PGT104 – Digital Electronics

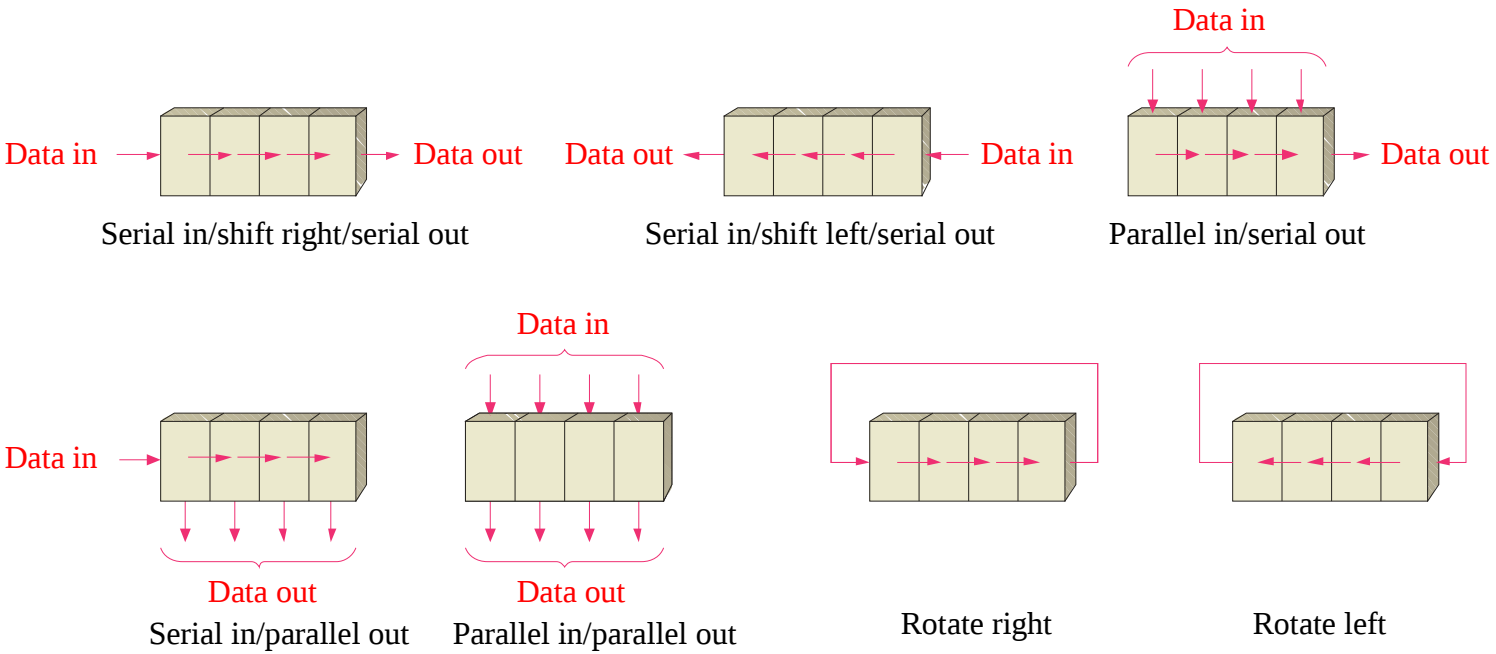
Part 6 – Sequential Logic Circuits

Disclaimer:

- Most of the contents (if not all) are extracted from resources available for Digital Fundamentals 10th Edition

Basic Shift Register Operations

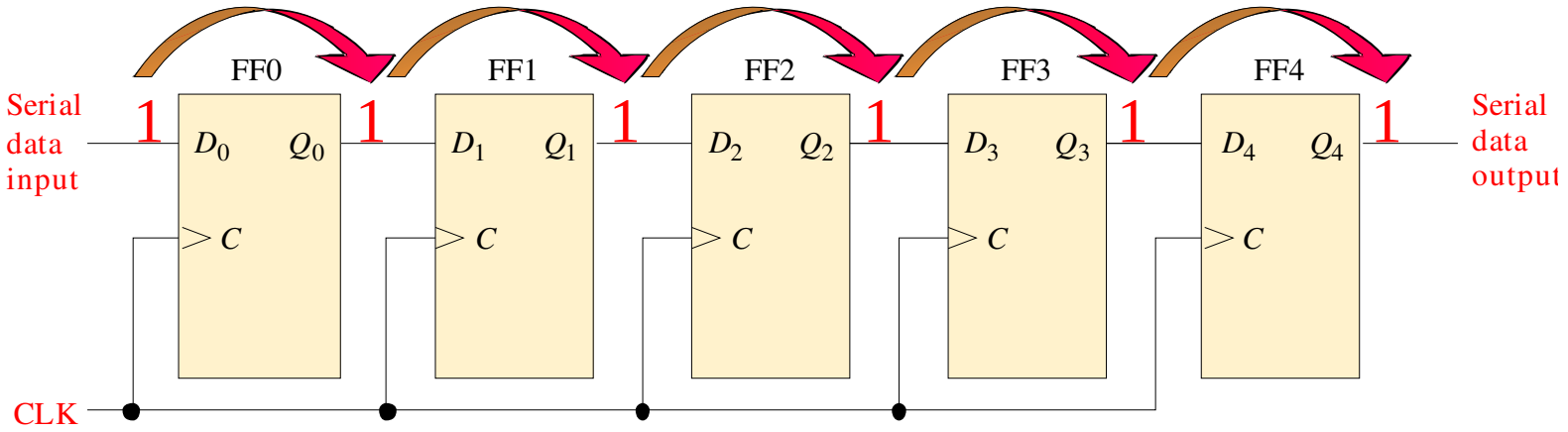
A shift register is an arrangement of flip-flops with important applications in storage and movement of data. Some basic data movements are illustrated here.



Serial-in/Serial out Shift Register

Shift registers are available in IC form or can be constructed from discrete flip-flops as is shown here with a five-bit serial-in serial-out register.

Each clock pulse will move an input bit to the next flip-flop. For example, a 1 is shown as it moves across.

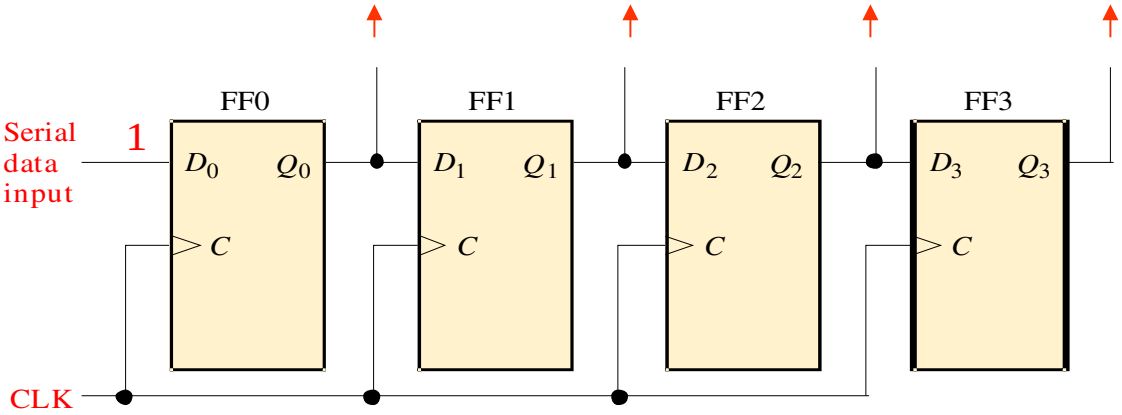


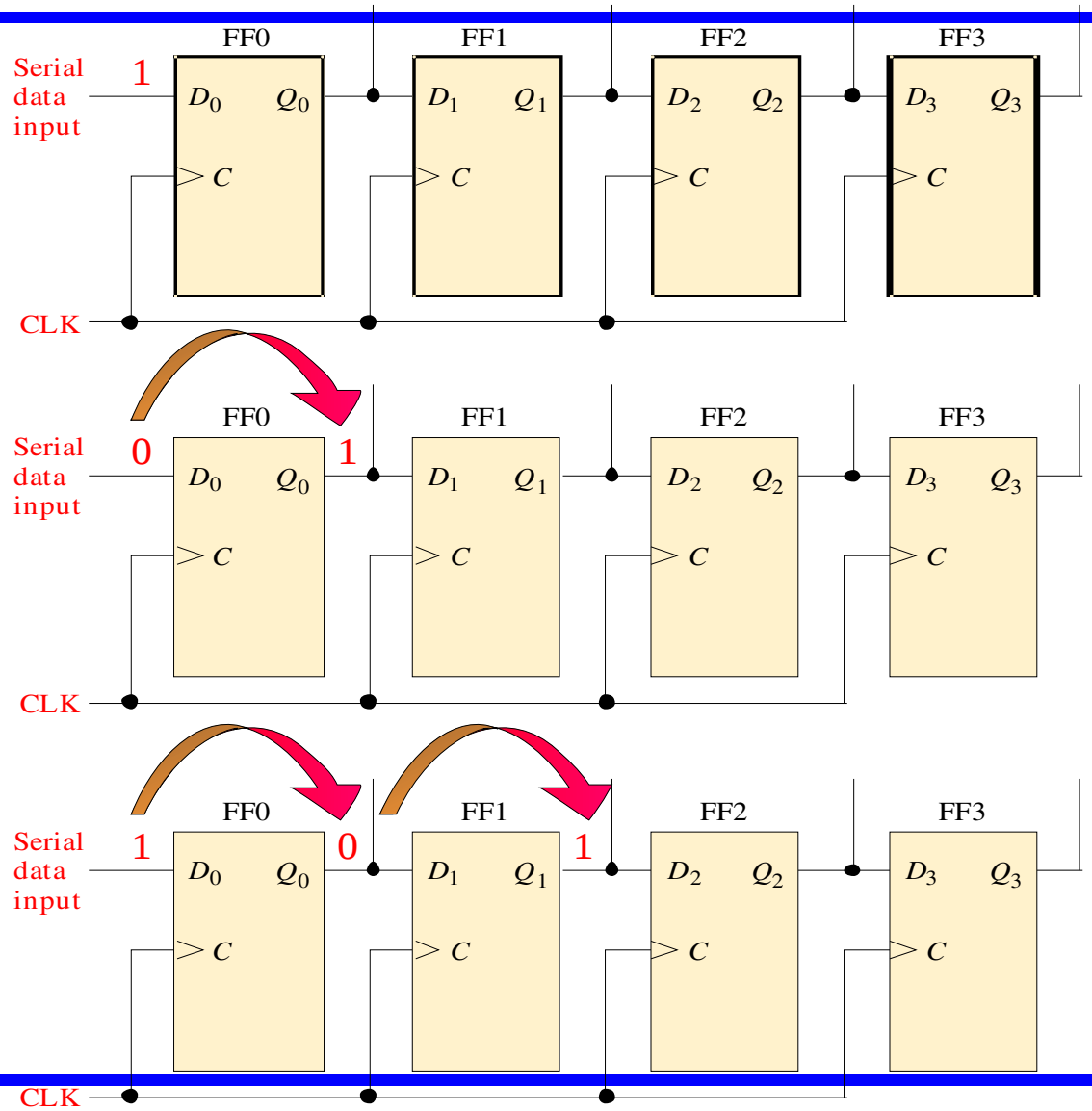
A Basic Application

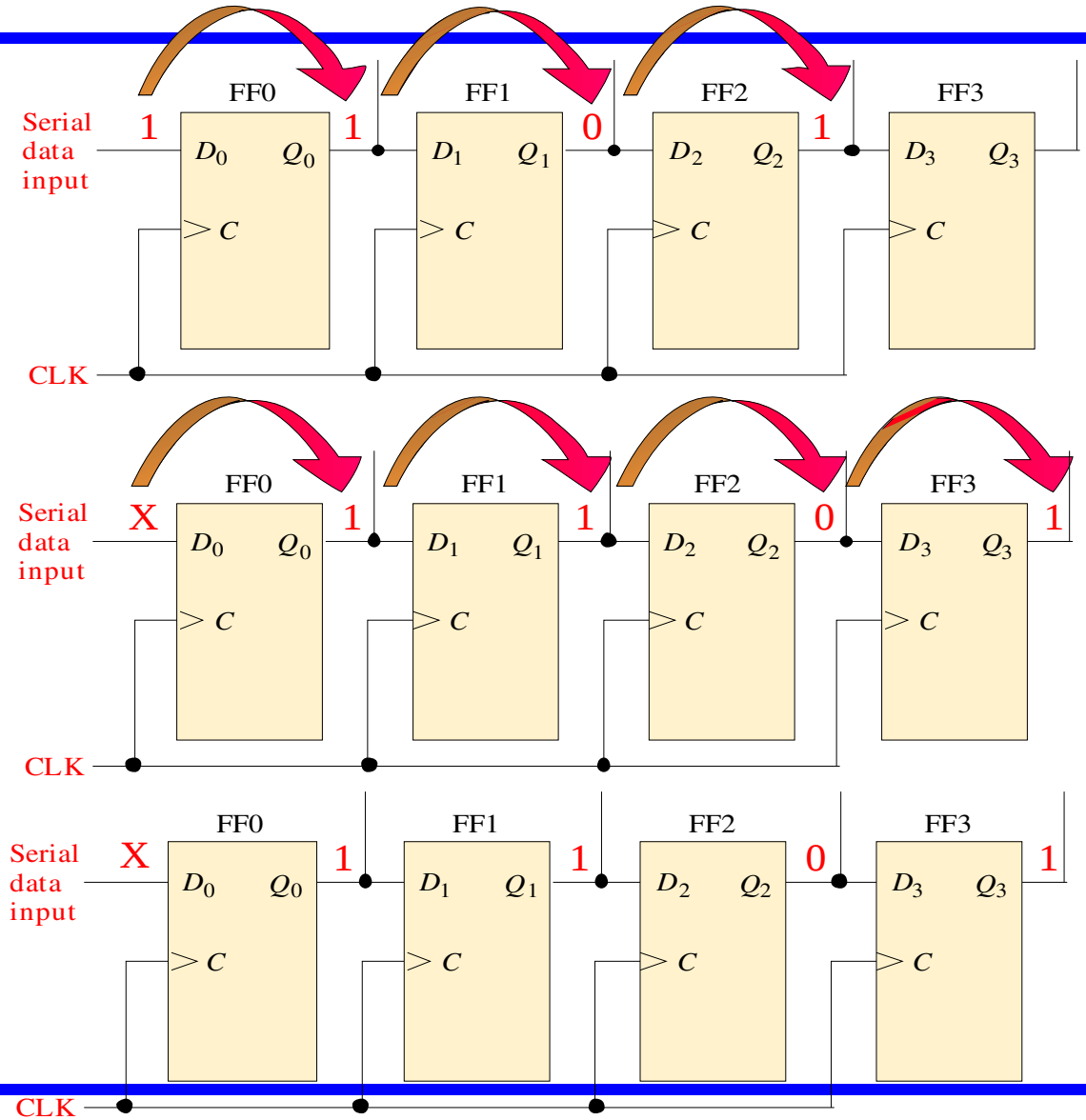
An application of shift registers is conversion of serial data to parallel form.

For example, assume the binary number 1011 is loaded sequentially, one bit at each clock pulse.

After 4 clock pulses, the data is available at the parallel output.

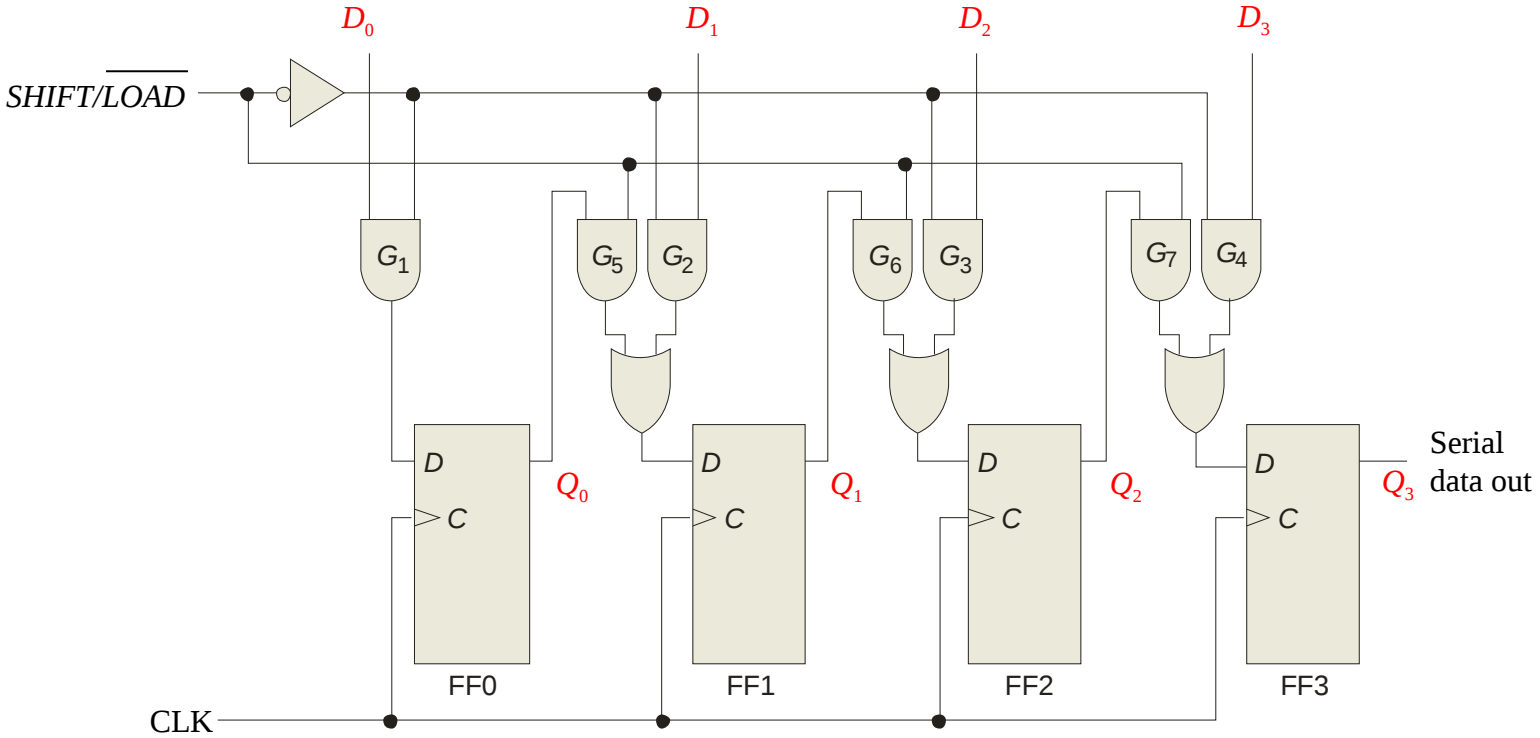



FF0
FF1
FF2
FF3
Serial data input
1
D_0
Q_0
D_1
Q_1
D_2
Q_2
D_3
Q_3
C
CLK
0
1
0
1
1


FF0
FF1
FF2
FF3
Serial data input
1
1
0
1
CLK
FF0
FF1
FF2
FF3
Serial data input
X
1
1
0
1
CLK
FF0
FF1
FF2
FF3
Serial data input
X
1
1
0
1
CLK
D0
Q0
D1
Q1
D2
Q2
D3
Q3
C

Parallel in/Serial out Shift Register

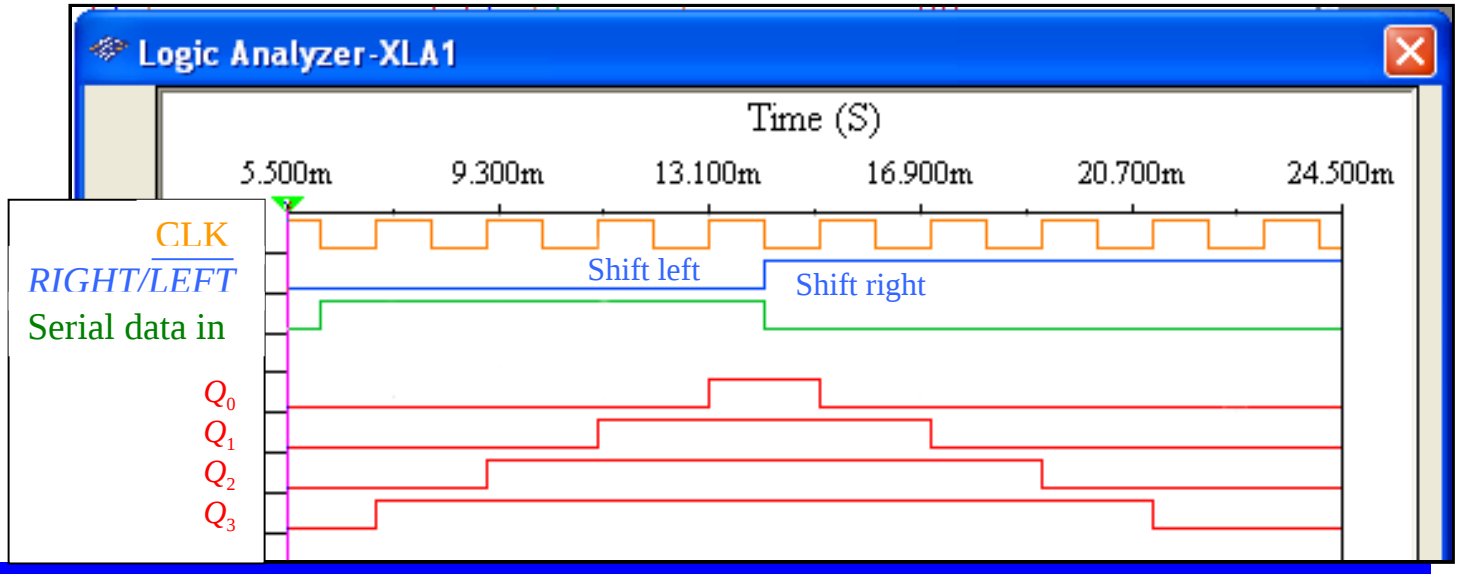
Shift registers can be used to convert parallel data to serial form. A logic diagram for this type of register is shown:



Bidirectional Shift Register

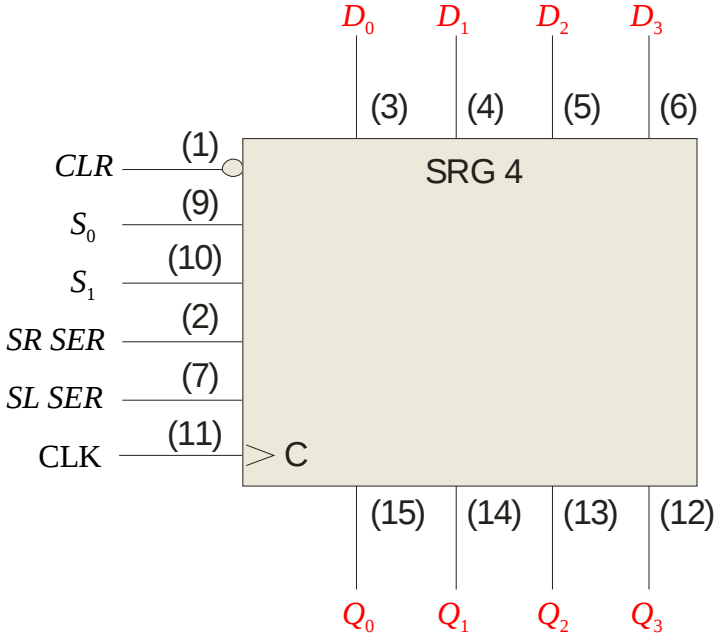
Bidirectional shift registers can shift the data in either direction using a *RIGHT/LEFT* input.

The logic analyzer simulation shows a bidirectional shift register such as the one shown in Figure 9-19 of the text. Notice the HIGH level from the Serial data in is shifted at first from Q_3 toward Q_0.



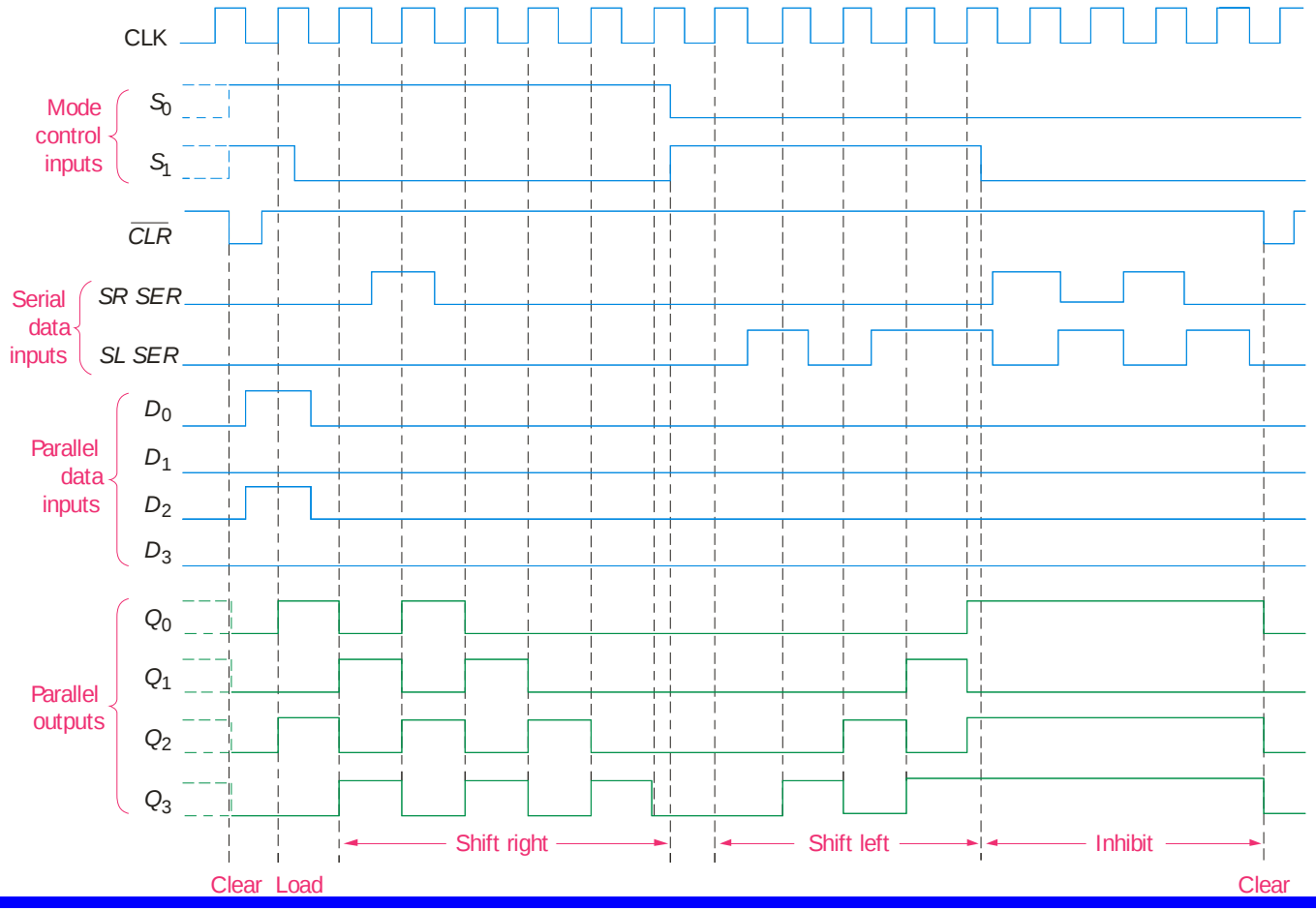
Universal Shift Register

A universal shift register has both serial and parallel input and output capability. The 74HC194 is an example of a 4-bit bidirectional universal shift register.



Sample waveforms are on the following slide…

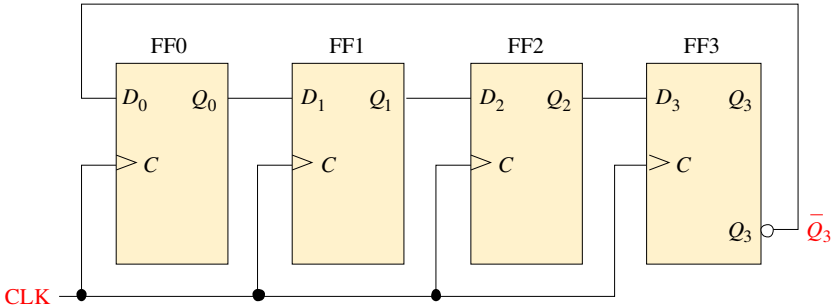
Universal Shift Register



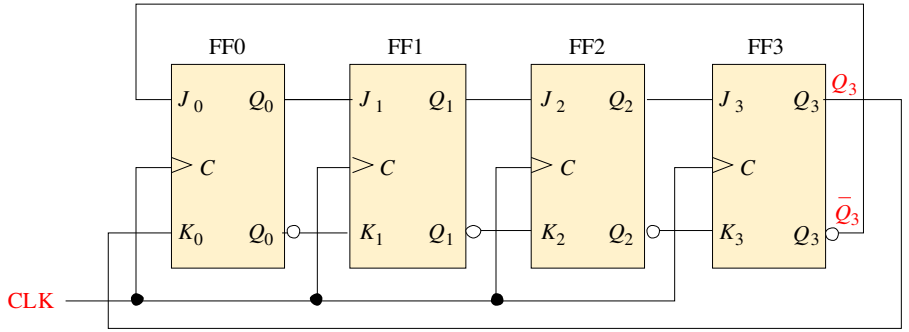
Shift Register Counters

Shift registers can form useful counters by recirculating a pattern of 0's and 1's. Two important shift register counters are the *Johnson counter* and the *ring counter*.

The Johnson counter can be made with a series of D flip-flops



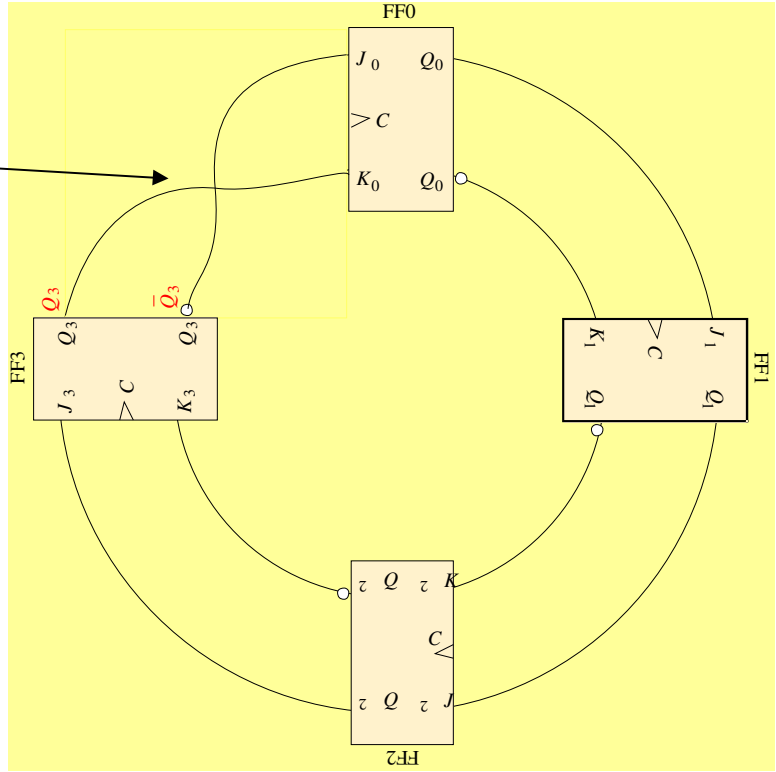
… or with a series of J-K flip flops. Here Q_3 and $\overline{Q}_3$ are fed back to the *J* and *K* inputs with a "twist".



Johnson Counter

Redrawing the same Johnson counter (without the clock shown) illustrates why it is sometimes called as a “twisted-ring” counter.

“twist”



Johnson Counter

The Johnson counter is useful when you need a sequence that changes by only one bit at a time but it has a limited number of states (2*n*, where *n* = number of stages).

The first five counts for a 4-bit Johnson counter that is initially cleared are:

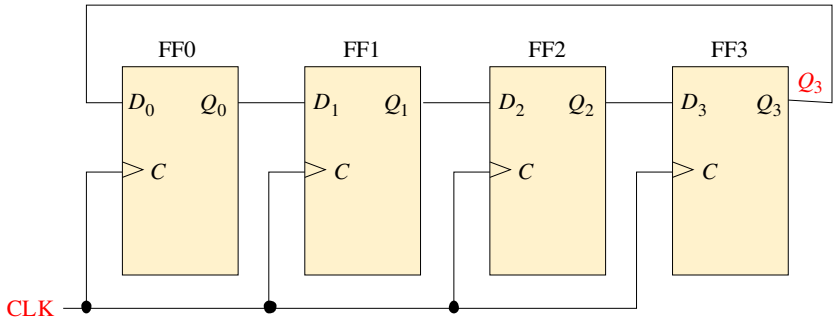
CLK	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1

What are the remaining 3 states?

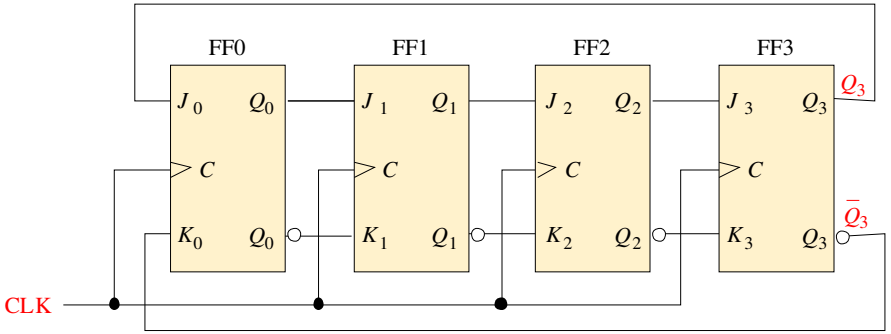
Ring Counter

The ring counter can also be implemented with either D flip-flops or J-K flip-flops.

Here is a 4-bit ring counter constructed from a series of D flip-flops. Notice the feedback.



Like the Johnson counter, it can also be implemented with J-K flip flops.

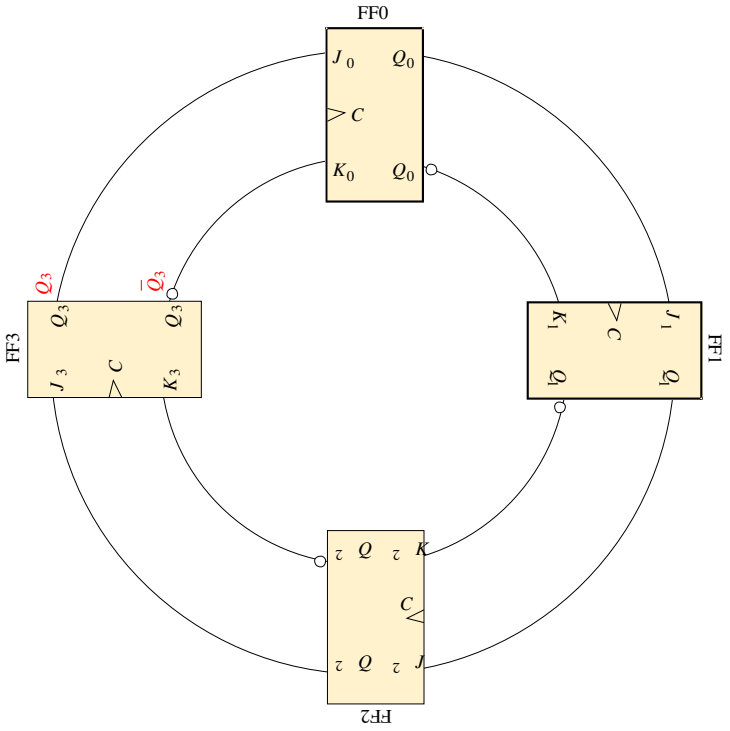


Ring Counter

Redrawing the Ring counter (without the clock shown) shows why it is a “ring”.

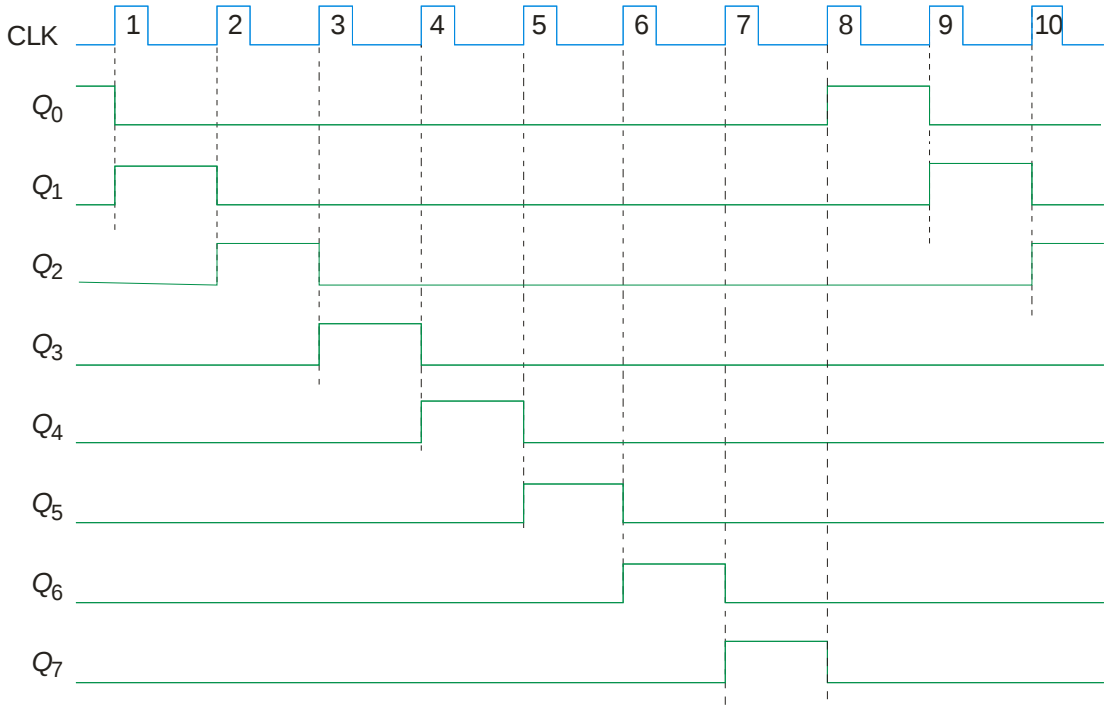
The disadvantage to this counter is that it must be preloaded with the desired pattern (usually a single 0 or 1) and it has even fewer states than a Johnson counter (*n*, where *n* = number of flip-flops.

On the other hand, it has the advantage of being self-decoding with a unique output for each state.



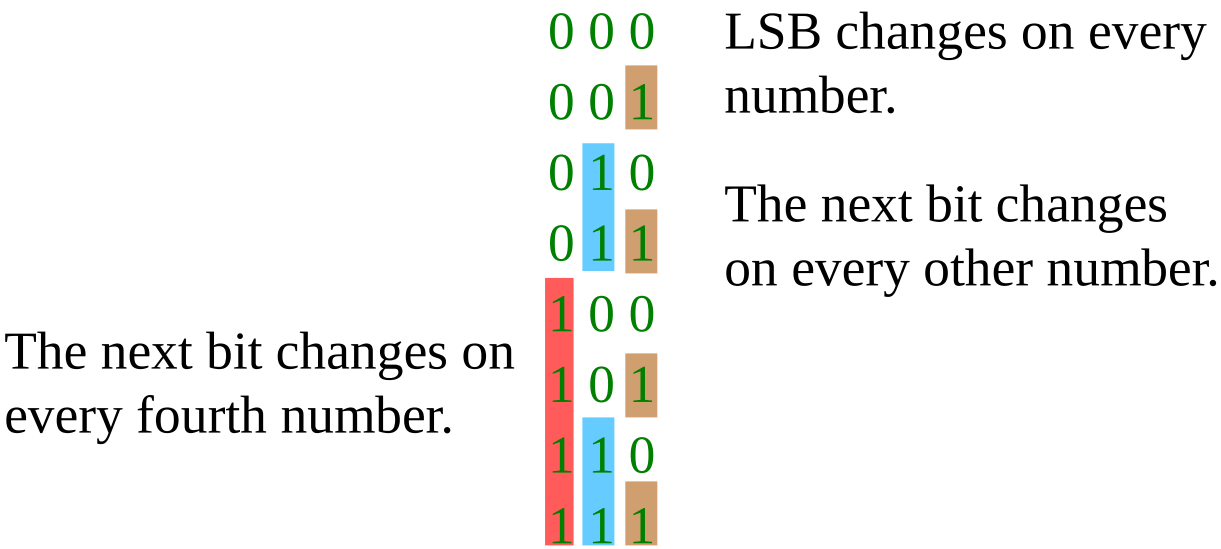
Ring Counter

A common pattern for a ring counter is to load it with a single 1 or a single 0. The waveforms shown here are for an 8-bit ring counter with a single 1.



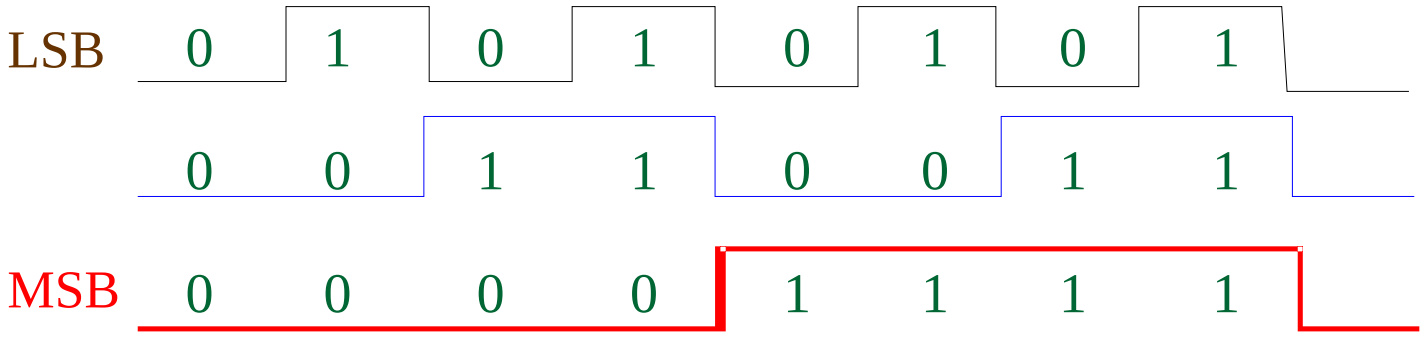
Counting in Binary

As you know, the binary count sequence follows a familiar pattern of 0's and 1's as described in Section 2-2 of the text.



Counting in Binary

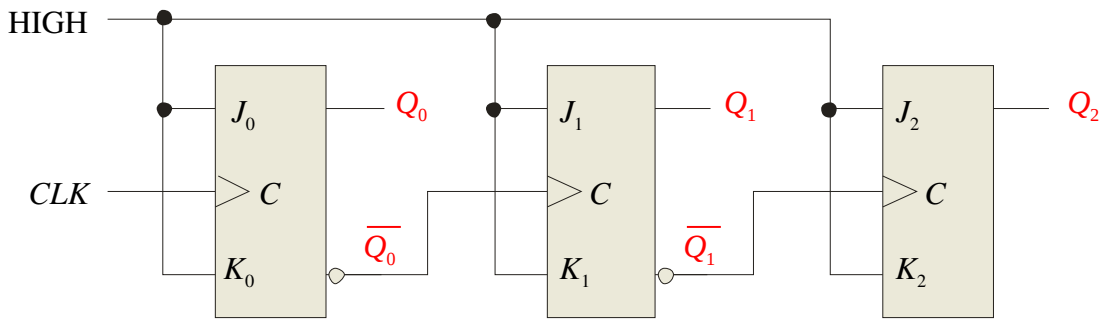
A counter can form the same pattern of 0's and 1's with logic levels. The first stage in the counter represents the least significant bit – notice that these waveforms follow the same pattern as counting in binary.



Three bit Asynchronous Counter

In an asynchronous counter, the clock is applied only to the first stage. Subsequent stages derive the clock from the previous stage.

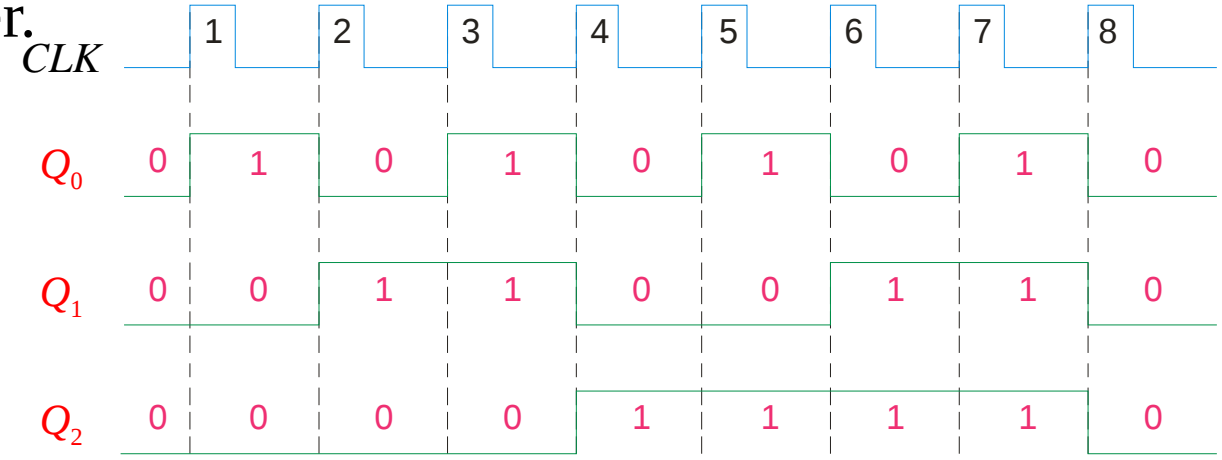
The three-bit asynchronous counter shown is typical. It uses J-K flip-flops in the toggle mode.



Waveforms are on the following slide…

Three bit Asynchronous Counter

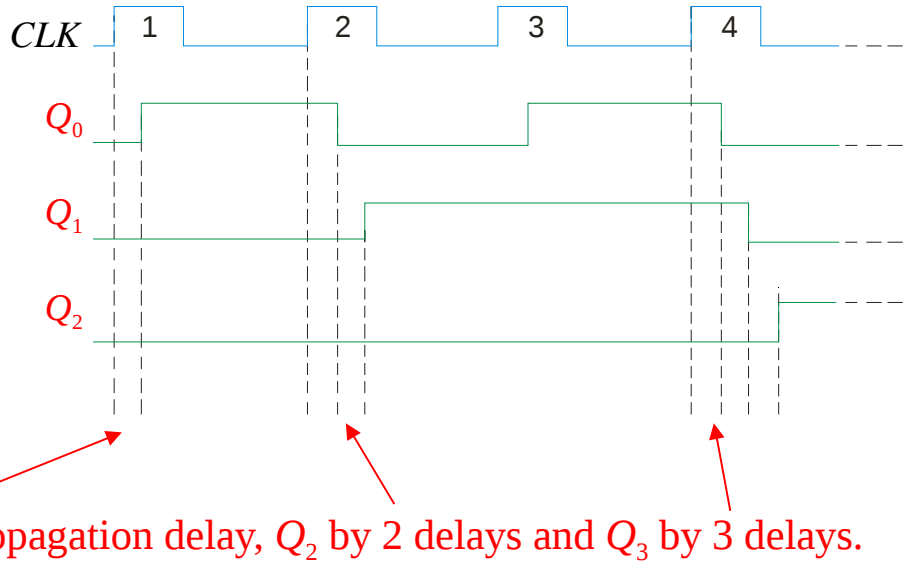
Notice that the Q_0 output is triggered on the leading edge of the clock signal. The following stage is triggered from Q_0. The leading edge of Q_0 is equivalent to the trailing edge of Q_0. The resulting sequence is that of an 3-bit binary up counter.



Propagation Delay

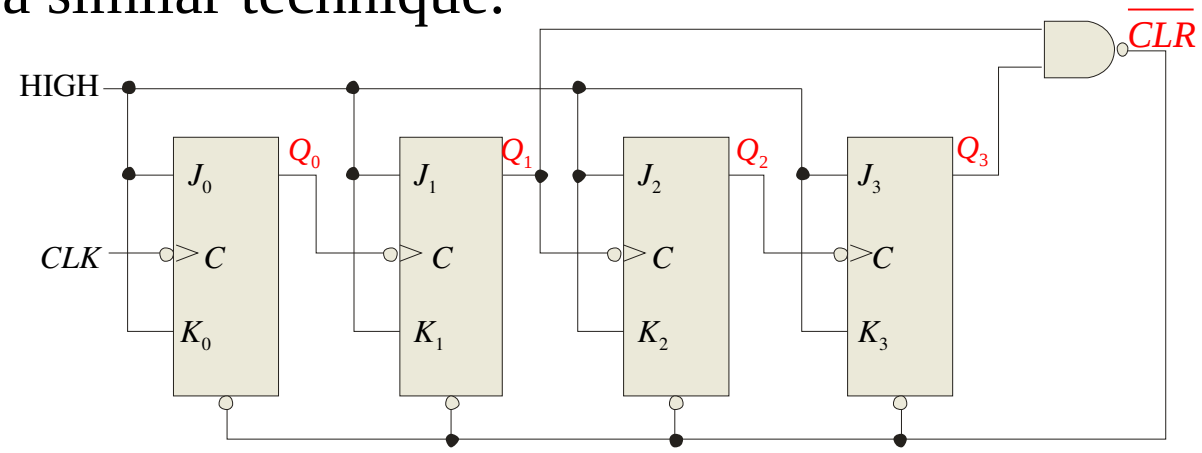
Asynchronous counters are sometimes called **ripple** counters, because the stages do not all change together. For certain applications requiring high clock rates, this is a major disadvantage.

Notice how delays are cumulative as each stage in a counter is clocked later than the previous stage.



Asynchronous Decade Counter

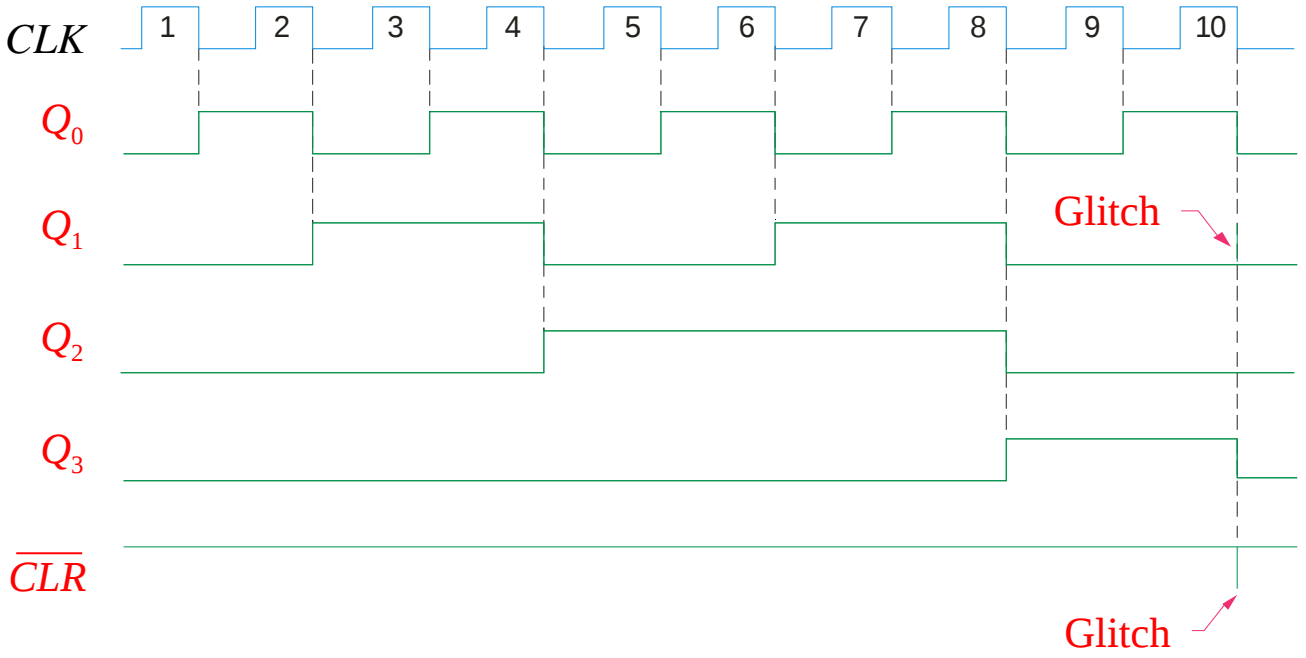
This counter uses partial decoding to recycle the count sequence to zero after the 1001 state. The flip-flops are trailing-edge triggered, so clocks are derived from the *Q* outputs. Other truncated sequences can be obtained using a similar technique.



Waveforms are on the following slide…

Asynchronous Decade Counter

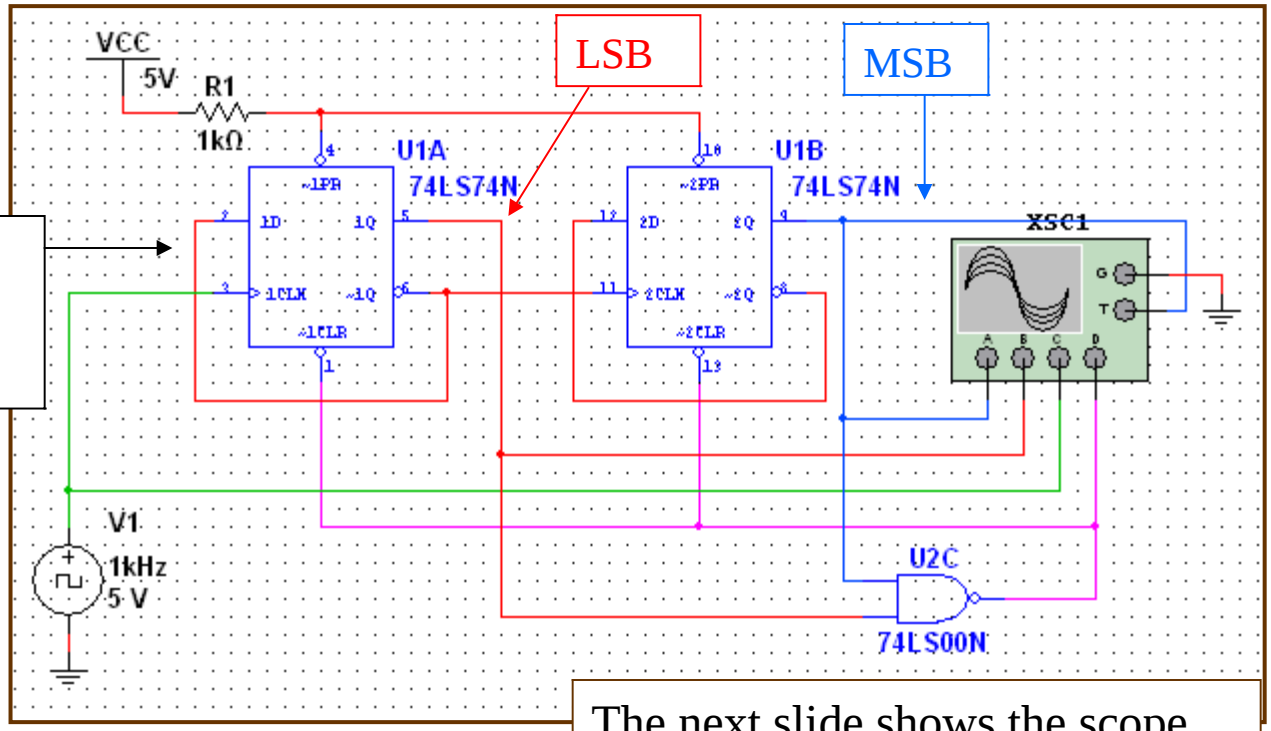
When Q_1 and Q_3 are HIGH together, the counter is cleared by a "glitch" on the $\overline{CLR}$ line.



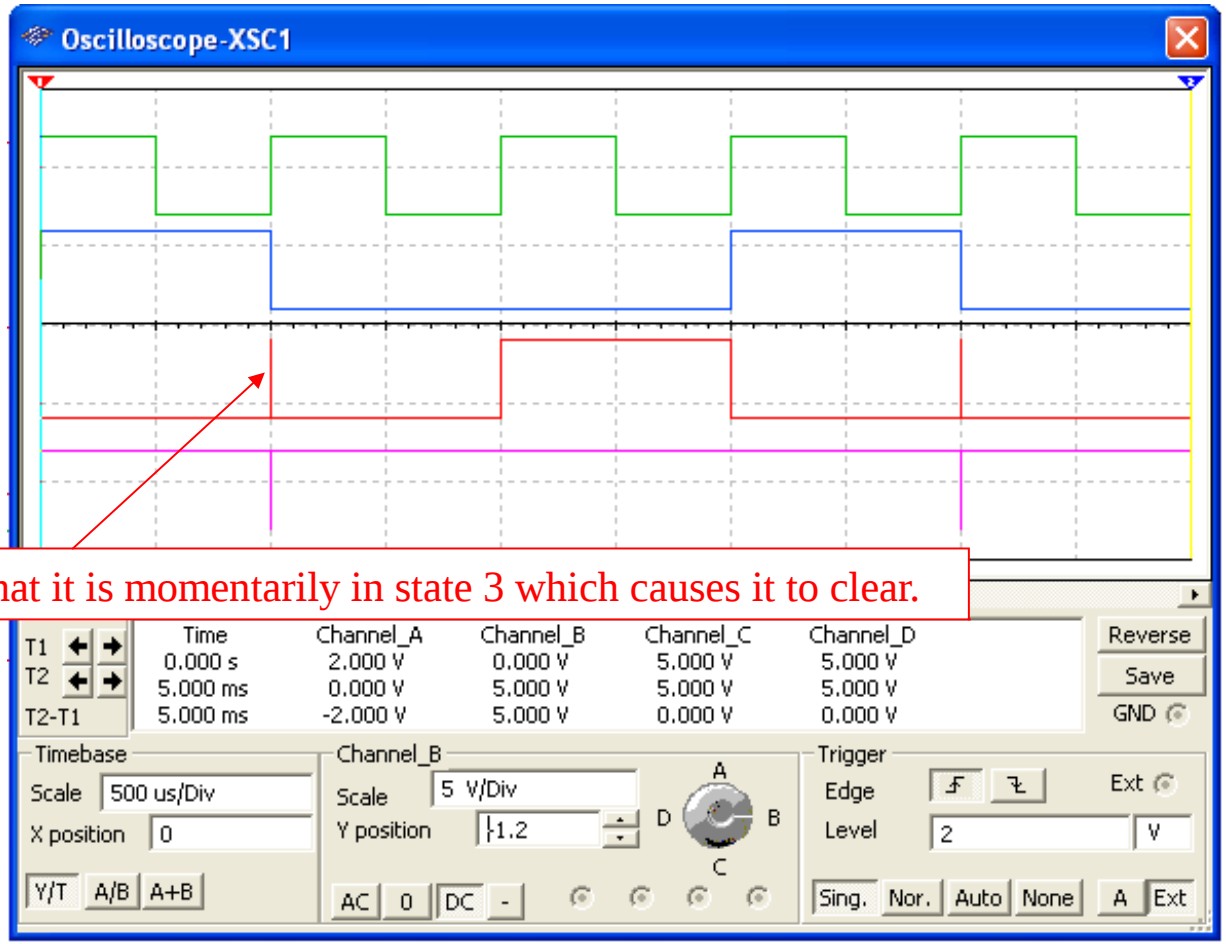
Asynchronous Counter Using D Flip-flops

D flip-flops can be set to toggle and used as asynchronous counters by connecting $\overline{Q}$ back to D. The counter in this slide is a Multisim simulation of one described in the lab manual. Can you figure out the sequence?

$\overline{Q}$ to D puts D flip-flop in toggle mode



The next slide shows the scope…

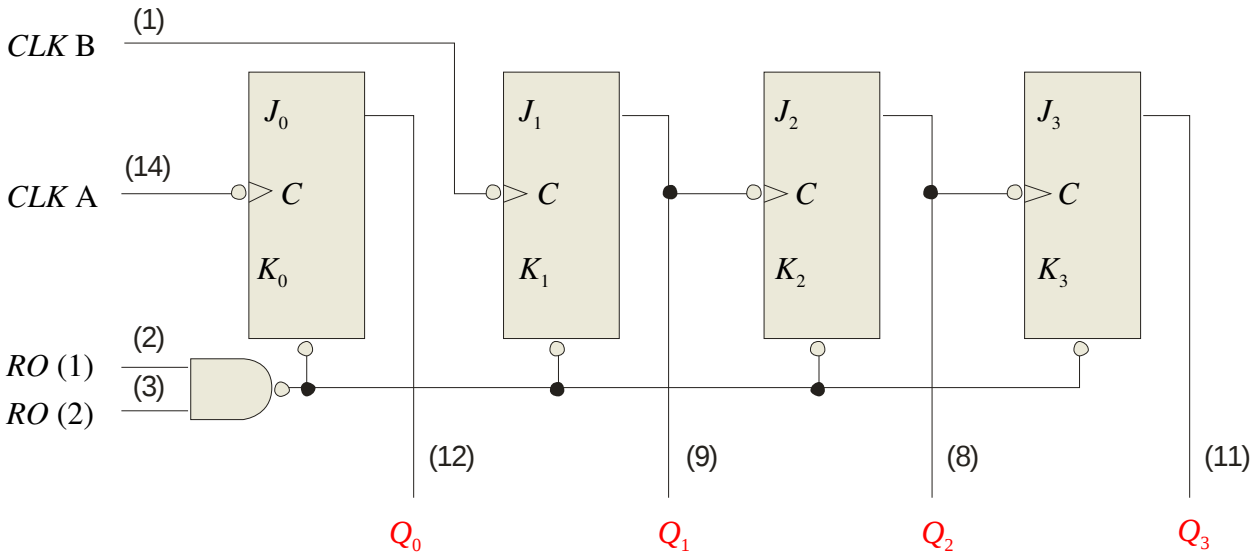


The sequence is 0 – 2 – 1 – ($\overline{CLR}$) (repeat)…

The 74LS93A Asynchronous Counter

The 74LS93A has one independent toggle J-K flip-flop driven by *CLK* A and three toggle J-K flip-flops that form an asynchronous counter driven by *CLK* B.

The counter can be extended to form a 4-bit counter by connecting Q_0 to the *CLK* B input. Two inputs are provided that clear the count.

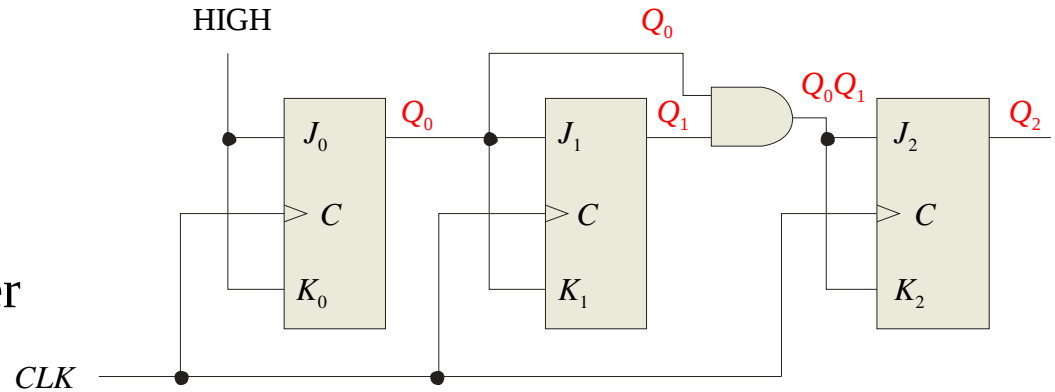


All J and K inputs are connected internally HIGH

Synchronous Counters

In a **synchronous counter** all flip-flops are clocked together with a common clock pulse. Synchronous counters overcome the disadvantage of accumulated propagation delays, but generally they require more circuitry to control states changes.

This 3-bit binary synchronous counter has the same count sequence as the 3-bit asynchronous counter shown previously.



The next slide shows how to analyze this counter by writing the logic equations for each input. Notice the inputs to each flip-flop…

Analysis of Synchronous Counters

A tabular technique for analysis is illustrated for the counter on the previous slide. Start by setting up the outputs as shown, then write the logic equation for each input. This has been done for the counter.

1. Put the counter in an arbitrary state; then determine the inputs for this state.

2. Use the new inputs to determine the next state: Q_2 and Q_1 will latch and Q_0 will toggle.

3. Set up the next group of inputs from the current output.

Outputs	Logic for inputs					
$Q_2\,Q_1\,Q_0$	$J_2 = Q_0Q_1$	$K_2 = Q_0Q_1$	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$
0 0 0	0	0	0	0	1	1
0 0 1	0	0	1	1	1	1
0 1 0						

4. Q_2 will latch again but both Q_1 and Q_0 will toggle.

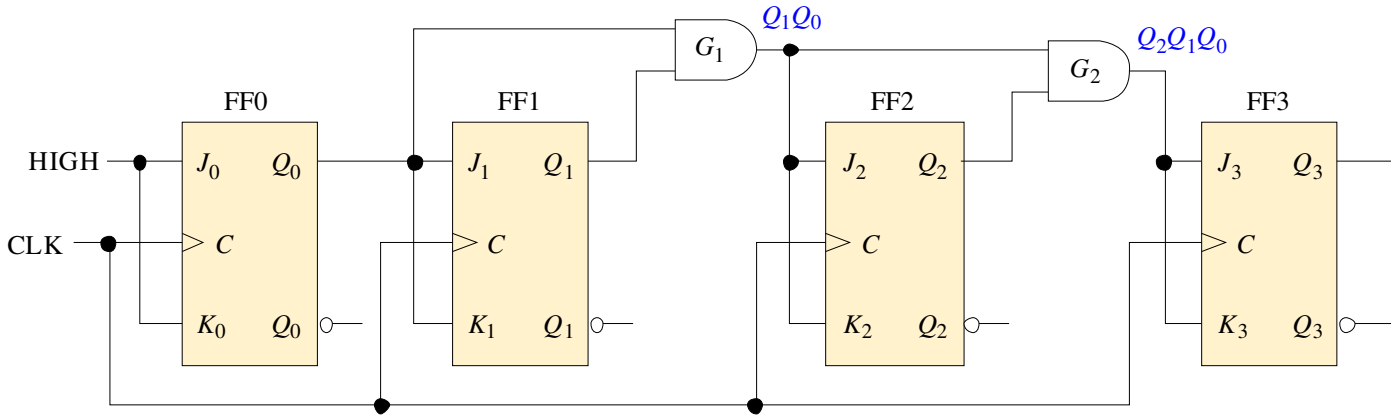
Continue like this, to complete the table. The next slide shows the completed table…

Analysis of Synchronous Counters

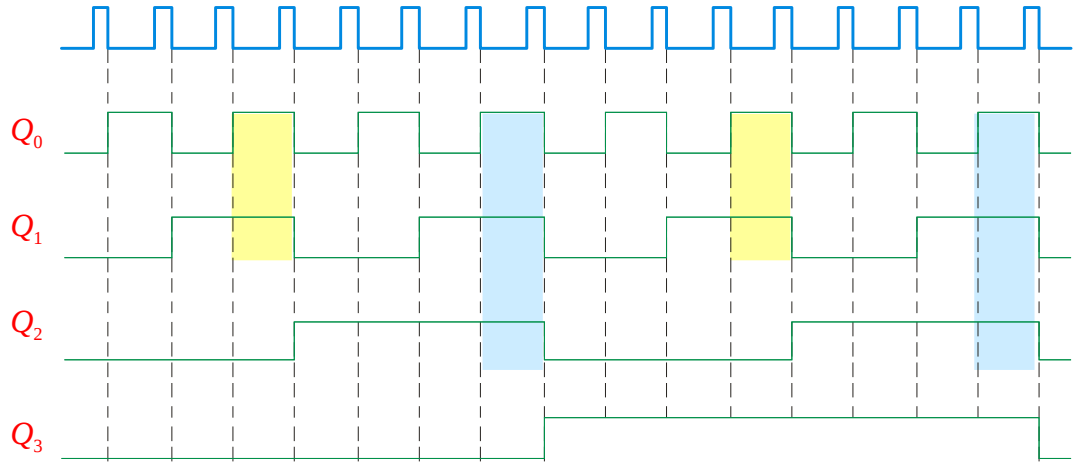
Outputs	Logic for inputs					
$Q_2\ Q_1\ Q_0$	$J_2 = Q_0Q_1$	$K_2 = Q_0Q_1$	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$
0 0 0	0	0	0	0	1	1
0 0 1	0	0	1	1	1	1
0 1 0	0	0	0	0	1	1
0 1 1	1	1	1	1	1	1
1 0 0	0	0	0	0	1	1
1 0 1	0	0	1	1	1	1
1 1 0	0	0	0	0	1	1
1 1 1	1	1	1	1	1	1
0 0 0						

At this points all states have been accounted for and the counter is ready to recycle…

A 4-bit Synchronous Binary Counter



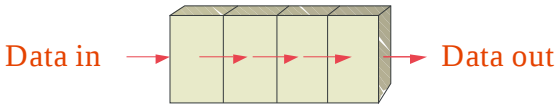
The 4-bit binary counter has one more AND gate than the 3-bit counter just described. The shaded areas show where the AND gate outputs are HIGH causing the next FF to toggle.



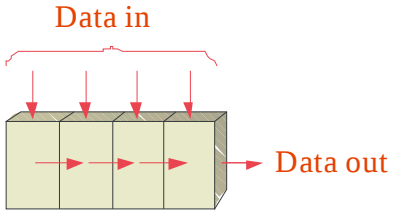
Quiz

The shift register that would be used to delay serial data by 4 clock periods is

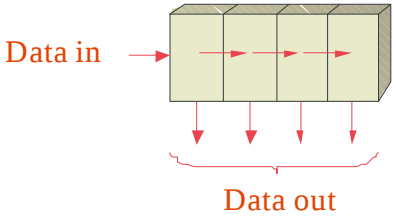
a.



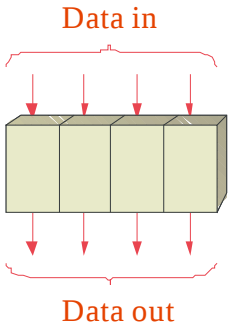
c.



b.



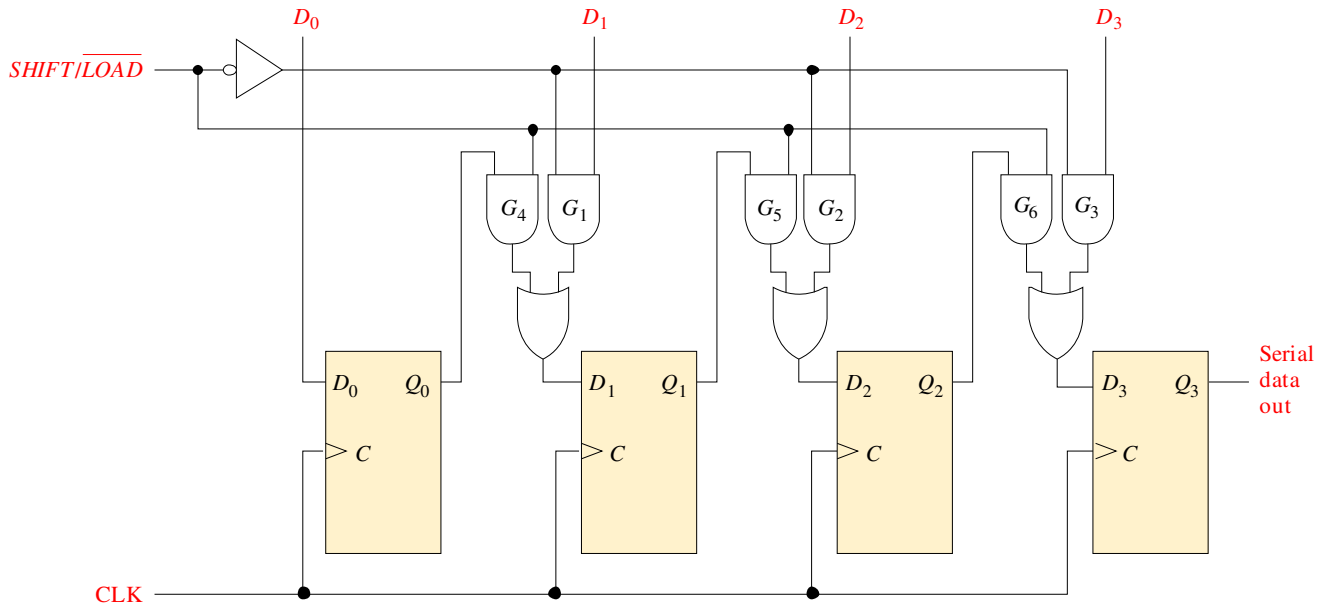
d.



Quiz

The circuit shown is a

a. serial-in/serial-out shift register
b. serial-in/parallel-out shift register
c. parallel-in/serial-out shift register
d. parallel-in/parallel-out shift register



Quiz

A 4-bit parallel-in/parallel-out shift register will store data for

a. 1 clock period

b. 2 clock periods

c. 3 clock periods

d. 4 clock periods

Quiz

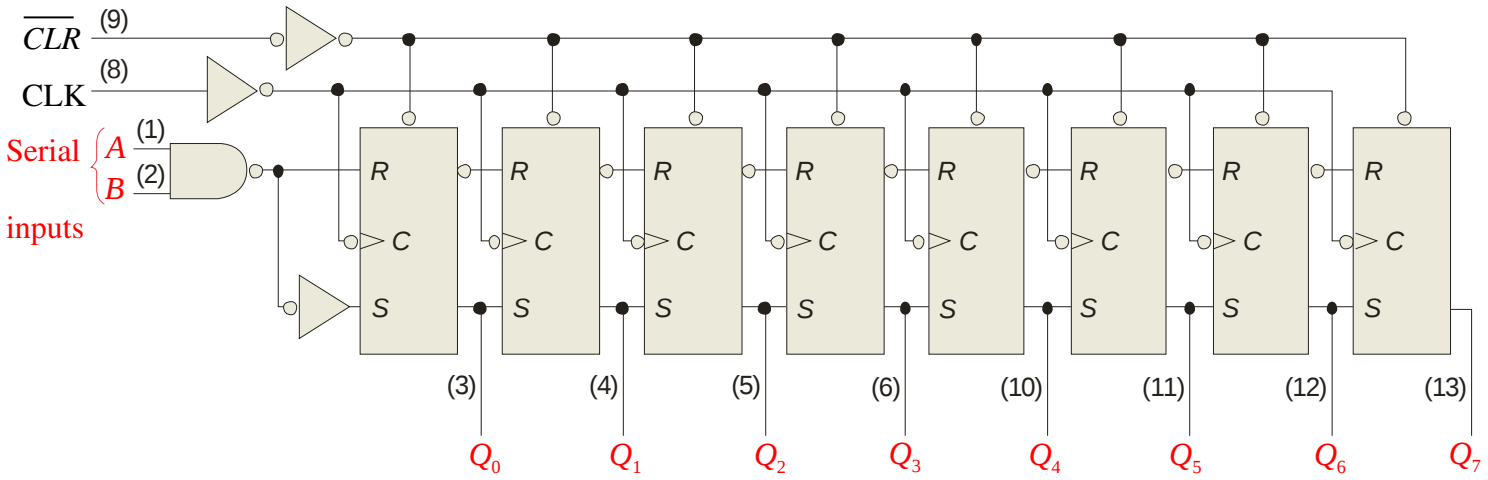
The 74HC164 (shown) has two serial inputs. If data is placed on the *A* input, the *B* input

a. could serve as an active LOW enable

b. could serve as an active HIGH enable

c. should be connected to ground

d. should be left open



Quiz

An advantage of a ring counter over a Johnson counter is that the ring counter

a. has more possible states for a given number of flip-flops

b. is cleared after each cycle

c. allows only one bit to change at a time

d. is self-decoding

Quiz

A possible sequence for a 4-bit ring counter is

a. … 1111, 1110, 1101 …

b. … 0000, 0001, 0010 …

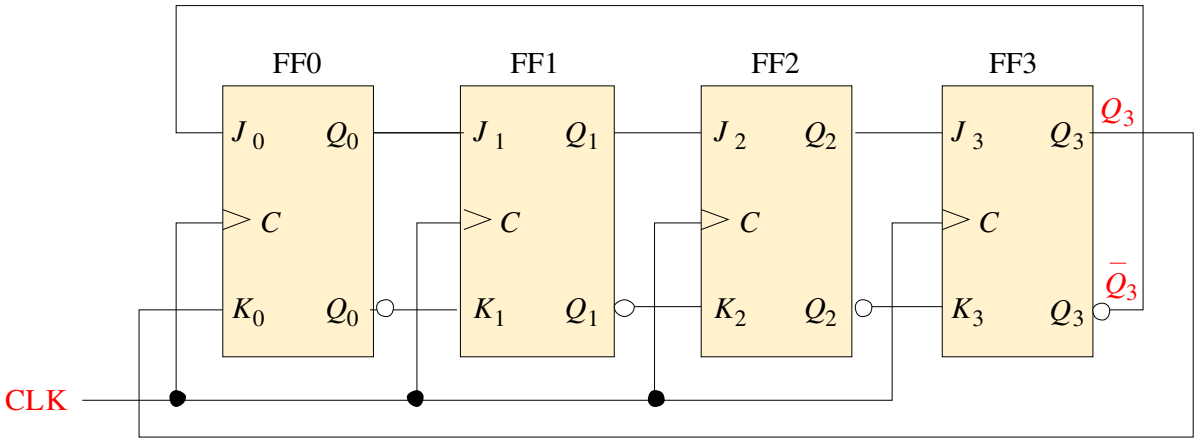
c. … 0001, 0011, 0111 …

d. … 1000, 0100, 0010 …

Quiz

The circuit shown is a

a. serial-in/parallel-out shift register
b. serial-in/serial-out shift register
c. ring counter
d. Johnson counter



Quiz

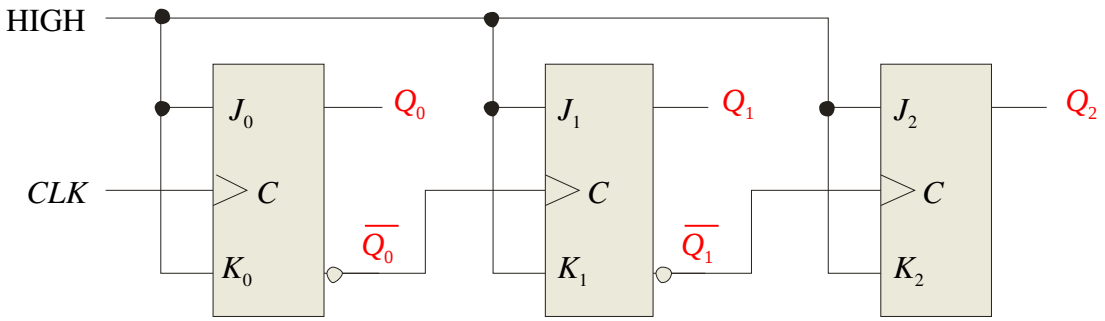
The counter shown below is an example of

a. an asynchronous counter

b. a BCD counter

c. a synchronous counter

d. none of the above



Quiz

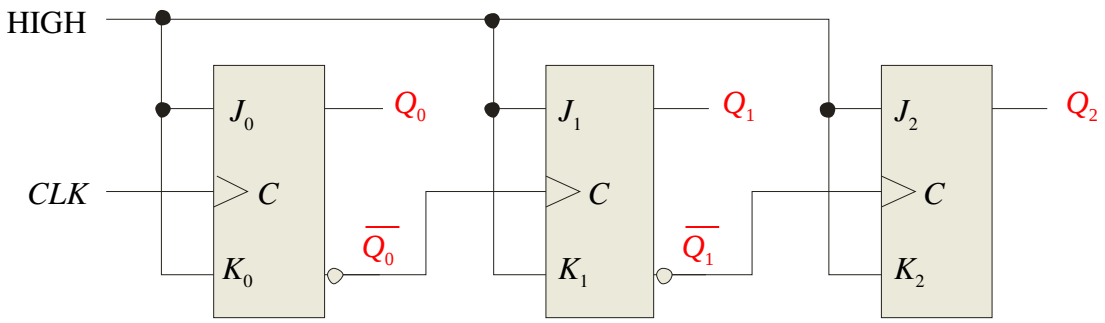
The Q_0 output of the counter shown

a. is present before Q_1 or Q_2

b. changes on every clock pulse

c. has a higher frequency than Q_1 or Q_2

d. all of the above



Quiz

To cause a D flip-flop to toggle, connect the

a. clock to the *D* input

b. *Q* output to the *D* input

c. $\overline{Q}$ output to the *D* input

d. clock to the preset input

Quiz

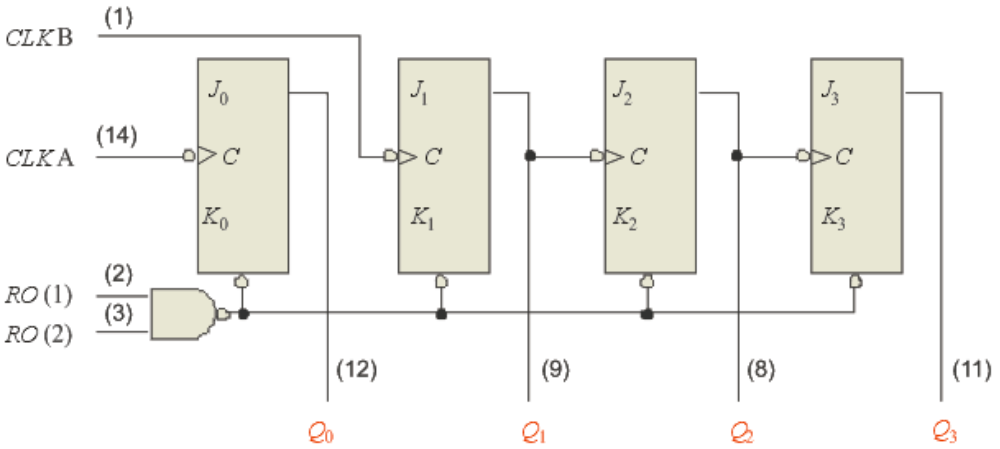
The 7493A asynchronous counter diagram is shown (*J*'s and *K*'s are HIGH.) To make the count have a modulus of 16, connect

a. Q_0 to RO(1) and RO(2) to

b. Q_3 to RO(1) and RO(2)

c. *CLK* A and *CLK* B together

d. Q_0 to *CLK* B



Quiz

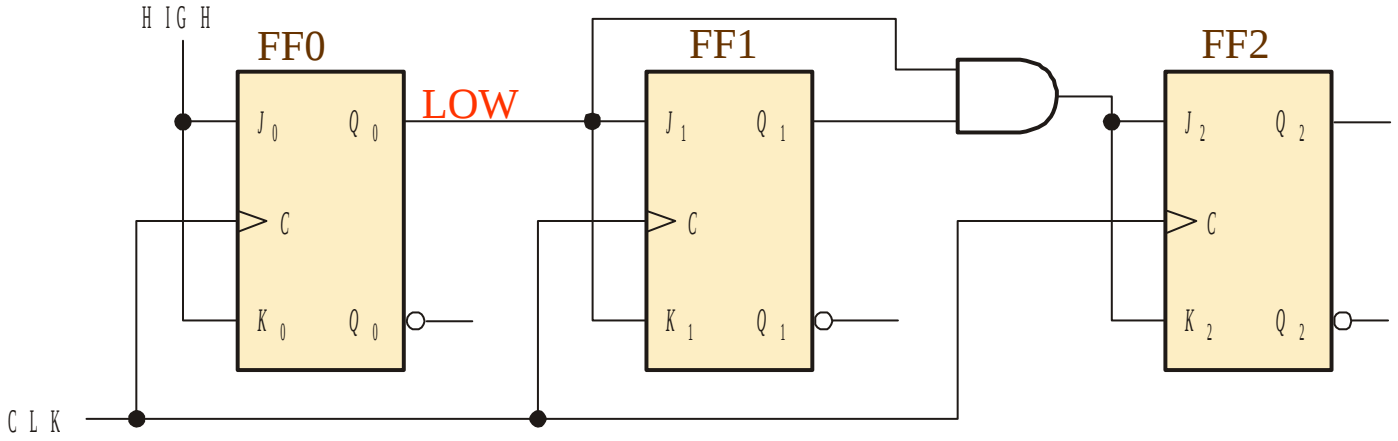
Assume Q_0 is LOW. The next clock pulse will cause

a. FF1 and FF2 to both toggle

b. FF1 and FF2 to both latch

c. FF1 to latch; FF2 to toggle

d. FF1 to toggle; FF2 to latch



Quiz

A 4-bit binary counter has a terminal count of

a. 4

b. 10

c. 15

d. 16

Quiz

Assume the clock for a 4-bit binary counter is 80 kHz. The output frequency of the fourth stage (Q_3) is

a. 5 kHz

b. 10 kHz

c. 20 kHz

d. 320 kHz

Quiz