

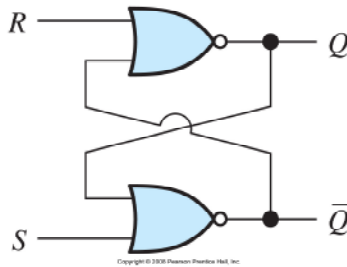
## EE334 Sequential Logic

The logic circuits we've seen so far (those built from AND, OR, NOT, etc.) immediately generate an output based on their inputs. Change the input and the output changes to match.

**Sequential logic circuits** are sensitive to input values, too, but their output also depends on their previous state. In essence, sequential logic circuits remember their old input values (past states). Not surprisingly, sequential logic circuits are the underlying technology of chip-based computer memory.

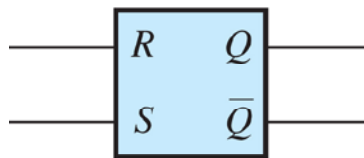
One of the building blocks of sequential circuits is the **flip-flop**. The flip-flop name comes from the fact that a flip-flop circuit can flip from one state to another and back again (from 0 to 1 and back to 0). There are several kinds of flip-flops, differing slightly in operation, but the same in that they each store one bit at a time. Flip-flops are built by connecting the logic gates we've already seen (AND, OR, NOT, etc.) to produce feedback (output fed back to the input). You can see the feedback loops in the SR flip-flop in problem #1.

- Shown below is the internal schematic for the **SR (set-reset) flip-flop**. This flip-flop has two outputs:  $Q$  and  $\sim Q$ . The  $\sim Q$  output is simply  $Q$  inverted. See if you can fill in the truth table for the SR flip-flop. **Hint:** start with the states  $R=0, S=1$  and  $R=1, S=0$ .



SR flip-flop			
S	R	Q	$\sim Q$
0	0		
0	1		
1	0		
1	1		

Usually, we don't show the internal circuitry of a flip-flop because the inputs and outputs of a flip-flop are the only things needed for diagrams. Here's the circuit symbol for an SR flip-flop.



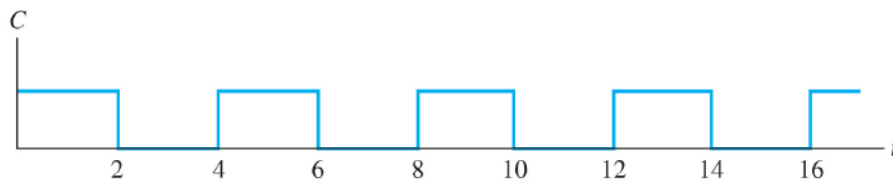
Circuit symbol for an SR flip-flop.

Clocked Flip-flops

Sometimes it's handy to have a flip-flop that changes state (is sensitive to its inputs) only at certain times. Such flip-flops are called **clocked flip-flops**. The term "clock" comes from the fact that for many digital circuits to work properly, the timing of events must be carefully controlled. The internals of a complex circuit goes something like this:

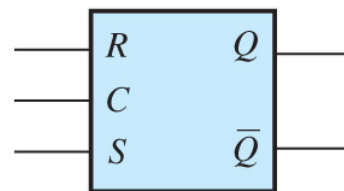
- a. On clock signal save new input values and last state.
- b. Inputs and last state used to compute next state. Non-saved value in flux during this time.
- c. Circuit reaches a steady state.
- d. Repeat.

The usual clock signal in a circuit oscillates between 0 and 1 with a regular pattern like a metronome. That said, a clock signal could be any signal that has the ability to change from 0 to 1 and back again. A typical clock signal is shown below.



2. Using the truth table to for a regular SR flip-flop (see problem #1) as a guide, complete the truth table below for a clocked SR flip-flop. This flip-flop can only change its output value when the clock signal, C, is high. The five possible states for the flip-flop are listed for your convenience. "X" means a value can be a 0 or a 1. The circuit symbol for a clocked SR flip-flop is also shown.

Clocked SR flip-flop			
S	R	C	Q
0	0	X	
0	1	1	
1	0	1	
1	1	1	
X	X	0	



Clocked SR flip-flop.

Some clocked devices are active (sensitive to changes) when the clock is low (0). Some are active when the clock is high (1). Some are sensitive to changes when the clock is transitioning from low to high. These devices are said to be **rising-edge sensitive**. Still others activate when the clock changes from high to low and are said to be **falling-edge sensitive**. If a device is rising-edge sensitive, the clock input on the device will feature a small triangle. On truth tables, the active part of the clock will be shown as an upward arrow:  $\uparrow$ . A falling-edge sensitive device features the triangle plus the familiar negate bubble. On truth tables, the active portion of the clock cycle is shown using a downward arrow:  $\downarrow$ . It doesn't matter that a flip-flop is falling-edge or rising-edge triggered—they both do the same job. *What does matter, at least to circuit designers, is when the flip-flop is sensitive to its inputs.*

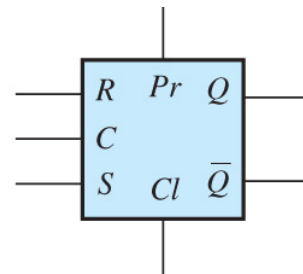
Asynchronous Inputs

Some clocked flip-flops feature inputs that ignore the clock signal. Thus they are called **asynchronous inputs**. *When activated, the asynchronous inputs change the state of the flip-flop immediately, independently of the clock. Moreover, while activated the asynchronous inputs override all other functions of the device.* Once the asynchronous inputs are deactivated, the device resumes normal operation. Note that the state of the device depends on what effect the asynchronous inputs had on the device.

There are two common asynchronous inputs on flip-flops: the Preset and the Clear. When activated the Preset input (abbreviated “Pr”) causes the non-inverted output to go to 1. Think of Preset as “preset to 1.” The Clear (abbreviated “Cl”) causes the non-inverted output to go to 0. Think of Clear as “clear the device by setting it to 0.”

- Based on the explanation of asynchronous inputs in the previous two paragraphs, complete the following truth table for an SR flip-flop with asynchronous preset and clear inputs. A circuit symbol for the SR flip-flop is shown, too.

Clocked SR Flip-flop With Asynchronous inputs					
Pr	Cl	S	R	C	Q
0	0	0	0	X	
0	0	0	1	↓	
0	0	1	0	↓	
X	X	1	1	↓	N/A
0	1	X	X	X	
1	0	X	X	X	
1	1	X	X	X	N/A



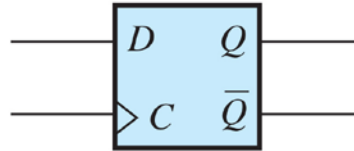
Clocked SR flip-flop with asynchronous Preset and Clear inputs.

- According to the truth table is the SR flip-flop in problem #3 falling-edge sensitive, rising-edge sensitive, active high, or active low?

EE 334  
Sequential Logic

There are three other common flip-flops in addition to the SR. Their truth tables (sometimes called excitation tables) and circuit symbols are shown below. For simplicity's sake, they are shown without Preset and Clear inputs. However, they can be added to any flip-flop.

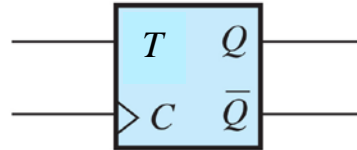
D Flip-flop		
C	D	Q
0	X	No change
1	X	No change
↑	0	0
↑	1	1



Rising-clock D flip-flop.

5. In your own words, write a description of the behavior of the D flip-flop.

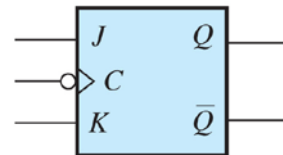
T Flip-flop		
C	T	Q
0	X	No change
1	X	No change
↑	0	No change
↑	1	Toggle



Rising-clock T flip-flop.

6. Write a description of the behavior of the T flip-flop.

JK Flip-flop				
C	J	K	Q	Comment
0	X	X	No change	Memory
1	X	X	No change	Memory
↓	0	0	No change	Memory
↓	0	1	0	Reset
↓	1	0	1	Set
↓	1	1	Toggle	



Falling-clock JK flip-flop.

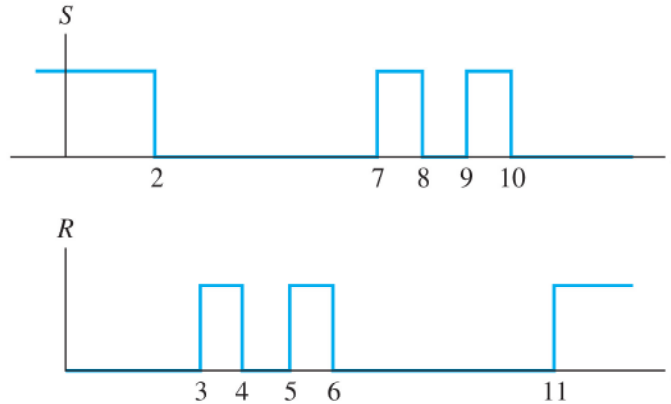
7. Highlight the rows of the JK flip-flop truth table where it behaves like an SR flip-flop.

8. Highlight the rows of the JK flip-flop truth table where it behaves like a T flip-flop.

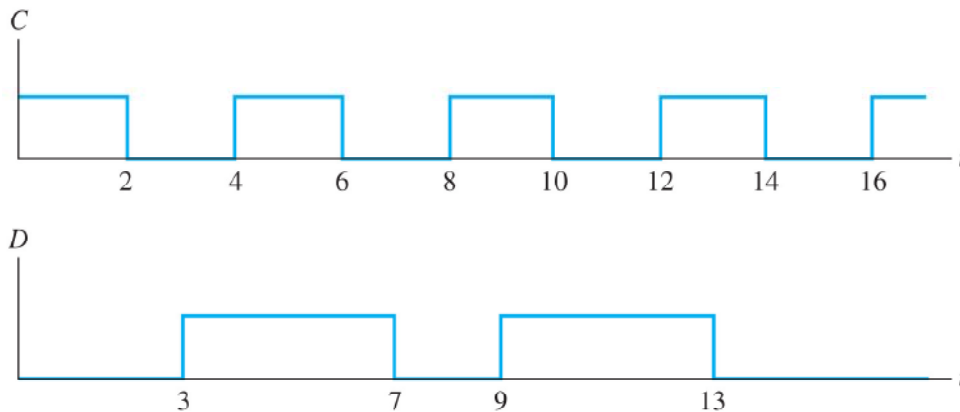
Timing Diagrams

A useful tool for understanding flip-flops or circuits containing flip-flops (or more generally any stateful circuit) is the timing diagram. A **timing diagram** is a pictorial representation of a circuit's inputs, outputs, and important signals.

9. Complete the following timing diagram for an SR flip-flop by adding the waveform for the Q output to the diagram below.



10. The input signals to a clocked rising-edge-triggered D flip-flop are shown below. Sketch the output Q to scale versus time. Assume that Q starts low.

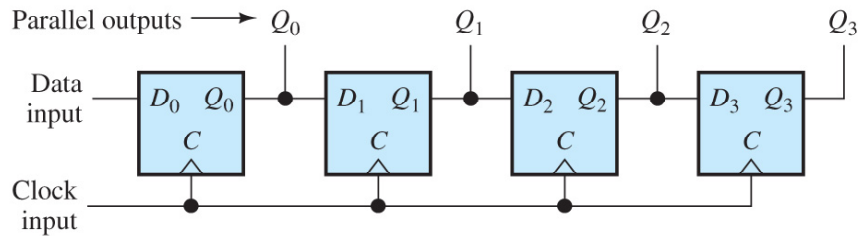


Shift Registers

One useful device we can build from flip-flops is the **shift register**. You can think of the shift register as an electronic bucket brigade. On the clock signal, flip-flops hand off their current bucket (a bit value of 0 or 1) to the next flip-flop in line whilst accepting the previous flip-flop's bucket.

11. Brainstorm on some uses for shift registers.

Shift registers are very simple devices. They are just a chain of flip-flops connected from output to input port as the figure below shows.



12. Complete the timing diagram below for each of the Q outputs (Q<sub>0</sub>, Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub>) for the flip-flop shift register shown above. Make sure your drawings are to scale with respect to time.

