Unit 8: Register, Counters and Memory Unit

Presented By:

Er. Ganesh Kumal

Contents

- Registers
- Shift Registers
- Superposition of Registers
- Generation of codes using registers
- Ripple
- Synchronous and Johnson Counters
- Design of multiple inputs circuit
- Random access memory (RAM)
- Memory decoding
- Error correction Code
- Output hazards races

Register

- Flip-flop can be used to store single bit binary data. To increase the storage capacity in terms of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a register.
- The n-bit register will consists of a n number of flip-flops and capable of storing an n-bit word.

Shift Register

- The binary data can be moved within the register from one flip-flop to another.
- The registers that allow such data transfers(either left or right) are called as shift registers.

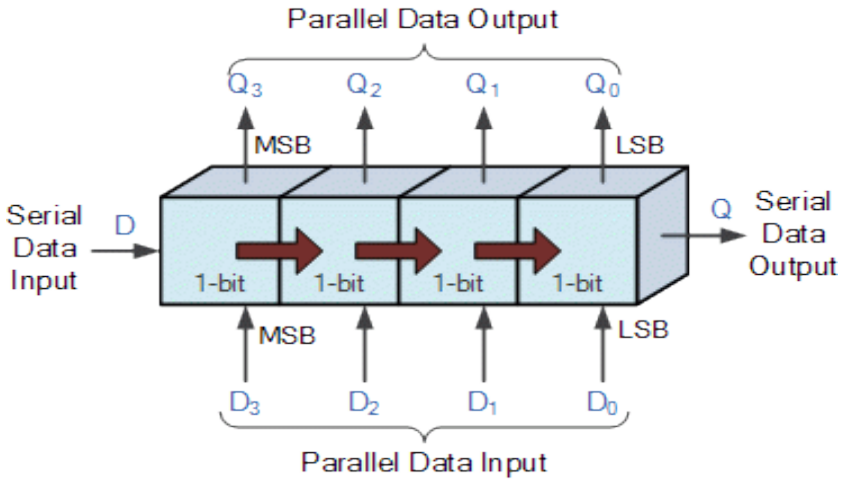
Types:

Shift registers are basically of 4 types. These are:

1. Serial In Serial Out (SISO)
2. Serial In Parallel Out (SIPO)
3. Parallel In Serial Out (PISO)
4. Parallel In Parallel Out (PIPO)

Types

- **Serial In Parallel-out (SIPO)** - the register is loaded with serial data, one bit at a time, with the stored data being available in parallel form.
- **Serial In Serial-out (SISO)** - the data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.
- **Parallel In Serial-out (PISO)** - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.
- **Parallel In parallel-out (PIPO)** - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.



Serial In Parallel Out

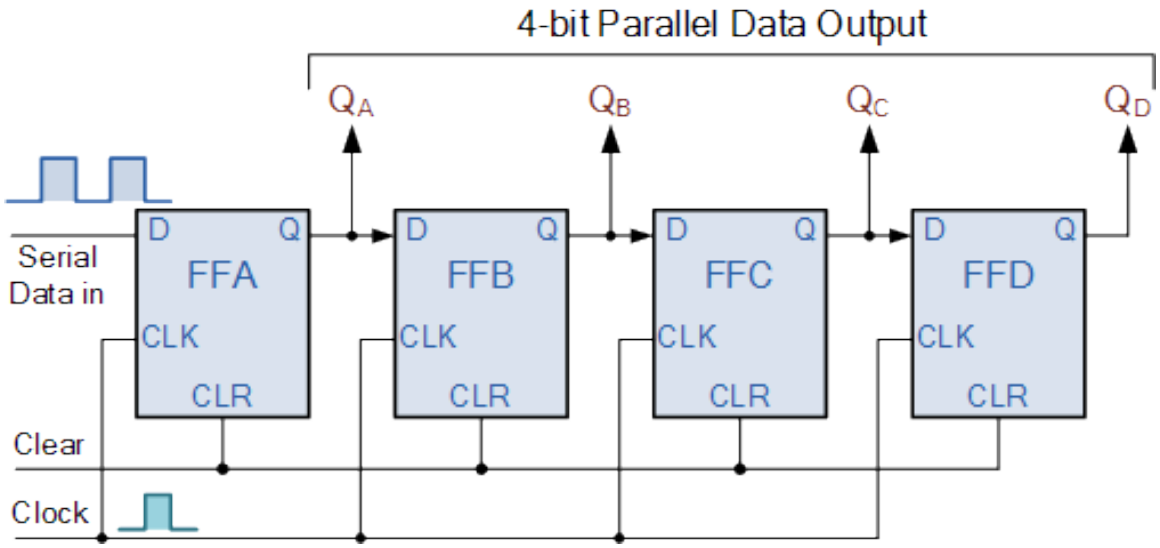


Figure: 4-bit Serial In Parallel Out

- In Serial-in Parallel-out shift register, data is entered serially and taken out in parallel fashion.
- 4 clock cycles are required to load 4-bit data serially.
- At 4^{th} clock pulse, all data bits are out.

Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

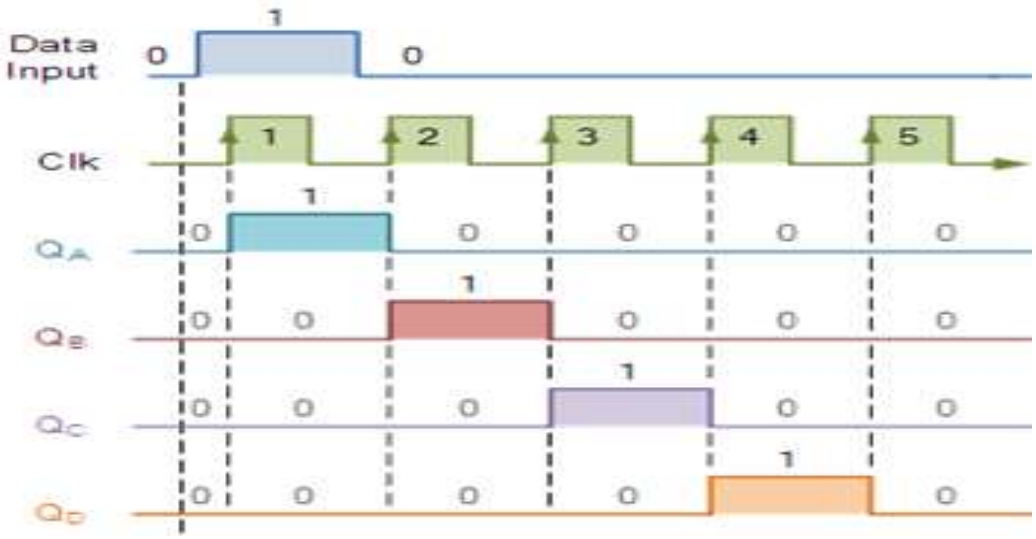
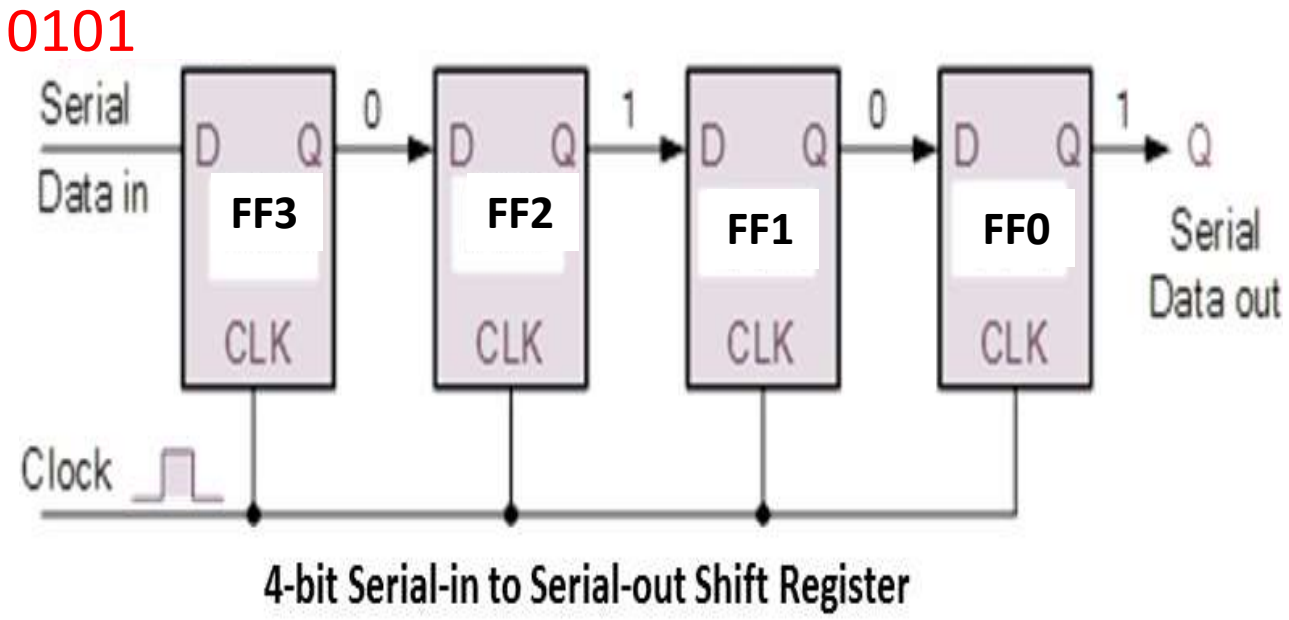


Figure: Truth table and Waveforms of 4-bit SIPO shift register

Serial In Serial Out



4-bit Serial-in to Serial-out Shift Register

- The circuit consists of 4 D flip-flops which are connected in a serial manner.
- All flip-flops are synchronous with each other since the same clock signal is applied to each flip-flop.
- If the number 0101 is going to enter, LSB should be applied first at FF3.
- So 4 clock pulses is required to enter the 4-bit data (0101) and LSB is out at the same time 4^{th} clock pulse. So other 3 clock pulses are required to out the other bits serially.

Truth table of 4-bit serial in serial out shift register

CLK	Q3	Q2	Q1	Q0
Initially	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	0	1	0
4	0	1	0	1
5	0	0	1	0
6	0	0	0	1
7	0	0	0	0

Serial In Complete

Serial out Complete

4-Bit Parallel In Parallel Out

This type of register also acts as a temporary storage device or as a time delay device similar to the SISO configuration above. The data is presented in a parallel format to the parallel input pins PA to PD and then transferred together directly to their respective output pins QA to QD by the same clock pulse.

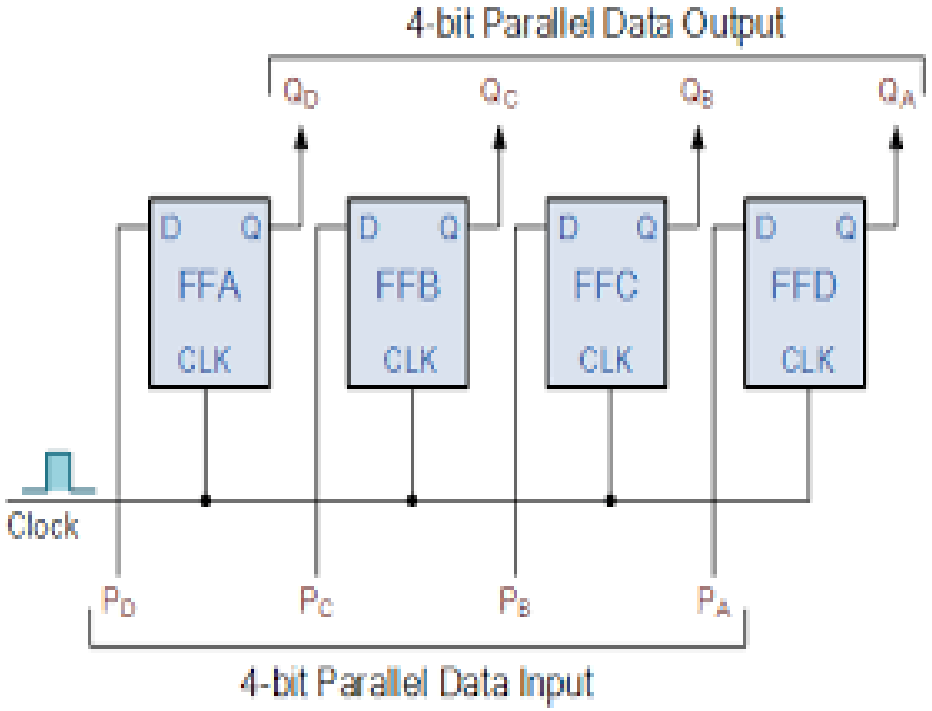


Figure: 4-bit parallel in serial out shift register

4-Bit Parallel In Serial Out

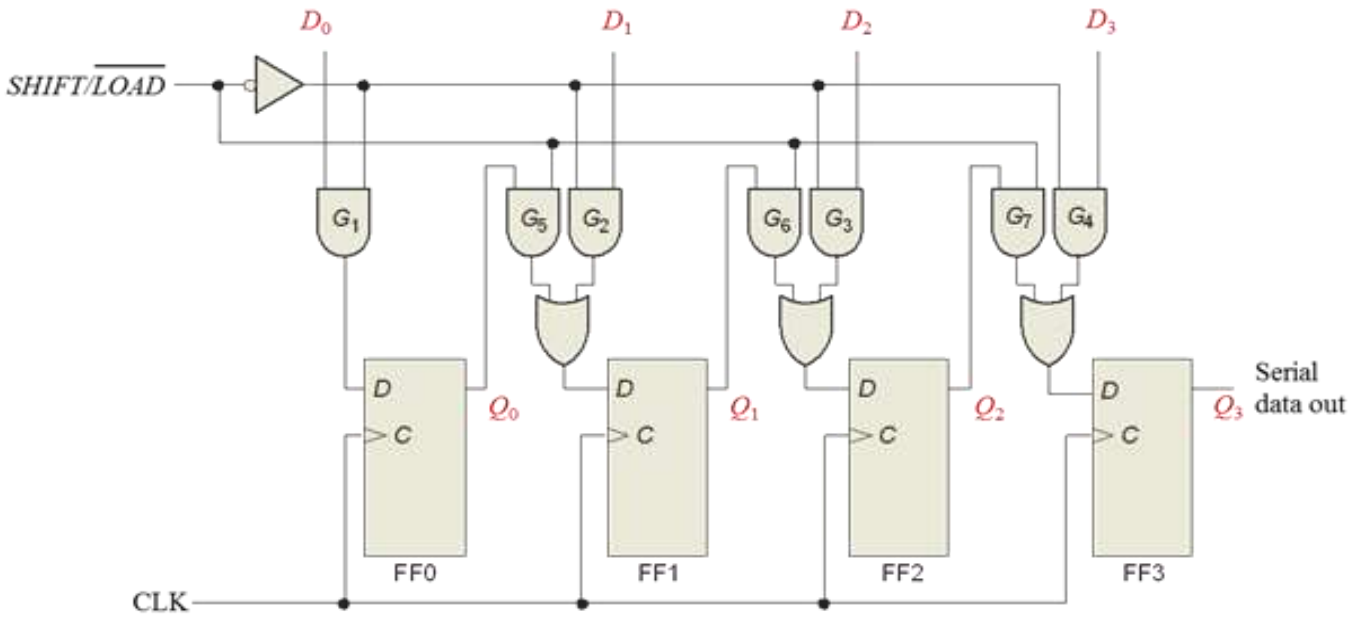


Figure: 4-bit parallel in serial out shift register

LOAD Mode:

When the SHIFT/ $\overline{LOAD}$ =0; the AND gate 1, 2, 3 and 4 become active. They will pass data D0, D1, D2 and D3 bits to the corresponding flip-flops.

- When SHIFT/ $\overline{\text{LOAD}}$ =1; Gate 1, 2, 3 and 4 are inactive. Hence the parallel loading of the data become impossible.
- But the Gate 5, 6 and 7 are active. Therefore shifting of data from left to right is possible by application of clock.
- First clock pulse is used for loading parallel data and remaining 3 clock pulses are required for shifting out the data serially from Q3.

CLK	Q0	Q1	Q2	Q3	Out from Q3
Initially	0	0	0	0	
1	1	0	1	0	0 (Parallel in complete)
2	0	1	0	1	1
3	0	0	1	0	0
4	0	0	0	1	1 (serial-out complete)

Figure: Truth table of 4-bit parallel in serial out shift register

Asynchronous and Synchronous Counter

Counter

- Number of flip-flops can be connected together to perform counting operation: such a group of flip-flops is a counter.
- Counters are classified into two broad categories according to the way they are clocked:
  i. Asynchronous (ripple) counter
  ii. Synchronous counter

Asynchronous Counter

In asynchronous counters, commonly called 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 hence the state of all flip-flops do not change simultaneously.

Synchronous Counter

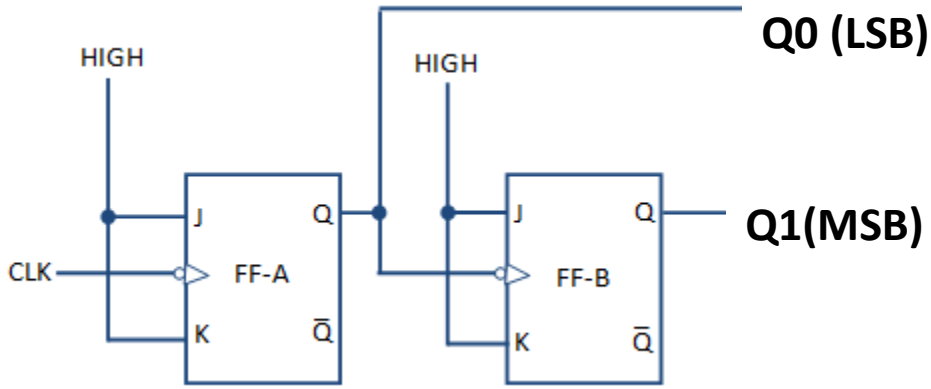
In a synchronous counter, the clock input is connected to all of the flip-flops. So they are clocked simultaneously and state changes by flip-flops occur at the same time.

Binary (Ripple) Counter

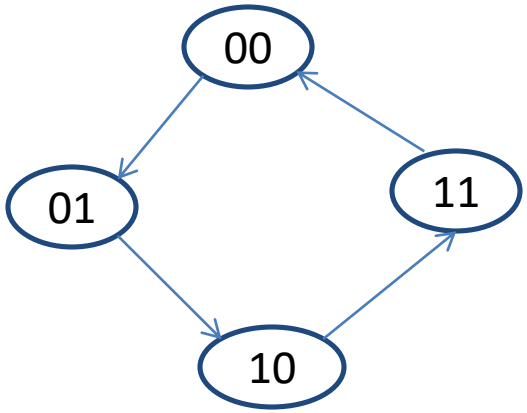
Binary (ripple) counter

- An asynchronous counter is one in which the flip-flop within the counter do not change states at exactly the same time because they do not have a common clock pulse.
- Ripple counter consists of series of flip-flops (T or J-K type), with the output of each flip-flop connected to the CP input of the next higher-order flip-flop.
  - 2 Bit asynchronous binary counter
  - 3 Bit asynchronous binary counter
  - 4 Bit asynchronous binary counter

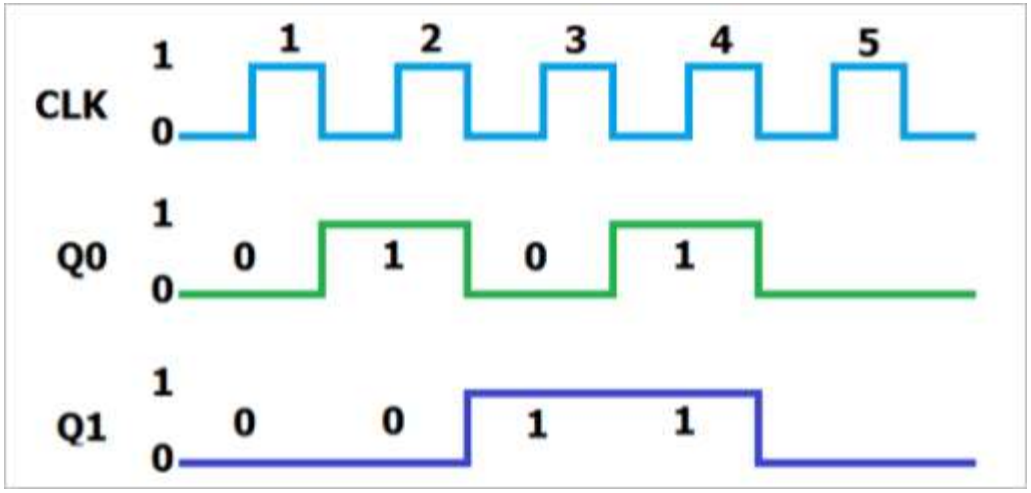
2-Bit Ripple Counter



(b) Circuit diagram



(a) State diagram



(d) waveform

CLK	Q1	Q0	
Initially	0	0	CLR=0
↓	0	1	CLR=1
↓	1	0	CLR=1
↓	1	1	CLR=1

(c) Truth table

Figure: 2-Bit ripple counter

2-Bit Ripple Counter

- There are 2 flip-flops (T or J-K) connected in series.
- All the flip-flops inputs are connected to logic high input.
- External clock input (CP) is connected to first flip-flop and Q output of first flip-flop is connected to next flip-flop.

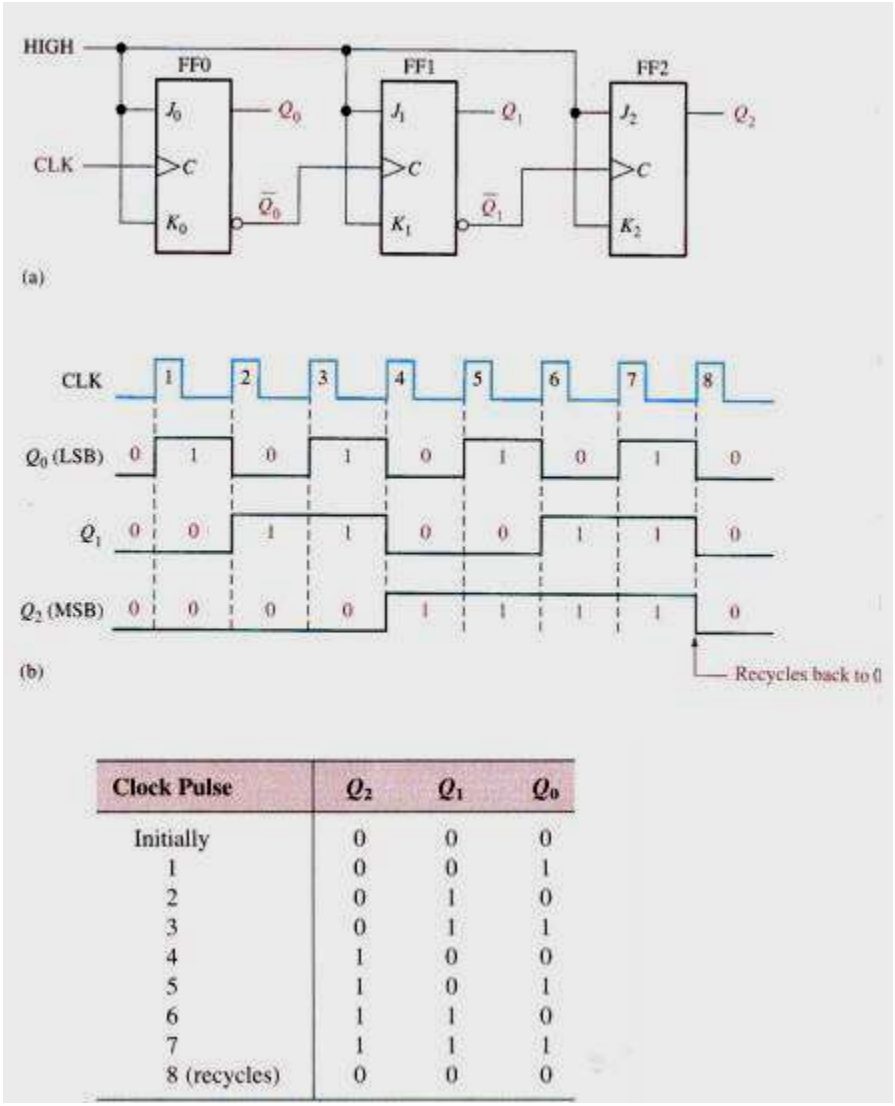
Operation:

- Initial state of each flip-flop in the absence of clock signal is $Q_1Q_0 = 00$; by inserting CLR=0. This is incremented by 1 at every clock pulse and reached to maximum, $Q_1Q_0=11$, and repeats when further clock pulse is applied.
- Q_0 toggle at every negative edge of clock pulse.
- Q_1 toggle for every Q_0 that goes from 1 to 0.

3-Bit Ripple Counter

- There are three flip-flops (T or J-K at toggle mode) connected in series.
- All the flip-flops inputs are connected to logic high.
- External clock input (CLK) is connected to first flip-flop and Q output of first flip-flop is connected to next flip-flop and so on (Positive Edge Triggering) and Q output is connected to next flip-flop and so on (Negative Edge Triggering)

3-Bit Ripple Up Counter(Positive Edge Triggering)



Clock Pulse	Q_2	Q_1	Q_0
Initially	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 (recycles)	0	0	0

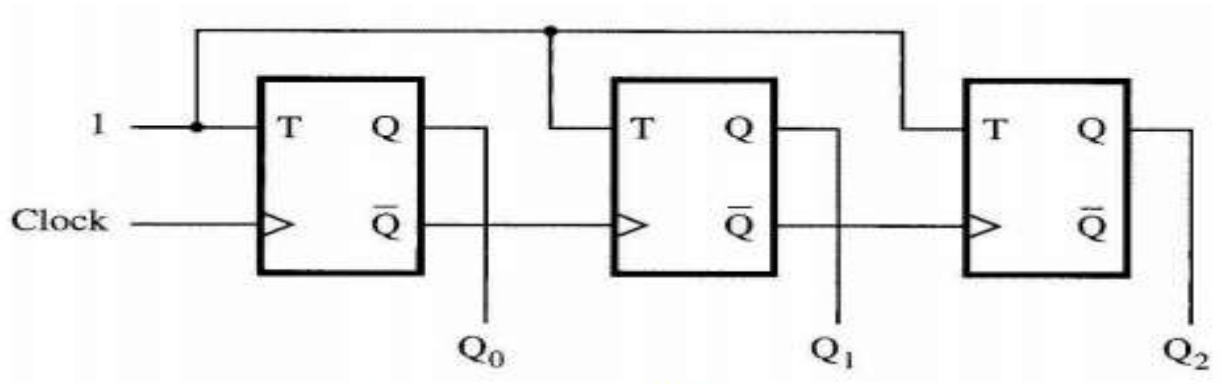
Operation:

1. The initial state of each flip-flop in the absence of clock signal is $Q_2Q_1Q_0$=000.
2. This is incremented by 1 at every clock pulse and reached to maximum(111) at 7^{th} clock pulse and repeats when further clock pulse is applied.
3. Q_0 toggle every positive edge of clock pulse.
4. Q_1 toggle for every Q_0 that goes from 1 to 0.
5. Q_2 toggle when Q_0 and Q_1 (both) goes from 1 to 0.

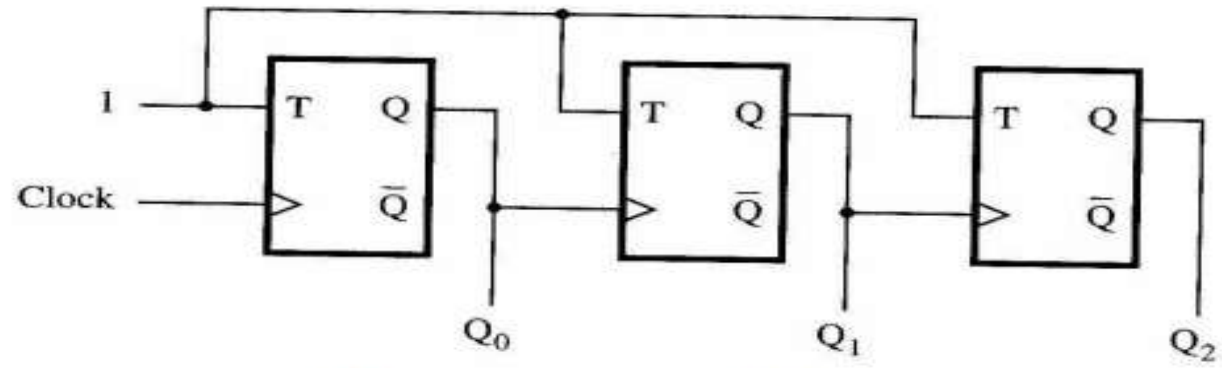
3-Bit Down Counter

For up counter:
+ve edge (Q0 given to CP input of next flip-flop)
-ve edge (Q0 given to CP input of next flip-flop)

For down counter
+edge (Q0 given to CP input of next flip-flop)
-edge ($\overline{Q0}$ given to CP input of next flip-flop)



3-bit up counter with T flip-flops



3-bit down counter with T flip-flops

3-Bit Up/Down Counter

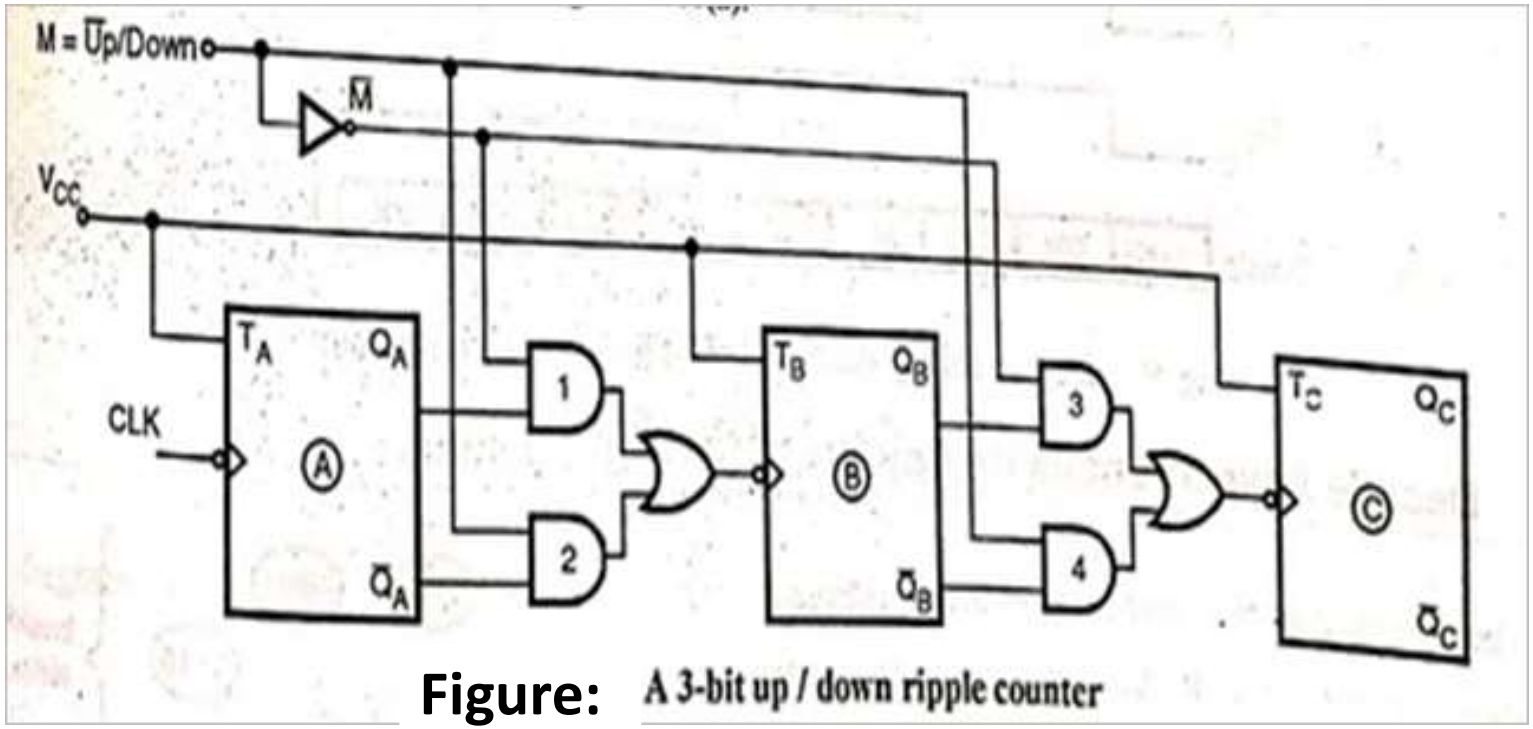


Figure: A 3-bit up / down ripple counter

Truth table of 3-bit up/down counter

COUNT-UP Mode (M=0)				*COUNT-DOWN Mode* (M=1)			
States	Q_C	Q_B	Q_A	*States*	Q_C	Q_B	Q_A
0	0	0	0	7	1	1	1
1	0	0	1	6	1	1	0
2	0	1	0	5	1	0	1
3	0	1	1	4	1	0	0
4	1	0	0	3	0	1	1
5	1	0	1	2	0	1	0
6	1	1	0	1	0	0	1
7	1	1	1	0	0	0	0

4-Bit Ripple Up Counter

- It works exactly the same way as a 2-bit or 3 bit asynchronous binary counter mentioned above, except it has 16 states due to the fourth flip-flop
- J-K inputs of all flip-flops are connected together and given to high input
- For negative edge triggering up counter, Q output is given to CLK input of next flip-flop.
- Initially all flip-flops are reset condition.
- When first clock pulse is given, FF_0 is set and other flip-flops remain previous state(count 0001)
- When second clk pulse is given, FF_1 is set and FF_0 is reset by toggling. (count 2)
- Similarly, at each clock pulse counting is increased by 1(binary) and at 15^{th} clock pulse all flip-flops are in set condition (i.e. 1111=15).
- At 16^{th} clock pulse all flip-flops are reset and recycles.

4-Bit Counter (Negative Edge Triggering)

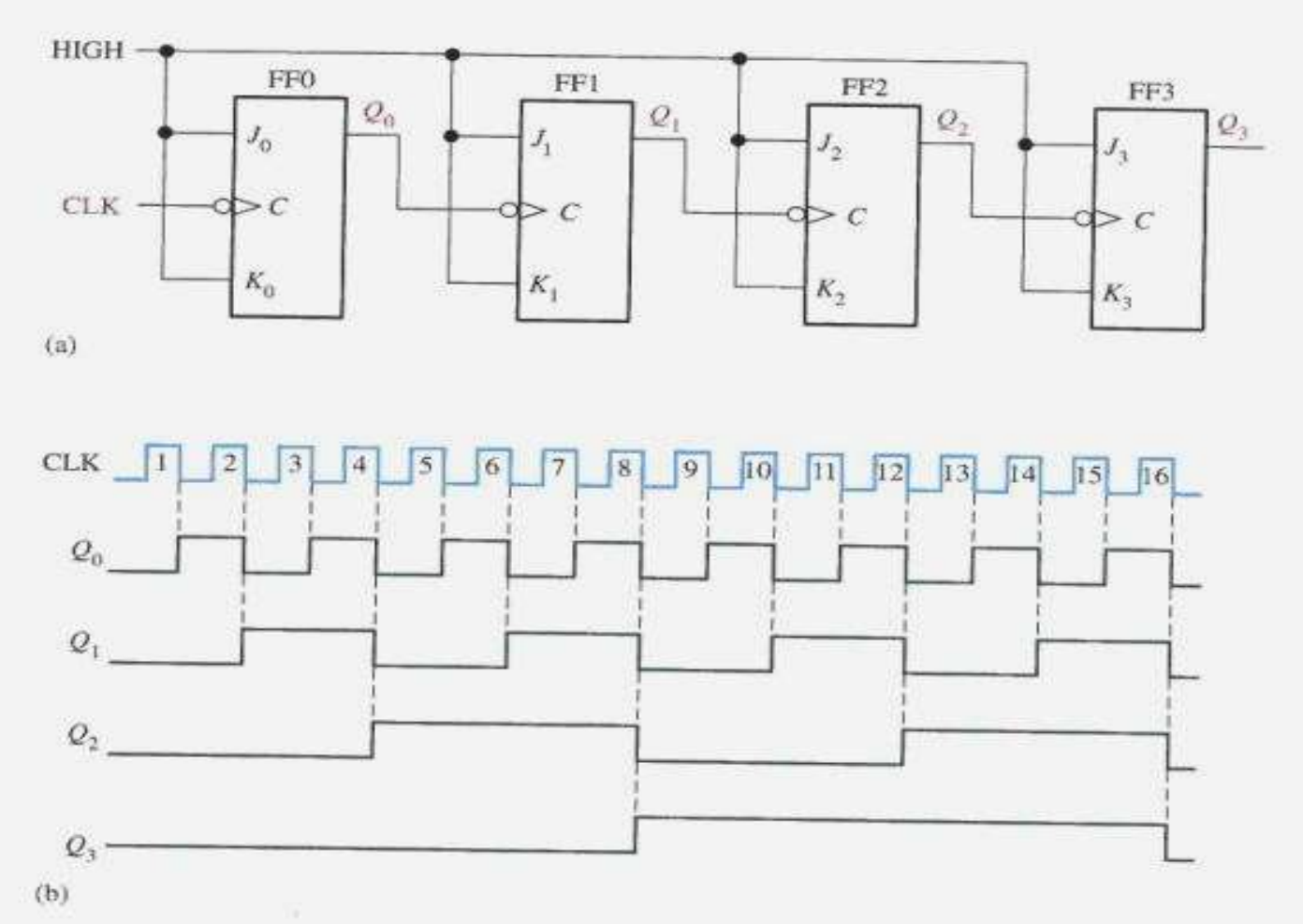


Figure: Circuit diagram and timing waveform

Asynchronous BCD (Decade) Counter

- The binary counters previously introduced have 2^n states. But counters with states less than this number are also possible. They are designed to have the number of states in their sequences, which are called truncated sequences. These sequences are achieved by forcing the counter to recycle before going through all of its normal states.
- A counter with ten states in its sequence is called a decade(Modulus 10) counter.
- Once the counter counts to ten (1010), all the flip-flops are being cleared. Notice that only Q1 and Q3 are used to decode the count of ten. This is called partial decoding, as none of the other states (zero to nine) have both Q1 and Q3 HIGH at the same time.
- Such a counter must have at least four flip-flops to represent each binary bit.

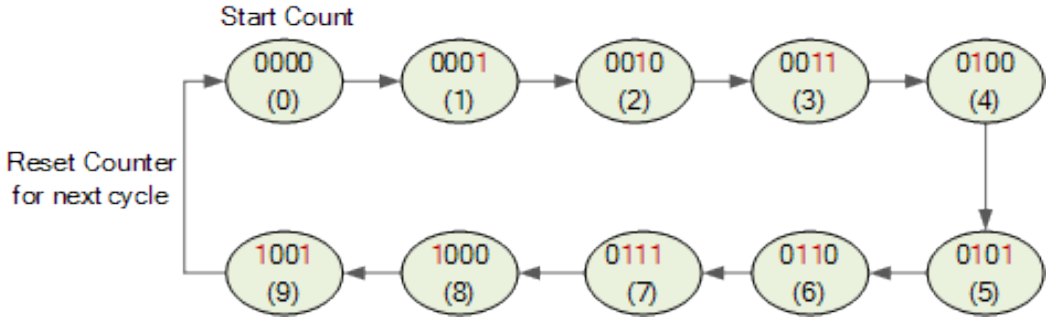
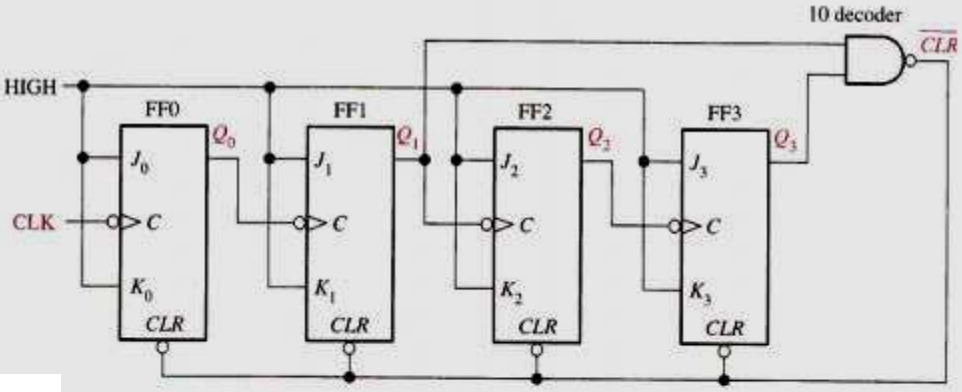


Figure: State diagram

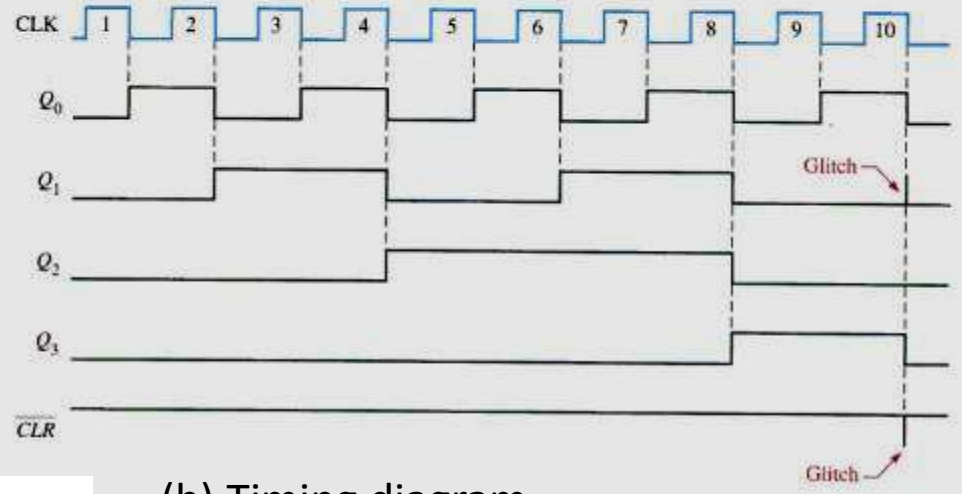
Asynchronous BCD(Decade) Counter

State	Q3	Q2	Q1	Q0
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(RESET)	1	0	1	0

(c) Truth Table



(a) Circuit diagram



(b) Timing diagram

Figure: Decade Counter

Modulus (5, 7, 11) Counter

- 2-bit counter is called MOD-4 counter.
- 3-bit counter is called MOD-8 counter.
- 4-bit counter is called MOD-16 counter.
- Modulus means states.

Example: Mod-5 Counter

- Total states= 5 (000, 001, 010, 011, 100) and reset all flip-flops at 101(5).

MOD-5 Counter

- Total states = 5(0 - 4) and rest at 101 (5).
- Maximum count = 5-1=4 (0 - 4)= 5 states

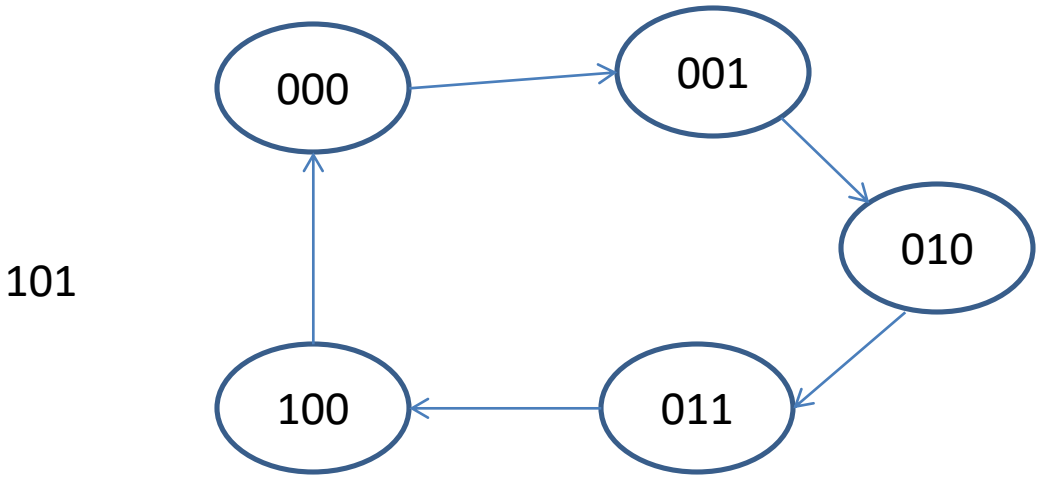


Figure: state diagram

States	Q3	Q2	Q1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5(reset)	1	0	1

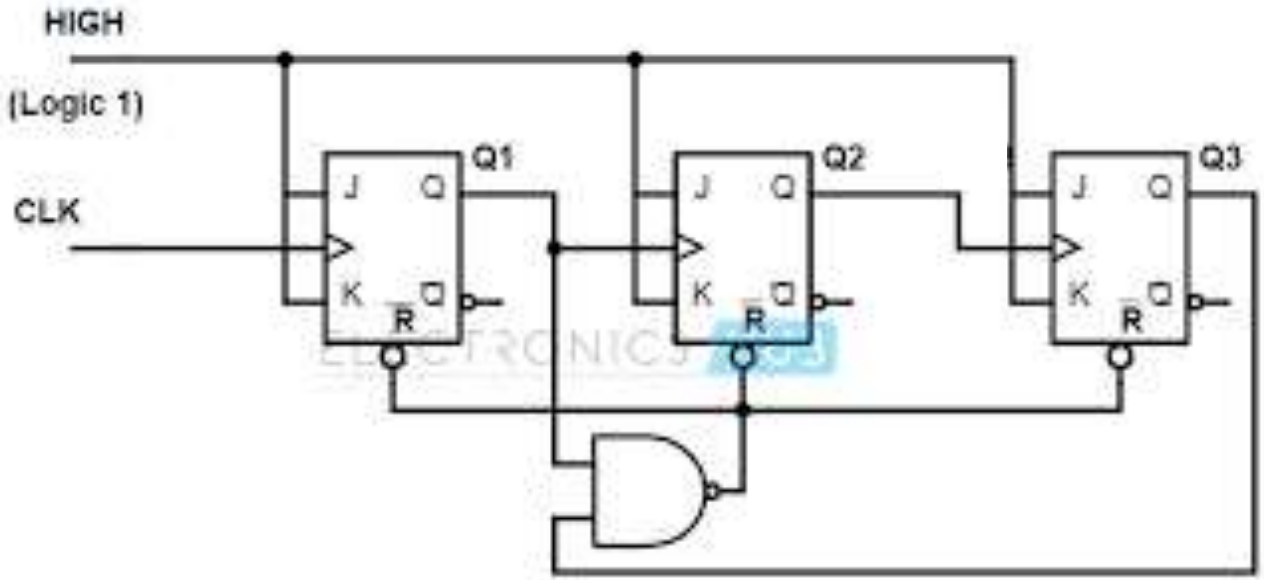
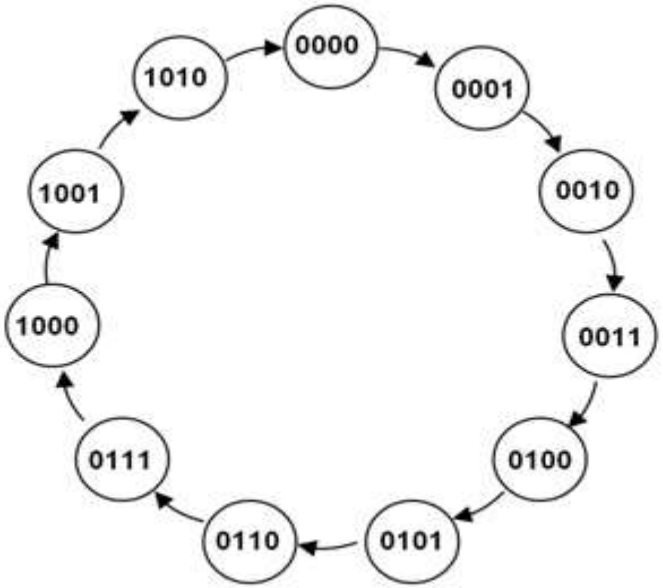


Figure: Mod-5 Counter

- The presence of NAND gate will alter the sequence of counter.
- The NAND output is connected to the RESET inputs of each flip-flop.
- As long as the NAND output HIGH; it will have no effect on the counter.
- When NAND output is LOW; it will reset all the flip-flops, so counter immediately goes to 000 state.

MOD-11 Counter



(a) State diagram

Clock	Q3	Q2	Q1	Q0
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(Reset)	1	0	1	1

(b) Truth Table

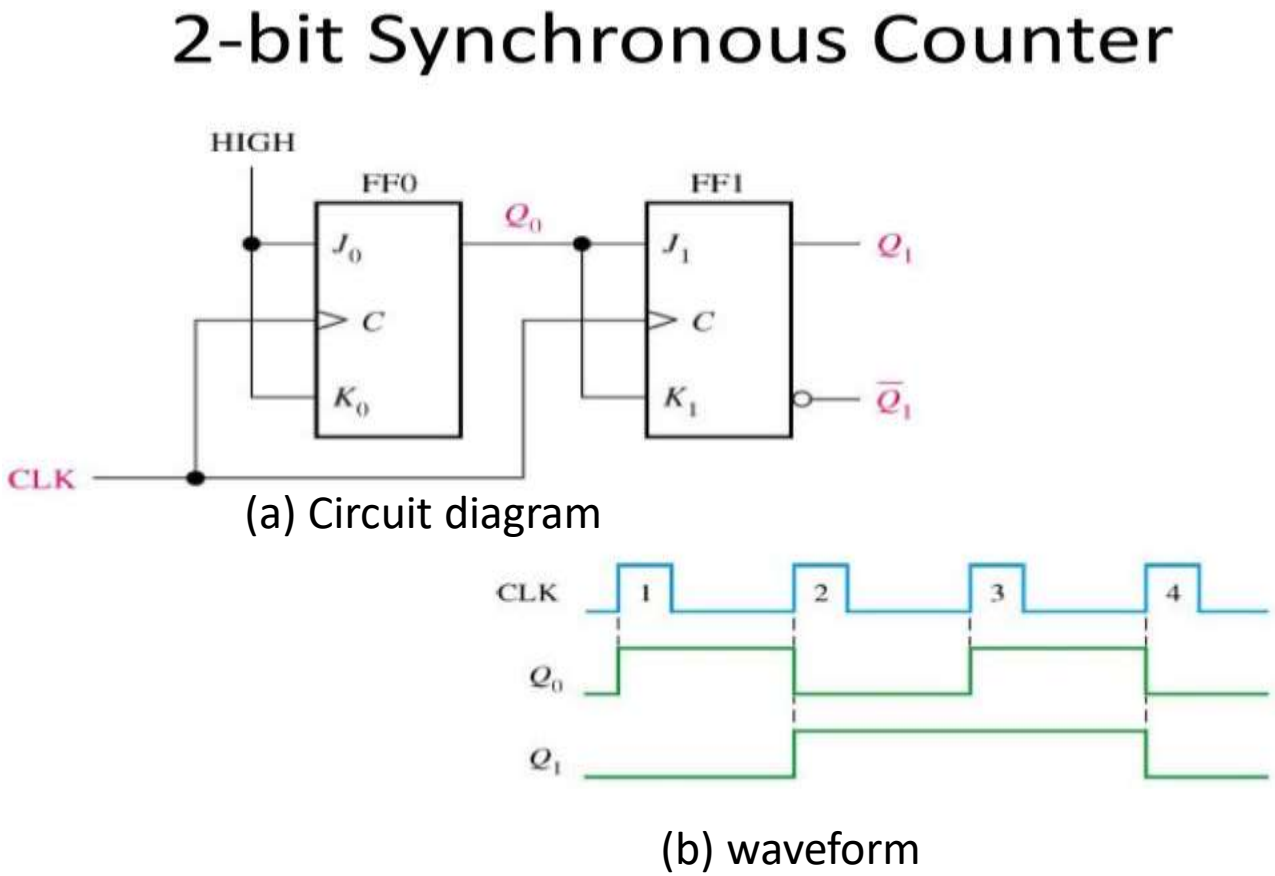
Draw Circuit diagram yourself

Figure: MOD – 11 Counter

Synchronous Counter

Design Procedure of synchronous counter:

1. Determine number of flip-flops & decide type of flip-flop.
2. Draw the excitation table of FF.
3. Make the state diagram and circuit excitation table.
4. Derive simplified equation using k-map.
5. Draw the logic diagram.



3-Bit Synchronous Counter

P.S			N.S			J-K Flip-flop Inputs					
Q2	Q1	Q0	Q2+	Q1+	Q0+	J2	K2	J1	K1	J0	K0
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

From the above excitation table, we can derive the equation of each FFs input using K-map.

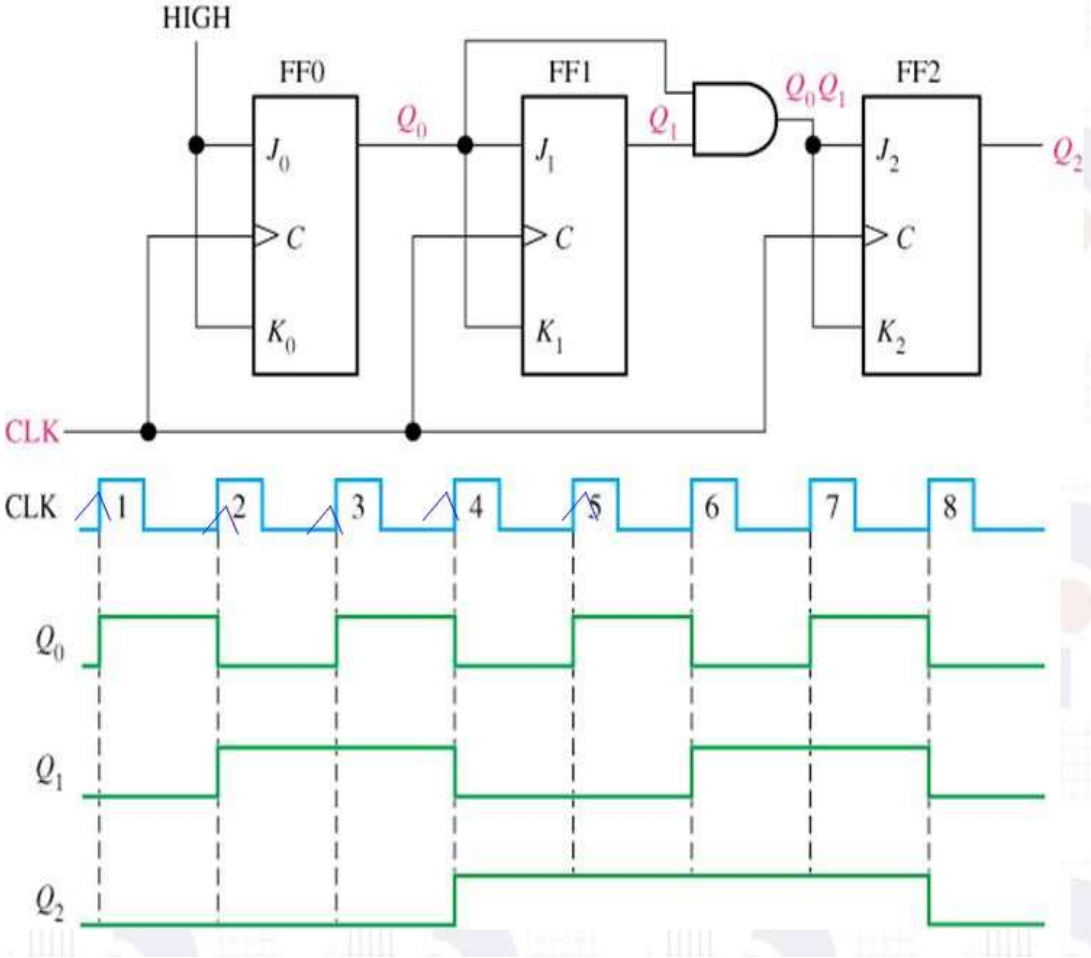
After Simplification we get:

$J_0 = K_0 = 1$ $J_1 = K_1 = Q_0$ $J_2 = K_2 = Q_0.Q_1$

From these equations we have to draw circuit diagram.

Synchronous binary Counter

A 3-bit synchronous binary counter.



Clk pulse	Q2	Q1	Q0
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 (REPEAT)	0	0	0

10

3-Bit Synchronous Down Counter

Step 1: Construct State table

Step 2: We are going to design by J-K FF, so from the information of state table construct excitation table of J-K FF .

Excitation table

Present State			Next State			J-K Flop-Flop Inputs					
Q2	Q1	Q0	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
1	1	1	1	1	0	X	0	X	0	X	1
1	1	0	1	0	1	X	0	X	1	1	X
1	0	1	1	0	0	X	0	0	X	X	1
1	0	0	0	1	1	X	1	1	X	1	X
0	1	1	0	1	0	0	X	X	0	X	1
0	1	0	0	0	1	0	X	X	1	1	X
0	0	1	0	0	0	0	X	0	X	X	1
0	0	0	1	1	1	1	X	1	X	1	X

Step 3: From the excitation Table, simplified J and K equation by K-map.

$J_0=K_0= 1$ $J_1=K_1= \overline{Q}_0$ $J_2= K_2 = \overline{Q}_0 .\overline{Q}_1$

Step 4: Draw a complete design as below.

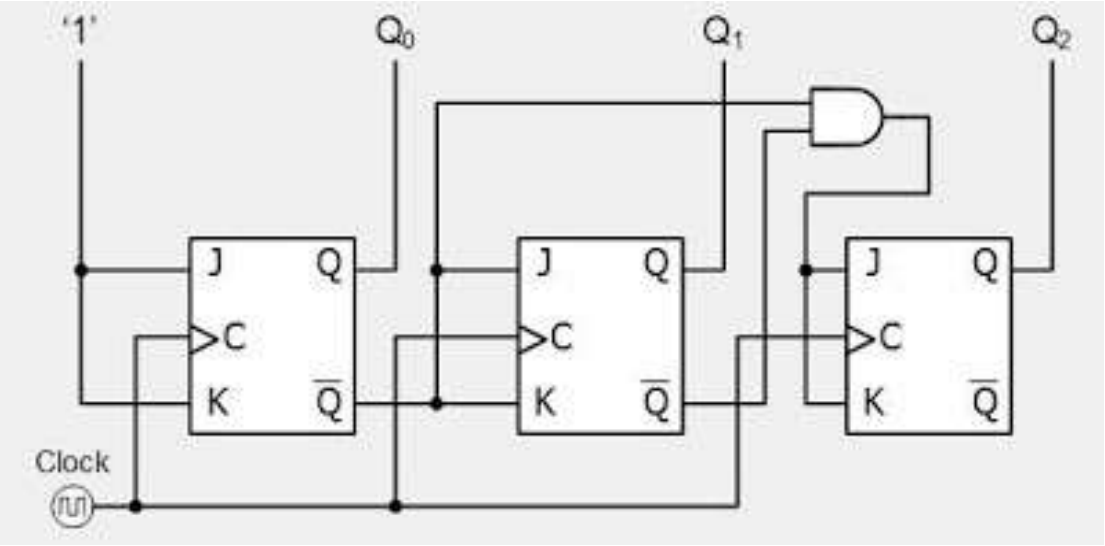
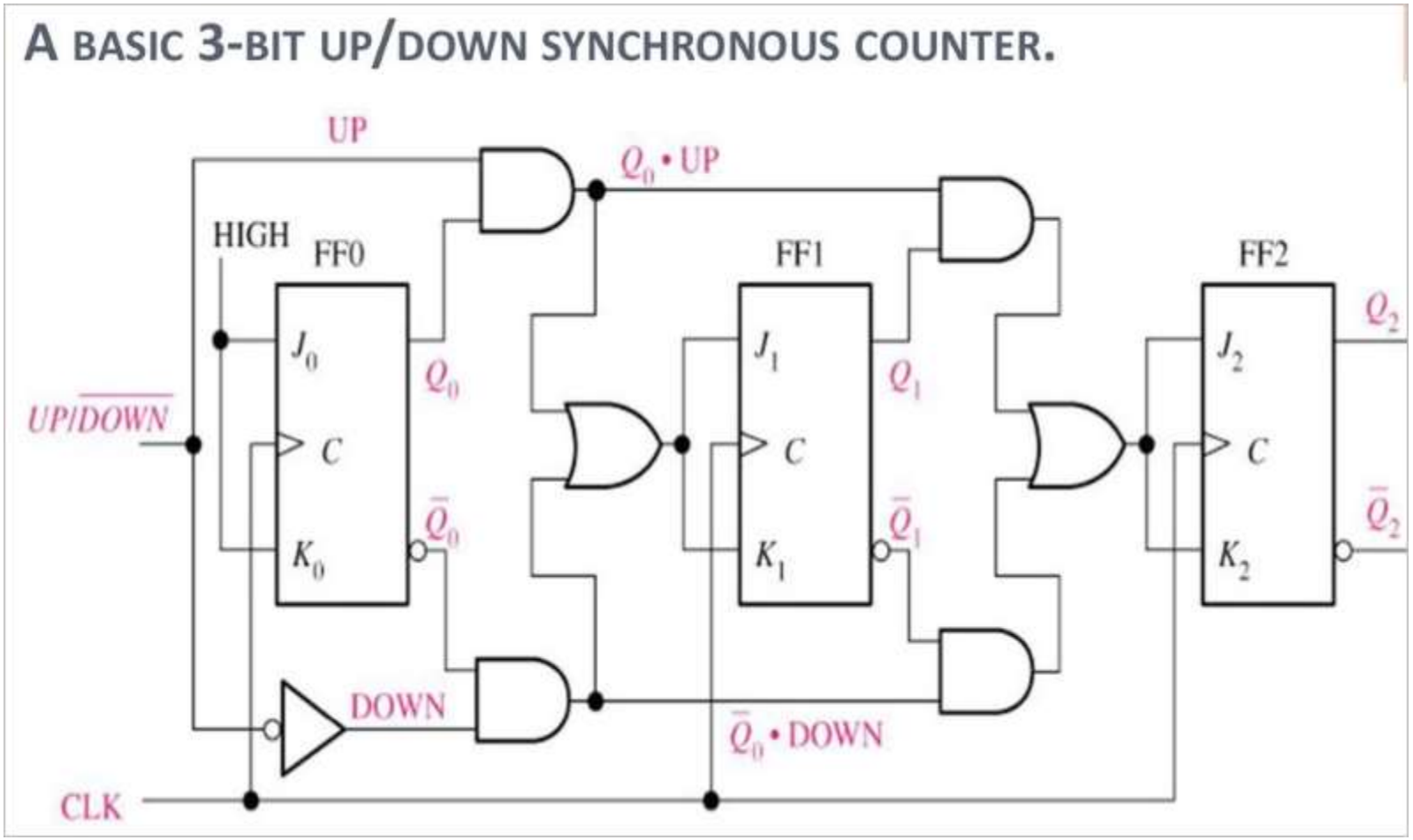


Figure: 3-bit synchronous down counter

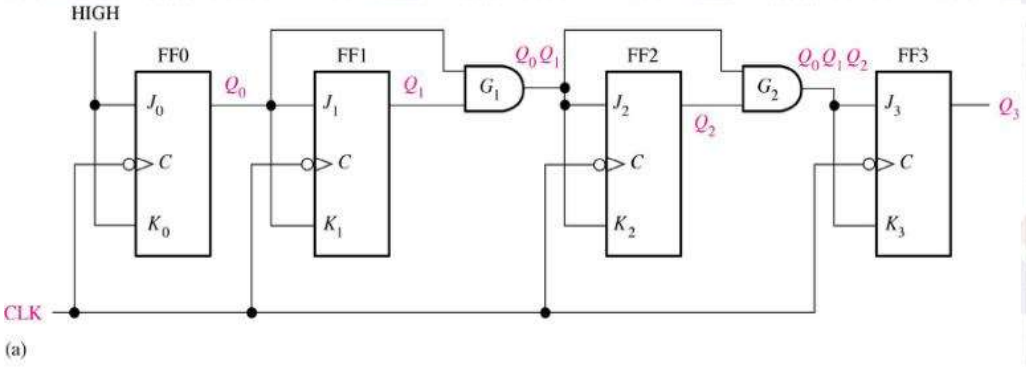
3-Bit Up/Down Synchronous Counter



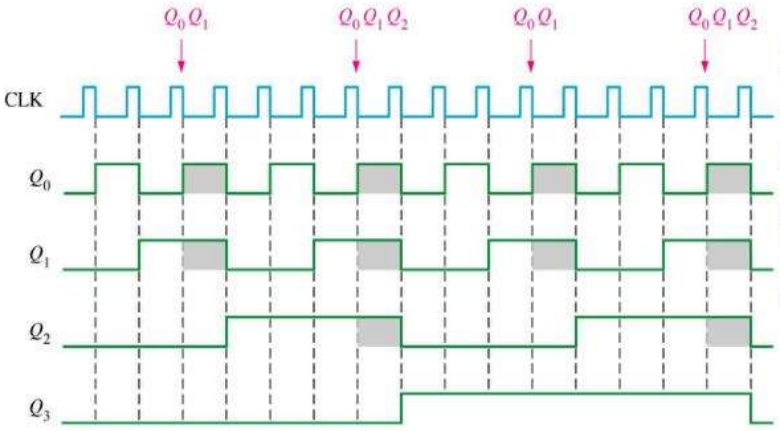
4-Bit Synchronous Counter

DIGITAL SYSTEMS TCE1111

Synchronous binary Counter



(a)

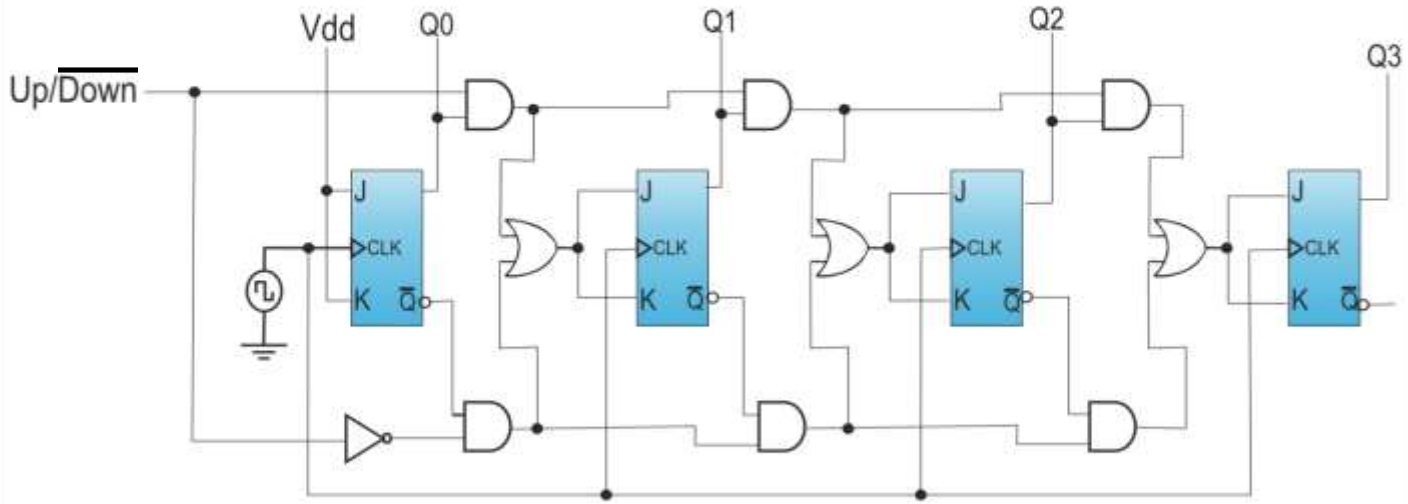


A 4-bit synchronous binary counter and timing diagram. Points where the AND gate outputs are HIGH are indicated by the shaded areas.

CLK PLUSE	Q3	Q2	Q1	Q0
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 REPEAT	0	0	0	0

Note: You have to write excitation table and find out J and K input of each flip-flop by mapping.

4-BIT SYNCHRONOUS "UP/DOWN" COUNTER



Synchronous BCD(Decade) Counter

A BCD decade counter exhibits a truncated binary sequence and goes from 0000 through the 1001 state. Rather than going from the 1001 state to the 1010 state, it recycles to the 0000 state.

Excitation table of BCD counter designed by J-K flip-flop

Present State				Next State				J-K Flip-Flop Inputs							
Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0	J3	K3	J2	K2	J1	K1	J0	K0
0	0	0	0	0	0	O	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	0	0	0	0	0	X	1	X	0	X	X	1

From the excitation table, we can find out the input of each flip-flip by mapping.

J0 = K0 =1

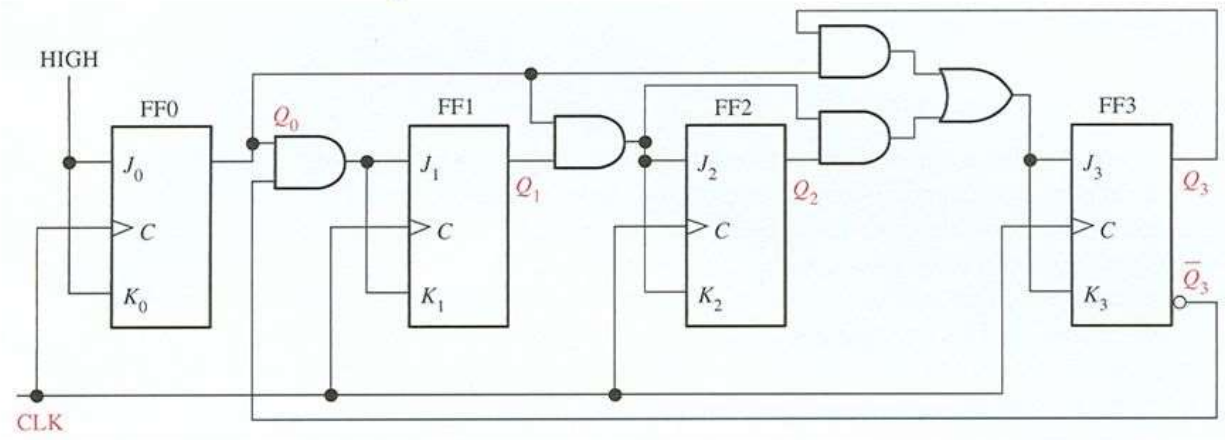
J1= = Q0. $\overline{Q3}$ K1 = Q3

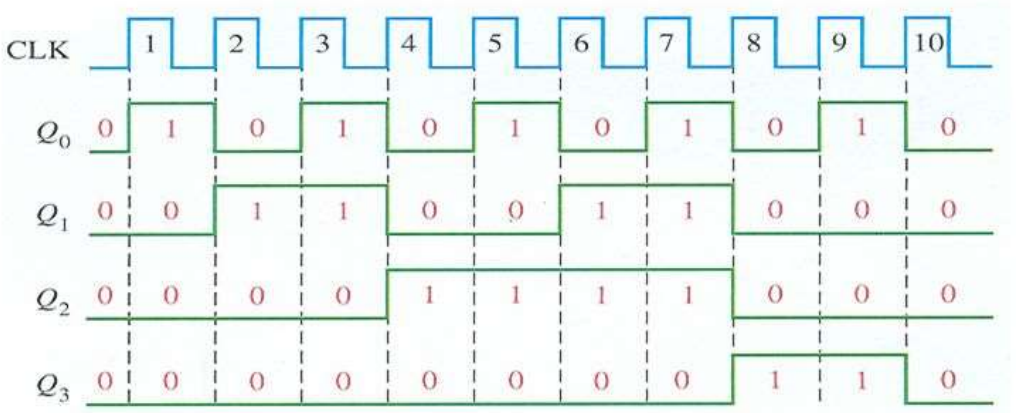
J2=K2= Q0.Q1

J3= Q0.Q1.Q2 and K3 = Q0

Draw the circuit yourself.

A 4 bit Synchronous Decade Counter





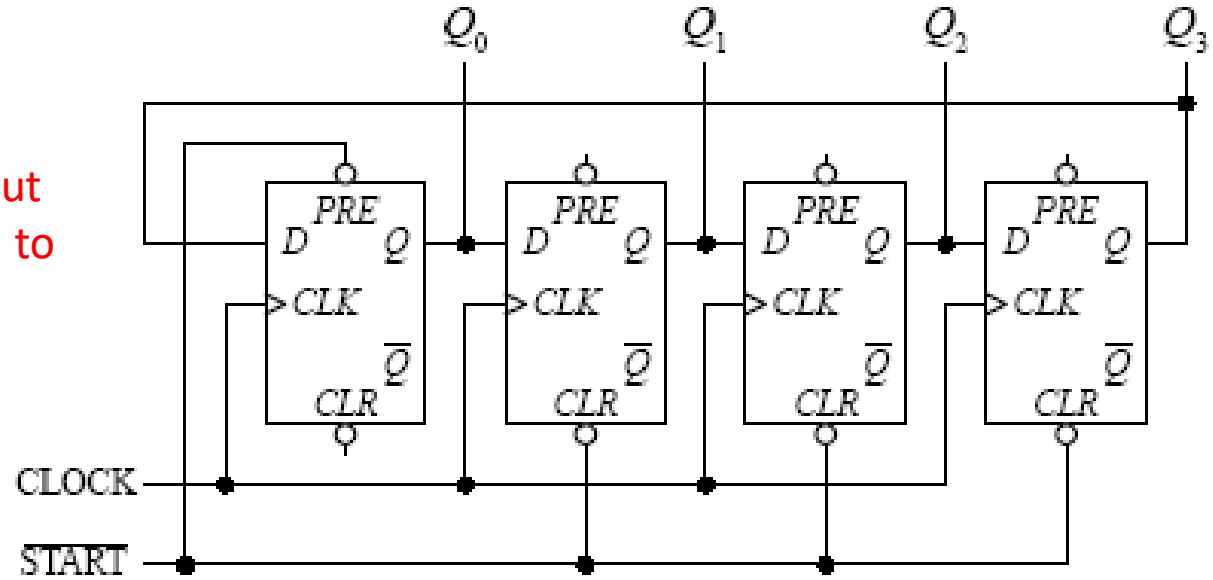
Truth Table:

Q3	Q2	Q1	Q0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

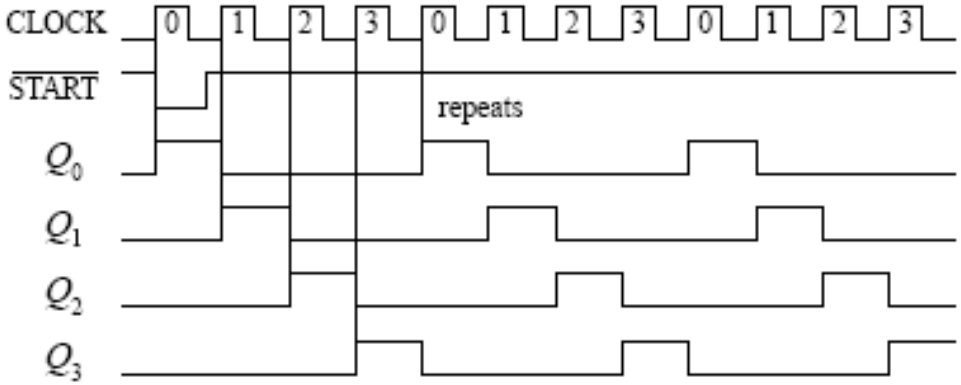
Q0 is toggle at every clock cycle, **so J0=K0= 1**
Q1 is toggle each time when Q0 = 1 and Q3 = 0, **so J1=K1=Q1.Q3**
Q2 is toggle when Q1 and Q0 both are high, **so J2=K2= Q1.Q0**
Q3 is toggle when Q2, Q1 and Q0 all are high OR Q0 qnd Q3 both are high. **SO J3=K3= Q2.Q1.Q0 + Q3.Q0**

Ring Counter

- Ring counter is a typical application of shift register.
- The only change is the output of last flip-flop is connected to the input of first flip-flop.



(a) Circuit diagram



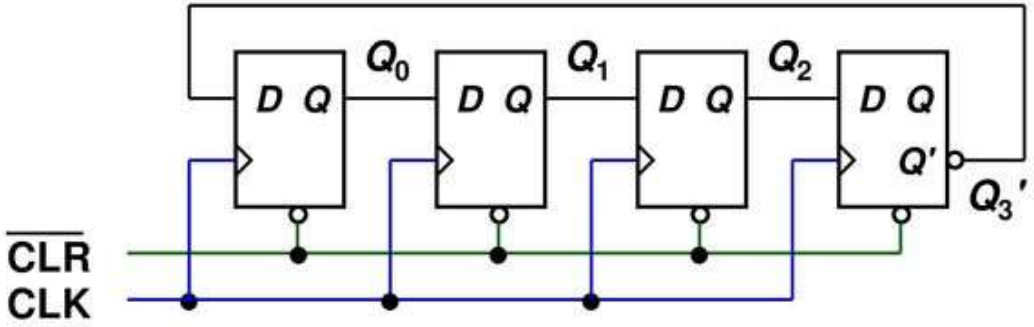
(b) Timing waveform

Johnson Counter

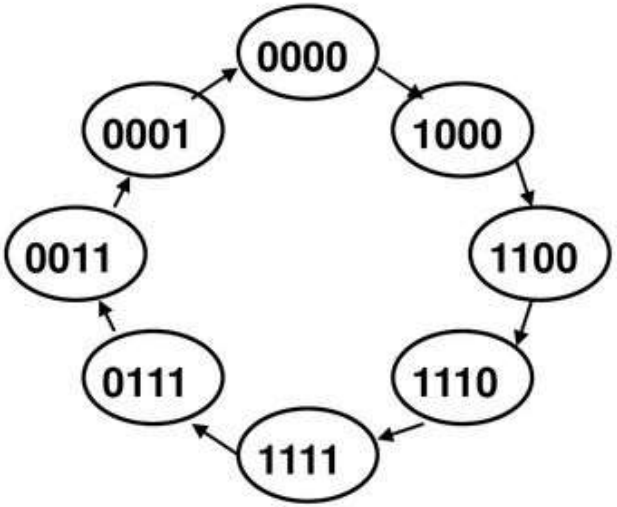
- The Johnson counter is a modification of ring counter. In this counter the inverted output of the last stage flip flop is connected to the input of first flip flop. If we use n flip flops to design the Johnson counter, it is known as 2n bit Johnson counter or Mod 2*n Johnson counter.
- If we use 4 flip-flops, it is a MOD -8 counter.
- Since only 8 (out of 16 possible states) are used, it is called a non binary counter.

Johnson Counters

- **Example**: A 4-bit (MOD-8) Johnson counter.



Clock	Q_0	Q_1	Q_2	Q_3
→0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1



Unit 8 END !!!

Any Queries ???