

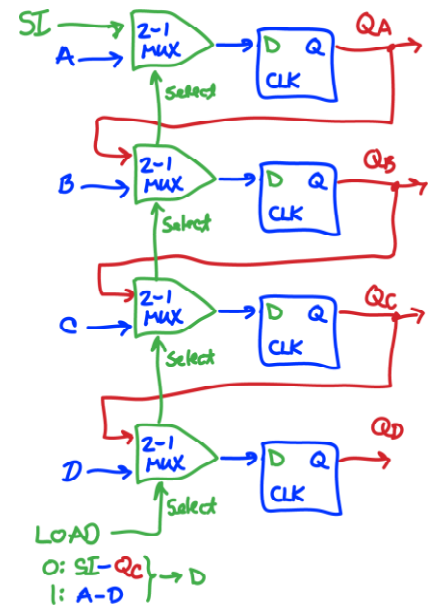
Laboratory Nine

Flip-Flops and Registers

Basic Concepts

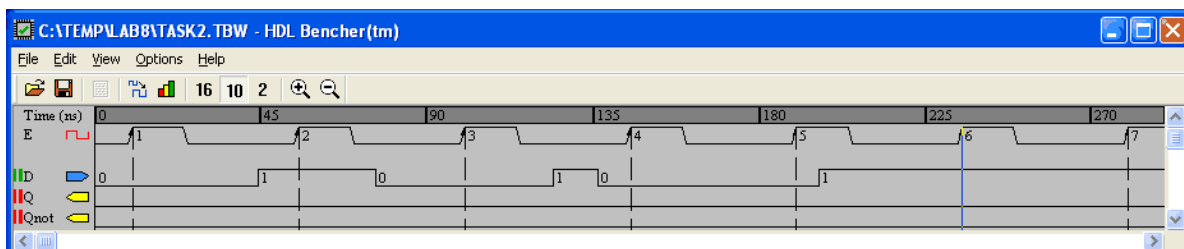
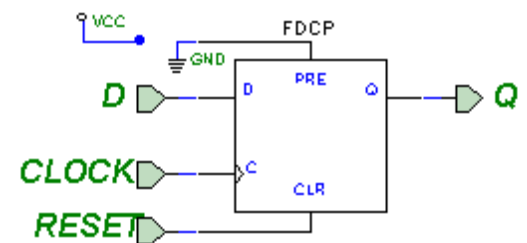
1. A *flip-flop* is a binary storage element designed specifically to work with a clock signal (**CLOCK**). There are 2 basic types of flip-flops, the **D (data) flip-flop** and the **JK flip-flop**. Flip-flops are designed to make state changes only on the *rising* or *falling edges* of the **CLOCK**.
2. A *shift register* consists of a string of **D** flip-flops, which are able to exchange data with their nearest neighbors, as seen by the red paths at right. The *serial input (SI)* determines what the first **D** flip-flop in the string will receive. 2-to-1 **MUX**s help also to allow a *parallel load*, seen in blue at right, selected by the green **LOAD** signal.
3. The traditional *counter* consists of **JK** flip-flops operating in the toggle mode to make count sequences.

Note: This prelab assignment is worth 10 points. Lab is still 10.



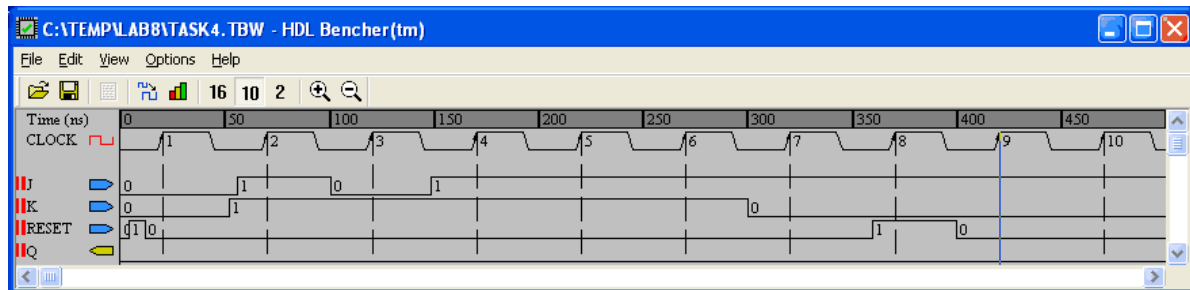
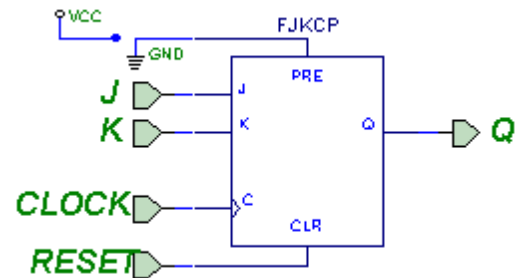
Task One: The 7474 D Flip-Flop

1. (Pre) Draw the **D flip-flop** shown in your lab notebook.
2. (Pre) Review your D-latch results from last week, and record the **Q** output changes below, keeping in mind that for this **D flip-flop**, state changes can only be performed on the rising edge of the **CLOCK**. Mark the mode of operation (*Set* or *Reset*) above each clock edge.
3. Simulate your results in **Xilinx**, using the **FDCP D-latch** block, shown at right. Route an additional signal, called **RESET**, to the **CLR** input to clear the flip-flop.
4. Assign ucf pins for 2 input switches, a debounced switch and 1 LED for **Q** and download your latch into the Digilent board. (The **CLOCK** input must be connected to the debounced switch.) While **RESET** is **high** in the **7474**, construct a truth table that illustrates the **D F/F** properties, having inputs **D** and **Q_n** and output **Q_{n+1}**.
5. Now, exercise the **CLOCK** and **D** inputs of the flip-flop according to your bench signals. To begin, clear the **Q** output by switching **RESET** from **high** to **low** to **high**.



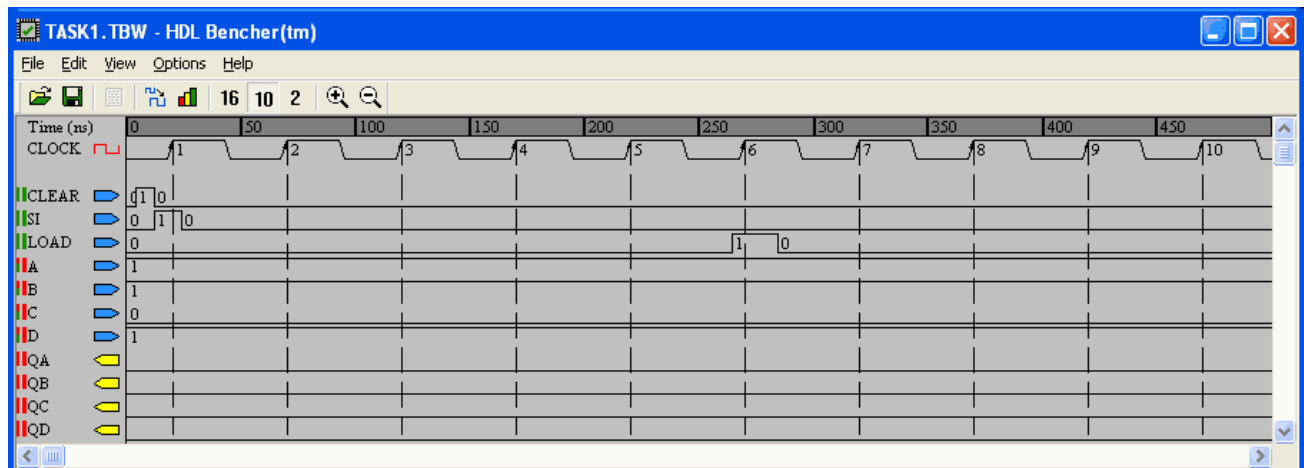
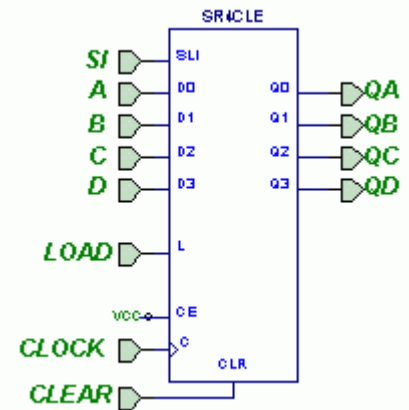
Task Two: The 74112 JK Flip-Flop

1. (Pre) Draw the **JK flip-flop** shown at right in your lab notebook. The JK truth table is shown in Table 5.7.
2. (Pre) Draw the timing diagram shown below for waveforms **CLOCK**, **J**, **K**, **RESET**, and complete the waveform for **Q**. Mark the mode of operation (*NC*, *Set*, *Reset*, *Toggle*) performed on each rising edge of the clock.
3. Simulate your results again in *Xilinx*, using the **FJKCP** model shown at right. Note again that the **PRE** and **CLR** signals are active-high with the **FJKCP**. Also, the **FJKCP** is positive-edge triggered.
4. Assign ucf pins for 3 input switches, a debounced switch (for **CLOCK**) and 1 LED for **Q** and download your latch into the Digilent board. With **RESET high**, construct a truth table that illustrates the **JK F/F** properties, having inputs **J** and **K** and Q_n and output Q_{n+1} .
5. Put the flip-flop in toggle mode and explain what happens each time you press the debounced switch.
6. Exercise the **RESET**, **CLOCK** and **JK** inputs of the **JK flip-flop** according to the bench signals below. Record the resulting responses on prelab waveforms in your lab book. How do your experimental waveforms differ from your simulation?



Task Three: 4-Bit Shift Register

1. (Pre) Draw the 4-bit shift register symbol given at right in your lab notebook. No pin numbers are necessary, here.
2. Draw the timing diagram shown below and label the input waveforms **CLOCK**, **CLEAR**, **SI** (serial-input), **LOAD** (synchronously load parallel inputs), **A-D** (parallel inputs) and blank output waveforms **QA-QD**.
3. Place a **SR4CLE** 4-bit shift register into *Xilinx* and simulate your results employing the sample waveforms shown below. Then download your design to the **Digilent** board and record the observed output **QA-QE** pattern on the blank waveforms in your notebook.



Task Four: The 74163 4-bit Counter

1. (Pre) Draw the **74163 MSI** counter (use Fig. 7.8 and the [MSI chip data sheet](#)), showing pin numbers.
2. (Pre) Complete the *simplified* state table (truth table) shown at right for the **Present** and **Next State (PS & NS)** outputs Q_D-Q_A . (See further Table 6.8, where it shows that $NS = PS + 1$) Assume that Q_A is considered the least significant bit, when counting.
3. Draw the **74163** timing diagram shown below and label the input waveforms **CLOCK**, **CLEAR** (synchronous-clear), **LOAD** (synchronous-load) and **D-A**. Label the (blank) output waveforms Q_D-Q_A . The **RC** output shall be ignored. Note that in the **74163**, **CLEAR** and **LOAD** are active-low and synchronous, rather than asynchronous signals as in the 74161.
4. Build a **VHDL** model of the **74163** with **Xilinx**, using the [74163 code](#) that your lab instructor will give you.
5. Simulate your **74163 VHDL** model in **Xilinx**, using the waveforms below. Observe (but do not record) the output Q_D-Q_A pattern.
6. Instead of downloading, we will wire the **74163 MSI** counter chip up with power, input and output connections on the red box, and verify that this counter counts. The **CLOCK** input must be debounced and the **Q** outputs should be connected to LEDs.
7. Repeat the counting experiment using an ordinary non-debounced switch for the clocking signal. You should see some skipped numbers.
8. Exercise the **74163** according to the waveforms below and record the results on the blank waveforms. How do they compare with the simulated results?

PS				NS			
Q _D	Q _C	Q _B	Q _A	Q _D	Q _C	Q _B	Q _A

