Design and implementation of an asynchronous MOD-60 counter

Objective

1. To design an asynchronous MOD-60 counter using only basic logic gates and flip-flops.
2. To simulate the counter circuit to verify its functionality.
3. To analyze the performance of the circuit based on simulation results.

Theory

The design of an asynchronous MOD-60 counter involves creating a circuit that can count from 0 to 59 (60 states in total) and then reset to 0. This counter can be realized by cascading two counters: a MOD-10 (decade counter) and a MOD-6 counter. Each counter is constructed using JK flip-flops and logic gates.

(1) **Asynchronous Counter**

In an asynchronous counter, the output of each flip-flop (FF) is used as the clock input for the next flip-flop. This design means that the flip-flops do not respond to the external clock pulses in exact synchrony; instead, their state changes sequentially.

- **First Flip-Flop**: In an asynchronous counter, the first flip-flop is directly driven by external clock pulses. Therefore, its state change is strictly synchronized with the external clock.
- **Subsequent Flip-Flops**: Starting from the second flip-flop, the clock input for each flip-flop comes from the output of the previous one. This means that each flip-flop's state change depends on the state of its predecessor, rather than directly responding to the external clock pulses.

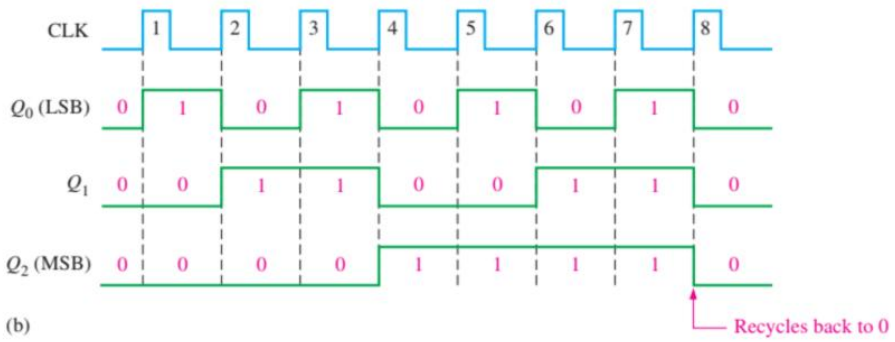

Fig 2-1 The timing diagram for an asynchronous counter

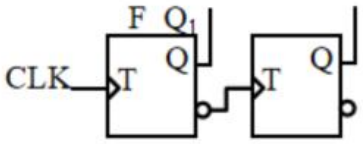

Fig 2-2 A 2-bit asynchronous counter

(2) Mod-N Counter

The MOD number of a counter is defined as the number of unique states it cycles through before returning to its initial state. For example, a MOD-6 counter will go through six states, typically from 000 to 110 in binary, before resetting back to 000.

A standard N-bit binary counter will naturally count from 0 to $2^N - 1$ before rolling over to 0. However, to implement a MOD number that is less than 2^N, additional logic is needed to reset the counter early. This is commonly achieved using combinational logic that monitors the counter's outputs and generates a reset signal when a certain count, which represents the MOD number, is reached.

1. **Flip-Flop Configuration:** Start by configuring the flip-flops in toggle mode, where each flip-flop toggles its output state with every clock pulse it receives.

2. **Detection Circuit:** Implement a detection circuit that monitors the outputs of the flip-flops and detects when the counter reaches the MOD number.

3. **Reset Signal Generation:** Use the output of the detection circuit to generate a reset signal. This signal is then fed to the reset inputs of all flip-flops, causing the counter to return to 0.

1) **Mod 10**

The MOD-10 counter, characterized by its ability to count ten distinct states from 0000 (0 in decimal) to 1001 (9 in decimal), is commonly referred to as a decade counter. It is also termed a Binary-Coded Decimal (BCD) counter due to its representation of decimal numbers in binary form.

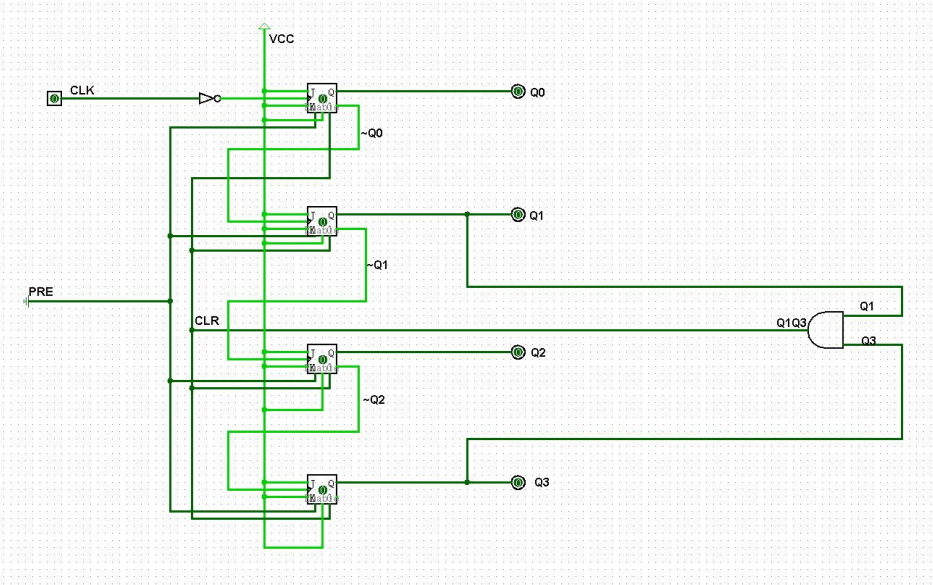

Fig 2-3 Mod-10 counter Circuit Diagram

The circuit diagram Fig 2-4 illustrates a 4-bit binary counter composed of four flip-flops, each representing a single bit in the binary count sequence. The flip-flops are arranged in a typical ripple counter configuration where the output of each flip-flop serves as the clock input to the next flip-flop in the series.

Circuit Diagram Description:

- **JK Flip-Flops:** Each flip-flop has a clear (CLR) and preset (PRE) input, along with complementary outputs (Q and not-Q). The clear and preset inputs can be used to reset or set the counter to a particular state.
- **Clock Input (CLK):** The clock input is connected to the first flip-flop, initiating the count sequence.
- **Output States (Q0, Q1, Q2, Q3):** The outputs are labeled for each bit, from the least significant bit (Q0) to the most significant bit (Q3). These outputs will

change state with each clock pulse, counting up in binary from 0 (0000) to 9 (1100).

- **Reset Input(CLR):** The CLR input responds to an external signal to immediately set the counter's output to zero. An AND gate connected to specific outputs (in this case, Q1 and Q3) is used to detect the reset condition and generate the required clear signal. When the counter reaches the binary count of 1010 (decimal 10), which is a temporary state, the AND gate outputs a high signal that activates the CLR input and resets the counter.

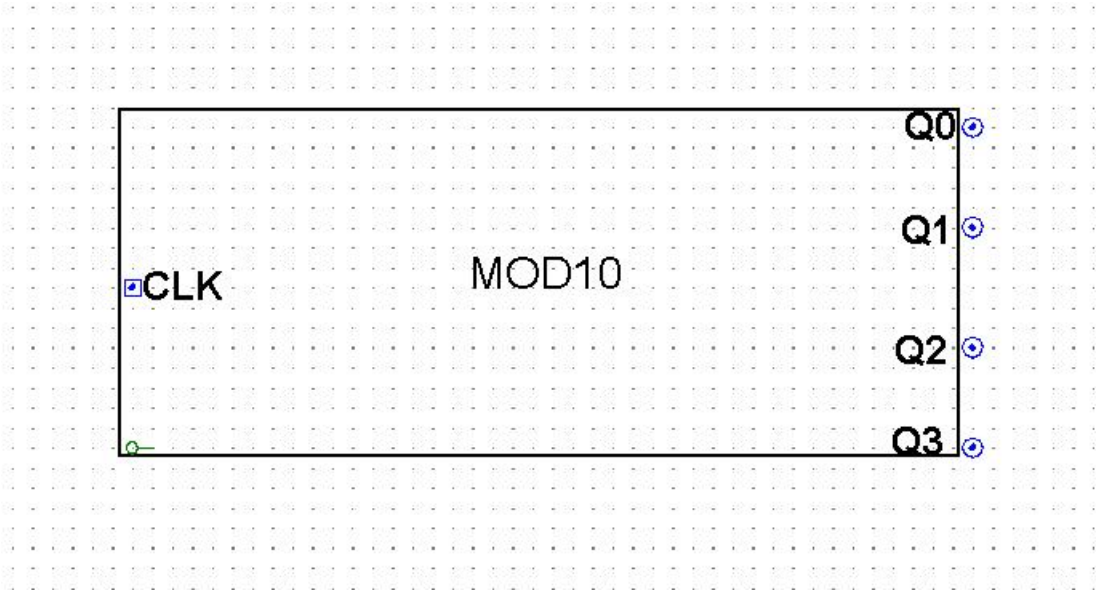

Fig 2-4 Package diagram of a Mod-10 counter

Fig 2-4 shows the package diagram of a MOD-10 counter, encapsulating the complexity of the internal circuitry within a simple rectangular symbol for ease of use in system-level diagrams.

2) Mod 6

The MOD-6 counter is designed to count six distinct states from 000 (0 in decimal) to 101 (5 in decimal) and is an essential component for applications requiring counting in a range less than a full decade. Unlike the MOD-10 counter, the MOD-6 counter must recycle back to zero after reaching the sixth count instead of the tenth.

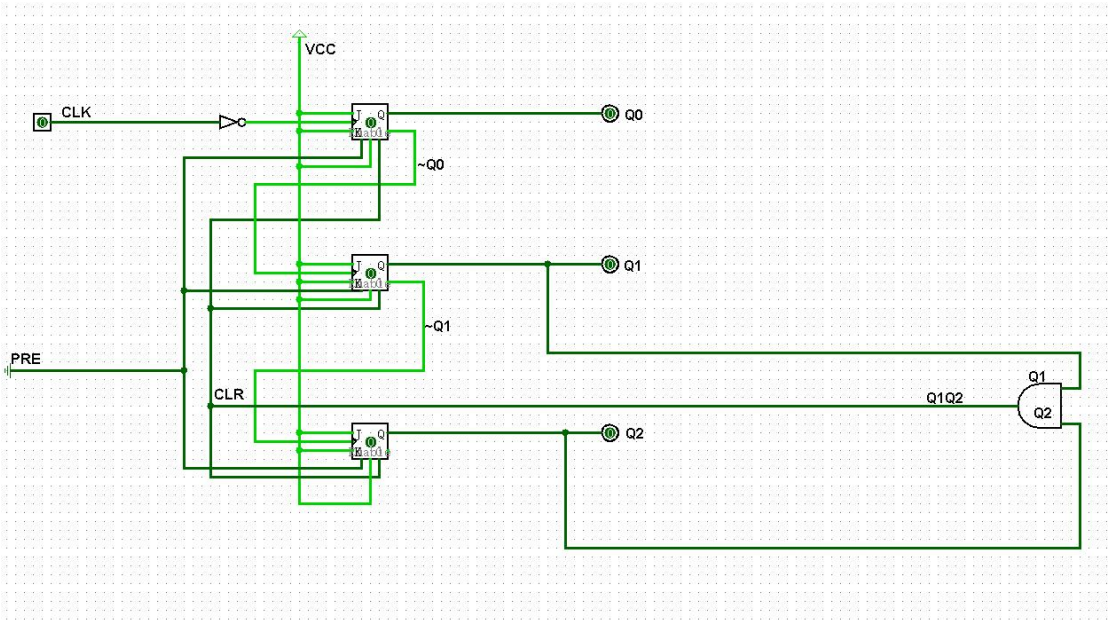

Fig 2-5 Mod-6 counter Circuit Diagram

The circuit diagram Fig 2-5 illustrates a 3-bit binary counter composed of three flip-flops, each representing a single bit in the binary count sequence. The flip-flops are arranged in a typical ripple counter configuration where the output of each flip-flop serves as the clock input to the next flip-flop in the series.

Circuit Diagram Description:

- **JK Flip-Flops:** Similarly to the MOD-10, each flip-flop would have clear (CLR) and preset (PRE) inputs for state control.
- **Clock Input (CLK):** The CLK input would be connected to the first flip-flop in the series, initiating the count sequence.
- **Output States (Q0, Q1, Q2):** Outputs for each bit, with Q0 being the least significant bit and Q2 the most significant bit, would represent the binary count. The counter would increment its binary value from 0 (000) to 5 (101).
- **Reset Logic for MOD-6 Counter**: Reset Condition: When both Q1 and Q2 are HIGH, the AND gate would output a HIGH signal, which, when connected to the CLR input of the flip-flops, would reset the counter back to zero.

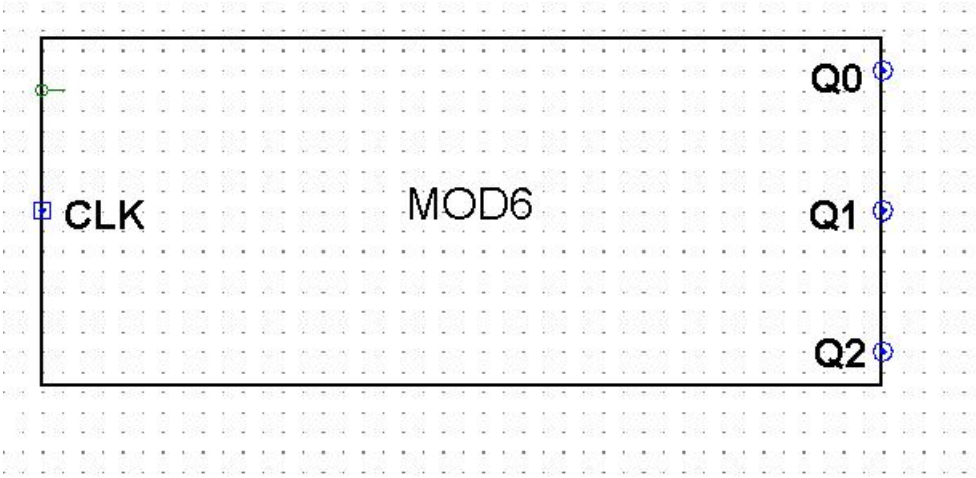

Fig 2-6 Package diagram of a Mod-6 counter

Fig 2-6 shows the package diagram of a MOD-6 counter, encapsulating the complexity of the internal circuitry within a simple rectangular symbol for ease of use in system-level diagrams. The counter is represented as a single module labelled 'MOD6', which abstracts the internal configuration of flip-flops and logic gates used to create the Mod6 counting functionality.

(3) Cascade Counter

To construct a MOD-60 counter, a MOD-10 counter and a MOD-6 counter are cascaded. The MOD-10 counter acts as the least significant digit, counting from 0 to 9, while the MOD-6 counter represents the most significant digit, counting from 0 to 5. Upon the MOD-10 counter's transition from 9 back to 0, it triggers the MOD-6 counter to increment, thus creating a combined count that cycles from 00 to 59 before resetting to 00.

The increment trigger from the MOD-10 to the MOD-6 counter is typically implemented by connecting the carry-out or a specific state output (like when all bits are high) of the MOD-10 counter to the clock input of the MOD-6 counter.

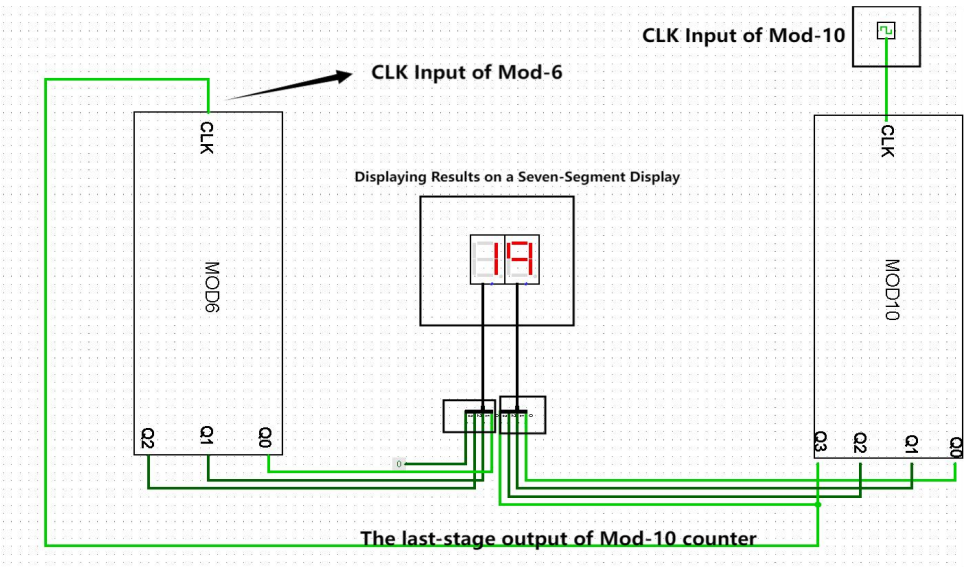

Fig 2-7 A cascaded counter of MOD-6 counter and MOD-10 counter circuit diagram

The circuit diagram Fig 2-7 would illustrate the interconnected MOD-10 and MOD-6 counters, showing the flow of signals between them and the logic involved in their cascading.

List of apparatus/components

Below are the lists of the components used in this circuit.

Gate	Quantity
2-Input AND Gate	2
Inverter	2
JK flip-flop	7
Hex Digit Display	2

Table 2-1 Gate List

IC No	Description	Quantity
CD4081	Quad 2-Input AND Gate	2
74HC112	Dual JK flip-flop with set and reset; negative-edge trigger	3
CD4511	BCD-to-7-Segment LED Latch Decoder	1

Table 2-2 IC List

Result

Q3	Q2	Q1	Q0	Decimal	Next State Action
0	0	0	0	0	Increment
0	0	0	1	1	Increment
0	0	1	0	2	Increment
0	0	1	1	3	Increment
0	1	0	0	4	Increment
0	1	0	1	5	Increment
0	1	1	0	6	Increment
1	0	0	0	7	Increment
1	0	0	1	8	Increment
1	0	1	0	9	Reset to 0

Fig 2-8 The truth table for the MOD-10 counter

The truth table in Fig 2-8 details the state of a 4-bit binary counter (MOD-10) that counts from 0 to 9 in decimal. The 'Next State Action' column indicates that the counter will increment on each clock pulse and reset to 0 after reaching the count of 9.

Q2	Q1	Q0	Decimal	Next State Action
0	0	0	0	Increment
0	0	1	1	Increment
0	1	0	2	Increment
0	1	1	3	Increment
0	0	0	4	Increment
1	0	1	5	Reset to 0

Fig 2-9 The truth table for the MOD-6 counter

The truth table in Fig 2-9 shows the states for a MOD-6 counter, which is a 3-bit binary counter counting from 0 to 5 in decimal. The 'Next State Action' column indicates that the counter will increment on each clock pulse and reset to 0 after reaching the count of 6.

Figure 2-10 and 2-11 illustrate the counter immediately after a reset has been triggered, with both displays showing '0'. This state demonstrates the counter's ability to reset correctly and start a new counting cycle. The reset mechanism, as designed, correctly identifies the transition from '59' to '60' and initializes both the MOD-6 and MOD-10 counters, returning the display to '00'.

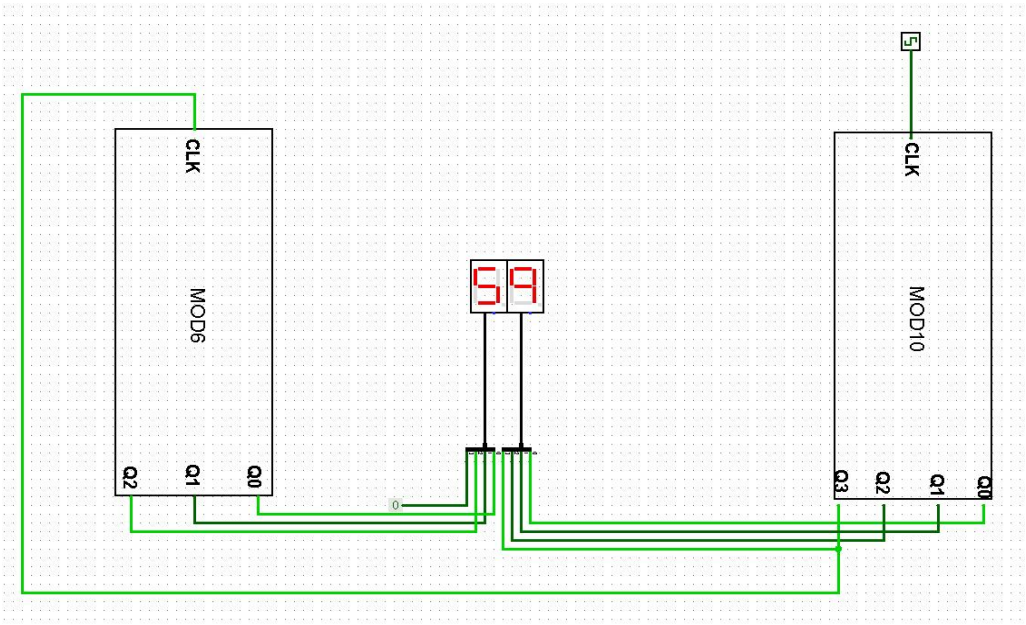

Fig 2-10 Final State Before Reset

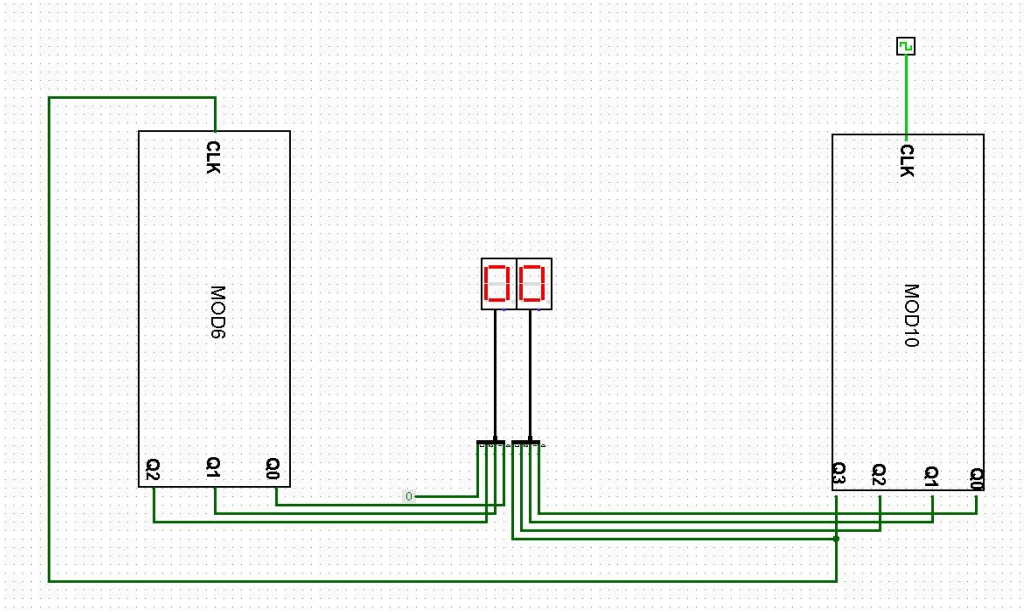

Fig 2-11 Initial State After Reset

The simulation results confirm the functionality of the MOD-60 counter, showcasing its capacity to count up to 59 and then recycle to 0 as expected.

Discussion and comment

Propagation Delay

In designing the asynchronous MOD-60 counter, propagation delay, setup time, and hold time are critical parameters that define the performance and reliability of the circuit. The propagation delay, $t_{PHL}\ and\ t_{PLH}$, is the time taken for a signal to transition from low to high and high to low, respectively. These delays are crucial in determining the maximum operating frequency of the counter.

During our experimental procedure, we opted for the 74HC112 JK Flip-Flops as the primary components in the MOD-60 counter design. This model was selected due to its balanced propagation delay of 17 ns for both $t_{PHL} and\ t_{PLH}$, which is conducive to achieving a responsive yet stable counting mechanism.

To derive the maximum operational clock frequency f_{max} ,we will assume the worst case the total propagation delay of the counter

$$t_{pd} = t_{PHL} = t_{PLH} = 17ns$$

$$\mathrm{T_{clk}} = \mathrm{N} * t_{pd}, where\ N\ is\ the\ number\ of bits$$

$$f_{max} = \frac{1}{\mathrm{T_{clk}}} = \frac{1}{N * t_{pd}}$$

Given that our MOD-60 counter utilizes a total of 7 JK flip-flops—representing a combination of a MOD-10 and a MOD-6 counter to encapsulate the full range of 60 states—we set N to 7.

$$f_{max} = \frac{1}{\mathrm{T_{clk}}} = \frac{1}{N * t_{pd}} = 8.40\ MHz$$

By incorporating this value into the calculation, we determine the upper limit for the clock frequency permissible for the counter's seamless operation without the risk of state-skipping or miscounting due to propagation delay.