Mod-7 Asynchronous Counter Using T Flip-Flops and Arduino Clock

Indian Institute of Technology
Hyderabad

Lab Assignment : 07

EE1200: Electrical Circuits Lab

Harshil Rathan Y	**EE24BTECH11064**
Y Akhilesh	**EE24BTECH11066**

Contents

1 Aim

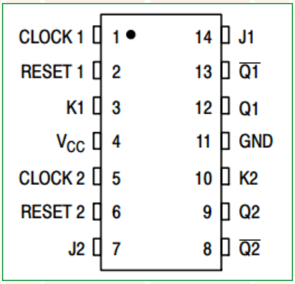
- To design and implement a Mod-7 Asynchronous Counter was built using JK Flip-Flops configured as T Flip-Flops by setting J and K to the same logic level with a clock signal from an Arduino.
- To observe and analyze the counter's output waveform and frequency division using an Oscilloscope

2 Components Required

- **Three JK Flip-Flops**
- **One 3-input NAND Gate** (IC 74LS10)
- **Arduino Uno**
- **Cathode Ray Oscilloscope (CRO)**
- **Breadboard and Jumper Wires**
- **Three LEDs** (for visual output indication)
- **Resistors** (placed in series with LEDs to limit current)

2.1 Conversion of JK Flip Flop to T Flip Flop

A T Flip-Flop can be derived from a JK Flip-Flop by connecting both J and K inputs together and supplying a logic HIGH (1). The modified JK Flip-Flop then functions as a T Flip-Flop. The circuit diagram is given below



- When T = 0, the flip-flop remains in its previous state.
- When T = 1, the flip-flop toggles its state on each clock pulse.

The characteristic equation is

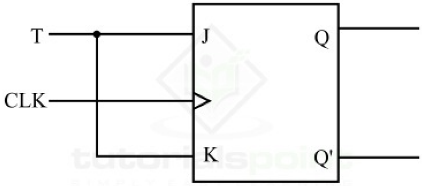
$$Q_{next} = J\overline{Q} + \overline{K}Q$$

Truth Table

Connect J and K inputs together

- J = K = T
- Clock signal remains the same as in the JK Flip-Flop

J	K	Q_n	Q_{n+1} (Next State)	Action
0	0	Q_n	Q_n	No Change
0	1	X	0	Reset
1	0	X	1	Set
1	1	Q_n	$\bar{Q}_n$	Toggle



3 T Flip-Flop

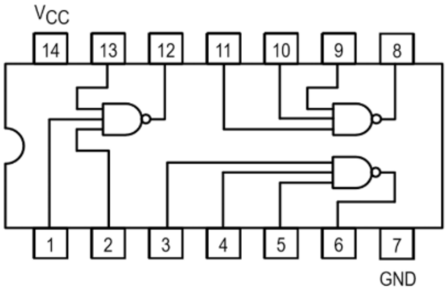
A T Flip-Flop is a type of sequential logic circuit used for counting and frequency division. It has two stable states: 0 and 1. The output of a toggles on each clock pulse when its T input is 1. The equation is given by

$$Q_{next} = T\overline{Q} + \overline{T}Q$$

Truth Table

T Input	Previous State (Q_n)	Next State (Q_{n+1})	Action
0	0	0	No Change
0	1	1	No Change
1	0	1	Toggle
1	1	0	Toggle

4 74LS10 IC



- The **74LS10 IC** contains **three independent 3-input NAND gates** in a single package.

- In this experiment, it is used to **detect when the counter reaches 7 (111 in binary)** and generate a reset signal.
- The output of the NAND gate goes **LOW when all three inputs are HIGH**, ensuring an automatic reset for the Mod-7 counter.
- Operates on a typical **supply voltage of 5V**, making it fully compatible with TTL circuits.
- Provides **fast response time and low power consumption**, making it ideal for digital logic applications.

5 Mod-7 Asynchronous Counter

A Mod-7 counter is a sequential digital circuit that counts from 0 to 6 (000 to 110 in binary) and then resets to 0. The number of flip-flops (N) needed for a Mod-M counter is determined by the formula

$$N = \lceil \log_2 M \rceil$$

For a Mod-7 counter ($M = 7$)

$$N = \lceil \log_2 7 \rceil = \lceil 2.81 \rceil = 3$$

Thus, 3 flip-flops are required to count from 000 (0) to 110 (6) in binary.

6 Connections

6.1 Setting Up the Breadboard and Power Connections

1) Place the **7476 JK Flip-Flop ICs** and the **74LS10 Triple 3-Input NAND Gate IC** on the breadboard.
2) Connect **Vcc (Pin 16) of both 7476 ICs and 74LS10 IC to 5V (Arduino or external supply)**.
3) Connect **GND (Pin 8) of both 7476 ICs and 74LS10 IC to Ground**.

6.2 Configuring JK Flip-Flops as T Flip-Flops

Each **JK Flip-Flop acts as a T Flip-Flop** when **J = K = 1 (HIGH)**, meaning the flip-flop toggles on every clock pulse.

6.2.1 Flip-Flop 1 (FF1) in 1st 7476 IC

1) Connect **Pin 2 (J) and Pin 4 (K) to 5V (HIGH)**.
2) Connect **Pin 3 (Clock) to Arduino Pin 9 (Clock Signal)**.
3) Connect **Pin 1 (Clear) to 5V** (active LOW, so we keep it HIGH).
4) Connect **Pin 5 (Q output) to Flip-Flop 2 Clock (Pin 11)**.
5) Connect **Pin 6 (Q') to Flip-Flop 2 Clock (Pin 11) for toggling**.

6.2.2 Flip-Flop 2 (FF2) in 1st 7476 IC

1) Connect **Pin 12 (J) and Pin 10 (K) to 5V (HIGH)**.
2) Connect **Pin 11 (Clock) to Q' of FF1 (Pin 6)**.
3) Connect **Pin 13 (Clear) to 5V**.
4) Connect **Pin 8 (Q) to Flip-Flop 3 Clock (Pin 3 of 2nd 7476 IC)**.
5) Connect **Pin 9 (Q') to Flip-Flop 3 Clock for toggling**.

6.2.3 Flip-Flop 3 (FF3) in 2nd 7476 IC

1) Connect **Pin 2 (J) and Pin 4 (K) to 5V (HIGH)**.
2) Connect **Pin 3 (Clock) to Q' of FF2 (Pin 9)**.
3) Connect **Pin 1 (Clear) to 5V**.
4) **Q output (Pin 5) will be the MSB of the counter**.

6.3 Implementing the Reset Logic Using 74LS10

To reset at count **7 (Q2 = Q1 = Q0 = 1)**:

1) Connect **Q outputs of FF1 (Pin 5), FF2 (Pin 8), and FF3 (Pin 5 of second IC) to the three inputs of a NAND gate (IC 74LS10 Pins 1, 2, 13)**.
2) Connect **NAND gate output (Pin 12 of 74LS10) to the Clear pins (Pin 1 and Pin 13) of both 7476 ICs**.
3) This ensures that when the counter reaches **111 (decimal 7)**, it resets to **000**.

6.4 Connecting LEDs for Output Visualization

1) Connect an **LED through a 220Ω resistor** to each **Q output** of the flip-flops:
   - **FF1: Pin 5 (Q) → LED1**
   - **FF2: Pin 8 (Q) → LED2**
   - **FF3: Pin 5 (Q) of 2nd IC → LED3**
2) This will allow visual observation of the **binary count (000 to 110 and then reset to 000)**.

6.5 Connecting Arduino for Clock Pulse Generation

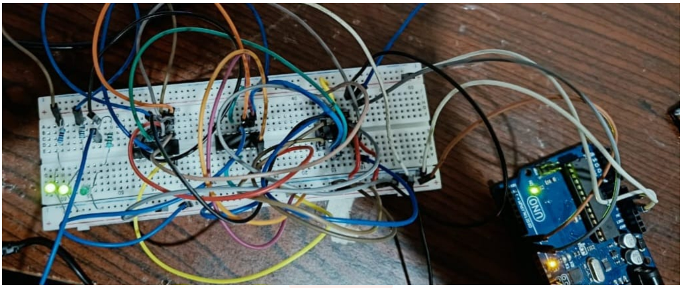
1) Connect **Arduino Pin 9** to **Clock input of FF1 (Pin 3 of first 7476 IC)**.
2) Upload the following Arduino code to generate clock pulses:

```
const int clockPin = 9; // Clock signal output pin
const int outputPin = 8; // Pin to be set HIGH

void setup() {
    pinMode(clockPin, OUTPUT);
    pinMode(outputPin, OUTPUT);
    digitalWrite(outputPin, HIGH); // Set pin 8 to HIGH
}
```

```
void loop() {
    digitalWrite(clockPin, HIGH);
    delay(5000); // 500ms HIGH
    digitalWrite(clockPin, LOW);
    delay(5000); // 500ms LOW
}
```

6.6 Breadboard Connections



7 Frequency Division in Asynchronous Counters

In an asynchronous counter that lacks a reset circuit (such as a NAND gate), each flip-flop inherently reduces the frequency of its preceding stage by a factor of two. The first flip-flop (FF1) operates at half the frequency of the clock signal, the second flip-flop (FF2) runs at one-fourth, and the third flip-flop (FF3) functions at one-eighth of the original clock frequency. This cascading effect emerges as each flip-flop toggles only on the falling edge of the preceding stage's output, leading to progressive frequency division. The resulting binary sequence produced by the counter outputs $(Q_0, Q_1, Q_2, ...)$ represents this hierarchical frequency reduction, where each bit corresponds to a power-of-two fraction of the input clock.

8 Results

Without a NAND Gate Reset Mechanism

8.0.1 Operation of the First Flip-Flop (FF1)

The first flip-flop operates by toggling its state at every clock pulse. As a result, its output frequency is:

$$f_{Q_0} = \frac{f_{clk}}{2} \tag{2.1}$$

8.0.2 Operation of the Second Flip-Flop (FF2)

The second flip-flop transitions on the negative edge of Q_0, meaning it changes state every two clock cycles, resulting in:

$$f_{Q_1} = \frac{f_{Q_0}}{2} = \frac{f_{clk}}{4} \tag{2.2}$$

8.0.3 Operation of the Third Flip-Flop (FF3)

Similarly, the third flip-flop toggles whenever Q_1 transitions from high to low, leading to a frequency output of:

$$f_{Q_2} = \frac{f_{Q_1}}{2} = \frac{f_{clk}}{8} \tag{2.3}$$

In general, for an asynchronous counter with n flip-flops, the frequency of the n^{th} stage is given by:

$$f_{Q_n} = \frac{f_{clk}}{2^{(n+1)}} \tag{2.4}$$

where n is indexed from 0 for the first flip-flop.

8.0.4 Time Period Computation

The clock signal has a time period defined as:

$$T_{clk} = \frac{1}{f_{clk}} \tag{2.5}$$

For each subsequent stage, the time period follows this pattern:

$$T_{Q_0} = 2T_{clk} \Rightarrow f_{Q_0} = \frac{1}{2T_{clk}} = \frac{f_{clk}}{2} \tag{2.6}$$

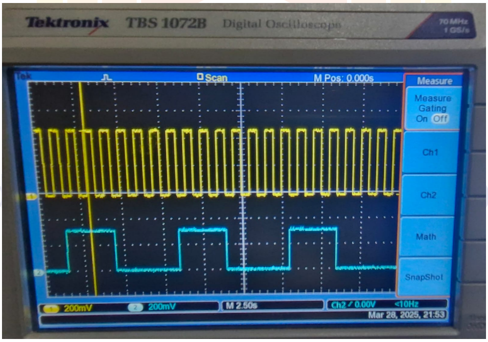
$$T_{Q_1} = 2T_{Q_0} = 4T_{clk} \Rightarrow f_{Q_1} = \frac{1}{4T_{clk}} = \frac{f_{clk}}{4} \tag{2.7}$$

$$T_{Q_2} = 2T_{Q_1} = 8T_{clk} \Rightarrow f_{Q_2} = \frac{1}{8T_{clk}} = \frac{f_{clk}}{8} \tag{2.8}$$

This mathematical validation confirms that, in the absence of a reset circuit like a NAND gate, the counter follows a natural binary frequency division:

$$f_{Q_0} = \frac{f_{clk}}{2}, \quad f_{Q_1} = \frac{f_{clk}}{4}, \quad f_{Q_2} = \frac{f_{clk}}{8}, \quad \ldots \tag{2.9}$$

With NAND Gate



We observe that the obtained frequency is

$$\frac{f}{7}$$

$$f_{\text{out}} = \frac{f_{\text{in}}}{M}$$

where $M = 7$ (modulus of the counter).

Thus,

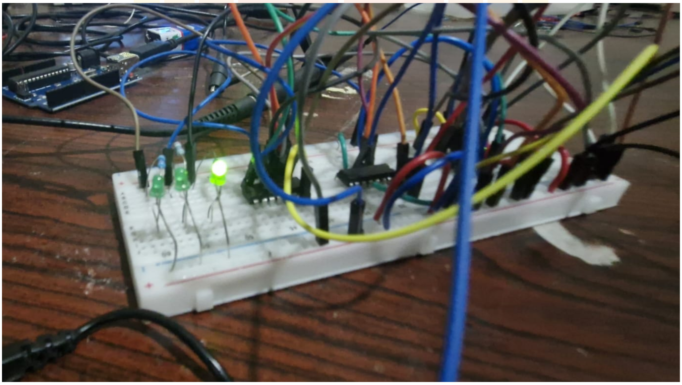
$$f_{\text{out}} = \frac{f_{\text{in}}}{7}$$

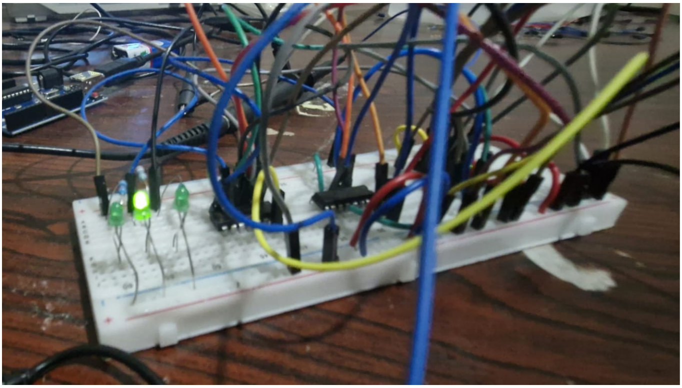
Thus, it is verified

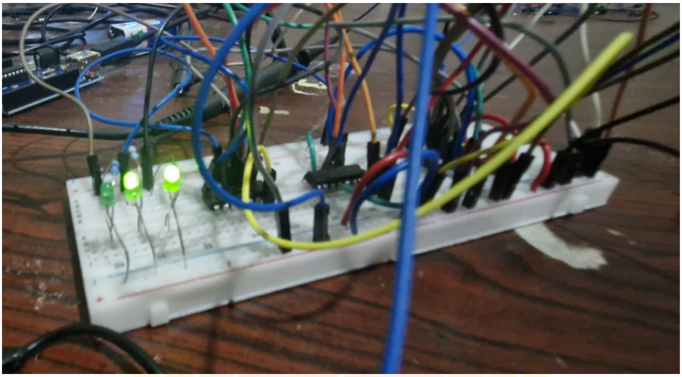
9 Verification using LED's

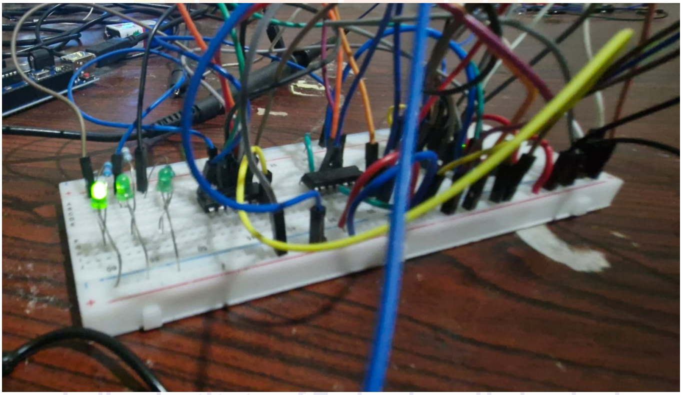
To verify the behavior of the Mod-7 asynchronous counter with a NAND gate reset, an LED display was used to visualize the state transitions of the flip-flops. The objective was to observe how the output frequency changed at each stage due to the counter operation and the reset mechanism implemented using the 3-input NAND gate (IC 74LS10).

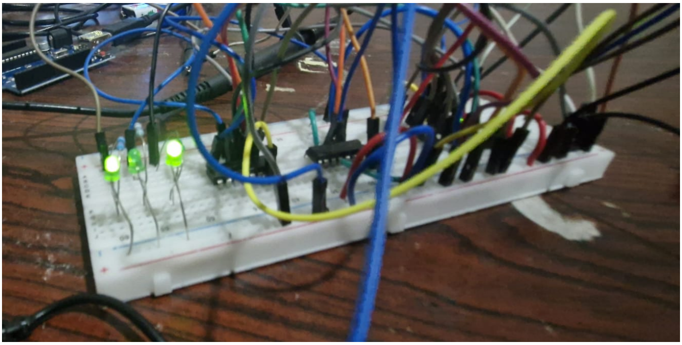
- The Q outputs of each T Flip-Flop were connected to individual LEDs to display the counter's binary states.
- The LEDs toggled according to the counter's binary sequence (000 to 110), allowing real-time observation of the frequency division pattern.
- When the counter reached 111 (7 in decimal), the NAND gate triggered a reset, forcing the counter back to 000.

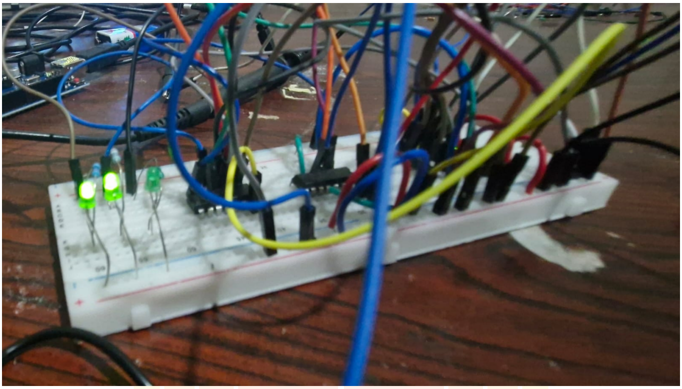


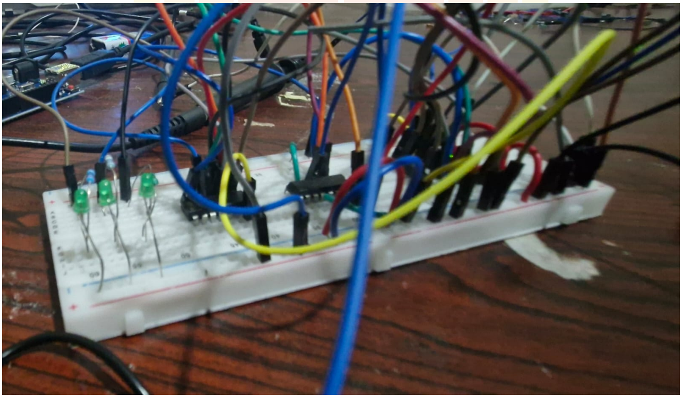












Decimal (Count Value)	**Binary** (Q_2 Q_1 Q_0)
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7 (Reset)	111 (Detected by NAND, resets to 000)