MODULE 4: Hours-10

Sequential Circuits – 2:

Characteristic Equations, Registers, Counters -Binary Ripple Counters, Synchronous Binary counters, Counters based on Shift Registers, Design of a Synchronous counters, Design of a Synchronous Mod-6 Counter using clocked JK Flip-Flops Design of a Synchronous Mod-6

Counter using clocked D, T, or SR Flip-Flops

Recommended readings:

1. Donald D Givone, "Digital Principles and Design ", Tata McGraw

Hill Edition, 2002.

Unit - 6.6, 6.7, 6.8,6.9 – 6.9.1 and 6.9.2

REGISTERS

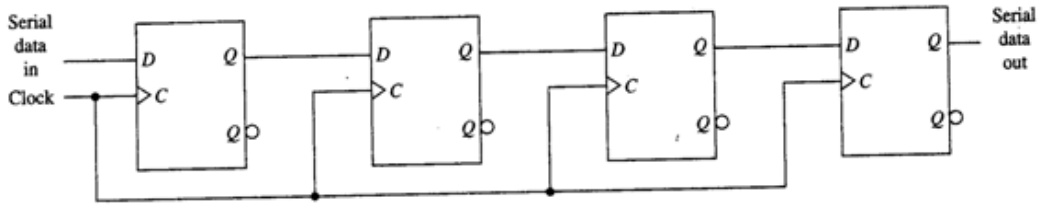


Fig. : Serial-In, Serial-Out Unidirectional Shift Register

- Register is a group of Flip-Flops.
- It stores binary information 0 or 1.
- It is capable of moving data left or right with clock pulse.
- Registers are classified as
    - Serial-in Serial-Out
    - Serial-in parallel Out
    - Parallel-in Serial-Out
    - Parallel-in parallel Out

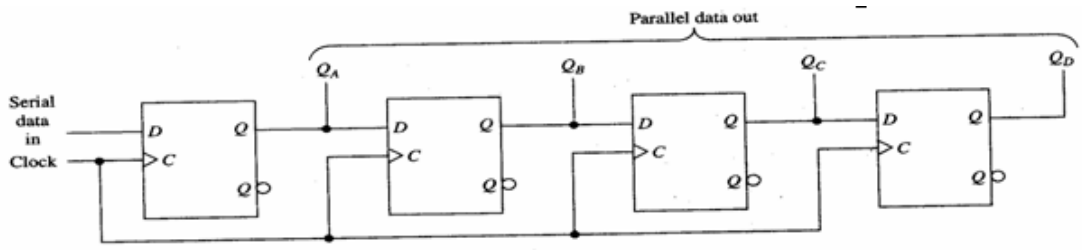


Fig. : Serial-In, Parallel-Out Unidirectional Shift Register

Parallel-in Unidirectional Shift Register

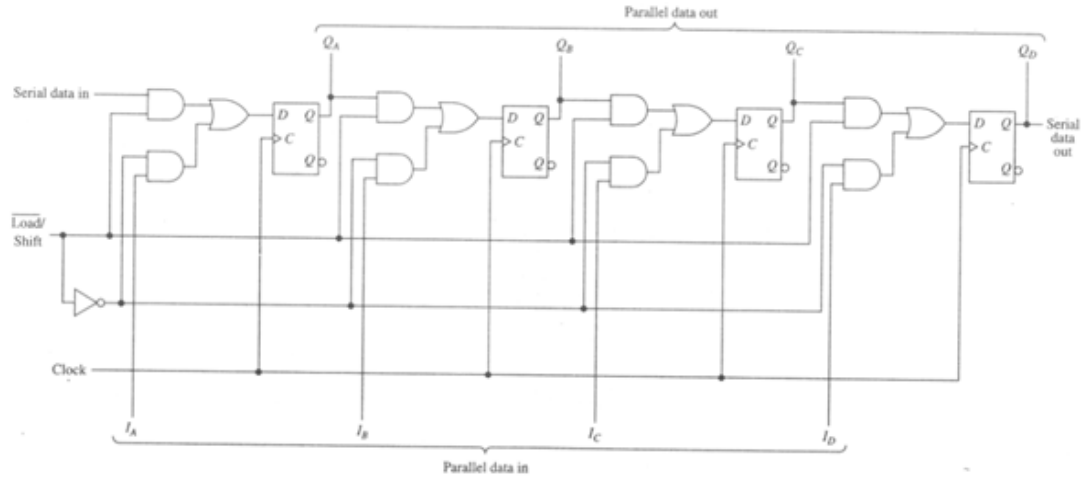
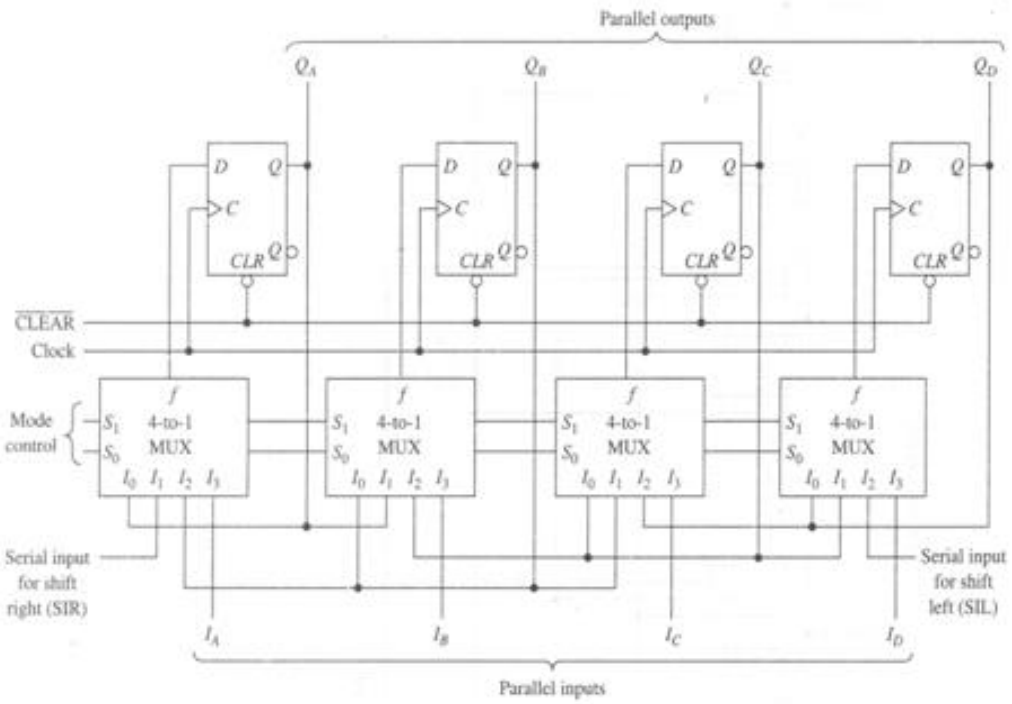


Fig. : Parallel-in Unidirectional Shift Register

- Parallel input data is applied at $I_AI_BI_CI_D$.
- Parallel output $Q_AQ_BQ_CQ_D$.
- Serial input data is applied to A FF.
- Serial output data is at output of D FF.
- $\bar{L}$/Shift is common control input.
- $\bar{L}$/S = 0, Loads parallel data into register.
- $\bar{L}$/S = 1, shifts the data in one direction.

Universal Shift Register



Select lines S_1	S_0	Register operation
0	0	Hold
0	1	Shift right
1	0	Shift left
1	1	Parallel load

- Bidirectional Shifting.
- Parallel Input Loading.
- Serial-Input and Serial-Output.
- Parallel-Input and Serial-Output.
- Common Reset Input.
- 4:1 Multiplexer is used to select register operation.

COUNTERS

- Counter is a register which counts the sequence in binary form.
- The state of counter changes with application of clock pulse.
- The counter is binary or non-binary.
- The total no. of states in counter is called as modulus.
- If counter is modulus-n, then it has n different states.
- State diagram of counter is a pictorial representation of counter states directed by arrows in graph.

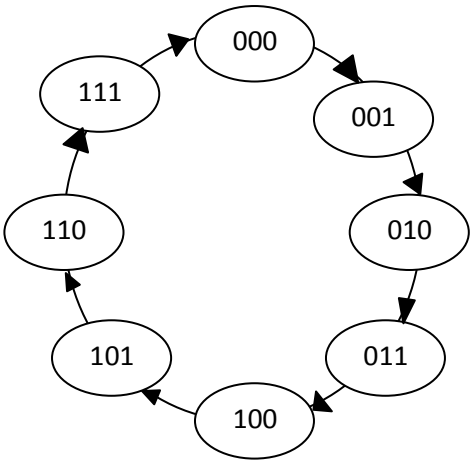


Fig. State diagram of mod-8 counter

Ripple and Arbitrary Counters

In this lesson, you will learn about:

_ Ripple Counters

_ Counters with arbitrary count sequence

Design of ripple Counters

Two types of counters are identifiable:

_ *Synchronous* counters, which have been discussed earlier, and

_ *Ripple* counters.

In ripple counters, flip-flop output transitions serve as a source for triggering other flipflops. In other words, clock inputs of the flip-flops are triggered by output transitions of other flip-flops, rather than a common clock signal.

Typically, T flip-flops are used to build ripple counters since they are capable of complementing their content (See Figure 1).

The signal with the pulses to be counted, i.e."*Pulse*", is connected to the clock input of the flip-flop that holds the **LSB** (FF # 1).

The output of each FF is connected to the clock input of the next flip-flop in sequence.

The flip-flops are negative edge triggered (bubbled clock inputs).

T=1 for all FFs (J = K= 1). This means that each flip-flop complements its value if C input goes through a negative transition (1 _ 0).

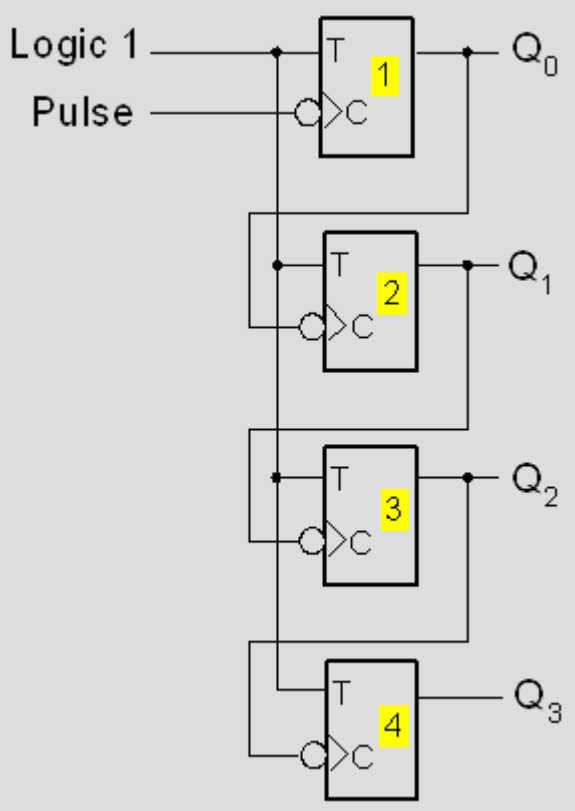


Figure 1: A ripple counter

The previous ripple up-counter can be converted into a down-counter in one of two ways:

_ Replace the negative-edge triggered FFs by positive-edge triggered FFs, or

_ Instead of connecting C input of FF Qi to the output of the preceding FF (Qi-1) connect it to the complement output of that FF (Q/i-1).

Advantages of Ripple Counters:

_ simple hardware and design.

Disadvantages of Ripple Counters:

_ They are asynchronous circuits, and can be unreliable and delay dependent, if more logic is added.

_ Large ripple counters are slow circuits due to the length of time required for the ripple to occur.

Counters with Arbitrary Count Sequence:

Design a counter that follows the count sequence: 0, 1, 2, 4, 5, 6. This counter can be designed with any flip-flop, but let's use the JK flip-flop.

Notice that we have two "*unused*" states (3 and 7), which have to be dealt with (see Figure 2). These will be marked by don't cares in the state table (Refer to the design of sequential circuits with unused states discussed earlier). The state diagram of this counter is shown in Figure 2.

In this figure, the unused states can go to any of the valid states, and the circuit can continue to count correctly. One possibility is to take state 7 (111) to 0 (000) and state 3 (011) to 4 (100).

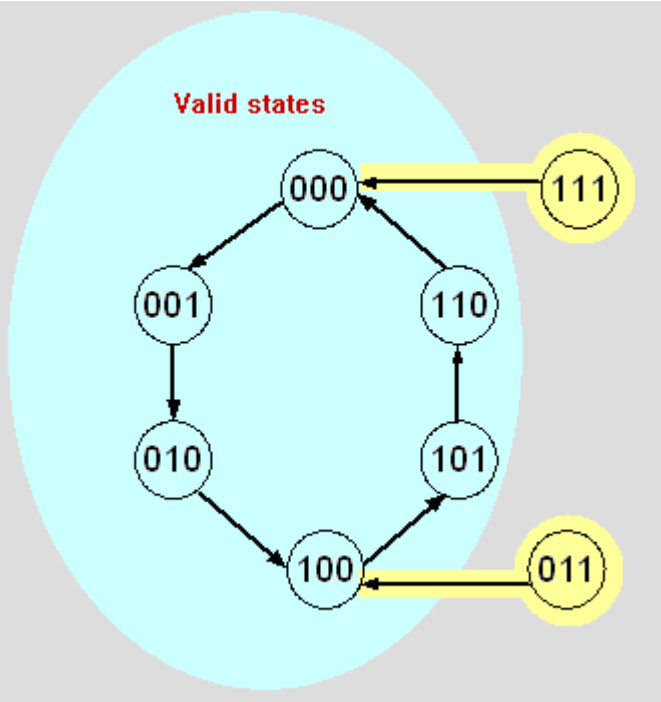


Figure 2: State diagram for arbitrary counting sequence

The design approach is similar to that of synchronous circuits. The state transition table is built as shown in Figure 3 and the equations for all J and K inputs are derived. Notice that we have used don't care for the unused state (although we could have used 100 as the next state for 011, and 000 as the next state of 111).

Unused states

Present State			Next State			Flip-Flop Inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
0	1	1	X	X	X	X	X	X	X	X	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	X	X	X	X	X	X	X	X	X

Figure 3: State table for arbitrary counting sequence

The computed J and K input equations are as follows:

JA = B KA =B

JB = C KB =1

JC = B/ KC =1

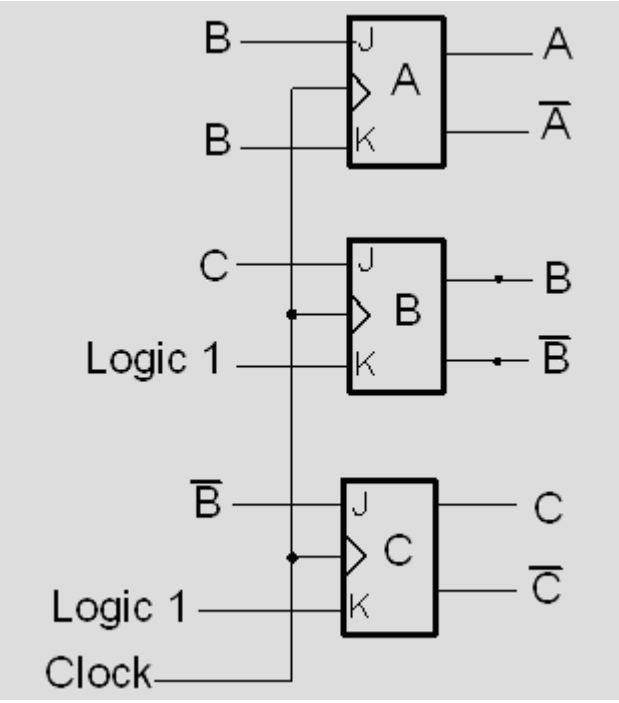
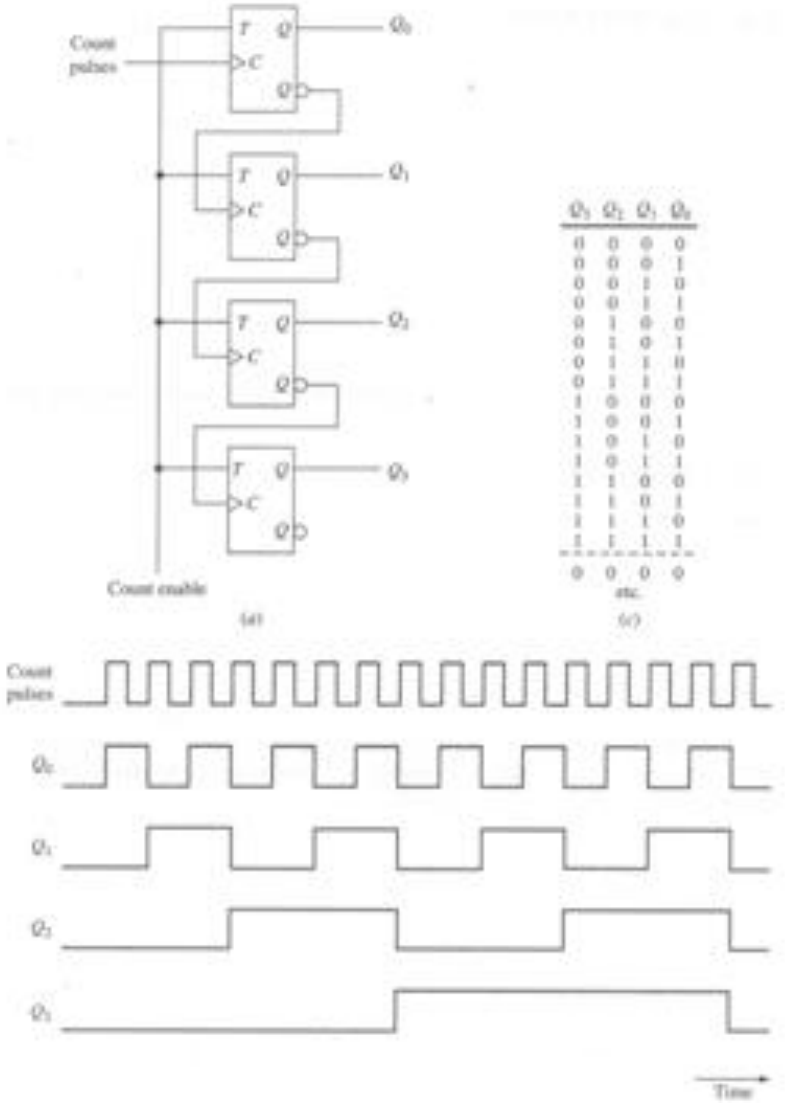


Figure 4: Circuit for arbitrary counting sequence

4-bit Binary Ripple Counter :



- All Flip-Flops are in toggle mode.
- The clock input is applied.
- Count enable = 1.
- Counter counts from 0000 to 1111.

Counters

In this lesson, the operation and design of Synchronous Binary Counters will be studied.

Synchronous Binary Counters (SBC)

Description and Operation

In its simplest form, a *synchronous binary counter* (**SBC**) receives a train of clock *pulses* as input and outputs the *pulse count* (Q*n-1* …. Q*2* Q*1* Q*0*).

An example is a 3-bit counter that counts from 000 upto 111. Each counter consists of a number of FFs. (Figure 1)

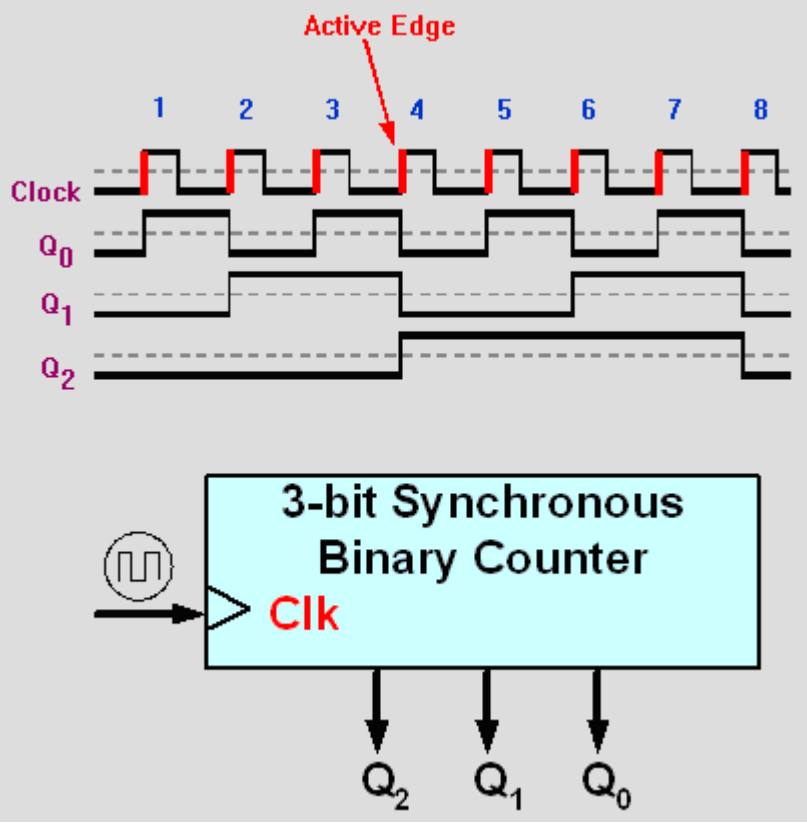


Figure 1: 3-bit SBC

In *synchronous* counters, all FFs are triggered by the same input clock.

An *n*-bit counter has *n*-FFs with 2*n* distinct *states*, where each state corresponds to a particular *count*.

Accordingly, the possible *counts* of an *n*-bit counter are 0 to (2*n*-1). Moreover an n-bit

counter has *n* output bits (Q*n-1* …. Q*2* Q*1* Q*0*).

After reaching the maximum count of (2*n*-1), the following clock pulse resets the count back to 0.

Thus, a 3-bit counter counts from 0 to 7 and back to 0. In other words, the output count actually equals (*Total # of input pulses* **Modulo** 2*n*).

Accordingly, it is common to identify counters by the modulus 2*n*. For example, a 4-bit counter provides a modulo 16 count, a 3-bit counter is a modulo 8 counter, etc.

Referring to the 3-bit counter mentioned earlier, each stage of the counter divides the frequency by 2, where the last stage divides the frequency by 2*n*, *n* being the number of bits. (Figure 2)

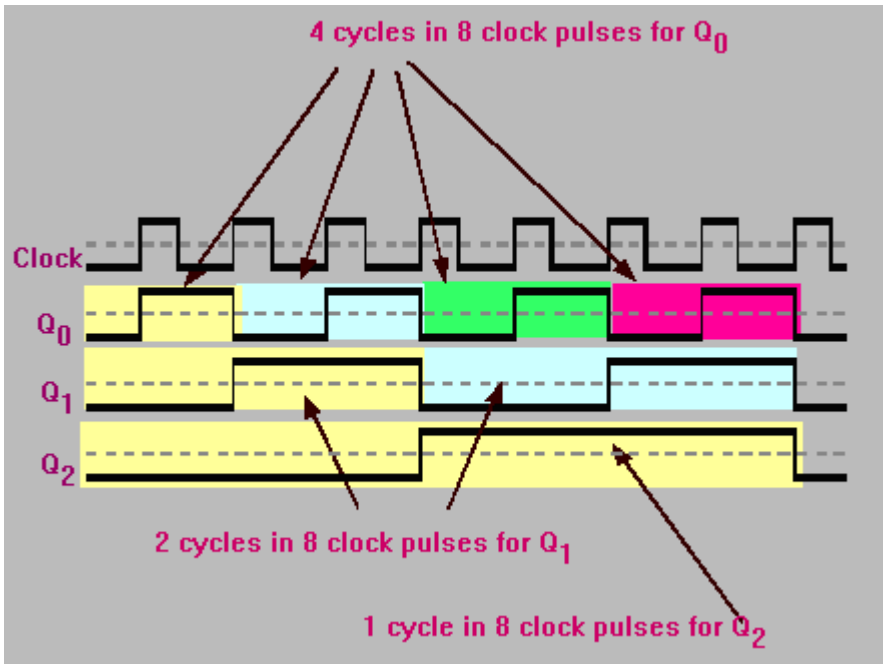


Figure 2: 3-bit SBC

Thus, if the frequency (i.e. no. of cycles/ sec) of clock is F, then the frequency of output waveform of Q0 is F/2, Q1 is F/4, and so on. In general, for n-bit counter, we have F/2n.

Design of Binary Counters (SBC)

Design procedure is the same as for other synchronous circuits.

A counter may operate without an external input (except for the clock pulses!)

In this case, the output of the counter is taken from the outputs of the flip-flops without

any additional outputs from gates.

Thus, there are no columns for the input and outputs in the state table; we only see the current state and next state…

Example Design a 4-bit SBC using JK flip-flops.

The counter has 4 FFs with a total of 16 states, (0000 to 1111) _ 4 state variables Q3 Q2 Q1 Q0 are required.

Present State				Next State				Flip-Flop Inputs							
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_{Q3}	K_{Q3}	J_{Q2}	K_{Q2}	J_{Q1}	K_{Q1}	J_{Q0}	K_{Q0}
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	1	0	1	0	X	0	0	X	1	X	X	1
1	0	1	0	1	0	1	1	X	0	0	X	X	0	1	X
1	0	1	1	1	1	0	0	X	0	1	X	X	1	X	1
1	1	0	0	1	1	0	1	X	0	X	0	0	X	1	X
1	1	0	1	1	1	1	0	X	0	X	0	1	X	X	1
1	1	1	0	1	1	1	1	X	0	X	0	X	0	1	X
1	1	1	1	0	0	0	0	X	1	X	1	X	1	X	1

Figure 3: State table for the example

Notice that the next state equals the present state plus one.

To design this circuit, we derive the flip-flop input equations from the state transition table. Recall that to find J & K values, we have to use:

_ The present state,

_ The next state, and

_ The JK flip-flop excitation table.

When the *count* reaches 1111, it resets back to 0000, and the count cycle is repeated.

Once the J and K values are obtained, the next step is to find out the simplified input equations by using K-maps, as shown in figure 4.

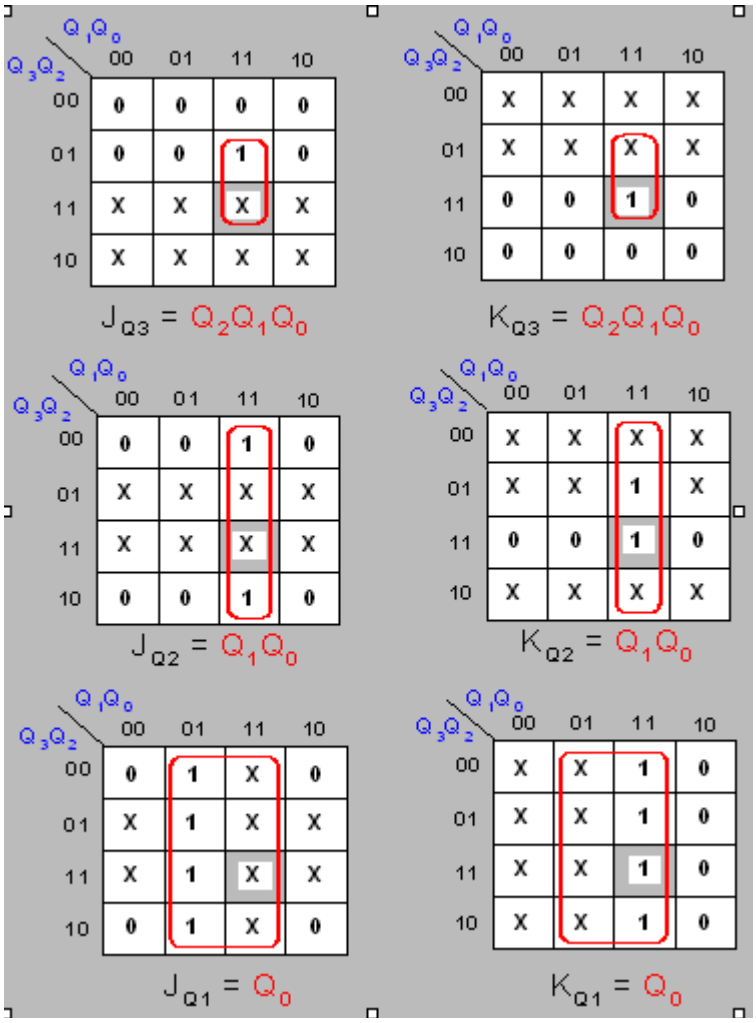


Figure 4: K-maps for the example

Notice that the maps for JQ0 and KQ0 are not drawn because the values in the table for these two variables either contain 1's or X's. This will result in JQ0 = KQ0 = 1

Note that the Boolean equation for J input is the same as that of the K input for all the FFs ⇒ Can use T-FFs instead of JK-FFs.

Count Enable Control

In many applications, controlling the counting operation is necessary ⇒ a count-enable (En) is required.

If En= 1 **then** counting of incoming clock pulses is enabled **Else if** (En =0), no incoming clock pulse is counted.

To accommodate the enable control, two approaches are possible.

1. Controlling the clock input of the counter

2. Controlling FF excitation inputs (JK, T, D, etc.).

Clock Control

Here, instead of applying the system clock to the counter directly, the clock is first ANDed with the En signal.

Even though this approach is simple, it is not recommended to use particularly with configurable logic, e.g. FPGA's.

FF Input Control (Figure 5)

In this case, the En =0 causes the FF inputs to assume the no change value (SR=00, JK=00, T=0, or D*i*=Q*i*).

To include En, analyze the stage when JQ1 = KQ1 = Q0, and then include En. Accordingly, the FF input equations of the previous 4-bit counter example will be modified as follows:

JQ0 = KQ0 = 1. EN = En

JQ1 = KQ1 = Q0. En

JQ2 = KQ2 = Q1.Q0. En

JQ3 = KQ3 = Q2.Q1.Q0. En

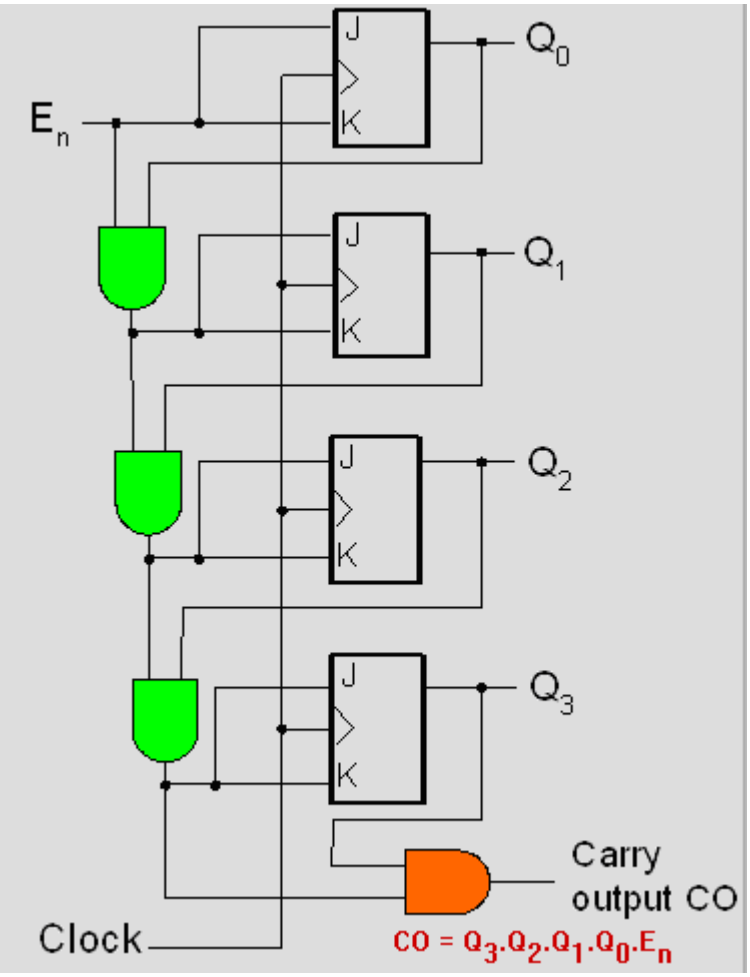


Figure 5: FF input control in counter

Thus, when En = 0, all J and K inputs are equal to zero, and the flip flops remain in the same state, even in the presence of clock pulses

When En = 1, the input equations are the same as equations of the previous example.

A carry output signal (CO) is generated when the counting cycle is complete, as seen in the timing diagram.

The CO can be used to allow cascading of two counters while using the same clock for both counters. In that case, the CO from the first counter becomes the En for the second counter. For example, two modulo-16 counters can be cascaded to form a modulo-256 counter.

Up-Down Binary Counters

In addition to counting up, a SBC can be made to count down as well.

A control input, S is required to control the direction of count.

IF S= 1, the counter counts up, otherwise it counts down.

FF Input Control

Design a Modulo-8 up-down counter with control input S, such that if S= 1, the counter counts up, otherwise it counts down. Show how to provide a count enable input and a carry-out (CO) output. (See figures 6 & 7)

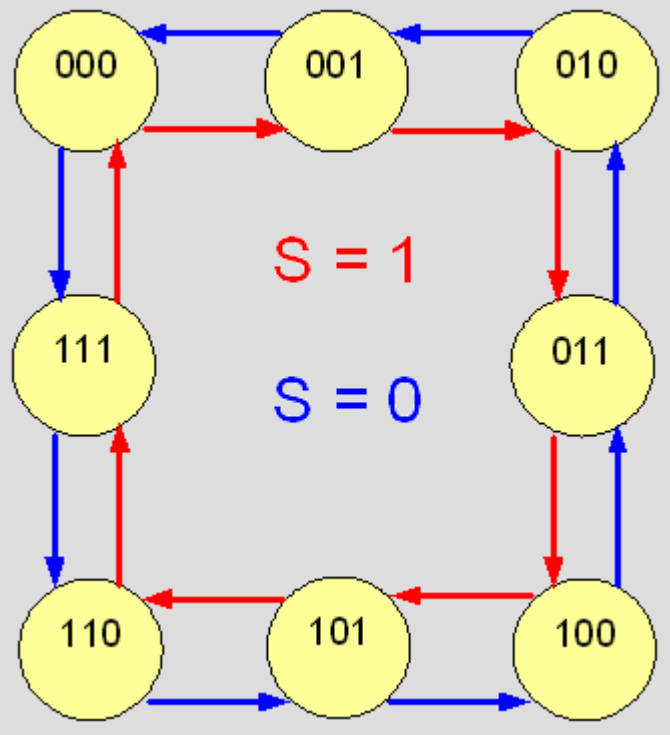


Figure 6: State diagram for FF input control example

Present State				Next State			Flip-Flop Inputs		
Q_2	Q_1	Q_0	S	Q_2	Q_1	Q_0	T_2	T_1	T_0
0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0	1
0	0	1	1	0	1	0	0	1	1
0	1	0	0	0	0	1	0	1	1
0	1	0	1	0	1	1	0	0	1
0	1	1	0	0	1	0	0	0	1
0	1	1	1	1	0	0	1	1	1
1	0	0	0	0	1	1	1	1	1
1	0	0	1	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0	1
1	0	1	1	1	1	0	0	1	1
1	1	0	0	1	0	1	0	1	1
1	1	0	1	1	1	1	0	0	1
1	1	1	0	1	1	0	0	0	1
1	1	1	1	0	0	0	1	1	1

Figure 7: State table for FF input control example

The equations are (see figure 8)

T0 = 1

T1 = Q0. S + Q□

0. S□

T2 = Q1.Q0. S + Q□

1. Q□

0. S□

The carry outputs for the next stage are: (see figure 8)

Cup = Q2.Q1.Q0 for upward counting.

Cdown = Q□

2.Q□

1.Q□

0 for downward counting.

The equations with En are (see figure 9)

T0 = En. 1

T1 = Q0. S. En + Q□

0. S□. En

T2 = Q1.Q0. S. En + Q□

1. Q□

0. S□. En

The carry outputs for the next stage, with En are (see figure 9):

Cup = Q2.Q1.Q0. En for counting up.

Cdown = Q□

2.Q□

1.Q□

0. En for counting down.

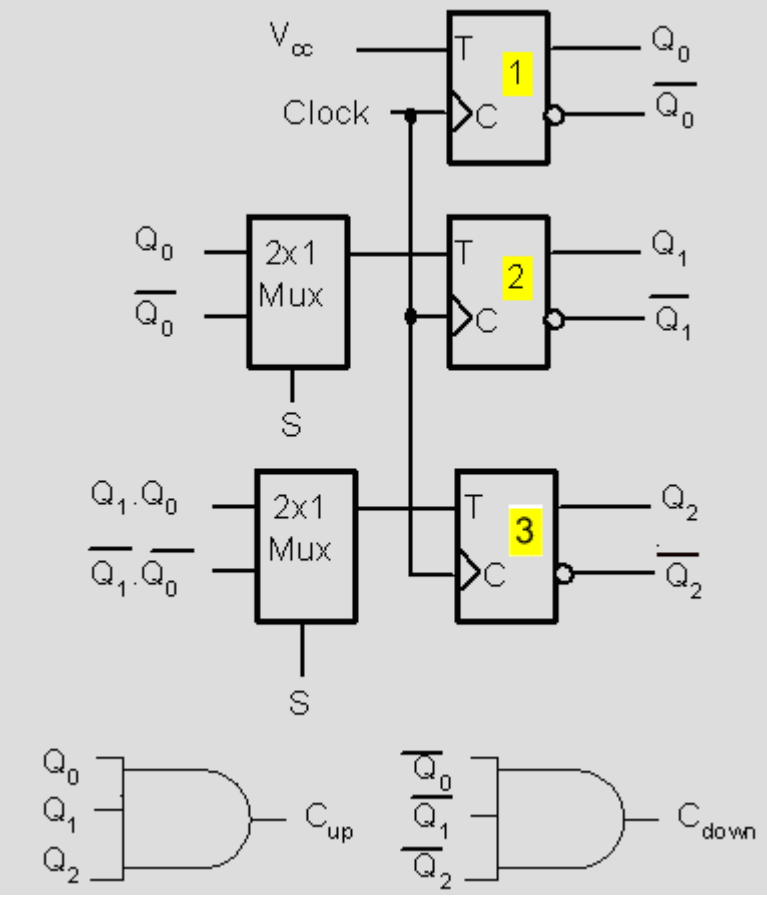


Figure 8: Circuit of up-down counter

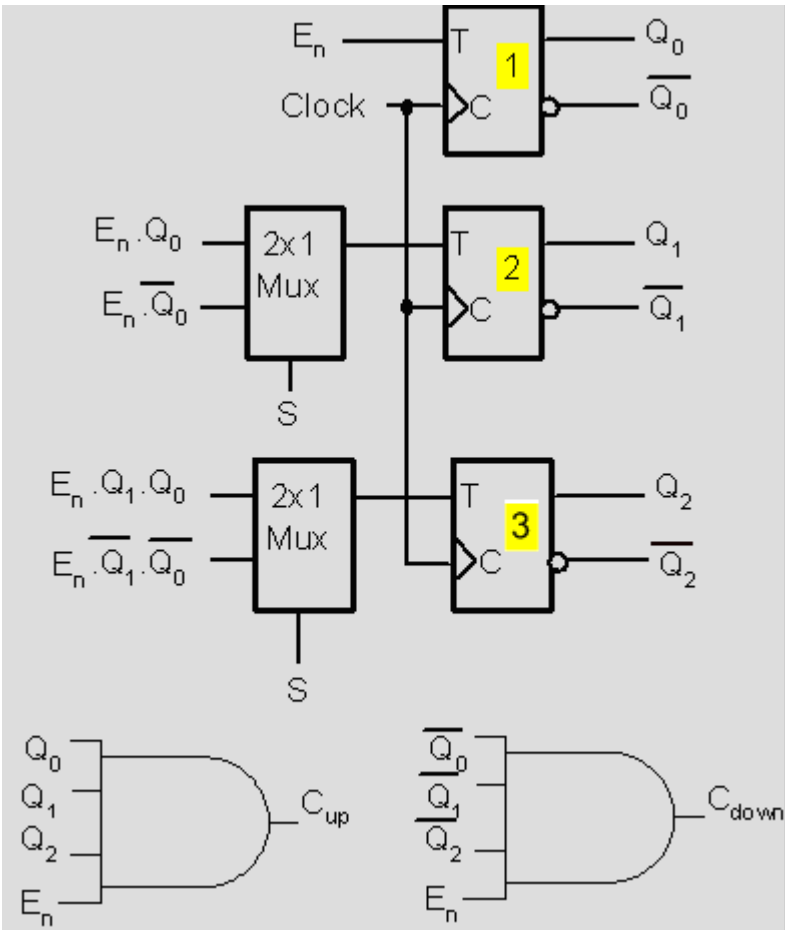
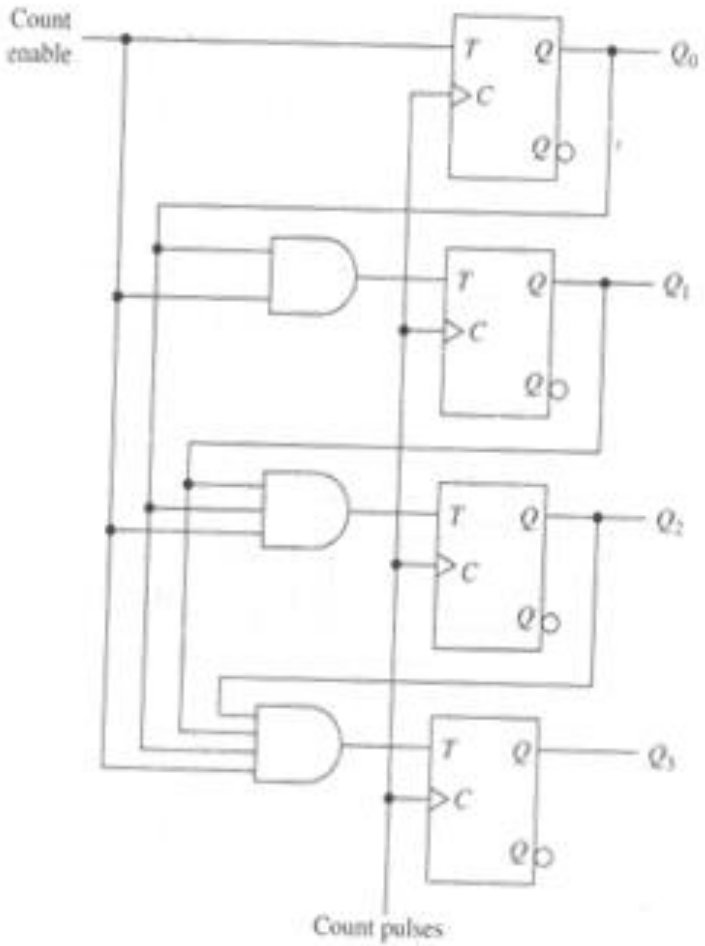


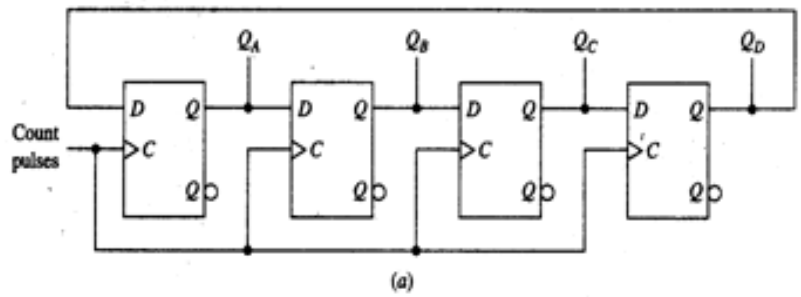
Figure 9: Circuit of up-down counter with En

Synchronous Binary Counter :



- The clock input is common to all Flip-Flops.
- The T input is function of the output of previous flip-flop.
- Extra combination circuit is required for flip-flop input.

Counters Based on Shift Register



(a)

Q_A	Q_B	Q_C	Q_D
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

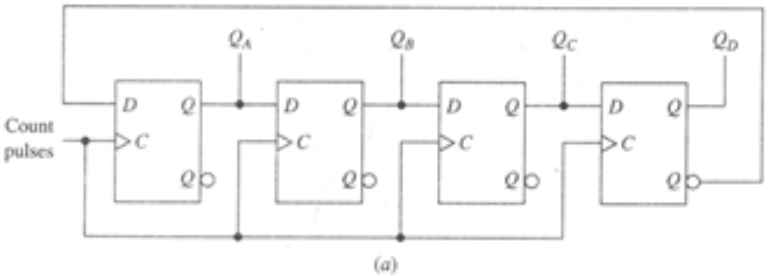
etc.

(b)

Mod-4 Ring Counter

- The output of LSB FF is connected as D input to MSB FF.
- This is commonly called as Ring Counter or Circular Counter.
- The data is shifted to right with each clock pulse.
- This counter has four different states.
- This can be extended to any no. of bits.

Twisted Ring Counter or Johnson Counter



(a)

Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

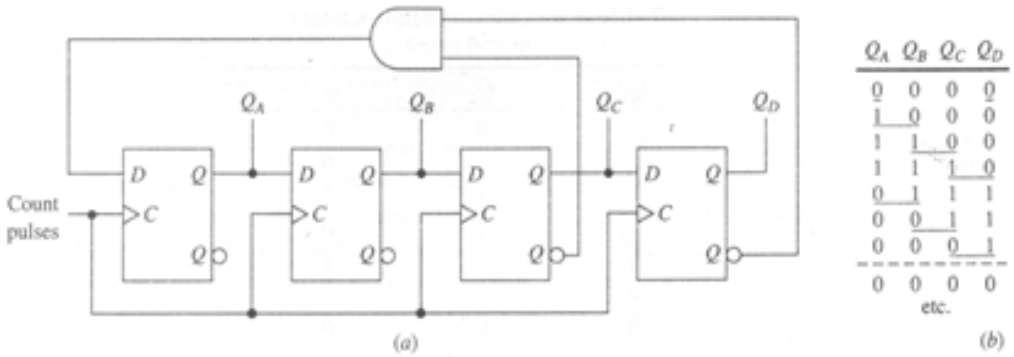
etc.

(b)

Mod-8 Johnson Counter

- The complement output of LSB FF is connected as D input to MSB FF.
- This is commonly called as Johnson Counter.
- The data is shifted to right with each clock pulse.
- This counter has eight different states.
- This can be extended to any no. of bits.

Mod-7 Twisted Ring Counter



Mod-7 Ring Counter

- The D input to MSB FF is $\overline{Q_D}.\overline{Q_C}$
- The counter follows seven different states with application of clock input.
- By changing feedback different counters can be obtained.

Design Procedure for Synchronous Counter

- The clock input is common to all Flip-Flops.
- Any Flip-Flop can be used.
- For mod-n counter 0 to n-1 are counter states.
- The excitation table is written considering the present state and next state of counter.
- The flip-flop inputs are obtained from characteristic equation.
- By using flip-flops and logic gate the implementation of synchronous counter is obtained.

Difference between Asynchronous and Synchronous Counter :

Asynchronous Counter	Synchronous Counter
1. Clock input is applied to LSB FF. The output of first FF is connected as clock to next FF.	1. Clock input is common to all FF.
2. All Flip-Flops are toggle FF.	2. Any FF can be used.
3. Speed depends on no. of FF used for n bit . $f_{max} = \frac{1}{n \times t_p}$	3. Speed is independent of no. of FF used. $f_{max} = \frac{1}{t_p}$
4. No extra Logic Gates are required.	4. Logic Gates are required based on design.
5. Cost is less.	5. Cost is more.

Recommended question and answer –unit-6

Jan -2009

b) *Explain the working principle of a mod-6 binary ripple counter, configured using positive edge triggered T-FF. Also draw the timing diagram.*
(8),

Ans. : Mod-8 ripple counter using T flip flop: For designing counter using T flip flop,

Flip-flops required are : 2n>= N

Here N = 8 ∴ n = 3
i.e. 3 FFs are required

Excitation table for T FF :

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

Transition table :

Present state			Next state			Flip-flop inputs		
A	B	C	A^+	B^+	C^+	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

K-map simplification :

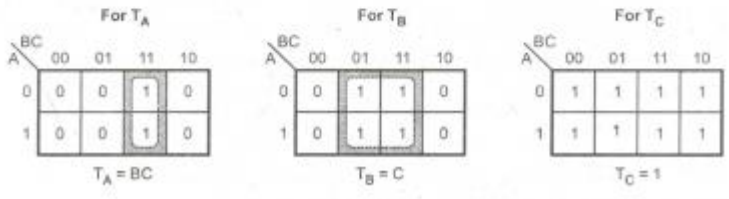


Fig. 6 (a)

Implement the counter :

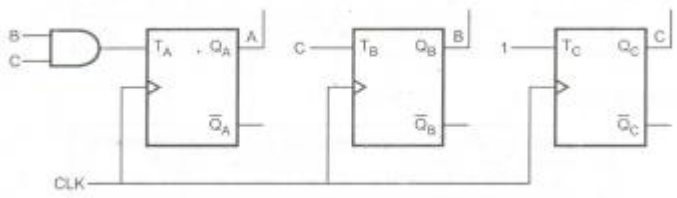


Fig. 6 (b)

Jan-2008

i) Synchronous and asynchronous circuits.

ii) Combinational and sequential circuits.

Ans. : i) Synchronous and asynchronous circuits :

Sr. No.	Synchronous sequential circuits	Asynchronous sequential circuits
1.	In synchronous circuits, memory elements are clocked flip-flops.	In asynchronous circuits, memory elements are either unclocked flip-flops or time delay elements.
2.	In synchronous circuits, the change in input signals can affect memory element upon activation of clock signal.	In asynchronous circuits 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.

b) *Explain the working of 4-bit asynchronous counter.*

Ans. : 4-bit asynchronous counter :

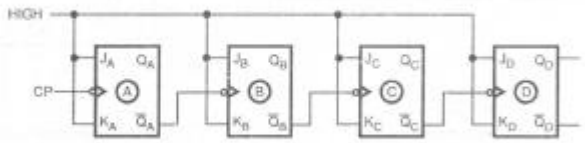


Fig. 10 4-bit asynchronous down counter

1) 4 flip-flops are employed to create a 4-bit asynchronous counter as shown.

2) The clock signal is connected to the clock input of only first stage flip-flop.

3) Because of the inherent propagation delay time through a flip-flop, two flip-flops never trigger simultaneously. Thus, it works in an asynchronous operation.

4) Output of the first flip-flop triggers the second flip-flop and so on.

S) At the output of flip-flops, we get the counted value of the counter.

c) *Explain Johnson counter with its circuit diagram and timing diagram.* (8)

Ans. : **4-bit Johnson counter :**

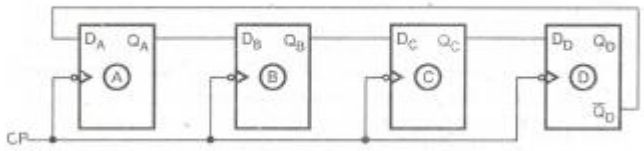


Fig. 11 Four-bit Johnson counter

1) Initially, the register is cleared.

:. all the outputs QA' *QSI QCI* Qo are zero.

2) The complement of Q 0 is 1 which is connected back to the D input of first stage.

:. DA is, 1.

:. The outPut becomes QA = 1, Qs = 0, Qc = a and Qo = O.

3) The next clock pulse produces Q A = 1, Q B = 1, Q C = a and Q 0 = O.

The sequence is' given as :

Clock Pulse	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

Table 3 Four-bit Johnson sequence

Aug-2008

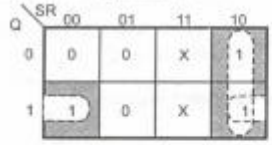
Q.5 a) *Derive the characteristics equations of the following flip-flops.*

i) SR flip-flops ii) JK flip-flop (10)

Ans. : i) Function table of SR flip-flop

S	R	Q	Clk	Q+
0	0	0	–	0
0	0	1	–	1
0	1	0	–	0
0	1	1	–	0
1	0	0	–	1
1	0	1	–	1
1	1	0	–	X
1	1	1	–	X
X	X	Q	0	Q
X	X	Q	1	Q

Characteristic equation :



$Q+ = S + \overline{R}\,Q$

Fig. 8 (a)

In words :

If SR = 10, on high going edge of clock the Q output becomes 1.

If SR = 01, Q becomes 0.

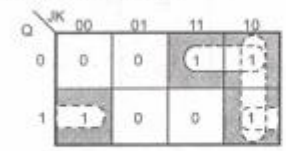
If SR = 11, Q raised condition.

If SR = 00, Q does not change.

ii) Function table of JK flip-flop

J	K	Q	Clk	Q^+
0	0	0	–	0
0	0	1	–	1
0	1	0	–	0
0	1	1	–	0
1	0	0	–	1
1	0	1	–	1
1	1	0	–	1
1	1	1	–	0
X	X	Q	0	Q
X	X	Q	1	Q

Characteristic equation



$Q+ = \overline{Q}\,J + Q\,\overline{K}$

Fig. 8(b)

In words :

If JK = 10, On high going edge of clock, the Q output becomes l.
If JK = 01, .Q becomes O.
If JK = 11, . Q toggles.
If JK = 00, Q does not change.

b) *Explain clearly the operation of an asynchronous inputs in a flip-flops with suitable example.* (6)

Ans. : The best example of operation of asynchronous input to flip-flop is counter. Use of counter is to count the clock pulse. For these counters the external clock signal is applied to one- flip-flop and then the output of preceding flip-flop is

connected to the clock of next flip-flop

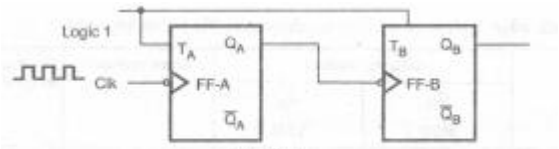


Fig. 9 (a)

Operation:

1) Initially both the flip-flops be in reset condition.

.. QBQA = 00

2) On the first negative going clock edge : As the 1st falling edge of the clock hits FF-A, it will toggle as TA = 1. Hence QA will be equal to 1. But for FF - B it has no changed from 0 to I, it is treated as the positive clock edge by FF - B.

So

$Q_B Q_A = 01$ ... After the first clk pulse

On second falling edge of clock pulse :

On arrival of second falling cloCk edge, FF-A toggles again, to make QA = O. This change in QA (from 1 to 0) acts as a negative clock edge for FF-B. So it will also toggle, and QB will become 1.

Hence after the second clock pulse the counter output are

$Q_B Q_A = 10$ ... After the second clk pulse

Both the outputs are changing their state. But both the changes do not take place

simultaneously, QA will change first from 1 to 0 and then QB will change from 0 to 1. This is due to propagation delay of FF-A. So both flip-flops will never triggered at the same instant. So it is asynchronous counter

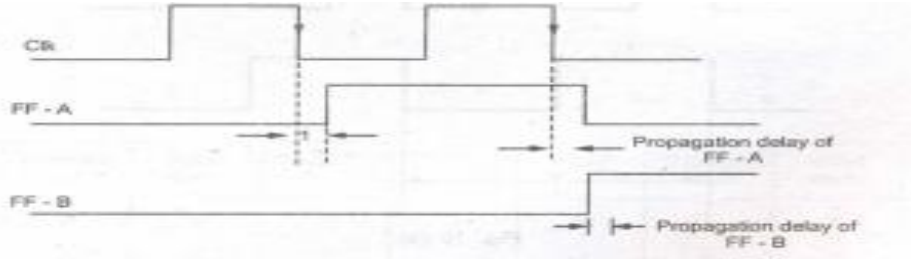


Fig. 9 (b)

For 4 clock edge operation will be as shown in the following table.

Clock	Counter output		State number	Decimal equation of counter output
	Q_B MSB	Q_A LSB		
Initially	0	0	–	0
1 ↓	0	1	1	1
2 ↓	1	0	2	2
3 ↓	1	1	3	3
4 ↓	0	0	4	0

c) *An edge triggered 'D' flip-flop is connected as shown in the Fig. 10. Assume that* An Q = 0 *initially and sketch the waveform and determine its frequency of the signal at 'Q' output. (4)*

Ans. :

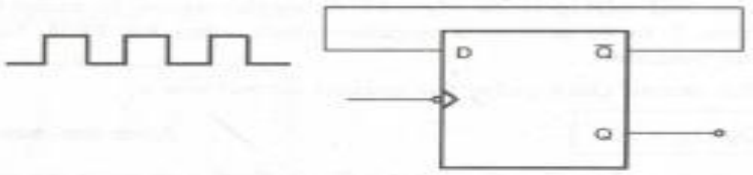


Fig. 10

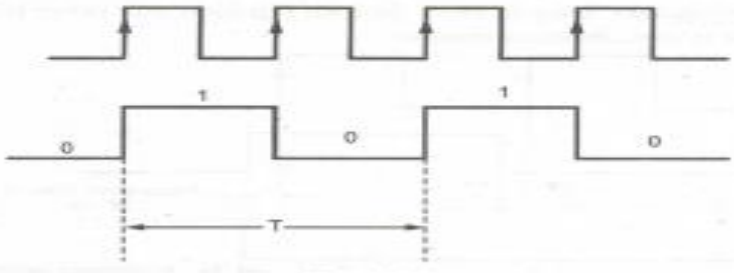


Fig. 10 (a)

$$f = \frac{1}{T} = f = \frac{f_{clock}}{2}$$

Aug-2007

c) *Write the truth table of the following flip flops :*
D, T, SR, JK.

Sol. : D :

CP	D	Q_{n+1}
↓	0	0
↓	1	1
0	X	Q_n

Truth table of D flip-flop

Table 2

Logic Diagram

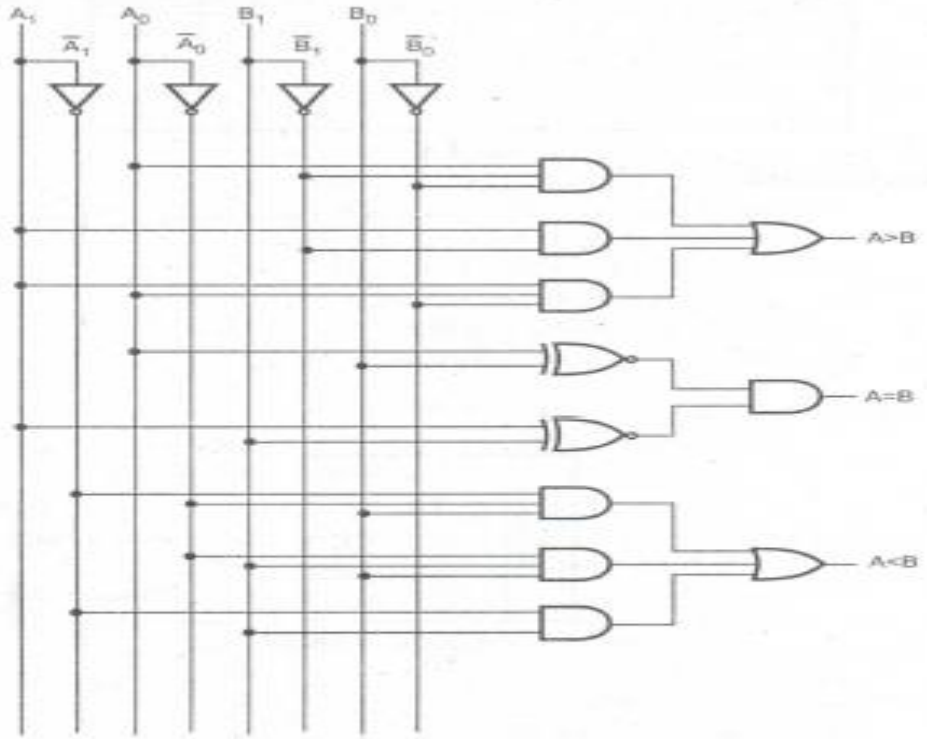


Fig. 9

b) *Implement the following Boolean function using 8:1 MUX :*
$F(A,B,C,D) = \Sigma m(1,2,5,9,10,14)$

Sol. : Implementation Table :

T :

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

≡

T	Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

(b) Truth table (3)

SR :

EN	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	No change (NC)
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	X	Indeterminate
1	1	1	1	X	
0	X	X	0	0	No change (NC)
0	X	X	1	1	

Table 4 Truth table for SR latch with enable input

JK :

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q}_n$

Truth table 5

Q.7 a) *Realize a 3 bit binary synchronous up counter using JK flip flop. Write the excitation table, transition table and logic diagram. Include preset, clear option.* [10]

Sol. : 3 - bit binary synchronous up counter using JK flip-flops :

Fig. 12 (a) shows 3-bit synchronous binary counter and its timing diagram. The state sequence for this counter is shown in Table 6.

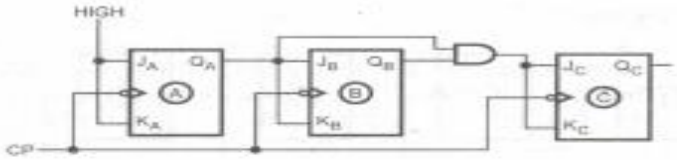


Fig. 12 (a) A three-bit synchronous binary counter

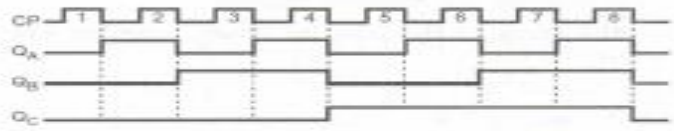


Fig. 12 (b) Timing diagram for 3-bit synchronous binary counter

CP	Q_C	Q_B	Q_A
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

Table 6 State sequence for 3-bit binary counter

Looking at Fig. 12 (b), we can see that Q_A changes on each clock pulse as we progress from its original state to its final state and then back to its original state. To produce this operation, flip-flop A is held in the toggle mode by connecting J and K inputs to HIGH. Now let us see what flip-flop B does. Flip-flop B toggles, when Q_A is 1. When Q_A is a 0, flip-flop B is in the no-change mode and remains in its present state. Looking at the Table 6 we can notice that flip-flop C has to change its state only when Q_B and Q_A both are at logic 1. This condition is detected by AND gate and