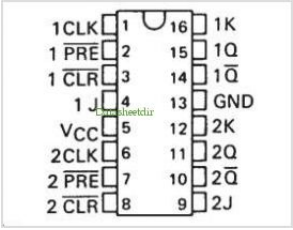
cs281: Computer Organization

Lab05 - JK Flip Flops and Sequential Circuits

1 Exploration: The JK Flip-Flop

The 7476 chip provides the hardware realization of *two* JK Flip-Flops, numbered 1 and 2. The pinouts for the 7476 are given below. In this exploration, we want to put the 7476 on our breadboard, and wire it up and experiment with it so that we can better understand its operation.



First, like with any chip, we need to provide GND and Vcc (+5) to the chip, so you will wire +5 to pin 5, and GND to pin 13. If you look at the labels for flip-flop 1, and for flip-flop 2, you can see the inputs and outputs we expect, ($1J$, $1K$, $1Q$, $1\overline{Q}$ for FF 1) . We also observe that each flip-flop has *its own* clock, 1CLK for FF 1. Finally, we see two other pins for each flip-flop, namely $1\overline{PRE}$ and $1\overline{CLR}$ for FF 1 and like signals for FF 2. These are inputs that allow us to *preset* (set to 1) and *clear* the value stored in a way that happens asynchronously to the state of the clock. The bar over each indicates that they are *active low*, meaning that they change the stored value when the input is 0 (rather than 1). For our purposes, we can wire both to +5V, so that they do not affect our experimentation.

Now I want you to manipulate the J and K inputs and the CLK for one of the flip-flops to gain experience with how it works. If you do things manually, you could wire all three to switches, or you could wire J & K to switches and CLK to the function generator, set to run at a very slow pace. You should wire the output Q to a Logic Indicator so that you can observe the value of the stored bit (0 or 1) and also observe, relative to the clock, when the stored value changes (this implies that you need the clock to be wired to a Logic Indicator for observation as well).

By the end of this exploration, you should have been able to set the storage using all combinations of J & K, and should also be able to answer the question, *"At what point in the clock cycle does the JK flip flop employed take on its new value?"*

2 Design: Sequence Detector

In a *Moore* circuit, like the ones introduced in lecture and in the prelab, the outputs depend only on the present state. A general model of a Moore sequential circuit consists of two combinational circuits – one to compute the next state and associated flip flop input values from the present state and the inputs, and a second circuit to compute the output based on the present state. These two combinational circuits

are combined with a set of 1-bit memories (collectively called the "state register"), which encode and remember the present state. The state register normally consists of D or JK flip-flops.

A *sequence detector* is a circuit that serially examines a string of 0's and 1's applied to the X input and can generate an output Z when the sequence matches a particular pattern. For this lab, the pattern you are looking for is the subsequence 111. For this specification, the circuit does not reset when the output is matched. A typical input sequence and the corresponding output sequence might be:

X	=	0	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0
Z	=	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

1. Design a Moore-style finite state machine that describes the operation of this circuit. Identify how many 1-bit memory elements are required to represent your set of states.
2. Draw a truth table that describes the next state function of the finite state machine.
3. Draw a truth table that describes the output function of the finite state machine.
4. Use K-maps to simplify the boolean expressions for each of your outputs.
5. Design a sequential circuit from your boolean expressions. Use JK flip-flops.
6. Implement your circuit on the breadboard.
7. Demonstrate for your assistant or instructor.

You should turn in, by Monday, a coherent and complete lab report that describes your design and gives pictures of each of the circuits in your work.