CALIFORNIA STATE UNIVERSITY
LOS ANGELES

Department of Electrical and Computer Engineering
EE-2449 Digital Logic Lab

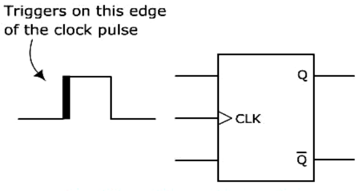
EXPERIMENT 10
INTRODUCTION TO SEQUENTIAL LOGIC

Text: Mano and Ciletti, *Digital Design, 5th Edition,* Chapter 5
Required chips: **7474:** dual D Flip-flop, **7476:** dual J-K Flip-flop, **7486**: quad 2-input XOR.

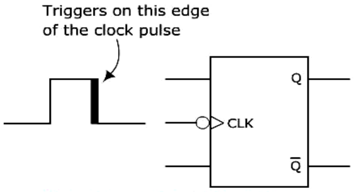
10.1 Sequential logic circuits are a type of logic circuit where the output of the circuit depends not only on the input, as in combinational logic circuits, but also on the sequence of past inputs that determine the present state of the circuit. For example, consider an elevator controller in a five story building: if someone on the third floor pushes the down button, should the elevator controller tell the elevator to go up or down to pick up the passenger on the third floor? You cannot answer that question without knowing what floor the elevator is currently on. If it is on the 4th or 5th floor, the elevator must go down to the 3rd floor. If it is on the 1st or 2nd floor it must go up to the 3rd floor. And if it is already on the 3rd floor it doesn't need to go up or down. The current floor of the elevator is one of the variables that would be used to determine the state of the elevator. There can be other state variables as well for the elevator control system. For example, there may be a state variable to indicate whether the elevator is currently moving up, currently moving down, or stationary.

Flip-flops are memory devices used to hold the state in a sequential logic circuit. A flip-flop can hold 1-bit of a state variable (i.e. one binary value). The output of a flip-flop, typically named Q, represents the state of the flip-flop. Usually the complement of the state, Q', is also available at the output. There are three types of inputs for flip-flops, the clock input, the synchronous input, and the asynchronous inputs.

- The clock input is used to control when the state should change or transition. We say that the clock triggers the state change (transition). There are different mechanisms for triggering a flip-flop. The most common is to trigger a flip-flop on either the rising (**positive**) edge of the clock ↑ or on its falling (**negative**) edge ↓. The following figures show the two types of triggers. Note that the triangle symbol (>) is used to show that it is an edge-triggered flip-flop and the bubble in front of the triangle indicates that it is a negative edge-triggered flip-flop.

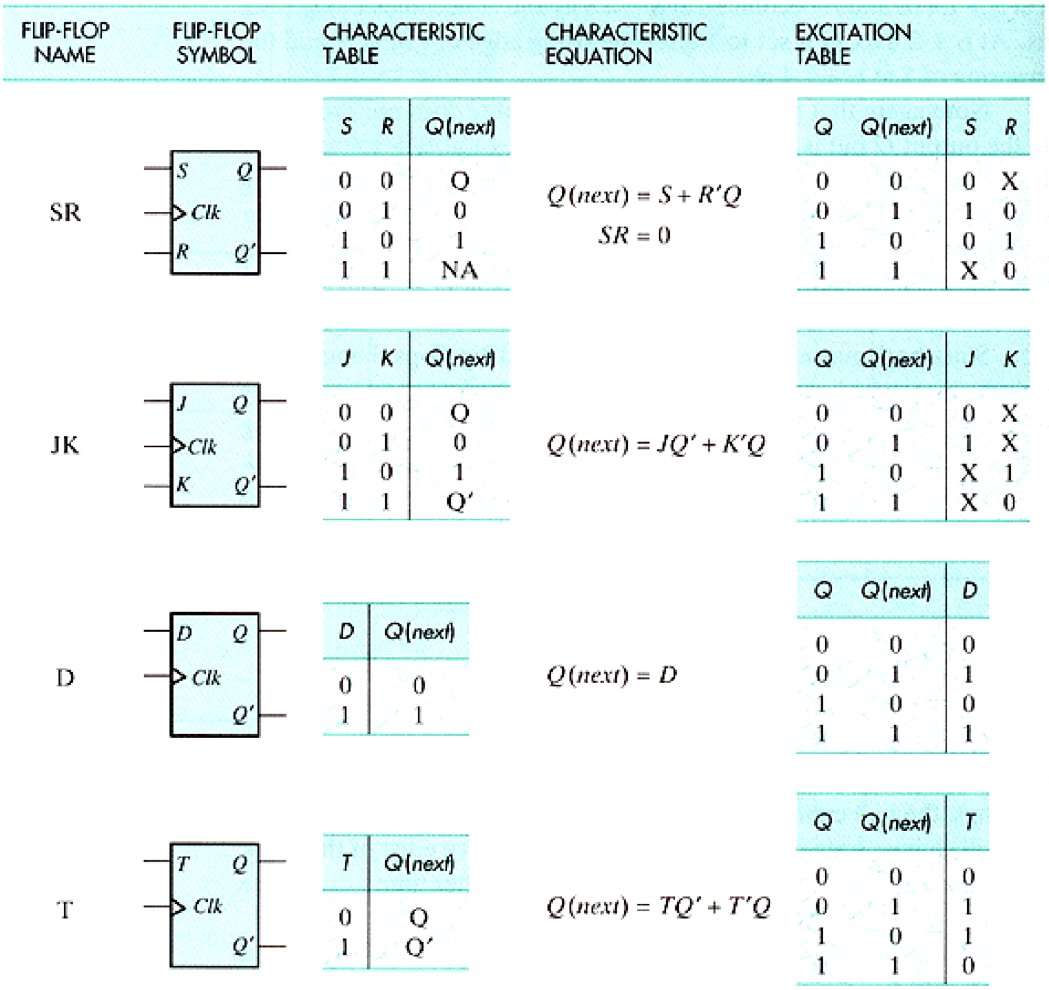


Positive Edge Triggering



Negative Edge Triggering

- The synchronous inputs are used to determine how the flip-flop should change state when it is triggered. ("synchronous" implies that the inputs are synchronized with the clock.) There are four types of flip-flops as shown in the figure below. The figure shows the name of the flip-flop (SR, JK, D, and T), the logic symbol for the flip-flop, the characteristic table of the flip-flop that describes how it behaves ("Q(next)" is the next state of the flip-flop after the trigger), the characteristic equation that captures the behavior in a Boolean expression, and the Excitation table that shows how to control the synchronous inputs to get the desired state transition (from the present state Q to the next state Q(next)). You will see various ways to describe the next state including Q(next), Q(t+1), Q_{t+1}, and Q^+. We'll use Q(t+1).



FLIP-FLOP NAME	FLIP-FLOP SYMBOL	CHARACTERISTIC TABLE	CHARACTERISTIC EQUATION	EXCITATION TABLE
SR	S Q Clk R Q'	S R | Q(next) 0 0 | Q 0 1 | 0 1 0 | 1 1 1 | NA	$Q(next) = S + R'Q$ $SR = 0$	Q Q(next) | S R 0 0 | 0 X 0 1 | 1 0 1 0 | 0 1 1 1 | X 0
JK	J Q Clk K Q'	J K | Q(next) 0 0 | Q 0 1 | 0 1 0 | 1 1 1 | Q'	$Q(next) = JQ' + K'Q$	Q Q(next) | J K 0 0 | 0 X 0 1 | 1 X 1 0 | X 1 1 1 | X 0
D	D Q Clk Q'	D | Q(next) 0 | 0 1 | 1	$Q(next) = D$	Q Q(next) | D 0 0 | 0 0 1 | 1 1 0 | 0 1 1 | 1
T	T Q Clk Q'	T | Q(next) 0 | Q 1 | Q'	$Q(next) = TQ' + T'Q$	Q Q(next) | T 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0

Let's look at the four types of flip-flops in more detail (note: changes in Q occur only at the triggering edge of the clock).

- SR Flip-flop – this is a set-reset flip-flop. When S = 1, R = 0, the flip-flop is set, Q(t+1) = 1. When R = 1, S = 0, the flip-flop is reset, Q(t+1) = 0. When both S and R are 0, there is no change, Q(t+1) = Q(t). The SR flip-flop has an undefined behavior when both S and R are 1. For this reason, SR flip-flops are not often used.

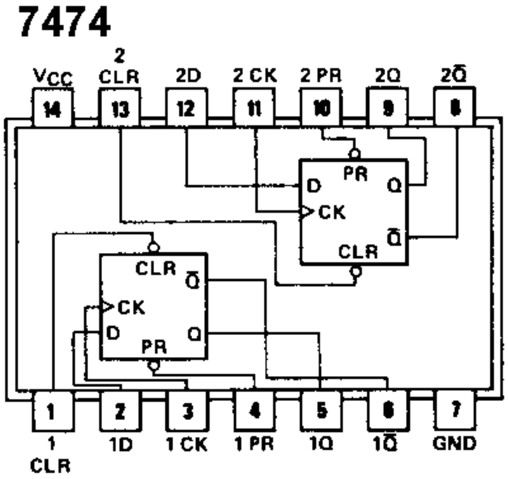
- JK Flip-flop – a JK flip-flop is similar to an SR flip-flop, where setting J = 1, K = 0, will set the flip-flop, K = 1, J = 0, will reset the flip-flop, and having both J and K equal to 0 will result in no change. The JK flip-flop differs from the SR since when both inputs are 1, the state will toggle (be complemented), that is, Q(t+1) = Q(t)'.
- D Flip-flop – this flip-flop is known as a **data** flip-flop. The state of the D flip-flop will follow the D input. When the flipflop is clocked, if D = 0, Q(t+1) will be 0 and if D = 1, Q(t+1) will be 1. Thus, the D flip-flop behaves as a memory element, remembering the value on the D input in the state Q; i.e. Q(t+1) = D.
- T Flip-flop – this is a toggle flip-flop. When T = 0, the state will be unchanged, Q(t+1) = Q(t), and when T = 1, the state will toggle (be complemented), that is Q(t+1) = Q(t)'.

- The third type of inputs are asynchronous inputs. Whereas synchronous inputs only effect the state of the flip-flop on the positive or negative edge of the clock (depending on the type of flip-flop), asynchronous inputs override the synchronous inputs (take priority over them) and also do not depend on the clock.

  When an asynchronous input is active, it will immediately effect the state of the flip-flop (it does not wait for the clock). Flip-flops usually have two asynchronous inputs, preset (PR) and clear (CLR). PR is used to set the state (set Q to 1); CLR is used to clear the state (clear Q to 0). The asynchronous inputs are often active-low which means that they will be active when a low voltage is applied to them and inactive when a high-voltage is applied. A bubble is often used to indicate that the asynchronous input is active-low; alternatively there may be a bar over its name or a slash / in front (like /CLR) to indicate that it is active-low.

  The asynchronous inputs are used to initialize flip-flops to a known state on power up or to reset a state to a known value during normal operation. When the asynchronous inputs are inactive, the flip-flop will be controlled by the clock and synchronous inputs. Note, you should not have both preset and clear inputs active at the same time as the effect on the state will be unpredictable.

10.2* Exploring the behavior of a D Flip-flop



At the right is the diagram for a 7474 dual D Flip-flop chip.

Answer the following questions in your lab notebook.

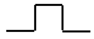
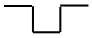
- Question: Are the flip-flops positive-edge triggered or negative-edge triggered? Explain how you know.
- Question: Are the preset (PR) and clear (CLR) asynchronous inputs active-high or active-low? Explain how you know.
- Question: Since there are two flip-flops, how many states can be represented using one 7474? Consider the fact that there are two state variables (Q1 and Q2). To help you figure out the answer, complete the following table below and include it in your lab note book. Indicate the total number of possible states that can be represented using Q1 and Q2.

Q1	Q2	States
0	0	0 (00)
0	1	1 (01)
1	0	
1	1	

Referring to the 7474 pin assignments above, draw a wiring diagram to test the behavior of a D flip-flop.

- Draw the logic diagram of one of the D flip-flops (not the entire chip) and label all of the pin numbers for the inputs and outputs accordingly.
- Include this power and ground table.

Unit	Chip	Power	Ground
1	7474	14	7

- In the diagram, connect the PR and CLR inputs so that they are initially inactive (you need to determine if connecting to power (high) or to ground (low) will make them inactive).
- Connect the clock input (CK) to the pushbutton switch (pulser). Although the 7474 triggers on the clock's positive edge, you can use either pulser output (normally high or low) since either way the clock pulse has a positive edge:  or  .

You may prefer to use Express SCH. You won't have to label pin numbers since all chip numbers and pin numbers including for Vcc and Ground are already included in the diagram. Therefore, a power and ground table is not needed. For the clock, instead of a pushbutton switch symbol, just use a "port" symbol from the Library Symbol listing:

In the wiring diagram, connect the D input to one of the toggle switches and connect the Q output to one of the LEDs and Q' to another LED. With ExpressSCH just use port symbols switch LED

After you complete the wiring diagram, wire the circuit as shown in your diagram using the 7474 chip, a pulser (CK), a toggle switch (D), and two LEDs (Q and Q'). Test your circuit following the characteristic table shown below.

1) In the first 3 lines, to make PR or CLR high or low, connect them to power or ground. Observe the outputs on the LEDs and insert them into the Next State columns. The X's indicate that changes in CK and D should have no effect on Q(t+1). Show that this is true for each of the first 3 lines of the table by toggling D up and down and then by pulsing the clock.

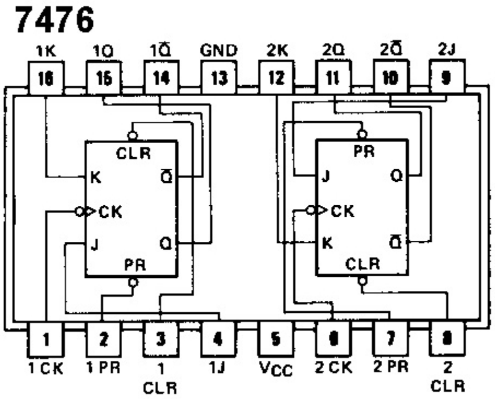
2) In the last 2 lines, PR and CLR are connected to power and so are inactive. For each line, toggle the clock *after* you have set D to the value shown. Then enter the results into the Next State columns.

Inputs				**Next State**	
PR	**CLR**	**CK**	**D**	**Q(t+1)**	**Q'(t+1)**
Low (0)	High (1)	X	X		
High (1)	Low (0)	X	X		
Low (0)	Low (0)	X	X		
High (1)	High (1)	↑	Low (0)		
High (1)	High (1)	↑	High (1)		

- Question: When PR is low, does changing the clock or D input have any effect on Q and Q'?
- Question: Repeat when CLR is low.
- Question: How does Q'(t+1) correspond to Q(t+1)? Are there any cases when it doesn't behave as expected?
- Question: What happens when both PR and CLR are high and you toggle D and pulse the clock?

Demonstrate your circuit to your instructor. Do not remove it--you'll need it later.

10.3* Exploring the behavior of a dual JK Flip-flop chip



Answer the following questions in your lab notebook.

- Are the flip-flops positive-edge triggered or negative-edge triggered? Explain how you know.
- Are the preset (PR) and clear (CLR) asynchronous inputs active-high or active-low? Explain how you know.

Referring to the 7476 pin assignments above, draw a wiring diagram to test the behavior of a JK flip-flop.

- Draw the logic diagram of one of the JK flip-flops (not the entire chip) and label all of the pin numbers for the inputs and outputs accordingly. (You may wish to use ExpressSCH, in which case a power and ground table like that in section 10.2 is not needed.)
- In the diagram, connect the PR and CLR inputs so that they are inactive (again, you need to determine if connecting to power (hi) or to ground (low) will make them inactive).
- Connect CK to the pulser. You want the flip-flop to change state on the clock's negative edge, but you can use either pulser output (normally high or low) since either way the clock pulse has a negative edge: or .
- Connect the J and K inputs to two of the toggle switches and connect the Q output to an LED.

After you complete the wiring diagram, connect the circuit as shown in your diagram using the 7476 chip, a pulser (CK), two toggle switches (J,K), and an LED (Q). We assume that Q'(t) and Q'(t+1) will always be complements of their respective Qs, so they are not included in the table at the right. Also, there is no need to include a column for CK since its use is implied in every state transition.

Notice in the table that in addition to Next State there is an additional column called Present State. The Present State is the state of the flip-flop (Q) *before* the clock trigger (↓). It is the state at time t, or Q(t). The Next State is the state of the flip-flop *after* the trigger; i.e. at time t+1, or Q(t+1).

External Inputs		**Present State**	**Next State**
J	**K**	**Q(t)**	**Q(t+1)**
0	0	0	
0	1	0	
1	0	0	
1	1	0	
0	0	1	
0	1	1	
1	0	1	
1	1	1	

Since PR and CLR affect the circuit just as they did in section 10.2, you won't have to test for their effects here, which is why they don't appear in this table. However, you will use them to set Present State values as shown in the table before you clock the flip-flop. Initially, connect then both to power (inactive).

Test your circuit line by line. In each case

1) Switch J and K high or low as indicated.

2) Then set Present State values as shown in the table by momentarily grounding either PR or CLR (leaving the other inactive).

3) Then press and release the pulser and observe Q(t+1)'s value on its LED. Enter that value into the Next State column of the table.

Demonstrate your circuit to your instructor.

10.4 Building T flip-flops from JK Flip-flops.

You can build a T flip-flop from a JK flip-flop by connecting both J and K inputs together. Its diagram would be similar to that in section 10.3, consisting of: a 7476 chip, a pulser clock (CK), and an LED (Q). Draw the wiring diagram, but instead of separate toggle switches for J and K, include only one which will be your T input. Connect it to both J and K,

Note: you will *not* be wiring up this circuit, just filling in its table.

If you were building and testing the circuit, you would go through the table line by line, activating either PR or CLR momentarily to set Q(t) to 1 or 0 as shown in this state table. You would then toggle the T switch high or low as the table indicates. And, finally, you would pulse the clock and then, from the Q output LED, you would fill in the Next State column.

External Input	**Present State**	**Next State**
T	**Q(t)**	**Q(t+1)**
0	0	
0	1	
1	0	
1	1	

Here you will just fill in the Next State column based on your understanding of T-flip-flop behavior from section 10.1. Then derive the equation for Q(t+1) in terms of T and Q(t).
Does it agree with the one in section 10.1?

Show your results to your instructor.

10.5* Building a T flip-flop using a D Flip-flop.

In a D flip-flop, the present value of its input, D(t) becomes its future output Q(t+1) after the clock trigger; i.e. Q(t+1) = D(t). So the values you placed in the column for Q(t+1) in the table of section 10.4 (above) would be the same in a column for D(t). Enter those values in this table.

External Input T	**Present State Q(t)**	**Flipflop Input D(t)**
0	0	
0	1	
1	0	
1	1	

But now the values in all three columns of the table exist *at the present time*--there is no future time (t+1) in this table. This transforms it into a truth table, one which describes a *combinational* rather than a sequential circuit (i.e. no flipflops) and from which you can derive an equation for D in terms of T and Q.

Derive this equation and name the cicuit element it describes. One of two inputs to this element is T (the other is ?) and its output goes to the D input of the flip-flop. Draw the wiring diagram which incorporates the D flip-flop of section 10.2 along with the element you just described, as well as a pulser for CK, a toggle switch for input T, and an LED for Q.

Now build the circuit and go through the state table line-by-line, entering the values for Q(t+1) as you go.

Note: the clock is omitted from the table since it is understood that Q only changes when the clock triggers. PR and CLR are also omitted. You needn't test their effects here--that was already done in section 10.2--but you will need to use them to change the present state of Q as you go through the table.

Input T	Present State Q(t)	Next State Q(t+1)
0	0	
0	1	
1	0	
1	1	

Before you clock the flip-flop *each time*, first do the following:

- Switch T to the desired value shown in the table.
- Use PR or CLR to set Q(t) to 1 or 0 as indicated in the table. To set Q(t) to 1, momentarily ground PR then reconnect it to power (leaving CLR connected to power--inactive). To clear Q(t) to 0, momentarily ground CLR then reconnect it to power (leaving PR connected to power--inactive).

Demonstrate for your instructor how you generated the values for Q(t+1) in the table.

- o Question: Do your experimental results in this table agree with those you entered in the table in section 10.4 ?

10.6 Designing a D flip-flop using a T Flip-flop.

Now, what if you wanted a D flip-flop, but only a T flip-flop and a gate are available. In this case, D would be the external input and T would be the flip-flop input. (Flip-flop output is again Q.)

External Input D	Present State Q(t)	Flip-flop Input T(t)
0	0	
0	1	
1	0	
1	1	

In the table at the right, fill in values for flip-flop input T that will give the desired transitions from Q(t) to Q(t+1) required by input D. As explained at the top of section 10.5a, the Q(t+1) values are the same as the D values--which is why no Q(t+1) column is included. (Example: in line 2, the present value of Q(t) is 1. What should be the value of Q(t+1) and what value of T will produce it?)

Finally, derive an equation for T in terms of D and Q.

- o Question: What gate does this represent? Does it look similar to one you've seen before?