

## Sequential Circuits: Flip-flops and Shift Registers

### Object

To investigate the properties of sequential circuits, as illustrated by latches and the J-K flip-flop.

### Parts

- (1) 7400 Quad 2-input NAND gate
- (2) 7476 Dual J-K Master-Slave flip-flops

### Study section

*Computer Systems*, Fourth Edition, Jones and Bartlett Publishers: Section 11.1, Latches and Clocked Flip-Flops.

### General information

Sequential circuits are interconnections of logic elements such that the output at a given time is dependent upon the state of the input values at some previous time. Flip-flops and latches are important circuits as used in computers, because they have the ability to retain the information present at a given time indefinitely, even though the original information is no longer present. Thus, they have a memory.

Arrays of flip-flops are often used as data registers, each flip-flop holding one bit of data. Data that is present only momentarily may be stored until a time when it is convenient to use the data. Since flip-flops have two distinct sets of output conditions, they are often known as bistable elements.

### Procedure

#### 1. The latch

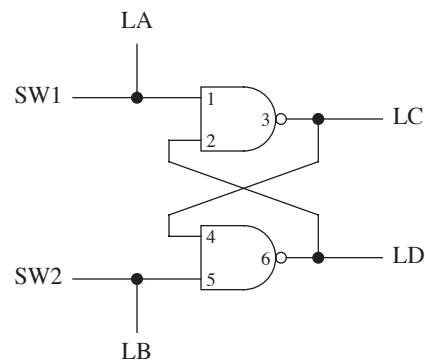
The simplest form of flip-flop in general use is the latch, which is constructed from a pair of gates. This type of latch is often used as a switch “debouncer”. Mechanical switches almost invariably “bounce” when they are activated due to mechanical construction. Thus, one switch depression may cause a series of rapid connections and disconnections lasting many milliseconds after the switch has been pressed or released.

Connect the circuit shown in Figure 1. Turn SW1 on and SW2 off as indicated in the second column of the table. This will set the initial state of the latch. Record the lamp output for the initial state. Perform the sequence of switch actions in the remaining columns of the table, recording the lamp outputs as you go. The remaining columns must be recorded in order as the outputs are not a function only of the input but of the state of the latch as well. LC and LD indicate that the circuit “remembers” indefinitely the last switch to be cycled off-on. Thus, the simple latch may be used to retain a momentary occurrence until it is desired to use the information.

A latch constructed of two cross-coupled NOR gates maintains its quiescent state with 00 input. Sending one input high then low either sets or resets the output. This latch is made of two cross-coupled NAND gates, which maintains its quiescent state with 11 input. Sending one input low then high either sets or resets the output. Note that when SW1 and SW2 are off simultaneously in the last column, both LC and LD are lit. This is an ambiguous or “not allowed” sit-

**Figure 1**

SW1	1	SW2 on	SW1 off	SW1 on	SW2 off	SW2 on	SW2 off	SW2 on	SW1 off	SW1 on	SW1 off	SW2 off
SW2	0											
LA												
LB												
LC												
LD												



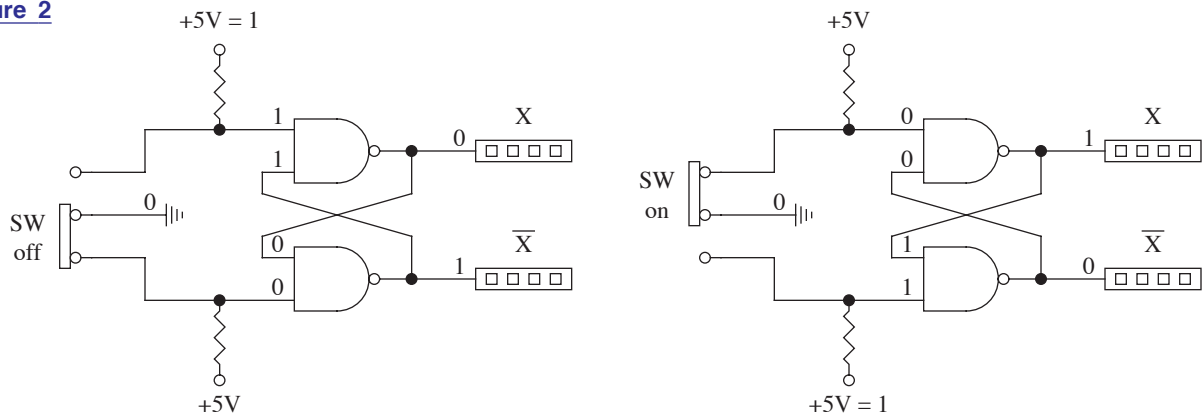
uation, because the state of LC and LD when the switches are returned to 11 will depend on which switch is released last. In all other columns, LC and LD should have complementary values.

## 2. The debounced switch

This section of the lab is only informational and requires no circuit construction.

To eliminate mechanical bounce, the switches labeled LOGIC SW are connected internally to X and Y on the control panel as shown in Figure 2 (for the X switch). In the off position the output of the top NAND gate is 0. When the switch is moved to the on position the output goes to 1 without bouncing.

Figure 2



The resistors in the circuit are called pull-up resistors. When the input to the gate is not shorted to ground, very little current flows through the resistor. Therefore, the voltage drop across the resistor is small (Ohm's law), and the input to the gate is about +5V for logic 1. The connection to +5V through the resistor pulls the input up to 1. When the input to the gate is shorted to ground through the switch, the input is 0. If the resistor were not there, power would be connected directly to ground through the switch, a most undesirable situation.

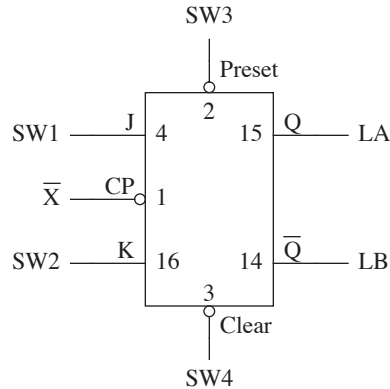
It is extremely important that all switch connections to the clock input of a flip-flop be debounced. Otherwise the flip-flop might be clocked more than once when the switch is pressed. A later part of this lab will attempt to demonstrate the bounce phenomenon by connecting the clock to a switch that is not debounced. Except for this one demonstration, from now on all clock inputs to flip-flops must be connected to the debounced X or Y logic switches.

### 3. The J-K flip-flop

The most versatile of all the different flip-flop types is the J-K. Because of the way it is internally connected, it may be used in a variety of configurations. We will explore some of these configurations that make this flip-flop so versatile.

Construct the circuit shown in Figure 3. Connect the clock input to the debounced switch  $\bar{X}$  (normally 1) with the switch off ( $X = 0$ ,  $\bar{X} = 1$ ). Connect inputs J, K, Preset, and Clear to SW1, SW2, SW3, and SW4 as shown in the figure. Set the JK input to 00. Set the Preset Clear inputs to 10 as shown in the first column, and record the initial lamp readings. Manipulate the Preset and Clear inputs as shown in the table. Record the light readings.

Figure 3

Note:

Vcc = Pin 5

Gnd = Pin 13

SW3	1	SW4 on	SW3 off	SW3 on	SW4 off	SW4 on	SW4 off	SW4 on	SW3 off	SW3 on
SW4	0									
LA										
LB										

From the light readings, you should see that when you use Preset and Clear as inputs the J-K flip-flop acts as a simple latch. This behavior is independent of the values of any of the other inputs. To verify that Preset and Clear act independently of the other inputs, set the JK input to 01, and repeat the procedure according the table of Figure 3. The lamp readings should be identical to the ones you recorded when the JK inputs were 00. You do not need to record the readings for JK values of 01.

Similarly, verify the behavior for JK values of 10 and 11. You do not need to record the readings for these settings.

The Preset and Clear inputs are often called “asynchronous” inputs because they override any other inputs being used. In a typical circuit all flip-flop “reset” inputs are connected to a “power on” reset pulse to initialize the entire system to a known condition.

Each column in the table of Figure 4 represents a switch action. When the column indicates Preset, press SW3 off then on to preset Q to 1. When the column indicates Clear, press SW4 off then on to clear Q to 0. When the column indicates CP, press the logic switch on then off, which will send  $\bar{X}$  off then on, to clock the flip-flop. Set JK = 00 and complete the first row in order, followed by the second row with JK = 01, and so on. Record the lamp readings after each action.

**Figure 4**

		Clear	CP	CP	CP	CP	Preset	CP	CP	CP	CP
JK = 00	LA										
	LB										
JK = 01	LA										
	LB										
JK = 10	LA										
	LB										
JK = 11	LA										
	LB										

It is evident from the table that a clock pulse will change the state of the flip-flop depending on the value of the JK inputs and the current state of the flip-flop. In fact, changing the J and K inputs has no effect on the outputs until the clock pulse occurs. To verify this statement, put the flip-flop in some state and change the J and K inputs several times with no clock pulse. The Q output of the flip-flop observed with LA will remain unchanged. This means that the J and K inputs may change at will without affecting the flip-flop's condition. At the time it is desired to “save” the information, a clock pulse is applied. The flip-flop now reflects the state of the J and K inputs at the time the clock pulse was applied, and remains in this condition until a new clock pulse occurs, or the Preset or Clear inputs are activated.

From the table in Figure 4, a set of rules for the output status of a J-K flip-flop as a function of the J-K inputs after the clock may be formed. Describe these rules for each combination of J and K inputs.

JK = 00

JK = 01

JK = 10

JK = 11

#### 4. The counter or divider

In this setup we will explore the use of the J-K flip-flop in one of its most common applications, that of a counter or divider circuit. Wire the circuit as shown in Figure 5 with SW1 on (high) and X off ( $\bar{X}$  high). The circuit requires two integrated circuit chips labeled IC1 and IC2.

Turn SW1 off, then on to reset all flip-flops. Record the states of LA through LD in the table under the column labeled Clear.

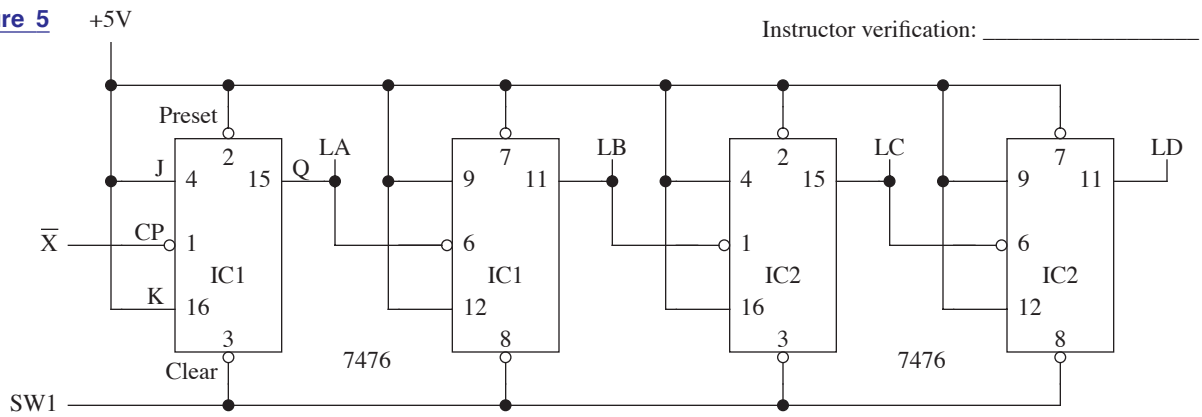
Switch X on then off ( $\bar{X}$  low then high) to clock the circuit and again record the light conditions of LA through LD under the column labeled Clock 1.

Continue clocking the circuit until all the entries of the table in Figure 5 are completed.

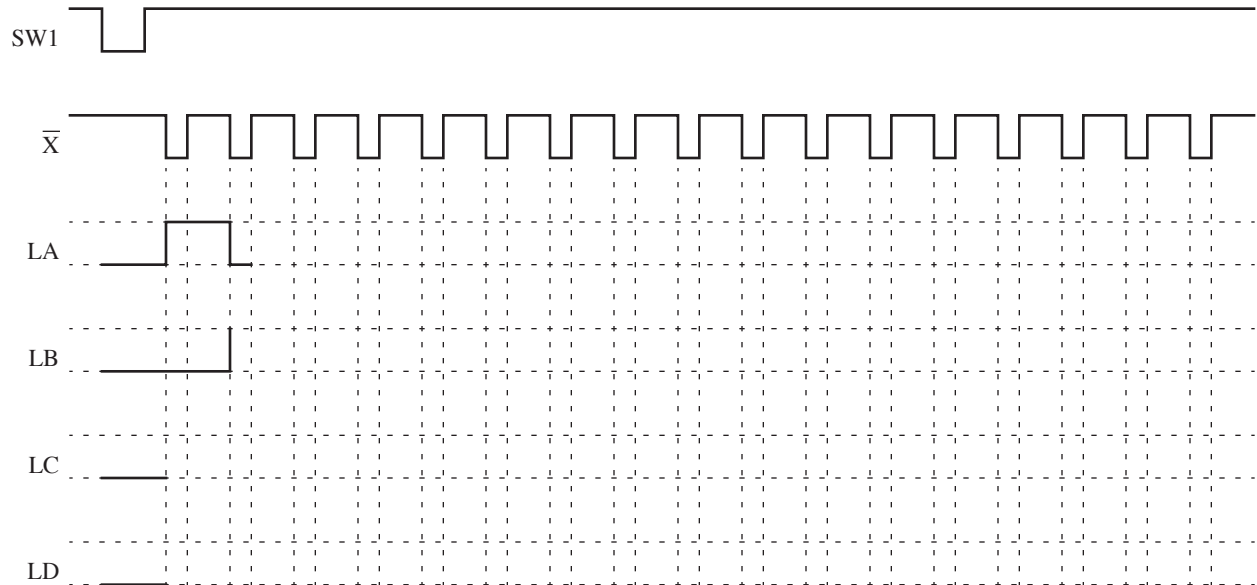
A useful device for illustrating the operation of a sequential circuit is a timing diagram. Such a diagram has been started in Figure 5. Complete the diagram using the information you have recorded in the table. Be careful to note whether the changes in the lamp readings occur on the leading or on the trailing edge of the clock pulse.

Do *not* tear down your circuit before completing the next part, as only one connection will need to change for the next setup.

Figure 5



	Clear	Clock 1	Clock 2	Clock 3	Clock 4	Clock 5	Clock 6	Clock 7	Clock 8	Clock 9	Clock 10	Clock 11	Clock 12	Clock 13	Clock 14	Clock 15	Clock 16	Clock 17
LA																		
LB																		
LC																		
LD																		



### 5. The bouncing switch

This part of the lab shows what happens when you do not use a debounced switch for a clock input. As described in Part 2 above, the logic switches are designed to eliminate the mechanical bounce that takes place when two metal parts come in contact with each other. The time between contacts is on the order of milliseconds, which is too short to be seen by the human eye but is fast enough to clock a flip-flop. The procedure is identical to the counter in Part 4, but with a bouncing switch used to clock the register instead of the debounced switch X.

Modify the circuit of Figure 5 by disconnecting the clock input from X and connecting it to SW2 instead. Turn SW2 on (high).

Send SW1 low then high to clear the shift register. All lights should be out. Record this condition in the column labeled Clear in the table of Figure 6.

Turn SW2 off and record the resulting condition in the column labeled SW2 = 0.

Turn SW2 on and record the resulting condition in the column labeled SW2 = 1.

Repeatedly clock the circuit with switch SW2, recording the light configuration each time.

**Figure 6**

	Clear	SW2 = 0	SW2 = 1	SW2 = 0	SW2 = 1	SW2 = 0	SW2 = 1	SW2 = 0	SW2 = 1	SW2 = 0	SW2 = 1	SW2 = 0
LA												
LB												
LC												
LD												

Compare the entries in this table with those in the table of Figure 5. Are there any differences? \_\_\_\_\_

It is impossible to predict how many times the switch will bounce when contact is made, so your results in this table will not be repeatable. It is even possible for a switch to occasionally not bounce at all when a mechanical contact is made. However, there should be some differences between this table and the one in Figure 5. From this table, it should be possible to tell how many times the switch bounced on one transition of SW2.

Does the bouncing effect appear only on the high-to-low transition of SW2, only on the low-to-high transition, or on both?

What was the smallest number of bounces on a single transition?

On which transition did it occur?

What was the largest number of bounces on a single transition?

On which transition did it occur?



## 6. The shift register

The shift register is at first glance very similar to the counter. However, its operation is entirely different, as is its use. A counter, as its name implies, counts the number of clock pulses applied to the input. A shift register, however, enters data present at the input of the first stage upon the application of a clock pulse, and transfers that data “downstream” with the application of succeeding clock pulses. Thus, the information appearing sequentially at the output of the last stage of a shift register is serially identical with the information that it has been delayed by a time equal to the time between clock pulses multiplied by the number of stages in the register.

You can also convert parallel data (that is, data bits appearing on several lines simultaneously) to serial data (that is, data appearing one bit at a time on a single line) by entering the data into the shift register on the Preset inputs of each stage and then using clock pulses to shift the data out of the last stage one bit at a time. Other uses will be discussed after we have explored the operation of the shift register more fully.

Construct the circuit illustrated in Figure 7. Switch SW1 will be used to “set up” the data to be entered into the shift register. The purpose of the inverter at the first stage is to allow the complement of the data to appear at the K input. Thus, if the data is a 1, the J input will be 1, the K input will be 0, and the clock pulse will cause a 1 to appear on the output of the first stage. However, if the data is 0, the J input will be 0, the K input will be 1, and the clock pulse will cause 0 to appear on the output of the first stage.

Turn off SW1 and send SW2 low then high to clear the shift register. All lights should be out. Record this condition in the column labeled Clear.

Turn on SW1. Leaving SW2 on, clock the circuit by sending X high then low ( $\bar{X}$  low then high). Enter the condition of the lights in the table in the column labeled Clock 1.

Turn off SW1 and clock the circuit again. Enter the status in the table in the row labeled Clock 2.

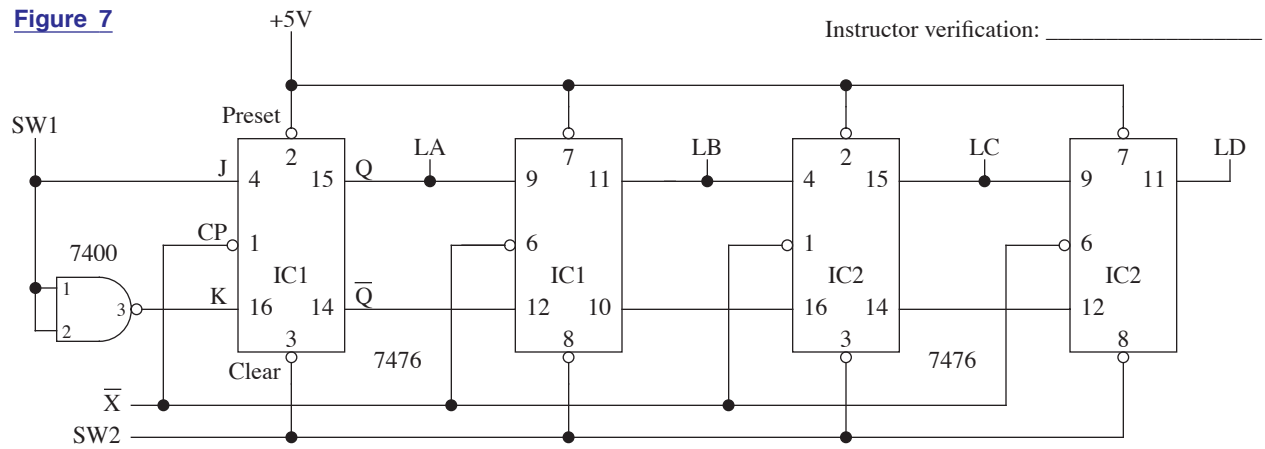
Turn on SW1, and clock the circuit again. Enter the status in the table in the row labeled Clock 3.

Turn off SW1, and leaving it off repeatedly clock the circuit, recording the light status each time until all lights stay off and the table is filled.

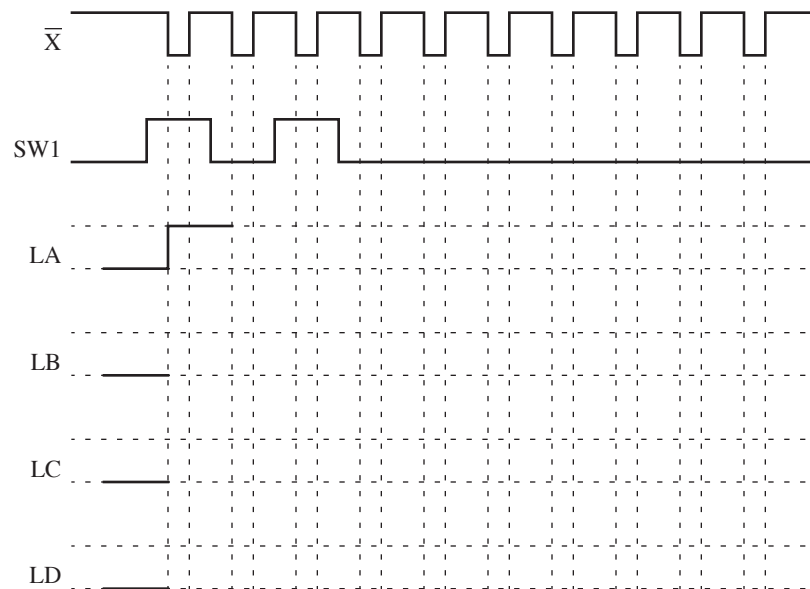
Do *not* tear down your circuit before completing the next part, as only a few connections will need to change for the next setup.

As you did for the counter, complete the timing diagram of Figure 7. The initial reset pulse to clear the register is not shown in the timing diagram. Be careful to note whether the changes in the lamp readings occur on the leading or on the trailing edge of the clock pulse.

Figure 7



	Clear	Clock 1	Clock 2	Clock 3	Clock 4	Clock 5	Clock 6	Clock 7	Clock 8	Clock 9	Clock 10
LA											
LB											
LC											
LD											



## 7. The circular shift register

A special case of the shift register is known as a circulating shift register. In this configuration, the output of the last stage is wired directly back to the input of the first stage. Thus, information contained in the register continuously circulates, advancing one stage with each clock pulse. We will enter the information in parallel through the Preset inputs.

Rewire the circuit of Figure 7, so that it is connected as shown in Figure 8. Note that only a few changes need to be made—the preset of the first stage is disconnected from +5V and replaced by SW1, and the output from the last stage is connected to the input of the first stage where SW1 was in the previous setup.

Turn the preset switch SW1 on so that it is connected to +5V along with the presets of the other stages. During operation the preset lines will all be high. To enter the initial data into the flip-flops we will clear the register then preset those which should contain 1.

Send SW2 low then high to clear the shift register. All lights should be out. Record this condition in the column labeled Clear.

Turn SW1 off and then back on to preset the register. The register should now contain the nibble 1000. (A nibble is half a byte.) Record this condition in the column labeled Preset.

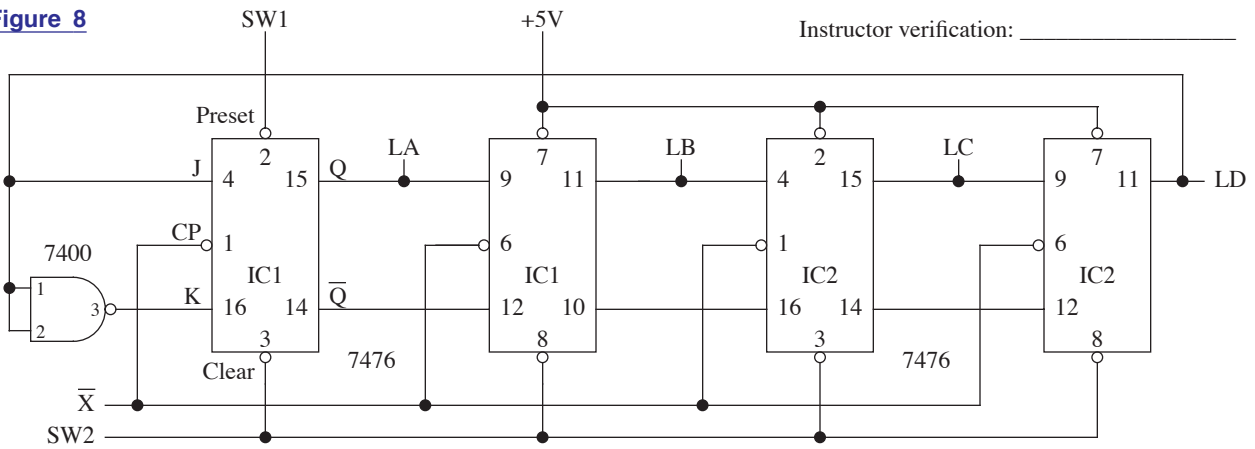
Repeatedly clock the circuit with switch X, recording the light configuration each time, until the spaces in the table have been filled.

Do *not* tear down your circuit before completing the next part, as only a few connections will need to change for the next setup.

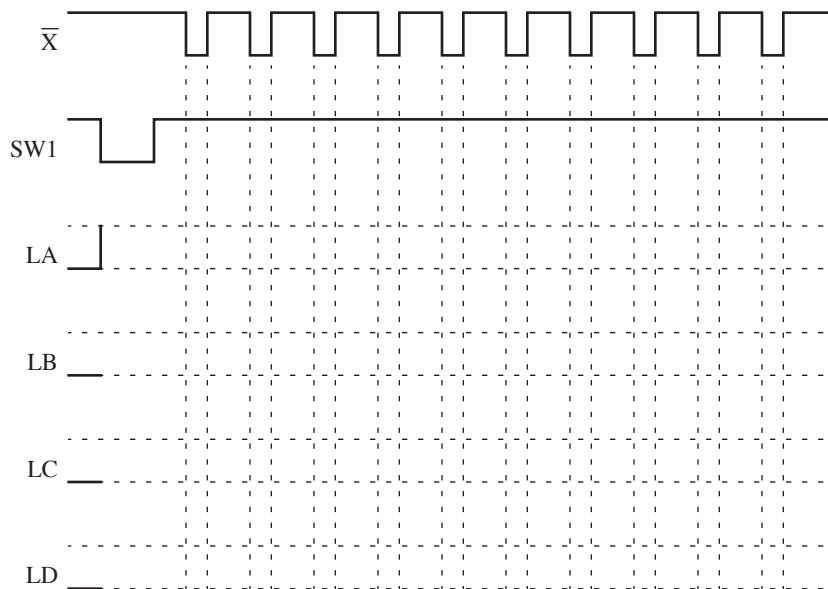
As you did previously, complete the timing diagram in Figure 8. Be careful to note whether the changes in the lamp readings occur on the leading or on the trailing edge of the clock pulse.

As you can see from the truth table and the timing diagram, the information that was entered into the register in parallel with SW1 is continuously circulating through the register. Thus, you have a four-bit memory whose locations appear at the output in sequence. If you were to build seven more shift registers, all of which were clocked with the same clock pulse, you would have a four-byte memory that presents an eight-bit byte at the outputs with every clock pulse. Multiple shift registers of much greater length (typically 1024 stages) are used extensively in such applications as refresh memories for graphics terminals, where information which is entered must be presented repeatedly at a rapid rate to provide a flicker free display.

Figure 8



	Clear	Preset	Clock 1	Clock 2	Clock 3	Clock 4	Clock 5	Clock 6	Clock 7	Clock 8	Clock 9	Clock 10
LA												
LB												
LC												
LD												



**Questions**

1. The outputs of a binary counter may be used either as a counter or as a frequency divider. Using the table and the timing diagram of Figure 5 for reference, describe each use.

Counter:

Frequency divider:

2. If a signal of 32 cycles per second is available, show how to obtain a signal of one cycle per second using J-K flip-flops. Show connections of the flip-flops including where to connect the input and where the output is obtained.

3. The basic differences between a counter and a shift register lie in the connections of the clock and the J-K inputs. Describe *in detail* the operation of each. This question requires you to state *more* than just which output is connected to which input. It requires you to explain how each signal affects each component to produce the observed behavior.