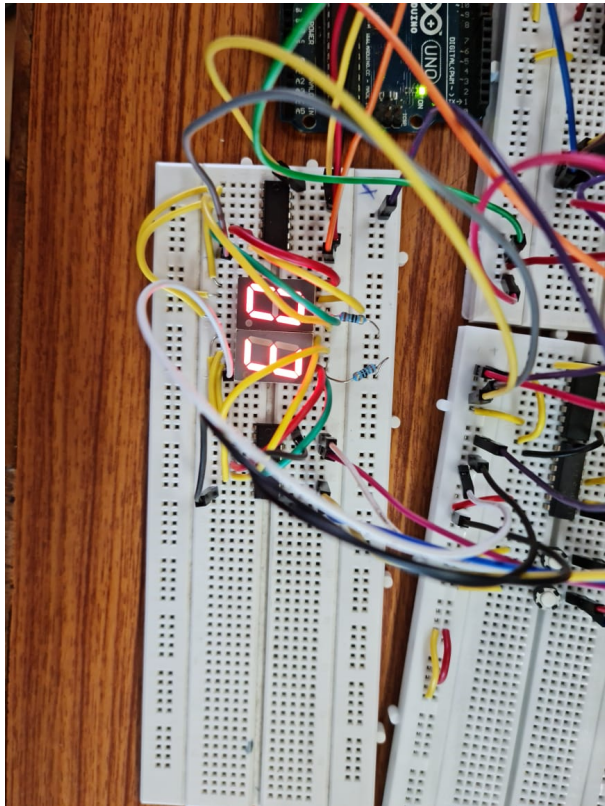
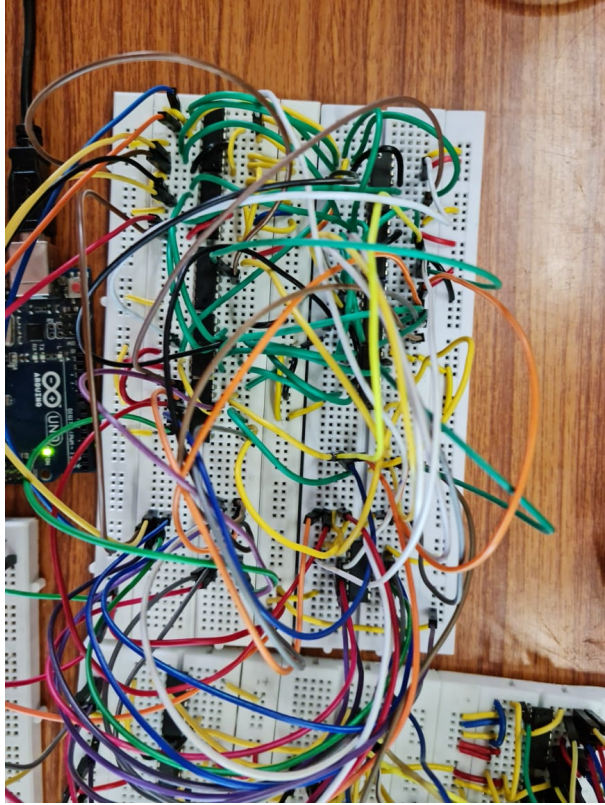


Figure 1: Actual implementation of the counter on breadboard





## State Transition Table (Up Count)

Present State	Next State (Up)	$T_3^{\text{up}}$	$T_2^{\text{up}}$	$T_1^{\text{up}}$	$T_0^{\text{up}}$
0000 (0)	0001 (1)	0	0	0	1
0001 (1)	0010 (2)	0	0	1	1
0010 (2)	0011 (3)	0	0	0	1
0011 (3)	0100 (4)	0	1	1	1
0100 (4)	0101 (5)	0	0	0	1
0101 (5)	0110 (6)	0	0	1	1
0110 (6)	0111 (7)	0	0	0	1
0111 (7)	1000 (8)	1	1	1	1
1000 (8)	1001 (9)	0	0	0	1
1001 (9)	0000 (0)	1	0	0	1

## State Transition Table (Down Count)

Present State	Next State (Down)	$T_3^{\text{down}}$	$T_2^{\text{down}}$	$T_1^{\text{down}}$	$T_0^{\text{down}}$
0000 (0)	1001 (9)	1	0	0	1
0001 (1)	0000 (0)	0	0	0	1
0010 (2)	0001 (1)	0	0	1	1
0011 (3)	0010 (2)	0	0	0	1
0100 (4)	0011 (3)	0	1	1	1
0101 (5)	0100 (4)	0	0	0	1
0110 (6)	0101 (5)	0	0	1	1
0111 (7)	0110 (6)	0	0	0	1
1000 (8)	0111 (7)	1	1	1	1
1001 (9)	1000 (8)	0	0	0	1

## 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  $T_1$ :**

		$Q_1Q_0$			
		00	01	11	10
$Q_3Q_2$	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  $T_2$ :**

		$Q_1Q_0$			
		00	01	11	10
$Q_3Q_2$	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  $T_3$ :**

		$Q_1Q_0$			
		00	01	11	10
$Q_3Q_2$	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  $T_1$ :**

		$Q_1 Q_0$			
		00	01	11	10
$Q_3 Q_2$	00	0	0	0	1
	01	1	0	0	1
	11	X	X	X	X
	10	1	0	X	X

Thus,

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

**For  $T_2$ :**

		$Q_1 Q_0$			
		00	01	11	10
$Q_3 Q_2$	00	0	0	0	0
	01	1	0	0	0
	11	X	X	X	X
	10	1	0	X	X

Thus,

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

**For  $T_3$ :**

		$Q_1Q_0$			
		00	01	11	10
$Q_3Q_2$	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 circuit is easier to build if we 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.  $T_3 = \overline{Q_2}T$

due to the type of ICs we have (3 input AND gates and 2 input OR gates)

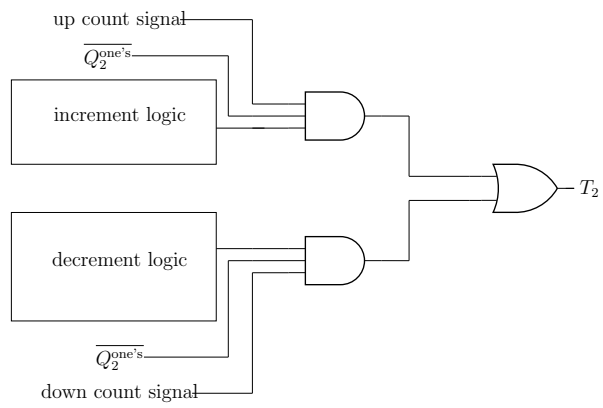
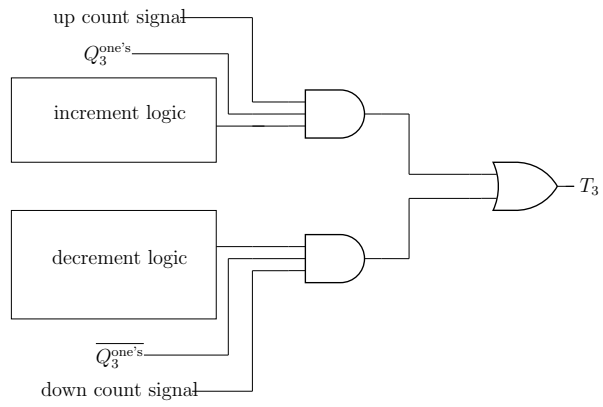
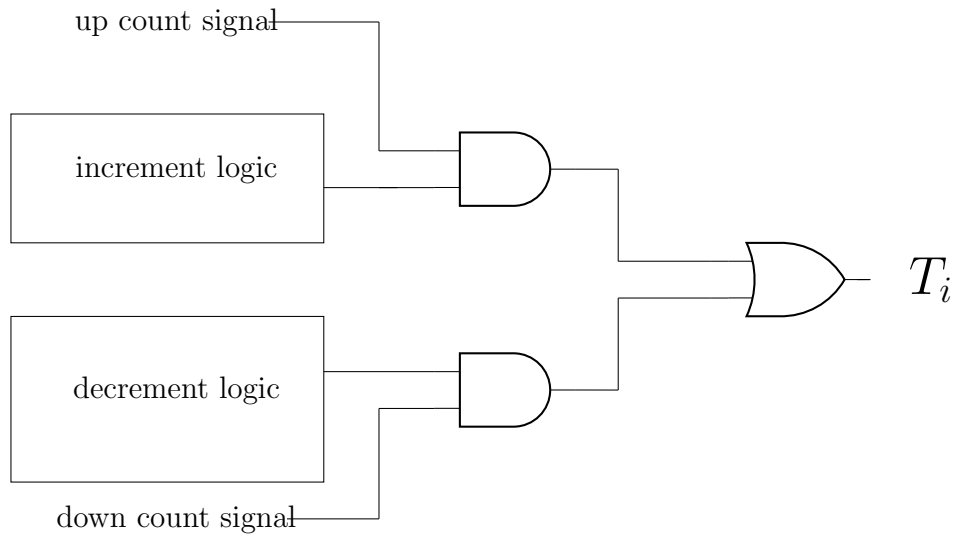
## Procedure

### One's Digit

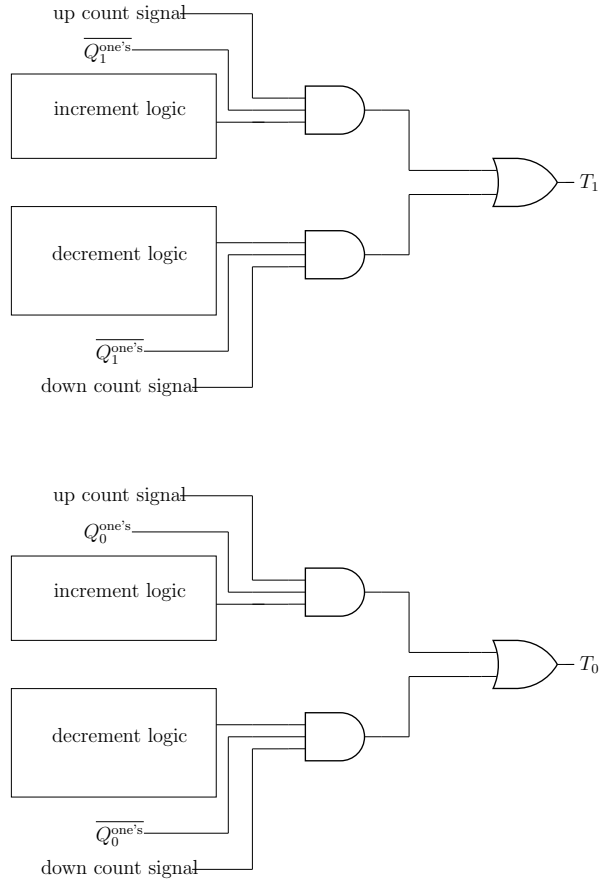
For each flip flop, the T pin should either receive incrementing combinational logic or decrementing logic based on whether the "up button" is pressed or the "down" button is pressed. This is achieved as shown the following diagram From this diagram it is clear that when the up count button is pressed, then the signal passed to the T pin of the flip flop will be the result of the incrementing synchronous counter k-map results, and likewise for when the down count button is pressed.

### Ten's Digit

For the ten's digit, we should increment tens place automatically (without any additional button presses) when the ones digit is 9 (1001 in binary). Likewise we will decrement the ten's place when one's place is zero (0000 in binary). The process is made clear by the following diagrams:







Therefore we can see that we are passing the incrementing logic to the respective T pins of the flip flop when the ones place is 9 ( $Q_3^{\text{one's}} = 1, Q_2^{\text{one's}} = 0, Q_1^{\text{one's}} = 0, Q_0^{\text{one's}} = 1$ ) and we pass decrement logic when ones place is zero ( $Q_3^{\text{one's}} = Q_2^{\text{one's}} = Q_1^{\text{one's}} = Q_0^{\text{one's}} = 0$ ).

## Circuit Diagram

### One's Digit

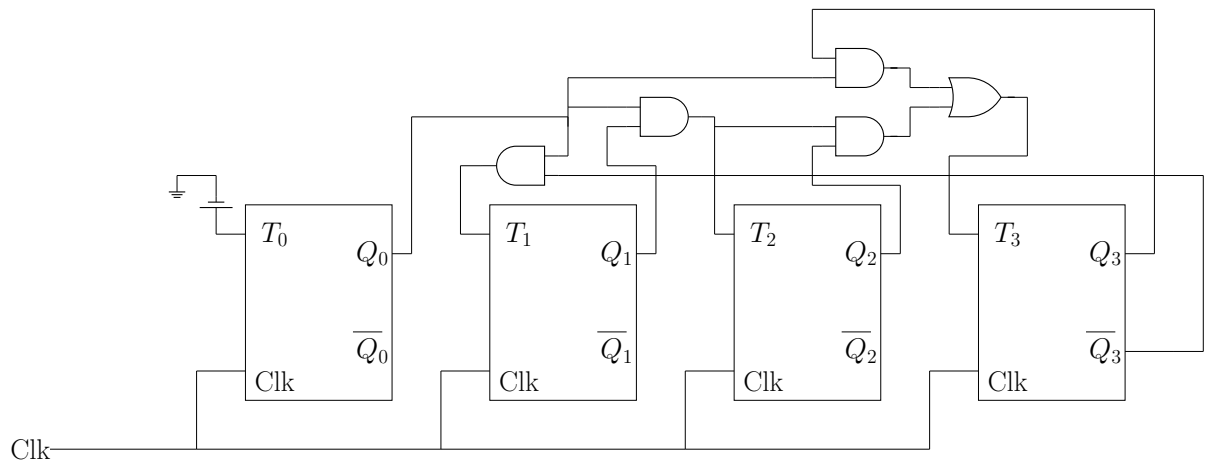


Figure 2: Circuit diagram representing the logic for incrementing.

The circuit can be represented as follows by abstracting away the and gates into a module

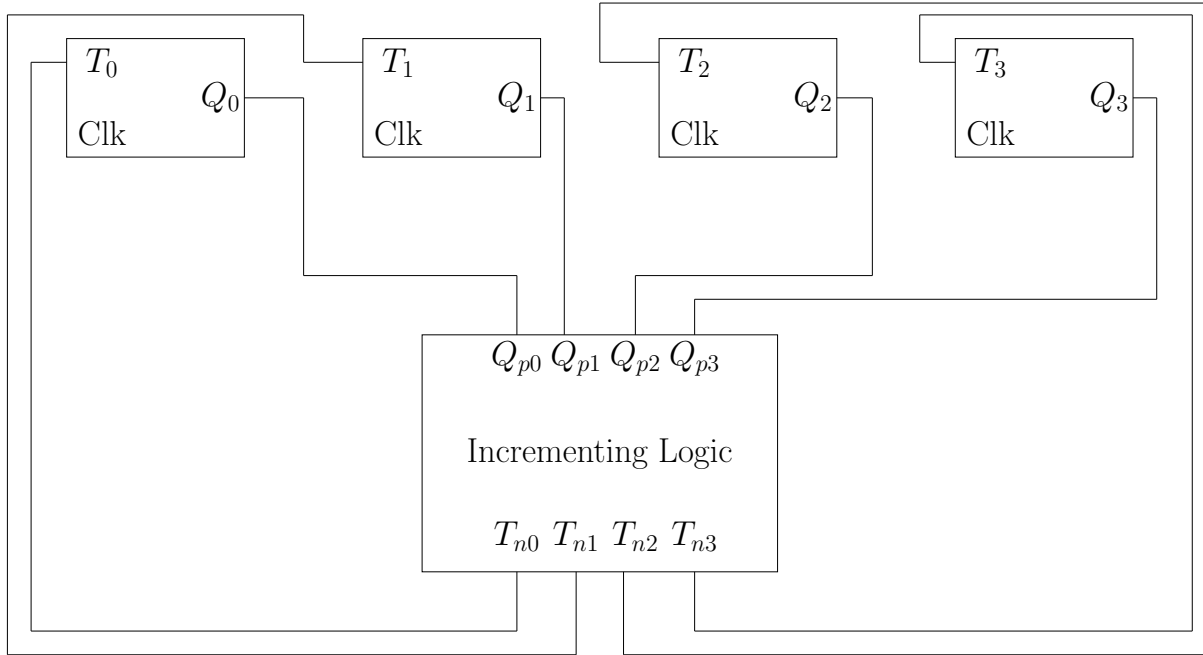


Figure 3: Simplified version of Circuit diagram

## Circuit Diagram for Decrementing

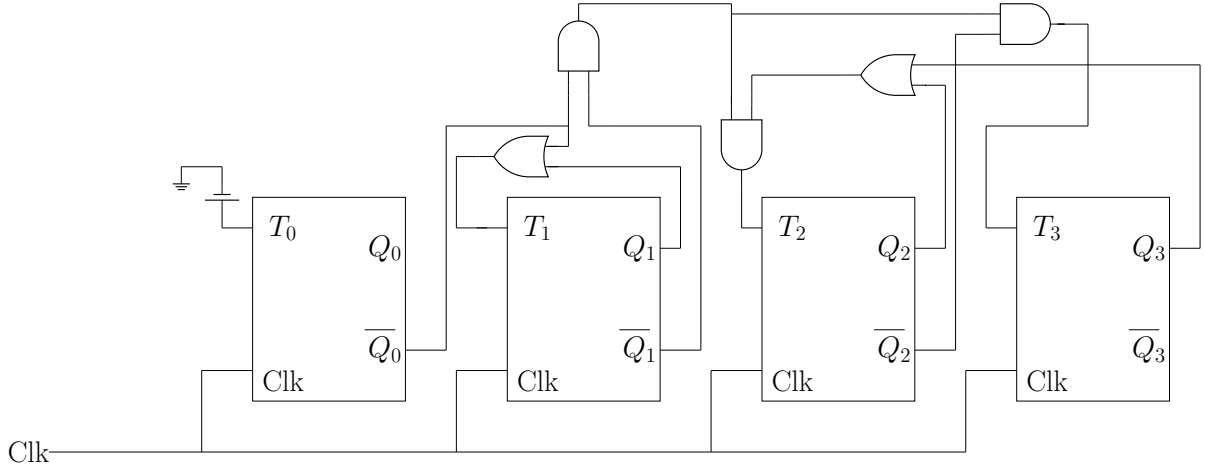


Figure 4: Circuit diagram representing the logic for decrementing.

Likewise we can build the circuit for decrementing logic,

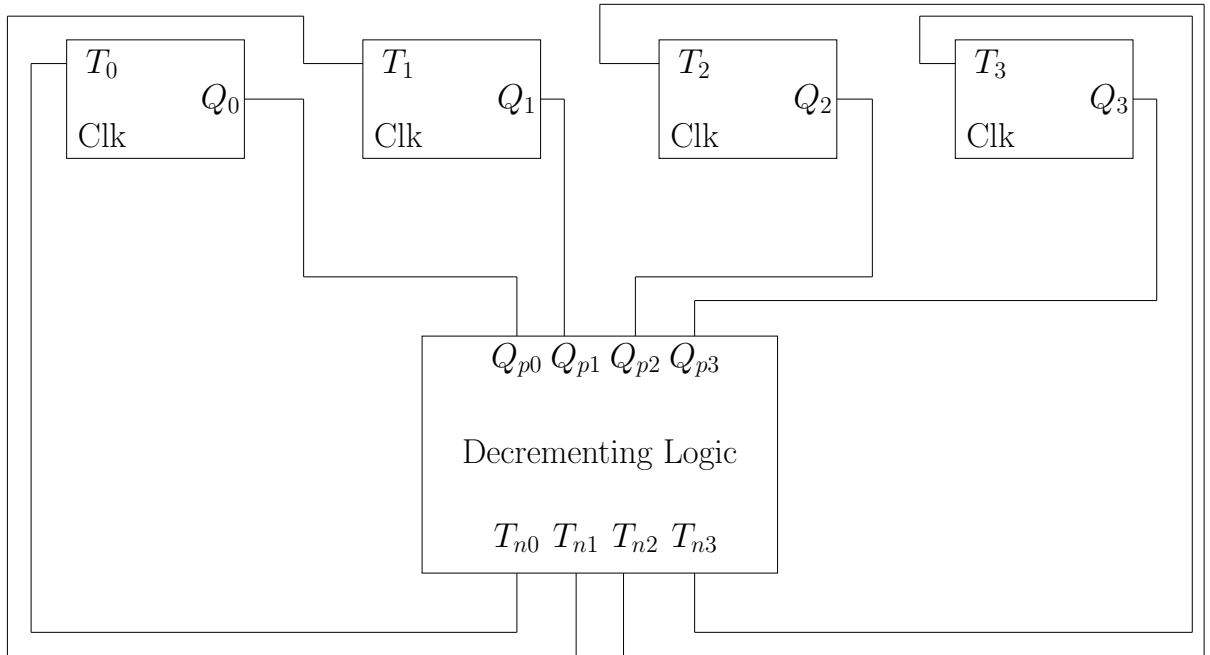


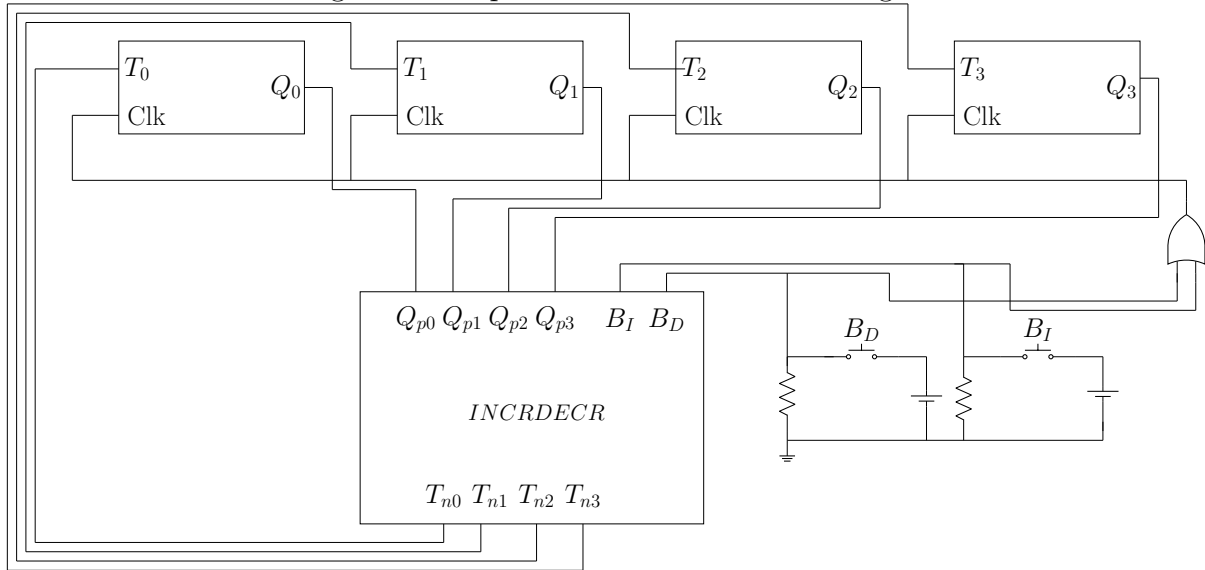
Figure 5: Simplified version of Circuit diagram

## Connecting Incrementing and Decrementing Logic

The connection of the incrementing and decrementing logic is done using previous logic as shown below in a separate module.

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.

Figure 6: Simplified circuit with button logic

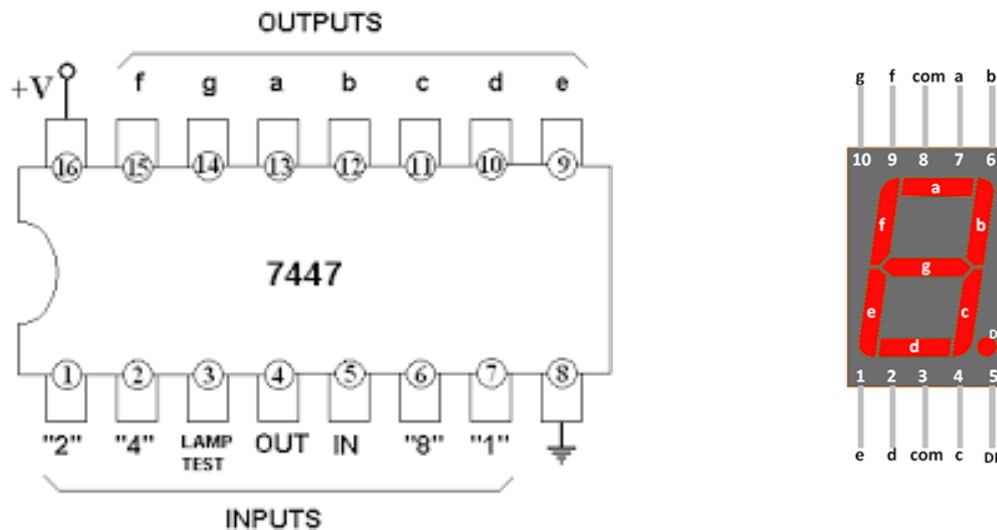


## Reset logic

It was asked that there should be a button to reset the counter to zero. This is implemented by connecting the  $\overline{CLEAR}$  of each T-Flip Flop to the reset button via a not gate, because the pin resets all the ICs to zero only when it gets low voltage.

## Seven Segment Display using BCD

Connect  $Q_3, Q_2, Q_1$  and  $Q_0$  of both the one's and ten's digit according to the datasheet provided by Texas instruments,



## Results

The working of the circuit is demonstrated in the video given in our github repository:

[Circuit Demonstration Video](#)