COUNTERS

Introduction

A counter is a sequential circuit used in digital electronics to count pulses. Counters are fundamental components in many digital systems such as timers, clocks, frequency dividers, and event counters. They can be implemented using flip-flops (like JK, D, or T flip-flops) and count in either binary or non-binary sequences.

Types of Counters

Counters are broadly categorized into the following types:

1.Asynchronous (Ripple) Counter

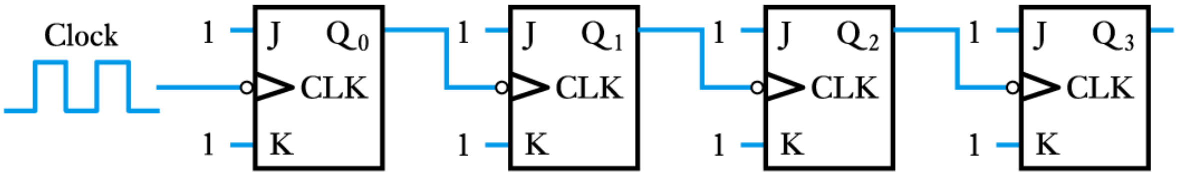
- Only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop means output of previous flip-flop is connected to clock input of next flip flop.

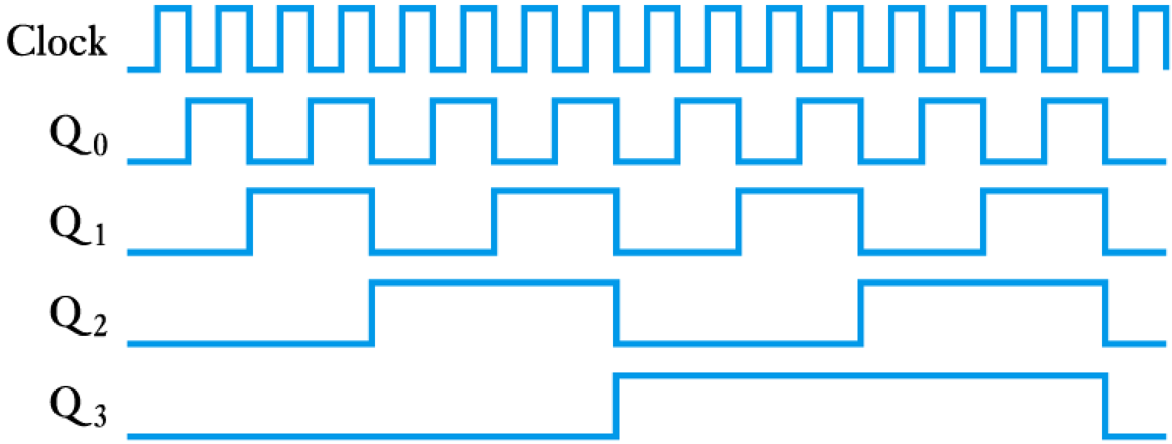
- Asynchronous counters are slower than synchronous counters because of the delay in the transmission of the pulses from flip-flop to flip-flop.

-Asynchronous counters are also **called *ripple-counters*** because of the way the clock pulse ripples it way through the flip-flops.

Advantage: Simple design.

Disadvantage: Propagation delay accumulates across stages.





2. Synchronous Counter

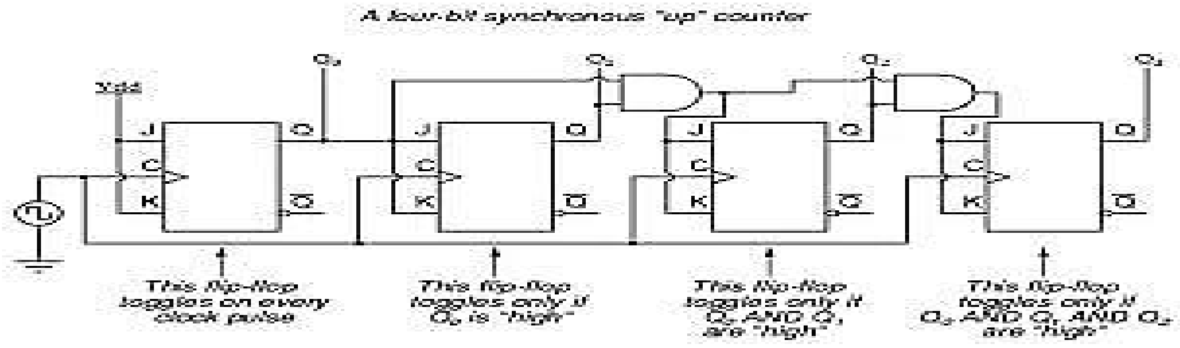
All flip-flops are clocked simultaneously by an external clock. Means clock input of all flip flops are connected to same external clock.

-Synchronous counters are faster than asynchronous counters because of the simultaneous clocking.

- Synchronous counters are an example of *state machine* design because they have a set of states and a set of transition rules for moving between those states after each clocked event.

Advantage: Faster operation due to simultaneous triggering.

Disadvantage: More complex wiring.



3. Binary Counters

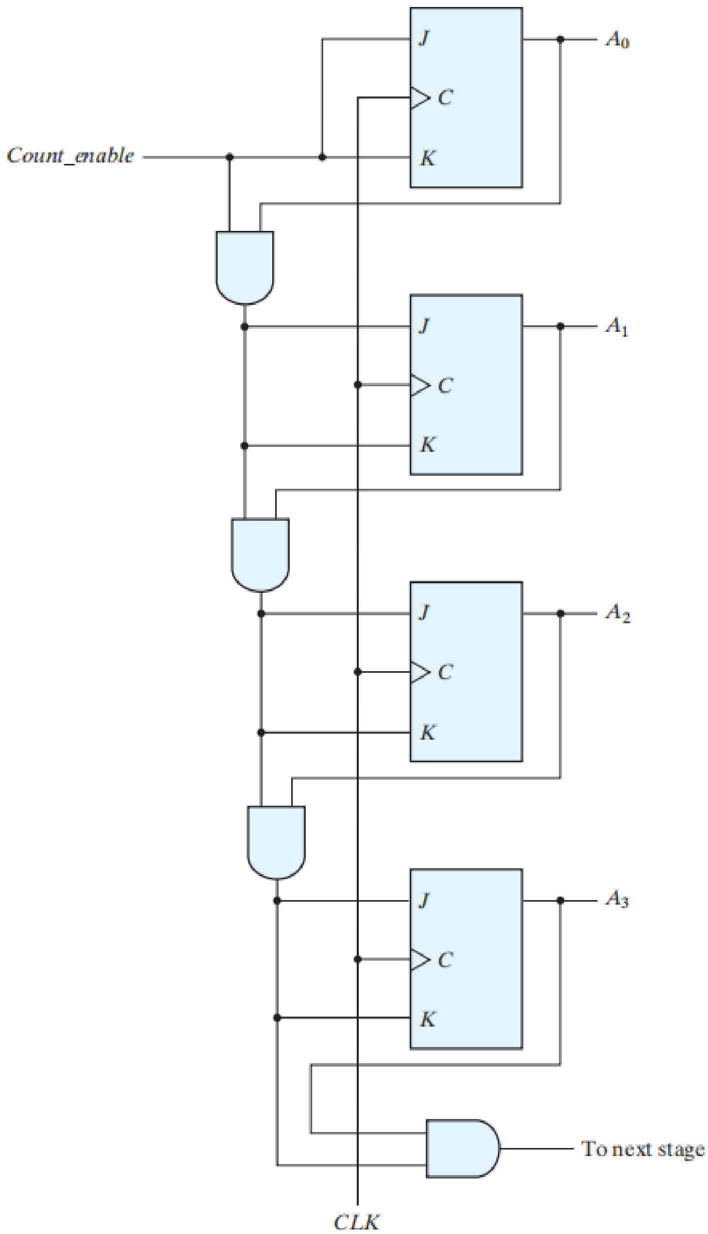
-A binary counter counts in a binary sequence. The number of states (or modulus) is given by (2^n), where (n) is the number of flip-flops.

-In a synchronous binary counter, the flip-flop in the least significant position is complemented with every pulse.

-A flip flop in any other position is complemented when all the bits in the lower significant positions are equal to 1 .

-For example, if the present state of a four-bit counter is $A3A2A1A0$ = 0011, the next count is 0100. $A0$ is always complemented. $A1$ is complemented because the present state of $A0$ = 1. $A2$ is complemented because the present state of $A1A0$ = 11. However,

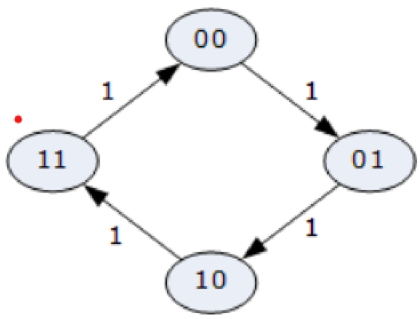
$A3$ is not complemented, because the present state of $A2A1A0$ = 011, which does not give an all-1's condition



3.1 Mod-n Counter

- A mod-n counter has `n` distinct states, resetting back to 0 after reaching n-1.

Clock Pulse	Present State			Next State		State Diagram
	Q_B	Q_A		Q_B	Q_A	
0 (start)	0	0	⇒	0	1	
1	0	1	⇒	1	0	
2	1	0	⇒	1	1	
3	1	1	⇒	0	0	
4 (repeat)	0	0	⇒	0	1	



- Example: A mod-4 counter counts from 0 to 3.

4. Up/Down Counter

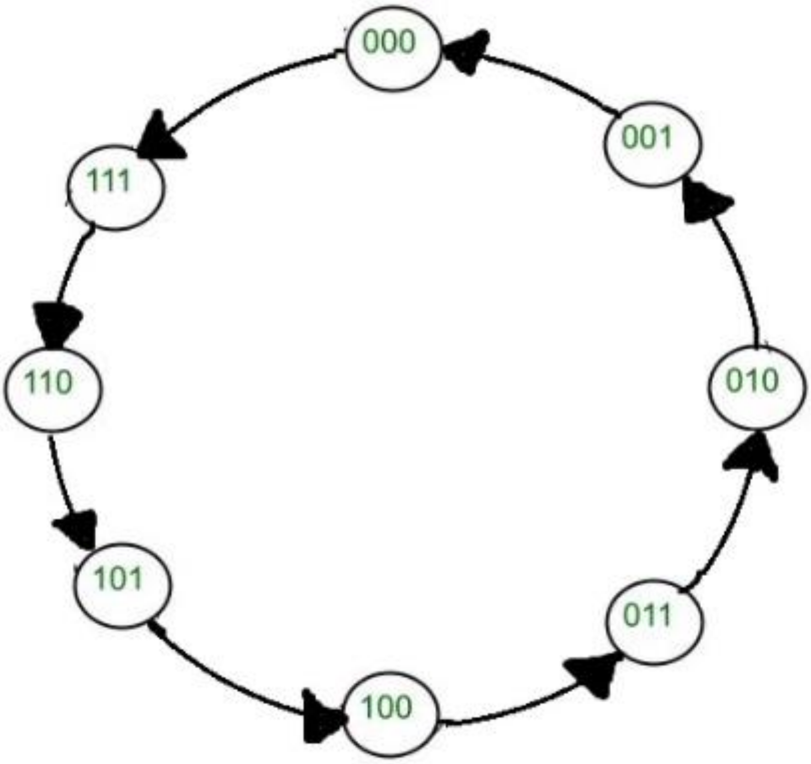
4.1 Up Counter

- Counts in ascending order (0, 1, 2, …).

4.2 Down Counter

- Counts in descending order (n, n-1, …).

Number of states = 2^{n}, where n is number of bits.

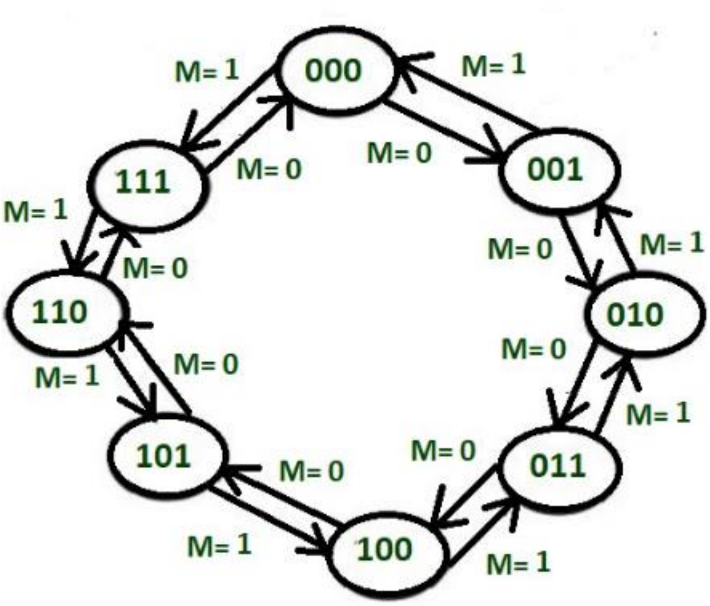


Previous state			Next state					
Q_3	Q_2	Q_1	Q^*_3	Q^*_2	Q^*_1	T_3	T_2	T_1
0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	1
1	0	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0	1
1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	0	0	1

Here T = 1, then there is output state(next state changes from previous state) changes i.e Q changes from 0 to 1 or 1 to 0
T= 0 then, there is no state output state changes i.e Q remains same

4.3 Up/Down Counter

- Can count both up and down based on control inputs.



State transition diagram for 3 bit up/down counting.

- When M=0 ,then the counter will perform up counting.
- When M=1 ,then the counter will perform down counting.

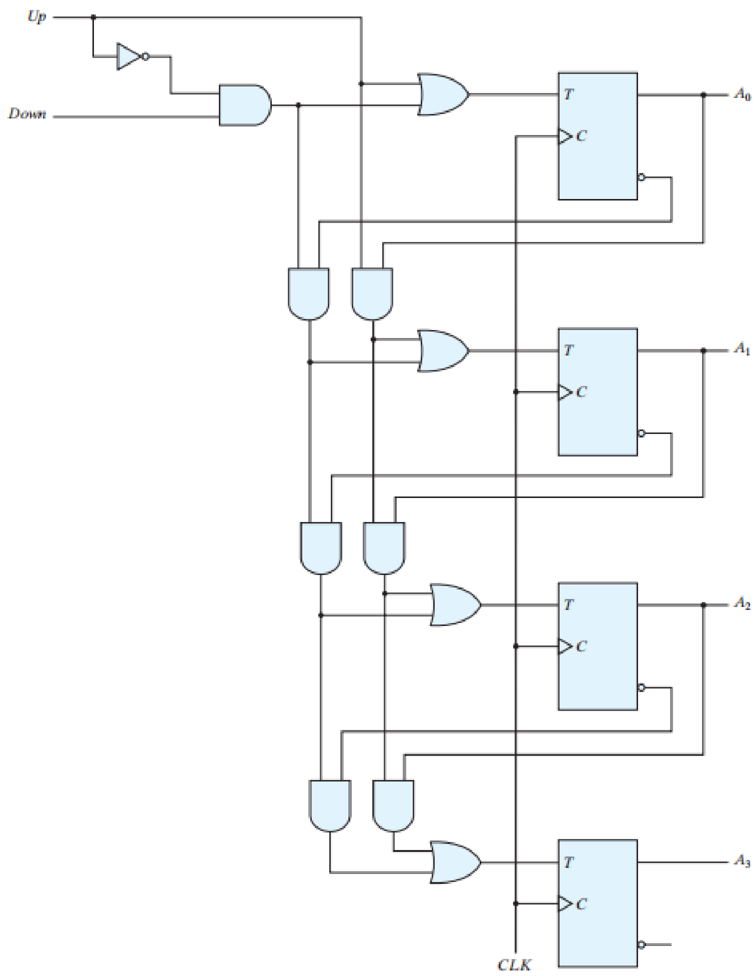


FIGURE 6.13
Four-bit up–down binary counter

5.BCD COUNTER

A **BCD (Binary Coded Decimal) counter** is a **mod-10 counter** that counts from 0 to 9 in binary and then resets to 0 on the next pulse. It is often used in digital clocks, calculators, and other devices where decimal numbers are needed.

Each decimal digit (0-9) is represented by a **4-bit binary code**, following the BCD format. The counter cycles through **10 states** (0000 to 1001) and resets to 0000 after reaching 9 (1001).

State Table for BCD Counter

Present State				Next State				Output	Flip-Flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

6.RING COUNTER

-A ring counter is a type of sequential counter, where the output of the last flip-flop is fed back to the input of the first flip-flop, forming a closed loop or "ring". It is commonly used in shift register circuits.

-A *ring counter* is a circular shift register with only one flip-flop being set at any particular time; all others are cleared.

-The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals

1000 → 0100 → 0010 → 0001 → 1000 → (repeats)

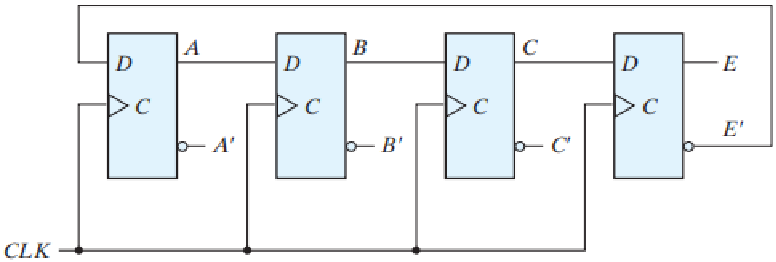
Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
4	0	0	0	1

- **Number of States**: A 4-bit ring counter has exactly **4 states**.

- **Modulus (Mod)**: A ring counter with n flip-flops has a modulus of **n** (for example, a 4-bit ring counter is mod-4).
- **Initial State**: Requires setting the first flip-flop to 1 (initialization).
- **Simple Design**: No complex logic gates are required between the flip-flops.

7.JOHNSON COUNTER

A Johnson counter, also known as a Twisted Ring Counter, is a type of shift register counter where the complemented output of the last flip-flop is fed back to the input of the first flip-flop. It differs from a standard ring counter by producing 2n unique states using n flip-flops, making it more efficient.



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Count sequence and required decoding

Applications of Counters

1. Timers – Used to generate delays.

2. Frequency Division– Dividing the frequency of a clock signal.

3. Digital Clocks– To keep track of hours, minutes, and seconds.

4. Event Counters – To count occurrences of specific events (e.g., visitors in a building).

5. Memory Addressing – In microprocessors to increment memory locations.

7. Advantages and Limitations

Advantages:

- Simple to design using flip-flops.
- Useful for many real-time applications like clocks and timers.

Limitations:

- Asynchronous counters suffer from propagation delays.
- More flip-flops increase complexity and power consumption

8. Conclusion

Counters are essential components in digital systems for tracking, controlling, and dividing time-based events. Depending on the requirements, either asynchronous or synchronous counters can be used, each with its trade-offs in speed and complexity.