Unit 7 Basics of Counters & Design of Counters

Structure:

7.1 Introduction

In the previous unit we studied about shift registers and their types. We also studied some ICs in relation to shift registers. In electronics, a counter is a logic device which is capable of counting the number of times an event has occurred. It can also display the count. The counter for its operation needs a trigger or prompt which conveys that an event has occurred. This trigger is usually supplied to a counter in the form of a signal called 'clock'. Depending on whether the count value increases/decreases, we have the following basic types of counters.

- Up counters - Increase count value
- Down counters - Decrease count value

Counters are generally implemented using basic storage elements such as flip-flops, latches or registers. There are many types of counters with different designs. They are also used in different applications.

Objectives:

By the end of Unit 7, the learners should be able to:

- explain the function of counters
- differentiate asynchronous and synchronous counters
- explain Johnson and ring counter
- explain the design of Modulo-N counters
- solve problems on counter design

7.2 Typical Counters

Counters are digital logic devices and as we know, widely used digital logic is binary in nature. So binary counters count in terms of binary numbers. They can even count BCD (Binary Coded Decimal) numbers. There are many commercially available counter ICs that can be used in electronic circuit/system design. For example, CMOS IC's in the 4XXX series implement different counters.

There are different types of counters which can count binary numbers. Counters can be classified based on how the clock is applied, number of flip-flops and sequential states. Some of the counter types are listed below.

- Asynchronous (ripple) counters
- Synchronous counters
- Up-Down counters
- Johnson counters
- Decade counters

In asynchronous or ripple counters, all the flip-flops are not clocked by the same clock and all flip-flops do not change their state in exact synchronism with the applied clock pulses. It is because, usually the clock pulse is applied to the first flip-flop and the output of the first flip flop is connected as a clock for the next flip-flop. Asynchronous counters are also called ripple counters because the flip-flops change their state in a ripple fashion i.e.the clock pulse fed into first flip-flop ripples through the other counters after propagation delays, like a ripple on water, hence the name Ripple Counter.

In synchronous counters, all the flip-flops are clocked by the same clock and all flip-flops change their state in exact synchronism with the applied clock pulses. So a common clock is connected to all the clock inputs of all the flip flops. The delay involved is equal to the propagation delay of one flip-flop only, irrespective of the number of flip-flops used to construct the counter.

An up counter is one which counts in upward direction in ascending order. On the other hand, a down counter counts in downward direction in descending order.

A counter which counts in both up and down directions is called an UP/DOWN counter.

A Johnson counter is basically a circulating shift register in which the output of the most significant stage is fed back to the input of the least significant stage.

The simplest counter circuits can be built using T flip-flops because the toggle feature is suited for the implementation of the counting operation.

Decade Counter is a type of counters which counts only from 0 through 9. Hence the modulus of decade counter is 10. This decade counter requires four flip-flops for its construction.

Modulus of a counter

Modulus of a counter is the number of different states that a counter can go through before it comes back to initial state to repeat the count sequence. An n-bit counter that counts through all its natural states and does not skip any of the states has a modulus of 2^n. For example, a 3 stage counter consists of cascade of 3 flip flops and can count maximum of eight counts. Hence the Modulus of this counter is 8. If a counter counts all the maximum possible counts, then it is called natural count and in this case the modulus of a counter is equal to natural count. Note that a counter with three flip flops can count maximum of 9 counts from 0 through 8 and suppose if it is designed to count only from 0 through 5, then modulus of the counter is 6. So it is called Modulo-6 counter. In this case the modulus of a counter is not equal to natural count.

Asynchronous Counters:

A counter may count up or count down or count up and down depending on the input control. The count sequence usually repeats itself. For example, for a 4- bit counter when counting up, the count sequence goes from 0000, 0001, 0010, ... 1110 , 1111 , 0000, 0001, ... etc. When counting down the count sequence goes in the opposite manner: 1111, 1110, ... 0010, 0001, 0000, 1111, 1110, ... etc.

3- Bit asynchronous up counter

The figure 7.1 shows a 3 bit asynchronous up counter along with timing diagram.

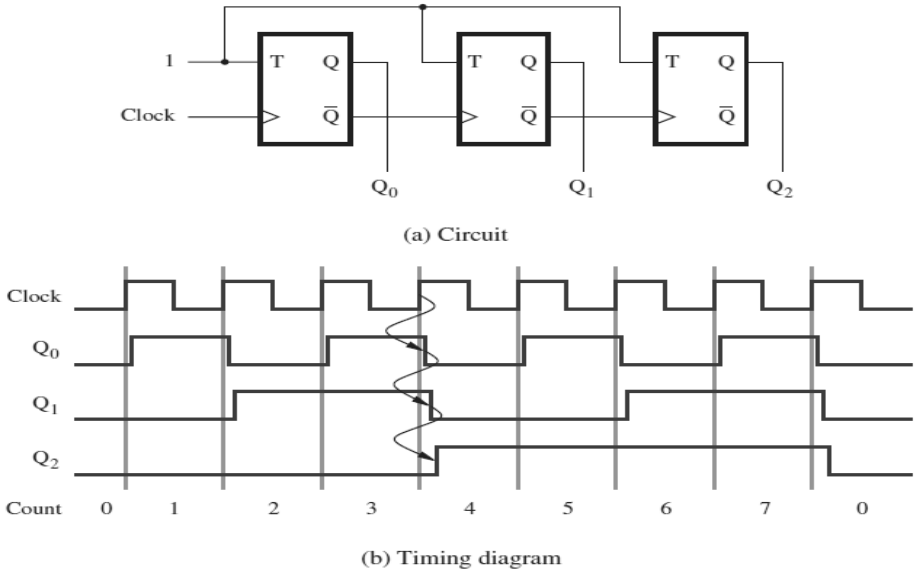


Figure 7.1: 3 –Bit Asynchronous Up Counter

(a) Circuit diagram

(b) Timing diagram

The table 7.1 shows the truth table of 3-Bit asynchronous Up counters

Table 7.1: Truth table of 3-Bit Asynchronous Up Counter

Clock Cycle	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	0	0	0
.		.	
.		.	
.		.	

Figure 7.1(a) shows a 3-bit counter implemented using T-flip flops. This counter can count from 0 to 7 in upward direction (i.e. Count Increases in ascending order). The *T* input of each flip-flop is 1(High), so that state of each flip-flop will be toggled at each positive edge of its clock. The clock input of the first flip-flop is connected to the *Clock* line and other two flip-flops recieve their clock inputs driven by the Q output of the preceding flip-flop. Therefore, they also toggle their state whenever the preceding flip-flop changes its state from Q = 1 to Q = 0, which results in a positive edge of the Q signal.

Figure 7.1(b) shows a timing diagram for the counter. The value of Q0 toggles once after each positive edge of the *Clock* signal. The second flip-flop is clocked by Q0, the value of Q1 changes after the negative edge of the Q0 signal. Similarly, Q2 changes its state after the negative edge of the Q1 signal. This is also evident from the truth table shown in table 7.1.

If we look at the values $Q_2Q_1Q_0$ in truth table, counting is in ascending order in the sequence 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, and so on. This is a modulo-8 counter. Since it counts in the upward direction, it is called an *up-counter*.

3-Bit asynchronous Down counter

The figure 7.2 shows a 3-bit asynchronous Down counter along with timing diagram and truth table. The difference between figure 7.1(a) and figure 7.2(a) is that only the clock inputs of the second and third flip-flops are driven by the Q outputs of the preceding flip-flops, instead of the $\overline{Q}$ outputs. The timing diagram shown in 7.2(b) shows how the counter counts during clock pulse. The truth table shown in table 7.2 clearly indicates the counter counts in down ward direction and the sequence is 7, 6, 5, 4, 3, 2, 1, 0, 7, and so on. Since the counter counts in the downward direction, it is called *down-counter*.

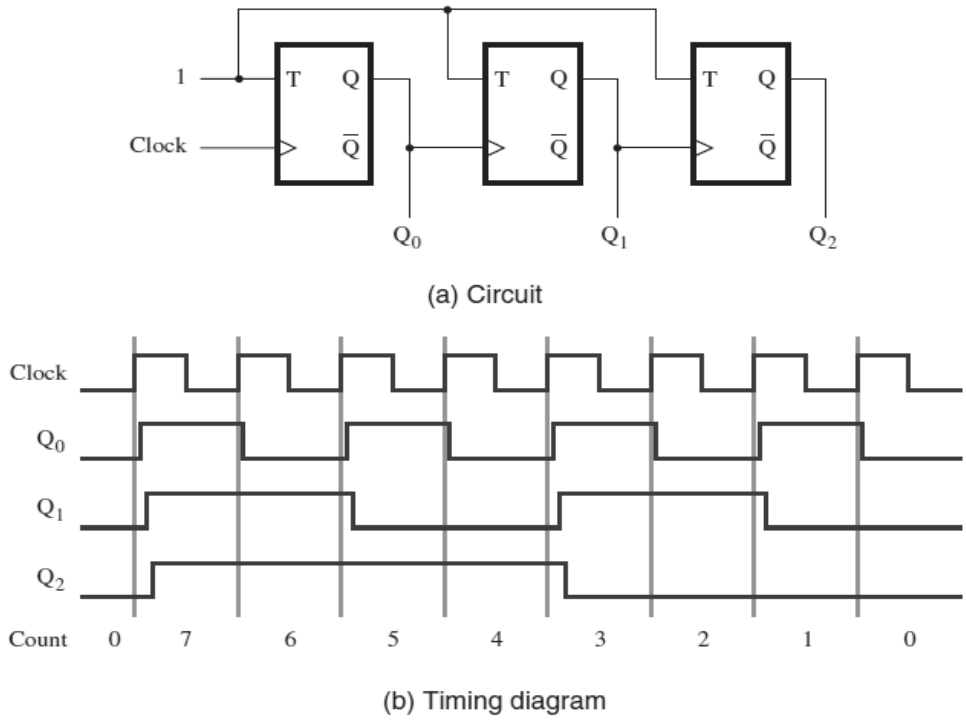


Figure 7.2: 3 –Bit asynchronous Up Counter

(a) Circuit diagram
(b) Timing diagram

Table 7.2: Truth Table of a 3-bit Asynchronous Down Counter

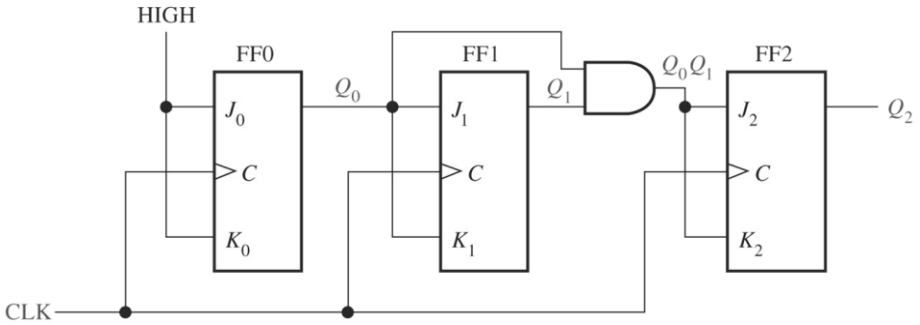
Clock Cycle	Q2	Q1	Q0
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0
6	0	0	1
7	0	0	0
8	1	1	1
.		.	
.		.	
.		.	

Synchronous Counters:

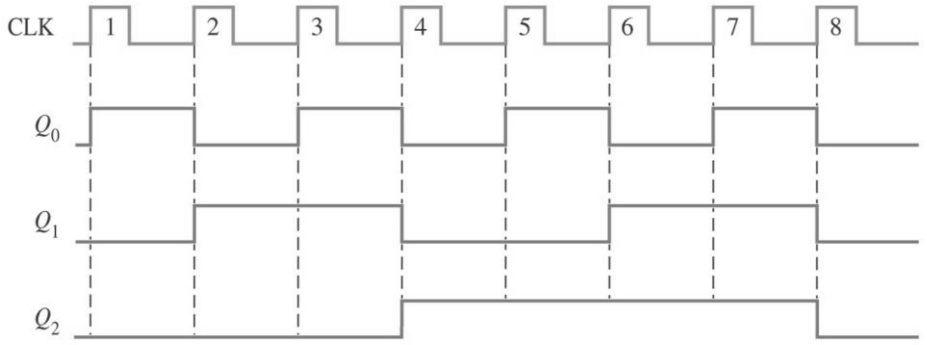
The asynchronous counters above are simple but not very fast. If a counter with a larger number of bits is constructed in this manner, then the delays caused by the cascaded clocking scheme may become too long to meet the desired performance requirements. We can build a faster counter by clocking all flip-flops at the same time as presented below.

Synchronous Up-Counter with T Flip-Flops

An example of a 3-bit synchronous up-counter is shown in figure 7.3(a) and the timing diagram is shown in figure 7.3(b).



(a)



(b)

Figure 7.3: 3 –Bit Synchronous Up Counter

(a) Logic circuit diagram
(b) Timing diagram

The table 7.3 shows the truth table of a 3-Bit Synchronous Binary Counter.

Table: 7.3: Truth table of a 3-Bit Synchronous Binary Counter

Clock Pulse	Q2	Q1	Q0
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

4-bit synchronous up-counter

A 4-bit synchronous up-counter using T flip-flop is shown in figure 7.4.

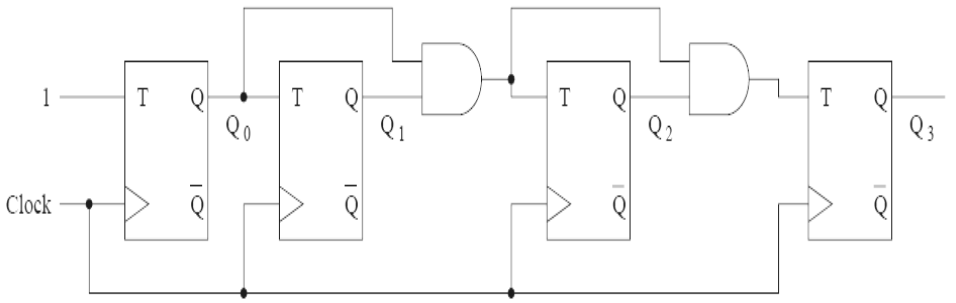
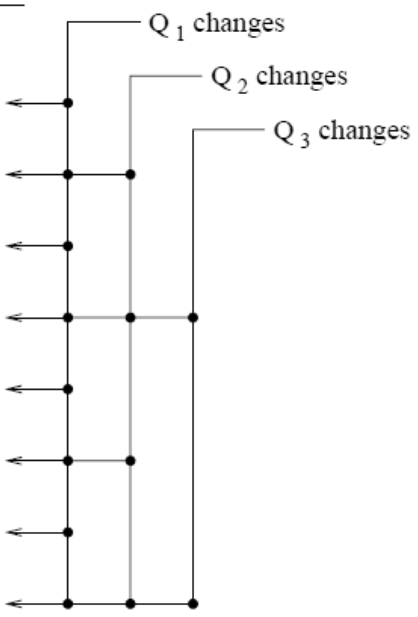


Figure 7.4: 4-bit synchronous up-counter

The table 7.4 shows the contents of such a 4-bit up-counter for sixteen consecutive clock cycles, assuming that the counter is initially 0.

Table 7.4: Contents of a 4−bit up−counter for 16 consecutive clock cycles

Clock Cycle	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0



Observing the pattern of bits in each row of the table, it is apparent that bit $Q0$ changes on each clock cycle. Bit $Q1$ changes only when $Q0 = 1$. Bit $Q2$ changes only when both $Q1$ and $Q0$ are equal to 1. Bit $Q3$ changes only when $Q2 = Q1 = Q0 = 1$. In general, for an n-bit up-counter, a give flip-flop changes its state only when all the preceding flip-flops are in the state $Q = 1$. Therefore, if we use T flip-flops to realize the 4-bit counter, then the T inputs should be defined as

$T0 = 1$

$T1 = Q0$

$T2 = Q0Q1$

$T3 = Q0Q1Q2$

In figure 7.4, instead of using AND gates of increased size for each stage, we use a factored arrangement. This arrangement does not slow down the response of the counter, because all flip-flops change their states after a propagation delay from the positive edge of the clock. Note that a change in the value of $Q0$ may have to propagate through several AND gates to reach the flip-flops in the higher stages of the counter, which requires a certain

amount of time. This time must not exceed the clock period. Actually, it must be less than the clock period minus the setup time of the flip-flops.

Figure 7.5 shows an example timing diagram for the 4-bit synchronous up-counter. It shows that the circuit behaves as a modulo-16 up-counter. Because all changes take place with the same delay after the active edge of the *Clock* signal, the circuit is called a *synchronous counter*.

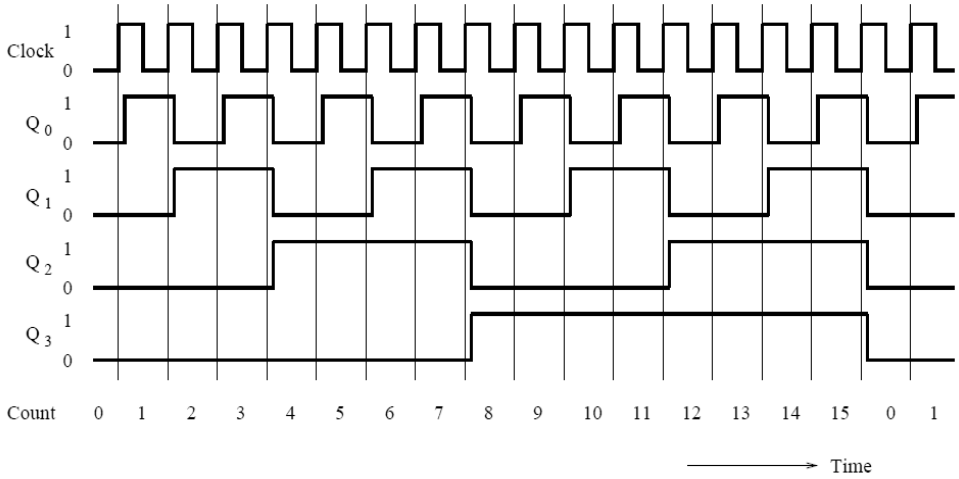


Figure 7.5: Timing diagram for a 4−bit synchronous up−counter

Now let see Johnson ring counter and ring counter.

7.2.1 Johnson Counter

A Johnson counter or Johnson Ring Counter or Twisted Ring Counter is constructed using serial-in and serial-out (SISO) shift register. Here the output of the last shift register is connected back to the input of the first after passing it through an inverter or the inverted output $\overline{Q}$ of the last flip-flop is connected back to the input of the first flip. Depending on the initial bit pattern stored in the shift register, the shift register content changes for every clock pulse and the bit pattern gets repeated after 2n clocks, where n is the number of bits in the shift register. These counters are also called “walking ring” counters and have applications like digital-to-analog converters (DAC) etc. The figure 7.6 shows the 4 bit Johnson Ring Counter using D flip-flop.

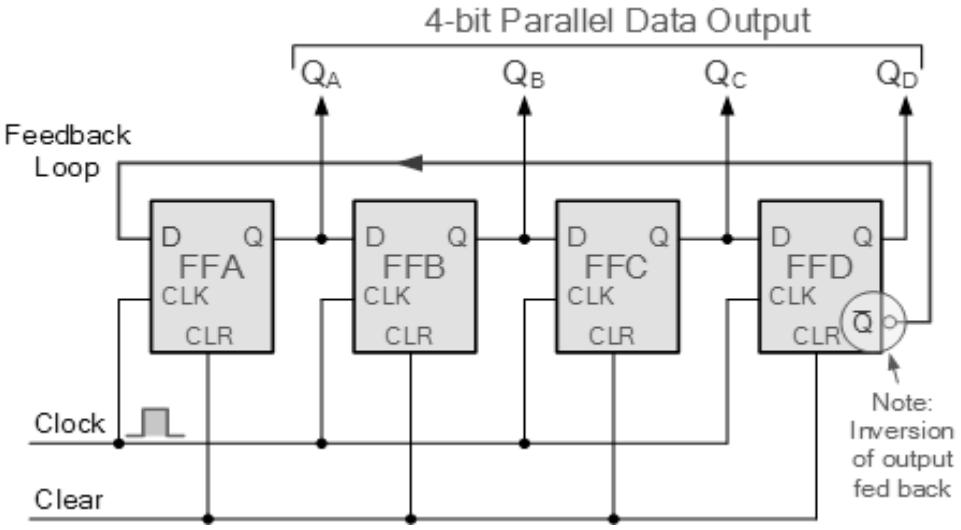


Figure 7.6: 4-bit Johnson Ring Counter

Operation:

Initially all the flip-flops are cleared to zero (0) by making asynchronous input Clear =1 so that the initial value of the outputs Q_A Q_B Q_C Q_D =0000. Now make the clear input=0 so that it becomes inactive. Now the operation of circuit depends only on the clock input. Also note that inverted output is $\overline{Q}$=1. Since the inverted output $\overline{Q}$ is connected to the input of the first flip flop, the input D=1. Now when 1st clock pulse is applied, shifting operation takes place and hence after the 1st clock pulse, Q_A Q_B Q_C Q_D =1000. The complete bit patterns of this circuit for the subsequent clock pulses is 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000... . and is shown in the table 7.5.

Table 7.5: Truth Table for a 4-bit Johnson Ring Counter

Clock Pulse No.	Q_A	Q_B	Q_C	Q_D
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
0	0	0	0	0

As the inverted output Q is connected to the input D, 8-bit pattern continually repeats. For example, “1000″, “1100″, “1110″, “1111″, “0111″, “0011″, “0001″, “0000″ and this is demonstrated in the table 7.5. The figure 7.7 shows the timing diagram (waveform) of a 4 bit Johnson counter.

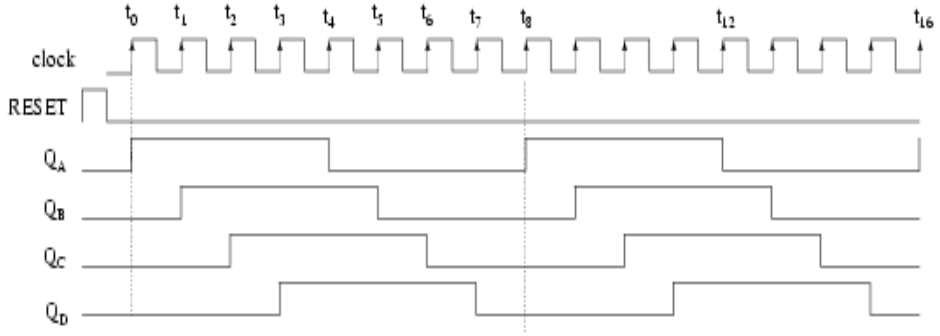


Figure 7.7: Waveform of a 4 bit Johnson Counter

From the timing diagram shown in figure 7.7, it is clear that after the 8th pulse t_8 (i.e. after second 0th pulse in the truth table 7.5), the sequence repeats again with the initial value Q_A Q_B Q_C Q_D =0000.,

Apart from counting or rotating data around a continuous loop, ring counters can also be used to detect or recognise various patterns or number values within a set of data. By connecting simple logic gates such as the AND or the OR gates to the outputs of the flip-flops the circuit can be made to detect a set of number or value.

Standard 2, 3 or 4-stage Johnson Ring Counters can also be used to divide the frequency of the clock signal by varying their feedback connections and divide-by-3 or divide-by-5 outputs are also available. For example, a 3-stage Johnson Ring Counter could be used as a 3-phase, 120 degree phase shift square wave generator by connecting to the data outputs at A, B and NOT-B. The standard 5-stage Johnson counter such as the commonly available CD4017 is generally used as a synchronous decade counter/divider circuit.

7.2.2 Ring Counter

A ring counter is a circulating shift register in which the output the last flip flop is fed back to input of the first flip-flop thus forming a loop. All but one flip-flop in a ring counter will be in the same state which could be either 0 or 1. Usually a bit “1” for a single flip-flop will be chosen, then the valid states of the 4 bit ring counter are 1000, 0100, 0010,0001 etc.

An example of a 4 bit ring counter constructed using S-R (set-reset) flip-flops is shown in figure 7.8(a). Output of each stage is fed to the S (set) input of next stage after AND'ing it with the 'INPUT' signal which can be considered to be a clock. The S input of the next stage is also fed to the R (reset) input of the current stage. The wave form of flip-flops FF1 through FF4 in response to the clock signal on 'INPUT' is shown in figure 7.8(b) (i.e. timing diagram).

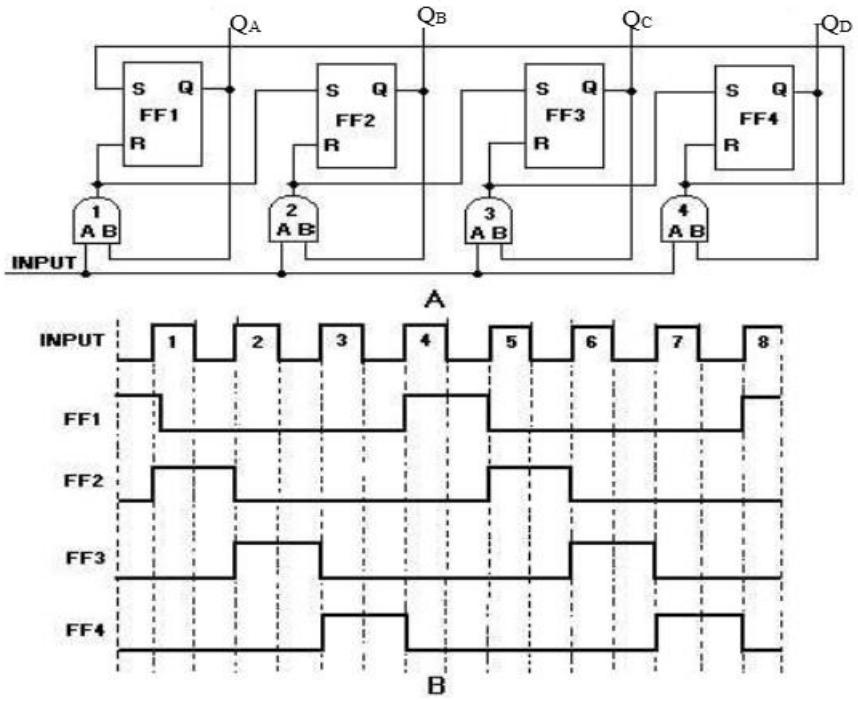


Figure 7.8: Ring Counter (a) Logic diagram counter (b) Timing diagram

The table 7.6 shows the truth table of 4-Bit ring counter

Table 7.6: Truth table of 4-Bit ring counter

Clock pulse	Q_A	Q_B	Q_C	Q_D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
7	0	0	0	1

Operation:

The operation of the circuit is straight forward. Assume that the initial output of first flip-flop is 1 and rest of the flip-flops and 'INPUT' are at 0. This means that A will be at 1 and B, C and D will be at 0. We can say that the counter state now is 1000 i.e. The Q output of first flip-flop is also connected to the B input of the first AND gate.

Let us apply a logic 1 to the 'INPUT' line which is connected to A inputs of all the AND gates. We know that as outputs (Q) of FF2 to FF4 are low, outputs of AND gates 2, 3, 4 will be 0. But the output of AND gate 1 is HIGH (1) as both of its inputs are at 1. This resets Q of FF1 to 0 and sets that of FF2 to 1. Hence B will be at logic 1 and A, C, D outputs will be at logic 0. Please note that in this process, the original state of FF1 appears to be shifted right by one position. The counter state now is 0100. Now if we apply one more pulse to the 'INPUT' line again, the same process repeats and the counter state would ne 0010.

Ring counter using D-flip flops:

A ring counter can also be constructed using d flip flops. The figure 7.9 shows a 4-Bit ring counter implemented using D-flip flops.

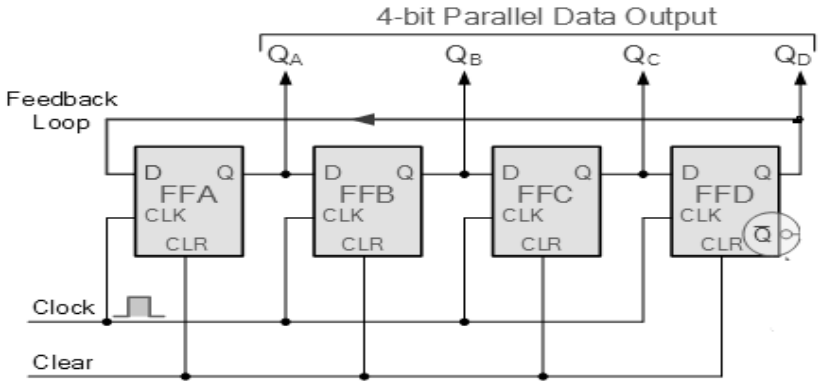


Figure 7.9: 4-Bit Ring counter using D-flip flops.

In the logic diagram shown in figure 7.9 above, the output of the last flip flop is fed back to the input of a first flip flop thereby forming a loop or a ring. Hence the name ring counters. A bit "1" will keep on circulating in a ring fashion when clock pulses are applied. The valid states of the 4 bit ring counter are 1000, 0100, 0010, 0001 etc. Note that during this operation, the asynchronous input Clear =0 making it inactive so that the entire operation will be based on the application of clock pulse.

7.3 Design of Modulo-N counters

In a counter, the sequence of states may follow a binary count or any other sequence of states. As we know counters count the number of occurrences of an event and are useful for generating timing sequences to control operations in a digital system. A counter that follows the binary sequence is called a binary counter. A single flip-flop used gives two states output and is referred to as mod-2 counter. With two flip-flops four output states can be counted in ascending or in descending way and is referred to as mod-4 or mod-2^2 counter. With 'n' flip-flops a mod-2^n counting is possible either of ascending or of descending type. So, an n-bit binary counter consists of n flip-flops and can count in binary from 0 to 2^n - 1.

To design an asynchronous counter to count till mod-N where N is a modulus of a counter and is not a power of 2, the following procedure is used.

- Find the number of flip-flops required $n = log_2 N$. calculated value is not an integer value if the N # 2^n then select n by rounding to the next integer value.
- First write the sequence of counting till N either in ascending or in descending way.
- Tabulate the value to reset the flip-flops in a mod-N count.
- Find the flip-flop outputs which are to reset from the tabulated value.
- Tap the output from these flip-flops and feed it to a suitable gate whose output is connected to the clear pin.

To design an synchronous counter, the following procedure can be used.

- Develop a state diagram for a given sequence.
- Develop a next-state table for a specific counter sequence.
- Create a FF transition table. (using JK Flip Flops)
- Use K-map to derive the logic equations. (Optional: Use Truth Table)
- Convert the logic equations to Boolean Algebra
- Implement a counter to produce a specified sequence of states.

Let's look at some examples of counter designs.

7.4 Problem on Counter Design

In this section we study some of the examples which show the design asynchronous and synchronous conters.

Problems on Asynchronous counter design

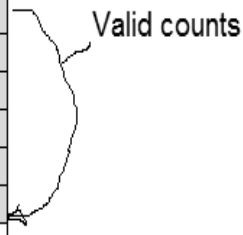
Problem 1: Design a Mod-6 counter using JK flip flop

Solution: The mod-6 counter counts from 0 through 5 (i.e. 000 to 101 in binary). After this this counter reset back to initial count 000 and it starts counting again. Only three flip flops are required, because $2^3 = 8$ and this number 8 is greater than mod-6.

Now we draw the table 7.7 to represent the desired output of the combinational circuit to reset FFs as

Table 7.7: The truth table for Mod 6-counter.

Clock 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



From the truth table it observed that we want the counter to count only from 000 to 101 and the count 110 is not required to be count. So the counter should reset to 000 immediately after the count 101. Observe that count 110 corresponds to Q2=1, Q1=1 and Q0=0. So we give the Q2 and Q1 as inputs to two input NAND gate so that whenever the value Q2=1 , Q1=1 appears the NAND gate outputs zero(o) and this out in turn activate the asynchronous inputs Clear (i.e. CLR) so that the counter resets back to 000 and it starts counting again. This mod-6 counter is shown in the figure 7.10. Note that J=K=1 so that each flip-flop acts as a Toggle flip-flop (i.e. T-flip flop).

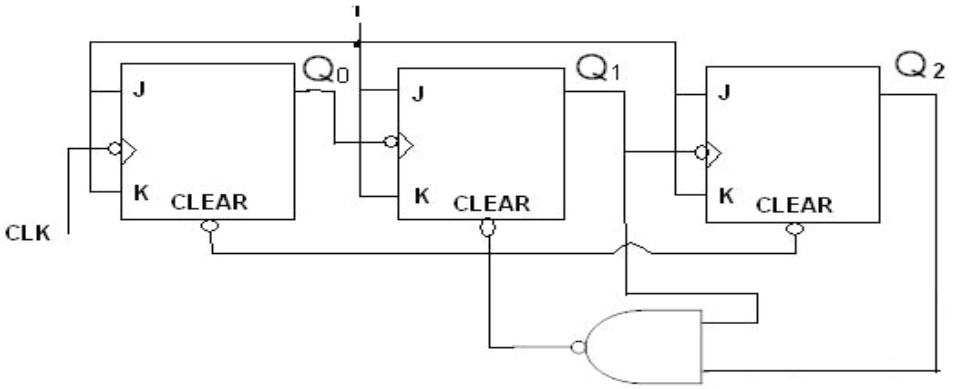


Figure 7.10: Mod-6 counter

2nd method:

We firstly draw the state diagram as shown in the figure 7.11.

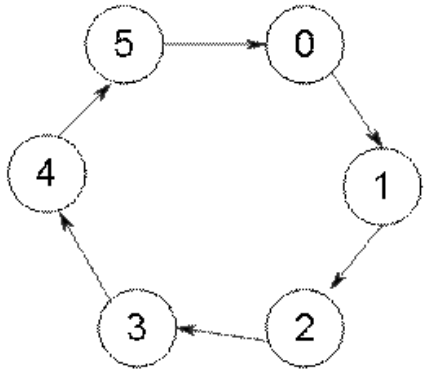


Figure 7.11: State diagram for Mod-6 counter

We draw the truth table to represent the desired output of the combinational circuit to reset FFs as shown in the table 7.8.

Table 7.8: The truth table for Mod 6-counter (for 2nd method).

Q2	Q1	Q0	OUTPUT
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Note that the output is 1 for required combinations or count and is 0 for the count 110 onwards.

Now we use k-map to get the simplified circuit as shown in the figure 7.12

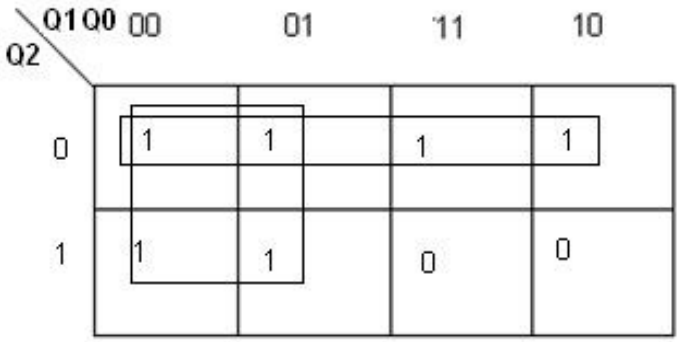


Figure 7.12: K-map for mod-6 design using 2nd method

And the equation we get is output = $\overline{Q}1+\overline{Q}2$.

$= \overline{Q1Q2}$

So, output= NAND(Q1 and Q2)

And the complete circuit is shown in figure 7.10.

Problem 2: Design the ripple counter whose output sequence is represented by the state diagram shown in figure 7.13.

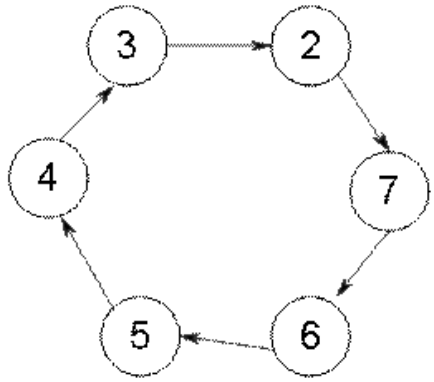


Figure 7.13: State diagram for problem 2

Solution: As we can see that it is a down counter so we'll be using Q bar of all flip-flops as clock to next flip-flops (negative edged FFs). We draw the table as shown in the table 7.9.

Table 7.9: Truth table for Problem 2

Q2	Q1	Q0	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

And using K-map (refer figure 7.14)

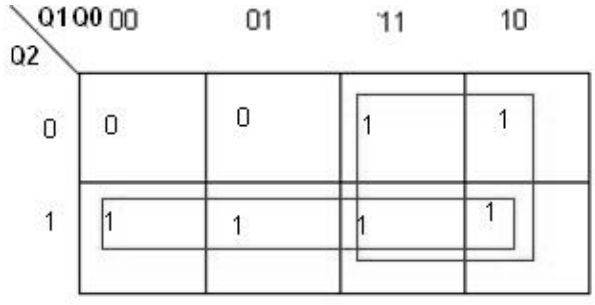


Figure 7.14: K-map for Problem 2

And the equation we get is

Output = Q2 + Q1

The circuit is shown in figure 7.15.

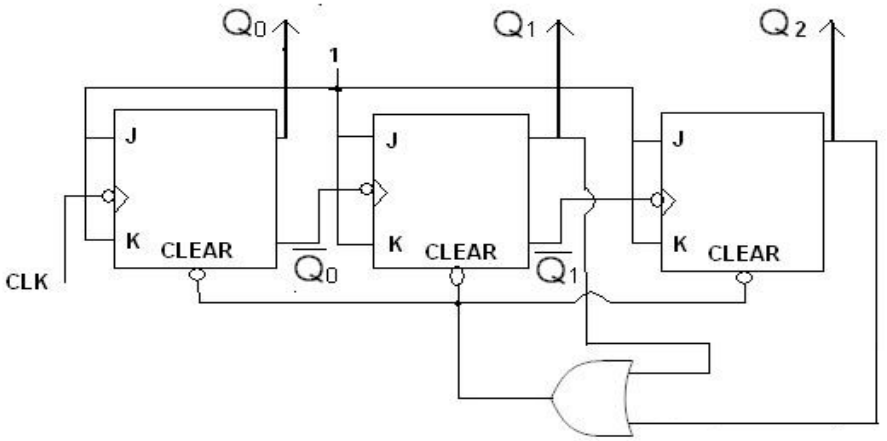


Figure 7.15: Logic circuit diagram for problem 2

Problems on Synchronous counter design

Problem 3: Design a Mod-5 synchronous counter

Solution: To design a mod-5 counter, the number of flip-flops required are found by the equation $2^n \geq N \geq 2^{n-1}$ where n is the number of flip-flops required and N is the number of states present in the counter. Now N=5, then n=3. So three flip flops are required to design a Mod-5 counter.

The state diagram for mod-5 counter is shown n figure 7.16.

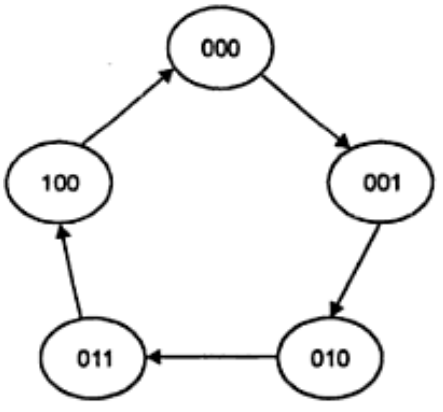


Figure 7.16: State diagram for Mod-5 counter

The state table for the counter is given in the table 7.10 and table 7.11 gives its excitation table.

Table 7.10: State table for the counter

Present state	*Next state*
000	001
001	010
010	011
011	100
100	000

Table 7.11: Excitation table for Mod-5 counter

Count Sequence			*Flip-flop inputs*					
A_2	A_1	A_0	JA_2	KA_2	JA_1	KA_1	JA_0	KA_0
0	0	0	0	X	0	X	1	X
0	0	1	0	X	1	X	X	1
0	1	0	0	X	X	0	1	X
0	1	1	1	X	X	1	X	1
1	0	0	X	1	0	X	0	X

The figure 7.17 shows the Karnaugh Map for Mod-5 Counter.

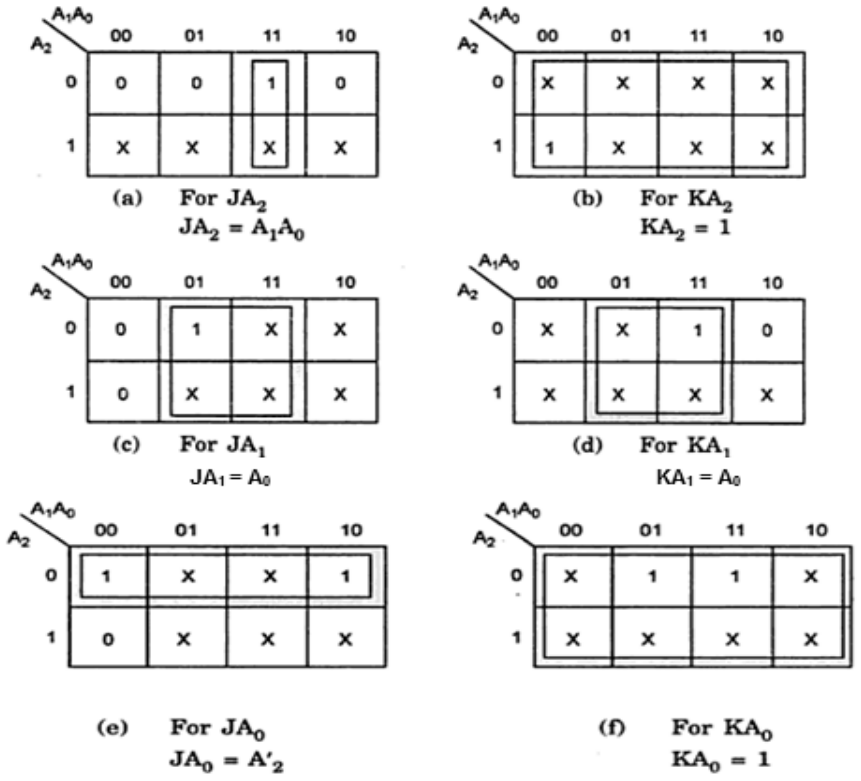


Figure 7.17: Karnaugh Map for Mod-5 counter

The figure 7.18 shows the logic circuit diagram for Mod-5 counter.

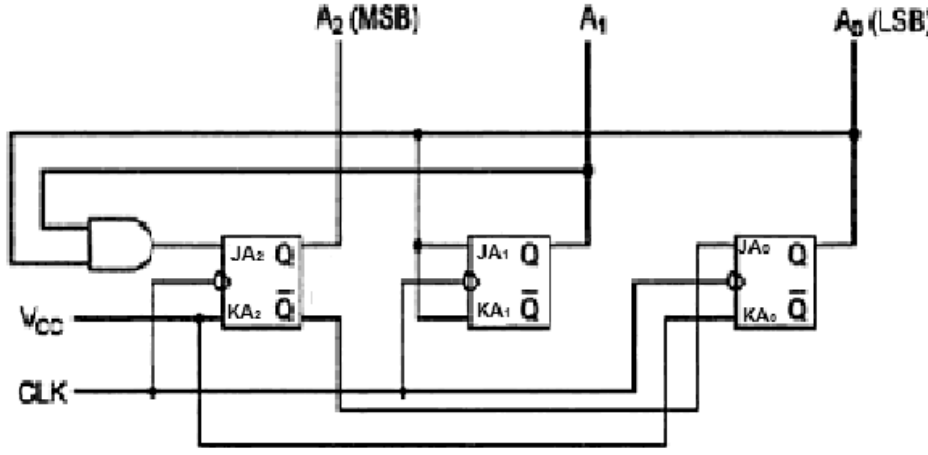


Figure 7.18: Logic diagram of a Mod-5 counter

Problem 4: Design a counter for the sequence of states shown in figure 7.19.

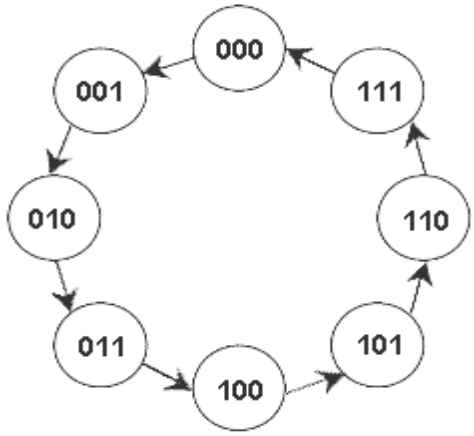


Figure 7.19: State diagram of a 3-bit binary counter.

Solution: The circuit has no inputs other than the clock pulse and no outputs other than its internal state (outputs are taken off each flip-flop in the counter). The next state of the counter depends entirely on its present state, and the state transition occurs every time the clock pulse occurs.

Once the sequential circuit is defined by the state diagram, the next step is to obtain the next-state table, which is derived from the state diagram in figure 7.19 and is shown in table 7.12.

Table 7.12: State table

Present State $Q_2\ Q_1\ Q_0$	Next State $Q_2\ Q_1\ Q_0$
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	1 0 1
1 0 1	1 1 0
1 1 0	1 1 1
1 1 1	0 0 0

Since there are eight states, the number of flip-flops required would be three. Now we want to implement the counter design using JK flip-flops.

Next step is to develop an excitation table from the state table, which is shown in table 7.13.

Table 7.13: Excitation table

Output State Transitions		Flip-flop inputs		
Present State $Q_2 Q_1 Q_0$	Next State $Q_2 Q_1 Q_0$	$J_2 K_2$	$J_1 K_1$	$J_0 K_0$
0 0 0	0 0 1	0X	0X	1X
0 0 1	0 1 0	0X	1X	X1
0 1 0	0 1 1	0X	X0	1X
0 1 1	1 0 0	1X	X1	X1
1 0 0	1 0 1	X0	0X	1X
1 0 1	1 1 0	X0	1X	X1
1 1 0	1 1 1	X0	X0	1X
1 1 1	0 0 0	X1	X1	X1

Now transfer the JK states of the flip-flop inputs from the excitation table to Karnaugh maps to derive a simplified Boolean expression for each flip-flop input. This is shown in figure 7.20.

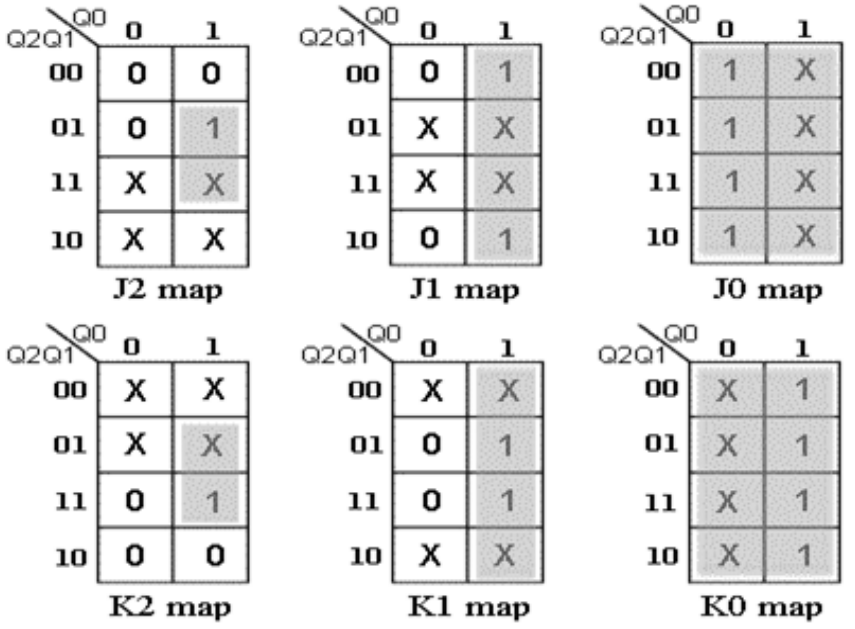


Figure 7.20: Karnaugh maps

The 1s in the Karnaugh maps of figure 7.20 are grouped with "don't cares" and the following expressions for the J and K inputs of each flip-flop are obtained:

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_1 * Q_0$$

The final step is to implement the combinational logic from the equations and connect the flip-flops to form the sequential circuit. The complete logic of a 3-bit binary counter is shown in figure 7.21.

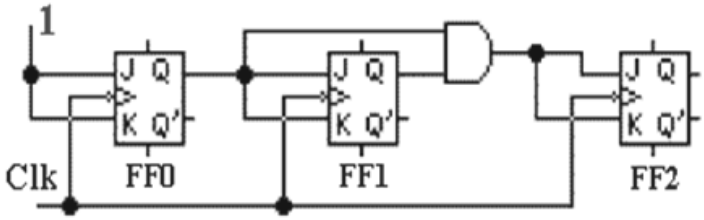


Figure 7.21: Logic of a 3-bit binary counter

Problem 5: Design a counter using T flip-flop as specified by the state diagram shown in figure in 7.22.

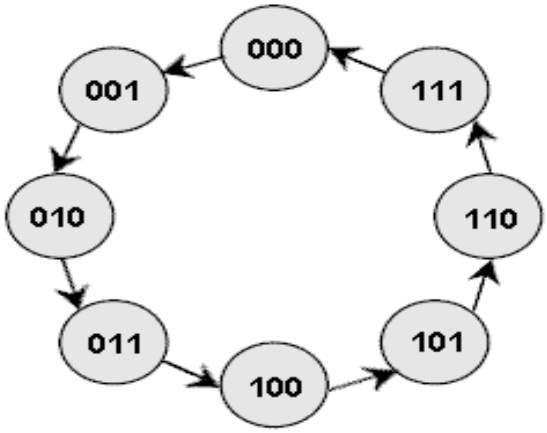


Figure 7.22: State diagram of a 3-bit binary counter.

Now derive the excitation table from the state table, which is shown in table 7.14.

Table 7.14: Excitation table for problem 5

Output State Transitions		Flip-flop inputs
Present State $Q_2\ Q_1\ Q_0$	Next State $Q_2\ Q_1\ Q_0$	$T_2\ T_1\ T_0$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

Next step is to transfer the flip-flop input functions to Karnaugh maps to derive simplified Boolean expressions, which is shown in figure 7.23.

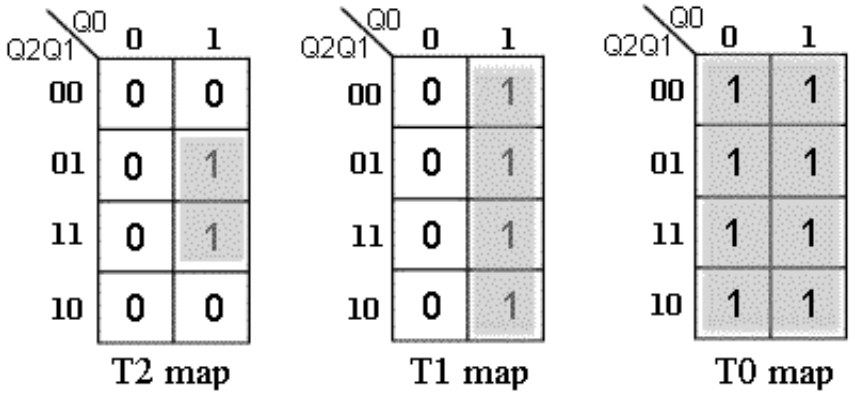


Figure 7.23: Karnaugh maps for problem 5

The following expressions are obtained:

$T_0 = 1$; $T_1 = Q_0$; $T_2 = Q_1 * Q_0$

Finally, draw the logic diagram of the circuit from the expressions obtained. The complete logic diagram of the counter is shown in figure 7.24.

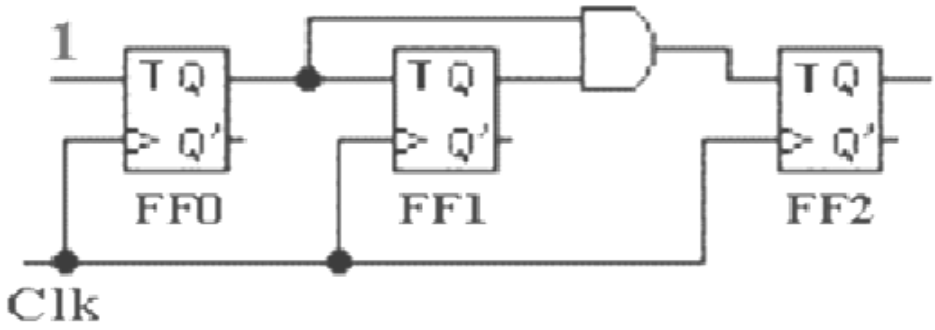


Figure 7.24: Karnaugh maps for problem 5

7.5 Summary

Let us recapitulate the important concepts discussed in this unit:

- In asynchronous or ripple counters, all the flip-flops are not clocked by the same clock and all flip-flops do not change their state in exact synchronism with the applied clock pulses.
- Modulus of a counter is the number of different states that a counter can go through before it comes back to initial state to repeat the count sequence.
- With 'n' flip-flops a mod-2^n counting is possible either of ascending or of descending type.
- The Johnson counter has an even-numbered cycle length of 2N where N is the number of stages in the register.
- The ring counter continuously circulates a stream of bits, all of which have same state except one bit. The differing bit may be 1 or 0. Hence possible 4 bit ring counter sequences are 0001, 0010, 0100, 1000, and the sequence is repeated.

Self Assessment Questions

1. Counters are designed using basic storage elements such as _____.
2. The counting range of a BCD counter is from _____ to _____.
3. The valid states of 3 bit ring counter with differing bit having a value of logic 1 are _____ , _____ and _____.
4. The length of a cycle of an n bit Johnson counter is _____.

5. A counter that follows the binary sequence is called a _____ counter.
6. The _____ counter counts from 0 through 5.

7.6 Terminal Questions

1. Explain the working of 4 bit Johnson counter with the help of neat diagram.
2. Explain the working of 4-bit ring counter. Draw the timing diagram.
3. Write a short note on design of modulo-N counters.
4. Design a Mod-5 synchronous counter

7.7 Answers

Self Assessment Questions

1. Flip-flops
2. 0 to 9
3. 001, 010 and 100
4. 2^n
5. Binary
6. Mod-6

Terminal Questions

1. Refer to sub-section 7.2.1
2. Refer to sub-section 7.2.2
3. Refer to section 7.3
4. Refer to section 7.4