

Flip Flops

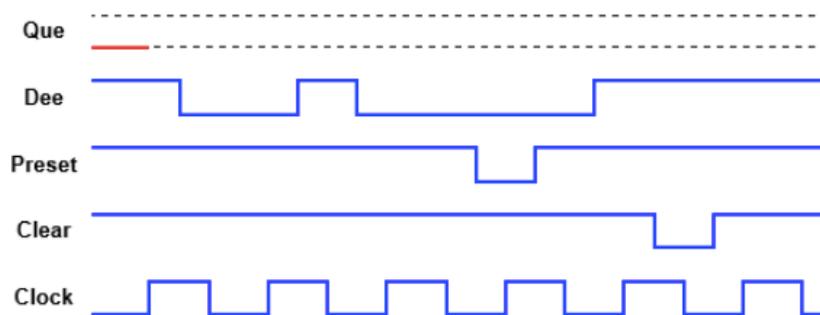
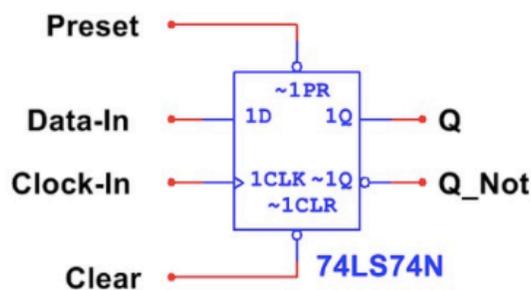
Introduction

Flip-flops are the fundamental building blocks of **sequential logic**. There are a variety of different flip-flop types and configurations. In this activity (and this course), we will only study two types of flip-flop: the D flip-flop that was introduced in Unit 1 and the J/K flip-flop. After reviewing the basic operation of the 74LS74 D and the 74LS76 J/K flip-flops, this activity will examine two applications of flip-flops.

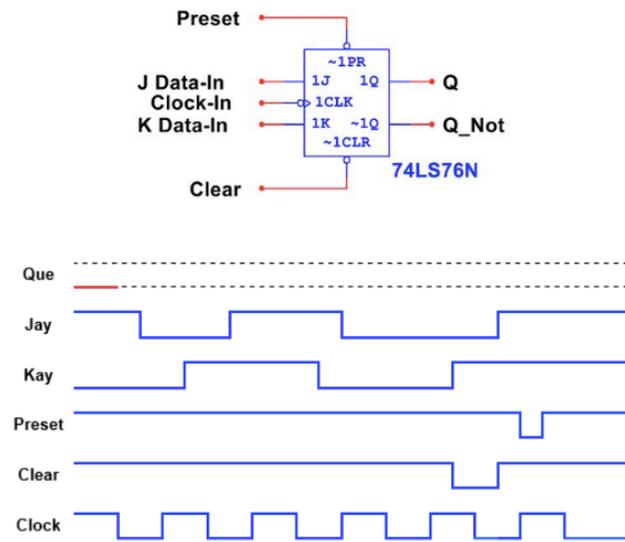


Before we jump into a discussion of practical applications of J/K or D flip-flops, let's revisit how flip-flops work.

1. For the 74LS74 D flip-flop below, complete the timing diagram for the output signal **Que**. Note that the **CLK** input for this flip-flop is a positive edge **trigger** and both the **PR** and **CLR** asynchronous inputs are active low.

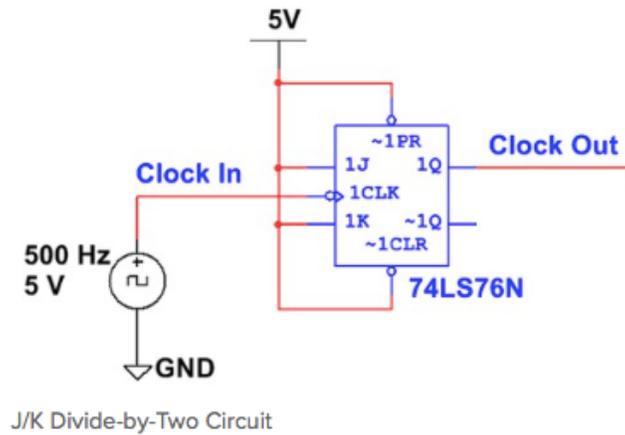


2. For the 74LS76 J/K flip-flop below, complete the timing diagram for the output signal **Que**. Note that the **CLK** input for this flip-flop is a negative edge trigger and both the **PR** and **CLR** asynchronous inputs are active low.

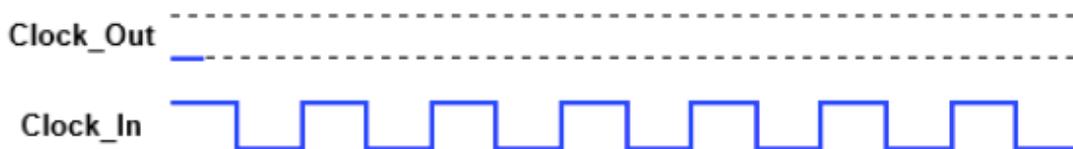


Let's examine some simple applications of the D and J/K flip-flops.

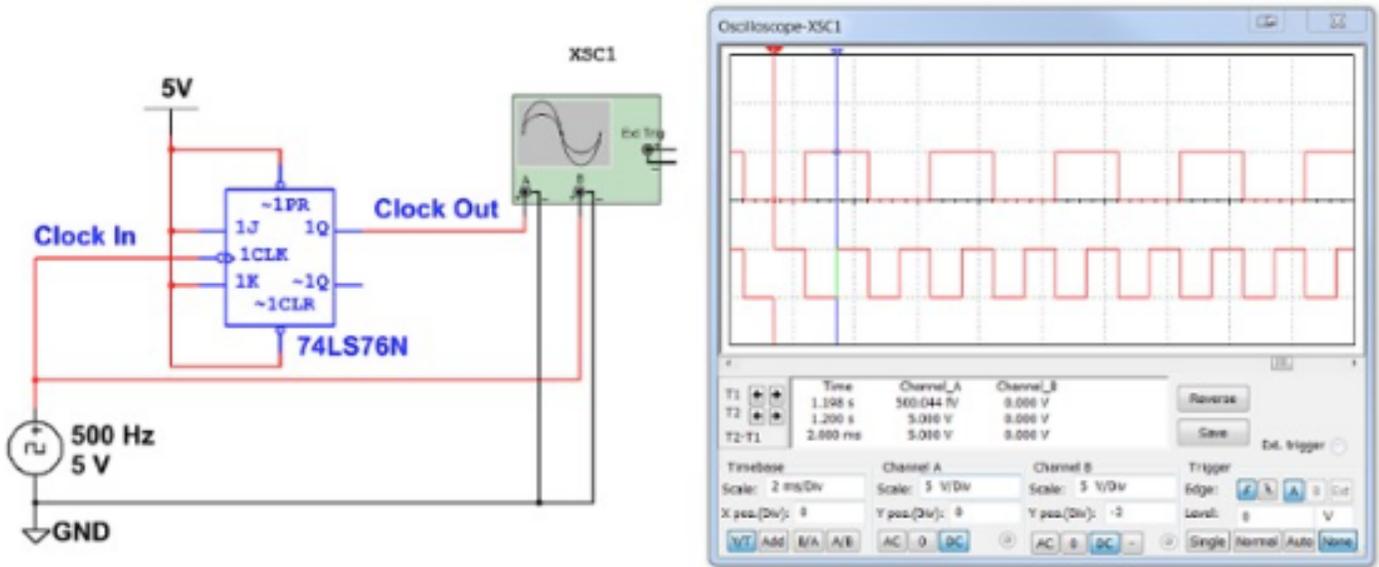
3. When flip-flops were discussed briefly back in Unit 1, we saw that a D flip-flop could be used to create a *Divide-by-Two* circuit. Remember, a *Divide-by-Two* circuit is one that generates a **clock** output that is half the **frequency** of the clock input. Likewise, a *Divide-by-Two* circuit can be implemented with a J/K flip-flop.



In your notebook, complete the timing diagram for a J/K *Divide-by-Two* circuit.

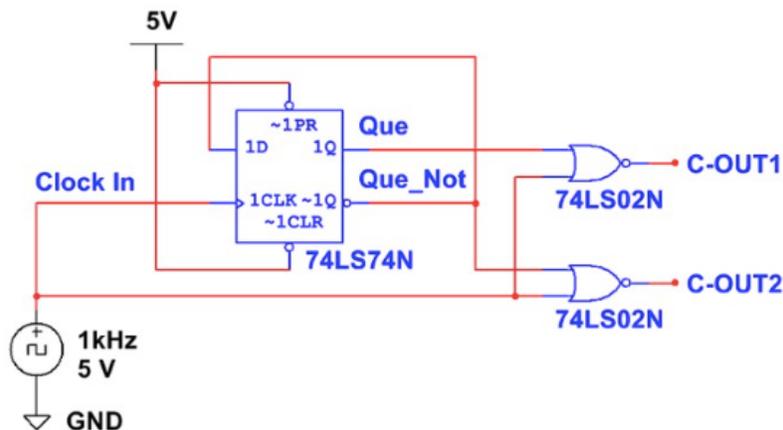


- Using the CDS, enter the *Divide-by-Two* circuit. Add an oscilloscope to monitor the two signals **Clock_In** and **Clock_Out**. Run the simulation and capture several **periods** of the output signal. Verify that the circuit is working as expected and that the output signal matches the predictions from step 3a. If the results do not match, review your work and make any necessary corrections.



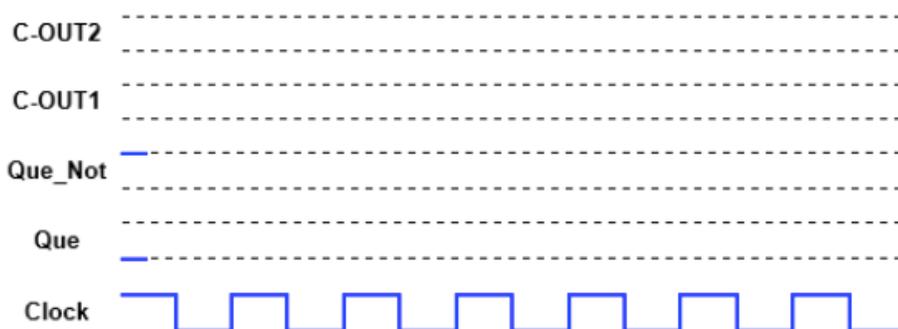
- Change the frequency of **Clock_In** to 20 kHz and re-simulate. What effect did this change have on the frequency of the output signal **Clock_Out**?

The circuit shown below generates two non-overlapping signals at the same frequency. These signals, **C-OUT1** and **C-OUT2**, were frequently used by early microprocessor systems that required four different clock transitions all synchronized by one clock.



Non-Overlapping Signal Generator

In your notebook, complete the timing diagram below for the *Non-Overlapping Signal Generator* circuit.



- a. Using the CDS, enter the *Non-Overlapping Signal Generator* circuit. Add an oscilloscope to monitor the three signals **Clock**, **C-OUT1**, and **C-OUT2**. Run the simulation and capture several periods of the output signals. Verify that the circuit is working as expected and the output signals match the predictions from step 5a. If the results do not match, review your work and make any necessary correction.
- b. The input signal, **Clock**, was a 1 kHz square wave with a 50% **duty cycle**. What is the frequency and duty cycle of the output signals **C-OUT1** and **C-OUT2**?
- c. Change the frequency of **Clock** to 2 KHz and re-simulate. What effect did this change have on the frequency of the output signals **C-OUT1** and **C-OUT2**?
- d. What effect did this change have on the duty cycle of the output signals **C-OUT1** and **C-OUT2**?