Sairam INSTITUTIONS

Sri SAI RAM ENGINEERING COLLEGE

INSTITUTE OF TECHNOLOGY

West Tambaram, Chennai - 44

YEAR	SEM
II	III

CS 8351

DIGITAL PRINCIPLES AND SYSTEM DESIGN (Common to CSE & IT)

UNIT NO. 3

3.7 COUNTERS

Version: 1.0

4 QUALITY EDUCATION

NBA NATIONAL BOARD of ACCREDITATION

NAAC A+

nirf

SYNCHRONOUS COUNTERS

Flip-Flops can be connected together to perform counting operations. Such a group of Flip- Flops is a **counter**. The number of Flip-Flops used and the way in which they are connected determine the number of states (called the modulus) and also the specific sequence of states that the counter goes through during each complete cycle.

Counters are classified into two broad categories according to the way

they are clocked:

Asynchronouscounters,

Synchronous counters.

In asynchronous (ripple) counters, the first Flip-Flop is clocked by the

external clock pulse and then each successive Flip-Flop is clocked by the

output of the preceding Flip-Flop.

In synchronous counters, the clock input is connected to all of the

Flip-Flops so that they are clocked simultaneously. Within each of these

two categories, counters are classified primarily by the type of sequence,

the number of states, or the number of Flip-Flops in the counter.

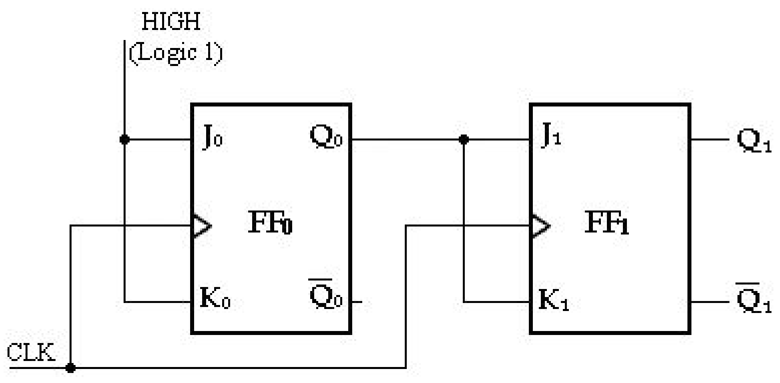
The term 'synchronous' refers to events that have a fixed time relationship with each other. In synchronous counter, the clock pulses are applied to all Flip- Flops simultaneously. Hence there is minimum propagation delay.

S.No	Asynchronous (ripple) counter	Synchronous counter
1	All the Flip-Flops are not clocked simultaneously.	All the Flip-Flops are clocked simultaneously.
2	The delay times of all Flip- Flops are added. Therefore there is considerable propagation delay.	There is minimum propagation delay.
3	Speed of operation is low	Speed of operation is high.
4	Logic circuit is very simple	Design involves complex logic circuit

	even for more number of states.	as number of state increases.
5	Minimum numbers of logic devices are needed.	The number of logic devices is more than ripple counters.
6	Cheaper than synchronous counters.	Costlier than ripple counters.

2-Bit Synchronous Binary Counter

In this counter the clock signal is connected in parallel to clock inputs of both the Flip-Flops (FF0 and FF1). The output of FF0 is connected to J1 and K1 inputs of the second Flip-Flop (FF1).

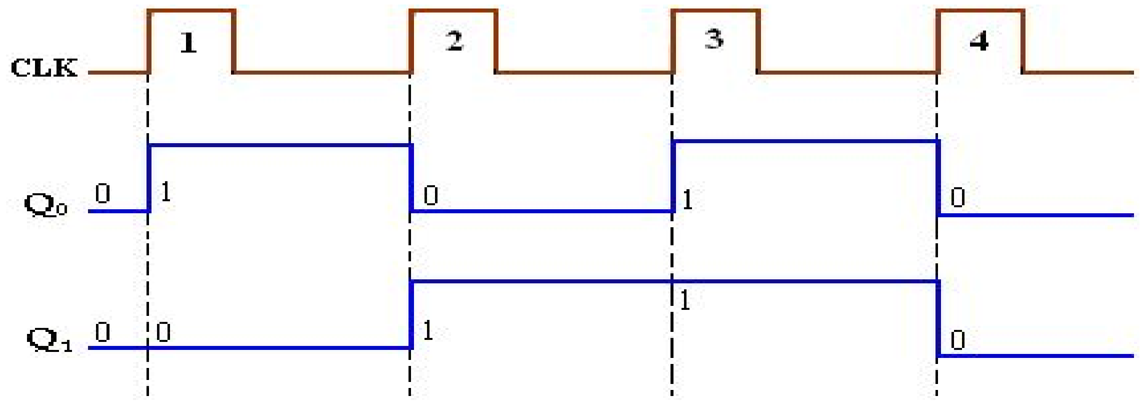


Assume that the counter is initially in the binary 0 state: i.e., both Flip-Flops are RESET. When the positive edge of the first clock pulse is applied, FF_0 will toggle because J_0= k_0= 1, whereas FF_1 output will remain 0 because J_1= k_1= 0. After the first clock pulse Q_0=1 and Q_1=0.

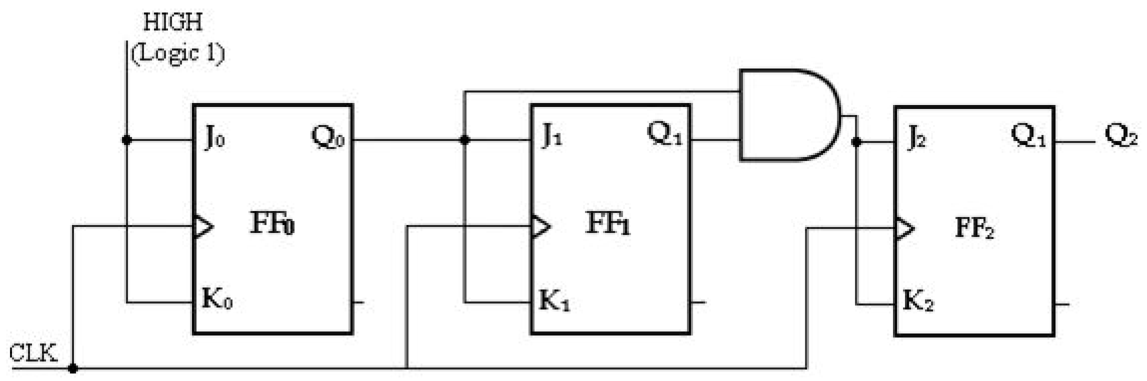
When the leading edge of CLK2 occurs, FF_0 will toggle and Q_0 will go LOW. Since FF_1 has a HIGH (Q_0 = 1) on its J_1 and K_1 inputs at the triggering edge of this clock pulse, the Flip-Flop toggles and Q_1 goes HIGH. Thus, after CLK2,Q0 = 0 and Q1 = 1.

When the leading edge of CLK3 occurs, FF0 again toggles to the SET state (Q0= 1), and FF1 remains SET (Q1 = 1) because its J1 and K1 inputs are both LOW (Q0 = 0). After this triggering edge, Q0 = 1 and Q1 = 1.

Finally, at the leading edge of CLK4, Q0 and Q1 go LOW because they both have a toggle condition on their J1 and K1 inputs. The counter has now recycled to its original state, Q0 = Q1 = 0.

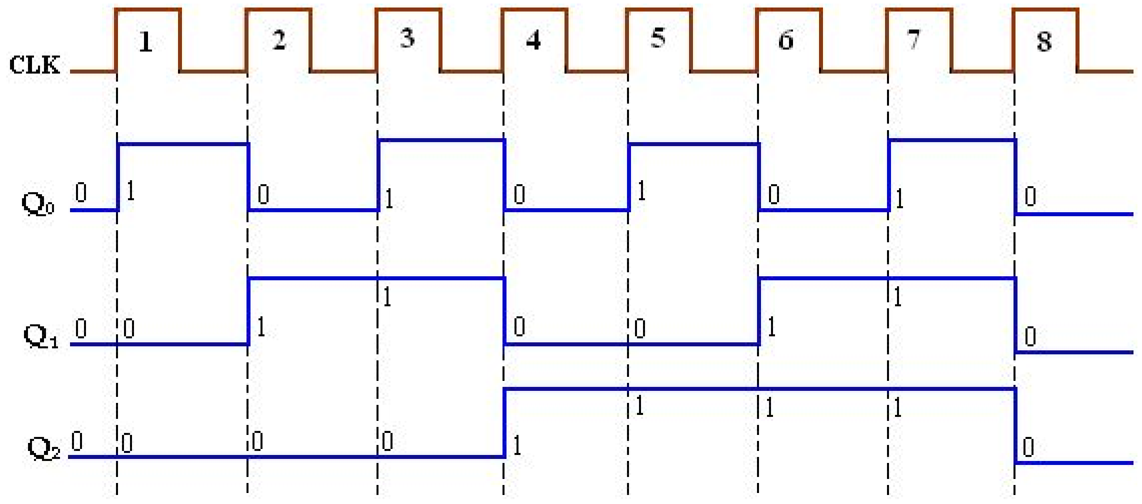


A 3 bit synchronous binary counter is constructed with three JK Flip-Flops and an AND gate. The output of FF0 (Q0) changes on each clock pulse as the counter progresses from its original state to its final state and then back to its original state. To produce this operation, FF0 must be held in the toggle mode by constant HIGH, on its J0 and K0 inputs.



The output of FF1 (Q1) goes to the opposite state following each time Q0= 1. This change occurs at CLK2, CLK4, CLK6, and CLK8. The CLK8 pulse causes the counter to recycle. To produce this operation, Q0 is connected to the J1 and K1 inputs of FF1. When Q0= 1 and a clock pulse occurs, FF1 is in the toggle mode and therefore changes state. When Q0= 0, FF1 is in the no-change mode and remains in its present state.

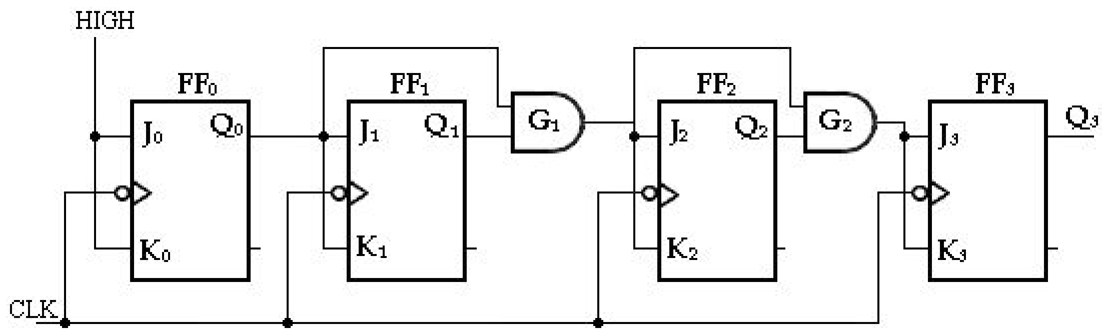
The output of FF2 (Q2) changes state both times; it is preceded by the unique condition in which both Q0 and Q1 are HIGH. This condition is detected by the AND gate and applied to the J2 and K2 inputs of FF3. Whenever both outputs Q0= Q1= 1,the output of the AND gate makes the J2= K2= 1 and FF2 toggles on the following clock pulse. Otherwise, the J2 and K2 inputs of FF2 are held LOW by the AND gate output, FF2 does not change state.



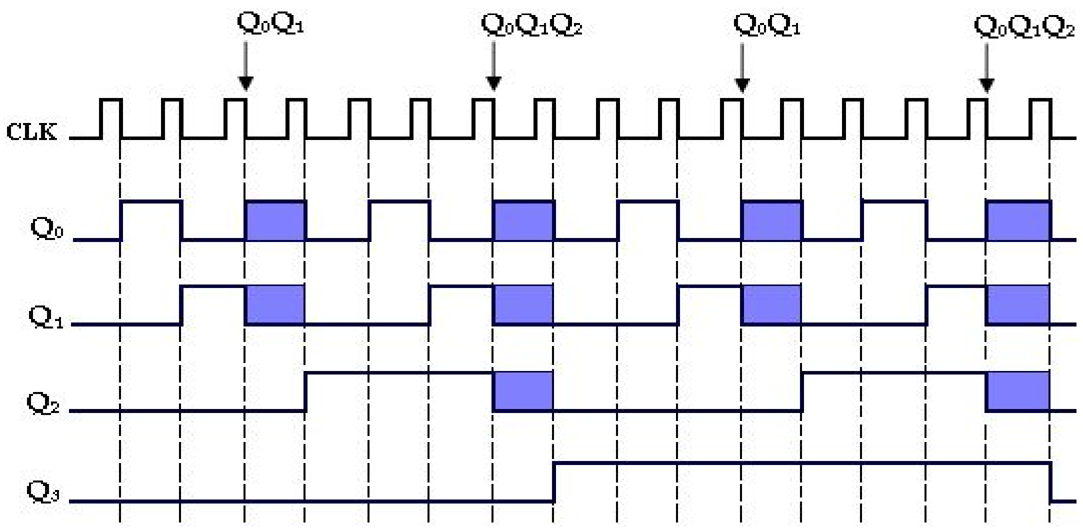
4-Bit

Synchronous Binary Counter

This particular counter is implemented with negative edge-triggered Flip- Flops. The reasoning behind the J and K input control for the first three Flip- Flops is the same as previously discussed for the 3-bit counter. For the fourth stage, the Flip- Flop has to change the state when Q0= Q1= Q2= 1. This condition is decoded by AND gate G3.

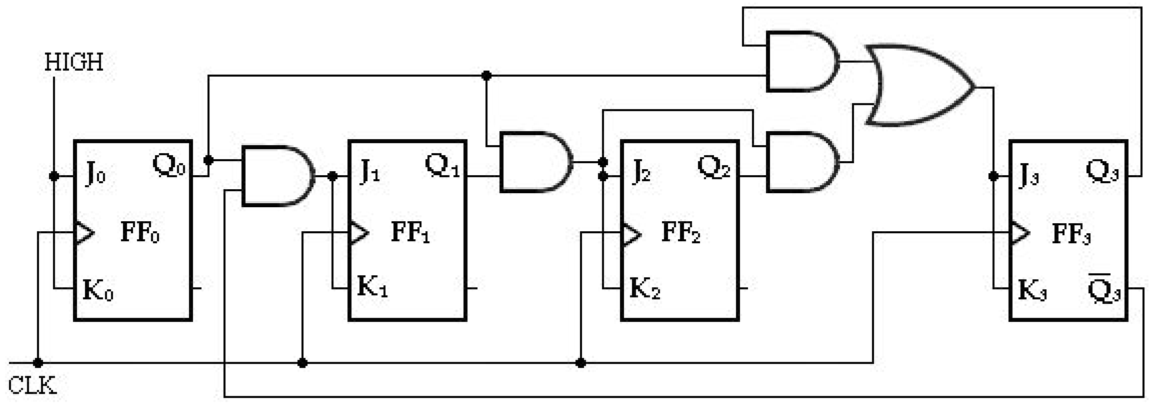


Therefore, when Q0= Q1= Q2= 1, Flip-Flop FF3 toggles and for all other times it is in a no-change condition. Points where the AND gate outputs are HIGH are indicated by the shaded areas.



4-Bit Synchronous Decade Counter: (BCD Counter)

BCD decade counter has a sequence from 0000 to 1001 (9). After 1001 state it must recycle back to 0000 state. This counter requires four Flip-Flops and AND/OR logic as shown below.



CLOCK PULSE	Q3	Q2	Q2	Q1
Initially	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

10	0	0	0	0

.

First, notice that FF$_0$ (Q$_0$) toggles on each clock pulse, so the logic equation for its J$_0$ and K$_0$ inputs is **$J_0 = K_0 = 1$**

This equation is implemented by connecting J0 and K0 to a constant HIGH level.

Next, notice from table, that FF$_1$ (Q$_1$) changes on the next clock pulse each time Q0 = 1 and Q3 = 0, so the logic equation for the J$_1$ and K$_1$ inputs is

$$J_1 = K_1 = Q_0Q_3'$$

This equation is implemented by ANDing Q0 and Q3 and connecting the gate output to the J1 and K1 inputs of FFl.

· Flip-Flop 2 (Q2) changes on the next clock pulse each time both Q0 = Q1 = 1. This requires an input logic equation as follows:

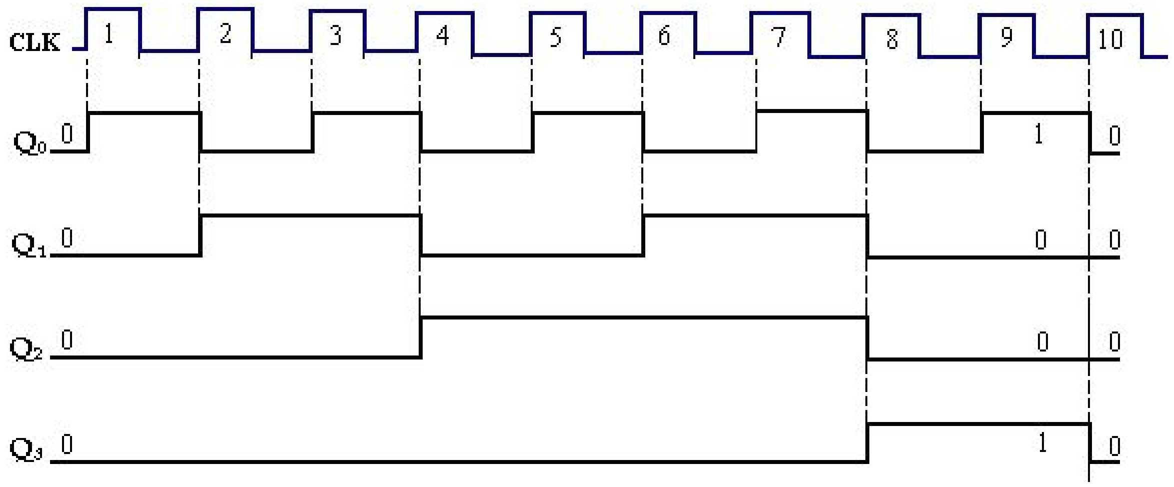
$$J_2 = K_2 = Q_0Q_1$$

This equation is implemented by ANDing Q$_0$ and Q$_1$ and connecting the gate output to the J$_2$ and K$_2$ inputs of FF$_3$.

Finally, FF3 (Q3) changes to the opposite state on the next clock pulse each time Q0 = 1, Q1 = 1, and Q2 = 1 (state 7), or when Q0 = 1 and Q1 = 1 (state 9). The equation for this is as follows:

$$J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$$

This function is implemented with the AND/OR logic connected to the J3 and K3 inputs of FF3.



Synchronous UP/DOWN Counter

An up/down counter is a bidirectional counter, capable of progressing in either direction through a certain sequence. A 3-bit binary counter that advances upward through its sequence (0, 1, 2, 3, 4, 5, 6, 7) and then can be reversed so that it goes through the sequence in the opposite direction (7, 6, 5, 4, 3, 2, 1,0) is an illustration of up/down sequential operation.

The complete up/down sequence for a 3-bit binary counter is shown in table below. The arrows indicate the state-to-state movement of the counter for both its UP and its DOWN modes of operation. An examination of Q0 for both the up and down sequences shows that FF0 toggles on each clock pulse. Thus, the J0 and K0 inputs of FF0 are,

$$J0= K0= 1$$

CLOCK PULSE	UP	Q_2	Q_1	Q_0	DOWN
0		0	0	0	
1		0	0	1	
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	

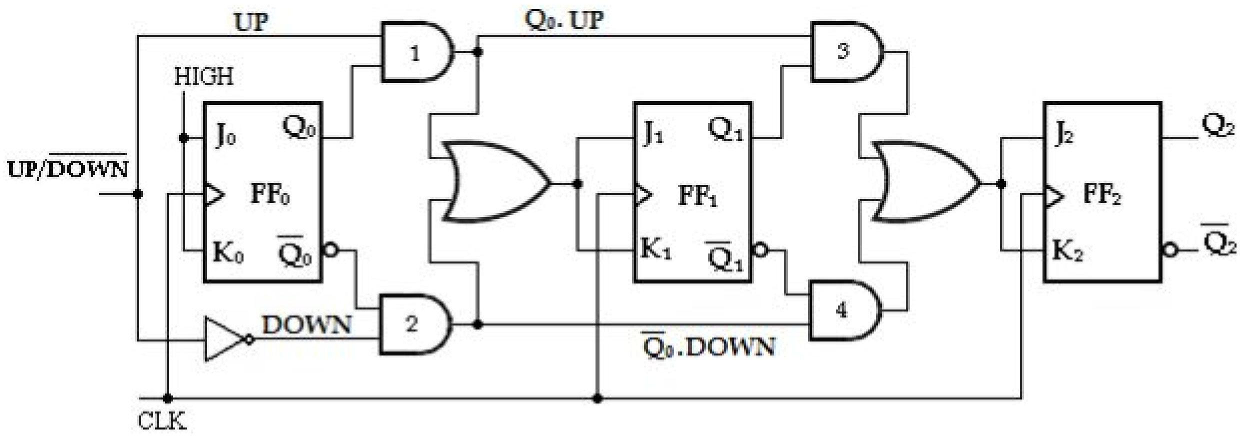
To form a synchronous UP/DOWN counter, the control input (UP/DOWN) is used to allow either the normal output or the inverted output of one Flip-Flop to the J and K inputs of the next Flip-Flop. When UP/DOWN= 1, the MOD 8 counter will count from 000 to 111 and UP/DOWN= 0, it will count from 111 to 000.

When UP/DOWN= 1, it will enable AND gates 1 and 3 and disable AND gates 2 and 4. This allows the Q0 and Q1 outputs through the AND gates to the J and K inputs of the following Flip-Flops, so the counter counts up as pulses are applied.

When UP/DOWN= 0, the reverse action takes place.

$J_1 = K_1 = (Q_0.UP) + (Q_0'.DOWN)$

$J_2 = K_2 = (Q_0.\ Q_1.UP) + (Q_0'.Q_1'.DOWN)$



MODULUS-N-COUNTERS

The counter with 'n' Flip-Flops has maximum MOD number 2n. Find the number of Flip-Flops (n) required for the desired MOD number (N) using the equation,

$$2n \geq N$$

(i) For example, a 3 bit binary counter is a MOD 8 counter. The basic counter can be modified to produce MOD numbers less than 2n by allowing the counter to skin those are normally part of counting sequence.

$n = 3$

$N = 8$

$2^n = 2^3 = 8 = N$

(i) MOD 5 Counter:

$2^n = N$

$2^n = 5$

$2^2 = 4$ less than N.

$2^3 = 8 > N(5)$

Therefore, 3 Flip-Flops are required.

(i) MOD 10 Counter:

$2^n = N = 10$

$2^3 = 8$ less than N.

$2^4 = 16 > N(10)$.

To construct any MOD-N counter, the following methods can be used.

1. Find the number of Flip-Flops (n) required for the desired MOD number (N) using the equation,

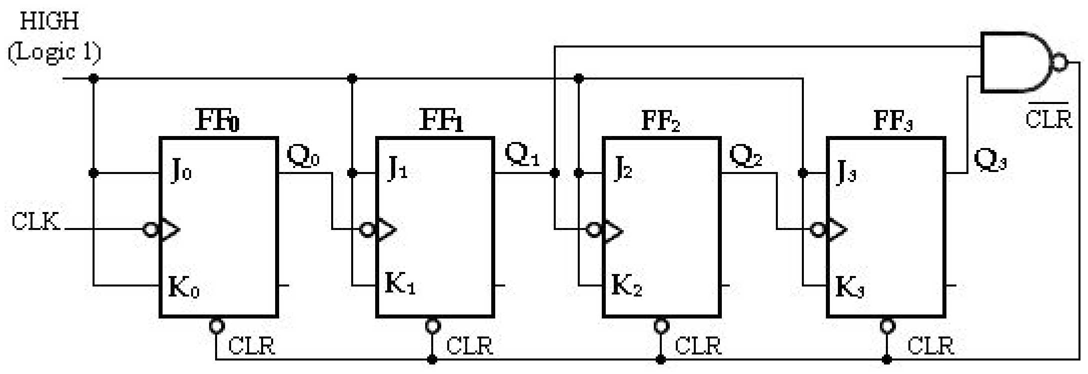
$2^n \geq N$.

2. Connect all the Flip-Flops as a required counter.
3. Find the binary number for N.
4. Connect all Flip-Flop outputs for which Q= 1 when the count is N, as inputs to NAND gate.
5. Connect the NAND gate output to the CLR input of each Flip-Flop.

When the counter reaches Nth state, the output of the NAND gate goes LOW, resetting all Flip-Flops to 0. Therefore the counter counts from 0 through N-1.

For example, MOD-10 counter reaches state 10 (1010). i.e., Q3Q2Q1Q0= 1 0 1 0. The outputs Q3 and Q1 are connected to the NAND gate and the output of the NAND gate goes LOW and resetting all Flip-Flops to zero. Therefore MOD-10 counter counts from 0000 to 1001. And then recycles to the zero value.

The MOD-10 counter circuit is shown below.


HIGH
(Logic 1)
CLK
FF0
FF1
FF2
FF3
J_0
K_0
J_1
K_1
J_2
K_2
J_3
K_3
Q_0
Q_1
Q_2
Q_3
$\overline{CLR}$
CLR
CLR
CLR
CLR

.