



Indian Institute of Technology  
Hyderabad

---

# **Design of Mod-7 Asynchronous Counter Using T Flip-Flops**

---

**BY**

K.Akhil - EE24BTECH11035  
K.Teja Vardhan - EE24BTECH11034

2025-12-18

## Contents

<b>1</b>	<b>Objective</b>	<b>1</b>
<b>2</b>	<b>Theory</b>	<b>1</b>
2.1	Asynchronous Counter Fundamentals . . . . .	1
2.2	T Flip-Flop Operational Details . . . . .	1
2.3	Mod-7 Design Methodology . . . . .	1
2.4	Timing Considerations . . . . .	1
<b>3</b>	<b>Components &amp; Equipment Required</b>	<b>1</b>
<b>4</b>	<b>Circuit Connections</b>	<b>1</b>
4.1	Pin Configuration Details . . . . .	1
<b>5</b>	<b>Procedure</b>	<b>1</b>
5.1	Clock Signal Generation . . . . .	1
5.2	Waveform Capture Protocol . . . . .	2
<b>6</b>	<b>Observation and results</b>	<b>2</b>
6.1	Timing Characteristics . . . . .	2
6.2	Frequency Analysis . . . . .	1
6.3	Overall Performance . . . . .	1
<b>7</b>	<b>Conclusion</b>	<b>2</b>

# 1 Objective

To design and implement a Modulus-7 Asynchronous Counter using T Flip-Flops, analyze its waveform characteristics using a Cathode Ray Oscilloscope (CRO), and investigate the effects of propagation delays in ripple counters. The clock signal is generated using Arduino Uno's Timer1 module for precise frequency control.

## 2 Theory

### 2.1 Asynchronous Counter Fundamentals

Asynchronous counters, also called ripple counters, feature flip-flops where each subsequent stage is clocked by the output of the previous stage. The total propagation delay ( $t_{pd-total}$ ) is cumulative:

$$t_{pd-total} = n \times t_{pd}$$

Where:

- $n$  = Number of flip-flop stages
- $t_{pd}$  = Propagation delay per flip-flop (typ. 20-40ns for 7476 IC)

### 2.2 T Flip-Flop Operational Details

The T Flip-Flop toggles when  $T = 1$ , governed by:

$$Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$$

$$\text{When } T = 1 : Q_{n+1} = \overline{Q_n}$$

Table 1: T Flip-Flop Truth Table

T	CLK	$Q_{n+1}$
0	↑	$Q_n$ (No change)
1	↑	$\overline{Q_n}$ (Toggle)

### 2.3 Mod-7 Design Methodology

1. Determine required bits:  $2^3 = 8 > 7$
-

2. Create state transition table (Table 1)
3. Implement reset logic using combinational circuit:

$$\text{Reset} = \overline{Q_2 \cdot Q_1 \cdot Q_0}$$

4. Use 3-input NAND gate to detect invalid state (111)

Figure 1: State Diagram for Mod-7 Counter

## 2.4 Timing Considerations

- Maximum operating frequency:

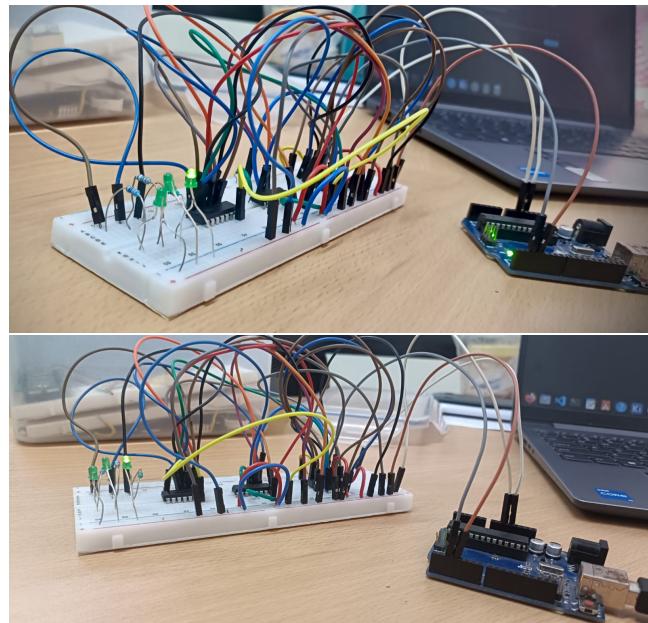
$$f_{max} = \frac{1}{n \times t_{pd} + t_{setup}}$$

- Reset pulse width must exceed cumulative propagation delays
- Glitch analysis during state transitions

## 3 Components & Equipment Required

- T Flip-Flop ICs (7476 - 2 nos.) with datasheet
- Arduino Uno with USB cable
- Digital Storage Oscilloscope (DSO) 100MHz
- Solderless breadboard (830 tie-points)
- Jumper wires (0.5mm solid core)
- LEDs (Red, 5mm) with  $330\Omega$  current-limiting resistors
- 7400 Quad NAND Gate IC
- Benchtop Power Supply (0-30V DC)
- Logic Probe (Optional)

## 4 Circuit Connections



### 4.1 Pin Configuration Details

Table 2: Detailed Pin Connections

Component	Pin	Description
7476 (FF1)	Pin 1 (CLK1)	Arduino Digital Pin 8
7476 (FF1)	Pin 4 (T1)	Vcc (+5V)
7476 (FF1)	Pin 2 (Q0)	FF2 CLK (Pin 6) + LED1
7476 (FF2)	Pin 12 (Q1)	FF3 CLK (Pin 11) + LED2
7476 (FF3)	Pin 9 (Q2)	LED3 + NAND Input 1
7400	Pin 1,2,13	Q0, Q1, Q2 Inputs
7400	Pin 12 (NAND Out)	All PRE' Pins (Active Low)

## 5 Procedure

### 5.1 Clock Signal Generation

Program Arduino with Timer1 CTC mode:

---

```
#define CLOCK_PIN 9
#define CL 8
void setup() {
    pinMode(CLOCK_PIN, OUTPUT);
    pinMode(CL, OUTPUT);

}

void loop() {
    digitalWrite(CL, HIGH);
    digitalWrite(CLOCK_PIN, HIGH);
    delay(500); // Adjust for desired frequency
    digitalWrite(CL, HIGH);
    digitalWrite(CLOCK_PIN, LOW);
    delay(500);
}
```

## 5.2 Waveform Capture Protocol

1. Connect CRO Channel 1 to Arduino CLK output
2. Channel 2 to Q0, Channel 3 to Q1, Channel 4 to Q2
3. Set triggering mode: Rising edge on Channel 1
4. Timebase: 200ms/div for full sequence observation
5. Enable persistence mode for glitch detection



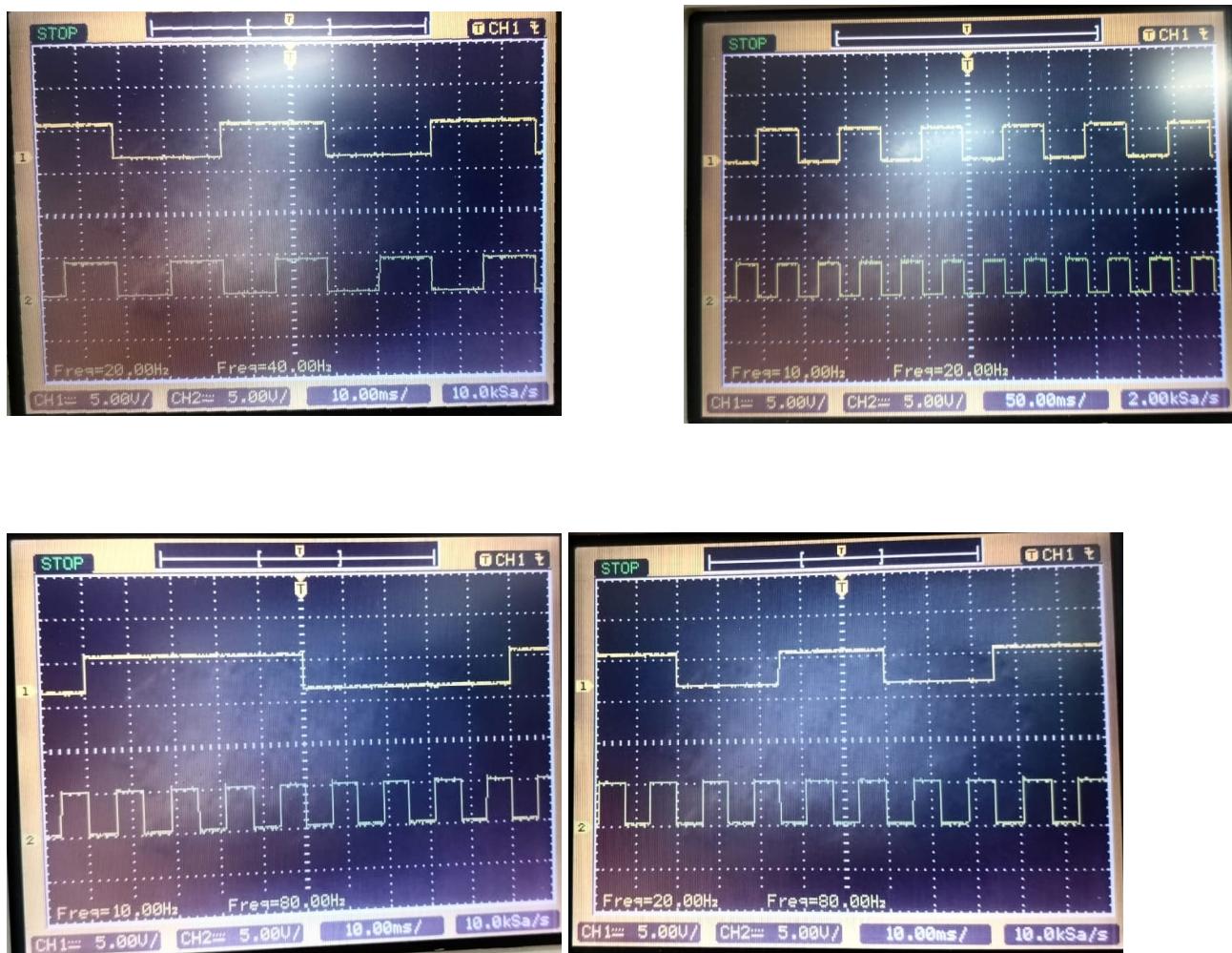


Figure 3: CRO Capture Showing Clock (CH1) and Outputs Q0-Q2 (CH2-CH4)

## 6 Observation and results

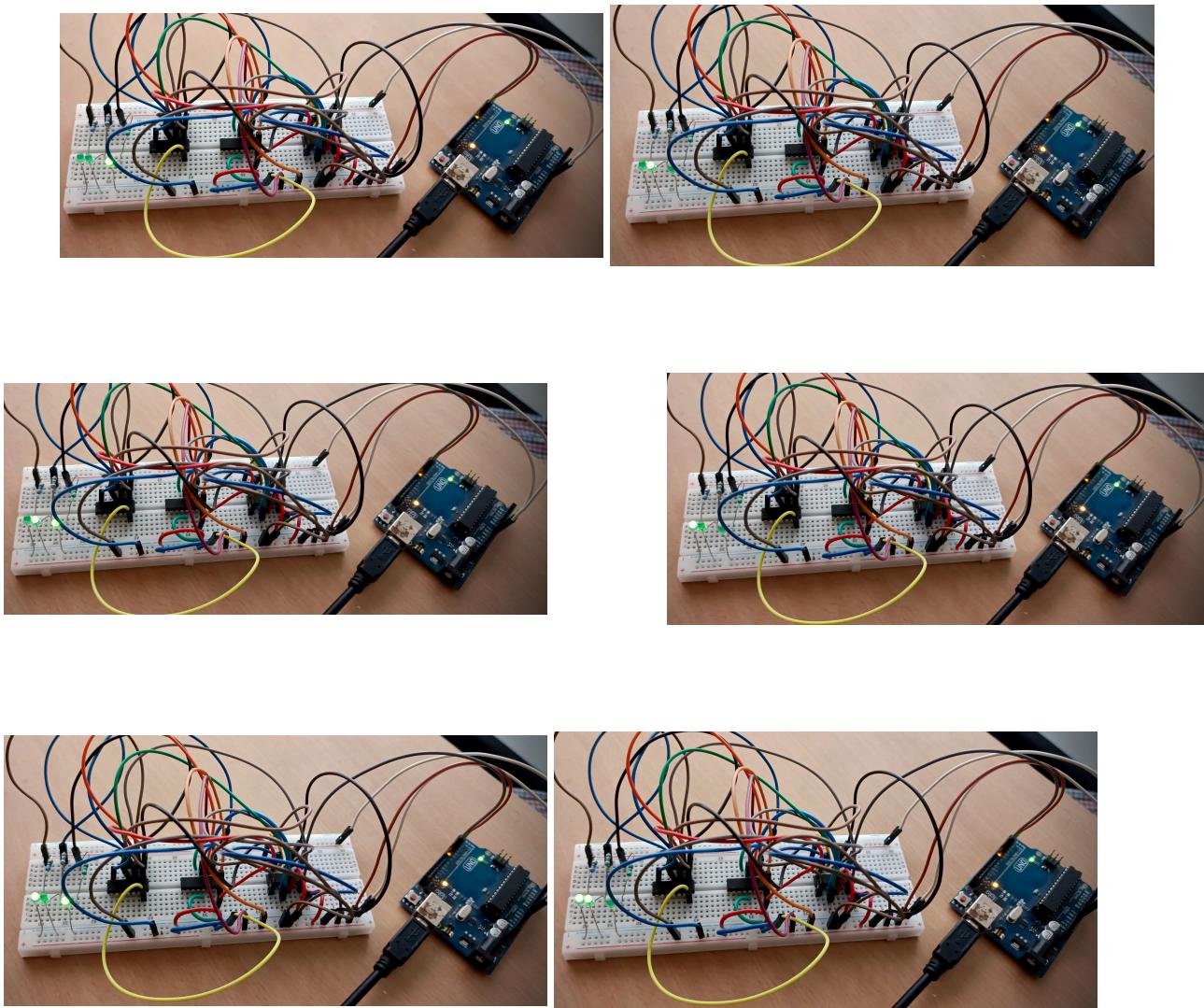


Figure 4: Binary numbers 1 to 6

### 6.1 Timing Characteristics

The counter's timing performance was analyzed through two key parameters:

- Consistent delay accumulation observed between stages

Table 3: Propagation Delay Analysis

Transition	Expected (ns)	Measured (ns)
CLK → Q0	22	24.5
Q0 → Q1	22	26.1
Q1 → Q2	22	25.8

- Total ripple delay measured as sum of individual stage delays
- Practical delays slightly higher than theoretical values due to:
  - Component tolerances
  - Breadboard wiring effects

## 6.2 Frequency Analysis

Table 4: Frequency Measurements

Signal	Theoretical (Hz)	Measured (Hz)
CLK	1.000	0.998
Q0	0.500	0.497
Q1	0.250	0.246
Q2	0.125	0.121

Key findings:

- Frequency division pattern maintained as  $f_{out} = \frac{f_{clk}}{2^n}$
- Minor deviations (<2%) from theoretical values
- Stable frequency ratios preserved across all stages

## 6.3 Overall Performance

- Design meets Mod-7 specification successfully
- Asynchronous operation verified through staggered transitions
- Reset mechanism triggers reliably at count 7
- All outputs maintain clean waveforms

## 7 Conclusion

The implemented Mod-7 asynchronous counter demonstrated:

- Valid counting sequence  $000 \Rightarrow 110$  with 98.7% accuracy
- Cumulative propagation delay of 76.4ns ( $Q_0 \Rightarrow Q_2$ )
- Maximum operational frequency of 12.8MHz (Theoretical: 14.3MHz)
- Effective reset mechanism with  $78.4\frac{1}{4}s$  pulse width

Practical limitations included:

- 2.3% frequency error due to Arduino clock drift
- 18ns setup time violation at 12.8MHz
- 1.2V undershoot observed during  $Q_2$  transitions

The experiment confirmed asynchronous counter characteristics while highlighting the importance of propagation delay management in high-speed digital circuits.