Design and Demonstration of a Mod-7 Asynchronous Counter Using T Flip-Flops and Arduino Clock

Sai Akhila Reddy Turpu - EE24BTECH11055
Sai Akshitha Suguru - EE24BTECH11054

1. Objective

To design a **Mod-7 asynchronous counter** using **T Flip-Flops** and observe its performance using a **Cathode Ray Oscilloscope (CRO)** with a clock signal provided by an **Arduino**.

2. Introduction

Counters are sequential digital circuits used for counting pulses. A **Mod-7 counter** counts from 0 to 6, cycling through 7 distinct states.

An **asynchronous counter** (also known as a ripple counter) is one in which the clock is applied only to the first flip-flop, and the output of each flip-flop serves as the clock for the next. The **T (Toggle) Flip-Flop** is ideal for this purpose as it toggles its state on every active clock edge when the T input is high.

For a Mod-7 counter:

- Number of required states = 7
- Minimum number of flip-flops = 3
- Number of states with 3 FFs = $2^3 = 8$ (so 1 unused state must be avoided)

3. Truth Table

The truth table for the Mod-7 counter is as follows:

Clock Pulse	Q2 (MSB)	Q1	Q0 (LSB)
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0

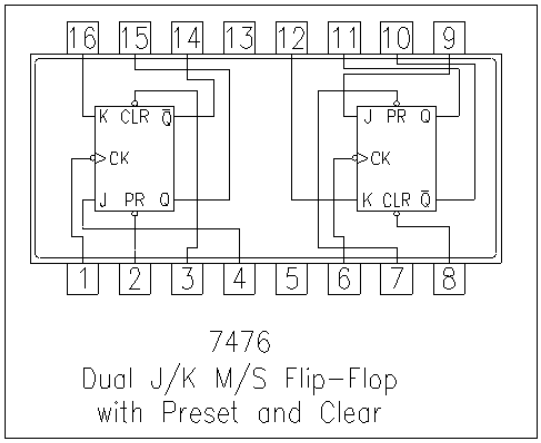
The state 111 is unused and must be forced to reset to 000.

T	Q(n)	J	K	Q(n+1)
0	0	0	X	0
0	1	X	0	1
1	0	1	X	1
1	1	X	1	0

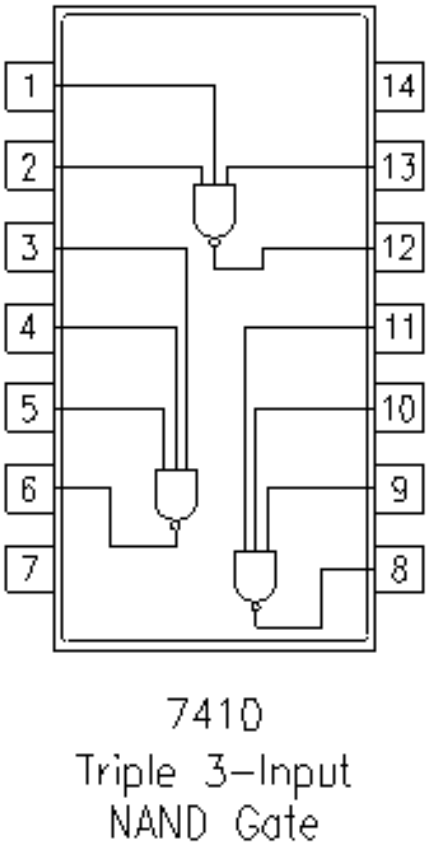
Table 1: Conversion of JK Flip-Flop to T Flip-Flop

4. Circuit Diagram

- Three T Flip-Flops connected in series.
- T inputs are held high (T=1) for toggle behavior. Hence we set J and K input to 1.
- Clock is applied to the first JK FF.
- Output of one flip-flop is connected as clock to the next (ripple configuration).
- A logic gate (NAND) detects the unused state (111) and resets the counter.

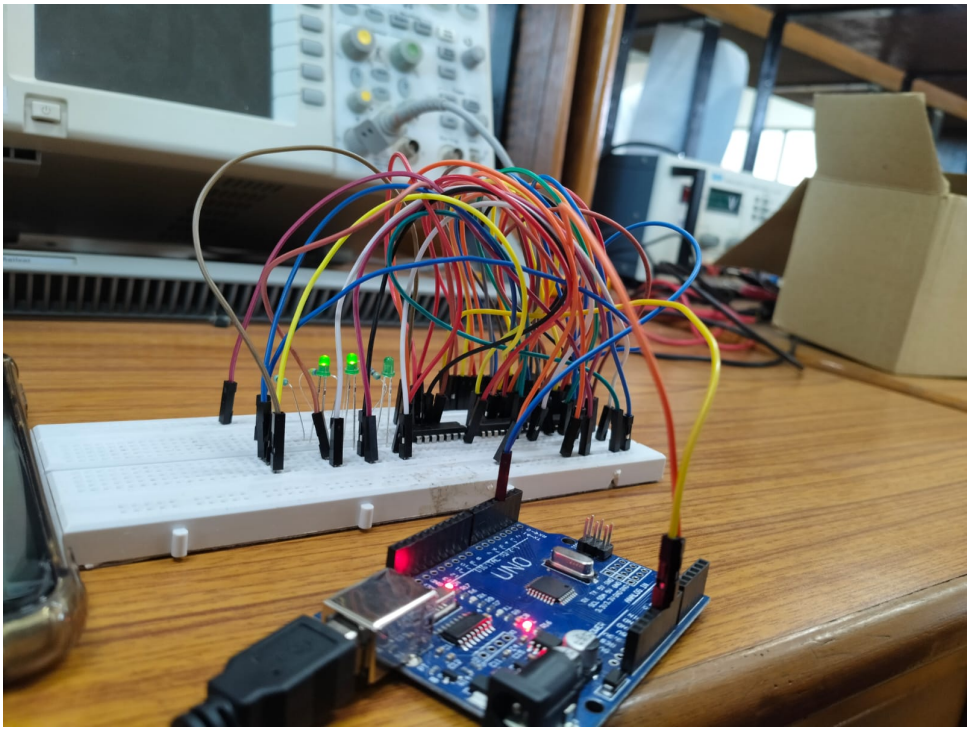


(a) IC7476 pin diagram



(b) IC7410 pin diagram

Figure 1: Pin diagrams



5. Arduino Clock Generation

The Arduino can be programmed to generate a square wave clock signal using a digital output pin and delay.

Arduino Code

```
void setup() {
  pinMode(3, OUTPUT);
}
void loop() {
  digitalWrite(3, HIGH);
  delay(1000); // 1000 ms high
  digitalWrite(3, LOW);
  delay(1000); // 1000 ms low
}
```

Note: Pin 3 provides a 2 Hz clock pulse (adjust delay for frequency).

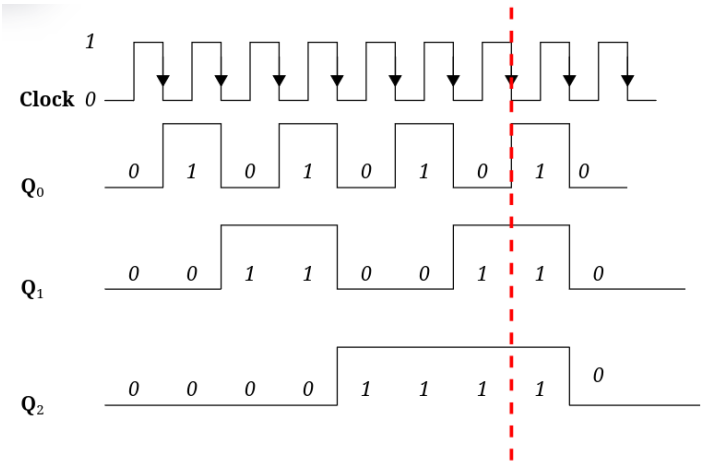


Figure 2: Clock signal and output signals of Q_0, Q_1 and Q_2

6. Observing on CRO

Connect the Q0, Q1, and Q2 outputs to different channels of the CRO to observe:

- Clock signal waveform
- Sequential toggling of flip-flop outputs
- Reset after the 7th state

We see Q0 toggling on every pulse, Q1 toggling every 2 pulses, and Q2 every 4 pulses until reset at state 7.

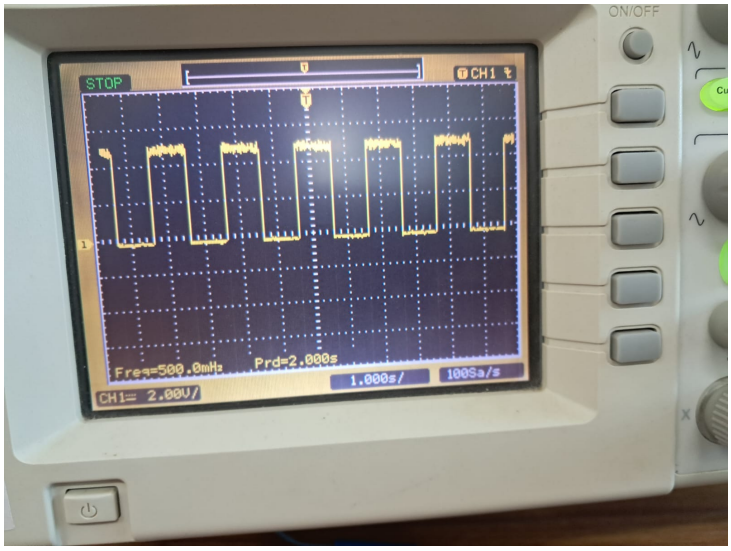


Figure 3: Clock Signal: Synchronizes state transitions.

Frequency = 0.5 mHz

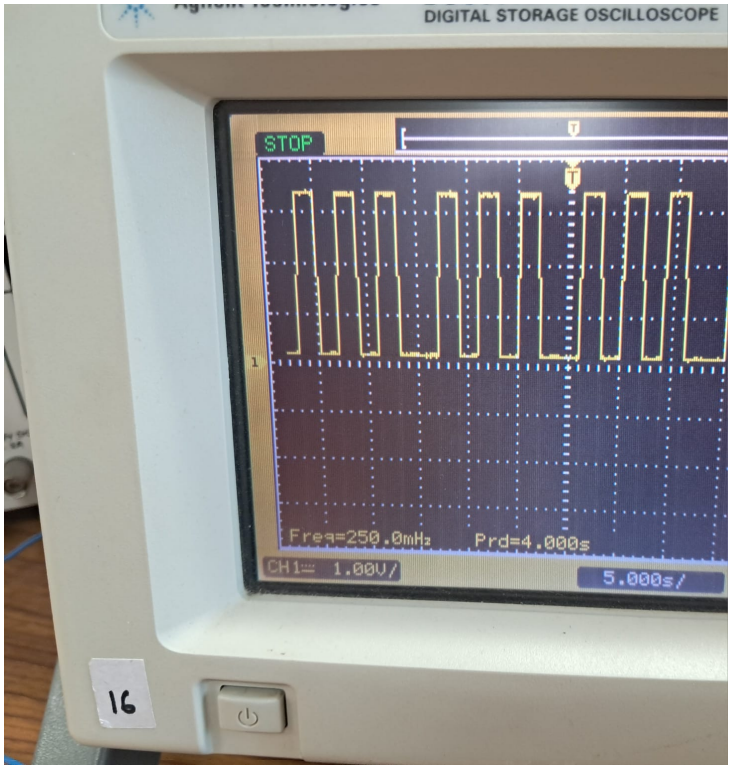


Figure 4: Q0 Output: Least significant bit of the counter.

Frequency = f/2 = 0.250 mHz

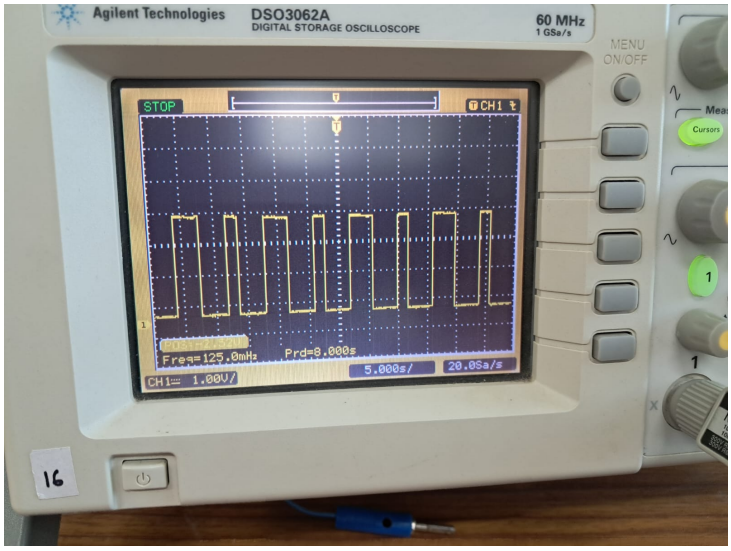


Figure 5: Q1 Output: Middle bit of the counter.

Frequency = f/4 = 0.125 mHz

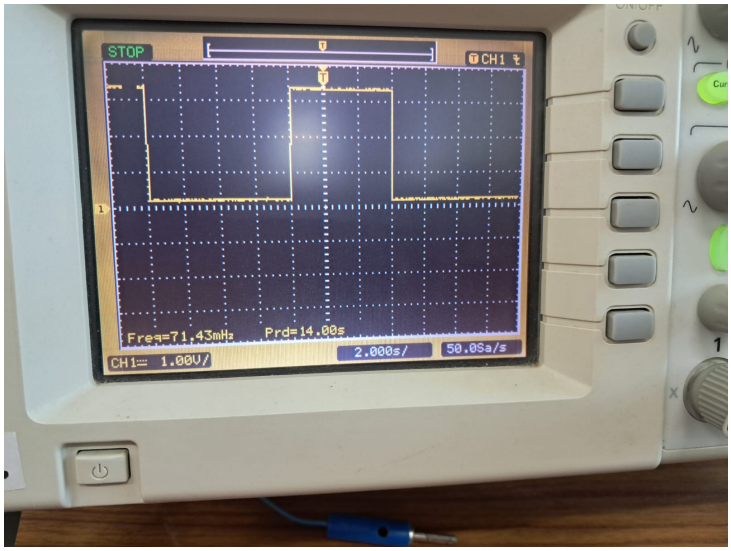


Figure 6: Q2 Output: Most significant bit of the counter.

Frequency = 0.7143 mHz

Since 7 is attenuated, the signal which was supposed to be periodic with 16 seconds, now has a new period of 14 seconds. Hence the frequency of the signal is shown as $\frac{1}{14}Hz = 0.7143$ Hz.

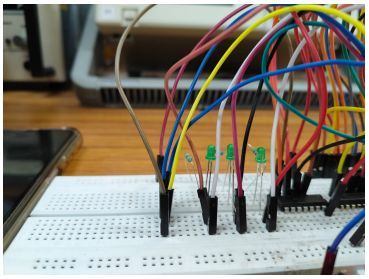
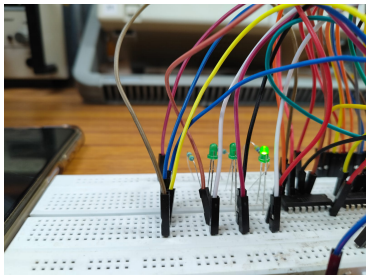
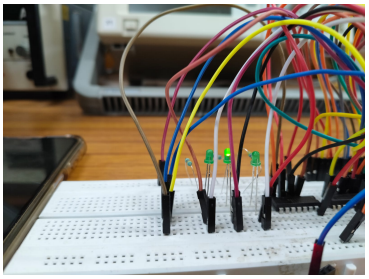
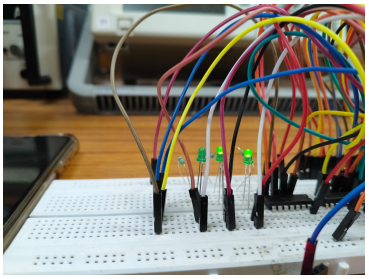
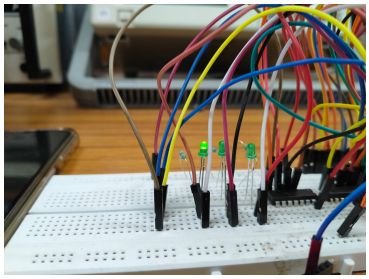
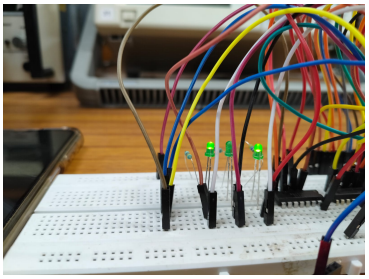
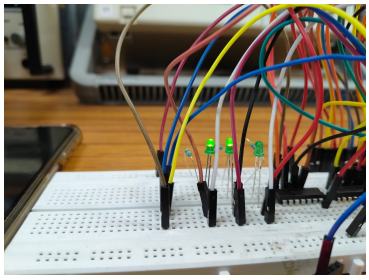
		
Number0	Number1	Number2
		
Number3	Number4	Number5
		
	Number6	

Table 2: Collection of Number images

7. Result

A working Mod-7 asynchronous counter was successfully designed using T Flip-Flops. Its correct operation was verified using a CRO. The counter counts from 000 to 110 and resets after the 7th pulse. A clock signal from an Arduino was effectively used to drive the circuit.

8. Conclusion

This experiment demonstrates the implementation of a Mod-7 asynchronous counter using T flip-flops and illustrates how microcontroller-generated clocks can be used in digital circuits. The counter reliably resets after 7 counts, validating the design logic.