Experiment Report: Study of Counter Circuits

Abhijnan Saraswat Gogoi
Roll no.: 2411072
PHY241 Course

12th November, 2025

Abstract

In this experiment, MOD-16 ripple up and down counters and a MOD-12 ripple up counter were constructed. The circuits were driven using a TTL-compatible square-wave clock input. The MOD-16 counters demonstrated the characteristic binary counting sequence and frequency division inherent to asynchronous counters built from JK flip-flops configured in toggle mode. The MOD-12 counter was implemented by integrating a 7420 NAND gate to detect the 1100 state and reset the system, yielding a 12-state sequence. The observed behavior matched theoretical predictions, with propagation delays small enough to be negligible in visual LED-based observations. The experiment highlights the practical operation of ripple counters, the role of gating logic in achieving non-binary moduli, and the importance of using clean TTL-compatible clock signals to avoid unstable or unpredictable behavior.

1 Objectives

To construct and study the working of the following circuits:

- MOD-16 ripple up counter.
- MOD-16 ripple down counter.
- MOD-12 ripple up counter.

2 Theory

Counters are sequential logic circuits that generate a specific sequence of states in response to clock pulses. They are widely used in digital systems for counting events, frequency division, timing applications, and digital clocks. A counter typically consists of a series of flip-flops connected in a manner that allows their outputs to represent binary numbers.

The JK flip-flop is a bistable device used to store a single bit of information. It has two inputs, J and K, and one clock input. Its characteristic behavior is given by the truth table:

J	K	Next State
0	0	No change
0	1	Reset (0)
1	0	Set (1)
1	1	Toggle

When used in counters, JK flip-flops are typically configured in toggle mode ($J = K = 1$), causing them to change state on every active clock edge. By connecting multiple flip-flops in series, binary counters of various moduli can be constructed.

2.1 Ripple Counters

A ripple counter (asynchronous counter) is formed by connecting the output of one flip-flop to the clock input of the next. The least significant bit (LSB) flip-flop receives the external clock, and each subsequent flip-flop toggles when the previous one transitions between states. The name *ripple* comes from the way state changes propagate sequentially through the flip-flops.

For a 4-bit ripple counter, the outputs Q_0, Q_1, Q_2, Q_3 represent a binary number from 0 to 15. The direction of counting depends on whether the flip-flops toggle with

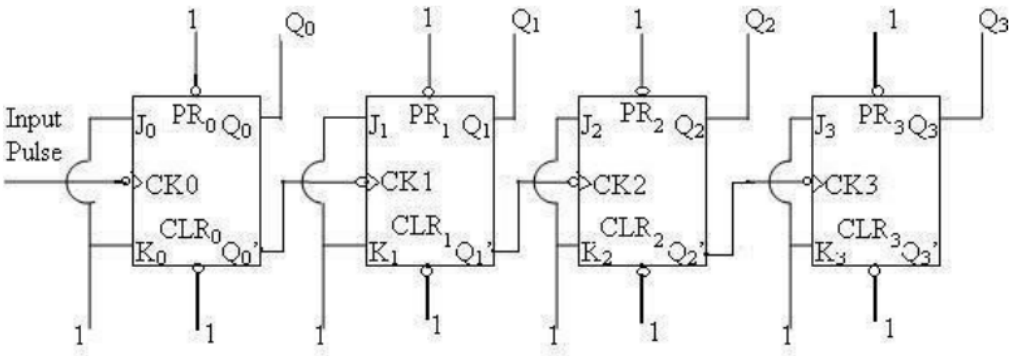

Figure 1: Circuit diagram for MOD-16 ripple down counter

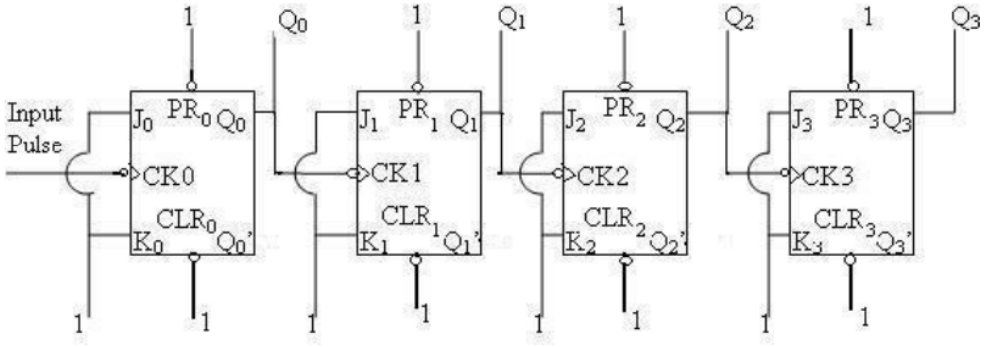

Figure 2: Circuit diagram for MOD-16 ripple up counter

respect to positive or negative transitions of the preceding outputs:

- **Up Counter:** Each flip-flop toggles on the positive-going edge of the output of the previous stage. The binary value increments with each clock pulse.
- **Down Counter:** Each flip-flop toggles on the negative-going edge of the previous stage. The binary value decrements with each pulse.

This difference ensures that the binary sequence progresses forward or backward as required.

2.2 MOD-n Counters

A MOD-n counter cycles through n distinct states before resetting to zero. For a 4-bit binary counter, the natural count is from 0 to 15 (MOD-16). To create counters with smaller moduli, additional logic must be used to detect specific states and reset the counter.

2.2.1 MOD-16 Counter

A MOD-16 counter uses two flip-flops, producing the sequence 0–1–2–3–...–15 and then resetting to 0. In the up-counter configuration, the state advances in ascending order; in the down-counter configuration, it decreases from 15 to 0. JK flip-flops remain in toggle mode, and the modulus is achieved by using four stages.

2.2.2 MOD-12 Counter

A MOD-12 counter requires four flip-flops (capable of counting up to 15), along with extra combinational logic to detect when the count reaches decimal 12 (1100_2). Upon detecting this state, the counter is asynchronously cleared to return to 0000. In this experiment, a 4-input NAND gate (IC 7420) is used to detect the reset condition. When $Q_3 = 1$ and $Q_2 = 1$ and $Q_1 = 0$ and $Q_0 = 0$, the NAND gate output transitions low, triggering the clear inputs of the flip-flops and returning the counter to zero.

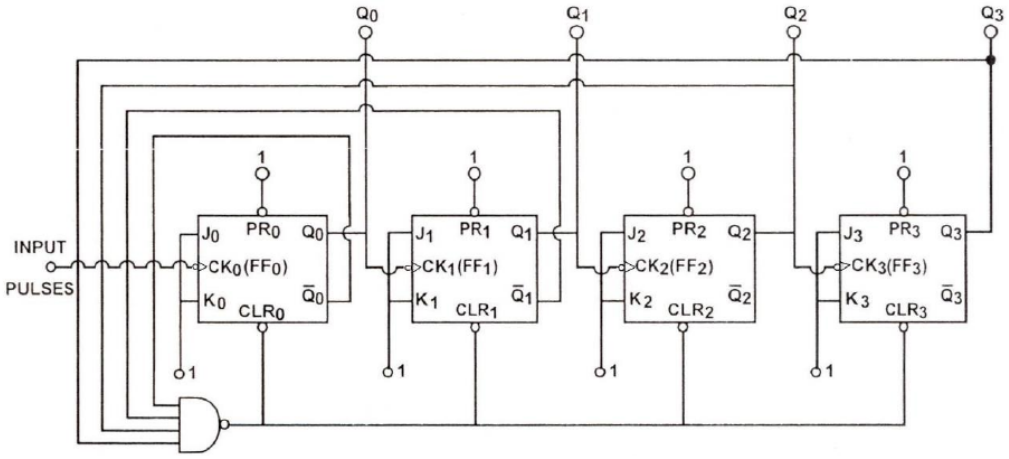

Figure 3: Circuit diagram for MOD-12 ripple up counter

3 Materials Used

(i) ICs (JK FF-7476 or 74112, 2 Nos; 4-input NAND-7420, 1 No), (ii) LEDs (four), (iii) 1kΩ resistors (four), (iv) breadboard, (v) connecting wires, (vi) function generator, (vii) 5V DC power supply.

4 Observations

Table 1: MOD-16 Binary Ripple Down Counter

Input Pulse	Binary Count Q_3	Q_2	Q_1	Q_0	Decimal Count
0	0	0	0	0	0
1	1	1	1	1	15
2	1	1	1	0	14
3	1	1	0	1	13
4	1	1	0	0	12
5	1	0	1	1	11
6	1	0	1	0	10
7	1	0	0	1	9
8	1	0	0	0	8
9	0	1	1	1	7
10	0	1	1	0	6
11	0	1	0	1	5
12	0	1	0	0	4
13	0	0	1	1	3
14	0	0	1	0	2
15	0	0	0	1	1
16	0	0	0	0	0

Table 2: MOD-16 Binary Ripple Up Counter

Input Pulse	Binary Count Q_3	Q_2	Q_1	Q_0	Decimal Count
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15
16	0	0	0	0	16

Table 3: MOD-12 Binary Ripple Up Counter

Input Pulse	Binary Count Q_3	Q_2	Q_1	Q_0	Decimal Count
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	0	0	0	0	0 (Reset)

5 Results

The observations of the LEDs for the three counter circuits are given in tables 1, 2 and 3. The MOD-16 and MOD-12 counters operated correctly under the clock inputs. The LED outputs followed the expected binary sequences for both up-counting and down-counting configurations for MOD-16. In the MOD-12 counter, the state returned to 0000 after reaching 1100, confirming that the external decoding using the 7420 NAND gate successfully generated the required asynchronous reset. No irregular transitions or missed toggles were observed during normal operation.

6 Discussion

The experimental results match the theoretical behavior of asynchronous (ripple) counters constructed using JK flip-flops configured in toggle mode. In the MOD-16 counters, the propagation of state changes from the least significant flip-flop to the most significant one produced the characteristic ripple effect, where each successive stage toggles at half the frequency of the previous stage. Although propagation delays are inherent to ripple counters, the delays were too small to affect the LED observations.

The MOD-12 counter demonstrated how additional combinational logic can be used to implement non-binary moduli. By detecting the count 1100 using the 7420 NAND gate, the system reset the flip-flops to zero, yielding a 12-state sequence instead of the natural 16-state sequence of a 4-bit counter. This matches the expected behavior of decoded asynchronous counters and confirms the effectiveness of integrating flip-flops with external gating logic.

The observations recorded in tables 1, 2, and 3 were obtained using a TTL-compatible square-wave clock signal. Earlier trials performed using a non-TTL clock waveform resulted in irregular toggling and unstable behaviour in the counters. This instability is likely due to improper logic-level thresholds or non-ideal rise and fall times in the input waveform, causing the flip-flops to misinterpret the clock edges. Using a proper TTL-level square wave ensured clean triggering and reliable operation of the counters.

7 Conclusions

The experiment successfully demonstrated the use of JK Flip-Flops to create MOD-16 up and down ripple counters and MOD-12 ripple up counters using sequential and combinatorial logic.

8 Precautions

1. Since the counters used in this experiment are asynchronous, the clock frequency must be kept sufficiently low to avoid errors caused by propagation delays between flip-flops.
2. The input clock should be a clean square wave with proper logic-level voltages (preferably TTL-compatible). Irregular or non-square waveforms can introduce false triggering or unstable counting.
3. All connections must be secure. Open or floating inputs in 7400-series ICs may be interpreted as HIGH, leading to incorrect or unpredictable counter behaviour.
4. Ensure a common ground of the function generator and the circuit.
5. Verify the supply voltage (5V) before powering the ICs. Over-voltage or unstable supply can damage 7400-series logic chips.