Module 3:

Lecturer : Yongsheng Gao

Room : Tech - 3.25

Email : yongsheng.gao@griffith.edu.au

Structure : 6 lectures
1 Tutorial

Assessment: 1 Laboratory (5%)
1 Test (20%)

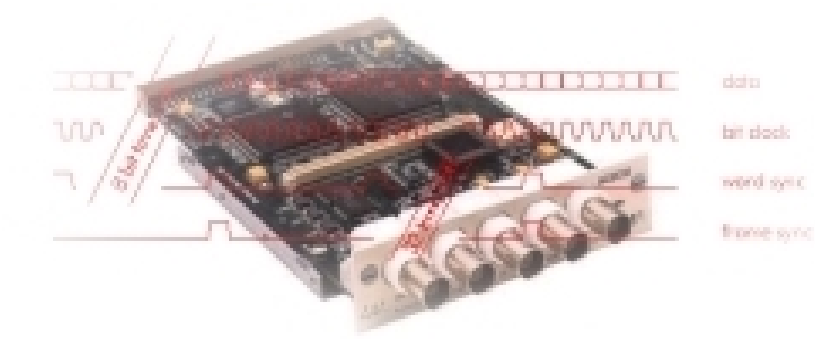
Textbook : Floyd, Digital Fundamental

Chapters : Chapter 9
Chapter 10

Synopsis : Asynchronous Counters, Synchronous Counters, Design of Synchronous Counters, Shift Registers, Johnson & ring counters, Applications etc.

Aim:

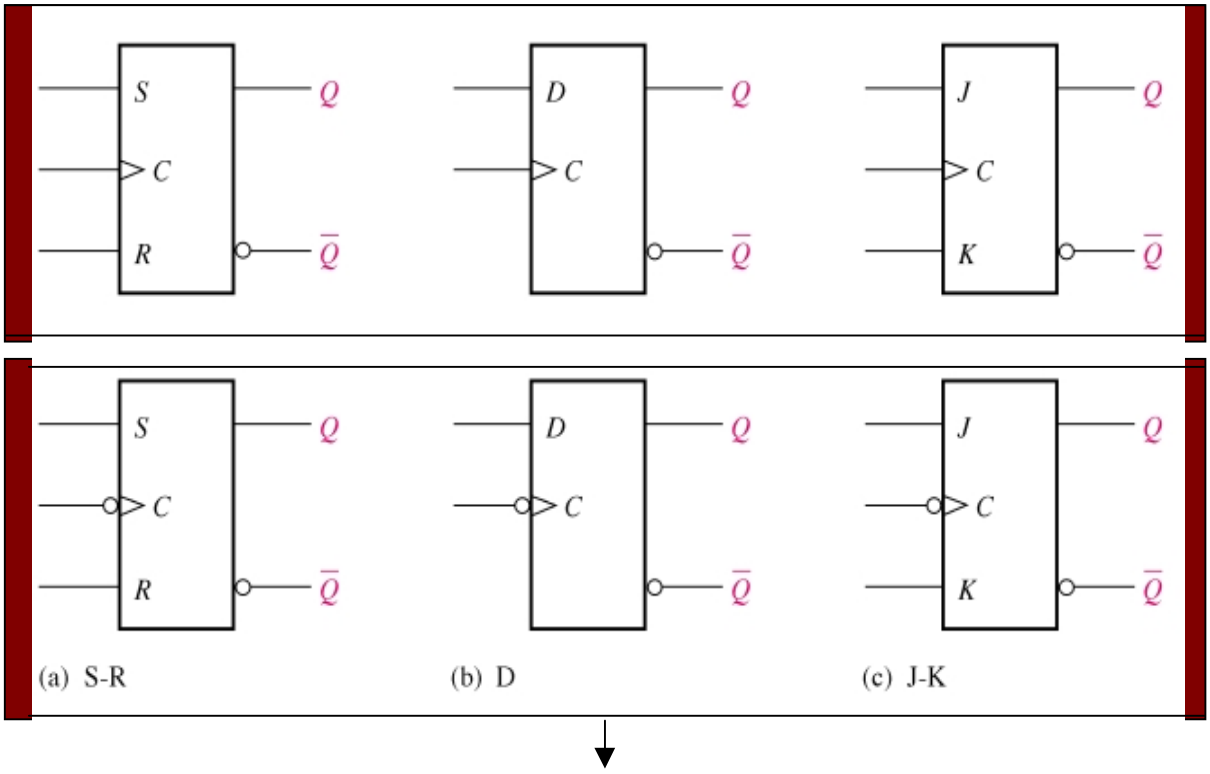
In the previous lectures, you have learnt D, S-R, J-K flip-flops. These flip-flops can be connected together to perform certain operations. In the following lectures, we will focus on a variety of sequential circuits used mostly as storage elements: registers, counters. We look at how to construct the different types of counter. We will also look at how to build registers for storage and shifting from simple D-type flip-flops sharing common clock lines.



Review of Flip-Flops

Flip-flops are synchronous bistable devices. The term synchronous means the output changes state only when the clock input is triggered. That is, changes in the output occur in synchronization with the clock.

Positive edge-triggered (without bubble at Clock input):
S-R, J-K, and D.



Negative edge-triggered (with bubble at Clock input):
S-R, J-K, and D.

An edge-triggered flip-flop changes states either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse on the control input. The three basic types are introduced here: S-R, J-K and D.

The S-R, J-K and D inputs are called synchronous inputs because data on these inputs are transferred to the flip-flop's output only on the triggering edge of the clock pulse. On the other hand, the direct set (SET) and clear (CLR) inputs are called asynchronous inputs, as they are inputs that affect the state of the flip-flop independent of the clock. For the synchronous operations to work properly, these asynchronous inputs must both be kept LOW.

Edge-triggered S-R flip-flop

The basic operation is illustrated below, along with the truth table for this type of flip-flop. The operation and truth table for a negative edge-triggered flip-flop are the same as those for a positive except that the falling edge of the clock pulse is the triggering edge.

Note that the S and R inputs can be changed at any time when the clock input is LOW or HIGH (except for a very short interval around the triggering transition of the clock) without affecting the output.

Edge-triggered J-K flip-flop

The J-K flip-flop works very similar to S-R flip-flop. The only difference is that this flip-flop has NO invalid state. The outputs toggle (change to the opposite state) when both J and K inputs are HIGH.

Edge-triggered D flip-flop

The operations of a D flip-flop is much more simpler. It has only one input addition to the clock. It is very useful when a single data bit (0 or 1) is to be stored. If there is a HIGH on the D input when a clock pulse is applied, the flip-flop SETs and stores a 1. If there is a LOW on the D input when a clock pulse is applied, the flip-flop RESETs and stores a 0. The truth table below summarize the operations of the positive edge-triggered D flip-flop. As before, the negative edge-triggered flip-flop works the same except that the falling edge of the clock pulse is the triggering edge.

Counters

1. Introduction

A counter is a sequential machine that produces a specified count sequence. The count changes whenever the input clock is asserted.

There is a great variety of counter based on its construction.

1. Clock: Synchronous or Asynchronous
2. Clock Trigger: Positive edged or Negative edged
3. Counts: Binary, Decade
4. Count Direction: Up, Down, or Up/Down
5. Flip-flops: JK or T or D

- A counter can be constructed by a synchronous circuit or by an asynchronous circuit. With a synchronous circuit, all the bits in the count change synchronously with the assertion of the clock. With an asynchronous circuit, all the bits in the count do not all change at the same time.

- A counter may count up or count down or count up and down depending on the input control.

- Because of limited word length, the count sequence is limited. For an n-bit counter, the range of the count is [0, 2^n-1]. The count sequence usually repeats itself. When counting up, the count sequence goes in this manner: 0, 1, 2, … 2^n-2, 2^n-1, 0, 1, …etc. When counting down the count sequence goes in the same manner: 2^n-1, 2^n-2, … 2, 1, 0, 2^n-1, 2^n-2, … etc.

  Example:

3-bit Up Counter	3-bit Down Counter
000	000
001	111
010	110
011	101
100	100
101	011
110	010
111	001

- The complement of the count sequence counts in reverse direction. If the uncomplemented output counts up, the complemented output counts down. If the uncomplemented output counts down, the complemented output counts up.

  Example:

3-bit Up Counter	Complement of the Count
000	111
001	110
010	101
011	100
100	011
101	010
110	001
111	000

- The natural count sequence is to run through all possible combinations of the bit patterns before repeating itself. External logic can be used to arbitrary cause the counter to start at any count and terminate at any count.

- A binary counter produces a count sequence similar to the binary numbers. A decade counter counts from 0 to 9, thus making it suitable for human interface. Other counters count to 12 making them suitable for clocks.

1.1 Uses of Counters

The most typical uses of counters are

- To count the number of times that a certain event takes place; the occurrence of event to be counted is represented by the input signal to the counter (Fig. 1.1a)
- To control a fixed sequence of actions in a digital system (Fig. 1.1b)
- To generate timing signals (Fig. 1.2a)
- To generate clocks of different frequencies (Fig. 1.2b)

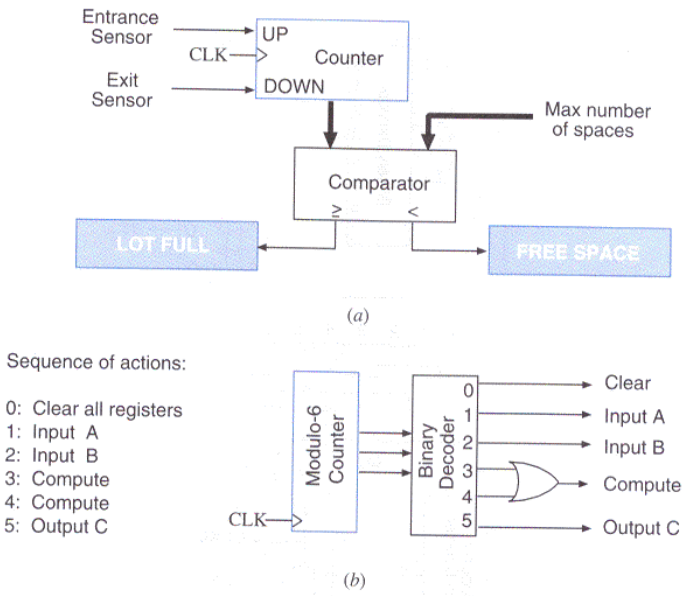


Figure 1.1

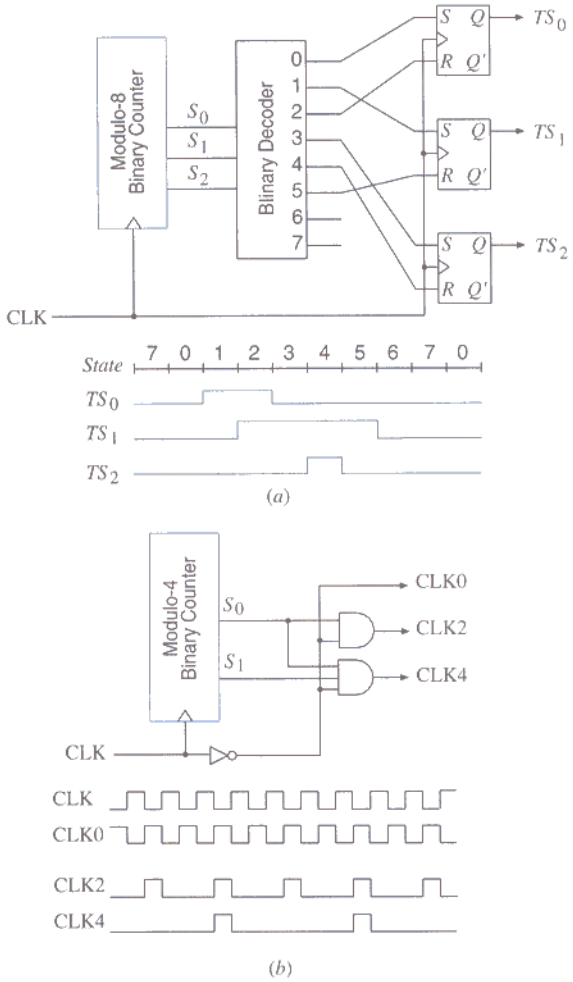


Figure 1.2 [Milos]

1.2 Two Classes of Counters

- Counters are classified into two categories:

    - Asynchronous Counters (Ripple counters)
    - Synchronous Counters

  Asynchronous & Synchronous

  Asynchronous: The events do not have a fixed time relationship with each other and do not occur at the same time.

  Synchronous: The events have a fixed time relationship with each other and do occur at the same time.

- Counters are classified according to the way they are clocked. In asynchronous counters, the first flip-flop is clocked by the external clock pulse and then each successive flip-flop is by clocked the output of the preceding flip-flop. In synchronous counters, the clock input is connected to all of the flip-flop so that they are clocked simultaneously.

2.0 Asynchronous Counters

An asynchronous counter is one in which the flip-flop within the counter do not change states at exactly the same time because they do not have a common clock pulse.

- 2 Bit asynchronous binary counter
- 3 Bit asynchronous binary counter
- 4 Bit asynchronous binary counter

The main characteristic of an asynchronous counter is each flip-flop derives its own clock from other flip-flops and is therefore independent of the input clock. Consequently, the output of each flip-flop may change at different time, hence the term asynchronous. From the asynchronous counter diagram above, we observed that the output of the first flip-flop becomes the clock input for the second flip-flop, and the output of the second flip-flop becomes the clock input for the third flip-flop etc.

For the first flip-flop, the output changes whenever there is a negative transition in the clock input. This means that the output of the first flip-flop produces a series of square waves that is half the frequency of the clock input. Since the output of the first flip-flop becomes the clock of the second flip-flop, the output of the second flip-flop is half the frequency of its clock, i.e. the output of the first flip-flop that in turn is half the frequency of the clock input. This behaviour, in essence is captured by the binary bit pattern in the counting sequence

2.1 2-Bit Asynchronous Binary Counter

Example 1:

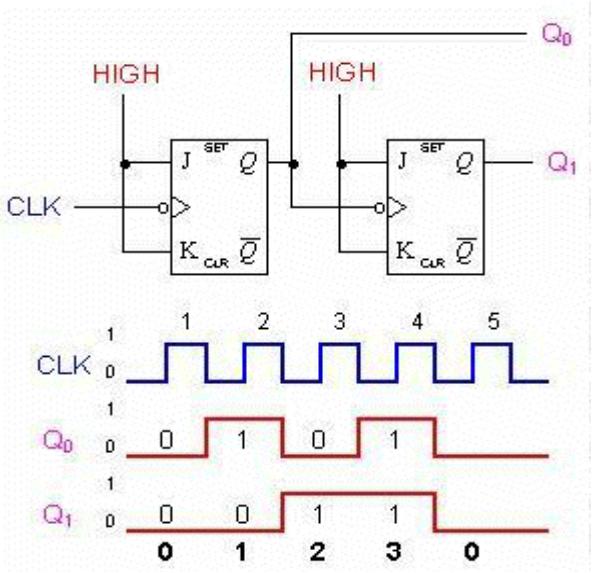


Figure 1.3a: two-bit asynchronous counter

- A two-bit asynchronous counter is shown on the left. The external clock is connected to the clock input of the first flip-flop (FF0) only. So, FF0 changes state at the falling edge of each clock pulse, but FF1 changes only when triggered by the falling edge of the Q output of FF0. Because of the inherent propagation delay through a flip-flop, the transition of the input clock pulse and a transition of the Q output of FF0 can never occur at exactly the same time. Therefore, the flip-flops cannot be triggered simultaneously, producing an asynchronous operation.

- Note that for simplicity, the transitions of Q0, Q1 and CLK in the timing diagram above are shown as simultaneous even though this is an asynchronous counter. Actually, there is some small delay between the CLK, Q0 and Q1 transitions.

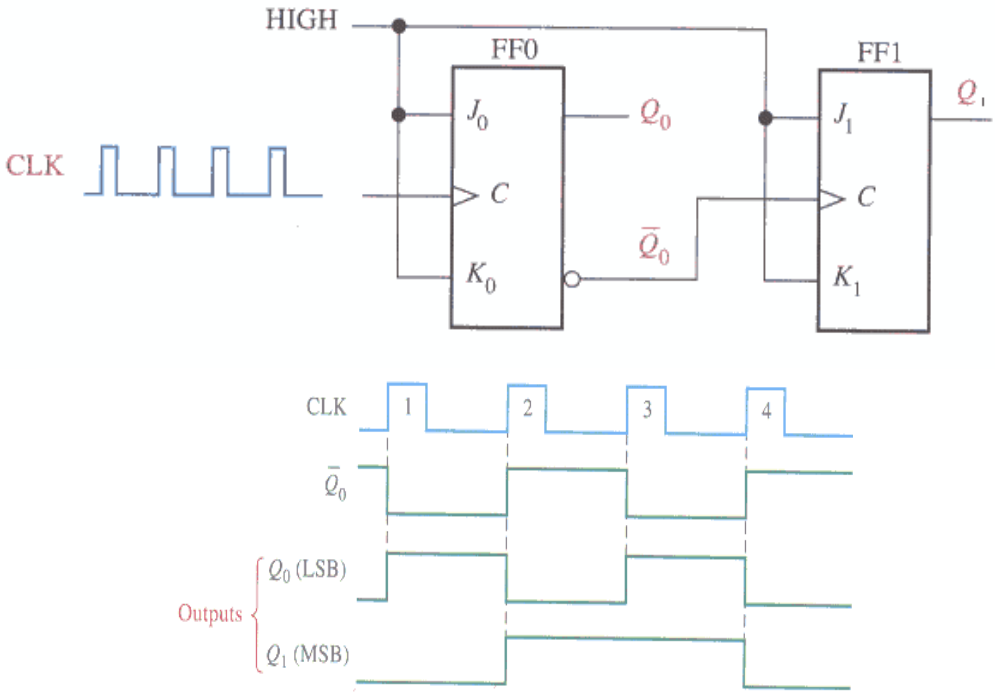
- Usually, all the CLEAR inputs are connected together, so that a single pulse can clear all the flip-flops before counting starts. The clock pulse fed into FF0 is rippled through the other counters after propagation delays, like a ripple on water, hence the name Ripple Counter.

- The 2-bit ripple counter circuit above has four different states, each one corresponding to a count value. Similarly, a counter with n flip-

flops can have 2 to the power n states. The number of states in a counter is known as its mod (modulo) number. Thus a 2-bit counter is a mod-4 counter.

- A mod-n counter may also be described as a divide-by-n counter. This is because the most significant flip-flop (the furthest flip-flop from the original clock pulse) produces one pulse for every n pulses at the clock input of the least significant flip-flop (the one triggers by the clock pulse). Thus, the above counter is an example of a divide-by-4 counter.

Example 2:



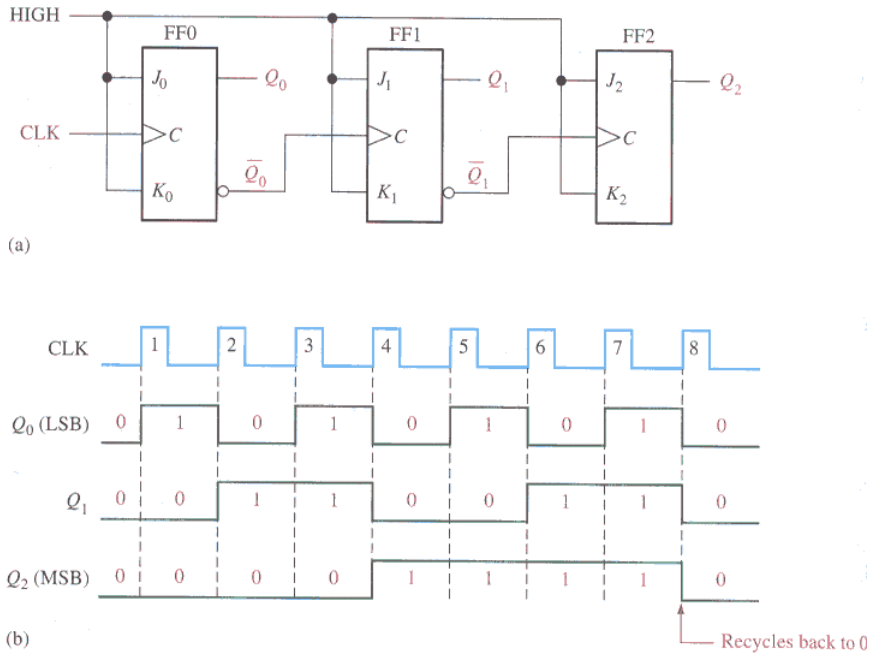
Clock Pulse	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

[Floyd]

Figure 1.3b: Two-bit asynchronous binary counter, timing diagram, binary state sequence

2.2 3-Bit Asynchronous Binary Counter

The following is a three-bit asynchronous binary counter and its timing diagram for one cycle. It works exactly the same way as a two-bit asynchronous binary counter mentioned above, except it has eight states due to the third flip-flop.

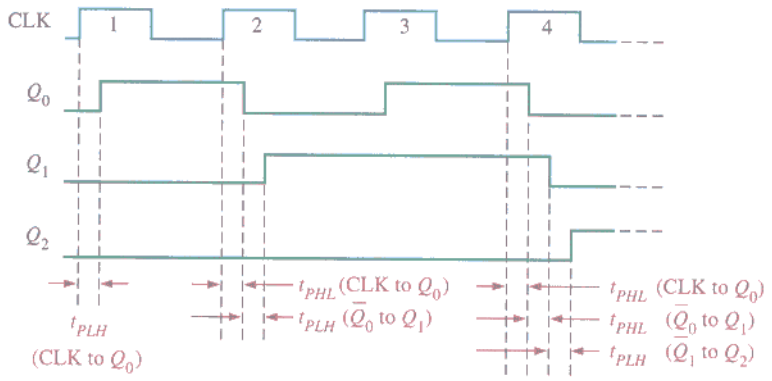


Clock Pulse	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0

[Floyd]

Figure 1.4a: Three-bit asynchronous binary counter, timing diagram, binary state sequence

Propagation Delay:



[Floyd]

Figure 1.4b: Propagation Delay in a 3-bit asynchronous binary counter

Asynchronous counters are commonly referred to as ripple counters for the following reason: The effect of the input clock pulse is first "felt" by FFO. This effect cannot get to FF1 immediately because of the propagation delay through FF0. Then there is the propagation delay through FF1 before FF2 can be triggered. Thus, the effect of an input clock pulse "ripples" through the counter, taking some time, due to propagation delays, to reach the last flip-flop.

2.3 4 Bit Asynchronous Binary Counter

The following is a 4-bit asynchronous binary counter and its timing diagram for one cycle. It works exactly the same way as a 2-bit or 3 bit asynchronous binary counter mentioned above, except it has 16 states due to the fourth flip-flop.

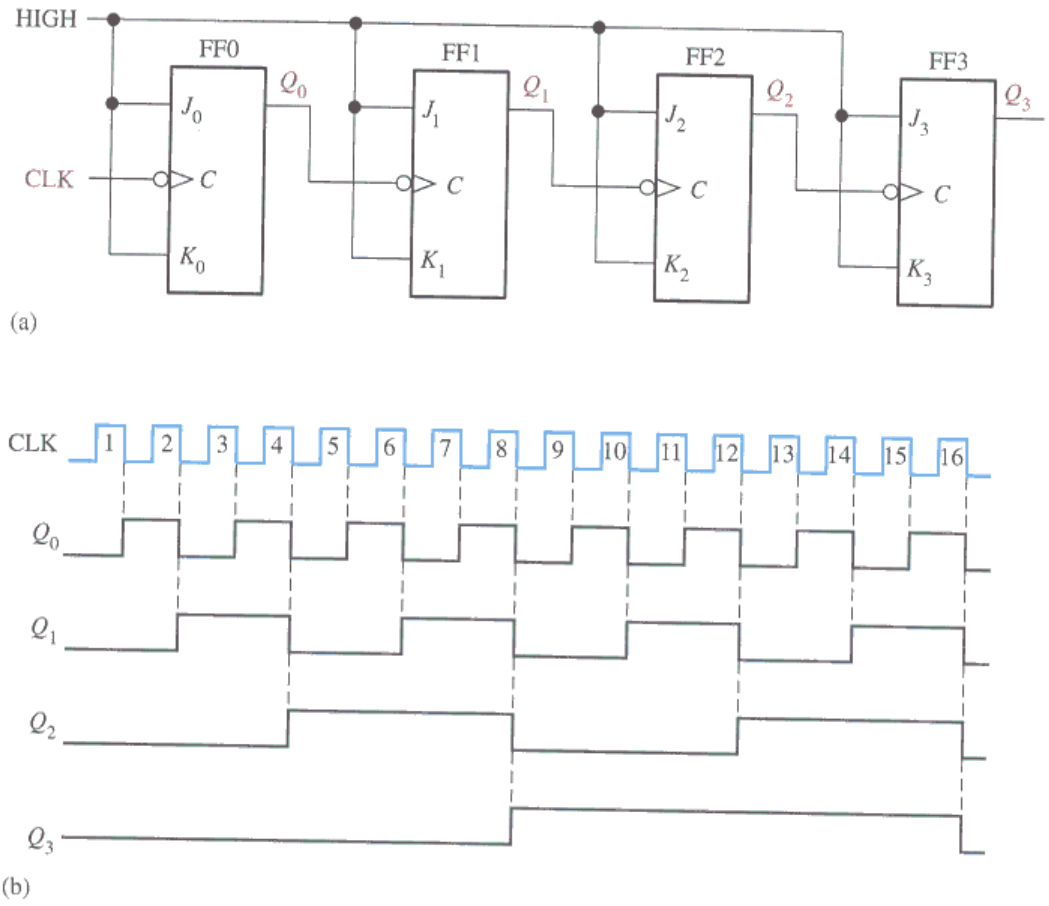


Figure 1.5: Four-bit asynchronous binary counter, timing diagram [Floyd]

2.4 Asynchronous Decade Counters

- The binary counters previously introduced have two to the power n states. But counters with states less than this number are also possible. They are designed to have the number of states in their sequences, which are called truncated sequences. These sequences are achieved by forcing the counter to recycle before going through all of its normal states.

- A common modulus for counters with truncated sequences is ten. A counter with ten states in its sequence is called a decade counter.

- The circuit below is an implementation of a decade counter.

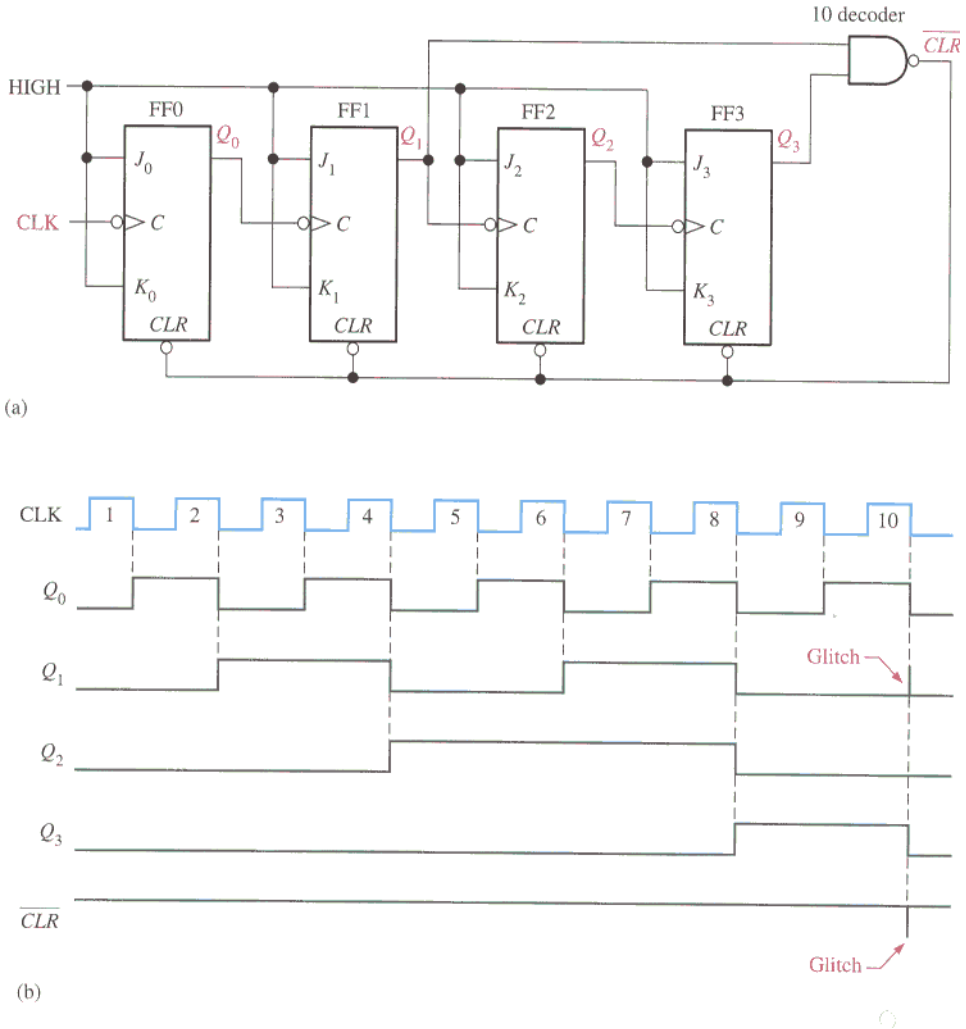


Figure 1.6: Asynchronous decade counter, timing diagram [Floyd]

- Once the counter counts to ten (1010), all the flip-flops are being cleared. Notice that only Q1 and Q3 are used to decode the count of ten. This is called partial decoding, as none of the other states (zero to nine) have both Q1 and Q3 HIGH at the same time.

- The sequence of the decade counter is shown in the table below:

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

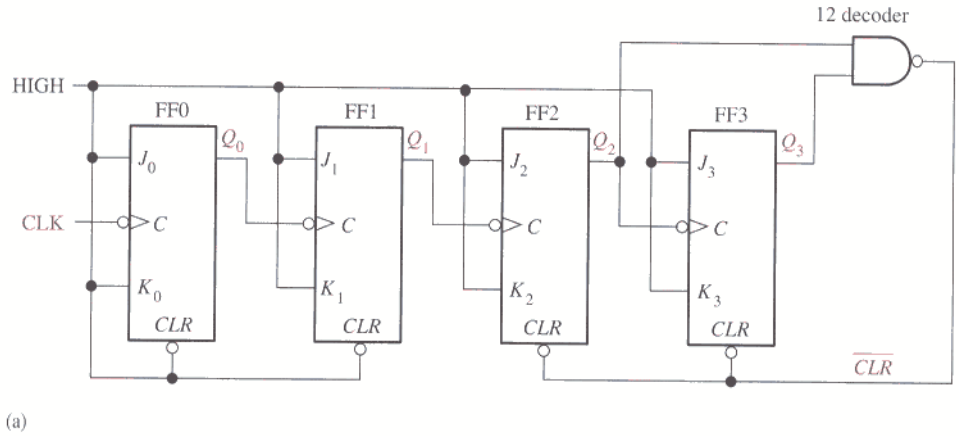
Recycles

1 1 0 0 (normal next state)

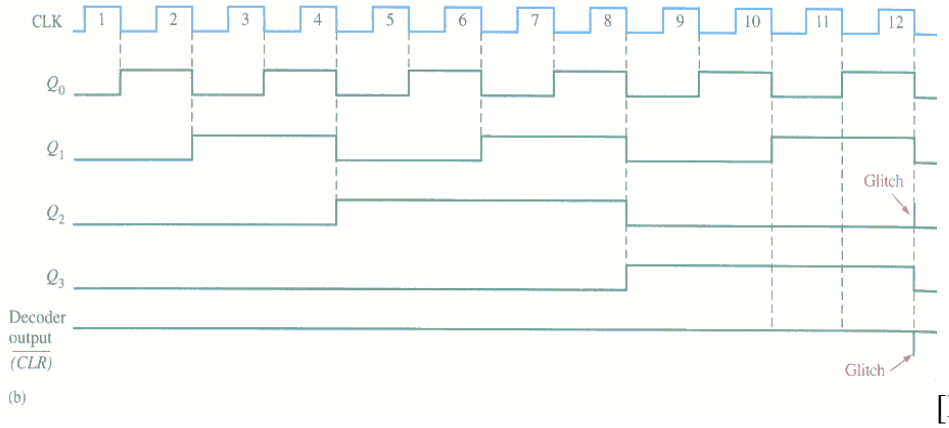
Glitch: Notice that there is a glitch on the Q_1 waveform. The reason for this glitch is that Q_1 must first go HIGH before the count of ten can be decoded. Not until several nanoseconds after the counter goes to the count of ten does the output of the decoding gate go LOW (both inputs are HIGH). Therefore, the counter is in the 1010 state for a short time before it is reset to 0000, thus producing the glitch on Q_1 and the resulting glitch on the $\overline{CLR}$ line that resets the counter.

Example: Modulus Twelve Asynchronous Counter

An Asynchronous counter can be implemented having a modulus of 12 with a straight binary sequence from 0000 through 1011.



(a)



(b)

[Floyd]

Figure 1.7: Asynchronous modulus-12 counter & timing diagram.

2.5 Asynchronous Up-Down Counters

In certain applications a counter must be able to count both up and down. The circuit below is a 3-bit up-down counter. It counts up or down depending on the status of the control signals UP and DOWN. When the UP input is at 1 and the DOWN input is at 0, the NAND network between FF0 and FF1 will gate the non-inverted output (Q) of FF0 into the clock input of FF1. Similarly, Q of FF1 will be gated through the other NAND network into the clock input of FF2. Thus the counter will count up.

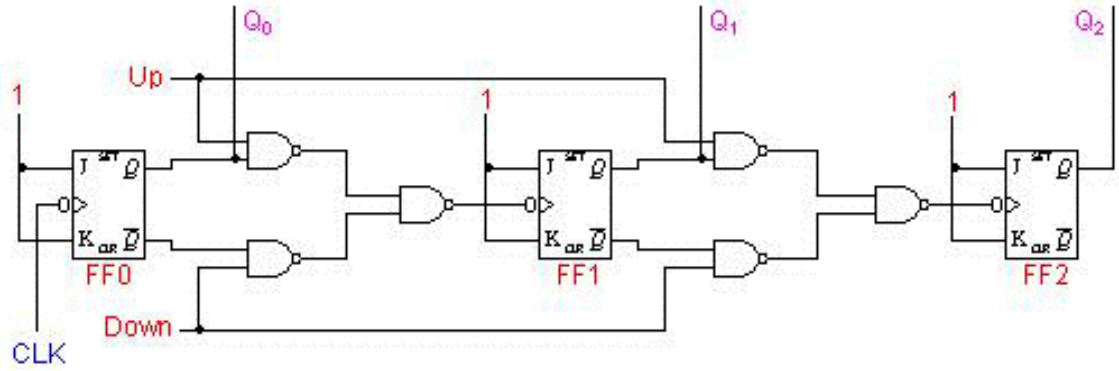


Figure 1.8: 3-bit up-down counter

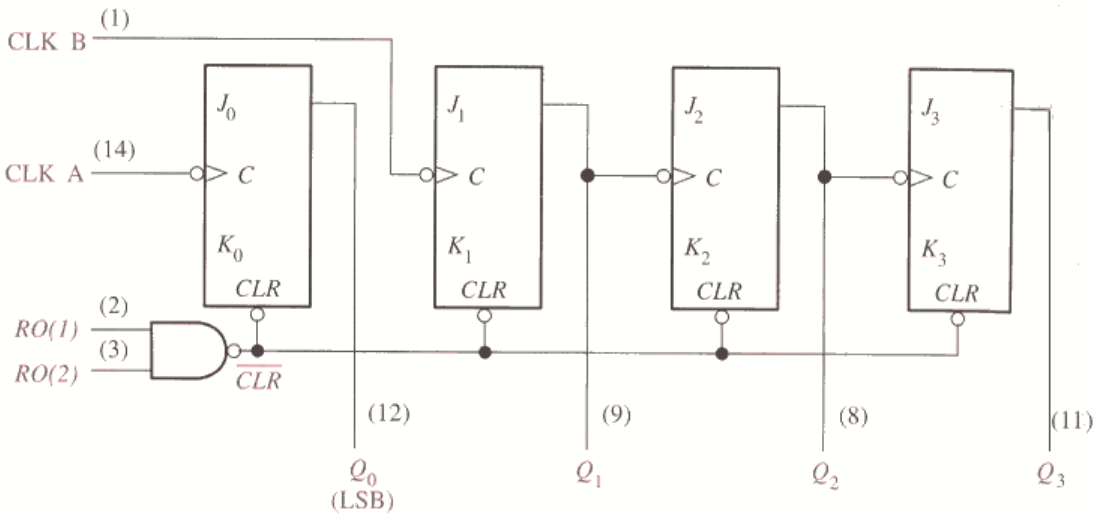
- When the control input UP is at 0 and DOWN is at 1, the inverted outputs of FF0 and FF1 are gated into the clock inputs of FF1 and FF2 respectively. If the flip-flops are initially reset to 0's, then the counter will go through the following sequence as input pulses are applied.

FF2	FF1	FF0
0	0	0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1

- Notice that an asynchronous up-down counter is slower than an up counter or a down counter because of the additional propagation delay introduced by the NAND networks.

2.6 Commercially Available Asynchronous Counters

Example 1: The 74LS93 Asynchronous Binary Counter

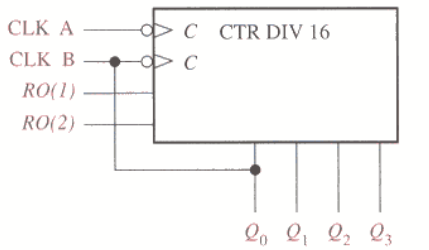


[Floyd]

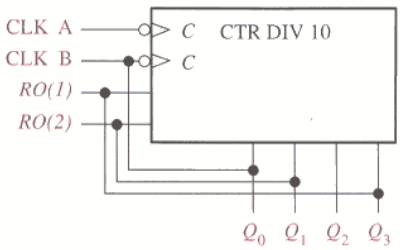
Figure 1.9: The 74LS93A 4-bit asynchronous binary counter logic diagram

Three configurations of the 74LS293 asynchronous counter:

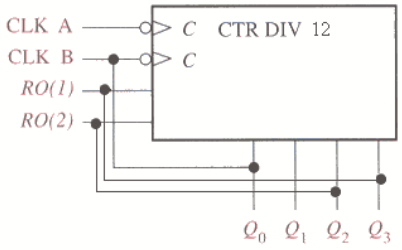
[Floyd]



(a) 74LS93A connected as a modulus-16 counter



(b) 74LS93A connected as a decade counter



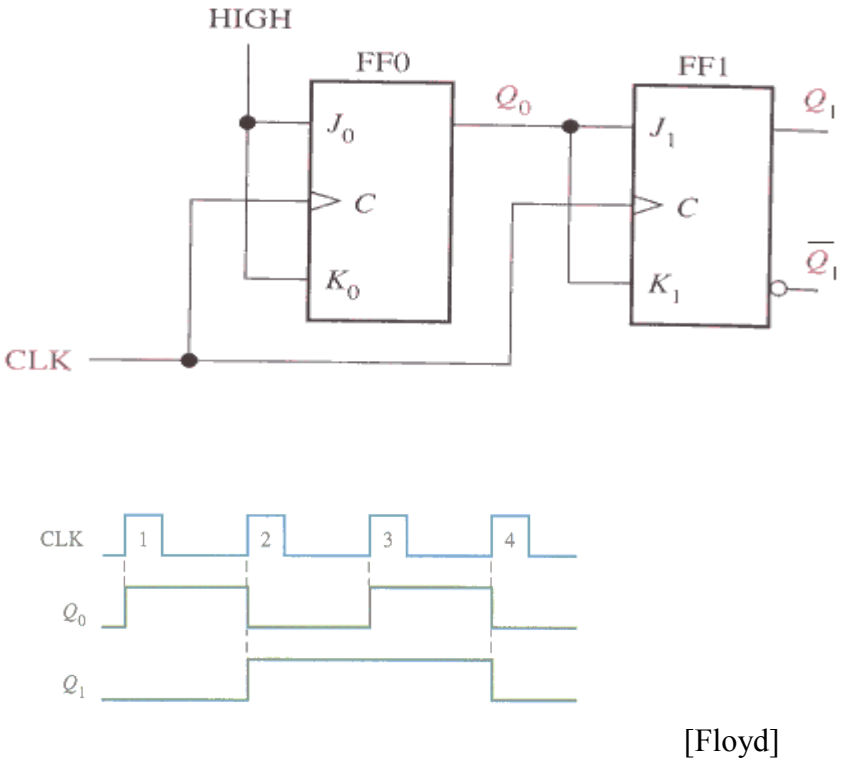
(c) 74LS93A connected as a modulus-12 counter

RO(1), R0(2) are the gated reset inputs. If both of these inputs are HIGH, the counter is reset to the 0000 state by $\overline{CLR}$.

3.0 Synchronous Counters

In synchronous counters, the clock inputs of all the flip-flops are connected together and are triggered by the input pulses. Thus, all the flip-flops change state simultaneously (in parallel).

3.1 2-Bit Synchronous Binary Counter



[Floyd]

Figure 3.1: Two-bit synchronous binary counter, timing diagram

Propagation Delay:

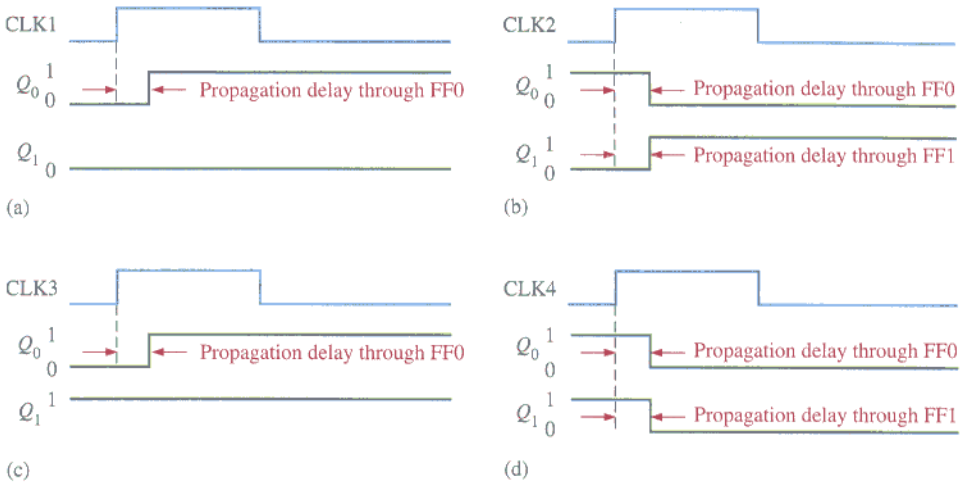


Figure 3.2

3.2 3-Bit Synchronous Binary Counter

The circuit below is a 3-bit synchronous counter. The J and K inputs of FF0 are connected to HIGH. FF1 has its J and K inputs connected to the output of FF0, and the J and K inputs of FF2 are connected to the output of an AND gate that is fed by the outputs of FF0 and FF1.

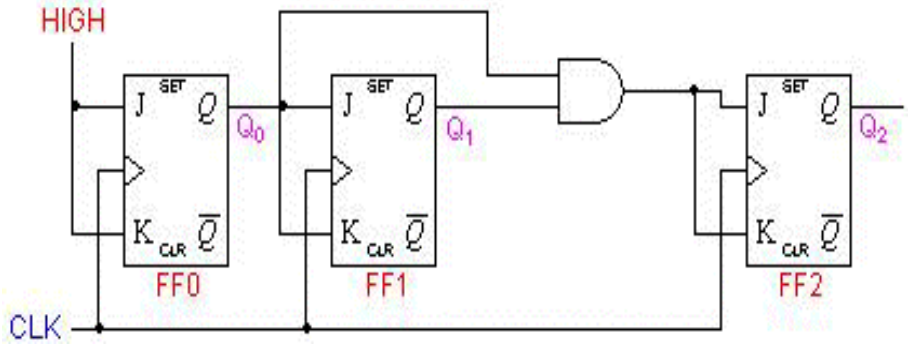


Figure 3.3a: A 3-bit synchronous binary counter

Pay attention to what happens after the 3rd clock pulse. Both outputs of FF0 and FF1 are HIGH. The positive edge of the 4th clock pulse will cause FF2 to change its state due to the AND gate.

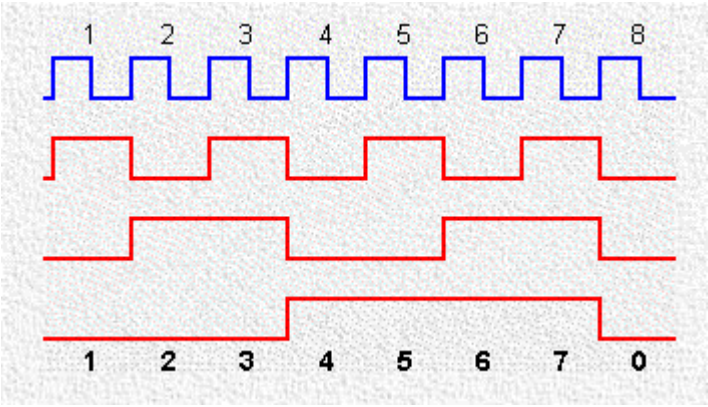


Figure 3.3b: Timing diagram

FF2	FF1	FF0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(recycles)

Figure 3.3c: Binary state sequence

The count sequence for the 3-bit counter is shown in Figure 3.3c.

The most important advantage of synchronous counters is that there is no cumulative time delay because all flip-flops are triggered in parallel. Thus, the maximum operating frequency for this counter will be significantly higher than for the corresponding ripple counter.

3.3 4-Bit Synchronous Binary Counter

Figure 3.4(a) shows a 4-bit synchronous binary counter and Figure 3.4(b) reveals its timing diagram.

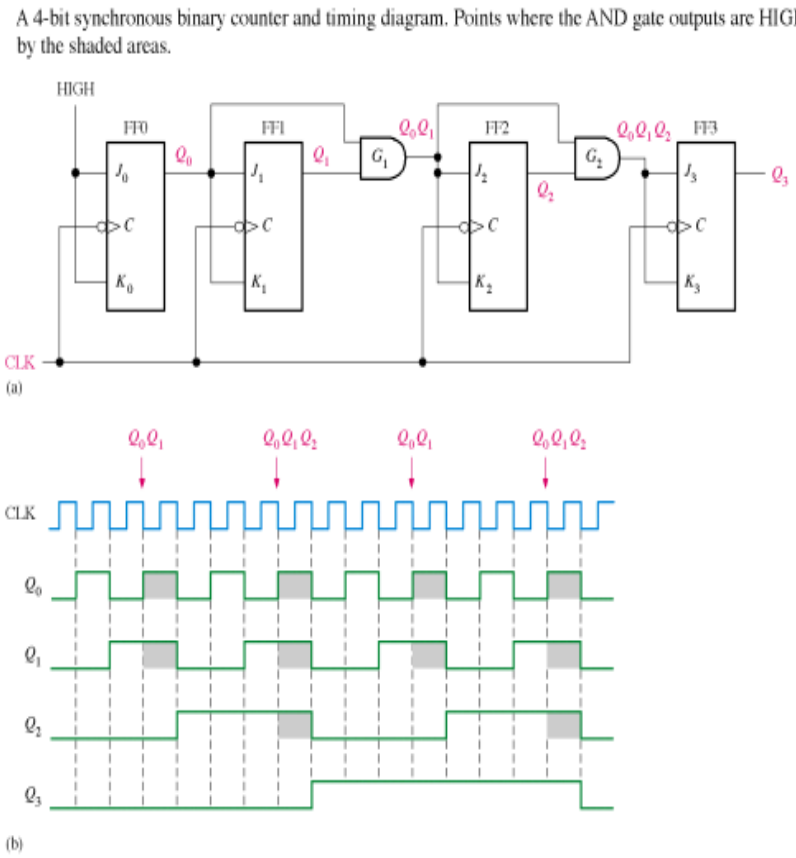


Figure 3.4: Four-bit synchronous binary counter, timing diagram

3.4 Synchronous Decade Counters

Similar to an asynchronous decade counter, a synchronous decade counter counts from 0 to 9 and then recycles to 0 again. This is done by forcing the 1010 state back to the 0000 state. This so called truncated sequence can be constructed by the following circuit.

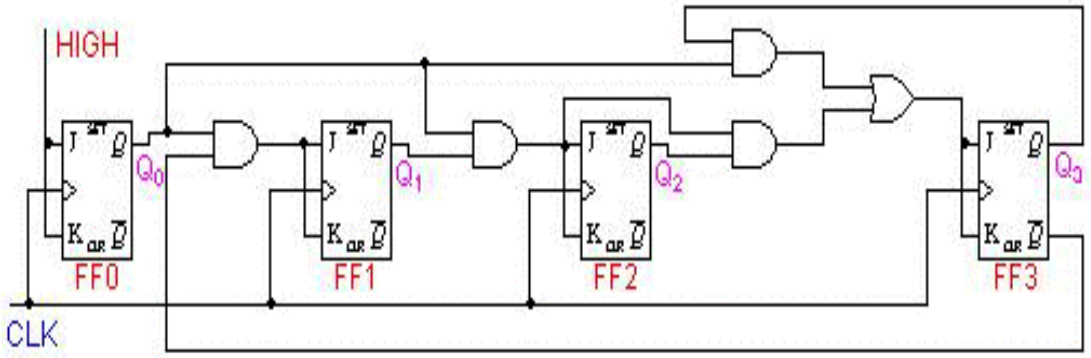


Figure 3.5a: A synchronous BCD decade counter

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Figure 3.5b: States of a BCD decade

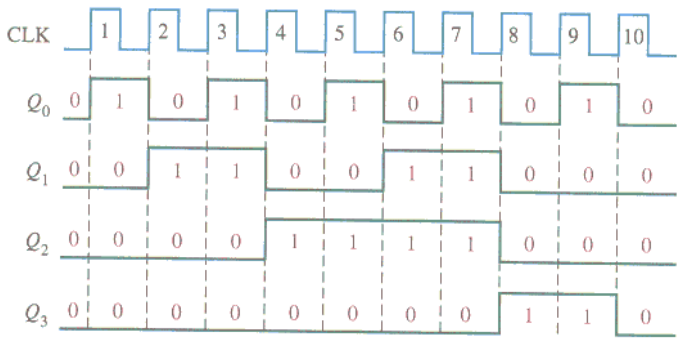


Figure 3.5c: Timing diagram for the BCD decade counter (Q_0 is the LSB)

From the sequence in the Figure 3.5b, we notice that:

- Q_0 toggles on each clock pulse.
- Q_1 changes on the next clock pulse each time $Q_0=1$ and $Q_3=0$.
- Q_2 changes on the next clock pulse each time $Q_0=Q_1=1$.
- Q_3 changes on the next clock pulse each time $Q_0=1$, $Q_1=1$ and $Q_2=1$ (count 7), or then $Q_0=1$ and $Q_3=1$ (count 9).

Flip-flop 2 (Q_2) changes on the next clock pulse each time both $Q_0=1$ and $Q_1=1$. Thus we must have

$$J_2 = K_2 = Q_0Q_1$$

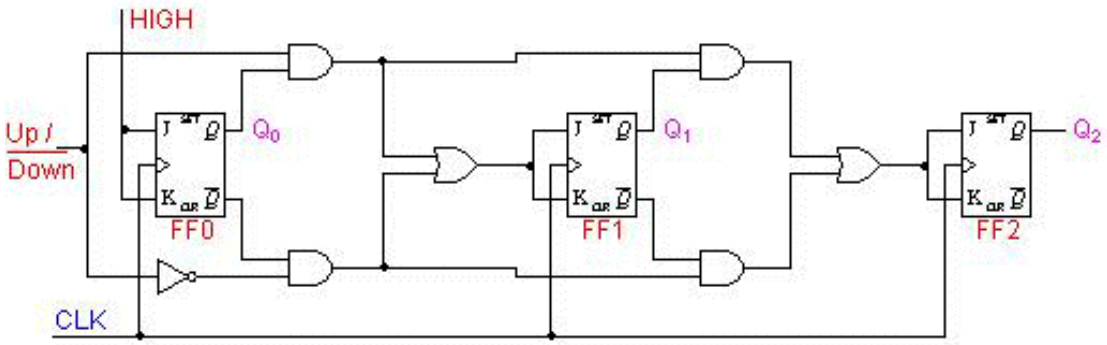
Flip-flop 3 (Q_3) changes to the opposite state on the next clock pulse each time $Q_0=1$, $Q_1=1$, and $Q_2=1$ (state 7), or when $Q_0=1$ and $Q_3=1$ (state 9). Thus we must have

$$J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$$

These characteristics are implemented with the AND/OR logic connected as shown in the logic diagram (Figure 3.5b).

3.5 Up-Down Synchronous Counters

A circuit of a 3-bit synchronous up-down counter and a table of its sequence are shown in Figure 3.6. Similar to an asynchronous up-down counter, a synchronous up-down counter also has an up-down control input. It is used to control the direction of the counter through a certain sequence.



Up

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

Down

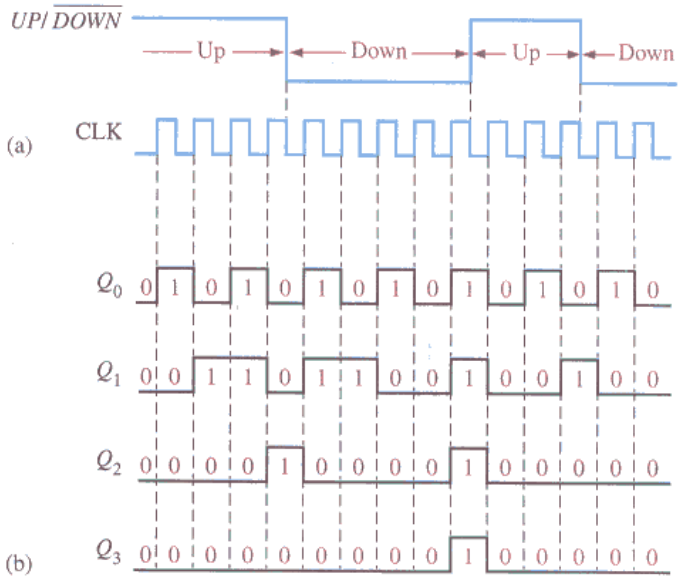
Figure 3.6: A basic 3-bit up/down synchronous counter and its up/down sequence

An examination of the sequence table shows:

- for both the UP and DOWN sequences, Q_0 toggles on each clock pulse.
- for the UP sequence, Q_1 changes state on the next clock pulse when $Q_0=1$.
- for the DOWN sequence, Q_1 changes state on the next clock pulse when $Q_0=0$.
- for the UP sequence, Q_2 changes state on the next clock pulse when $Q_0=Q_1=1$.
- for the DOWN sequence, Q_2 changes state on the next clock pulse when $Q_0=Q_1=0$.

These characteristics are implemented with the AND, OR & NOT logic connected as shown in the Figure 3.6

Example: 4-bit synchronous up-down counter [Floyd]



Q_3	Q_2	Q_1	Q_0	
0	0	0	0	
0	0	0	1	
0	0	1	0	UP
0	0	1	1	
0	1	0	0	
0	0	1	1	
0	0	1	0	
0	0	0	1	DOWN
0	0	0	0	
1	1	1	1	
0	0	0	0	
0	0	0	1	UP
0	0	1	0	
0	0	0	1	DOWN
0	0	0	0	