Design and Implementation of a Mod-7 Asynchronous Counter

EE24BTECH11021-ESHAN RAY
EE24BTECH11048-NITHIN.K

March 27, 2025

Abstract

This lab report presents the design, implementation, and testing of a Mod-7 asynchronous counter using JK flip-flops. The counter operates with an external clock signal provided by an Arduino and its performance is verified using a Cathode Ray Oscilloscope (CRO).

1 Introduction

A Mod-7 counter is a sequential circuit that counts from 0 to 6 and then resets to 0. Using JK flip-flops in a toggle mode, the counter progresses through seven states. The asynchronous nature means that each flip-flop's clock input is triggered by the output of the previous flip-flop.

2 Components and Equipment

- JK Flip-Flops (e.g., 7476 IC)
- NAND gate(e.g. , 7410 IC)
- Arduino (as a clock source)
- 3 LED's
- Cathode Ray Oscilloscope (CRO)
- Resistors and Capacitors (as needed)
- Connecting Wires and Breadboard
- Power Supply (5V DC)

3 Circuit Design

The Mod-7 counter is implemented using three JK flip-flops. The JK flip-flops are configured in toggle mode by connecting both J and K inputs to logic HIGH (1). The clock for the first flip-flop is provided by the Arduino, while subsequent flip-flops receive their clock signal from the previous flip-flop's output.

3.1 Conversion of JK Flip-Flop to T Flip-Flop

A JK Flip-Flop can be converted into a T Flip-Flop by making the following modifications:

T	Q (Previous State)	Q (Next State)
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: Truth Table for T Flip-Flop

J	K	Q (Previous State)	Q (Next State)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Table 2: Truth Table for JK Flip-Flop

Q	T	J	K	Q (Next State)
0	0	0	X	0
0	1	1	X	1
1	0	X	0	1
1	1	X	1	0

Table 3: Truth Table for JK to T Flip-Flop

1. Connect both J and K inputs together.
2. Use a single toggle (T) input instead of separate J and K inputs.
3. The T input determines the flip-flop's behavior:
    - When $T = 0$, the output remains unchanged.
    - When $T = 1$, the output toggles.

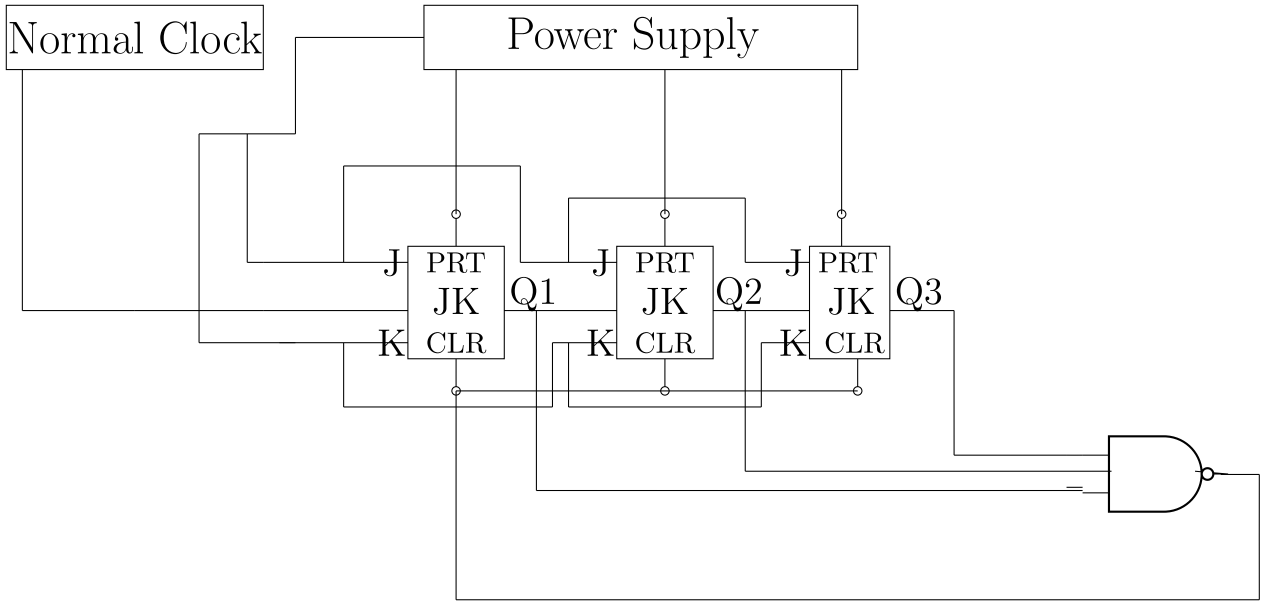
Since the Mod-7 counter requires toggle functionality, we connect J = K = 1 for each flip-flop, effectively making them act as T Flip-Flops.

3.2 State Transition Table

Current State			Next State		
Q3	Q2	Q1	Q3	Q2	Q1
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0

Table 4: State Transition Table for Mod-7 Counter

3.3 Circuit Diagram



Circuit Diagram for Mod-7 Counter using JK Flip-Flops

4 Implementation

The Arduino generates a clock signal using the following code:

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

void loop() {
    digitalWrite(CLOCK_PIN, HIGH);
    delay(100);
    digitalWrite(CLOCK_PIN, LOW);
    delay(100);
}
```

5 Procedure

1. **Circuit Assembly**

(a) Connect three JK flip-flops in cascade to form the Mod-7 counter.

(b) Configure each JK flip-flop in toggle mode by connecting $J = 1$ and $K = 1$.

(c) Use a NAND gate to reset the counter when the state reaches 7 (111).

(d) Connect LEDs to the output of each flip-flop to visualize the counting sequence.

(e) Provide power (5V) and ground connections to all ICs.

2. **Clock Signal Generation**

   (a) Use an Arduino to generate a square wave clock signal at a fixed frequency.

   (b) Connect the Arduino clock output to the clock input of the first flip-flop.

   (c) Verify the clock signal using an oscilloscope.

3. **Counter Operation Testing**

   (a) Power the circuit and observe the LED sequence.

   (b) Verify the counting sequence from 000 to 110 and check that it resets to 000.

   (c) Use a CRO to capture and analyze the waveforms of the clock and output states.

   (d) Measure the frequency division to confirm $f_{Q1} = \frac{f_{clk}}{2}$, $f_{Q2} = \frac{f_{clk}}{4}$, and $f_{Q3} = \frac{f_{clk}}{8}$.

4. **Analysis and Verification**

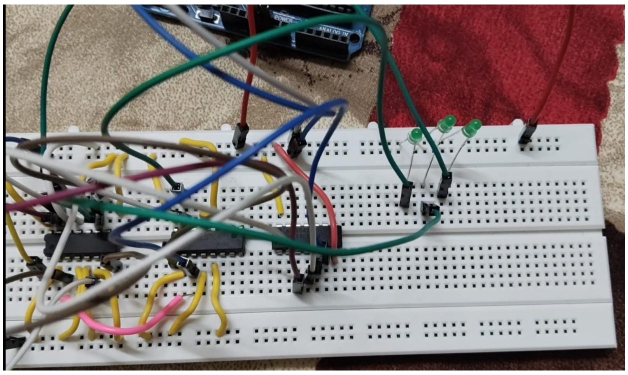
   (a) Compare the observed outputs with the theoretical state transition table.

   (b) Ensure proper synchronization of flip-flops and confirm correct reset behavior.

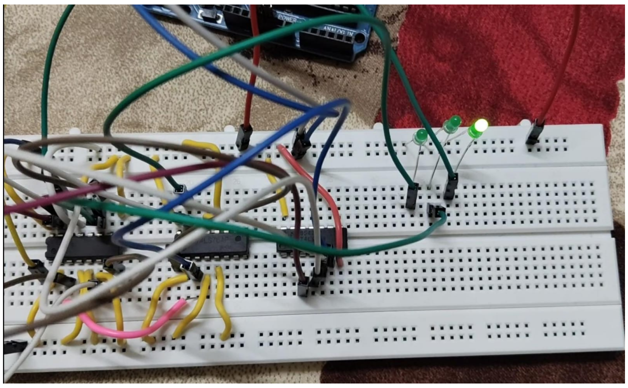
6 Observations and Results

The counter output was verified using a Cathode Ray Oscilloscope (CRO), showing the expected sequence from 000 to 110 before resetting. The waveform analysis confirmed the correct operation of the Mod-7 counter. The

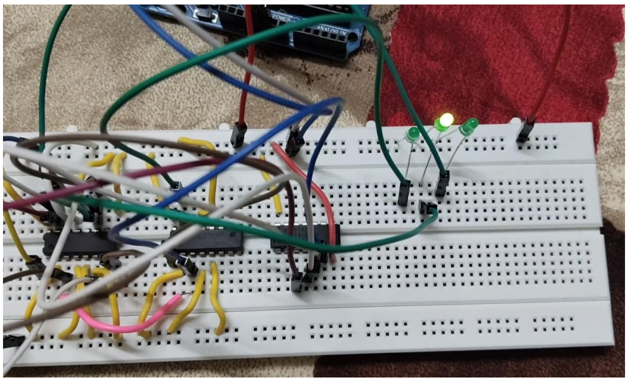
below pictures are of the LED counter:
The LED blinking represents 1 and the order is Q3, Q2, Q1 from left to right:



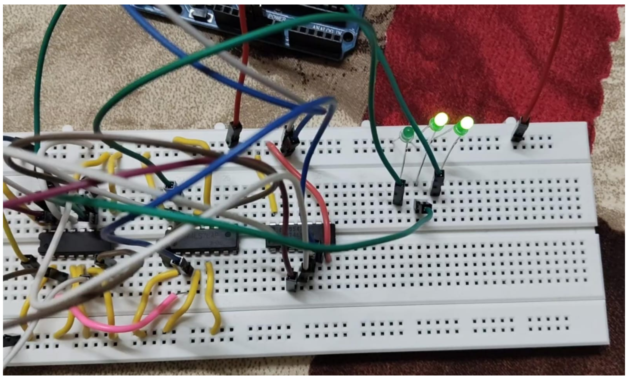
0 0 0 = 0



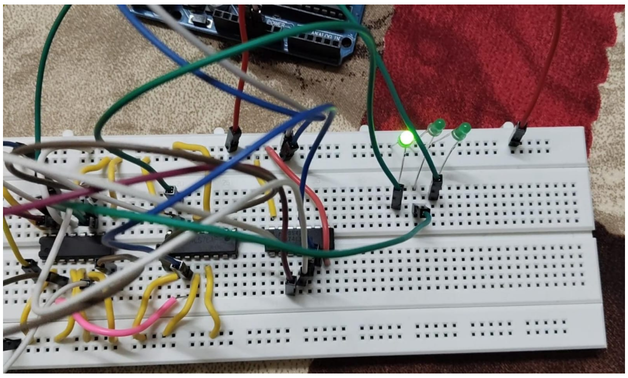
0 0 1 = 1



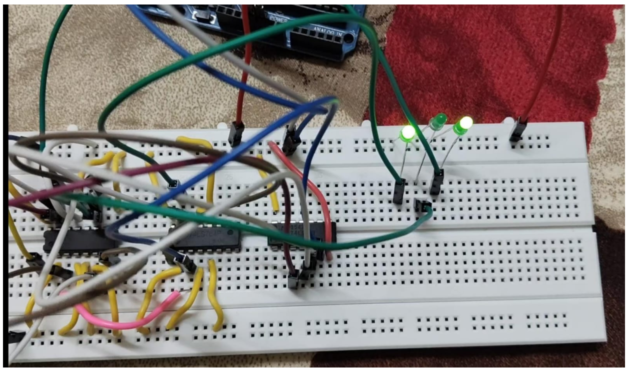
0 1 0 = 2



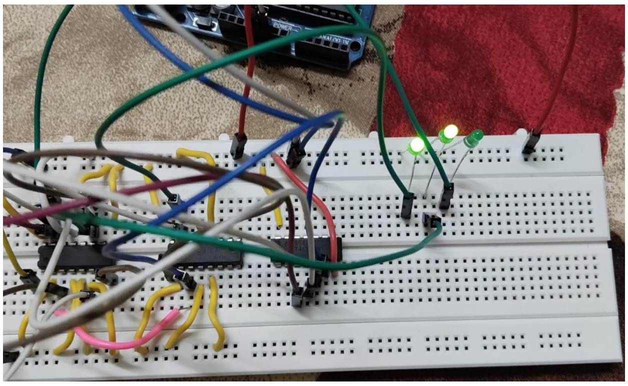
$0\ 1\ 1 = 3$



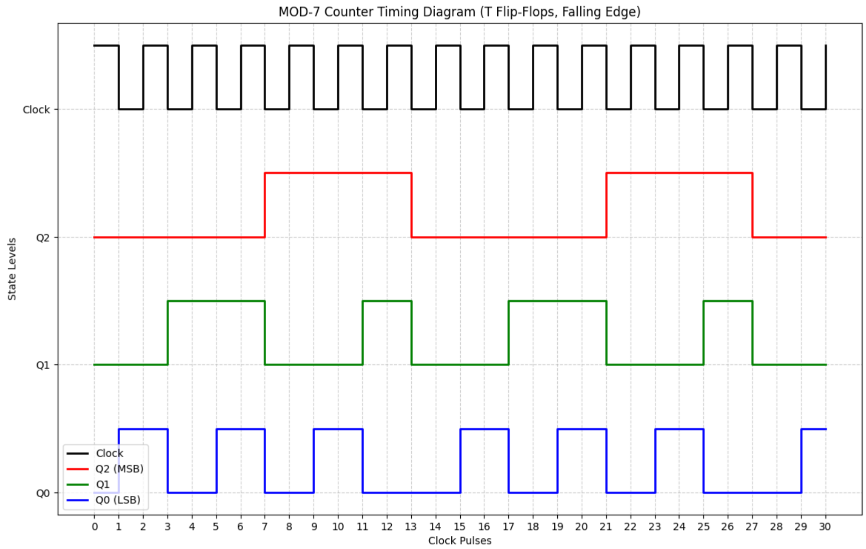
$1\ 0\ 0 = 4$



$1\ 0\ 1 = 5$



$1\ 1\ 0 = 6$


MOD-7 Counter Timing Diagram (T Flip-Flops, Falling Edge)
State Levels
Clock
Q2
Q1
Q0
Clock
Q2 (MSB)
Q1
Q0 (LSB)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Clock Pulses

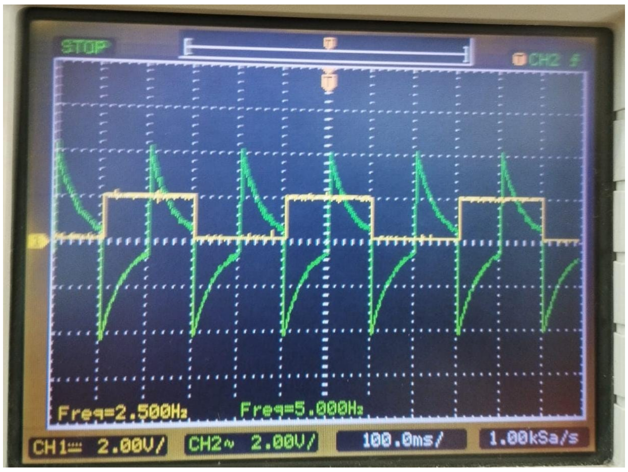


Figure 1: Clock and Q1

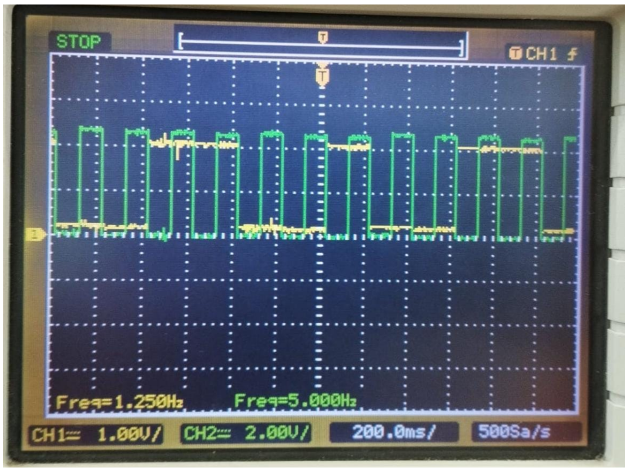


Figure 2: Clock and Q2

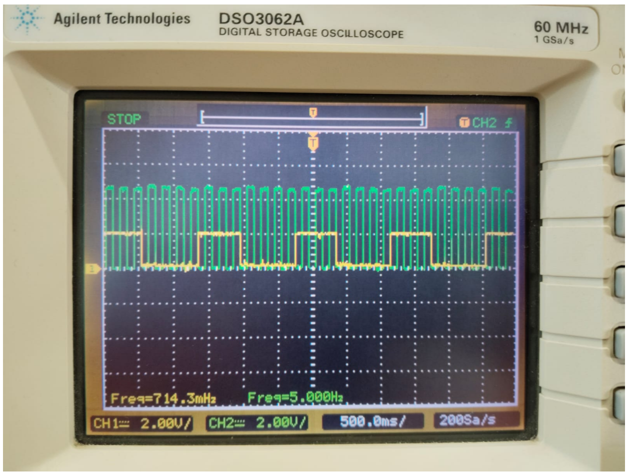


Figure 3: Clock and Q3

7 Conclusion

The Mod-7 asynchronous counter was successfully designed and tested using JK flip-flops. The circuit counted correctly through seven states and reset as expected. The Arduino provided a stable clock signal, and the output was verified using a CRO.