# Lab 4: Shift register

## What is a shift register?

A K-bit shift register holds K bits of memory, and has a capability of performing the shift operation. Suppose we have an 8-bit shift register, and the current bit string in the shift register is:

0110 1001

After performing a "left-shift" operation, all the bits in the register move "left" by 1 bit position, discarding the leftmost bit and injecting a new rightmost bit of value 0. The result looks like this:

1101 0010

---- ---^

Traditionally, the discarded leftmost bit is said to have been “put in the bit bucket.” A more general version of the left-shift operations allows the value of the inserted bit to be specified. Suppose, in the previous example, the value of the shift-in bit is 1, then the result would be:

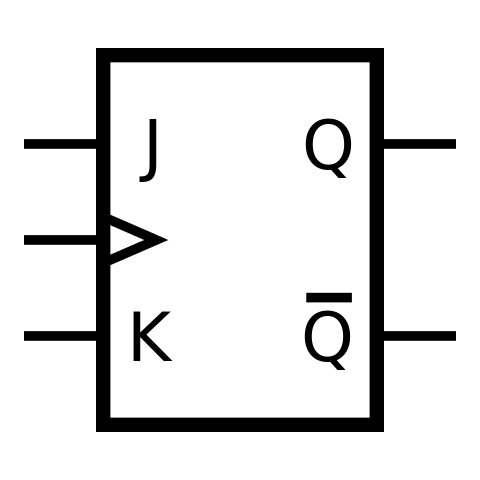
1101 0011

---- ---^

## 

## 

## J-K flip flop



To build a shift register, one can use flip-flops - one flip-flop for every bit of memory in the register.

There are many types of flip-flops. Most flip-flops respond to clock input. The simplest flip-flop in terms of functionality, is the D flip-flop: when the clock input triggers, the value that the flip-flop (memory) holds becomes the value of the input D, and is made available to other circuits as flip-flop outputs Q and Q’.

The J-K flip flop that we will use in this lab is more complex. It has 2 inputs, in addition to the clock, called J and K. When the clock input arrives, the two inputs J and K command the storage (memory) of a bit as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Characteristic Table** | | | **Excitation Table** | | | |
| **J** | **K** | **Flip-Flop Behavior** | **Q(t)** | **Q(t+1)** | **J** | **K** |
| 0 | 0 | Hold current memory bit value: Q(t+1) = Q(t) | 0 | 0 | 0 | X |
| 0 | 1 | Reset the memory bit to 0: Q(t+1) = 0 | 0 | 1 | 1 | X |
| 1 | 0 | Set the memory bit to 1: Q(t+1) = 1 | 1 | 0 | X | 1 |
| 1 | 1 | Invert current memory bit: Q(t+1) = ​NOT( Q(t) ) | 1 | 1 | X | 0 |

The left half of the above is called the **characteristic table**, because it tells the “character” of the flip-flop, or how the memory bit changes according to the control inputs. The right half of the table is called the **excitation table** because is shows what inputs to J and K are needed to “excite” or drive the flip flop from one memory state to another.

## Control logic for JK Flip-flop in a Shift Register

In this lab, you will build the circuit for a 2-bit shift register. The 74LS112 chip, holding 2 J-K flip-flops, will hold the 2 bits of memory. The 74LS112 flip-flops are negative-edge triggered, meaning their outputs change on the negative going edge of the clock signal.

The register should be clocked using the 555 timer chip, set to clock at 1 second intervals.

There are three data input signals to the register: D0, D1 and SHIFT\_IN. All three input signals are created by pushbutton voltage divider input circuits.

There are two control inputs to the register, also implemented using pushbutton voltage divider circuits, called LOAD and SHIFT. These control inputs will direct register operation as follows:

* LOAD input, a high-active signal:  
  When this input is active, the register will load the current values of the data inputs, D0 and D1, where D0 is the least significant bit. With an active LOAD command input, Q0 of the flip-flop holding the least significant bit, FF0, becomes the value of the D0 input after the negative clock edge, and Q1 of FF1 becomes the value of the D1 input. In other equations, Q0(t+1)=D0 and Q1(t+1)=D1 with the negative clock edge.
* SHIFT input, a high-active signal:  
  When this input is active, the register should perform a 1-bit left shift with every negative edge in the clock signal. That is, Q0(t) output of flip-flop zero (FF0) is the value of the SHIFT\_IN data input at the time of the negative clock edge, and Q1(t+1) output of flip-flop 1 (FF1) becomes the value once held by FF0, the value of Q0(t). The bit shifted out of the most significant position must go into a bit bucket that you implement.
* SHIFT and LOAD are not supposed to be active at the same time. So we will not push both of the LOAD and SHIFT pushbuttons at the same time.
* When LOAD and SHIFT are both not active, then register should retain the current stored bits for for as long as electrical power is supplied. Mathematically, this means Q0(t+1) = Q0(t) and Q1(t+1) = Q1(t).

The register outputs are those from the J-K flip-flops: Q0 and Q1.

Fill in the missing entries in a truth table for the above operation description.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LOAD | SHIFT | Q0(t+1) | Q1(t+1) | Meaning |
| 0 | 0 | Q0(t) | Q1(t) | Hold |
| 0 | 1 | SHIFT\_IN |  |  |
| 1 | 0 |  | D1 |  |
| 1 | 1 |  |  |  |

The four register inputs Clock, Shift\_In, D0, and D1 are to be made visible with green LEDs. The two register outputs Q0 and Q1 are to be visible using red LEDs.

Designing the circuits around the JK Flip-Flops

To implement the register, you will need design the Boolean functions for the J and K inputs for both flip-flops and build your circuit design. These circuits control the flip-flops by appropriately commanding them with the necessary J and K logic values, and then we wait for the clock signal’s falling (negative) edge to trigger the action in the flip-flops.

The new aspect of designing the J and K circuits is that we have to consider the sequential, not combinatorial, behavior of the flip flops. We want to use the J and K inputs to *excite* or drive the flip-flop to the correct next state.

For our needs here, the JK characteristic table can be focused on using the JK as a D-type flip-flop to load in data from D-inputs or to load in data from the *shift path*. This gives this table, where the Toggle T-type flip-flop) capability of a JK flip flop is not needed:

|  |  |  |
| --- | --- | --- |
| J | K | Operation |
| 0 | 0 | Hold |
| D | D’ | Q(t+1) = D |
| 1 | 1 | Toggle (invert Q(t) |

Then we can derive the Boolean expression for J0 as:

J0 = (SHIFT and SHIFT\_IN) or (LOAD and D0), while

K0 = (SHIFT and SHIFT\_IN’) or (LOAD and D0’).

You should now determine the logic equations for J1 and K1.

Now you can draw the circuit diagram, choose the correct chips to use, and build it on your breadboard.

You may find that, like a new video game, the “action” of your shift register is too fast to easily control in the way you wish. If this is so, then you can practice at half the speed by revising the clock circuit to run half as fast using available lab kit components. This little hardware “cheat” is just fine for you to use in this lab, even when demonstrating your circuit. With enough practice you may want to remove the cheat.

New chip: 74LS112 - two J-K flip-flops

This webpage contains a list of datasheets for 74LS112 chip: <http://www.datasheetcatalog.com/datasheets_pdf/7/4/L/S/74LS112.shtml>

Here are some notes about the chip you are going to use:

* The J-K flip-flops on 74LS112 are triggered by negative edges: they are triggered when clock input goes from high to low, as opposed to the counter chip, 74163, triggered when clock goes from low to high. However, since we use the same timer chip and external components to drive the flip-flops, the trigger still happens once per second.
* The J-K flip-flops on 74LS112 have two additional control inputs for each flip-flop: PRESET and RESET, both of which are low-active and asynchronous. If RESET is active, the flip-flop output changes to 0; if PRESET is active, the flip-flop output changes to 1. However, these inputs are **asynchronous**, the PRESET/RESET will happen **at once** (immediately) when they are active, rather than waiting for the clock to trigger the operation. Therefore, technically you will not use these inputs, so please deactivate them by connecting these inputs to a constant voltage.

In lab next week:

* Demonstrate your circuit to your TA. [50 pts]
* Show your completed table above. [20 pts]
* Show your circuit schematic. [30 pts]

Use Blackboard to submit your work in the form of a PDF file.