UNIT III

SYNCHRONOUS SEQUENTIAL CIRCUITS

FLIP-FLOPS

Q1a)Explain the different types of flipflops

(or)

b) Discuss about the different types of flipflops

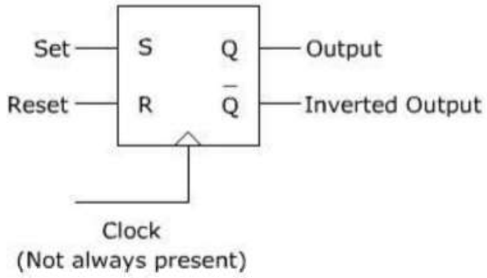
Flip-flops are the first stage in sequential logic design which incorporates memory (storage of
previous states).

Flip-flops that we will look at include the following:

- SR type Flip-flop or Set / Reset
- T type Flip-flop or Triggered /Toggle
- D type Flip-flop or Data / Delay
- JK type Flip-flop

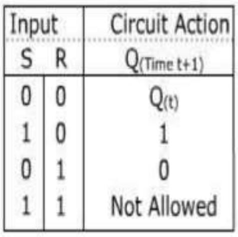
1.SR Flip-flop - (Set / Reset)

This type of flip-flop has two inputs: Set and Reset. Two outputs: Q and Q' (Q' being the inverse of Q). The SR flip-flop can also have a clock input for a level driven circuit as opposed to a pulse driven circuit.

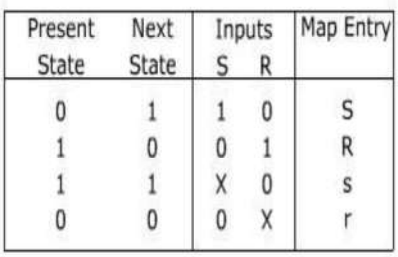


The operation of an SR flip-flop is as follows: The Set input will make Q go to 1 i.e. will 'set' the output. The Reset input will make the output Q go to 0 i.e. reset the output. The scenerio of having both Set and Reset at logic 1 is not allowed as this is not a logical pair of inputs.

Knowing the above, we can layout the operating characteristics and the state change table



Input		Circuit Action
S	R	$Q_{(Time\ t+1)}$
0	0	$Q_{(t)}$
1	0	1
0	1	0
1	1	Not Allowed

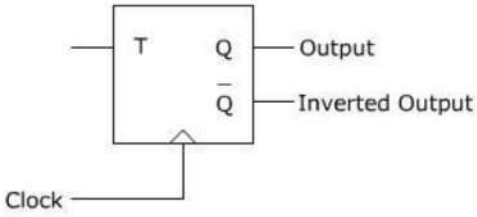


Present State	Next State	Inputs		Map Entry
		S	R	
0	1	1	0	S
1	0	0	1	R
1	1	X	0	s
0	0	0	X	r

There are a few different ways SR flip-flops can be made. They can be pulse driven or clock (and therefore level are used) driven. For the state change diagram above either a pulse or level input can be implied. When using the state change table to describe pulses, a '1' implies a pulse should be applied, where '0' implies that no pulse should exist at this state. For either a pulse driven circuit or a clock driven circuit, the following applies: An 'X' means a pulse / level may or may not be applied. The reason behind this is because no matter of the input (0 or 1), the output will always go to the same value. It is because of this fact that this is considered a 'don't care' input. Hence 's' and 'r' are 'don't care' sets of 'S' and 'R' respectively ('s' leads to the same output as 'S' and that is why 's' is a subset of 'S').

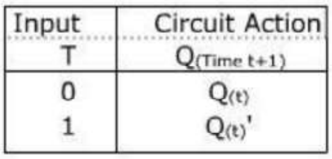
2.T flip-flop (Triggered / Toggle)

The T type flip-flop is a single input device: T (trigger). Two outputs: Q and Q' (where Q' is the inverse of Q).

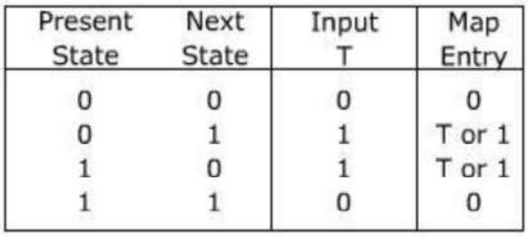


The operation of the T type flip-flop is as follows: A '0' input to 'T' will make the next state the same as the present state (i.e. T = 0 present state = 0 therefore next state = 0). However a '1' input to 'T' will change the next state to the inverse of the present state (i.e. T = 1 present state = 0 therefore next state = 1).

Knowing the above, we can now formalize the operating characteristics and the state change table



Input T	Circuit Action $Q_{(Time\ t+1)}$
0	$Q_{(t)}$
1	$Q_{(t)}'$



Present State	Next State	Input T	Map Entry
0	0	0	0
0	1	1	T or 1
1	0	1	T or 1
1	1	0	0

The T type flip-flop is an edge driven device. Therefore you should not associate 1 and 0 with levels, but instead 1 should be considered as a pulse, and 0 as no pulse.

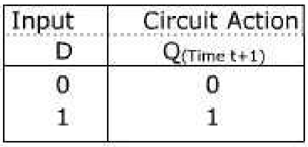
Notice that if a clock signal was tied to T, the output Q would be a clock signal at approximatly half the frequency of T. This property makes the T flip-flop a good candidate for applications such as frequency division.

3.D type flip-flop (Delay)

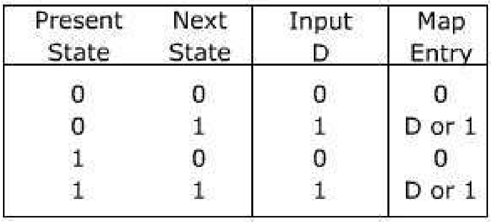
The D type flip-flop has one data input 'D' and a clock input. The circuit edge triggers on. the clock input. The flip-flop also has two outputs Q and Q' (where Q' is the reverse of Q).

The operation of the D type flip-flop is as follows: Any input appearing (present state) at the input D, will be produced at the output Q in time T+1 (next state). e.g. if in the present state we have D = 0 and Q = 1, the next state will be D = anything and Q = 0.

Knowing the above, we can now generate the state change table and the operating characteristics.



Input D	Circuit Action $Q_{(Time\ t+1)}$
0	0
1	1



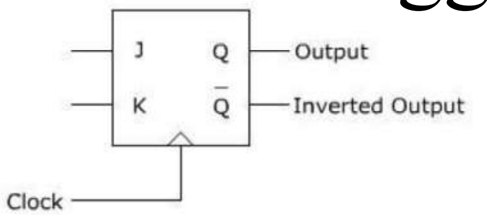
Present State	Next State	Input D	Map Entry
0	0	0	0
0	1	1	D or 1
1	0	0	0
1	1	1	D or 1

The operation of the D type delays any input by exactly one clock cycle (given an instantaneous response time i.e. a perfect flip-flop). Cascading several D type flip-flops together can produce delaying circuits; possible applications could be for matching time delays in digital television systems.

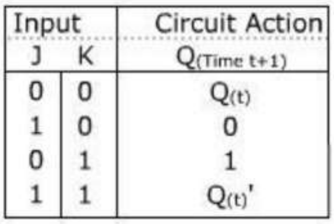
4.JK flip-flop

The JK type flip-flop consists of two data inputs: J and K, and one clock input.

There are again two outputs Q and Q' (where Q' is the reverse of Q).



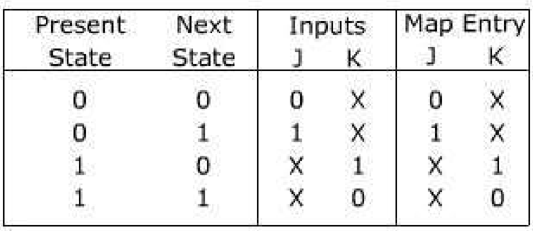
The JK flip-flop operations are quite complicated to understand by text alone. So here we will simply see the operating characteristics diagram and then discuss it.



Input		Circuit Action
J	K	$Q_{(Time\ t+1)}$
0	0	$Q_{(t)}$
1	0	0
0	1	1
1	1	$Q_{(t)}'$

- When J=K=0, the current output will carry through to the next state. e.g. Current state Q = Next state Q
- When J=0 and K=1, the next state output will be put to 0. This happens regardless of the present state output.
- When J=1 and K=0, the next state output will be put to 1. This happens regardless of the present state output.
- When J=K=1, the next state output will be the inverse of the current state output. e.g.
  Current state Q' = Next state Q.

Knowing the above we can now construct the state change table:



Present State	Next State	Inputs		Map Entry	
		J	K	J	K
0	0	0	X	0	X
0	1	1	X	1	X
1	0	X	1	X	1
1	1	X	0	X	0

Flip Flop Conversion

Q2a) Explain the flipflop conversion

(or)

b) convert one flip flop to another flipflop

The following flipflop conversions will be explained

SR Flip Flop to JK Flip Flop

- JK Flip Flop to SR Flip Flop
- SR Flip Flop to D Flip Flop
- D Flip Flop to SR Flip Flop
- JK Flip Flop to T Flip Flop
- JK Flip Flop to D Flip Flop
- D Flip Flop to JK Flip Flop

Follow these steps for converting one flip-flop to the other.

- Consider the characteristic table of desired flip-flop.
- Fill the excitation values inputs of given flip-flop for each combination of present state and next state. The excitation table for all flip-flops is shown below.

Present State	Next State	SR flip-flop inputs		D flip-flop input	JK flip-flop inputs		T flip-flop input
Qtt	**Qt+1t+1**	**S**	**R**	**D**	**J**	**K**	**T**
0	0	0	x	0	0	x	0
0	1	1	0	1	1	x	1
1	0	0	1	0	x	1	1
1	1	x	0	1	x	0	0

- Get the simplified expressions for each excitation input. If necessary, use Kmaps for simplifying.
- Draw the circuit diagram of desired flip-flop according to the simplified expressions using given flip-flop and necessary logic gates.

SR Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of SR flip-flop to other flip-flops.

- SR flip-flop to D flip-flop
- SR flip-flop to JK flip-flop
- SR flip-flop to T flip-flop

Now, let us convert few flip-flops into other. Follow the same process for remaining flipflop conversions.

SR flip-flop to D flip-flop conversion

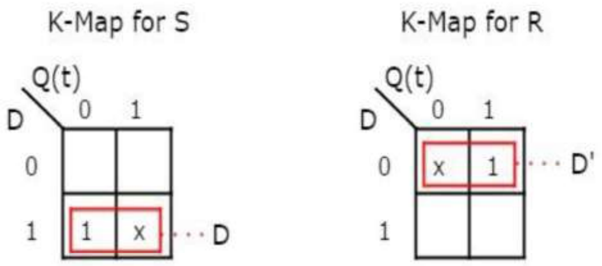
Here, the given flip-flop is SR flip-flop and the desired flip-flop is D flip-flop. Therefore, consider the following characteristic table of desired flip-flop ie) D flip-flop.

D flip-flop input	Present State	Next State
D	**Q**tt	**Q**t+1t+1
0	0	0
0	1	0
1	0	1
1	1	1

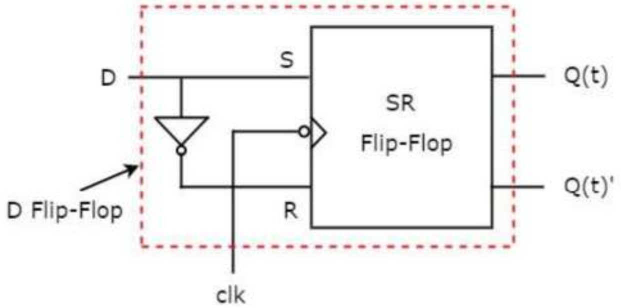
We know that SR flip-flop has two inputs S & R. So, write down the excitation values of SR flip-flop for each combination of present state and next state values. The following table shows the characteristic table of D flip-flop along with the excitation inputs of SR flip-flop.

D flip-flop input	Present State	Next State	SR flip-flop inputs	
D	**Q**tt	**Q**t+1t+1	**S**	**R**
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

We can use 2 variable K-Maps for getting simplified expressions for these inputs. The k-Maps for S & R are shown below.



So, we got S = D & R = D' after simplifying. The circuit diagram of D flip-flop is shown in the following figure.



This circuit consists of SR flip-flop and an inverter. This inverter produces an output, which is complement of input, D. So, the overall circuit has single input, D and two outputs Q(t)& Q(t'). Hence, it is a **D flip-flop**. Similarly, you can do other two conversions.

D Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of D flip-flop to other flip-flops.

- D flip-flop to T flip-flop
- D flip-flop to SR flip-flop
- D flip-flop to JK flip-flop

Now, let us convert few flip-flops into other. Follow the same process for remaining flipflop conversions.

D flip-flop to T flip-flop conversion

Here, the given flip-flop is D flip-flop and the desired flip-flop is T flip-flop. Therefore, consider the following characteristic table of desired flipflop ie) T flip-flop.

T flip-flop input	Present State	Next State
T	Qtt	Qt+1t+1
0	0	0
0	1	1
1	0	1
1	1	0

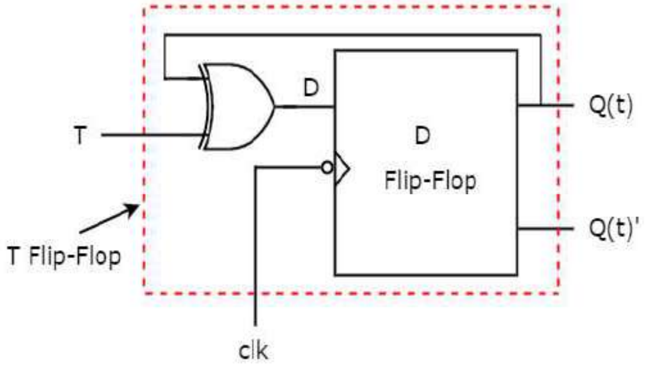
We know that D flip-flop has single input D. So, write down the excitation values of D flip-flop for each combination of present state and next state values. The following table shows the characteristic table of T flip-flop along with the excitation input of D flip-flop.

T flip-flop input	Present State	Next State	D flip-flop input
T	Qtt	Qt+1t+1	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

From the above table, we can directly write the Boolean function of D as below.

$$D= =T\oplus Q(t)$$

So, we require a two input Exclusive-OR gate along with D flip-flop. The circuit diagram of T flip-flop is shown in the following figure.



This circuit consists of D flip-flop and an Exclusive-OR gate. This Exclusive-OR gate produces an output, which is Ex-OR of T and Qtt. So, the overall circuit has single input, T and two outputs Qtt & Qtt'. Hence, it is a T flip-flop. Similarly, you can do other two conversions.

JK Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of JK flip-flop to other flip-flops.

- JK flip-flop to T flip-flop
- JK flip-flop to D flip-flop
- JK flip-flop to SR flip-flop

Now, let us convert few flip-flops into other. Follow the same process for remaining flipflop conversions.

JK flip-flop to T flip-flop conversion

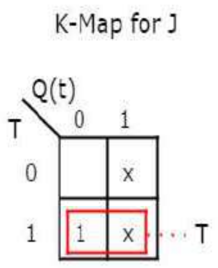
Here, the given flip-flop is JK flip-flop and the desired flip-flop is T flip-flop. Therefore, consider the following characteristic table of desired flipflop ie) T flip-flop.

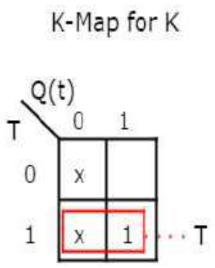
T flip-flop input	Present State	Next State
T	Qtt	Qt+1t+1
0	0	0
0	1	1
1	0	1
1	1	0

We know that JK flip-flop has two inputs J & K. So, write down the excitation values of JK flip-flop for each combination of present state and next state values. The following table shows the characteristic table of T flip-flop along with the excitation inputs of JK flipflop.

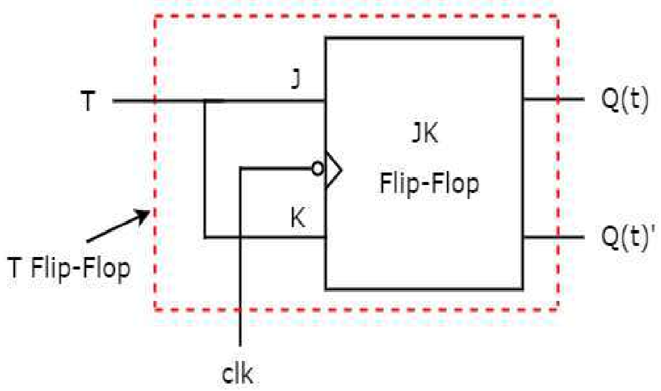
T flip-flop input	Present State	Next State	JK flip-flop inputs	
T	Qtt	Qt+1t+1	J	K
0	0	0	0	x
0	1	1	x	0
1	0	1	1	x
1	1	0	x	1

We can use 2 variable K-Maps for getting simplified expressions for these two inputs. The k-Maps for J & K are shown below.





So, we got, J = T & K = T after simplifying. The circuit diagram of T flip-flop is shown in the following figure.



This circuit consists of JK flip-flop only. It doesn't require any other gates. Just connect the same input T to both J & K. So, the overall circuit has single input, T and two outputs Qtt& Qtt'. Hence, it is a T flip-flop. Similarly, you can do other two conversions.

T Flip-Flop to other Flip-Flop Conversions

Following are the three possible conversions of T flip-flop to other flip-flops.

- T flip-flop to D flip-flop
- T flip-flop to SR flip-flop
- T flip-flop to JK flip-flop

Now, let us convert few flip-flops into other. Follow the same process for remaining flipflop conversions.

T flip-flop to D flip-flop conversion

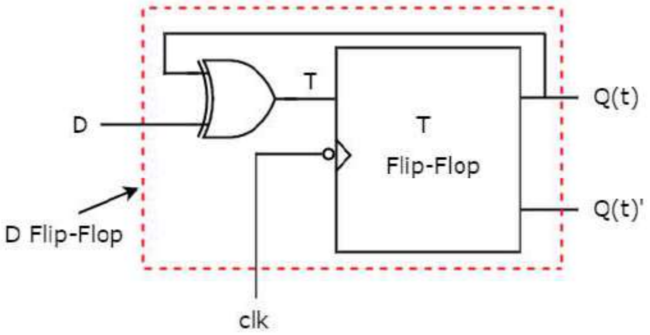
Here, the given flip-flop is T flip-flop and the desired flip-flop is D flip-flop. Therefore, consider the characteristic table of D flip-flop and write down the excitation values of T flip-flop for each combination of present state and next state values. The following table shows the characteristic table of D flip-flop along with the excitation input of T flip-flop.

D flip-flop input	Present State	Next State	T flip-flop input
D	Qtt	Qt+1t+1	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

From the above table, we can directly write the Boolean function of T as below.

$$T=D\oplus Q(t)$$

So, we require a two input Exclusive-OR gate along with T flip-flop. The circuit diagram of D flip-flop is shown in the following figure.



This circuit consists of T flip-flop and an Exclusive-OR gate. This Exclusive-OR gate produces an output, which is Ex-OR of D and Qt. So, the overall circuit has single input, D and two outputs Qt & Qt'. Hence, it is a D flip-flop. Similarly, you can do other two conversions.

COUNTERS

Q3a) Discuss about counters

(or)

b) Explain the different types of counters?

Counters and registers belong to the category of MSI sequential logic circuits. They have similar architecture, as both counters and registers comprise a cascaded arrangement of more than one flipflop with or without combinational logic devices. Counters are mainly used in counting applications, where they either measure the time interval between two unknown time instants or measure the frequency of a given signal, registers are primarily used for the temporary storage of data present at the output of a digital circuit before they are fed to another digital circuit.

Counters are mainly classified into 2.

1. Synchronous Counters
2. Ripple (Asynchronous) Counter

1. Synchronous Counters

Synchronous counters are simple state machines made out of flip flops and logic gates. They have two parts, a register made out of flip flops and a decoder made out of logic gates. A register is a simple group of flip flops that are all clocked at the same time. In this way they can hold the counters output value until the next clock cycle. The decoder, decodes the current count and generates the correct value for the next count to the flop flops. For example in a simple up counter the decoder would always output the current count plus one. The major advantage of Synchronous Counters is that all the bits of their output change at the same time. In a synchronous counter, also known as a parallel counter, all the flip-flops in the counter change state at the same time in synchronism with the input clock signal.

Binary 4-bit Synchronous Up Counter

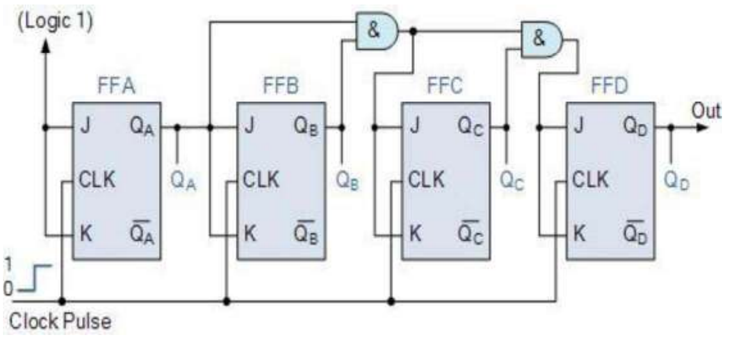
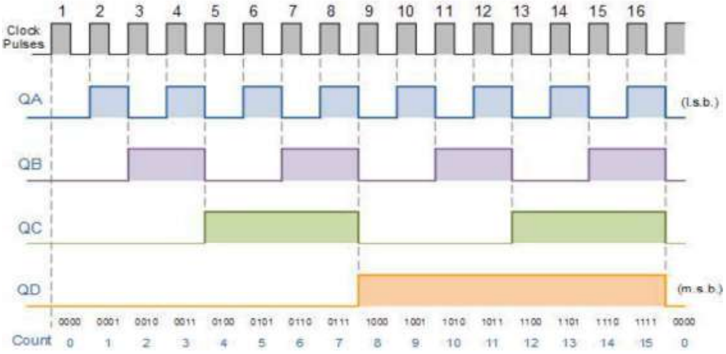


Figure : 4-bit Synchronous Up Counter

Here the external clock pulses (pulses to be counted) are fed directly to each of the J-K flip-flops in the counter chain and that both the J and K inputs are all tied together in toggle mode, but only in the first flip-flop, flip-flop FFA (LSB) are they connected HIGH, logic 1 allowing the flip-flop to toggle on every clock pulse. Then the synchronous counter follows a predetermined sequence of states in response to the common clock signal, advancing one state for each pulse.

The J and K inputs of flip-flop FFB are connected directly to the output Q_A of flip-flop FFA, but the J and K inputs of flip-flops FFC and FFD are driven from separate AND gates which are also supplied with signals from the input and output of the previous stage. These additional AND gates generate the required logic for the JK inputs of the next stage.

4-bit Synchronous Counter Waveform Timing Diagram.



Because this 4-bit synchronous counter counts sequentially on every clock pulse the resulting outputs count upwards from 0 (0000) to 15 (1111). Therefore, this type of counter is also known as a 4-bit Synchronous Up Counter.

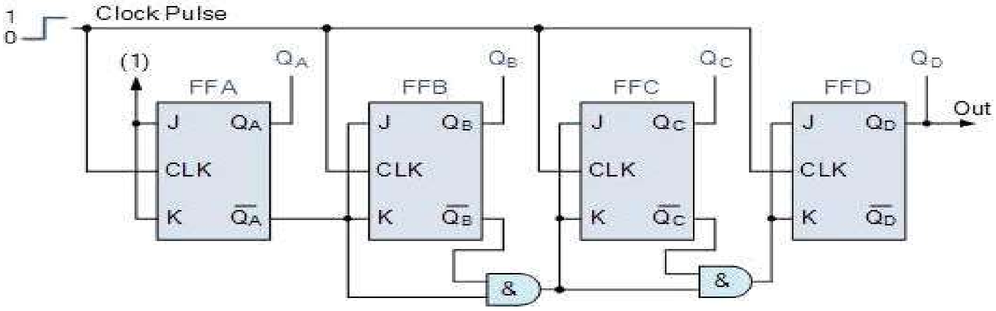


Figure 4-bit Synchronous Down Counter

Binary 4-bit Synchronous Down Counter

In this, Binary 4-bit Synchronous Down Counter we can easily construct a 4-bit Synchronous Down Counter by connecting the AND gates to the Q output of the flip-flops as shown to produce a waveform timing diagram the reverse of the above. Here the counter starts with all of its outputs HIGH (1111) and it counts down on the application of each clock pulse to zero, (0000) before repeating again.

Decade 4-bit Synchronous Counter

A 4-bit decade synchronous counter can also be built using synchronous binary counters to produce a count sequence from 0 to 9. A standard binary counter can be converted to a decade (decimal 10) counter with the aid of some additional logic to implement the desired state sequence. After reaching the count of 9(1001), the counter recycles back to 0 (0000).

Decade 4-bit Synchronous Counter

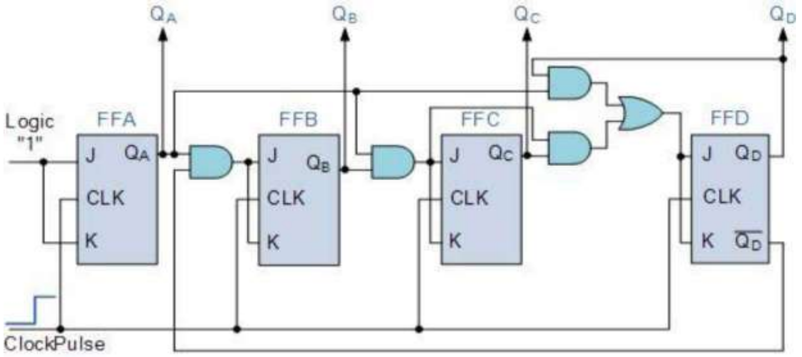


Figure : Decade 4-bit Synchronous Counter

Ripple (Asynchronous) Counter

Ripple counters are the simplest type of counters. They are nothing more than toggle flip flops connected in a chain to divide each others output frequency by two. The result is a binary count. They are called ripple counters because the new count ripples through them. The major disadvantage of ripple counters is that because of new count "rippling" through the flip flops all the bits of the count arrive at different times. In a ripple counter, also called an asynchronous counter or a serial counter, the clock input is applied only to the first flip-flop, also called the input flip-flop, in the cascaded arrangement. The clock input to any subsequent flip-flop comes from the output of its immediately preceding flip-flop.

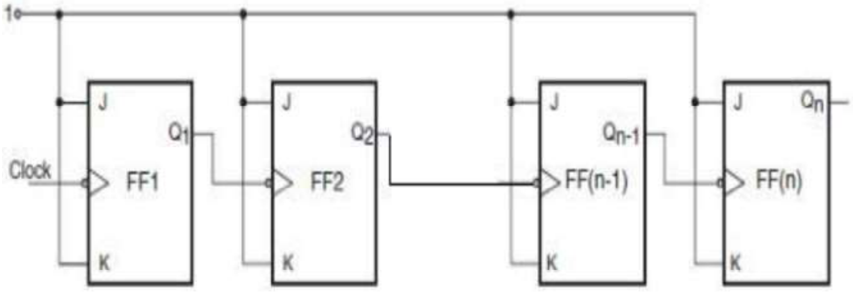


Figure :Generalized block schematic of n-bit binary ripple counter.

Figure shows the generalized block schematic arrangement of an n-bit binary ripple counter. As a natural consequence of this, not all flip-flops change state at the same time. The second flip-flop can change state only after the output of the first flip-flop has changed its state. That is, the second flip-flop would change state a certain time delay after the occurrence of the input clock pulse owing to the fact that it gets its own clock input from the output of the first flip-flop and not from the input clock. This time delay here equals the sum of propagation delays of two flip-flops, the first and the second flip-flops.

UP/DOWN Counters

Counters are also available in integrated circuit form as UP/DOWN counters, which can be made to operate as either UP or DOWN counters. An UP counter is one that counts upwards or in the forward direction by one LSB every time it is clocked. A four-bit binary UP counter will count as 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011,1100, 1101, 1110, 1111, 0000, 0001, _ _ _ and so on. A DOWN counter counts in the reverse direction or downwards by one LSB every time it is clocked. The four-bit binary DOWN counter will count as0000, 1111, 1110, 1101, 1100, 1011, 1010, 1001, 1000, 0111, 0110, 0101, 0100, 0011, 0010, 0001, 0000, 1111, _ _ _ and so on. Some counter ICs have separate clock inputs for UP and DOWN counts, while others have a single clock input and an UP/DOWN control pin. The logic status of this control pin decides the counting mode. As an example, ICs 74190 and 74191 are four-bit UP/DOWN counters in the TTL family with a single clock input and an UP/DOWN control pin. While IC 74190 is a BCD decade counter, IC74191 is a binary counter. Figure shows a 3-bit binary UP/DOWN counter.

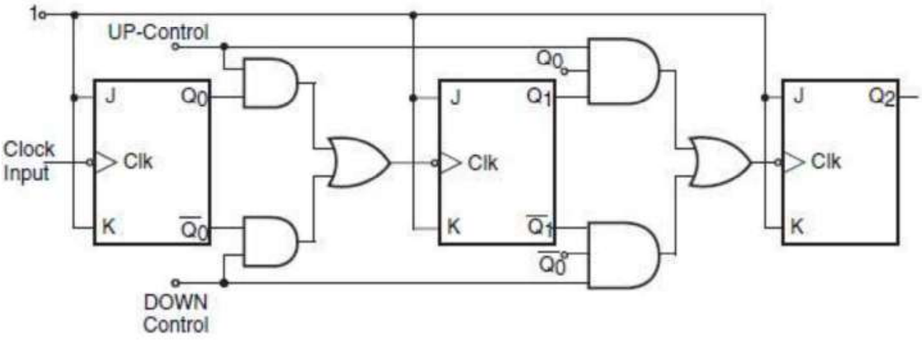


Figure : 3 bit UP/DOWN Counter

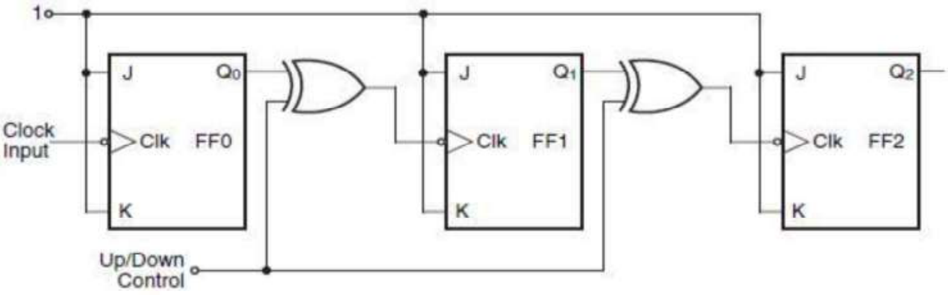


Figure: 3 bit UP/DOWN counter with a common clock input control is logic _0‘. In this case the clock input of each flip-flop other than the LSB flip-flop is fed from the normal output of the immediately preceding flip-flop. The counter counts downwards when the UP control input is logic _0‘ and DOWN control is logic _1‘.Figure shows another possible configuration for a 3-bit binary ripple UP/DOWN counter. It has a common control input. When this input is in logic _1‘ state the counter counts downwards, and when it is in logic _0‘ state it counts upwards.

Shift Register

Q4a) Explain in detail about shift registers?

(or)

b)Describe the different types of shift registers

A shift register is a digital device used for storage and transfer of data. The data to be stored could be the data appearing at the output of an encoding matrix before they are fed to the main digital system for processing or they might be the data present at the output of a microprocessor before they are fed to the driver circuitry of the output devices. The shift registers can also be configured to construct some special types of counter that can be used to perform a number of arithmetic operations such as subtraction, multiplication, division, complementation, etc. Based on the method used to load data onto and read data from shift registers, they are classified as

- serial-in serial-out (SISO) shift registers,
- serial-in parallel-out (SIPO) shift registers,
- parallel-in serial-out (PISO) shift registers and
- parallel-in parallel-out (PIPO) shift registers.

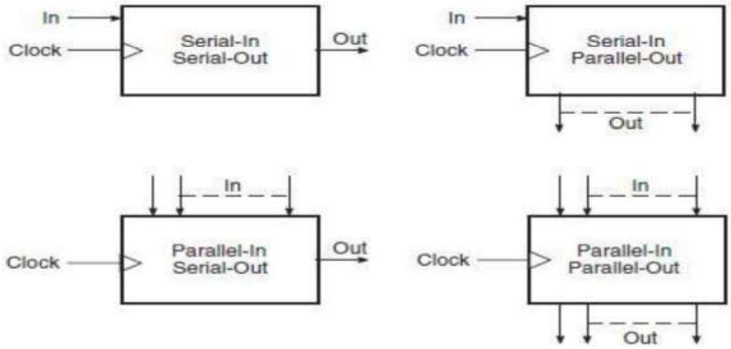


Figure : Circuit representation of shift registers

Serial-In Serial-Out Shift Register

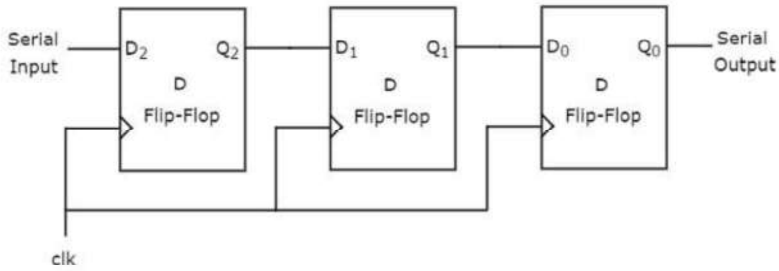


Figure : SISO shift register

Figure shows the basic 3-bit serial-in serial-out shift register implemented using D flip-flops. The circuit functions as follows. A reset applied to the CLEAR input of all the flip-flops resets their Q outputs to 0s.

Serial-In Parallel-Out Shift Register

A serial-in parallel-out shift register is architecturally identical to a serial-in serial-out shift register except that in the case of the former all flip-flop outputs are also brought out on the IC terminals. The gated serial inputs A and B control the incoming serial data, as a logic LOW at either of the inputs inhibits entry of new data and also resets the first flip-flop to the logic LOW level at the next clock pulse. Logic HIGH at either of the inputs enables the other input, which then determines the state of the first flip-flop. Data at the serial inputs may be changed while the clock input is HIGH or LOW, and the register responds to LOW-to-HIGH transition of the clock.

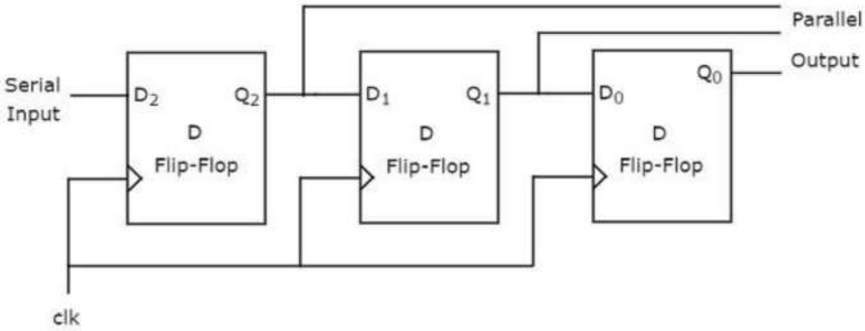


Figure : SIPO shift register

Parallel-In Serial-Out Shift Register

We will explain the operation of a parallel-in serial-out shift register with the help of the logic diagram of a practical device available in IC form. The parallel-in or serial-in modes are controlled by a SHIFT/LOAD input. When the SHIFT/LOAD input is held in the logic HIGH state, the serial data input AND gates are enabled and the circuit behaves like a serial-in serial-out shift register. When the SHIFT/LOAD input is held in the logic LOW state, parallel data input AND gates are enabled and data are loaded in parallel, in synchronism with the next clock pulse. Clocking is accomplished on the LOW-to-HIGH transition of the clock pulse via a two-input NOR gate. Holding one of the inputs of the NOR gate in the logic HIGH state inhibits the clock applied to the other input. Holding an input in the logic LOW state enables the clock to be applied to the other input. An active LOW CLEAR input overrides all the inputs, including the clock, and resets all flip-flops to the logic _0‘ state.

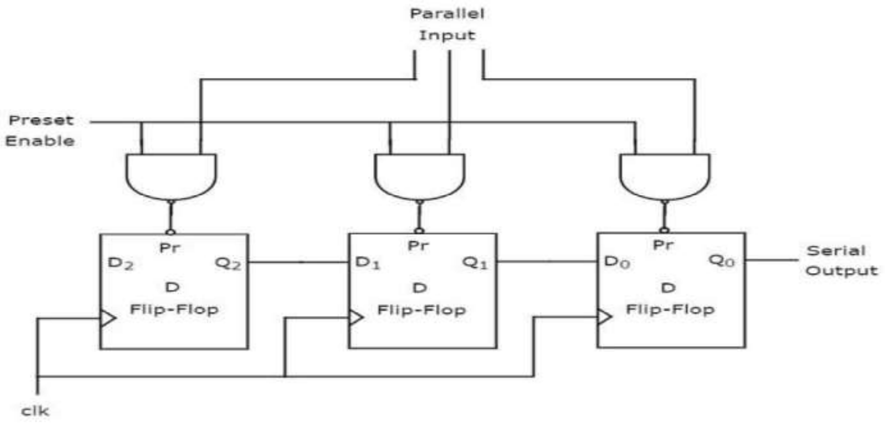


Figure : PISO shift register

Parallel-In Parallel-Out Shift Register

The hardware of a parallel-in parallel-out shift register is similar to that of a parallel-in serial-out shift register. If in a parallel-in serial-out shift register the outputs of different flip-flops are brought out, it becomes a parallel-in parallel-out shift register. In fact, the logic diagram of a parallel-in parallel-out shift register is similar to that of a parallel-in serial-out shift register. As an example, IC74199 is an eight-bit parallel-in parallel-out shift register.

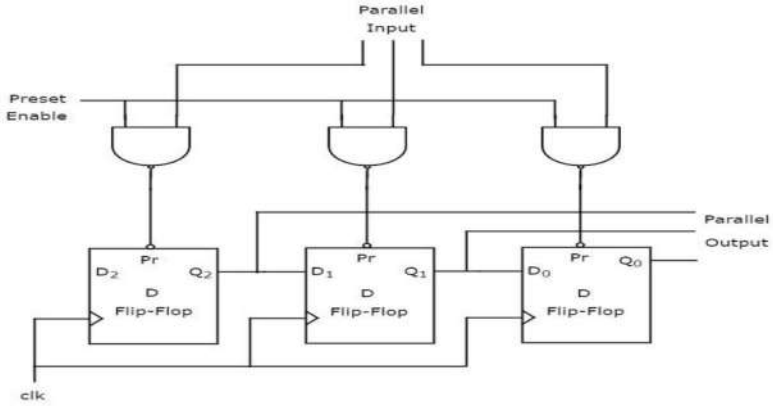


Figure : PIPO shift register

Shift Register Counters

Q4a) explain the types of shift register counters

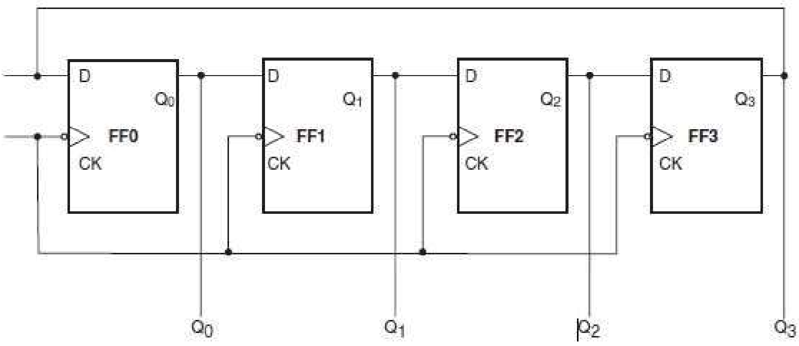
(or)

b) Discuss about the types of shift register counters

Both counters and shift registers are some kinds of cascade arrangement of flip-flops. A shift register, unlike a counter, has no specified sequence of states. However, if the serial output of the shift register is fed back to the serial input, we do get a circuit that exhibits a specified sequence of states. The resulting circuits are known as shift register counters. Depending upon the nature of the feedback, we have two types of shift register counter, namely the ring counter and the shift counter, also called the Johnson counter.

1.Ring Counter

A ring counter is obtained from a shift register by directly feeding back the true output of the output flip-flop to the data input terminal of the input flip-flop. If D flip-flops are being used to construct the shift register, the ring counter, also called a circulating register, can be constructed by feeding back the Q output of the output flip-flop back to the D input of the input flip-flop.



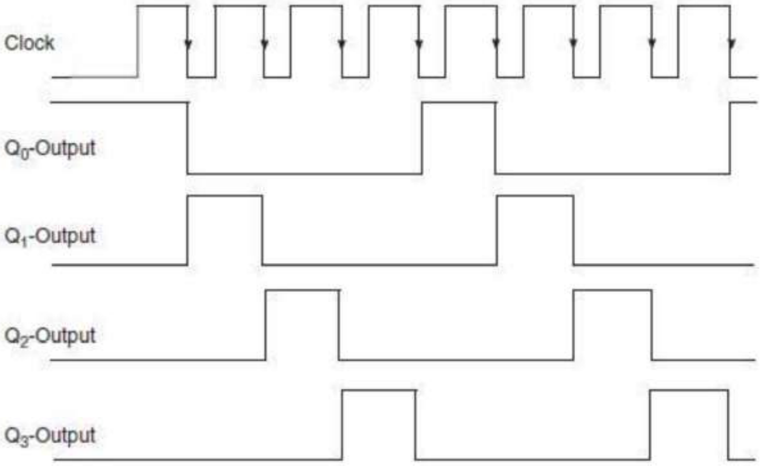


Figure : Timing waveforms of the four bit ring counter

2. Shift Counter

A shift counter on the other hand is constructed by having an inverse feedback in a shift register. For instance, if we connect the Q output of the output flip-flop back to the K input of the input flip-flop and the Q output of the output flip-flop to the J input of the input flip-flop in a serial shift register, the result is a shift counter, also called a Johnson counter.

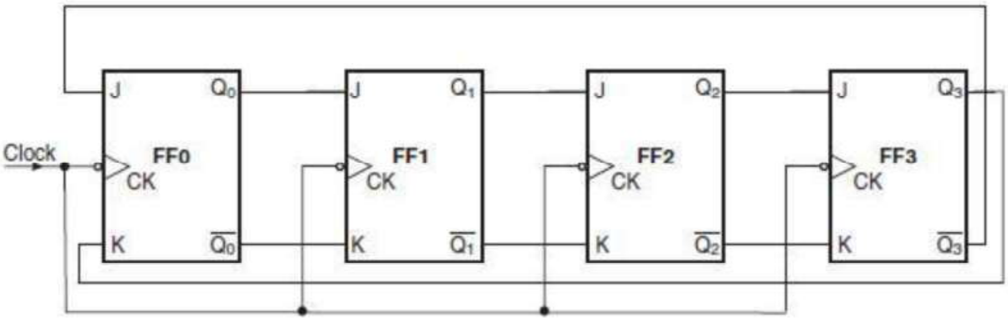


Figure :four bit shift counter

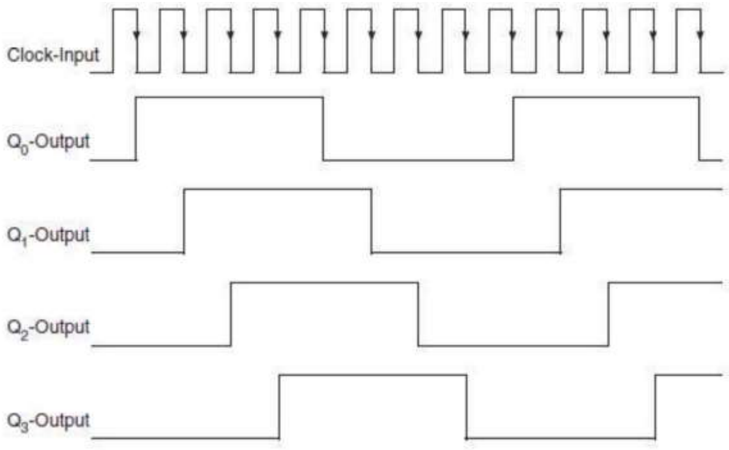


Figure : Timing waveforms of the shift counter

Synchronous and Asynchronous sequential circuits

We have already introduced to synchronous sequential circuits. These circuits are further classified depending on the timing of their signals: Synchronous sequential circuits and Asynchronous Sequential Circuits. In synchronous sequential circuits signals can affect the memory elements only at discrete instants of time. In asynchronous sequential circuits change in input signals can affect memory element at any instant of time.

Q5a) Difference between synchronous and asynchronous sequential circuits

(or)

b) Discuss about synchronous and asynchronous sequential circuits

Sl.No	Synchronous sequential circuits	Asynchronous sequential circuits

1.	In synchronous circuits memory elements are clocked flipflops	In asynchronous circuits memory elements are either unclocked flipflops or time delay elements.
2.	The change in input signals can affect memory element upon activation of clock signal	The change in input signals can affect memory element at any instant of time
3.	The maximum operating speed of clock depends on time delays involved	Because of absence of clock asynchronous circuits can operate faster than synchronous circuits
4.	Easier to design	More difficult to design

Table: Comparison between synchronous sequential circuits and asynchronous sequential circuits

Clocked sequential circuits

Q6a) Discuss about clocked sequential circuits

(or)

b)Explain in detail about synchronous or clocked sequential circuits

In synchronous or clocked sequential circuits clocked flipflops are used as memory elements which change their individual states in synchronism with the periodic clock signal. The change in states of flipflop and change in state of the entire circuit occur at the transition of the clock signal.The states of the output of the flipflop in the sequential circuit give the state of the sequential circuit.

Present state

The status of all state variables at some time t, before the next clock edge represent condition called present state.

Next state

The status of all state variables at some time t+1 represent a condition called next state.

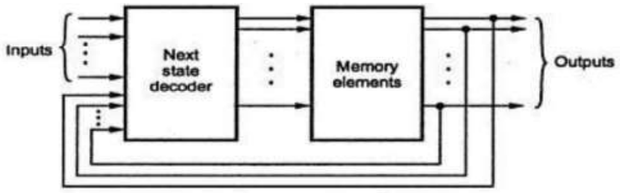
The synchronous or clocked sequential circuits are represented by two models.

Moore model: The output depends only on the present state of the flipflops.

Mealy model: The outputs depends on both the present state of the flipflops and on the inputs

Moore model

When the output of the sequential circuit depends only on the present state of the flipflop the sequential circuit is referred to as Moore model. Let us see one example of Moore model.



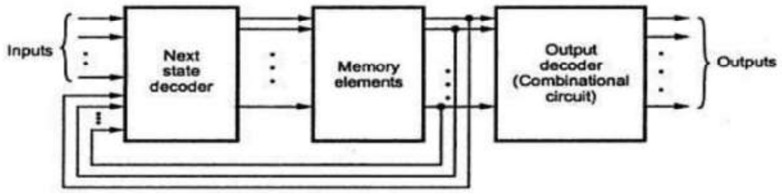


Figure: Moore circuit model with an output decoder

Mealy model:

When the output of the sequential circuit depends on both the present state of the flipflops and on the inputs, the sequential circuit is referred to as mealy model. Figure shows the sample mealy model.

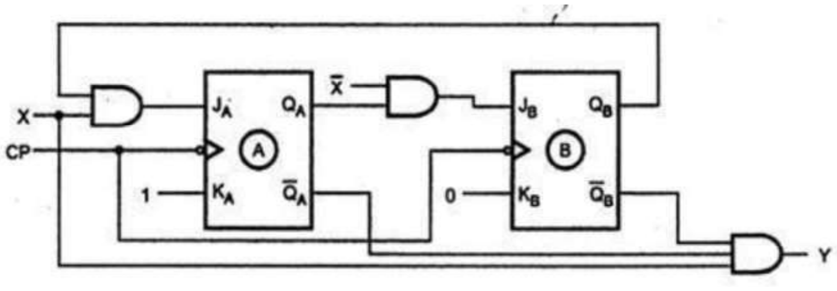
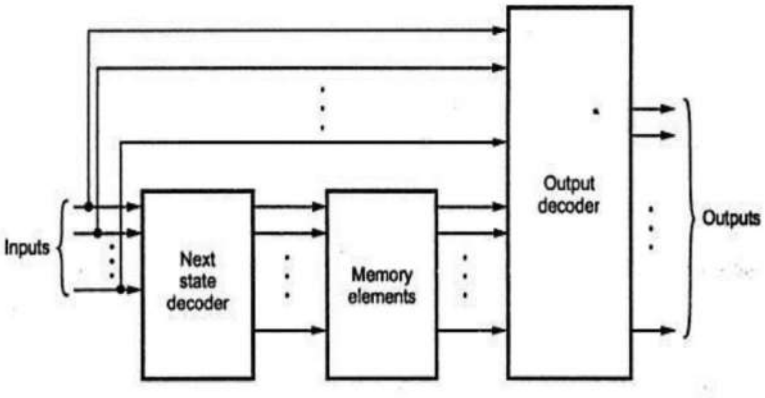


Figure: Example of mealy model

Looking at Figure we can easily realize that, changes in the input within the clock pluses cannot affect the state of the flip-flop. However, they can affect the output of the circuit. Due to this, if the input variations are not synchronized with the clock, the derived output also not be synchronized with the clock and we get false output (as it is a synchronous sequential network). The false outputs can be eliminated by allowing input to change only at the active transition of the clock (in our example HIGH –to-LOW).In general form the Mealy model can be represented with in its block schematic as shown as in figure.



`

Figure: Mealy circuit model

Moore Vs Mealy Circuit Models

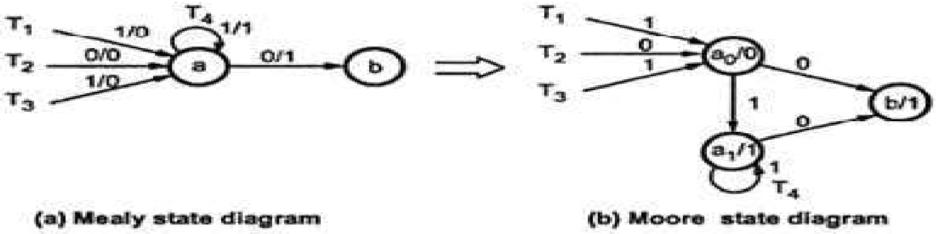
Q7a) Compare moore and mealy model
(or)
b)Difference between moore and mealy model

Sl no	Moore model	Mealy model
1.	**Its output is a function of present state only.**	**Its output is a function of present state as well as present input.**
2.	**Input changes does not affect the output.**	**Input changes may affect the output of the circuit.**
3.	**Moore model requires more number of states for implementing same function.**	**It requires less number of states for implementing same function.**

Conversion of Models

Rules to convert Mealy to Moore Model

If all the transitions in a Mealy model to a particular state are associated with only one type of output(either 0 or 1) then in corresponding Moore model that output becomes state output. This is illustrated in figure. Here T1, T2 and T3 are the paths leading to state a and corresponding outputs are all zeros. Thus we can say state a output is 0. When input is 1, state a leads to state b and there is no other input path. Thus we can say state b output is 0(the corresponding output of input path).

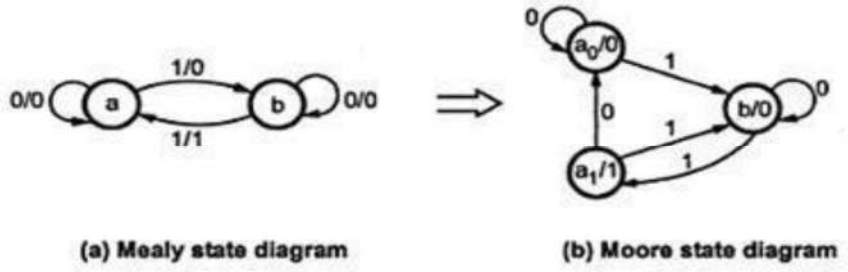


(a) Mealy state diagram

(b) Moore state diagram

Figure Rule1: Mealy to Moore conversion

2.If the outputs of all transitions in a Mealy model to a particular state are not same we need to insert intermediate state variables. This is illustrated in figure



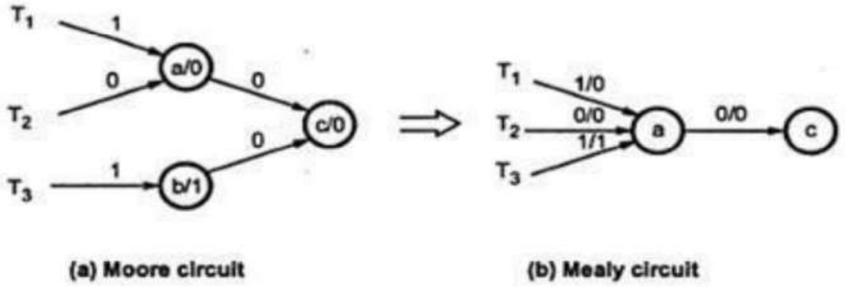
(a) Mealy state diagram

(b) Moore state diagram

Figure Rule 2: Mealy to Moore conversion

Rules to convert Moore to Mealy Model

1. **1.** If the state of transition from two different states of the same input leads to common state then one state can be eliminated. This is illustrated in figure



(a) Moore circuit

(b) Mealy circuit

Here, states a and b leads to common state c when input is 0. Therefore, we can eliminate state a or b.

Analysis of clocked sequential circuits

Q8a) Explain the Analysis of clocked sequential circuits

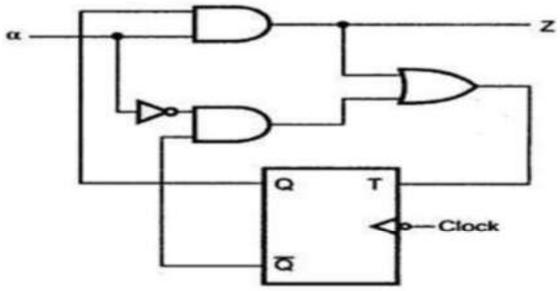
(or)

b) Steps to analyse the clocked sequential circuits

The behaviour of a sequential network is determined from the inputs, the outputs, and the states of its flipflops. Both the outputs and the next state are function of the inputs and the present state (in case of Moore circuits the outputs are function of only present state).

The analysis of sequential circuit consists of obtaining a table or a diagram for the time sequence of inputs, outputs and internal states. The success of analysis or design of sequential network depends largely on the aids and systematic techniques such as transition table, state table, state diagrams and state equations used in these processes.

Consider the sequential circuit to be analyzed as shown in figure



Let us see the steps to analyze the given synchronous sequential circuit

1. **Determine the flip-flop input equations and the output equations from the sequential circuit.**

$$Z = \alpha Q$$

$$T = \alpha Q + \bar{\alpha}\bar{Q}$$

2. **Derive the transition equation.**

The transition equation for T flip-flop is

$$Q^+ = T \oplus Q$$

$$Q^+ = (\alpha Q + \bar{\alpha}\bar{Q}) \oplus Q$$

3. **Plot the next step map for each flip-flop**

For Q^+

Q \ α	0	1
0	1	0
1	1	0

$$Q^+ = (\alpha Q + \bar{\alpha}\bar{Q}) \oplus Q$$

4.Plot the transition table

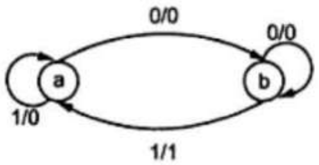
Present State	Next State		Output	
	α = 0	α = 1	α = 0	α = 1
Q	Q^+	Q^+	Z	Z
0	1	0	0	0
1	1	0	0	1

5. Draw the state table

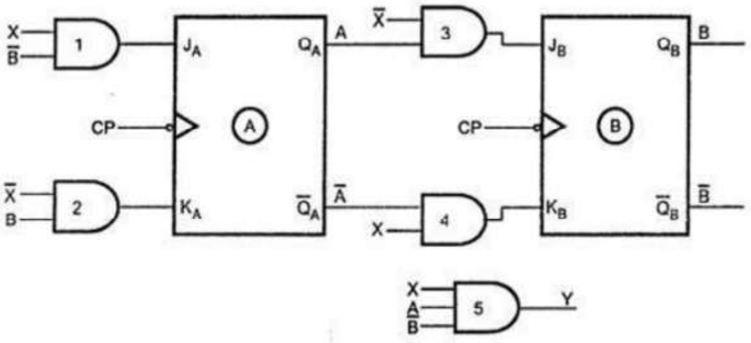
Here new symbols to binary codes are assigned. They are a=0, b=1

Present State	Next State		Output	
	α = 0	α = 1	α = 0	α = 1
a(0)	b	a	0	0
b(1)	b	a	0	1

6. Draw state diagram



Example: Construct the transition table, state table and state diagram for the Mealy sequential circuit given in figure



Solution : 1. Determine the flip-flop input equations and the output equations from the sequential circuit.

$$Y = XA\bar{B}$$

$$J_A = X\bar{B} \qquad K_A = \bar{X}B$$

$$J_B = \bar{X}A \qquad K_B = X\bar{A}$$

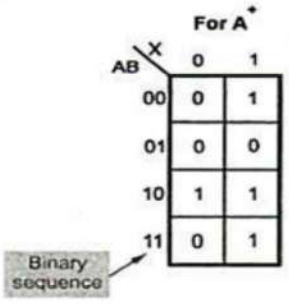
2. Derive the transition equations

The transition equations for JK flip-flops can be derived from the characteristic equation of JK flip-flop as follows:

We know that for JK flip-flop,

$$\begin{aligned}
Q^{+} &= J\bar{Q} + \bar{K}Q \\
\therefore\ A^{+} = Q_A^{+} &= J_A\bar{Q}_A + \bar{K}_A Q_A \\
&= X\bar{B}\bar{Q}_A + \overline{\bar{X}B}\,Q_A \\
&= X\bar{B}\bar{A} + (X+\bar{B})\,A \\
\text{and} \qquad B^{+} = Q_B^{+} &= J_B\bar{Q}_B + \bar{K}_B Q_B \\
&= \bar{X}A\bar{Q}_B + \overline{X\bar{A}}\,Q_B \\
&= \bar{X}A\bar{B} + (\bar{X}+A)\,B
\end{aligned}$$

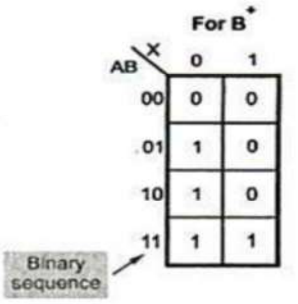
3. Plot a next-state maps for each flip-flop The next-state maps are



For A^{+}

AB \ X	0	1
00	0	1
01	0	0
10	1	1
11	0	1

$A^{+} = X\bar{B}\bar{A} + (X+\bar{B})A$



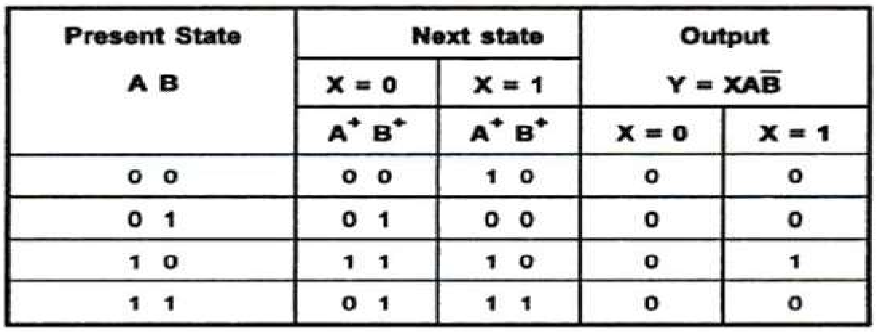
For B^{+}

AB \ X	0	1
00	0	0
01	1	0
10	1	0
11	1	1

$B^{+} = \bar{X}A\bar{B} + (\bar{X}+A)B$

4. Plot the transition table

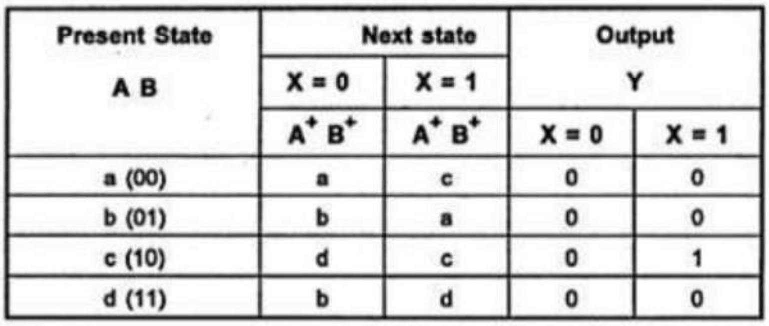
The transition table can be formed by combining the above two maps.



Present State	Next state		Output	
A B	X = 0	X = 1	$Y = XA\overline{B}$	
	$A^+ B^+$	$A^+ B^+$	X = 0	X = 1
0 0	0 0	1 0	0	0
0 1	0 1	0 0	0	0
1 0	1 1	1 0	0	1
1 1	0 1	1 1	0	0

5. Draw the state table

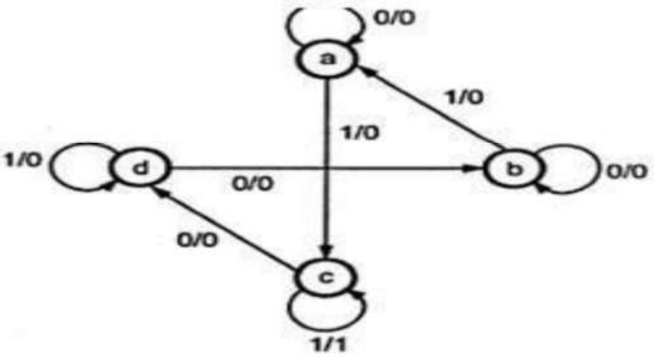
By assigning a = 00, b = 01, c = 10, d = 11 we can write state table from the transition table as shown below.



Present State	Next state		Output	
A B	X = 0	X = 1	Y	
	$A^+ B^+$	$A^+ B^+$	X = 0	X = 1
a (00)	a	c	0	0
b (01)	b	a	0	0
c (10)	d	c	0	1
d (11)	b	d	0	0

6. Draw the state diagram

From the state table we can draw the state diagram



Design of Clocked Sequential Circuits

Here, we will summarize the recommended steps for the design of a clocked sequential circuits.

1. It is necessary to first obtain the state table for the given circuit information such a state diagram, a timing-diagram, or other pertinent information.
2. The number of states may be reduced by state reduction technique if the sequential circuit can be categorized by input-output relationships independent of the number of states.
3. Assign binary values to each state in the state table.
4. Determine the number of flip-flops needed and assign a letter symbol to each.
5. Choose the type of flip-flop to be used.
6. From the state table, derive the circuit excitation and output tables.
7. Using the K-map or any other simplification method, derive the circuit output functions and flip-flop input functions.
8. Draw the logic diagram.

State Reduction

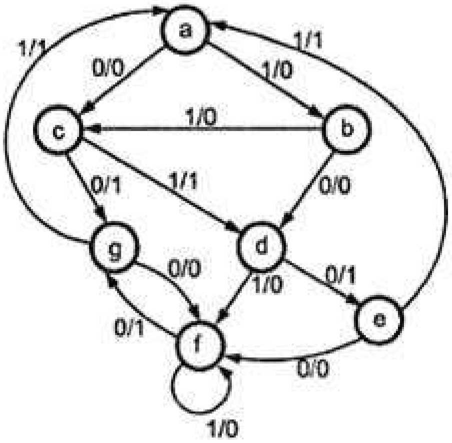
Q9. a) Explain in detail about state reduction techniques?

(or)

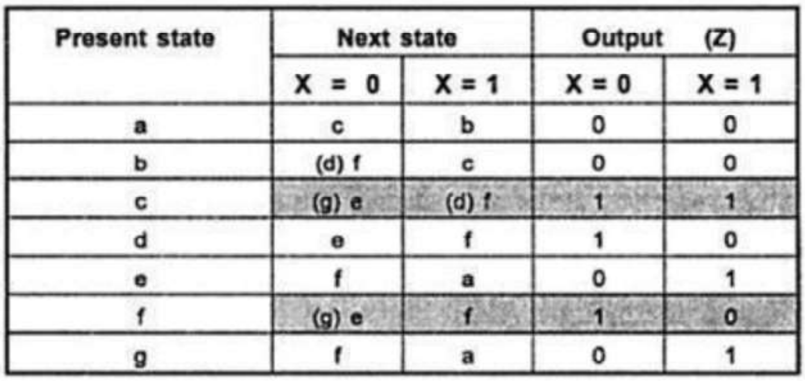
b) Design a clocked sequential circuit

The state reduction technique basically avoids the introduction of redundant states. The reduction in redundant states reduce the number of required flipflops and logic gates, reducing the cost of the final circuit. The two states are said to be redundant or equivalent, if every possible set of inputs generate exactly same output and same next state. When two states are equivalent, one of them can be removed without altering the input-output relationship.

Example 1 : Design a clocked sequential circuit for state diagram shown in the Fig.

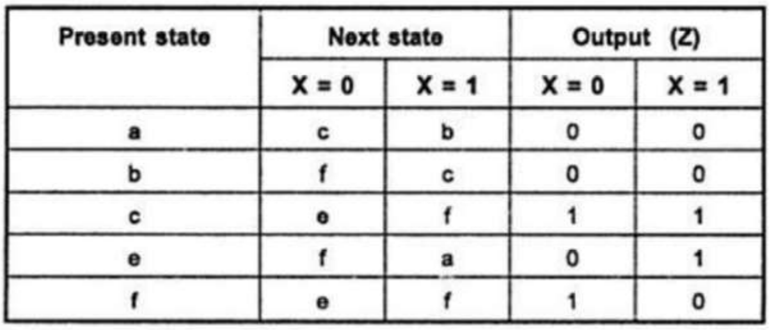


Solution: The state table for the sequential circuit will be as shown in Table 9.18 (a).



Present state	Next state		Output (Z)	
	X = 0	X = 1	X = 0	X = 1
a	c	b	0	0
b	(d) f	c	0	0
c	(g) e	(d) f	1	1
d	e	f	1	0
e	f	a	0	1
f	(g) e	f	1	0
g	f	a	0	1

As states e and g are eqivalent, we eliminate the state g as shown in the sate table. After replacing g by e, we can notice that the states d and f are eqivalent. Thus one of them say f can be eliminated. Then the reduced state table is as shown in Table



Present state	Next state		Output (Z)	
	X = 0	X = 1	X = 0	X = 1
a	c	b	0	0
b	f	c	0	0
c	e	f	1	1
e	f	a	0	1
f	e	f	1	0

No each state is assigned with binary values. Since there are five states, number of flip-flops required is 3 and 3-bit binary numbers are assigned to the states as shown below.

a = 000, b = 001, c = 010, e = 011, f = 100

If D flip-flops are used in design, the excitation Table 9.18 (c) is as given below.

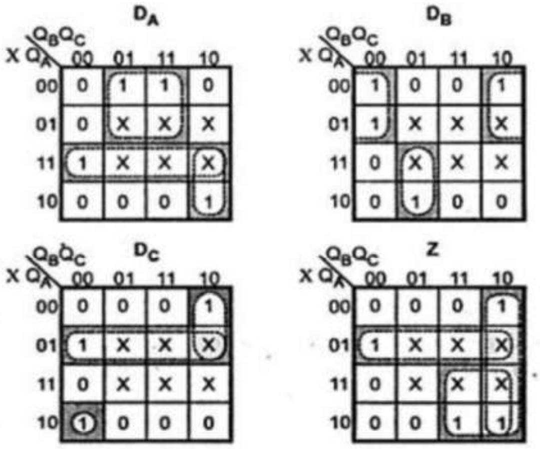
Input	Present	state		Next	state		Output
X	Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	Z
0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0
0	1	0	0	0	1	1	1
1	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	1	0	0	1
1	0	1	1	0	0	0	1
1	1	0	0	1	0	0	0

The flip-flop inputs DA, DB and DC are not included in the excitation table as they equal to the Next state.

$$D_A = Q_{A+1},\quad D_B = Q_{B+1} \quad \text{and} \quad D_C = Q_{C+1}$$

Mapping for D_A, D_B, D_C and Z (Assuming X, don't care, care condition for unused states).

K-map simplification



Hence the design equations are :

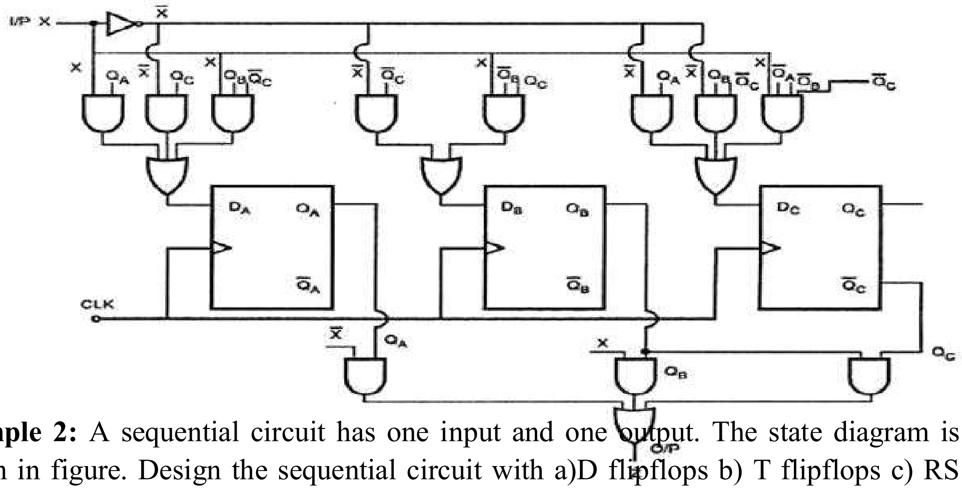
$$D_A = X\,Q_A + \bar{X}\,Q_C + X\,Q_B\bar{Q}_C$$

$$D_B = \bar{X}\,\bar{Q}_C + X\,\bar{Q}_B\,Q_C$$

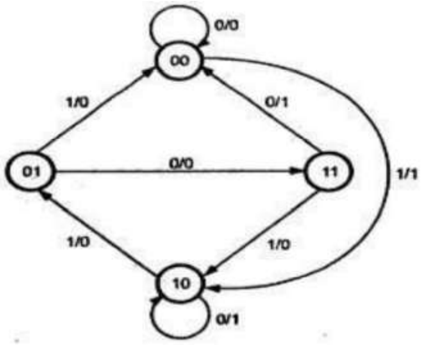
$$D_C = \bar{X}\,Q_A + \bar{X}\,Q_B\,\bar{Q}_C + X\,\bar{Q}_A\,\bar{Q}_B\,\bar{Q}_C$$

$$Z = \bar{X}\,Q_A + X\,Q_B + Q_B\,\bar{Q}_C$$

The circit diagram is as shown in Fig.



Example 2: A sequential circuit has one input and one output. The state diagram is shown in figure. Design the sequential circuit with a)D flipflops b) T flipflops c) RS flipflops d) JK flipflops



Solution: The state table for the state diagram

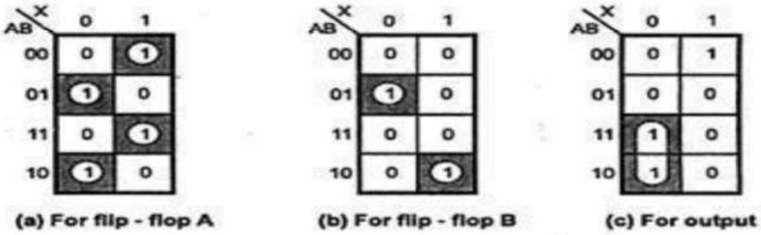
Present state		Next state		Output	
		X = 0	X = 1	X = 0	X = 1
A	B	AB	AB	Y	Y
0	0	0 0	1 0	0	1
0	1	1 1	0 0	0	0
1	0	1 0	0 1	1	0
1	1	0 0	1 0	1	0

As seen from the state table there is no equivalent states. Therefore, no reduction in the state diagram. The state table shows that circuit goes through four states, therefore we require flip-flops (number of states = 2^m, where m = number of flip-flops). Since two flip-flops are required first is denoted as **A** and second is denoted as **B**.

i) Design using D flip-flops

As mentioned earlier, for D flip-flops next states are nothing but the new present states. Thus, we can directly use next states to determine the flip-flop input with the help of K-map simplification.

K-map simplification



AB \ X	0	1
00	0	1
01	1	0
11	0	1
10	1	0

(a) For flip - flop A

AB \ X	0	1
00	0	0
01	1	0
11	0	0
10	0	1

(b) For flip - flop B

AB \ X	0	1
00	0	1
01	0	0
11	1	0
10	1	0

(c) For output

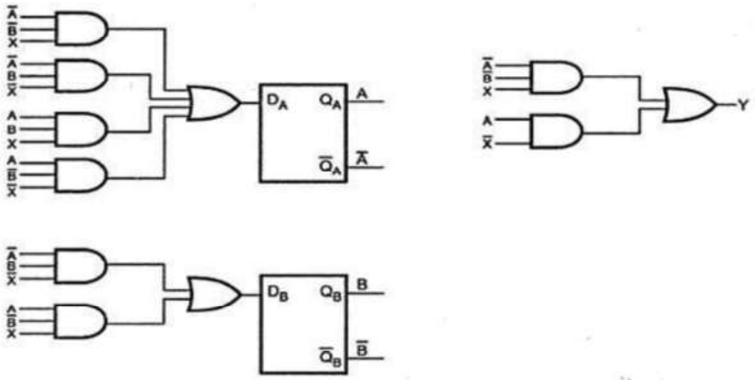
Fig. 9.15

$$D_A = \bar{A}\,\bar{B}\,X + \bar{A}\,B\,\bar{X} + ABX + A\,\bar{B}\,\bar{X} \quad \text{and}$$

$$D_B = \bar{A}\,B\,\bar{X} + A\,\bar{B}\,X$$

$$Y = \bar{A}\,\bar{B}\,X + A\,\bar{X}$$

With these flip-flop input functions and circuit output function we can draw the logic diagram as follows.

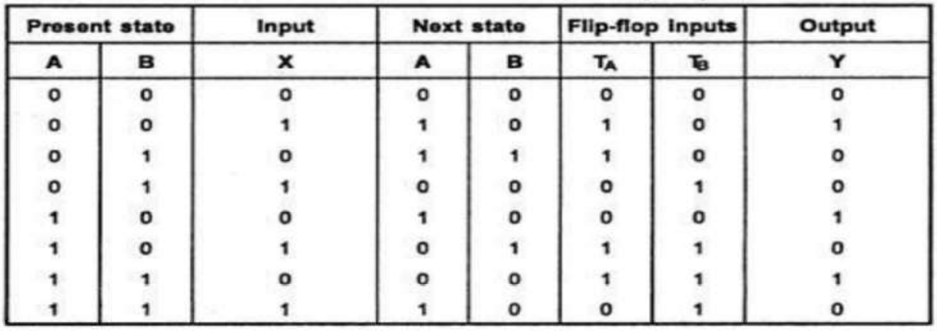


ii) Design using T flip-flops

a) Write the excitation table for T flipflop

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

b) Determine the excitation table for the given circuit

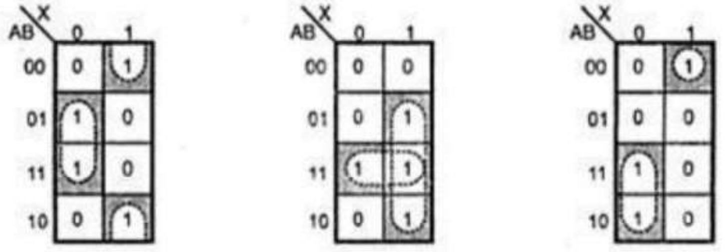


Present state		Input	Next state		Flip-flop inputs		Output
A	B	X	A	B	T_A	T_B	Y
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	1
0	1	0	1	1	1	0	0
0	1	1	0	0	0	1	0
1	0	0	1	0	0	0	1
1	0	1	0	1	1	1	0
1	1	0	0	0	1	1	1
1	1	1	1	0	0	1	0

In the first row of circuit excitation table shows that there is no change in the state for both flip-flops. The transition from 0 -> 0 for T flip-flop requires input T to be at logic 0. The second row shows the flip-flop A has transition 0 -> 1. It requires the input TA to be at logic 1. Similarly, we can find inputs for each flip-flop for each row in the table by referring present state, next state and excitation table.

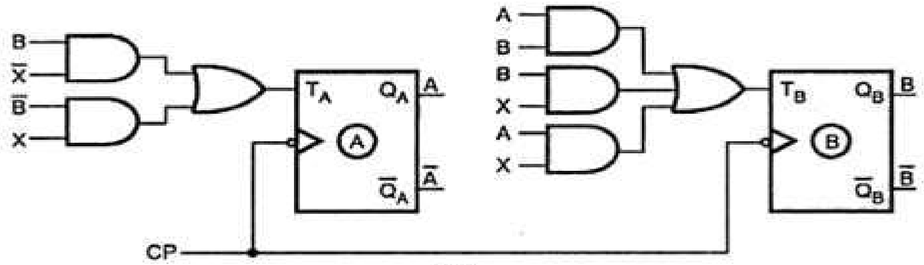
Let us use K-map simplification to determine the flip-flop input functions and circuit output functions.

K-map simplification

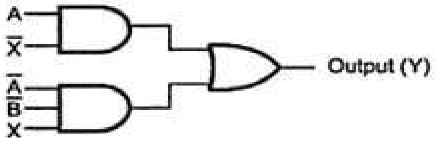


(a) For flip-flop A (b) For flip-flop B (c) For output

With these flip-flop input functions and circuit output function we can draw the logic diagram as follows.



(a)



iii) Design using RS flip-flops

a) Write the excitation table for RS flipflop

Q_n	Q_{n+1}	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

b) Determine the excitation table for the given circuit

Present state		Input	Next state		Flip-flop inputs				Output
A	B	X	A	B	R_A	S_A	R_B	S_B	Y
0	0	0	0	0	X	0	X	0	0
0	0	1	1	0	0	1	X	0	1
0	1	0	1	1	0	1	0	X	0
0	1	1	0	0	X	0	1	0	0
1	0	0	1	0	0	X	X	0	1
1	0	1	0	1	1	0	0	1	0
1	1	0	0	0	1	0	1	0	1
1	1	1	1	0	0	X	1	0	0

The first row of circuit excitation table shows that there is no change in the state for both flip-flops. The transition from 0 -> 0 for RS flip-flop requires inputs R and S to be X and 0, respectively. Similarly, we can determine the inputs for each flip-flop for each row in the table by referring present state, next state and excitation table. Let us use K-map simplification to determine the flip-flop input functions and circuit output functions.

K-map simplification

AB \ X	0	1
00	X	0
01	0	X
11	1	0
10	0	1

(a) For R_A

AB \ X	0	1
00	0	1
01	1	0
11	0	X
10	X	0

(b) For S_A

AB \ X	0	1
00	X	X
01	0	1
11	1	1
10	X	0

(c) For R_B

AB \ X	0	1
00	0	0
01	X	0
11	0	0
10	0	1

(d) For S_B

AB \ X	0	1
00	0	1
01	0	0
11	1	0
10	1	0

(e) For output

Therefore input function for $R_A = AB\bar{X} + A\bar{B}X$

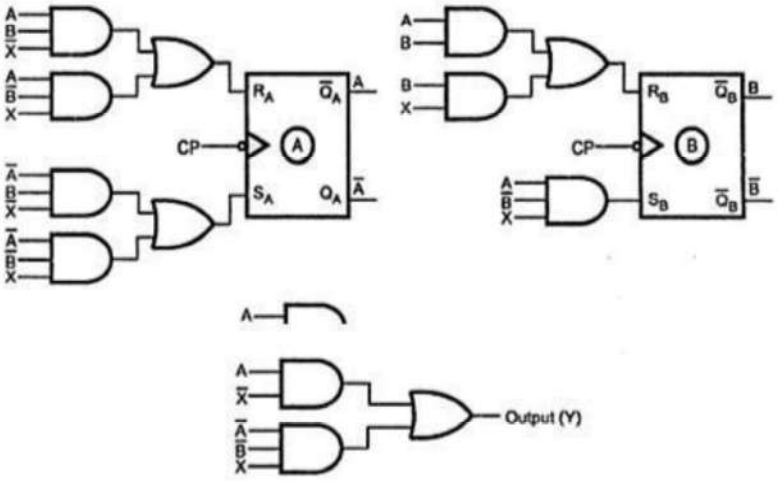
$$S_A = \bar{A}B\bar{X} + \bar{A}\bar{B}X$$

$$R_B = AB + BX$$

$$S_B = A\bar{B}X \text{ and,}$$

Circuit output function $= A\bar{X} + \bar{A}\bar{B}X$

With these flip-flop input functions and circuit output function we can draw the logic diagram as follows.



iv) Design using JK Flip-flops

a) Write the excitation table for JK flipflop

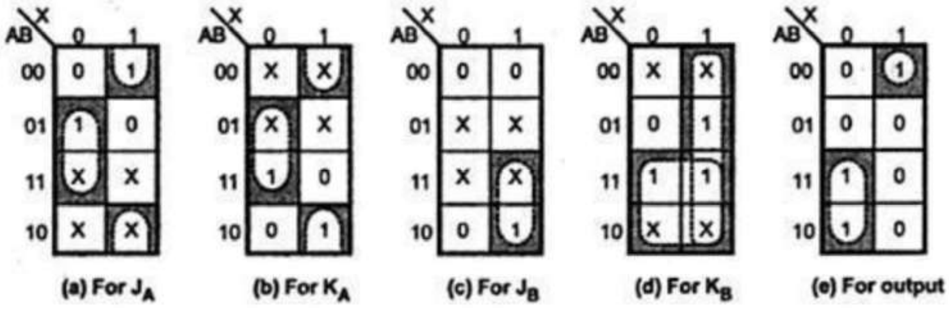
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

b) Determine the excitation table for the given circuit

Present state		Input	Next state		Flip-flop inputs				Output
A	B	X	A	B	J_A	K_A	J_B	K_B	Y
0	0	0	0	0	0	X	0	X	0
0	0	1	1	0	1	X	0	X	1
0	1	0	1	1	1	X	X	0	0
0	1	1	0	0	0	X	X	1	0
1	0	0	1	0	X	0	0	X	1
1	0	1	0	1	X	1	1	X	0
1	1	0	0	0	X	1	X	1	1
1	1	1	1	0	X	0	X	1	0

The first row of circuit excitation table shows that there is no change in the state for both flip-flops. The transition from 0 -> 0 for JK flip-flop requires inputs J and K to be 0 and X, respectively. Similarly, we can determine the inputs for each flip-flop for each row in the table referring present state, next state and excitation table. Let us use K-map. Simplification to determine the flip-flop input functions and circuit output functions.

K-map simplification


(a) For J_A

(b) For K_A

(c) For J_B

(d) For K_B

(e) For output

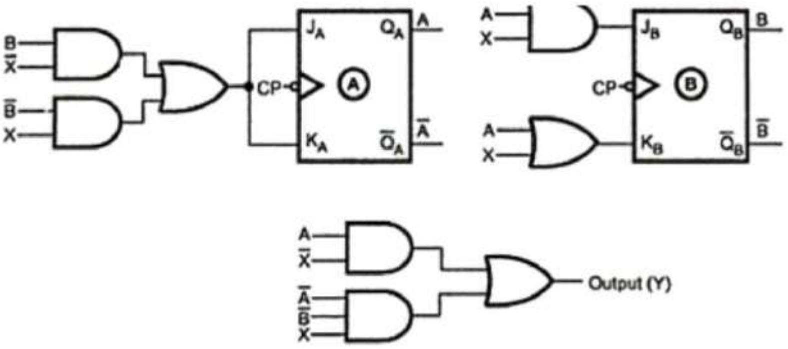
Therefore, input function for $J_A = B\bar{X} + \bar{B}X$

$K_A = B\bar{X} + \bar{B}X$

$J_B = AX$

$K_B = A + X$

Circuit output function $= A\bar{X} + \bar{A}\,\bar{B}X$



State Assignment

Q10a)Explain in detail about state assignment techniques?

(or)

b)Design a clocked sequential circuit

In sequential circuits is defined in terms of its inputs, present states, next state and outputs. To generate desired next state at particular present state and inputs, it is necessary to have specific flipflop inputs. These flipflop inputs are described by a set of boolean functions called flipflop input functions. To determine the flipflop input functions, it is necessary to represent states in the state diagram using binary values instead of alphabets. This procedure is known as state assignments. We must assign binary values to the states in such a way that it is possible to implement flipflop input functions using minimum logic gates.

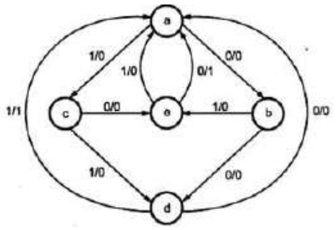
Rules for state assignments

There are two basic rules for making state assignments.

Rule 1: State having the same NEXT STATES for a given input condition should have assignments which can be grouped into logically adjacent cells in a K-map.

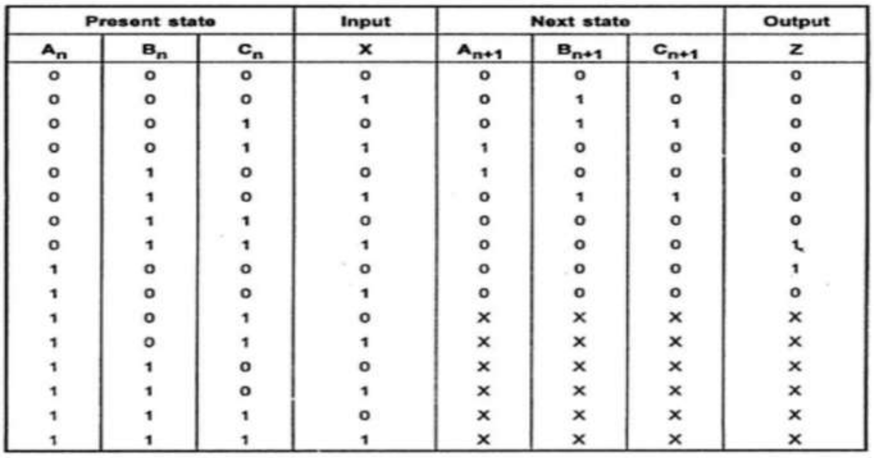
Rule 2: States that are the NEXT STATES of a single state should have assignment which can be grouped into logically adjacent cells in a K-map.

Example 1: Design a sequential circuit for a state diagram shown in figure. Use state assignment rules for assigning states and compare the required combinational circuit with random state assignment.



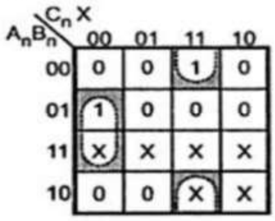
Using random state assignment we assign.

a = 000, b = 001, c = 010, d = 011 and e = 100. The excitation table with these assignments is as given inTable



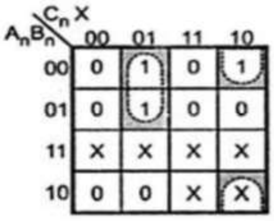
Present state			Input	Next state			Output
A_n	B_n	C_n	X	A_{n+1}	B_{n+1}	C_{n+1}	Z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	1	1	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

K-map simplification



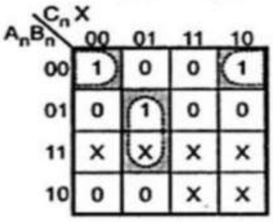
$A_nB_n \backslash C_nX$	00	01	11	10
00	0	0	1	0
01	1	0	0	0
11	X	X	X	X
10	0	0	X	X

$D_A = B_n\overline{C}_n\overline{X} + \overline{B}_nC_nX$



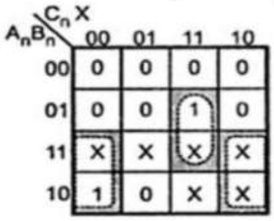
$A_nB_n \backslash C_nX$	00	01	11	10
00	0	1	0	1
01	0	1	0	0
11	X	X	X	X
10	0	0	X	X

$D_B = \overline{A}_n\overline{C}_nX + \overline{B}_nC_n\overline{X}$



$A_nB_n \backslash C_nX$	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	X	X	X	X
10	0	0	X	X

$D_C = \overline{A}_n\overline{B}_n\overline{X} + B_n\overline{C}_nX$



$A_nB_n \backslash C_nX$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	1	0	X	X

$Z = B_nC_nX + A_n\overline{X}$

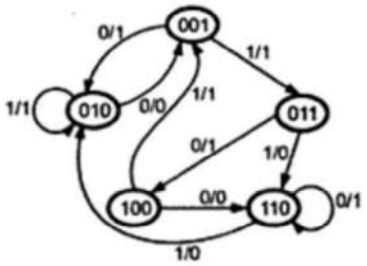
Design with unused state

There are occasions when a sequential circuit may use less than the available this maximum number of states. We can consider the unused states as don‘t care conditions and can be used to simplify the input functions to flip-flops.

Let us consider one example, First we will design the given sequential circuit without using unused states and then we will design the given sequential circuit using unused states.

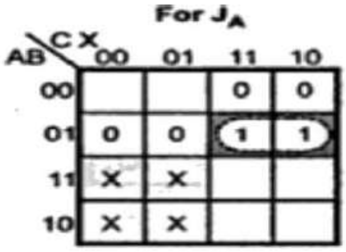
Example 2: Design the sequential circuit for the state diagram shown in figure. Use JK flip-flops

Solution: The state table for given state diagram is as follows



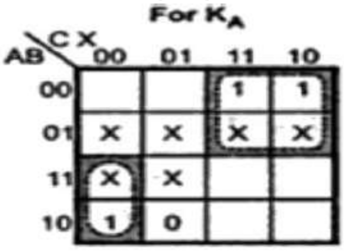
Present State			Input	Next State			Flip-flop inputs						Output
A	B	C	X	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C	Y
0	0	1	0	0	1	0	0	X	1	X	X	1	1
0	0	1	1	0	1	1	0	X	1	X	X	0	1
0	1	0	0	0	0	1	0	X	X	1	1	X	0
0	1	0	1	0	1	0	0	X	X	0	0	X	1
0	1	1	0	1	0	0	1	X	X	1	X	1	1
0	1	1	1	1	1	0	1	X	X	0	X	1	0
1	0	0	0	1	1	0	X	0	1	X	0	X	0
1	0	0	1	0	0	1	X	1	0	X	1	X	1
1	1	0	0	1	1	0	X	0	X	0	0	X	1
1	1	0	1	0	1	0	X	1	X	0	0	X	0

For J_A



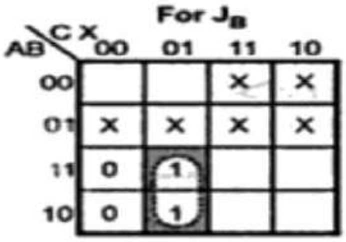
AB \ CX	00	01	11	10
00			0	0
01	0	0	1	1
11	X	X		
10	X	X		

For K_A



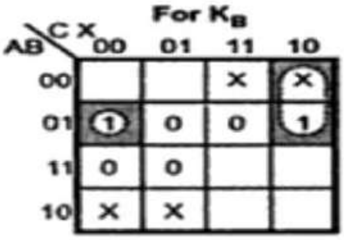
AB \ CX	00	01	11	10
00			1	1
01	X	X	X	X
11	X	X		
10	1	0		

For J_B



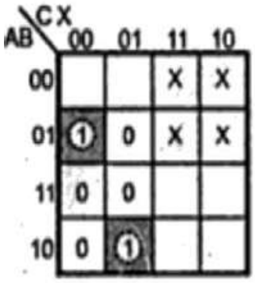
AB \ CX	00	01	11	10
00			X	X
01	X	X	X	X
11	0	1		
10	0	1		

For K_B



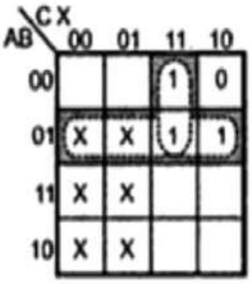
AB \ CX	00	01	11	10
00			X	X
01	1	0	0	1
11	0	0		
10	X	X		

For J_C



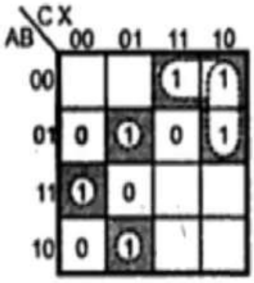
AB \ CX	00	01	11	10
00			X	X
01	1	0	X	X
11	0	0		
10	0	1		

For K_C



AB \ CX	00	01	11	10
00			1	0
01	X	X	1	1
11	X	X		
10	X	X		

For Output



AB \ CX	00	01	11	10
00			1	1
01	0	1	0	1
11	1	0		
10	0	1		

Therefore, input functions for

$$J_A = \overline{A}BC$$

$$K_A = A\overline{CX} + \overline{A}C$$

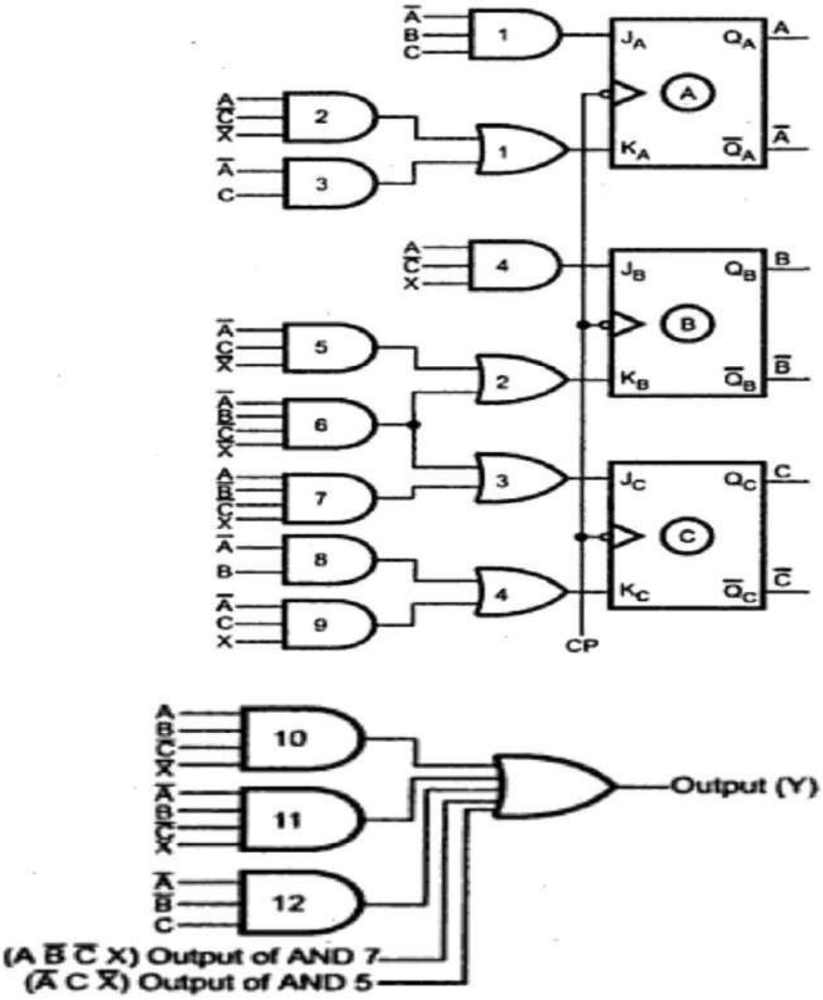
$$J_B = A\overline{C}X$$

$$K_B = \overline{A}B\overline{C}\,\overline{X} + \overline{A}C\overline{X}$$

$$J_C = \overline{A}B\overline{C}\,\overline{X} + A\overline{B}\,\overline{C}X$$

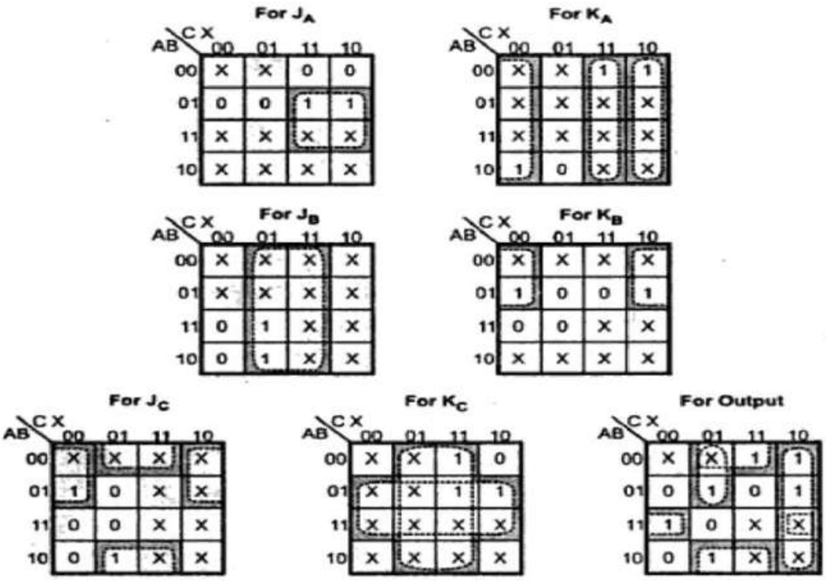
$$K_C = \overline{A}B + \overline{A}CX \text{ and}$$

$$\text{Circuit output function, } Y = AB\,\overline{C}\;\overline{X} + \overline{A}\,B\overline{C} + A\overline{B}\,\overline{C}X + \overline{A}\,\overline{B}C + \overline{A}C\overline{X}$$



Let us see the circuit design with the use of unused states. These unused states 000,101 and 111 are considered as a don't cares and are used to simplify the kmaps as follows:

K-map simplification



For J_A

AB \ CX	00	01	11	10
00	X	X	0	0
01	0	0	1	1
11	X	X	X	X
10	X	X	X	X

For K_A

AB \ CX	00	01	11	10
00	X	X	1	1
01	X	X	X	X
11	X	X	X	X
10	1	0	X	X

For J_B

AB \ CX	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	0	1	X	X
10	0	1	X	X

For K_B

AB \ CX	00	01	11	10
00	X	X	X	X
01	1	0	0	1
11	0	0	X	X
10	X	X	X	X

For J_C

AB \ CX	00	01	11	10
00	X	X	X	X
01	1	0	X	X
11	0	0	X	X
10	0	1	X	X

For K_C

AB \ CX	00	01	11	10
00	X	X	1	0
01	X	X	1	1
11	X	X	X	X
10	X	X	X	X

For Output

AB \ CX	00	01	11	10
00	X	X	1	1
01	0	1	0	1
11	1	0	X	X
10	0	1	X	X

Therefore, input functions for

$$J_A = BC$$

$$K_A = \overline{X} + C$$

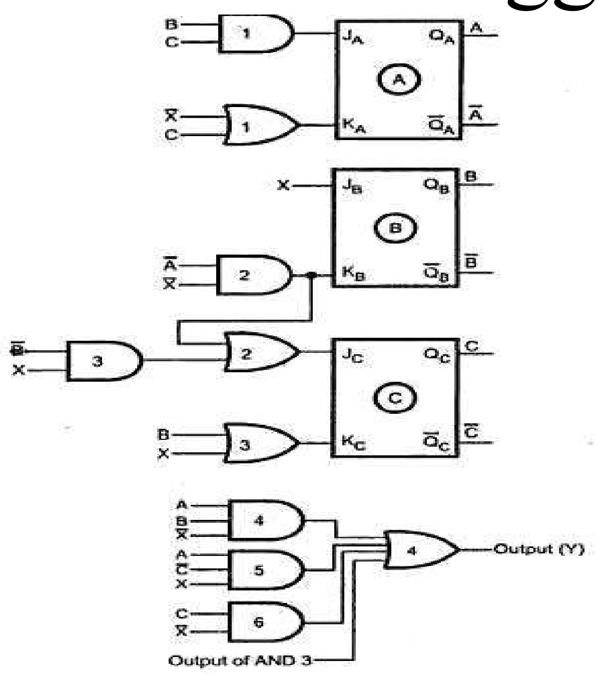
$$J_B = X$$

$$K_B = \overline{A}\,\overline{X}$$

$$J_C = \overline{A}\,\overline{X} + \overline{B}X \quad \text{and}$$

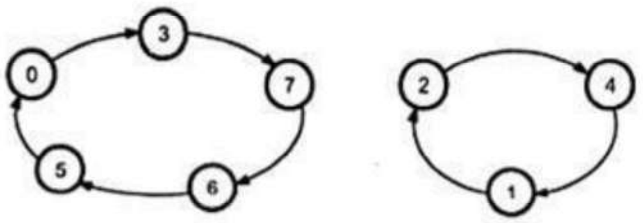
$$K_C = B + X$$

The circuit output function $Y = AB\overline{X} + A\overline{C}X + \overline{B}X + C\overline{X}$



Lockout Conditions

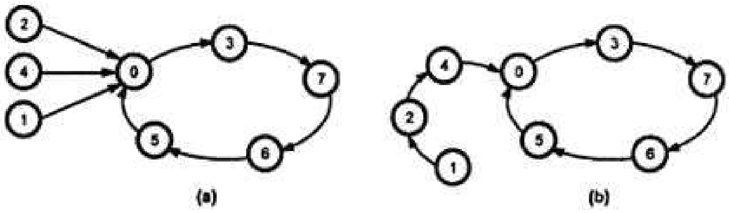
In a counter if the next state of some unused state is again an unused state and if by chance the counter happens to find itself in the unused states and never arrived at a used state then the counter is said to be in lockout conditions. This is illustrated in the Fig.



(
a
)

Desired sequence (b) Unused state forming lockout

The circuit that goes in lockout condition is called bushless circuit. To make sure that the counter will come to the initial state from any unused state, the additional logic circuit is necessary. To ensure that the lockout does not occur, the counter should be designed by forcing the next state to be the initial state from the unused states as shown in figure.

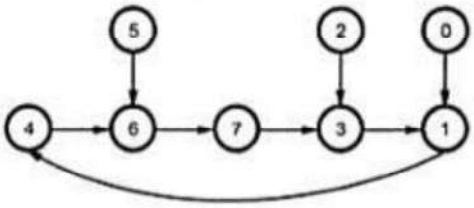


For example, as shown in Fig, actually it is not necessary to force all unused states into initial state. Forcing any one state is sufficient. Because, if counter initially goes to unused state which is not forced, it will go to another unused state. This will continue until it reaches the forced unused state. Once forced unused state is reached next state is used state, and circuit is lock free circuit. This is illustrated in Fig

Examle 3:*Design a synchronous counter for*

4 → 6 → 7 → 3 → 1 → 4 ...

Avoid lockout condition. Use JK type design.



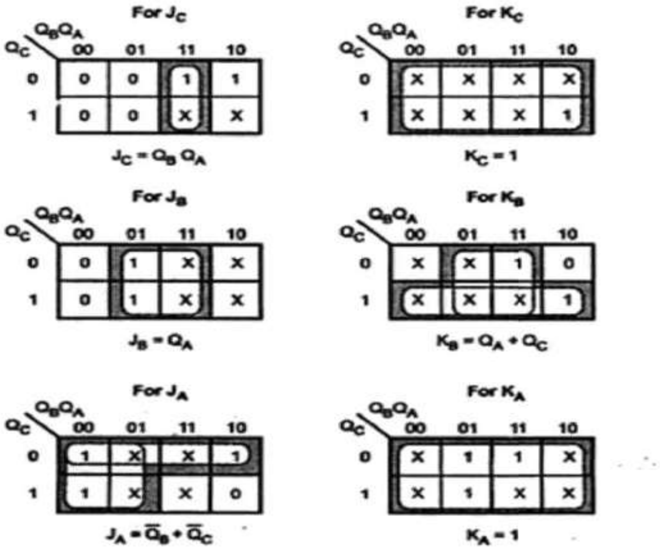
Solution : State diagram

Here, states 5,2 and 0 are forced to go into 6,3 and 1 state, respectively to avoid lockout condition.

Excitation table

Present states			Next states			Flip-flop inputs					
A	B	C	A_{+1}	B_{+1}	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	1	0	0	1	X	0	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	X	X	1	X	0
1	0	0	1	1	0	X	0	1	X	0	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	1	1	X	1	X	0	X	0

K-map simplification

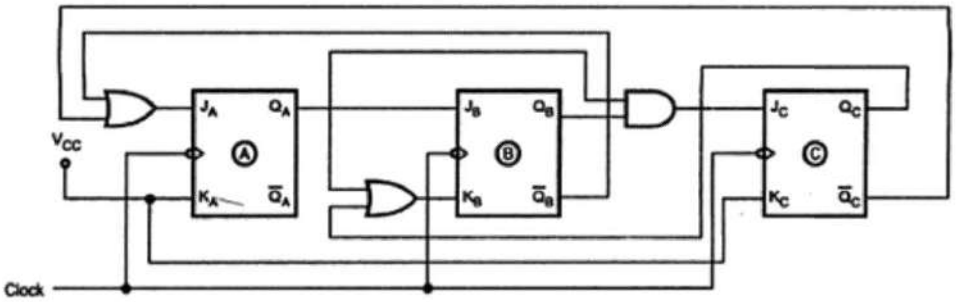


$J_C = Q_B Q_A$ $K_C = 1$

$J_B = Q_A$ $K_B = Q_A + Q_C$

$J_A = \bar{Q}_B + \bar{Q}_C$ $K_A = 1$

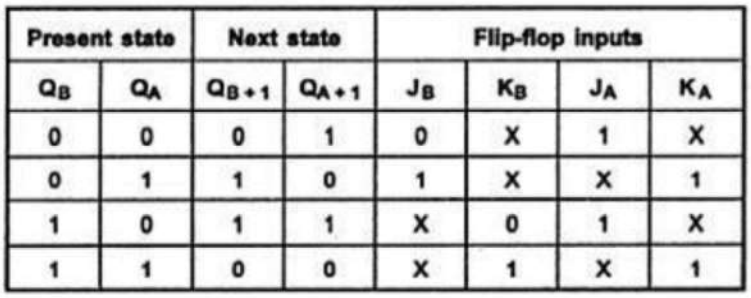
Logic diagram



Example4 : Design a synchronous conter with states 0,1,2,3,0,1……..using JK FFs.

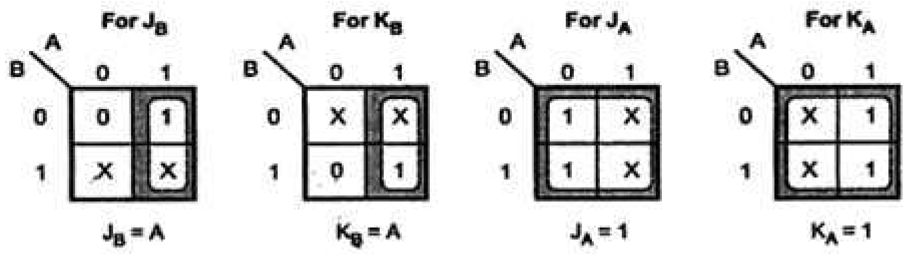
Solution:

Transition table

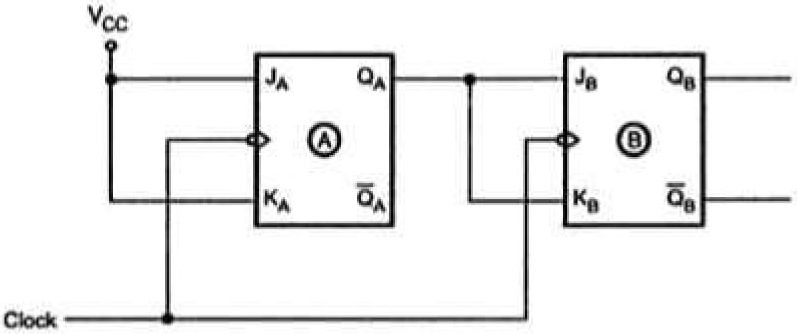


Present state		Next state		Flip-flop inputs			
Q_B	Q_A	Q_{B+1}	Q_{A+1}	J_B	K_B	J_A	K_A
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

K-map simplification



Logic diagram



2marks

1. **What are the classifications of sequential circuits?**
   (or)
   Write the classifications of sequential circuits?

   The sequential circuits are classified on the basis of timing of their signals into two types. They are:

   i. Synchronous sequential circuit.
   ii. Asynchronous sequential circuit.

2. **Define Flip flop.**
   (or)
   What is flipflop
   The basic unit for storage is flip flop. A flip-flop maintains its output state either at 1 or 0 until directed by an input signal to change its state.

3. **What are the different types of flip-flop?**
   (or)
   Write the different types of flip-flop?
   There are various types of flip flops. Some of them are mentioned below they are:

   a. SR flip-flop
   b. D flip-flop