
COUNTERS

# COUNTERS

## TYPES OF COUNTER

- 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)
- Counters are classified into two broad categories according to the way they are clocked:

  Asynchronous counters

  Synchronous counters.

# COUNTERS

## TYPES OF COUNTER

- 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.

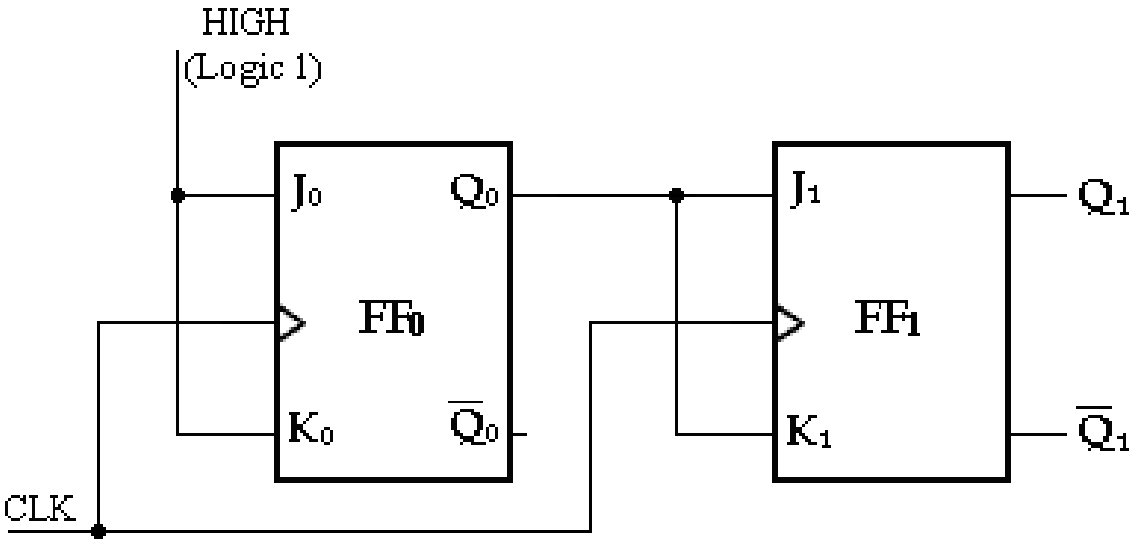
# COUNTERS

## SYNC. Vs ASYNC. COUNTER

| 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 even for more number of states. | Design involves complex logic circuit 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. |

# COUNTERS

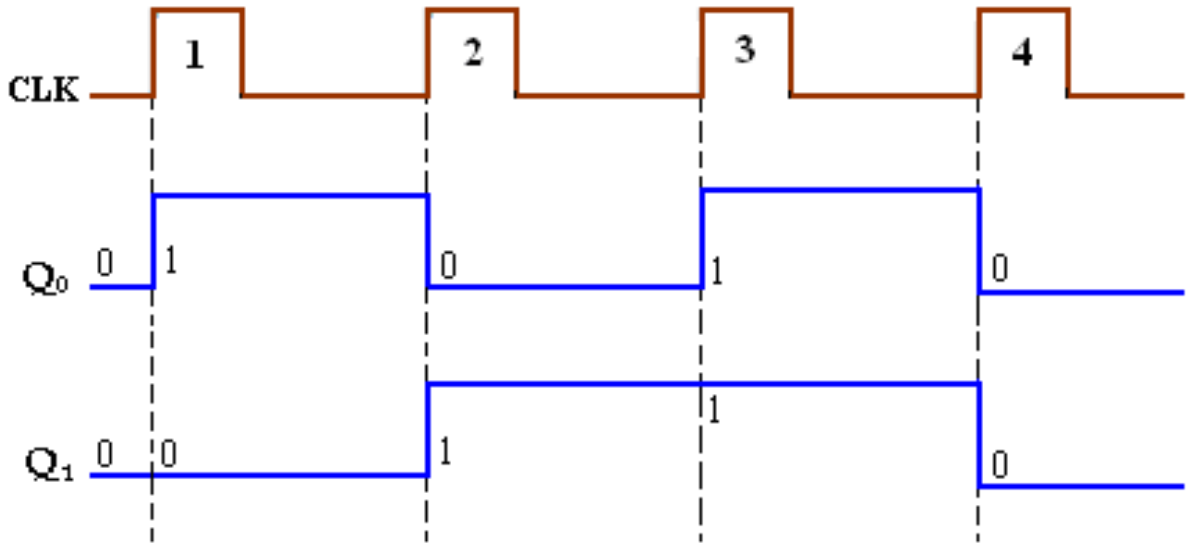
## 2-BIT SYNC. UP COUNTER




2-Bit Synchronous Binary Counter

# COUNTERS

## 2-BIT SYNC. UP COUNTER




Timing diagram

# COUNTERS

## Design steps of synchronous counter

- Find the number of flip flops using $2n \geq N$, where N is the number of states and n is the number of flip flops
- Choose the type of flip flop
- Draw the state diagram of the counter
- Draw the excitation table of the selected flip flop and determine the excitation table for the counter
- Use K-map to derive the flip flop input functions

# COUNTERS

## Design steps of synchronous counter

Design 3-bit synchronous up counter using JK flip flops.

***Step 1: Find the number of flip flops.***

A flip flop stores only one bit, hence for a 3 bit counter, 3 flip flops(n=3) are needed to design the counter.

Number of states = $2^n = 2^3 = 8$ states

***Step 2: Choose the type of flip flop.***

Since the type of flip flop is given in the problem, let us use JK flip flops.

***Step 3: Write the sequence of the counter***

000, 001, 010, 011, 100, 101, 110, 111

***Obtain excitation table for the counter.***

- We know, the excitation table for JK flip flop
- Excitation table for the 3-bit synchronous counter is determined from the excitation table of JK flip flop

# COUNTERS

## Design steps of synchronous counter

| Clock | Present State | | | Next State | | | Flip flop Inputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $Q_C$ | $Q_B$ | $Q_A$ | $Q_{C+1}$ | $Q_{B+1}$ | $Q_{A+1}$ | $J_C$ | $K_C$ | $J_B$ | $K_B$ | $J_A$ | $K_A$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | x | 0 | x | 1 | x |
| 2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | x | 1 | x | x | 1 |
| 3 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | x | x | 0 | 1 | x |
| 4 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | x | x | 1 | x | 1 |
| 5 | 1 | 0 | 0 | 1 | 0 | 1 | x | 0 | 0 | x | 1 | x |
| 6 | 1 | 0 | 1 | 1 | 1 | 0 | x | 0 | 1 | x | x | 1 |
| 7 | 1 | 1 | 0 | 1 | 1 | 1 | x | 0 | x | 0 | 1 | x |
| 8 | 1 | 1 | 1 | 0 | 0 | 0 | x | 1 | x | 1 | x | 1 |

Excitation table for JK flip flop

| $Q_n$ | $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | x |
| 0 | 1 | 1 | x |
| 1 | 0 | x | 1 |
| 1 | 1 | x | 0 |

# COUNTERS

## Design steps of synchronous counter

***Step 5: Derive the flip flop input functions***

**For $J_C$**

| $Q_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | X | X | X | X |

$J_C = Q_BQ_A$

**For $K_C$**

| $Q_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | X | X | X |
| 1 | 0 | 0 | 1 | 0 |

$K_C = Q_BQ_A$

**For $J_B$**

| $Q_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | X | X |
| 1 | 0 | 1 | X | X |

$J_B = Q_A$

**For $K_B$**

| $Q_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | X | 1 | 0 |
| 1 | X | X | 1 | 0 |

$K_B = Q_A$

**For $J_A$**

| $Q_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | X | X | 1 |
| 1 | 1 | X | X | 1 |

$J_A = 1$

**For $K_A$**

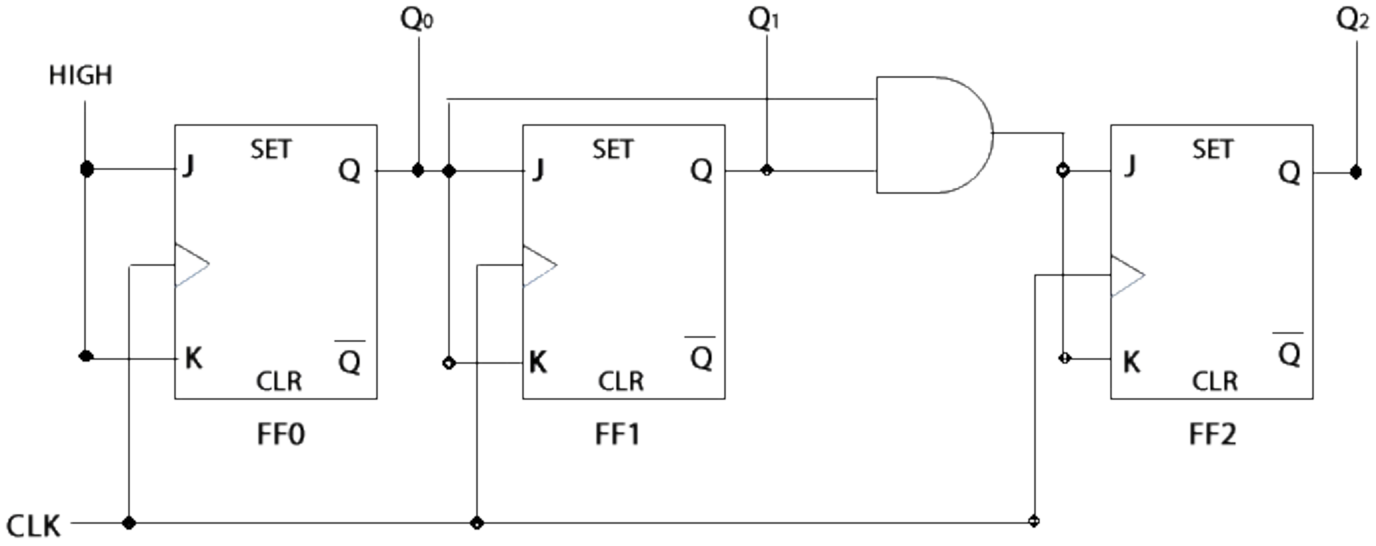
| $Q_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | 1 | 1 | X |
| 1 | X | 1 | 1 | X |

$K_A = 1$

# COUNTERS

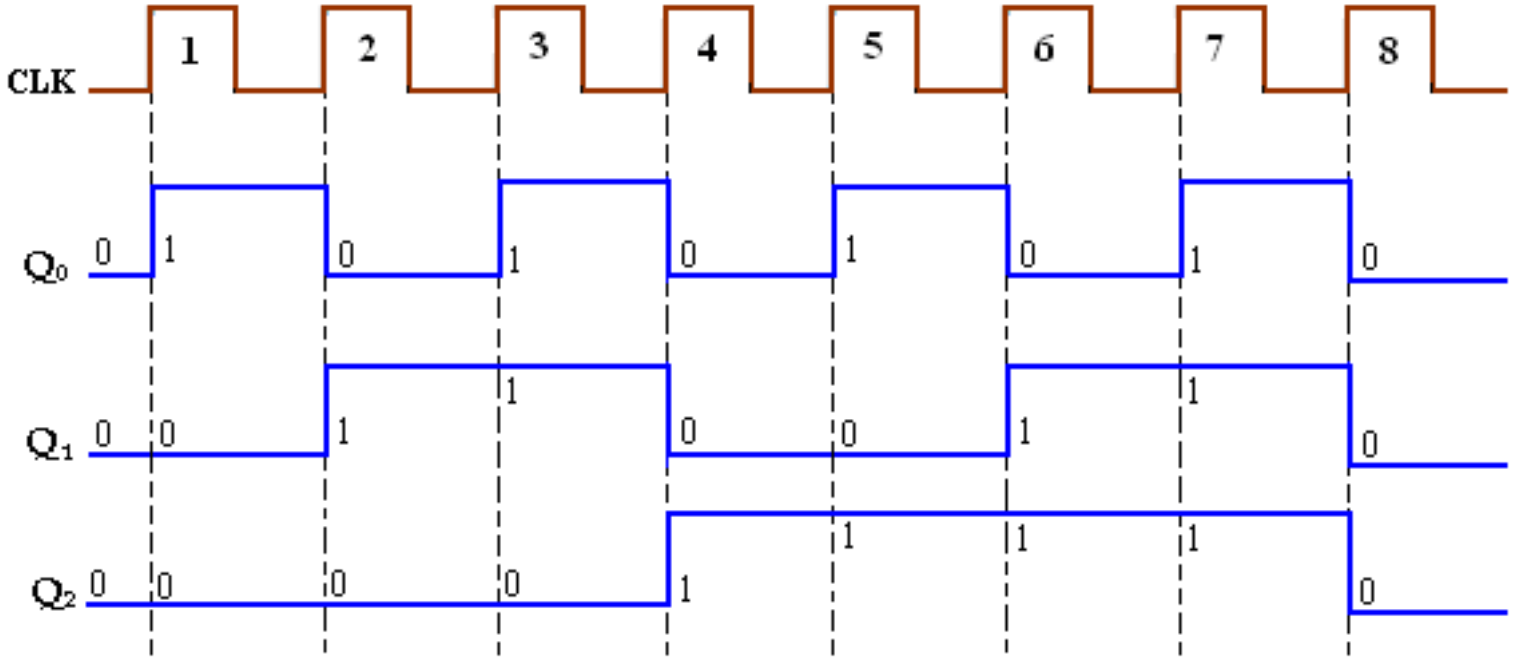
## Design steps of synchronous counter

***Step 6: Draw the logic diagram of the counter.***



# COUNTERS

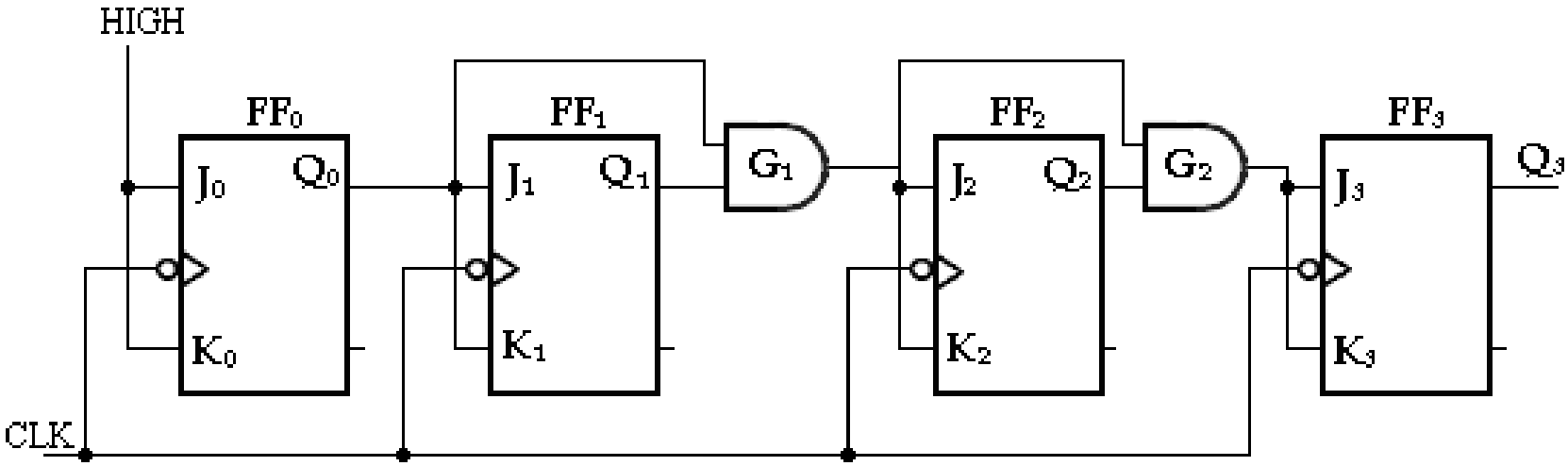
## 3-BIT SYNC. UP COUNTER




Timing diagram

# COUNTERS

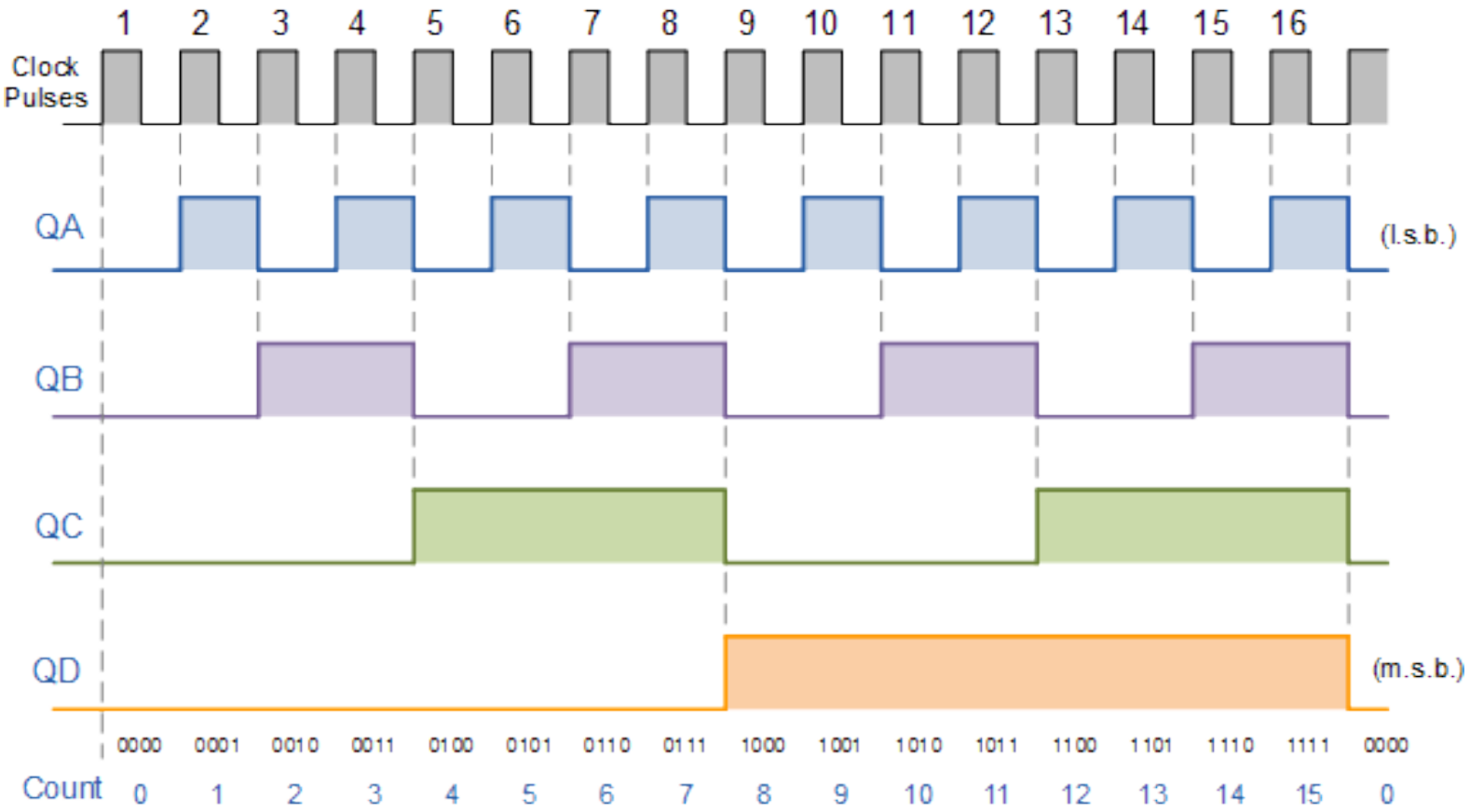
## 4-BIT SYNC. UP COUNTER




4-Bit Synchronous Binary Counter

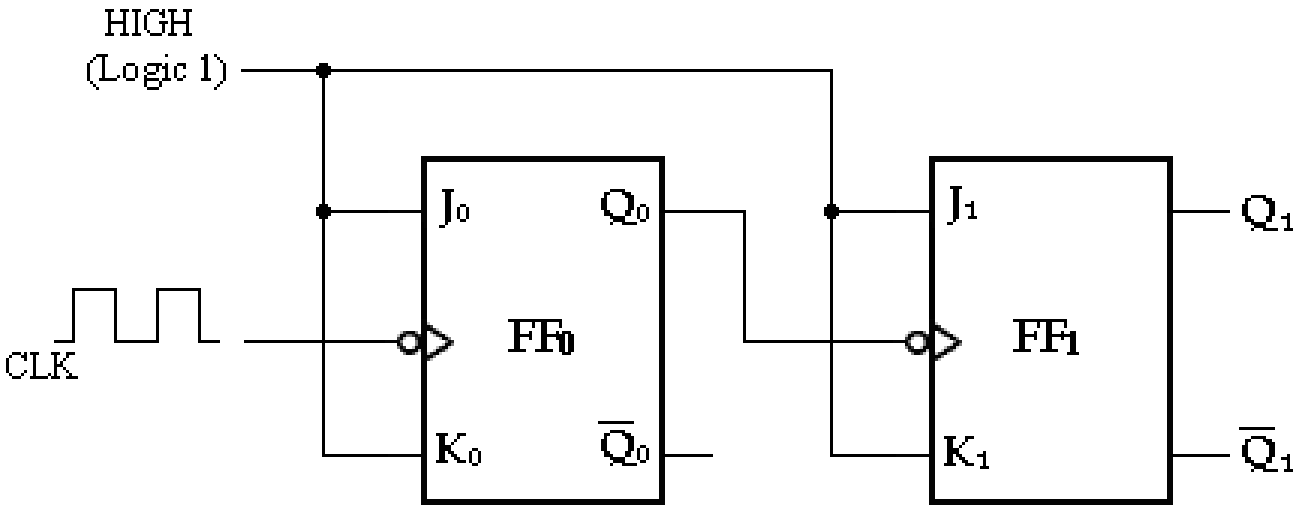
# COUNTERS

## 4-BIT SYNC. UP COUNTER



# COUNTERS

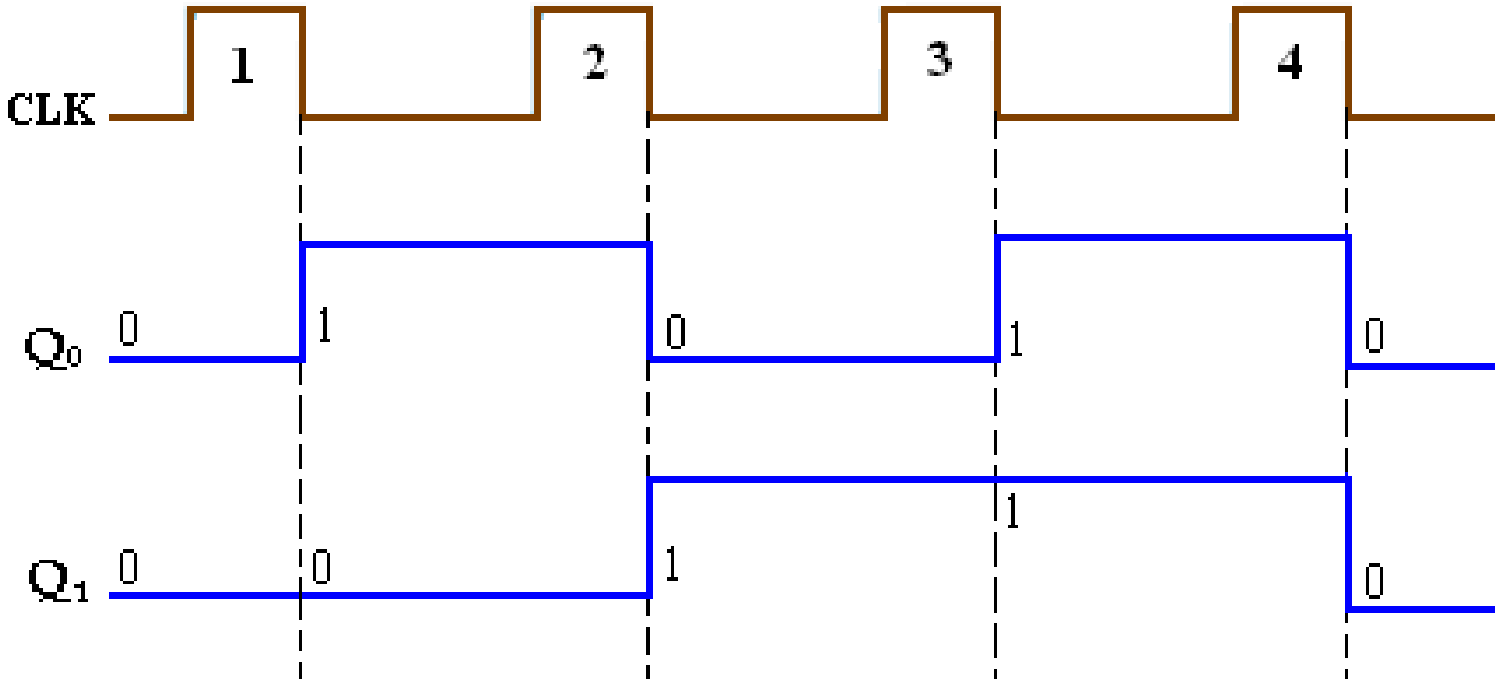
## 2-BIT ASYNC. UP COUNTER




2-Bit Asynchronous Binary Counter

# COUNTERS

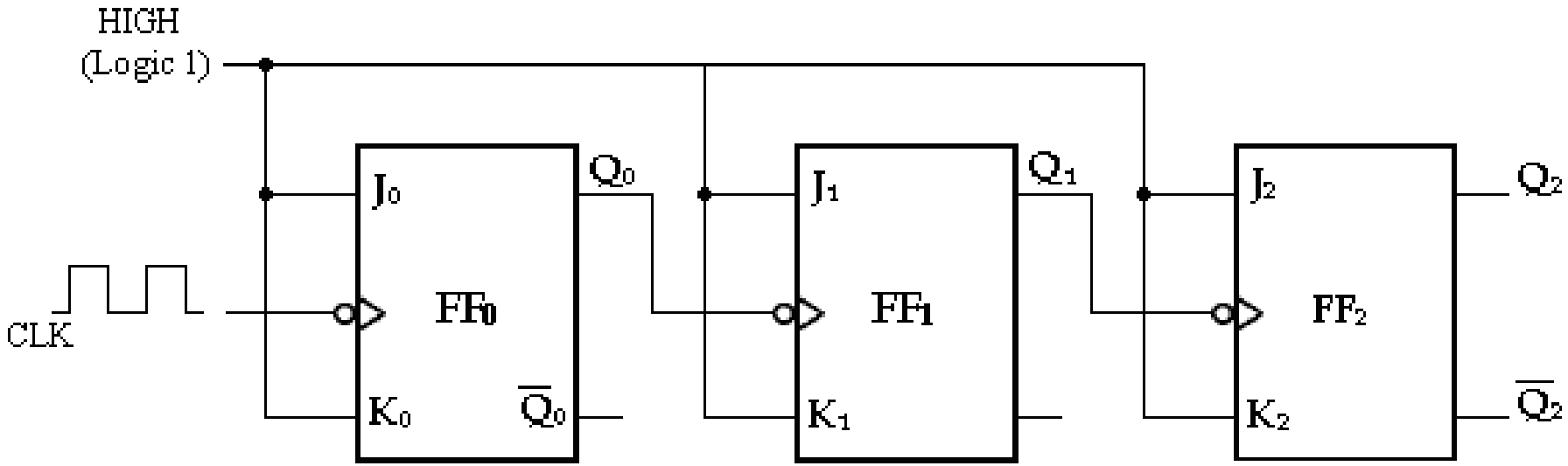
## 2-BIT ASYNC. UP COUNTER




Timing diagram for 2-bit counter

# COUNTERS

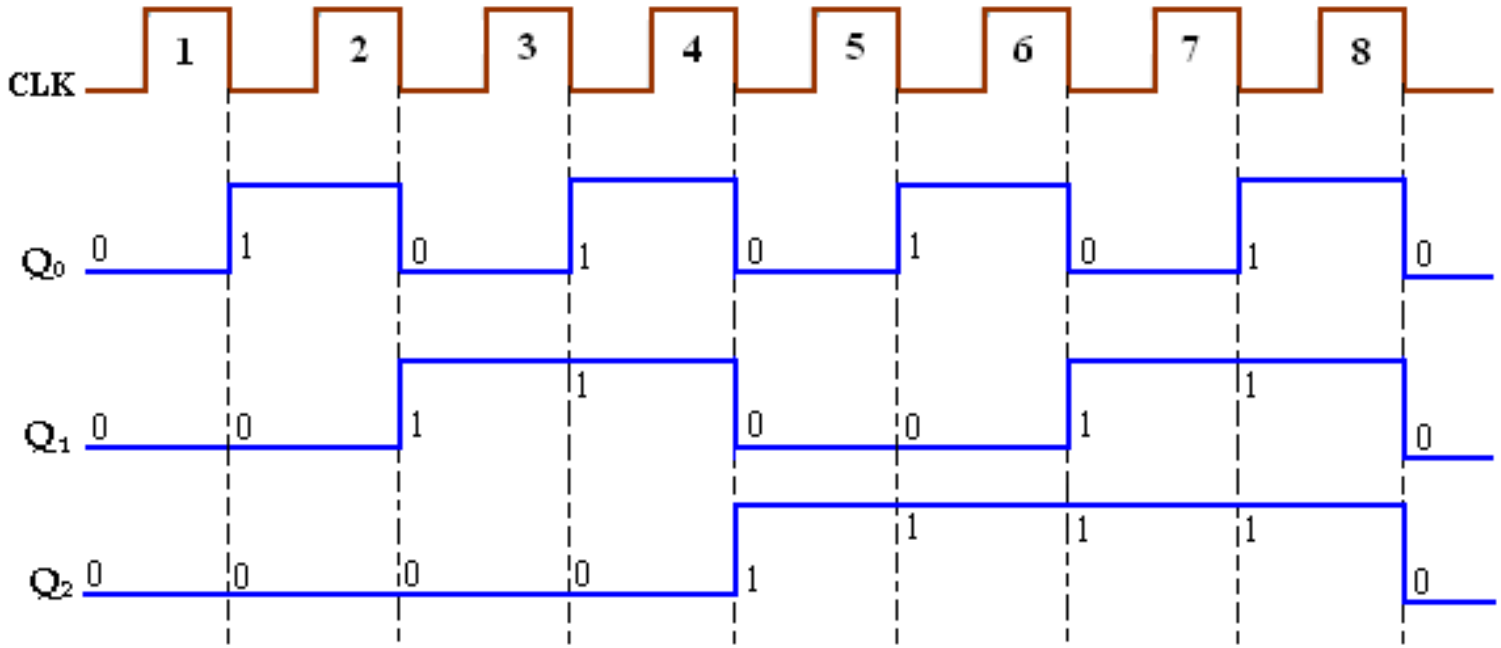
## 3-BIT ASYNC. UP COUNTER




3-Bit Asynchronous Binary Counter

# COUNTERS

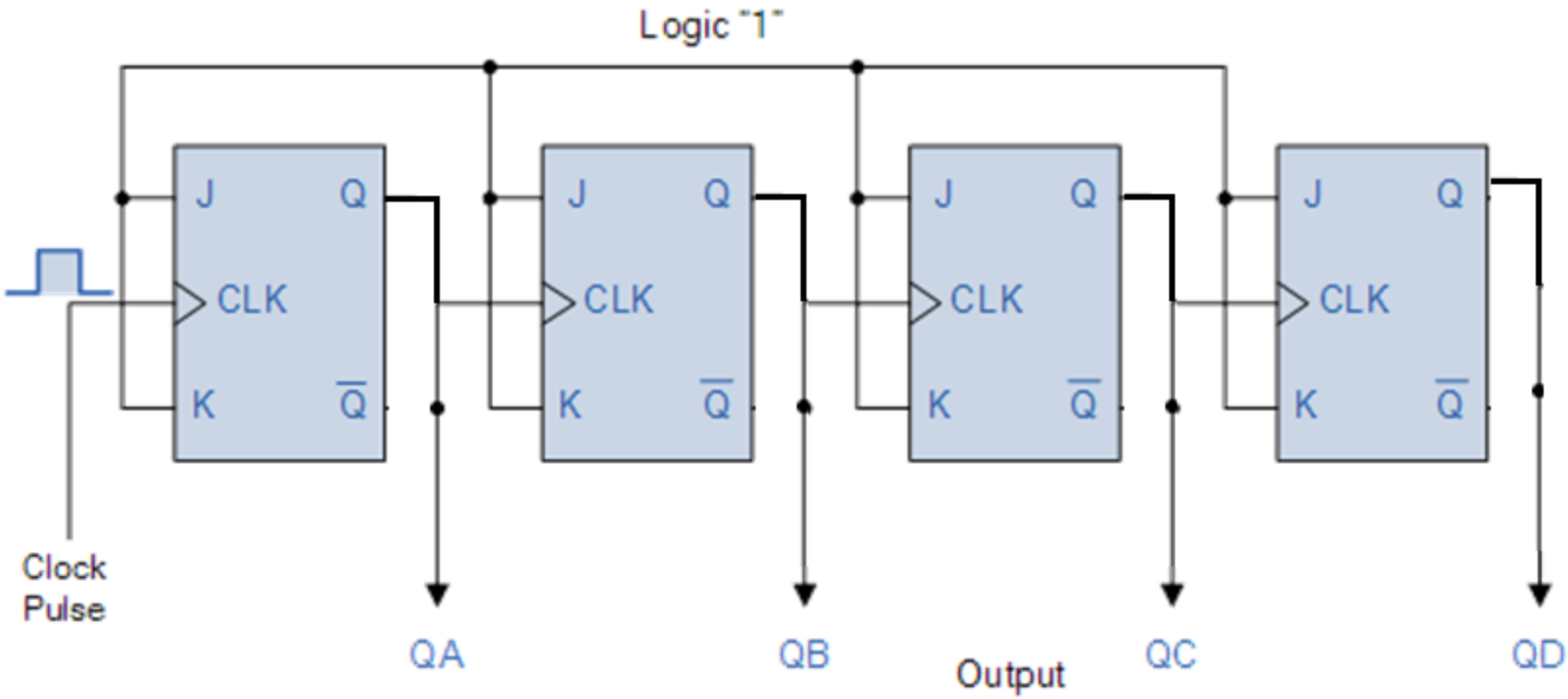
## 3-BIT ASYNC. UP COUNTER




Timing diagram

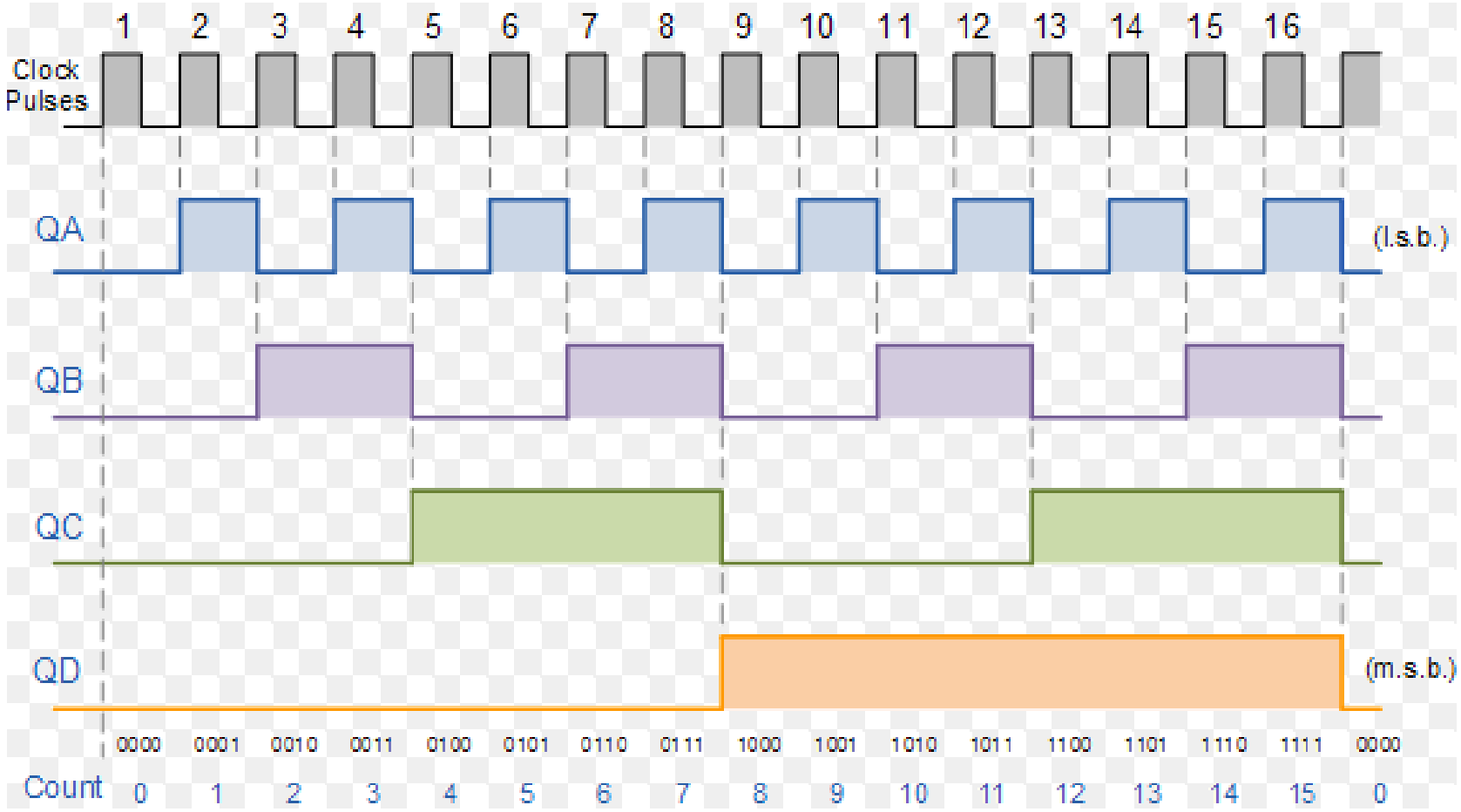
# COUNTERS

## 4-BIT ASYNC. UP COUNTER



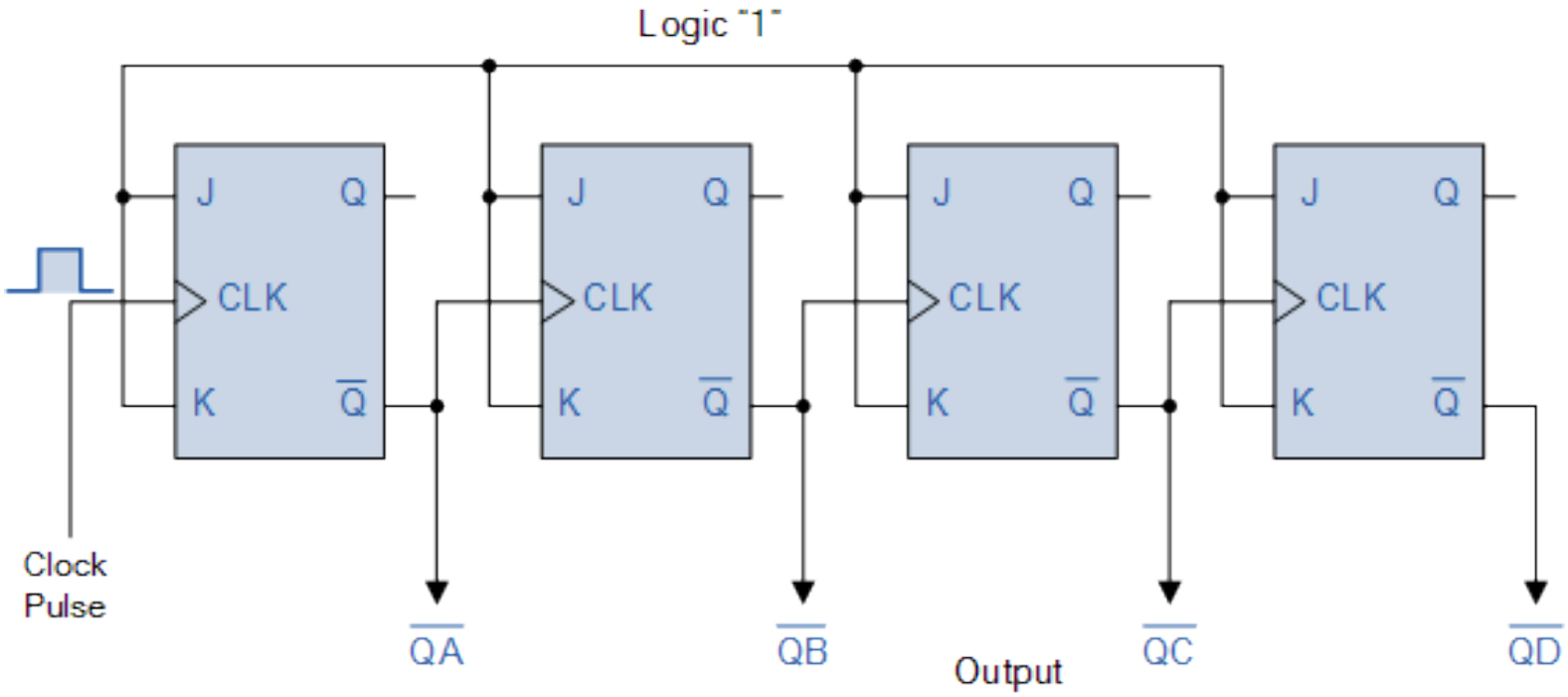
# COUNTERS

## 4-BIT ASYNC. UP COUNTER



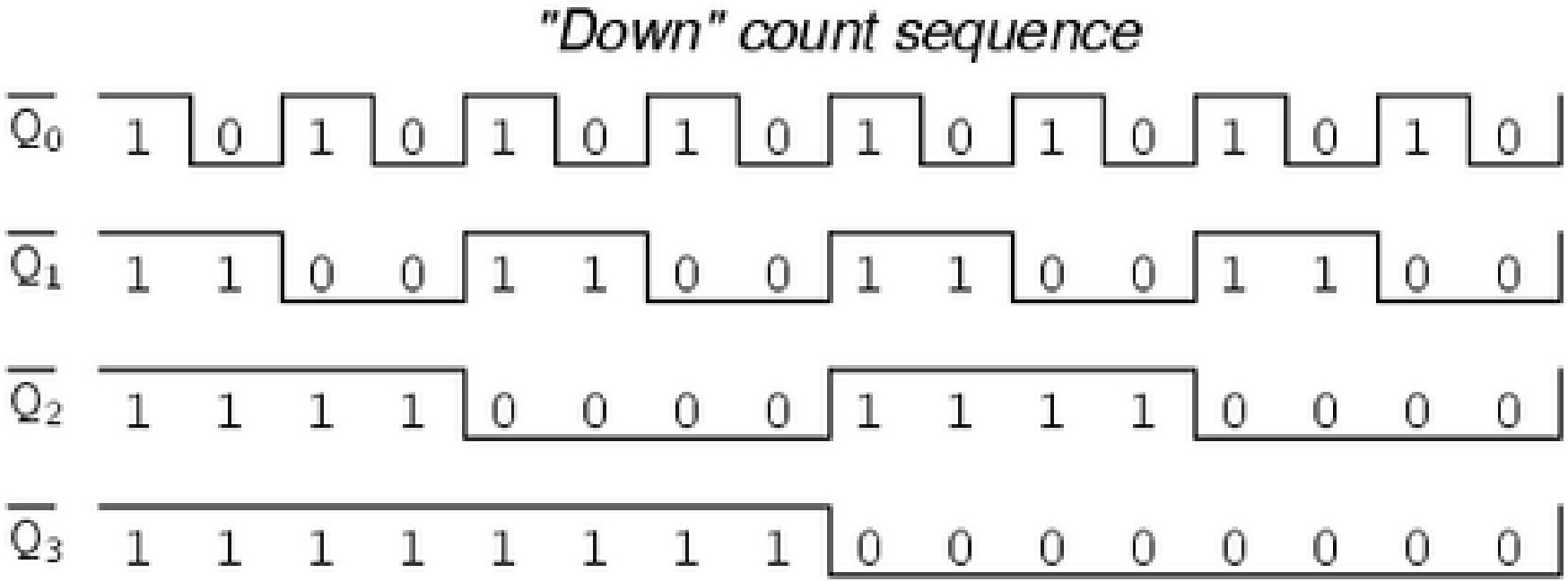
# COUNTERS

## 4-BIT ASYNC. DOWN COUNTER



# COUNTERS

## 4-BIT ASYNC. DOWN COUNTER



# COUNTERS

## SYNC. MOD COUNTER DESIGN

- The counter can be designed using any types of Flip flop. But in general T-Flip flop is used to design counter.
- The use of MOD counter in to counter value for specific number of times.
- For example, MOD-5 Counter means it can counter the values from 0 to 4 and it get reset. So, it count in the sequence of 000,001,0110,011,100,000,001,etc.,
- The number of flip flop required to design MOD counter is depends on the number of count it performs.

# COUNTERS

## SYNC. MOD COUNTER DESIGN

1. Determine the number of Flip-Flop needed
2. Choose the type of Flip Flop and its excitation table
3. Determine Transition table
4. K-Map simplification procedures for driving expressions
5. Draw the logic diagram

# COUNTERS

## Example:1 Design of MOD-6 Counter using JK Flip flop

**Step 1 : Find number of flip-flops required to build the counter.**

Flip-flops required are : $2^n \geq N$.

Here $N = 6 \quad \therefore n = 3$

i.e. Three flip-flops are required.

**Step 2 : Write an excitation table for JK flip-flop.**

| $Q_n$ | $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

# COUNTERS

## Example:1 Design of MOD-6 Counter using JK Flip flop

**Step 3 : Determine the transition table.**

| Present state | | | Next state | | | Flip-flop inputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_A$ | $Q_B$ | $Q_C$ | $Q_{A+1}$ | $Q_{B+1}$ | $Q_{C+1}$ | $J_A$ | $K_A$ | $J_B$ | $K_B$ | $J_C$ | $K_C$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | x | 0 | x | 1 | x |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | x | 1 | x | x | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | x | x | 0 | 1 | x |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | x | x | 1 | x | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | x | 0 | 0 | x | 1 | x |
| 1 | 0 | 1 | 0 | 0 | 0 | x | 1 | 0 | x | x | 1 |
| 1 | 1 | 0 | x | x | x | x | x | x | x | x | x |
| 1 | 1 | 1 | x | x | x | x | x | x | x | x | x |

# COUNTERS

## Example:1 Design of MOD-6 Counter using JK Flip flop

**Step 4 : K-map simplification for flip-flop inputs.**

For $J_A$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | X | X | X | X |

$J_A = Q_B Q_C$

For $K_A$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | X | X | X |
| 1 | 0 | 1 | X | X |

$K_A = Q_C$

For $J_B$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | X | X |
| 1 | 0 | 0 | X | X |

$J_B = \overline{Q}_A Q_C$

For $K_B$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | X | 1 | 0 |
| 1 | X | X | X | X |

$K_B = Q_C$

For $J_C$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | X | X | 1 |
| 1 | 1 | X | X | X |

$J_C = 1$

For $K_C$

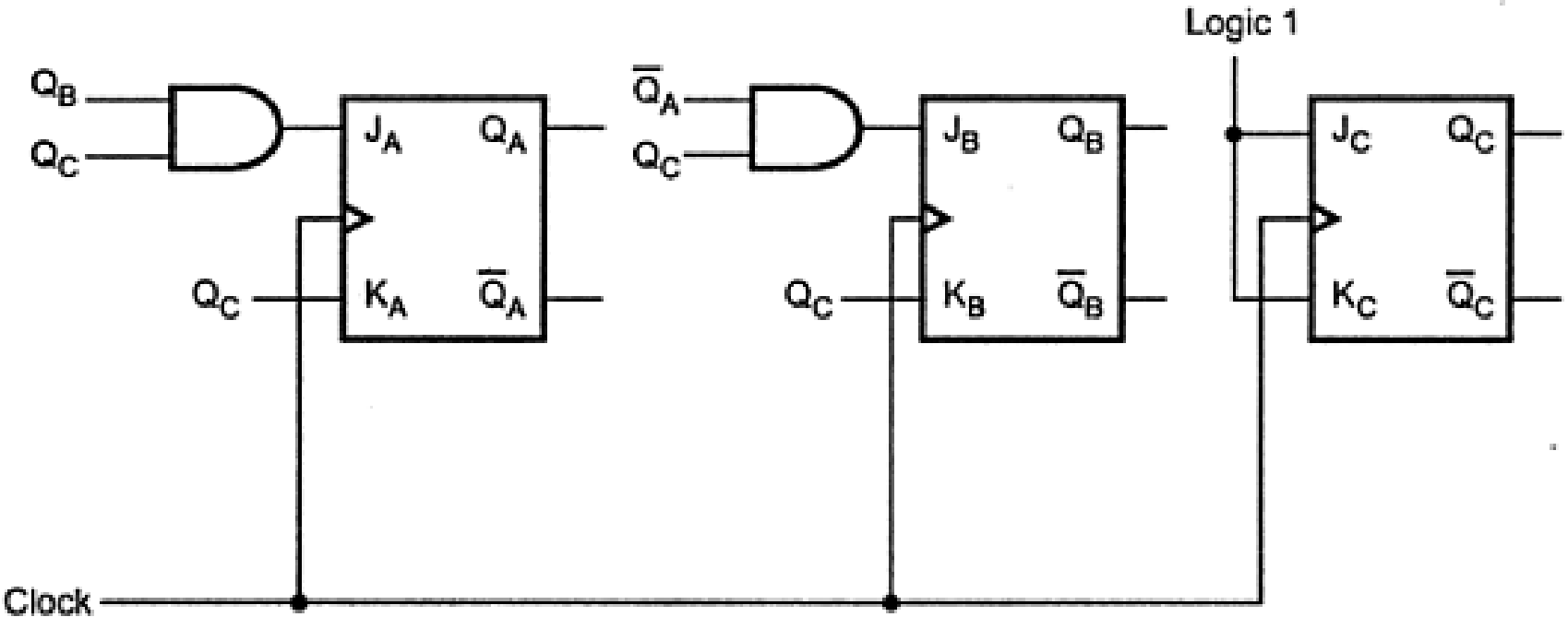
| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | 1 | 1 | X |
| 1 | X | 1 | X | X |

$K_C = 1$

# COUNTERS

## Example:1 Design of MOD-6 Counter using JK Flip flop

**Step 5 : Implement the counter.**



# COUNTERS

## Example:2 Design of MOD-6 Counter using T Flip flop

**Step 1 : Find number of flip-flops required to build the counter.**

Flip-flops required are : $2^n \geq N$

Here $N = 6 \quad \therefore \quad n = 3$

i.e. Three flip-flops are required.

**Step 2 : Write an excitation table for T flip-flop.**

| $Q_n$ | $Q_{n+1}$ | T |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# COUNTERS

## Example:2 Design of MOD-6 Counter using T Flip flop

**Step 3 : Determine the transition table.**

| Present state | | | Next state | | | Flip-flop inputs | | |
|---|---|---|---|---|---|---|---|---|
| $Q_A$ | $Q_B$ | $Q_C$ | $Q_{A+1}$ | $Q_{B+1}$ | $Q_{C+1}$ | $T_A$ | $T_B$ | $T_C$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | x | x | x | x | x | x |
| 1 | 1 | 1 | x | x | x | x | x | x |

# COUNTERS

## Example:2 Design of MOD-6 Counter using T Flip flop

**Step 4 : K-map simplification for flip-flop inputs.**

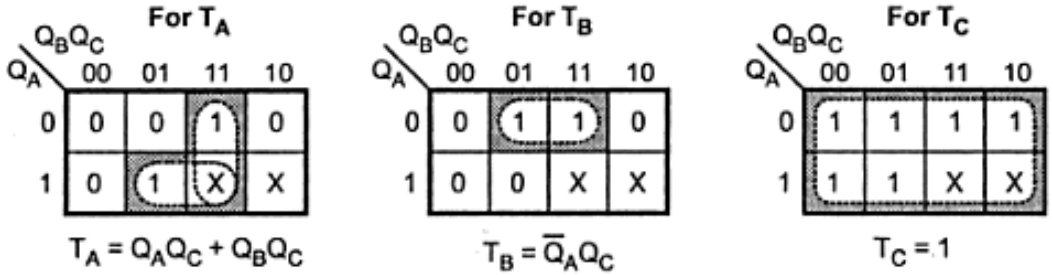
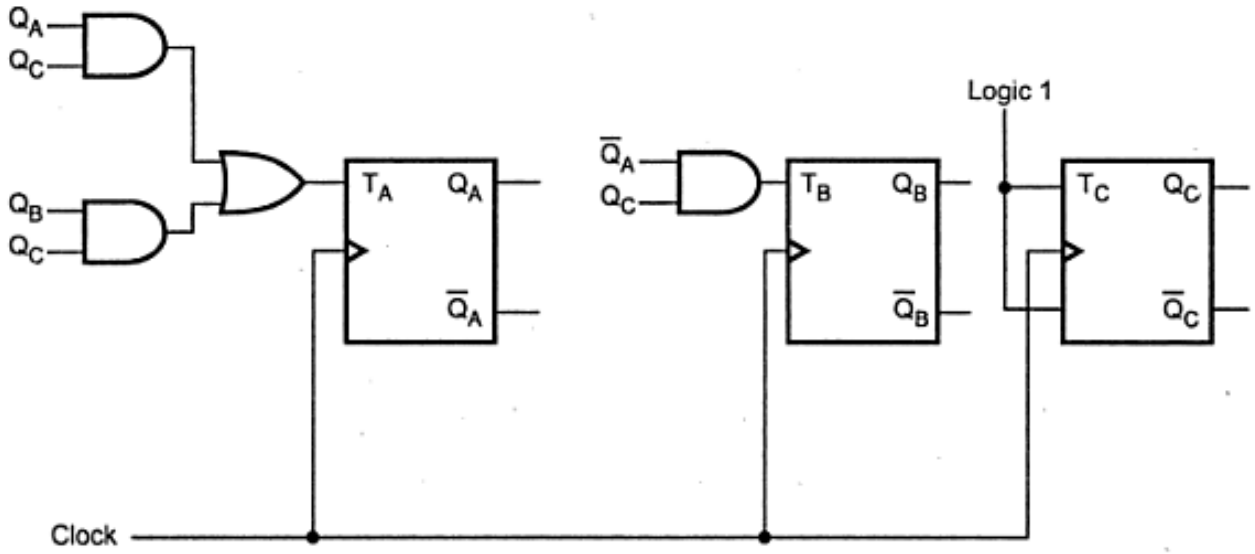
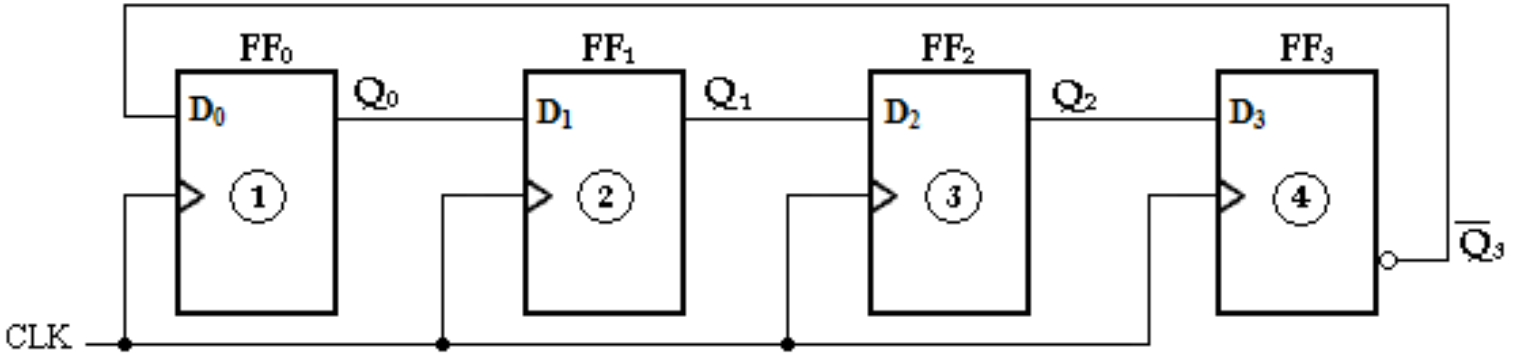



**Fig. 8.82**

**Step 5 : Implement the counter.**



## JOHNSON COUNTERS



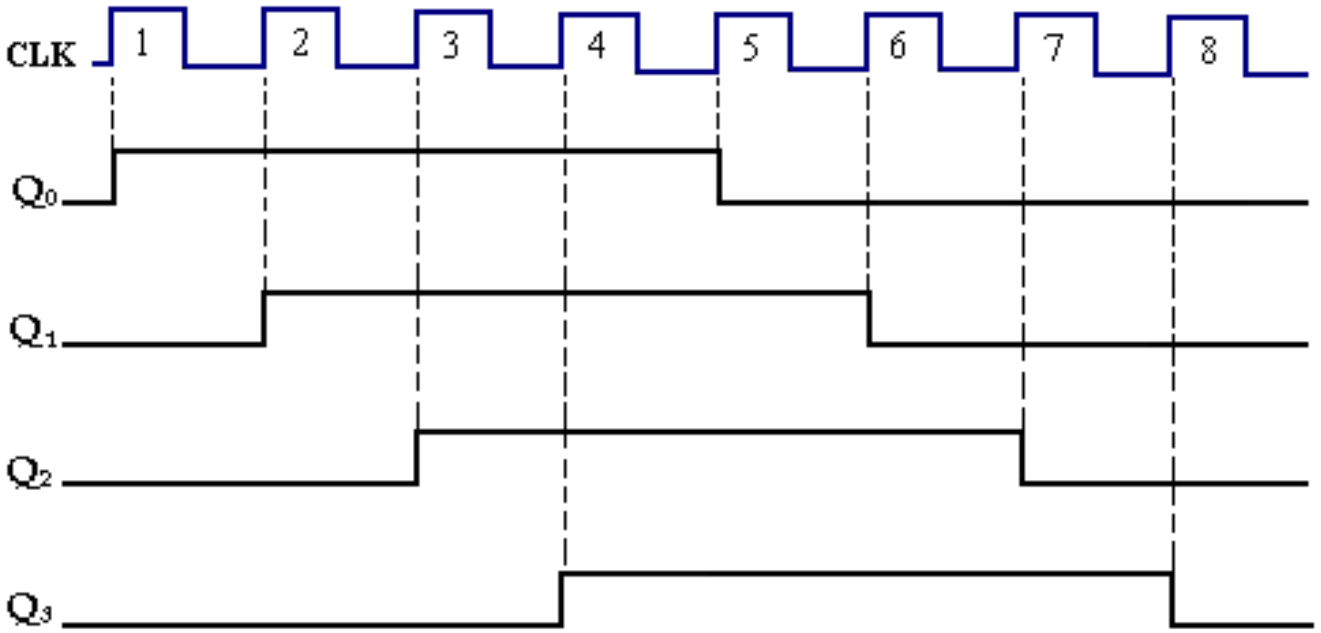

4-Bit Johnson Counter

# COUNTERS

## JOHNSON COUNTERS

| Clock Input | Q3 | Q2 | Q1 | Q0 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 0 |
| 7 | 1 | 1 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 |

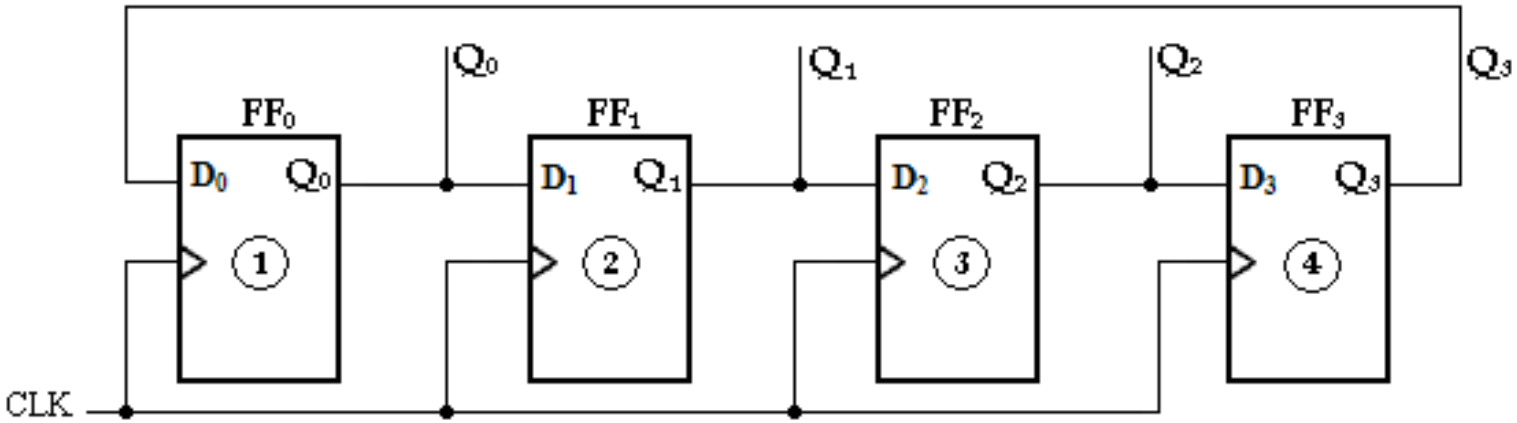
## JOHNSON COUNTERS




Time sequence for a 4-bit Johnson counter

# COUNTERS

## RING COUNTERS




Ring counter

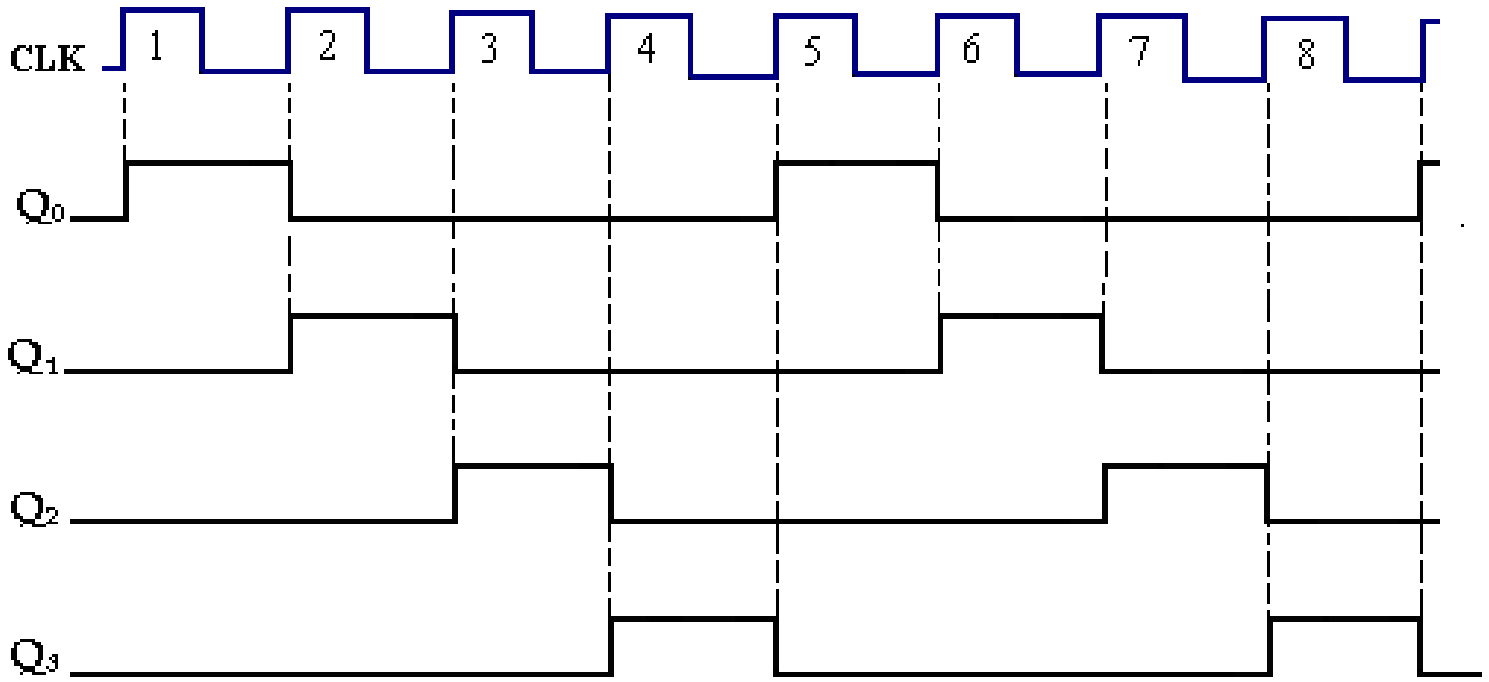
# COUNTERS

## RING COUNTERS

| Clock Input | Q3 | Q2 | Q1 | Q0 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 1 |
| 6 | 0 | 0 | 1 | 0 |
| 7 | 0 | 1 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 |

# COUNTERS

## RING COUNTERS




Time sequence for a Ring counter

# COUNTERS

```
module upordown_counter( Clk,  reset,  UpOrDown,  Count );
  input Clk,reset,UpOrDown;
  output [3 : 0] Count;
  reg [3 : 0] Count = 0;
   always @(posedge(Clk) or posedge(reset))
    begin
      if(reset == 1)
         Count <= 0;
      else
         if(UpOrDown == 1)   //Up mode selected
            if(Count == 15)
               Count <= 0;
            else
               Count <= Count + 1; //Incremend Counter
         else  //Down mode selected
            if(Count == 0)
               Count <= 15;
            else
               Count <= Count - 1; //Decrement counter
   end
  endmodule
```

# COUNTERS

Example: Mod N Counter

```
// # (parameter N = 10,   parameter WIDTH = 4)

module modN_ctr  ( input   clk,   input   rstn,   output  reg[WIDTH-1:0] out);

  always @ (posedge clk) begin
    if (!rstn) begin
      out <= 0;
    end else begin
      if (out == N-1)
        out <= 0;
      else
        out <= out + 1;
    end
  end
endmodule
```

# COUNTERS

```
module tb;
  parameter N = 10;
  parameter WIDTH = 4;
    reg clk;
  reg rstn;
  wire [WIDTH-1:0] out;

  modN_ctr u0  (   .clk(clk),  .rstn(rstn), .out(out));

  always #10 clk = ~clk;

  initial begin
    {clk, rstn} <= 0;
      repeat(2) @ (posedge clk);
    rstn <= 1;
      repeat(20) @ (posedge clk);
    $finish;
  end
endmodule
```