# Digital Electronics & Logic Design

(EC 207)

**Dr. Vivek Garg**

Department of Electronics Engineering

S. V. National Institute of Technology (SVNIT)
Surat

**email**: vivekg@eced.svnit.ac.in; vivekgarg0101@gmail.com

# Course Outline

- **PN DIODE AND TRANSITOR** **(04 Hours)**

  PN Diode Theory, PN Characteristic and Breakdown Region, PN Diode Application as Rectifier, Zener Diode Theory, Zener Voltage Regulator, Diode as Clamper and Clipper, Photodiode Theory, LED Theory, 7 Segment LED Circuit Diagram and Multi Colour LED, LASER Diode Theory and Applications, Bipolar Junction Transistor Theory, Transistor Symbols And Terminals, Common Collector, Emitter and Base Configurations, Different Biasing Techniques, Concept of Transistor Amplifier, Introduction to FET Transistor And Its Feature.

- **WAVESHAPING CIRCUITS AND OPERATIONAL AMPLIFIER** **(06 Hours)**

  Linear Wave Shaping Circuits, RC High Pass and Low Pass Circuits, RC Integrator and Differentiator Circuits, Nonlinear Wave Shaping Circuits, Two Level Diode Clipper Circuits, Clamping Circuits, Operational Amplifier OP-AMP with Block Diagram, Schematic Symbol of OP-AMP, The 741 Package Style and Pinouts, Specifications of Op-Amp, Inverting and Non-Inverting Amplifier, Voltage Follower Circuit, Multistage OP-AMP Circuit, OP-AMP Averaging Amplifier, OP-AMP Subtractor.

- **BOOLEAN ALGEBRA AND SWITCHING FUNCTIONS** **(04 Hours)**

  Basic Logic Operation and Logic Gates, Truth Table, Basic Postulates and Fundamental Theorems of Boolean Algebra, Standard Representations of Logic Functions- SOP and POS Forms, Simplification of Switching Functions-K-Map and Quine-Mccluskey Tabular Methods, Synthesis of Combinational Logic Circuits.

- **COMBINATIONAL LOGIC CIRCUIT USING MSI INTEGRATED CIRCUITS** **(07 Hours)**

# Course Outline

Binary Parallel Adder; BCD Adder; Encoder, Priority Encoder, Decoder; Multiplexer and Demultiplexer Circuits; Implementation of Boolean Functions Using Decoder and Multiplexer; Arithmetic and Logic Unit; BCD to 7-Segment Decoder; Common Anode and Common Cathode 7-Segment Displays; Random Access Memory, Read Only Memory And Erasable Programmable ROMS; Programmable Logic Array (PLA) and Programmable Array Logic (PAL).

- **INTRODUCTION TO SEQUENTIAL LOGIC CIRCUITS** **(04 Hours)**

  Basic Concepts of Sequential Circuits; Cross Coupled SR Flip-Flop Using NAND or NOR Gates; JK Flip-Flop Rise Condition; Clocked Flip-Flop; D-Type and Toggle Flip-Flops; Truth Tables and Excitation Tables for Flip-Flops; Master Slave Configuration; Edge Triggered and Level Triggered Flip-Flops; Elimination of Switch Bounce using Flip-Flops; Flip-Flops with Preset and Clear.

- **SEQUENTIAL LOGIC CIRCUIT DESIGN** **(06 Hours)**

  Basic Concepts of Counters and Registers; Binary Counters; BCD Counters; Up Down Counter; Johnson Counter, Module-N Counter; Design of Counter Using State Diagrams and Table; Sequence Generators; Shift Left and Right Register; Registers With Parallel Load; Serial-In-Parallel-Out (SIPO) And Parallel-In-Serial-Out(PISO); Register using Different Type of Flip-Flop.

- **REGISTER TRANSFER LOGIC** **(04 Hours)**

  Arithmetic, Logic and Shift Micro-Operation; Conditional Control Statements; Fixed-Point and Floating-Point Data; Arithmetic Shifts; Instruction Code and Design Of Simple Computer.

- **PROCESSOR LOGIC DESIGN** **(03 Hours)**

  Processor Organization; Design of Arithmetic Logic Unit; Design of Accumulator.

- **CONTROL LOGIC DESIGN** **(04 Hours)**

  Control Organization; Hard-Wired Control; Micro Program Control; Control Of Processor Unit; PLA Control.

# Course Text and Materials


1. Schilling Donald L. and Belove E., "Electronics Circuits- Discrete and Integrated", 3rd Ed., McGraw-Hill, 1989, Reprint 2008.
2. Millman Jacob, Halkias Christos C. and Parikh C., "Integrated Electronics", 2nd Ed., McGraw-Hill, 2009.
3. Taub H. and Mothibi Suryaprakash, Millman J., "Pulse, Digital and Switching Waveforms", 2nd Ed., McGraw-Hill, 2007.
4. Mano Morris, “Digital Logic and Computer Design”, 5th Ed., Pearson Education, 2005.
5. Lee Samual, “Digital Circuits and Logic Design”, 1st Ed., PHI, 1998.

## Shift Registers

- **An array of flip flop is used to store a group of bits.**
- **To store n bits, n flip flops are cascaded.**
- **Types:** 1. Based on input and outputs

  Serial In Serial Out

  Serial in Parallel Out

  Parallel in Serial Out

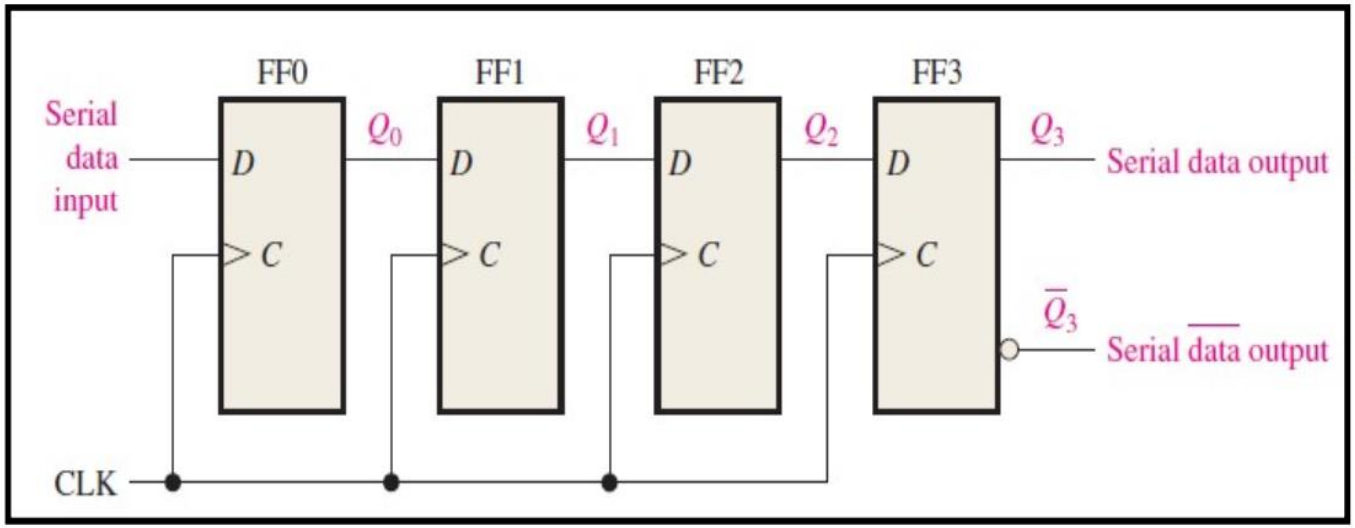
  Parallel in Parallel Out

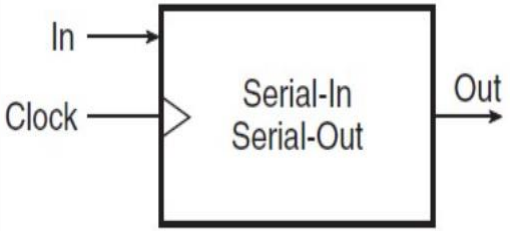
  2. Based on application

  Shift Register

  Storage Register

## Serial In Serial Out



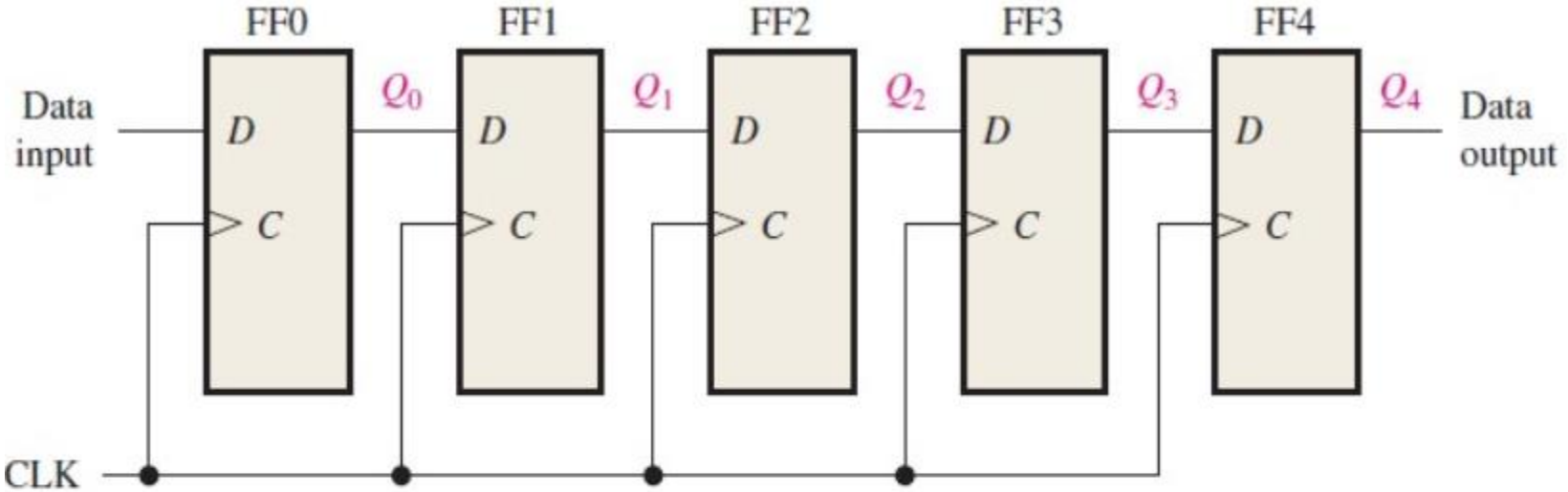

- For n bit storage n clock pulse is required.
- It provides n clock pulse delay
- To get n bit serial out, n-1 clock pulses are required.

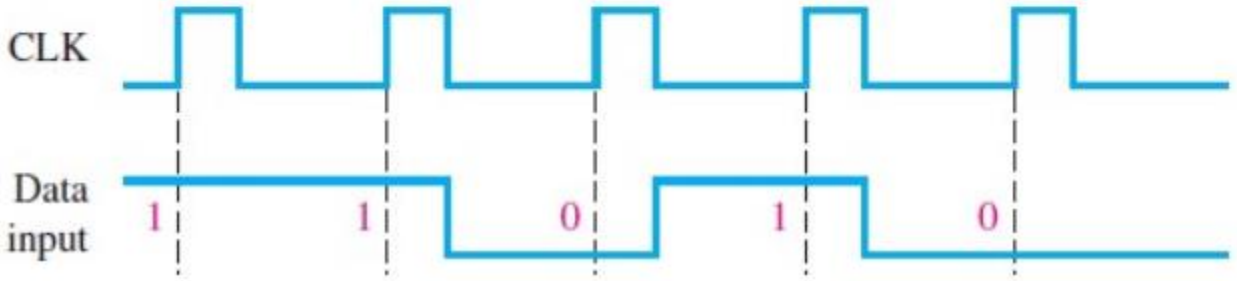
## Serial In Serial Out

| CLK | FF0 ($Q_0$) | FF1 ($Q_1$) | FF2 ($Q_2$) | FF3 ($Q_3$) |
|---|---|---|---|---|
| Initial | 0 | 0 | 0 | 0 |
| 1 | **0** | 0 | 0 | 0 |
| 2 | **1** | **0** | 0 | 0 |
| 3 | **0** | **1** | **0** | 0 |
| 4 | **1** | **0** | **1** | **0** |

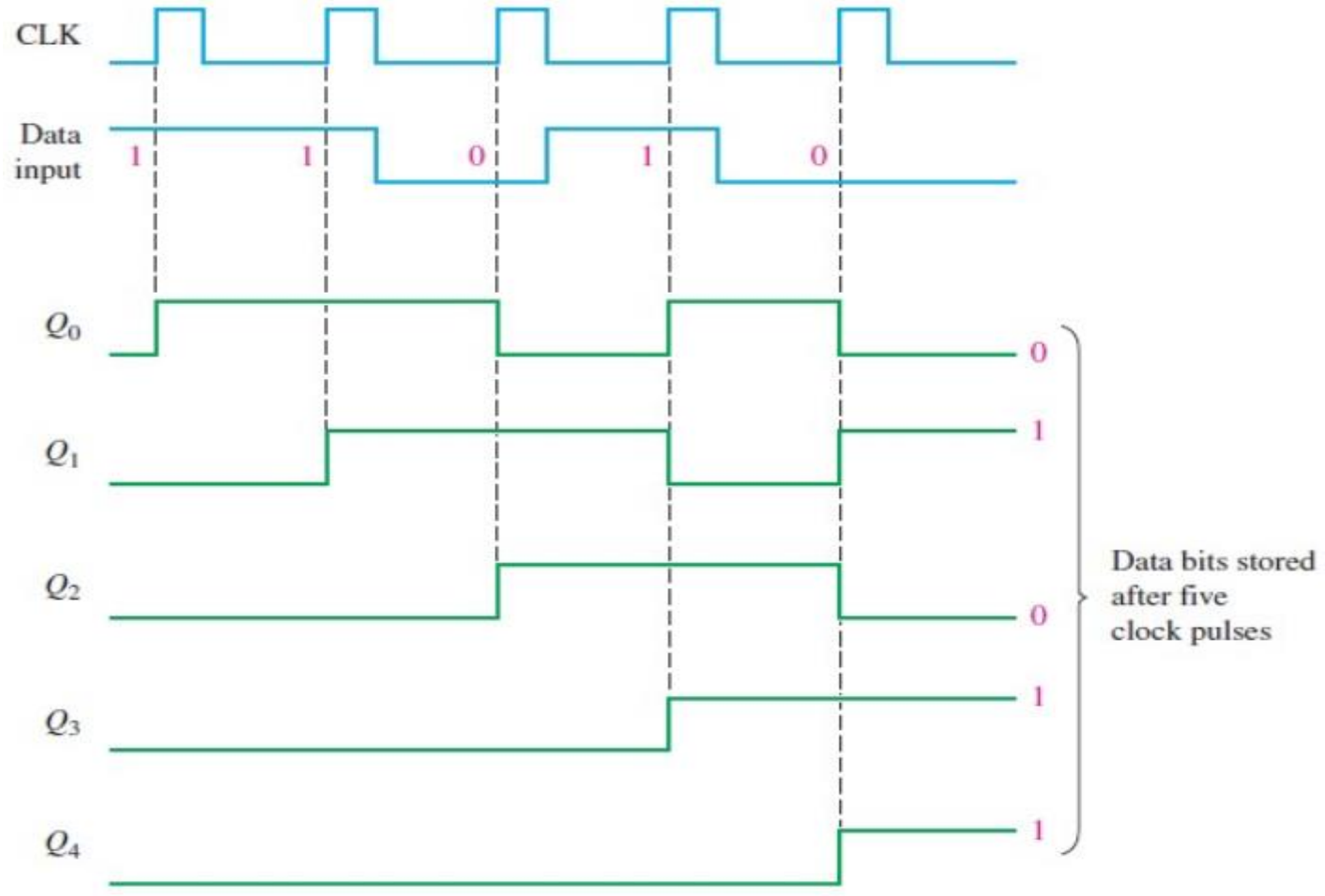
| CLK | FF0 ($Q_0$) | FF1 ($Q_1$) | FF2 ($Q_2$) | FF3 ($Q_3$) |
|---|---|---|---|---|
| Initial | **1** | **0** | **1** | **0** |
| 5 | 0 | **1** | **0** | **1** |
| 6 | 0 | 0 | **1** | **0** |
| 7 | 0 | 0 | 0 | **1** |
| 8 | 0 | 0 | 0 | 0 |

## Serial In Serial Out

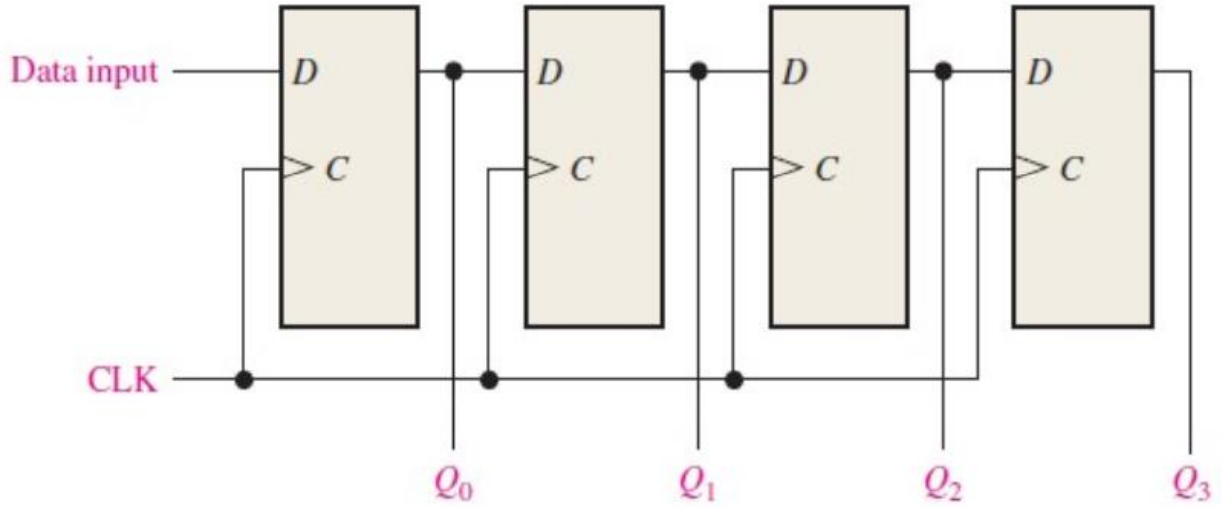


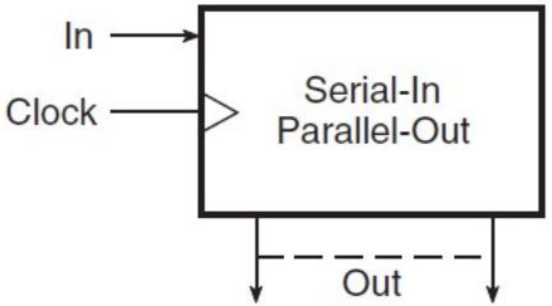
## Serial In Serial Out



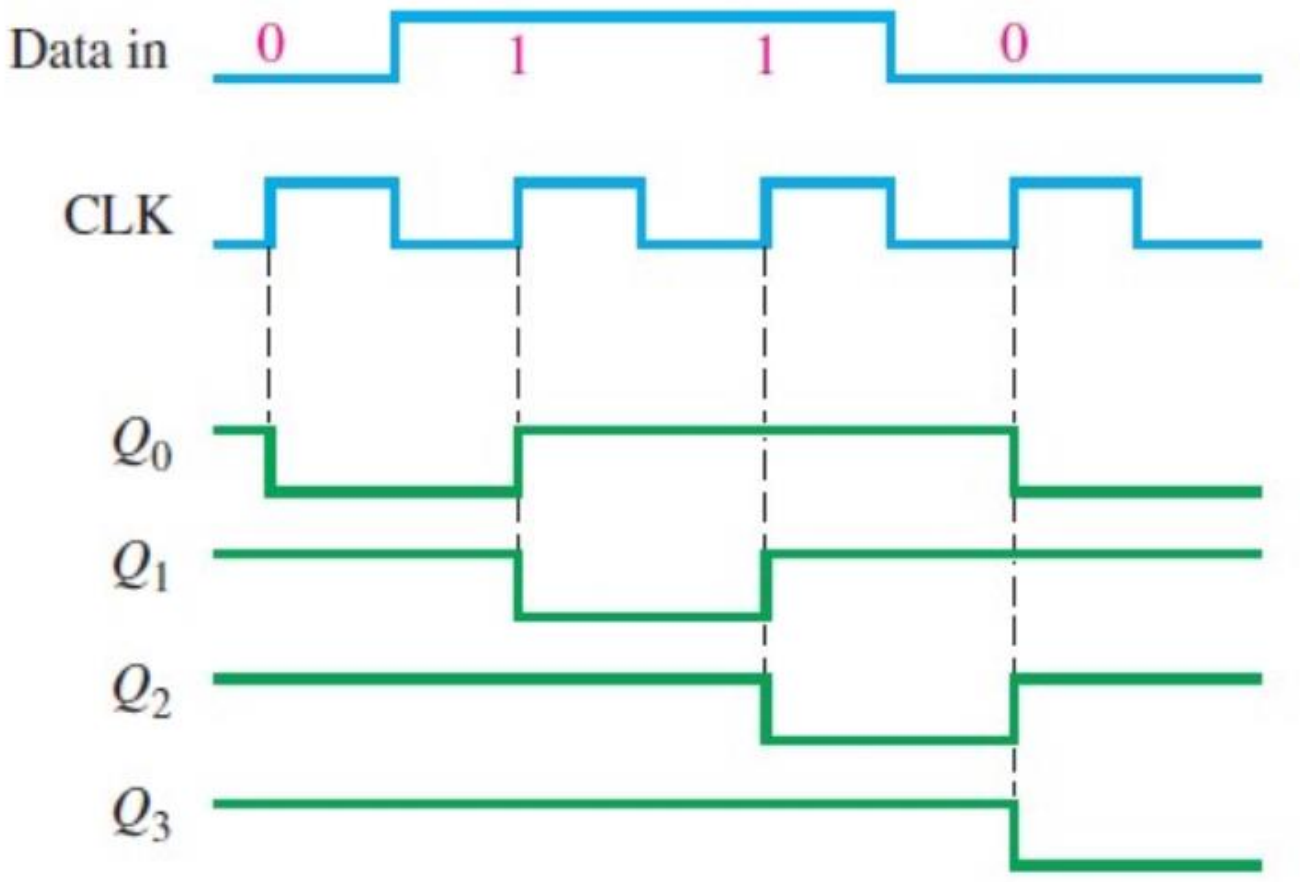
## Serial In Parallel Out



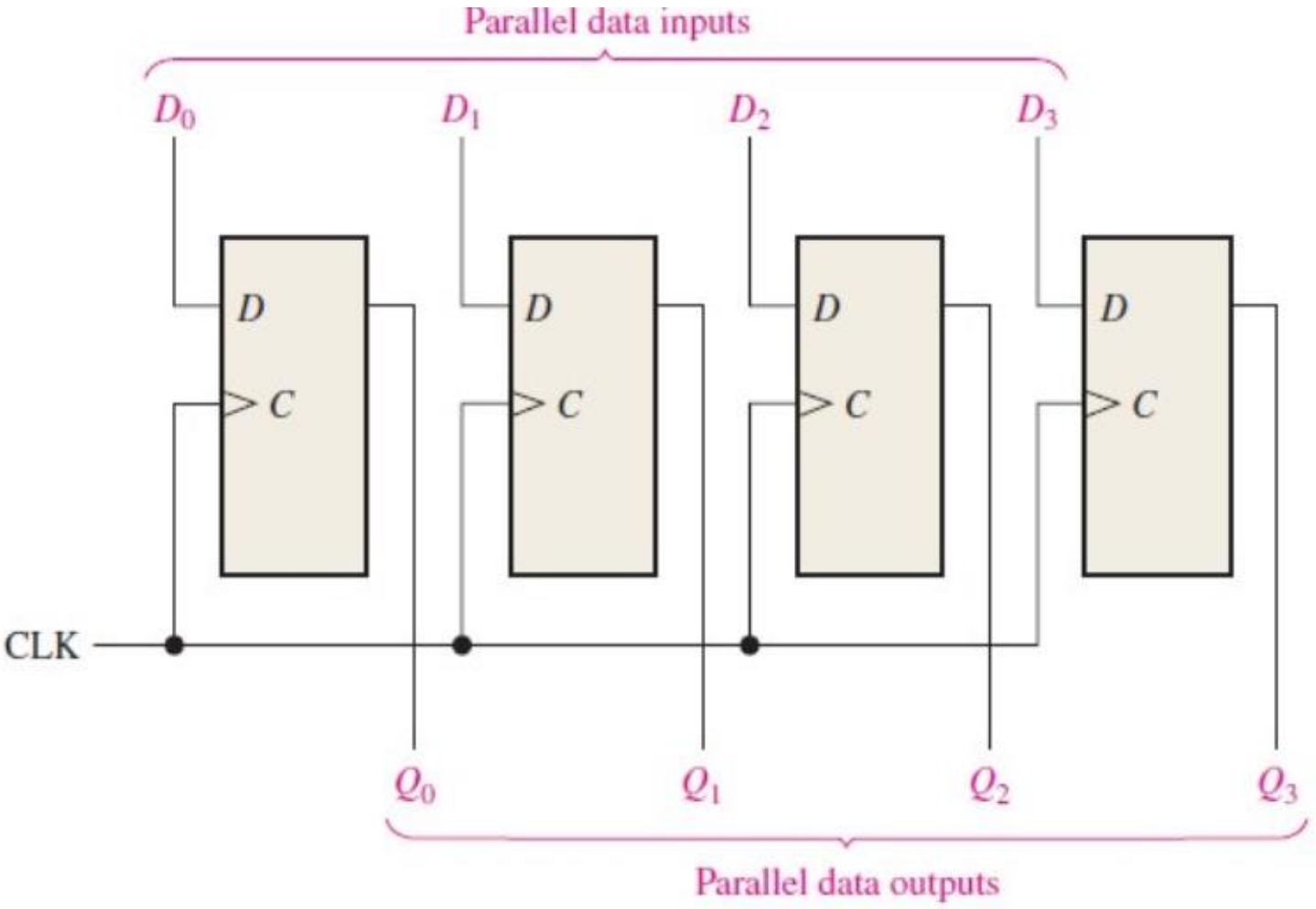

- For n bit storage n clock pulse is required.
- For Parallel output, it requires 0 clock pulses.
- It is used as serial to parallel convertor



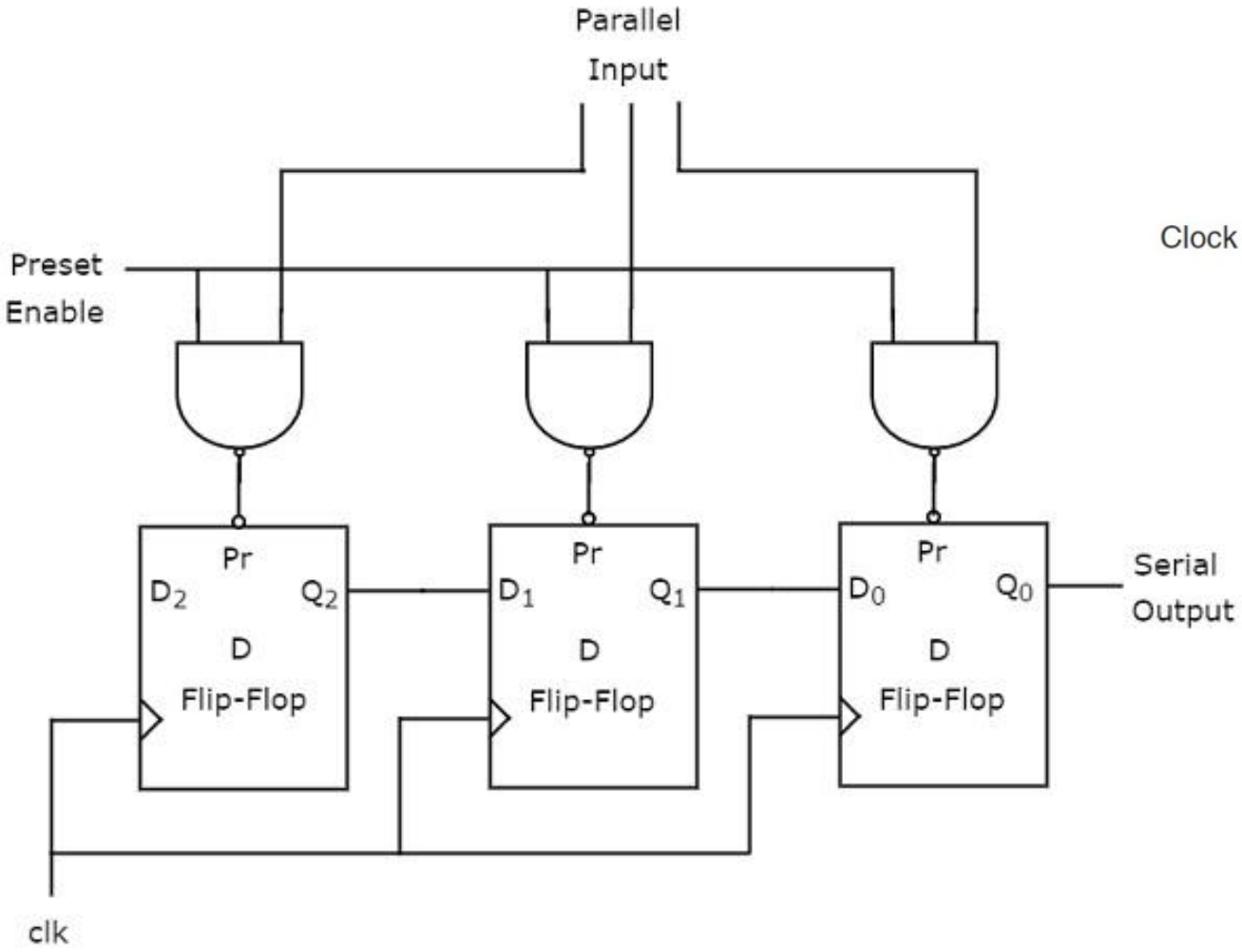
## Serial In Parallel Out

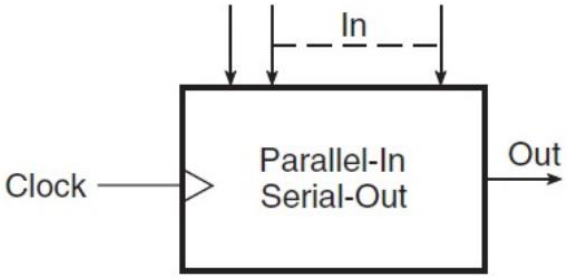


## Parallel In Serial Out

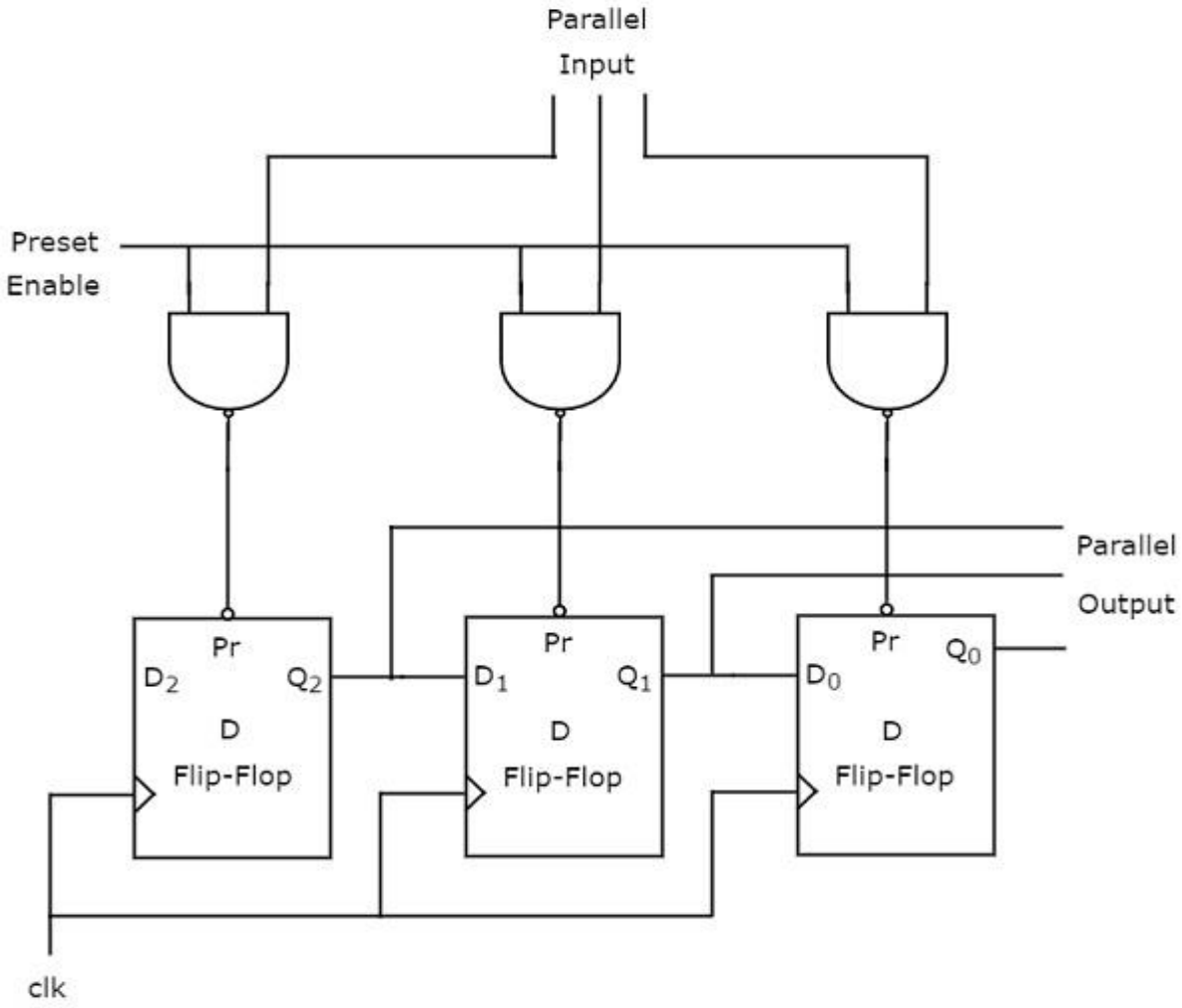


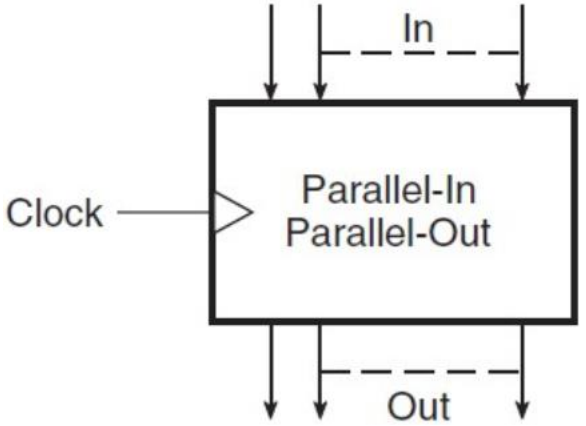
## Parallel In Serial Out



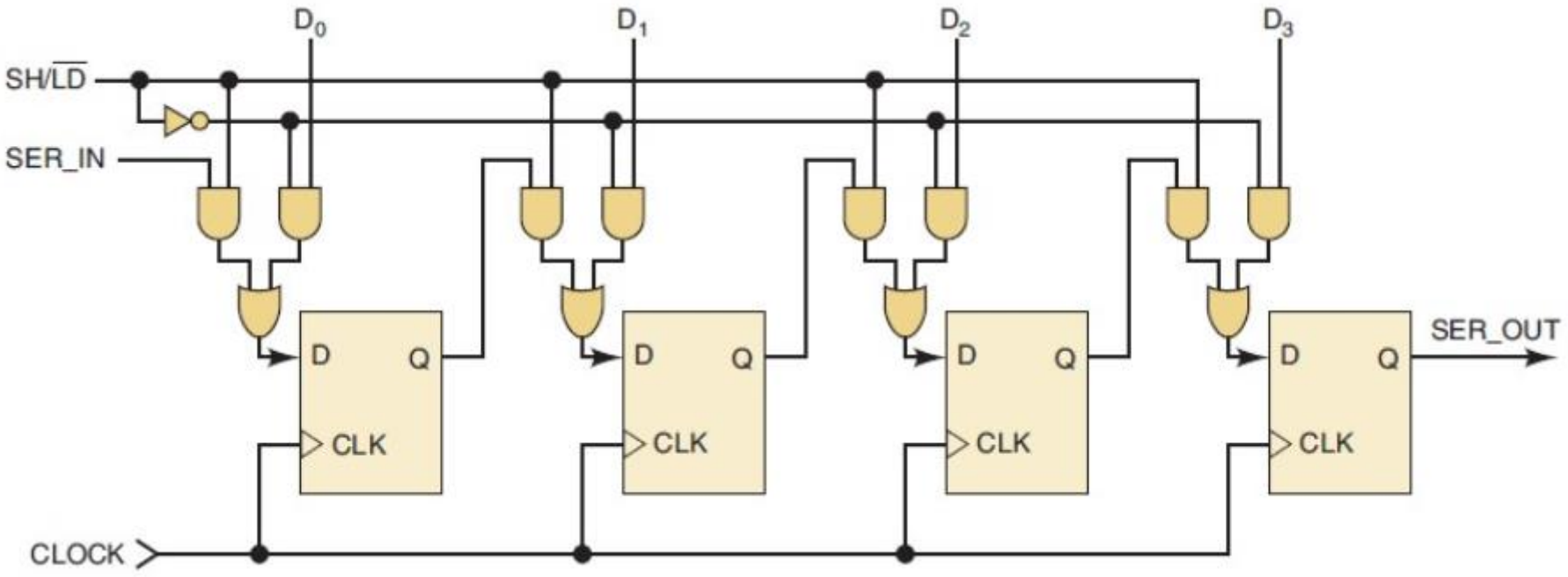



## Parallel In Parallel Out

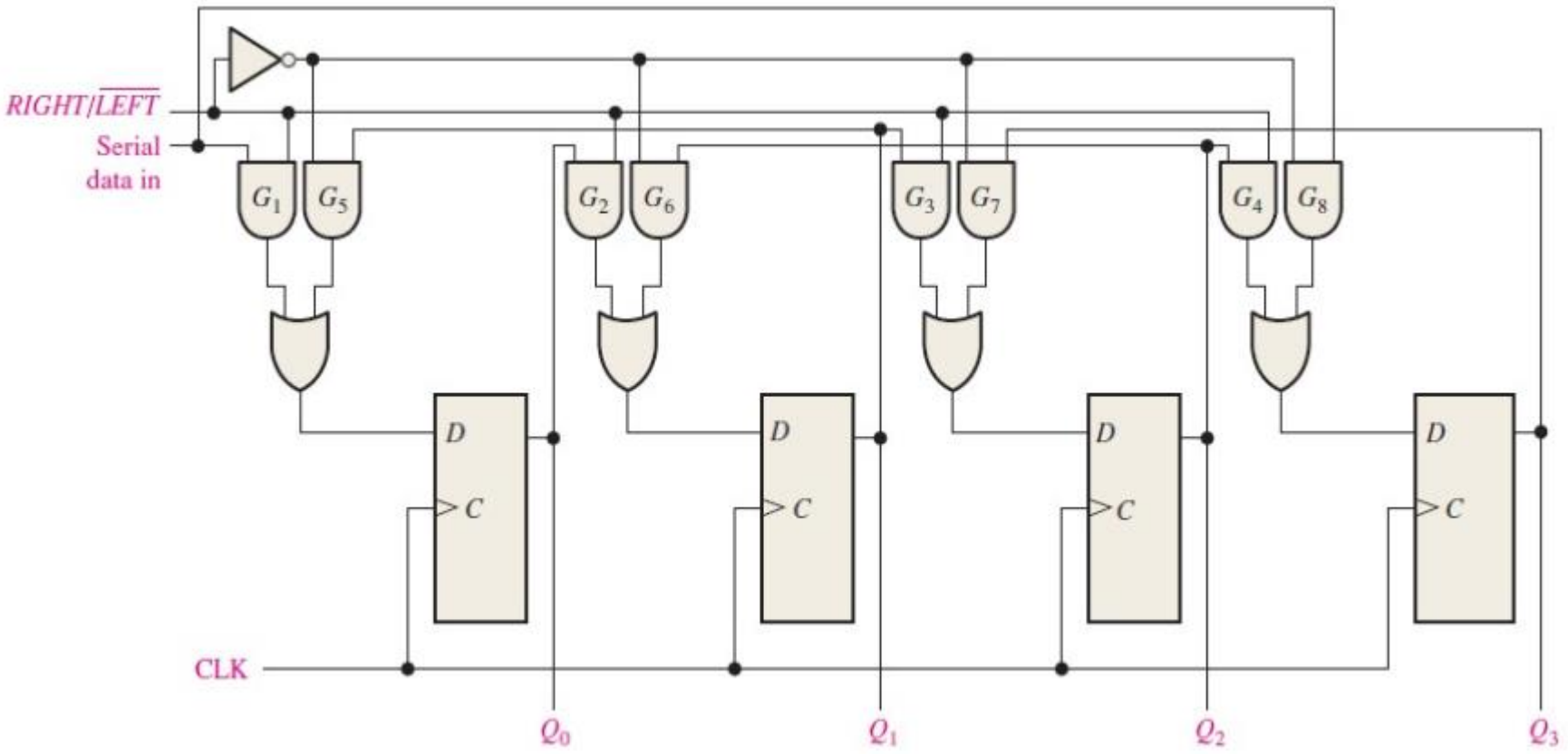


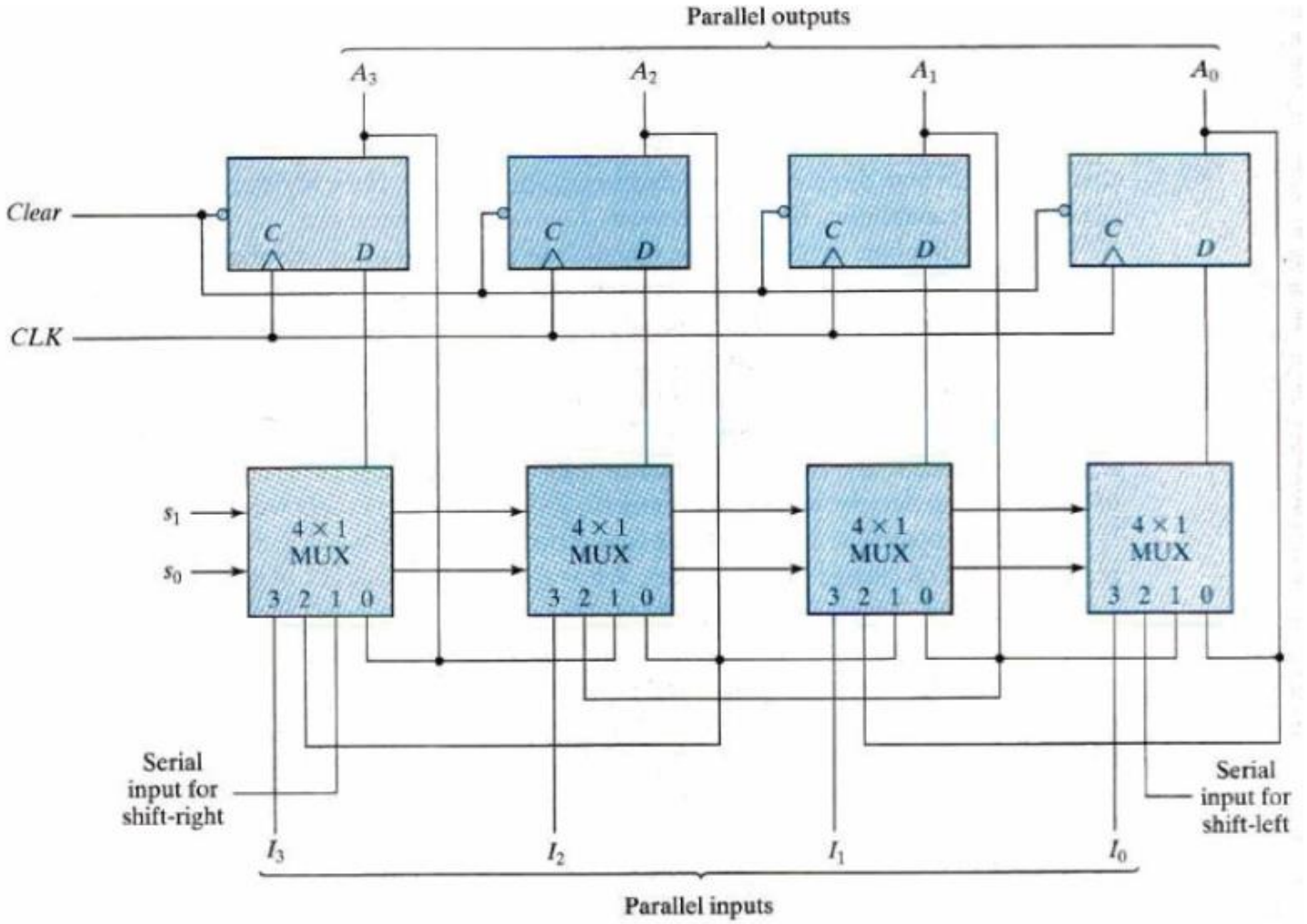
## Parallel/Serial Input and Serial Output



## Bi-Directional Shift Registers

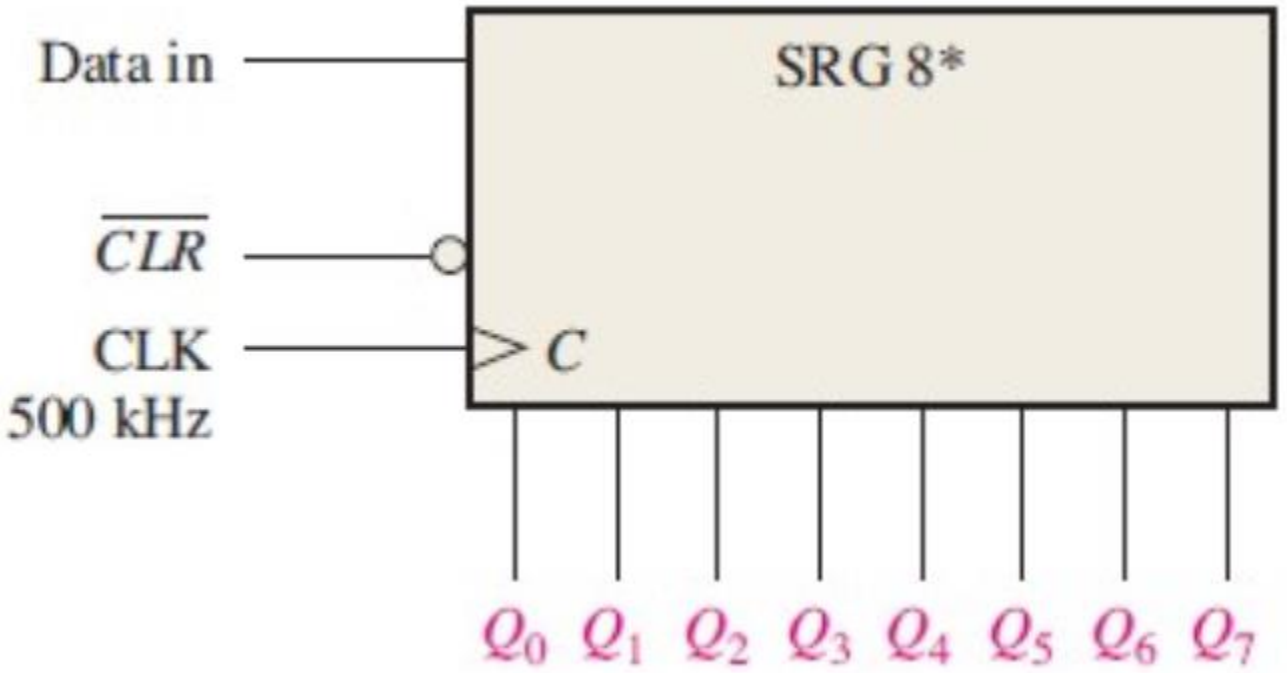


## Universal Shift Registers

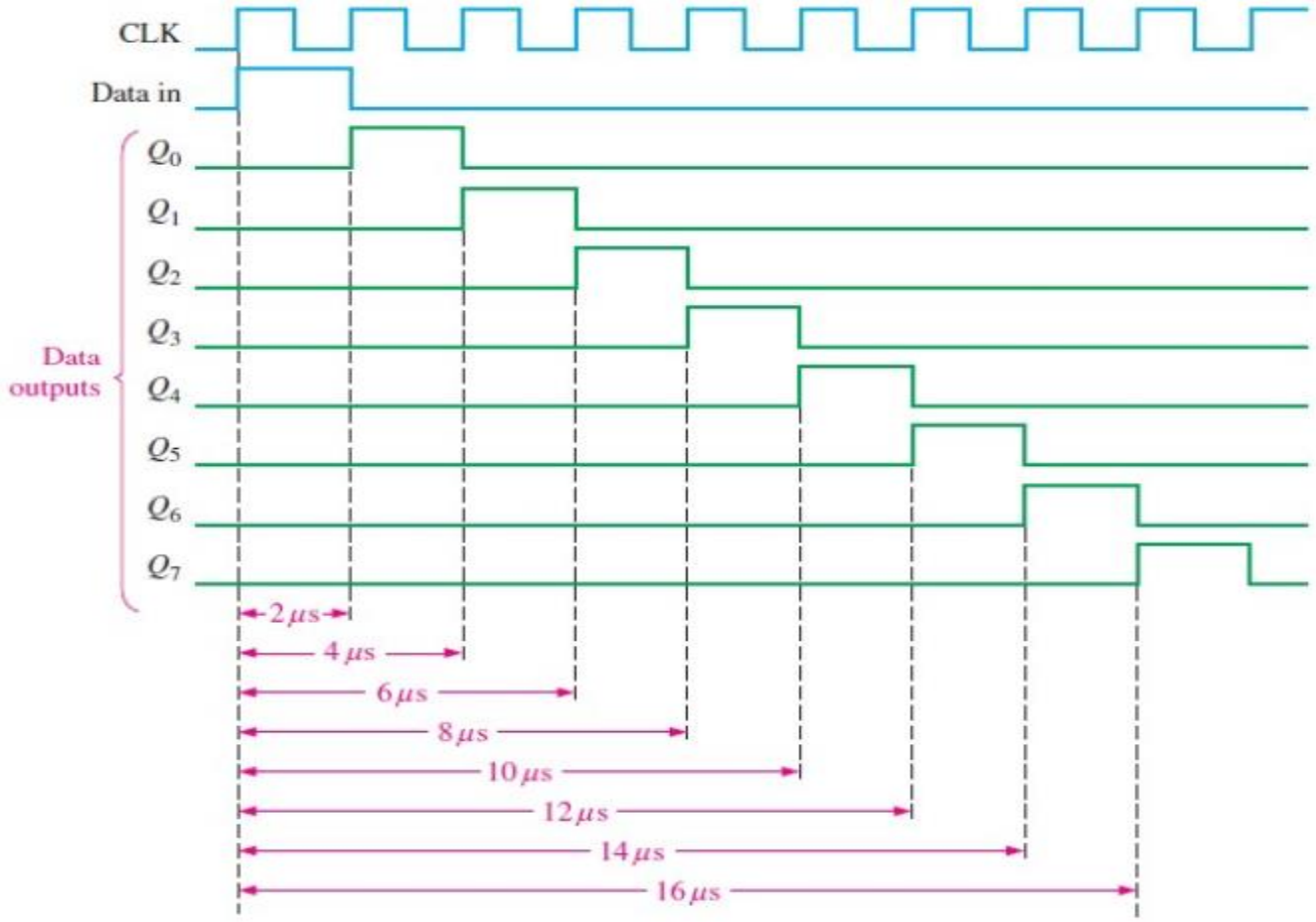



| Mode Control | | |
|---|---|---|
| $s_1$ | $s_0$ | Register Operation |
| 0 | 0 | No change |
| 0 | 1 | Shift right |
| 1 | 0 | Shift left |
| 1 | 1 | Parallel load |

## Shift Registers : Time Delay



## Shift Registers : Time Delay



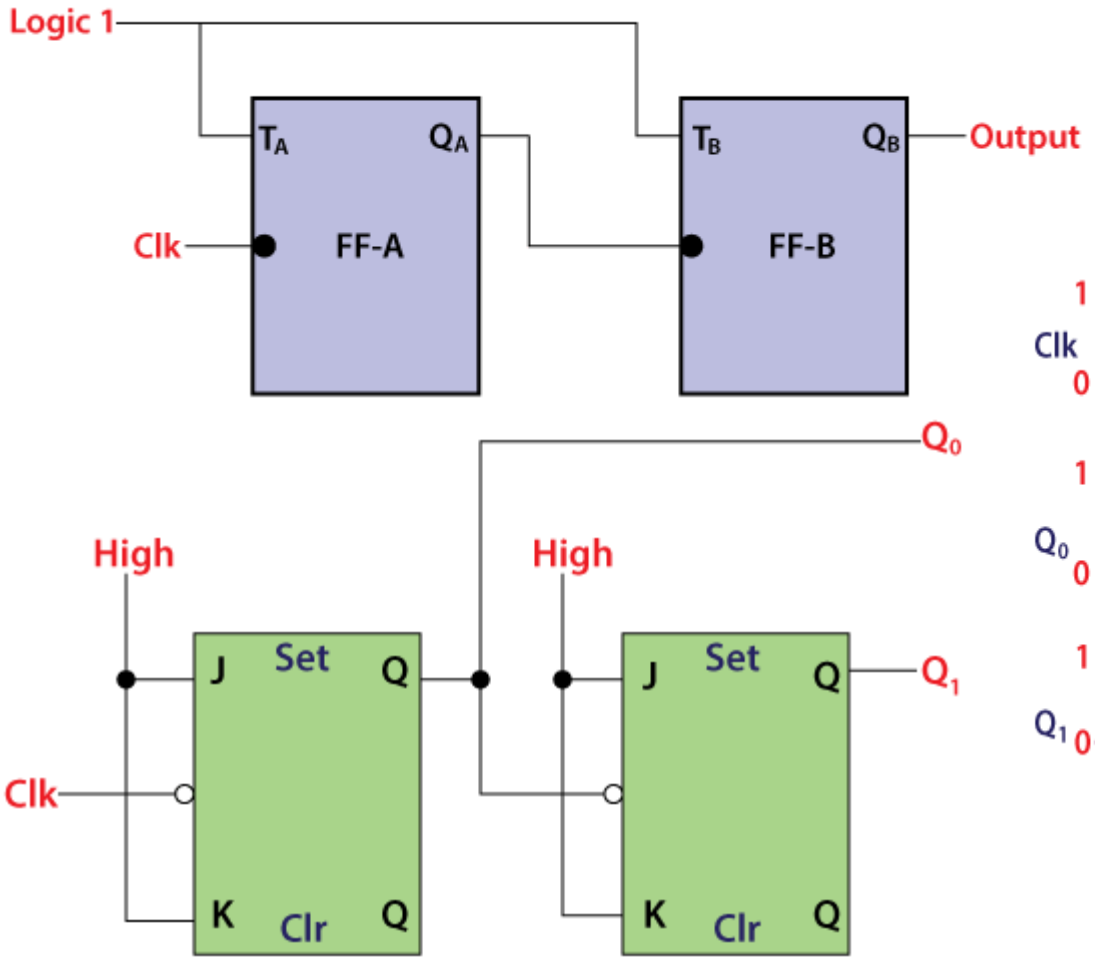
## Counters

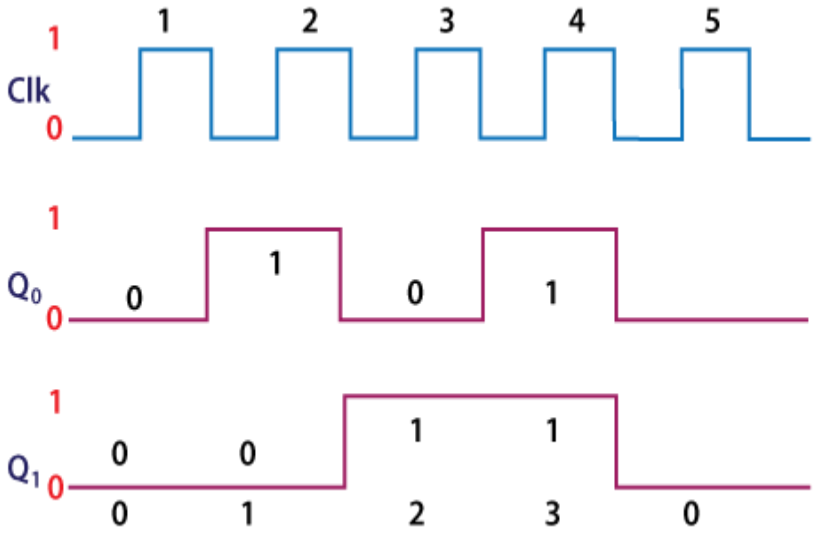
- Used to count number of clock pulse is applied.
- Frequency divider
- Time/Frequency measurement
- Pulse Width
- Waveform Generator
- With ‘n’ FF, maximum possible stages are $2^n$.
- **Types of counters**: (Based on the wav of clock pulse is applied):

  a) Asynchronous

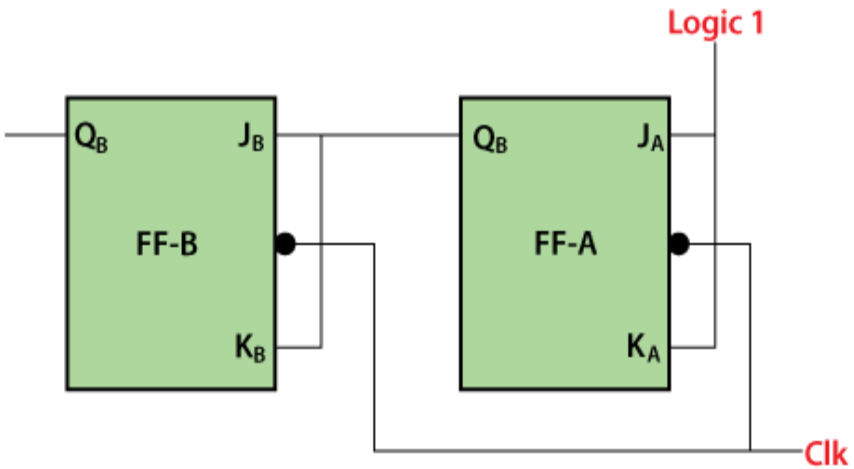
  b) Synchronous

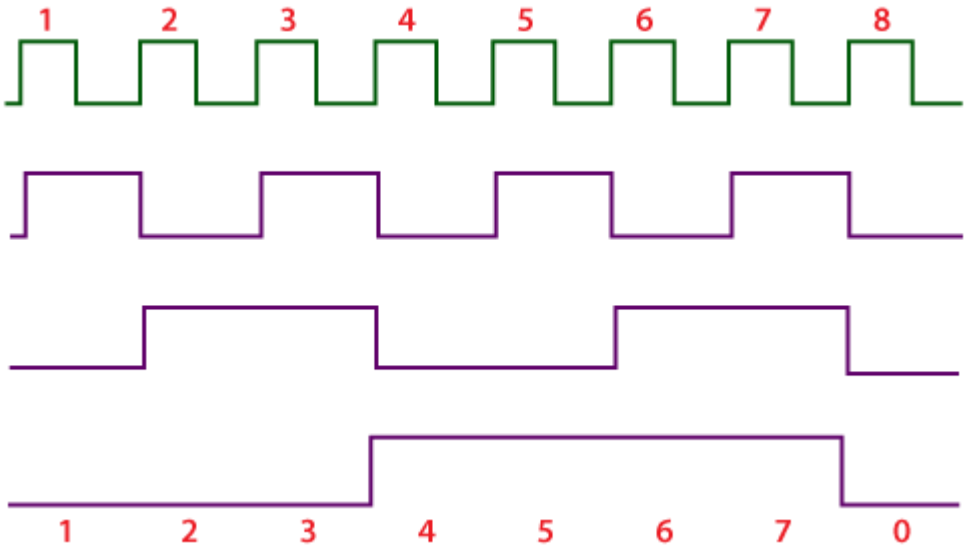
## Asynchronous



## Synchronous



## Ripple/Asynchronous/Serial Counter

- Cascaded arrangement of Flip-Flops where the output of one Flip-Flop drives the input of the following Flip-Flop.
- The clock input to any subsequent flip-flop comes from the output of its immediately preceding flip-flop. For instance, the output of the first flip-flop acts as the clock input to the second flip-flop, the output of the second flip-flop feeds the clock input of the third flip-flop and so on.
- The modulus (MOD number) of a counter is the number of unique states it goes through before it comes back to the initial state to repeat the count sequence.
- An n-bit counter that counts through all its natural states and does not skip any of the states has a modulus of 2n.
- We can see that such counters have a modulus that is an integral power of 2, that is, 2, 4, 8, 16 and so on.
- These can be modified with the help of additional combinational logic to get a modulus of less than 2n.

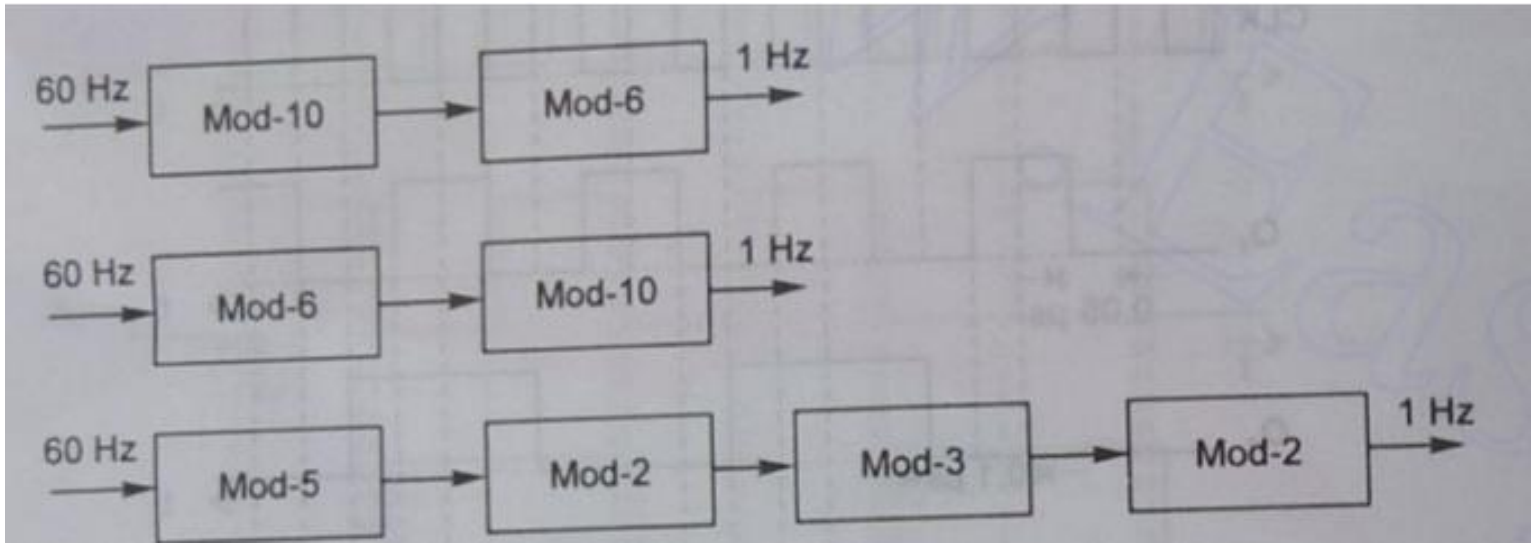
## Ripple/Asynchronous/Serial Counter

- For given modulus value, number of flip flops required:

$$modulus \leq 2^N$$

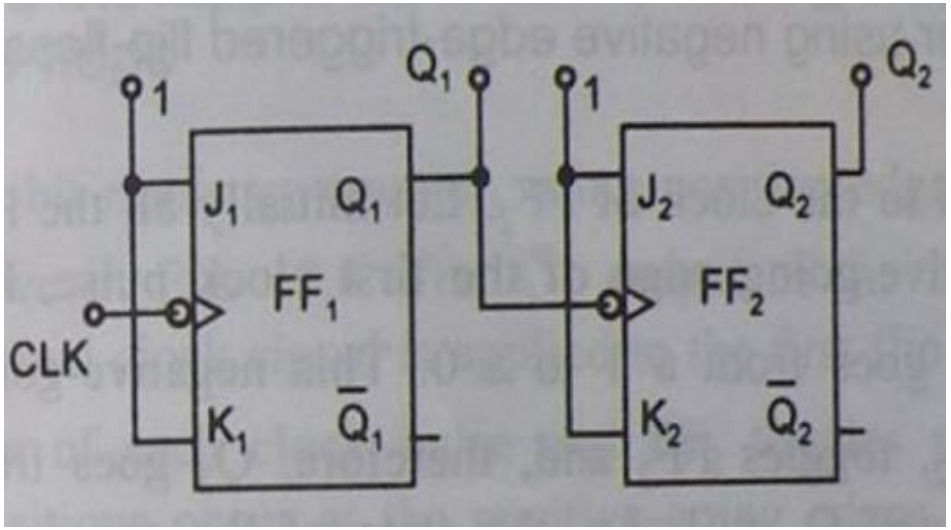
- It is desired to design a binary ripple counter that is capable of counting the number of items passing on a conveyor belt Each time an item passes a given point, a pulse is generated that can be used as a clock input. If the maximum number of items to be counted is 6000, determine the number of flip-flops required.

## Ripple/Asynchronous/Serial Counter

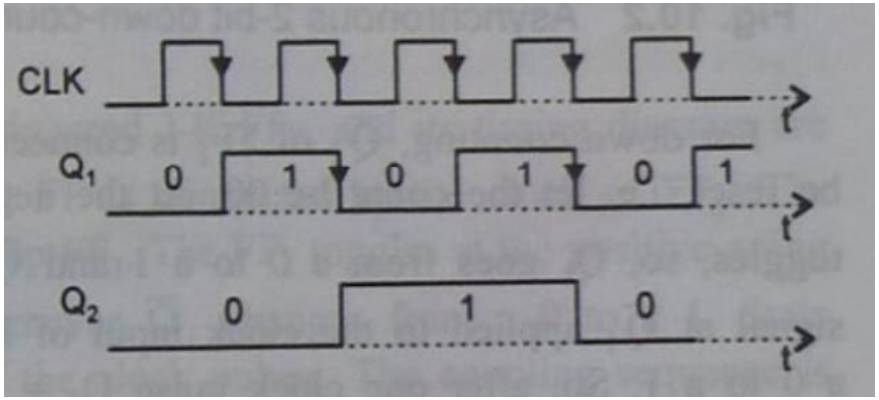


## Ripple/Asynchronous/Serial Counter

2 Bit Negative Edge Triggered UP Counter

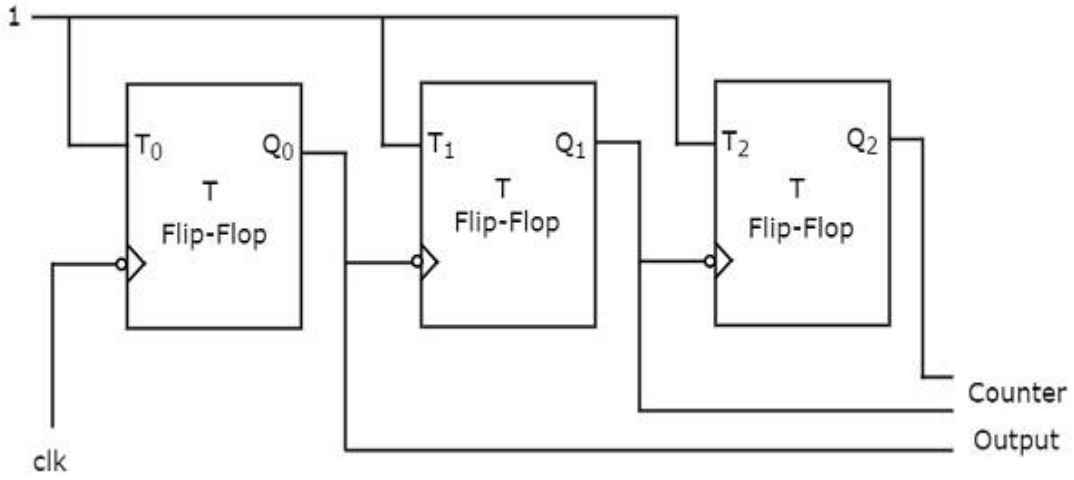



| Clock Pulse | Q2 | Q1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |
| 4 | 0 | 0 |



## Ripple/Asynchronous/Serial Counter

3 Bit Negative Edge Triggered UP Counter

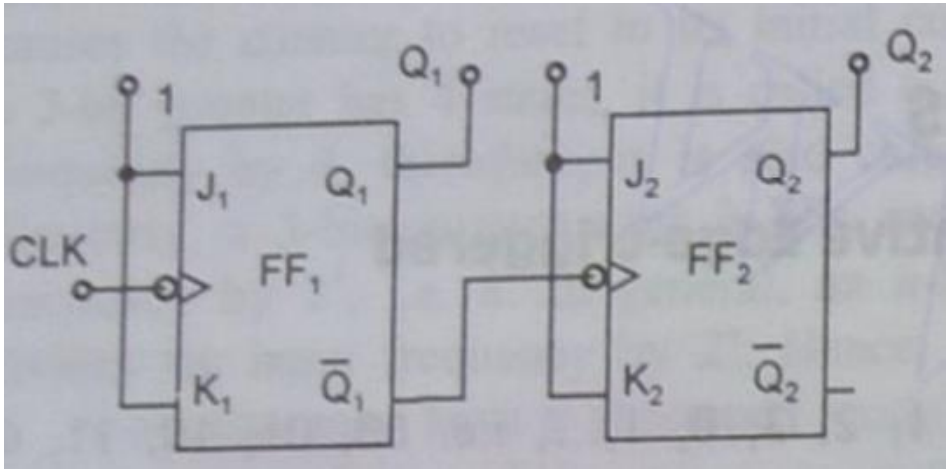


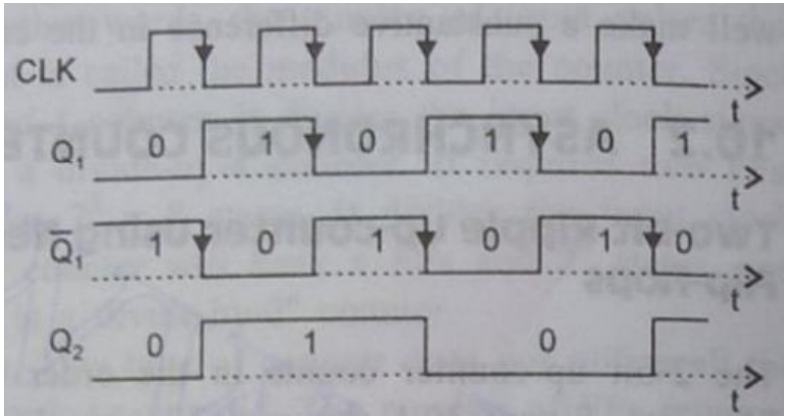

| Clock Pulse No. | Q3 | Q2 | Q1 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |
| 8 | 0 | 0 | 0 |

## Ripple/Asynchronous/Serial Counter

2 Bit Negative Edge Triggered Down Counter

| Clock Pulse | Q2 | Q1 | Q1′ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 |



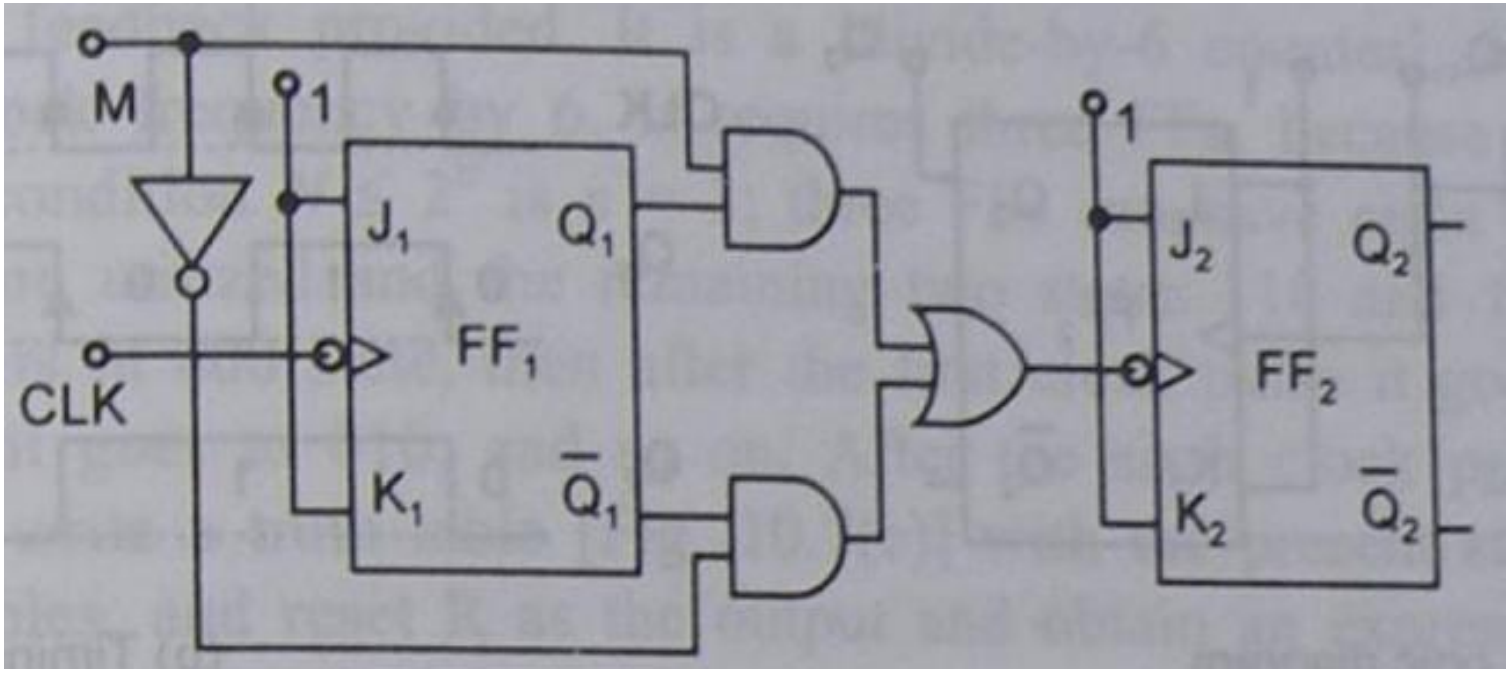



## Ripple/Asynchronous/Serial Counter

3 Bit Negative Edge Triggered Down Counter

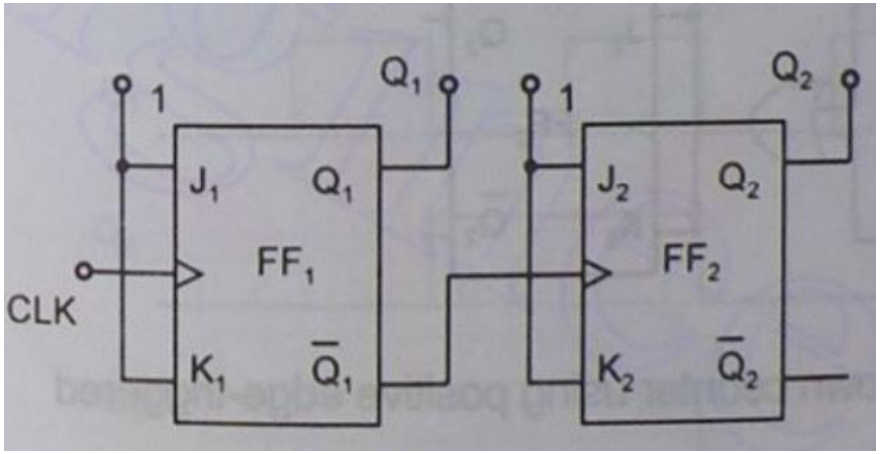
| Clock Pulse No. | Q3 | Q2 | Q2' | Q1 | Q1' |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 2 | 1 | 1 | 0 | 0 | 1 |
| 3 | 1 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 0 | 0 | 1 |
| 7 | 0 | 0 | 1 | 1 | 0 |
| 8 | 0 | 0 | 1 | 0 | 1 |

## Ripple/Asynchronous/Serial Counter (UP/Down Counter)

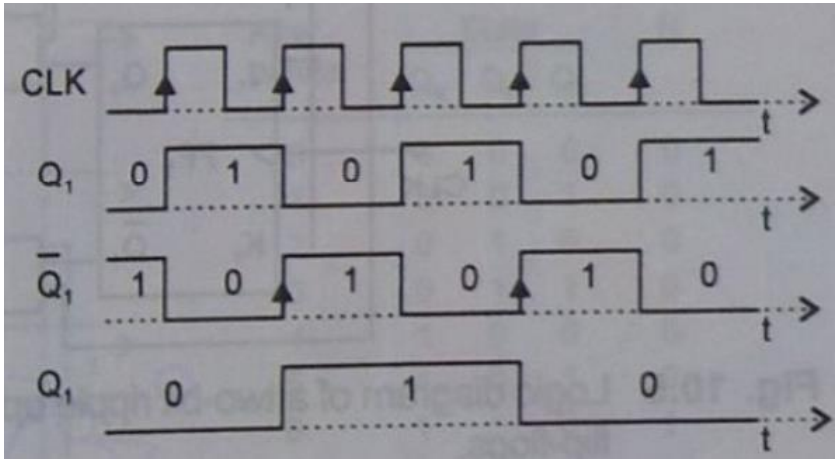


## Ripple/Asynchronous/Serial Counter

2 Bit Positive Edge Triggered Up Counter

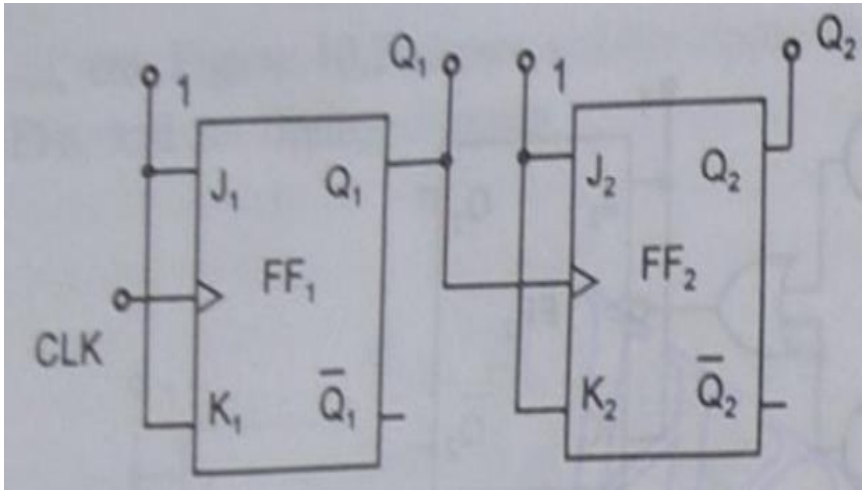



| Clock Pulse | Q2 | Q1 | Q1′ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 |

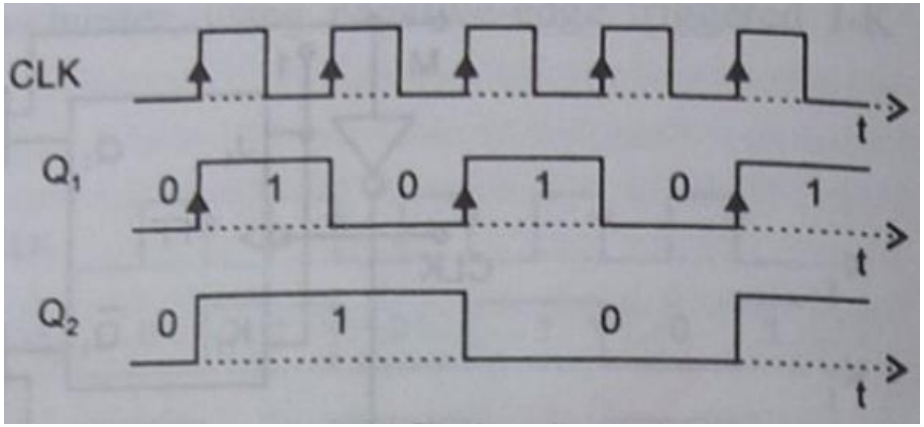


## Ripple/Asynchronous/Serial Counter

2 Bit Positive Edge Triggered Down Counter

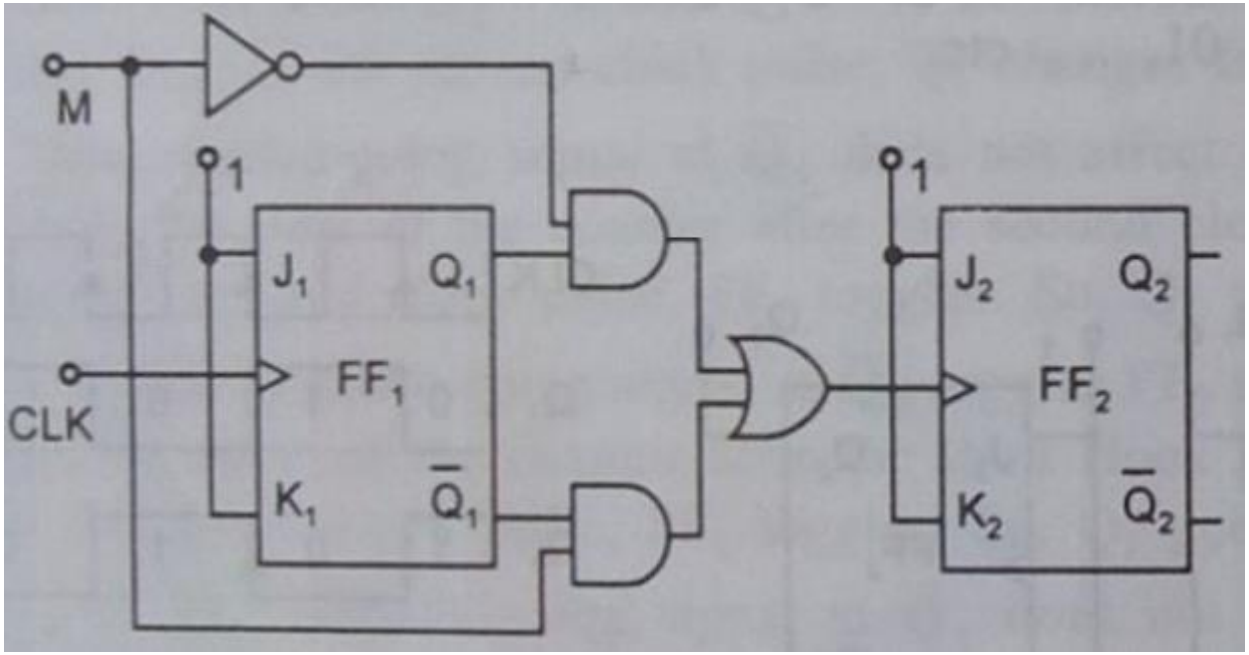



| Clock Pulse | Q2 | Q1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 1 | 0 |
| 3 | 0 | 1 |
| 4 | 0 | 0 |



## Ripple/Asynchronous/Serial Counter

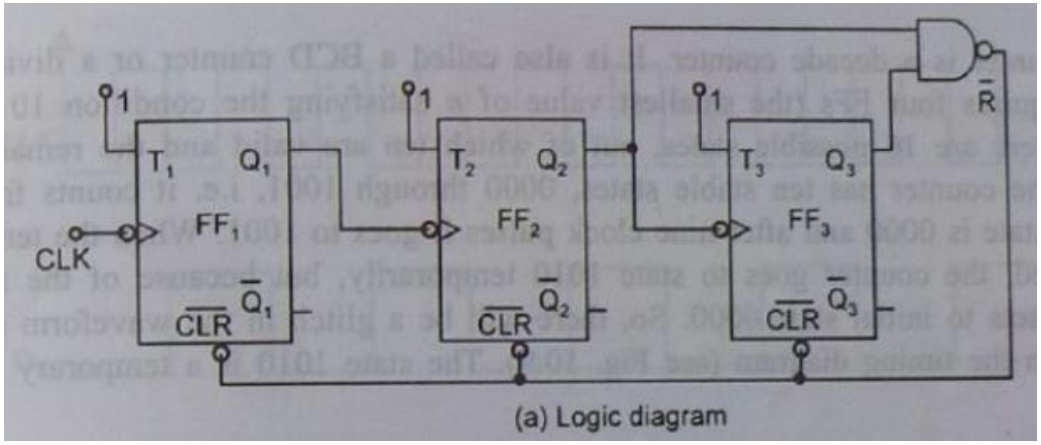
2 Bit Positive Edge Triggered UP/Down Counter



## Ripple/Asynchronous/Serial Counter

Mod 6 Counter Design

| After pulses | State $Q_3$ | $Q_2$ | $Q_1$ | R |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| | ↓ | ↓ | ↓ | |
| | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 1 | 0 |



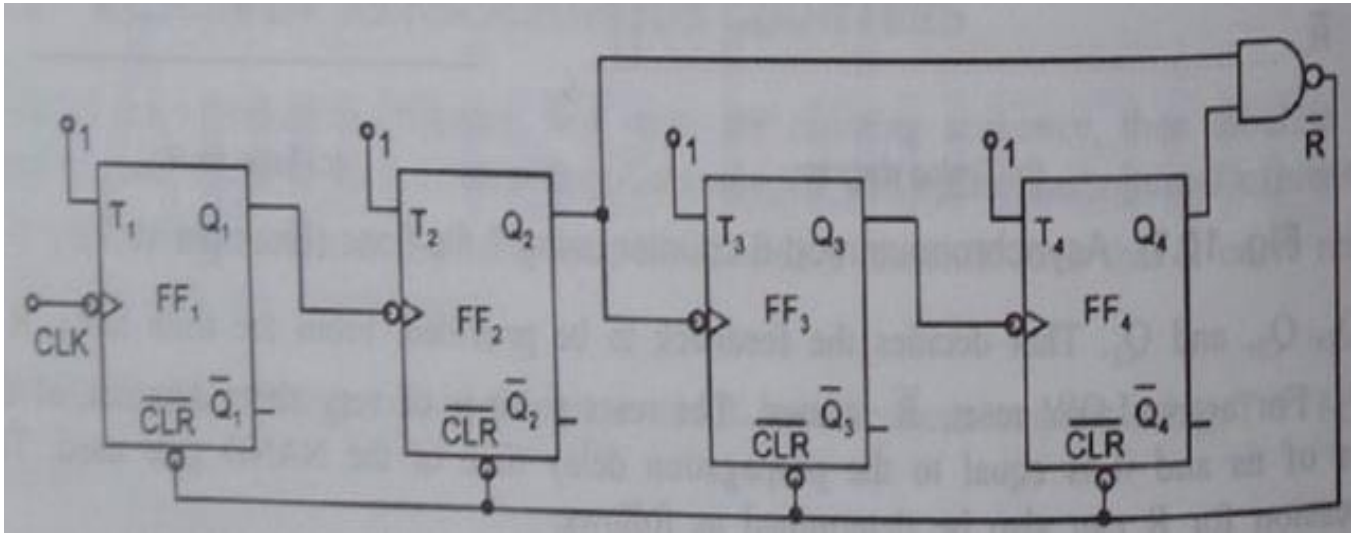
(a) Logic diagram

## Ripple/Asynchronous/Serial Counter

Mod 10 Asynchronous Counter Design

| After pulses | Count $Q_4$ | $Q_3$ | $Q_2$ | $Q_1$ |
|---|---|---|---|---|
| 0 | 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 |

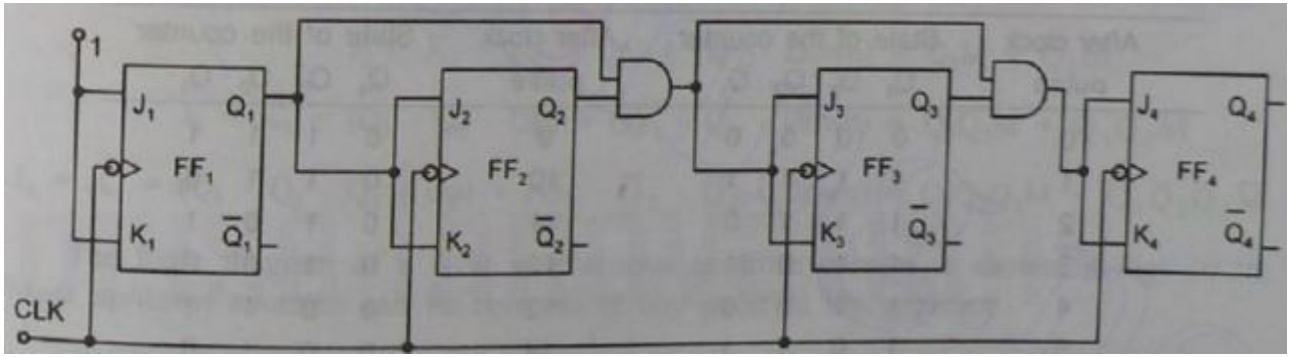
(a) Count table



## Synchronous Counter

4 Bit UP Counter Design

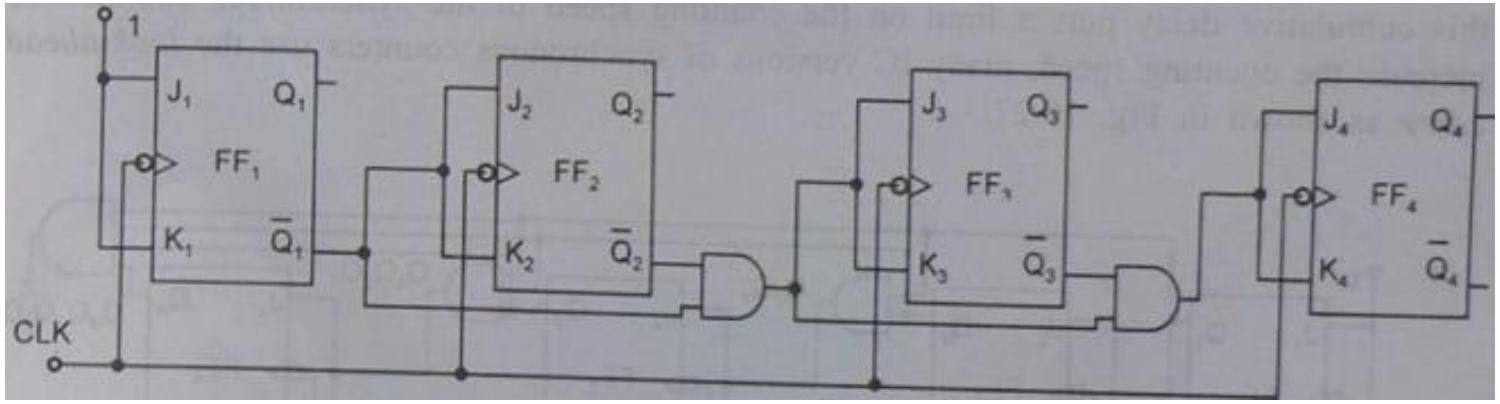
| After clock pusle | State of counter $Q_4$ | $Q_3$ | $Q_2$ | $Q_1$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 |
| 14 | 1 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 |
| 16 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 1 |



## Synchronous Counter

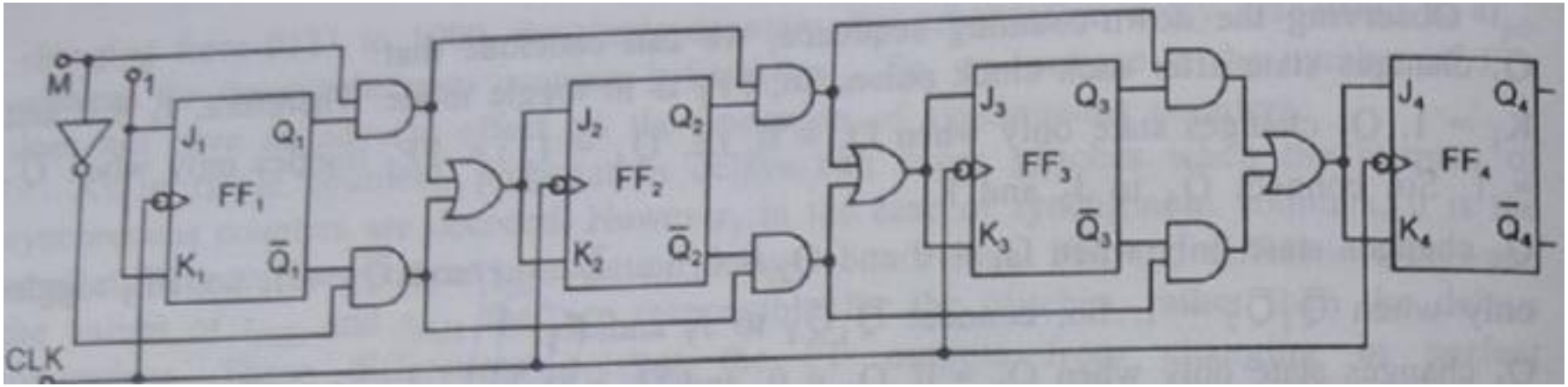
4 Bit Down Counter Design

| After clock pulse | State of the counter $Q_4$ $Q_3$ $Q_2$ $Q_1$ | After clock pulse | State of the counter $Q_4$ $Q_3$ $Q_2$ $Q_1$ |
|---|---|---|---|
| 0 | 0 0 0 0 | 9 | 0 1 1 1 |
| 1 | 1 1 1 1 | 10 | 0 1 1 0 |
| 2 | 1 1 1 0 | 11 | 0 1 0 1 |
| 3 | 1 1 0 1 | 12 | 0 1 0 0 |
| 4 | 1 1 0 0 | 13 | 0 0 1 1 |
| 5 | 1 0 1 1 | 14 | 0 0 1 0 |
| 6 | 1 0 1 0 | 15 | 0 0 0 1 |
| 7 | 1 0 0 1 | 16 | 0 0 0 0 |
| 8 | 1 0 0 0 | 17 | 1 1 1 1 |

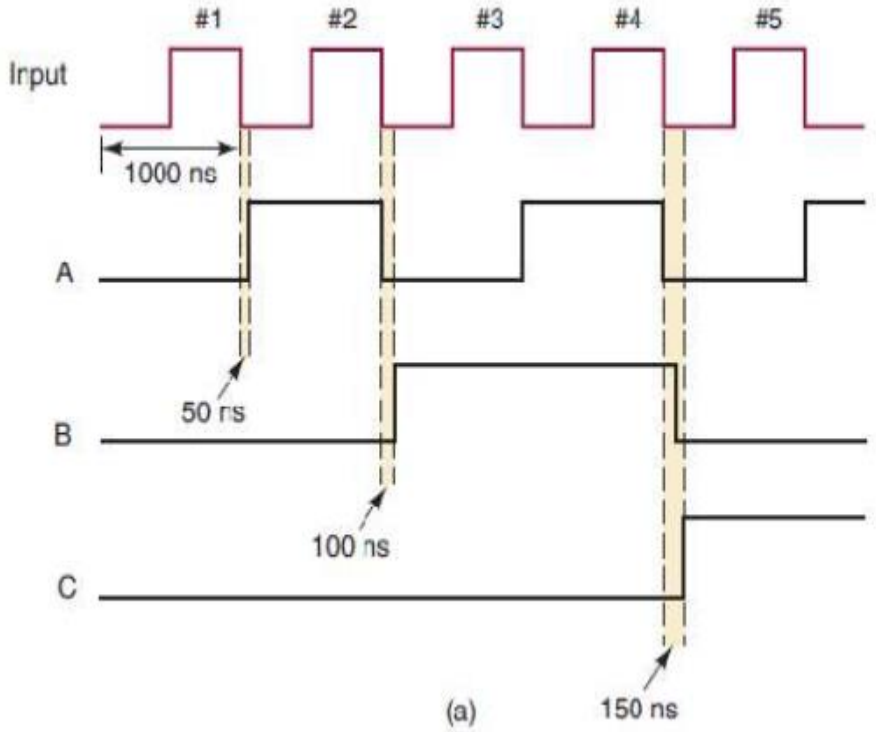


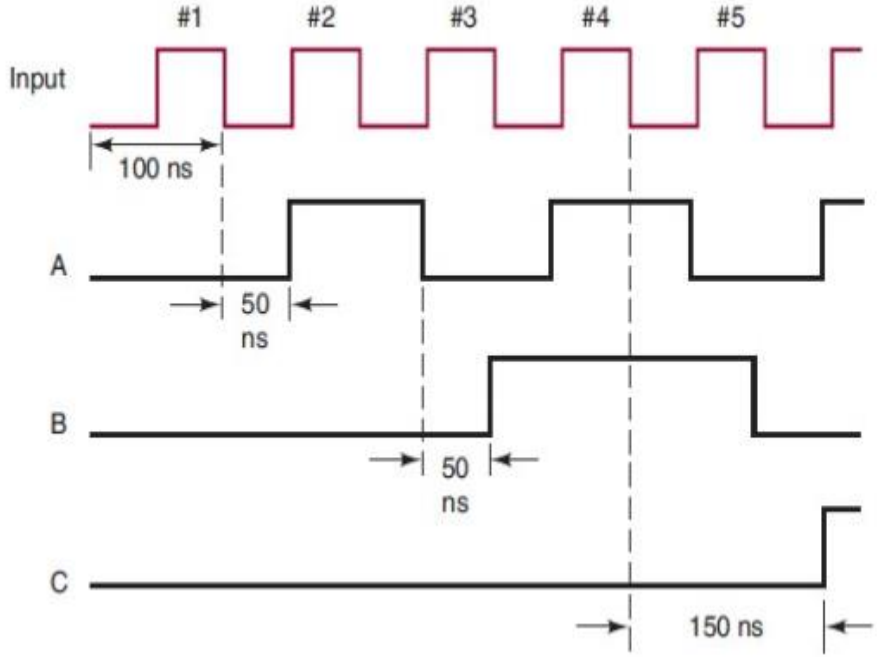
## Ripple/Asynchronous/Serial Counter

4 Bit UP/Down Counter Design



## Ripple/Asynchronous/Serial Counter

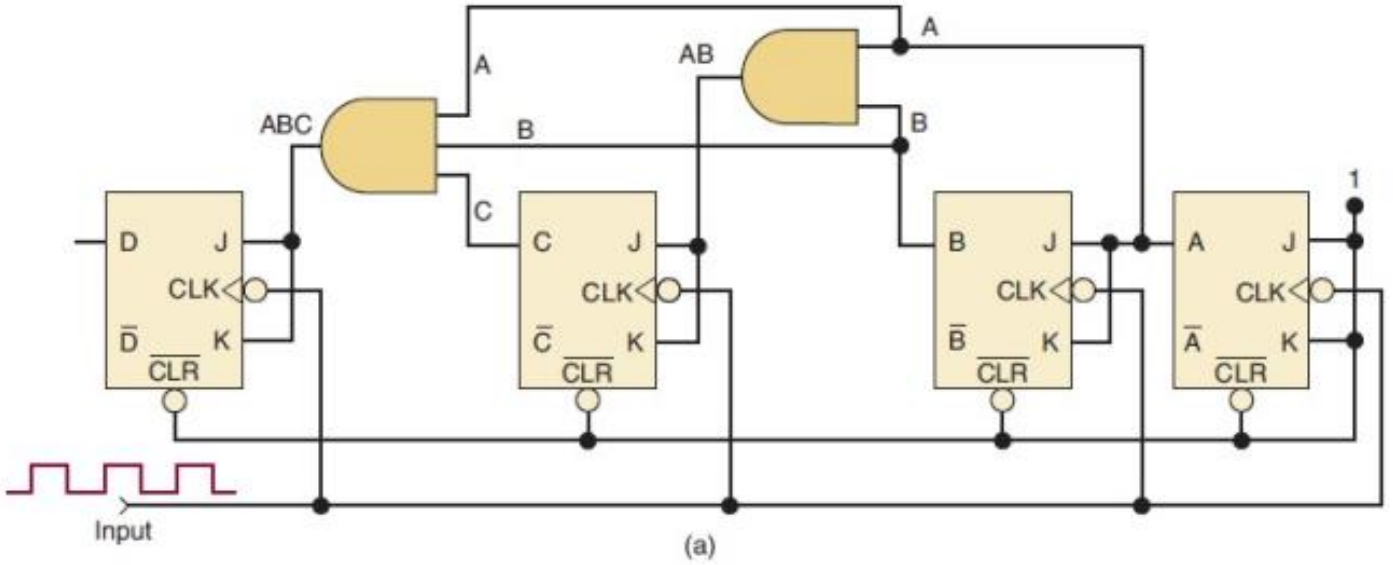



$$T_{\text{clock}} \geq N \times t_{\text{pd}}$$

$$f_{\text{max}} = \frac{1}{N \times t_{\text{pd}}}$$

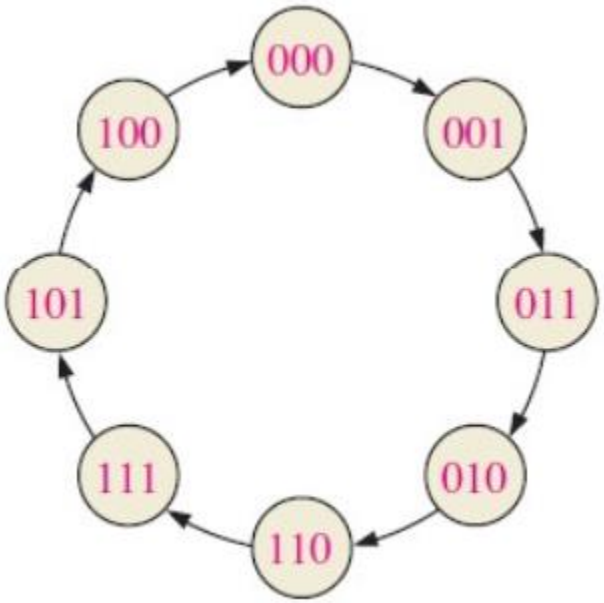
A certain J-K flip-flop has tpd = 12 ns. What is the largest MOD counter that can be constructed from these FFs and still operate up to 10 MHz?




$$\text{total delay} = \text{FF } t_{pd} + \text{AND gate } t_{pd}$$

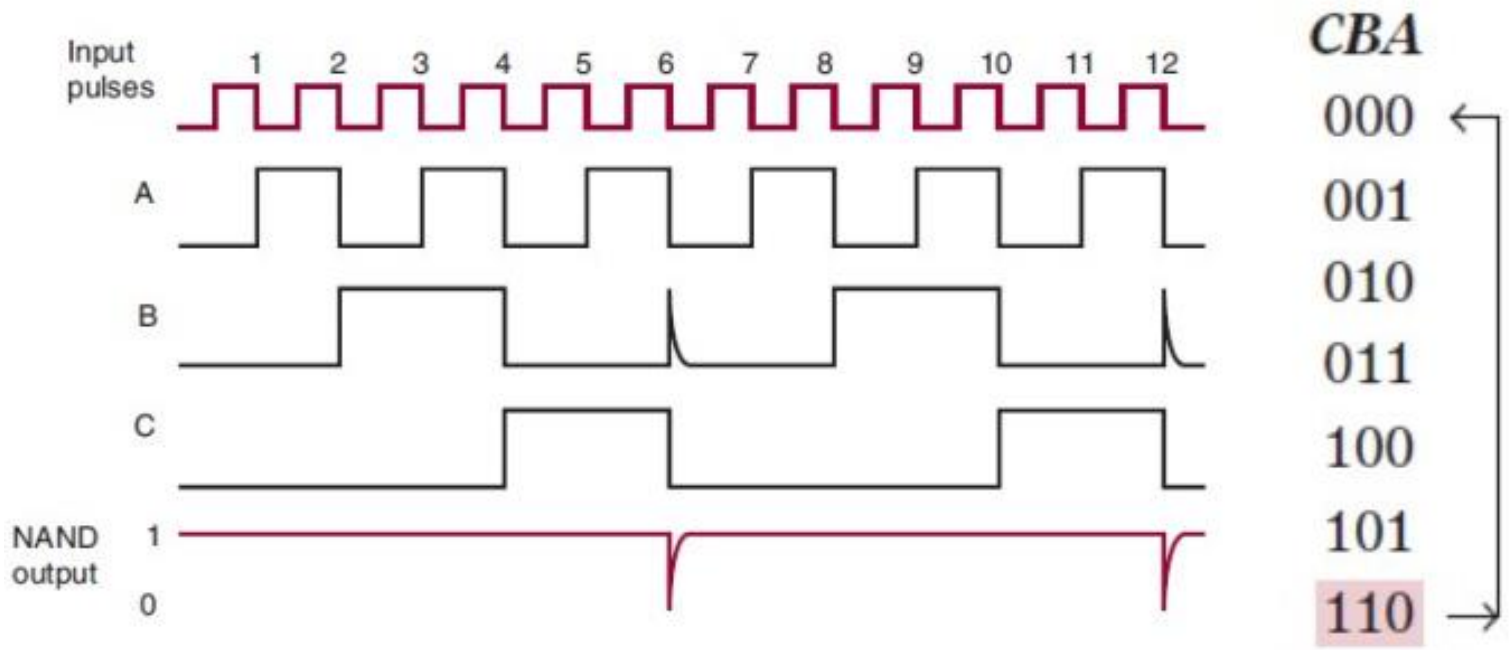
**Determine fmax for the synchronous counter of Figure above if tpd for each FF is 50 ns and tpd for each AND gate is 20 ns. Compare this value with fmax for a MOD-16 ripple counter.**

## State Diagram

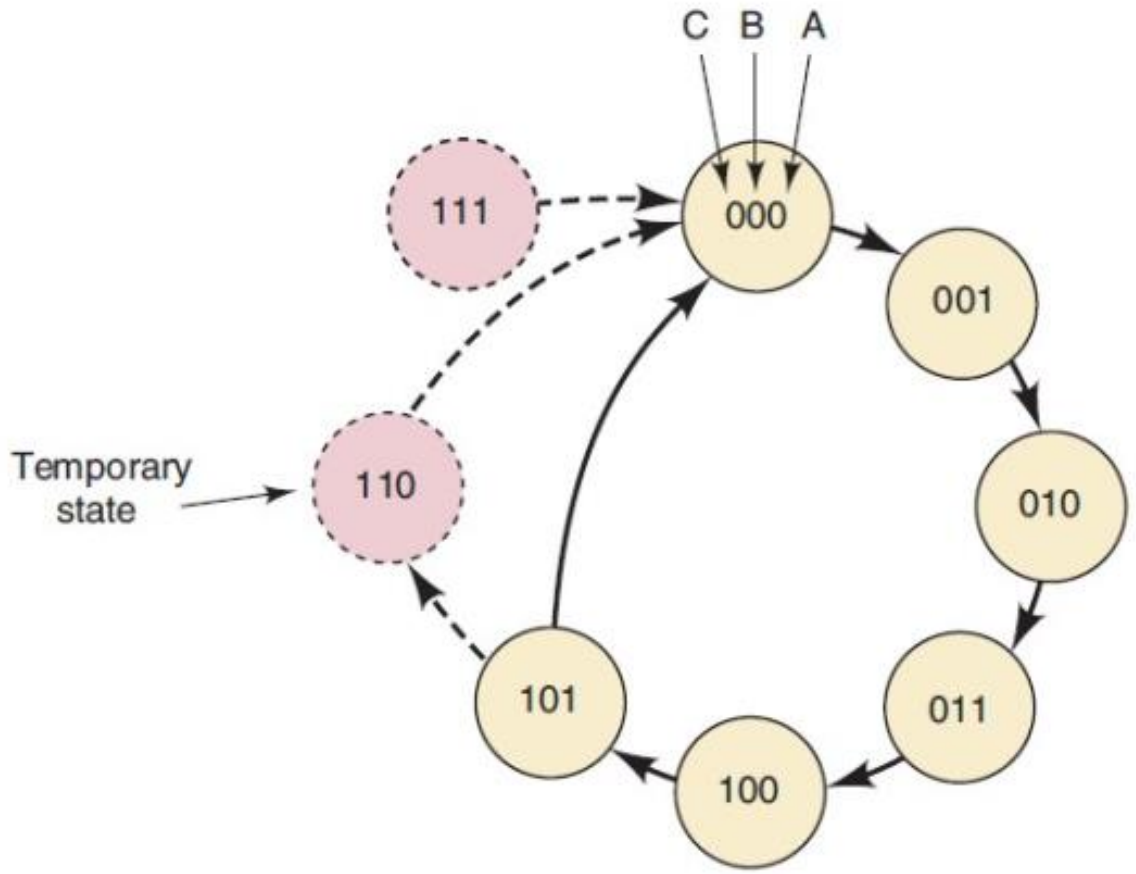



| Present State | | | Next State | | |
|---|---|---|---|---|---|
| $Q_2$ | $Q_1$ | $Q_0$ | $Q_2$ | $Q_1$ | $Q_0$ |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |

## Mod 6 Counter



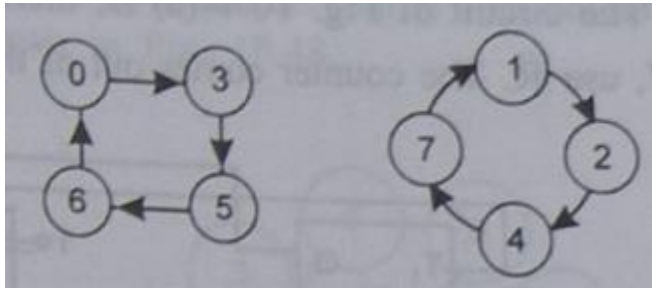
## State Transition Diagram



## Synchronous Counter Design

1. Specify the counter sequence and draw a state diagram.
2. Derive a next-state table from the state diagram.
3. Develop a transition table showing the flip-flop inputs required for each transition. The transition table is always the same for a given type of flip-flop.
4. Transfer the *J* and *K* states from the transition table to Karnaugh maps. There is a Karnaugh map for each input of each flip-flop.
5. Group the Karnaugh map cells to generate and derive the logic expression for each flip-flop input.
6. Implement the expressions with combinational logic, and combine with the flip-flops to create the counter.

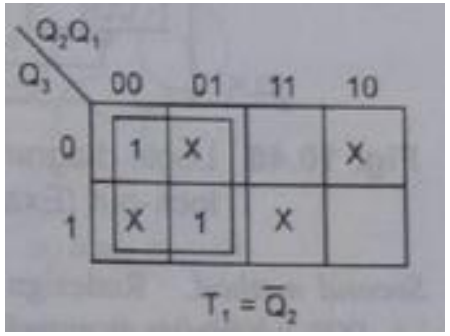
**Design a counter using T Flip-Flops that goes through the states 0-3-5-6-0...**

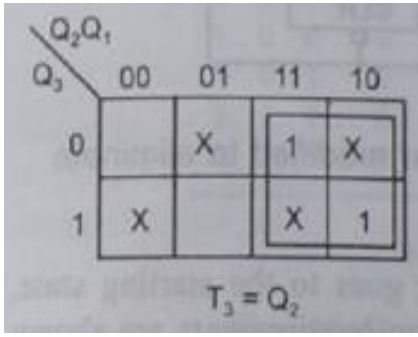


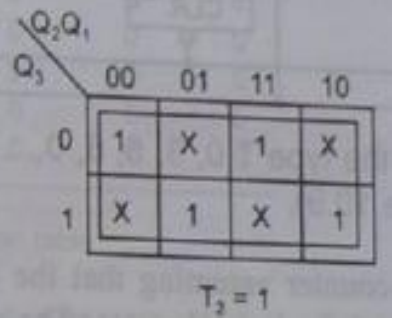

| PS | | | NS | | | Required excitation | | |
|---|---|---|---|---|---|---|---|---|
| $Q_3$ | $Q_2$ | $Q_1$ | $Q_3$ | $Q_2$ | $Q_1$ | $T_3$ | $T_2$ | $T_1$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Valid States – 0,3,5,6

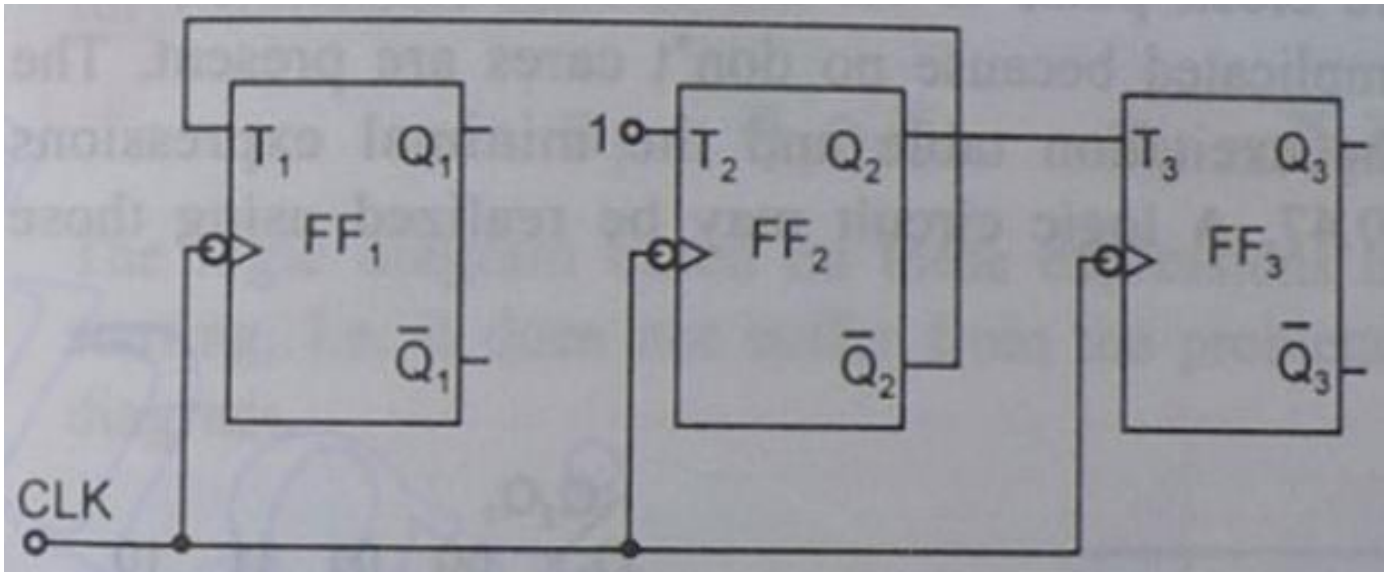
Invalid States – 1,2,4,7



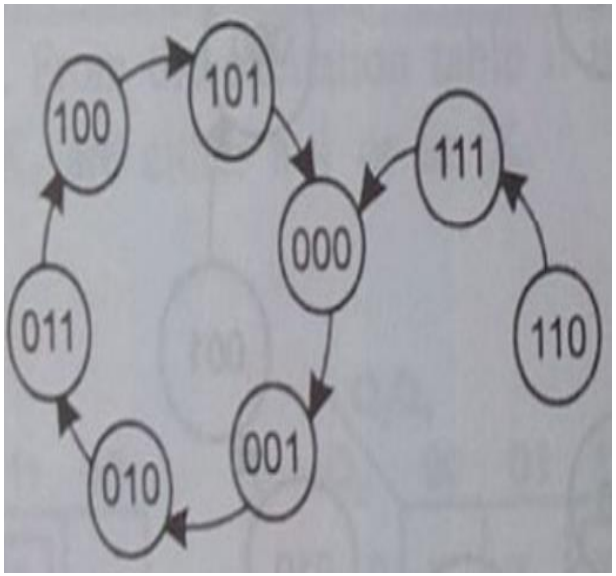



# Digital Logic Circuits

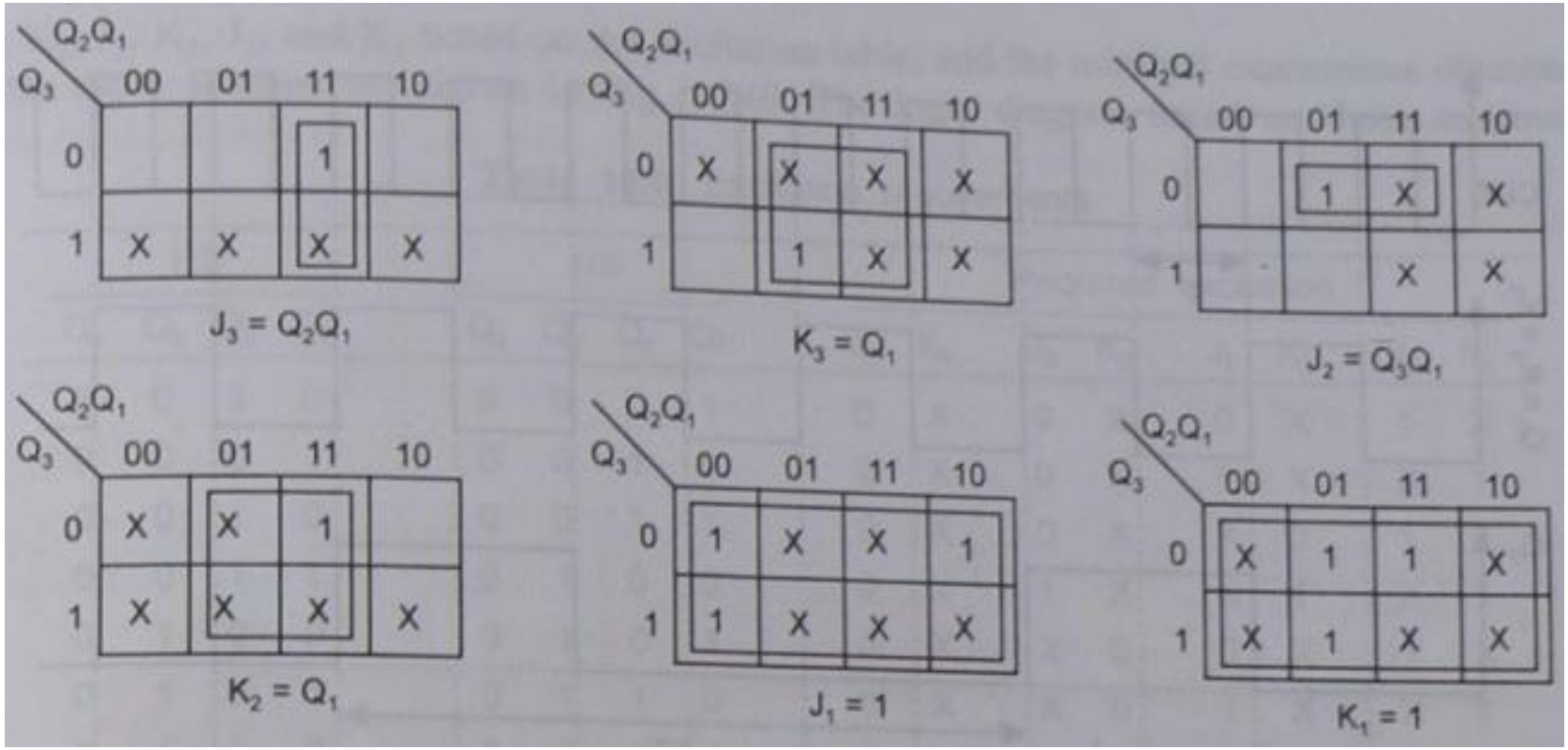


## MOD 6 Counter

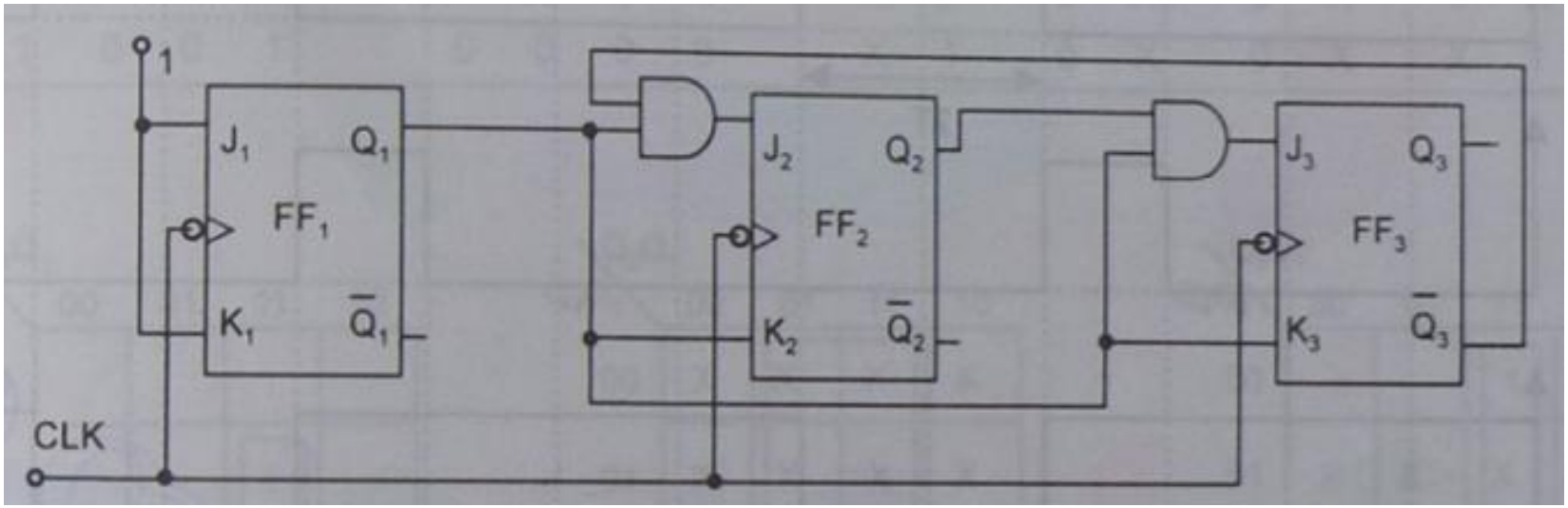



| PS | | | NS | | | Required excitation | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_3$ | $Q_2$ | $Q_1$ | $Q_3$ | $Q_2$ | $Q_1$ | $J_3$ | $K_3$ | $J_2$ | $K_2$ | $J_1$ | $K_1$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | X | 0 | X | 1 | X |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | X | 1 | X | X | 1 |
| 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 |

## MOD 6 Counter

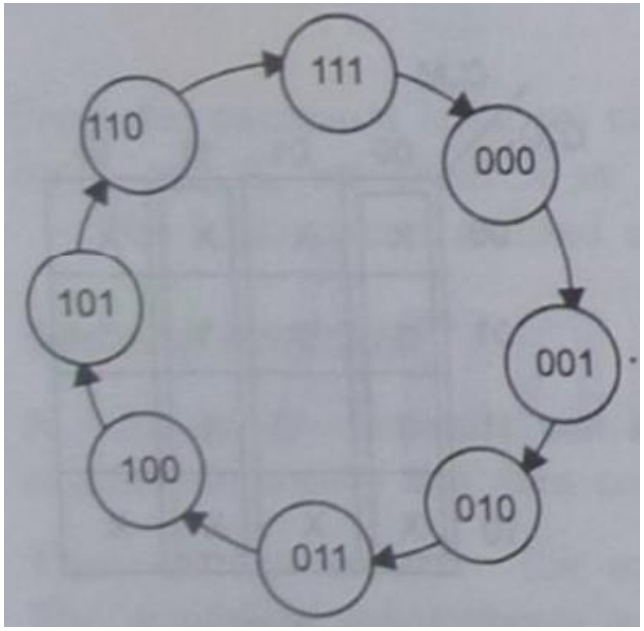


## MOD 6 Counter

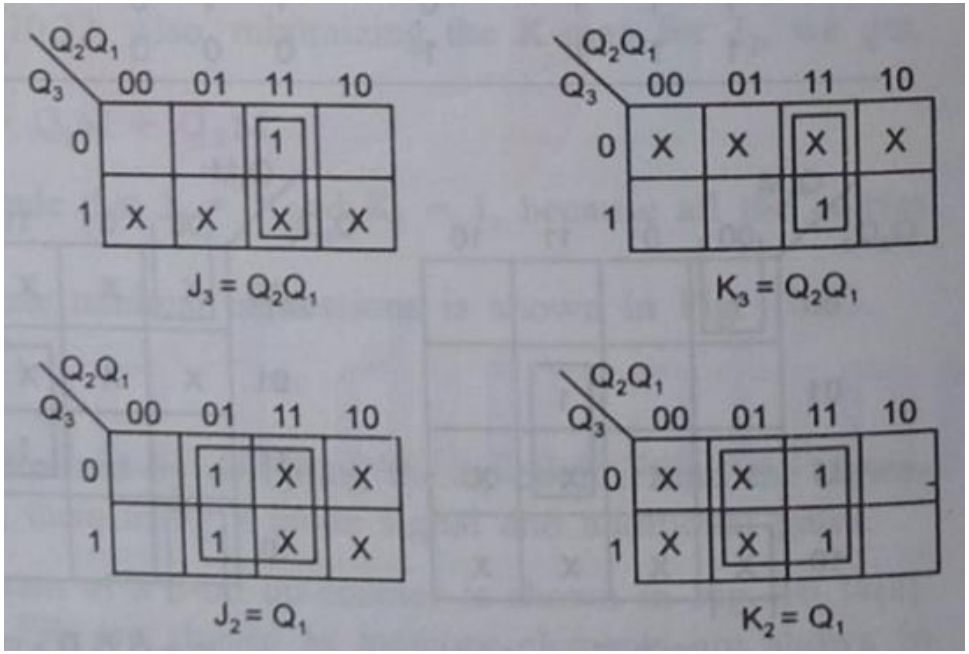


## 3 Bit UP Counter

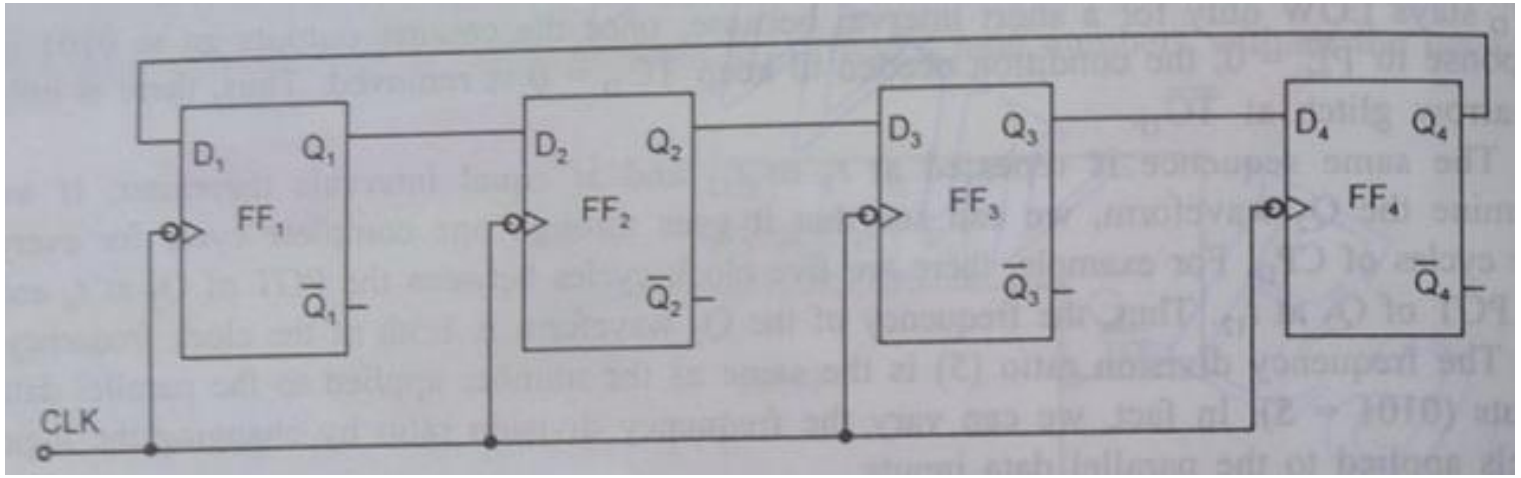
| PS | | | NS | | | Required excitation | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_3$ | $Q_2$ | $Q_1$ | $Q_3$ | $Q_2$ | $Q_1$ | $J_3$ | $K_3$ | $J_2$ | $K_2$ | $J_1$ | $K_1$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | X | 0 | X | 1 | X |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | X | 1 | X | X | 1 |
| 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 | 1 | 1 | 0 | X | 0 | 1 | X | X | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | X | 0 | X | 0 | 1 | X |
| 1 | 1 | 1 | 0 | 0 | 0 | X | 1 | X | 1 | X | 1 |

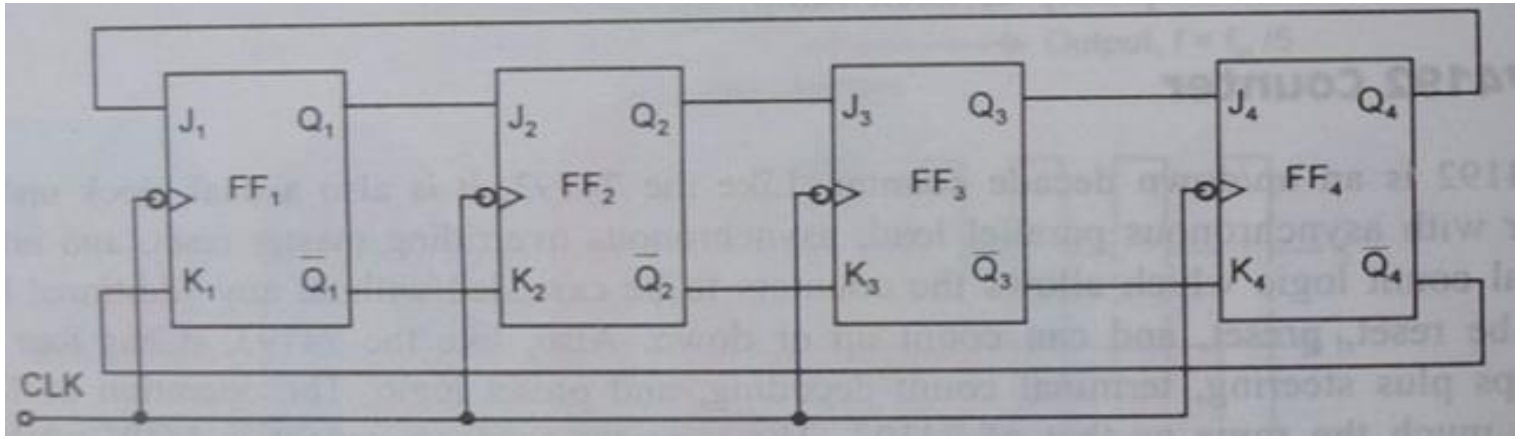


## 3 Bit UP Counter



## Ring Counter

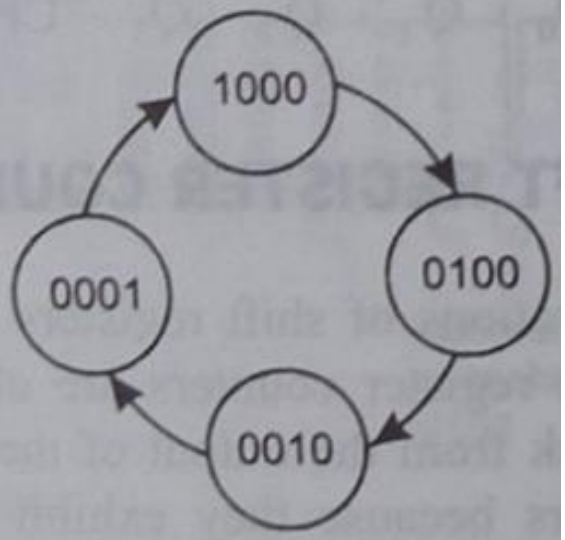


## Ring Counter

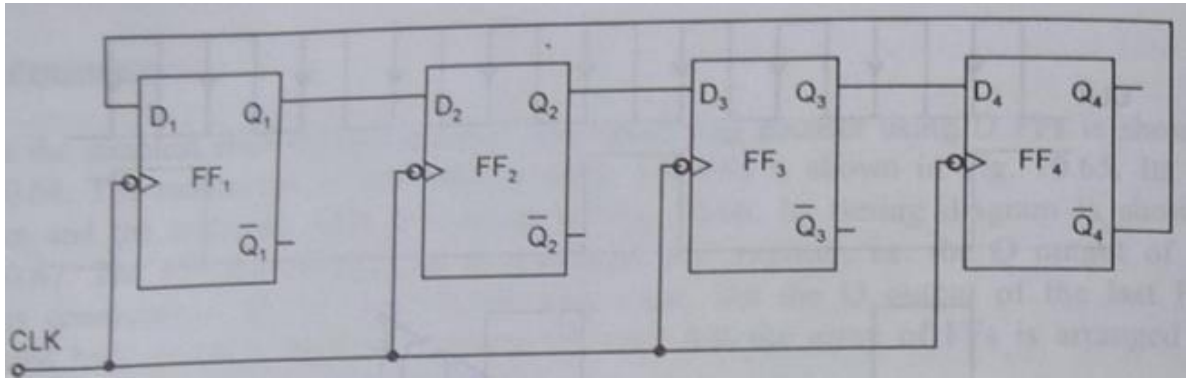
| $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | After clock pulse |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 1 | 3 |
| 1 | 0 | 0 | 0 | 4 |
| 0 | 1 | 0 | 0 | 5 |
| 0 | 0 | 1 | 0 | 6 |
| 0 | 0 | 0 | 1 | 7 |

(a) Sequence table




(b) State diagram

## Johnson Counter or Twisted Ring Counter




| $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | After clock pulse |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 1 | 1 | 0 | 3 |
| 1 | 1 | 1 | 1 | 4 |
| 0 | 1 | 1 | 1 | 5 |
| 0 | 0 | 1 | 1 | 6 |
| 0 | 0 | 0 | 1 | 7 |
| 0 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 0 | 9 |