CS 250 Fall 2017 Lab 04

Shift Register

**Problem 1 [20 points]** Fill in the missing entries in the truth table for the above operations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LOAD** | **SHIFT** | **Q0(t+1)** | **Q1(t+1)** | **Meaning** |
| 0 | 0 | Q0(t) | Q1(t) | Hold |
| 0 | 1 | SHIFT\_IN | Q0(t) | SHIFT |
| 1 | 0 | D0 | D1 | LOAD |
| 1 | 1 | X | X | X |

**Problem 2 [10 points]** Write the full logic equations for K0, J1, and K1 in terms of the available control and input signals.

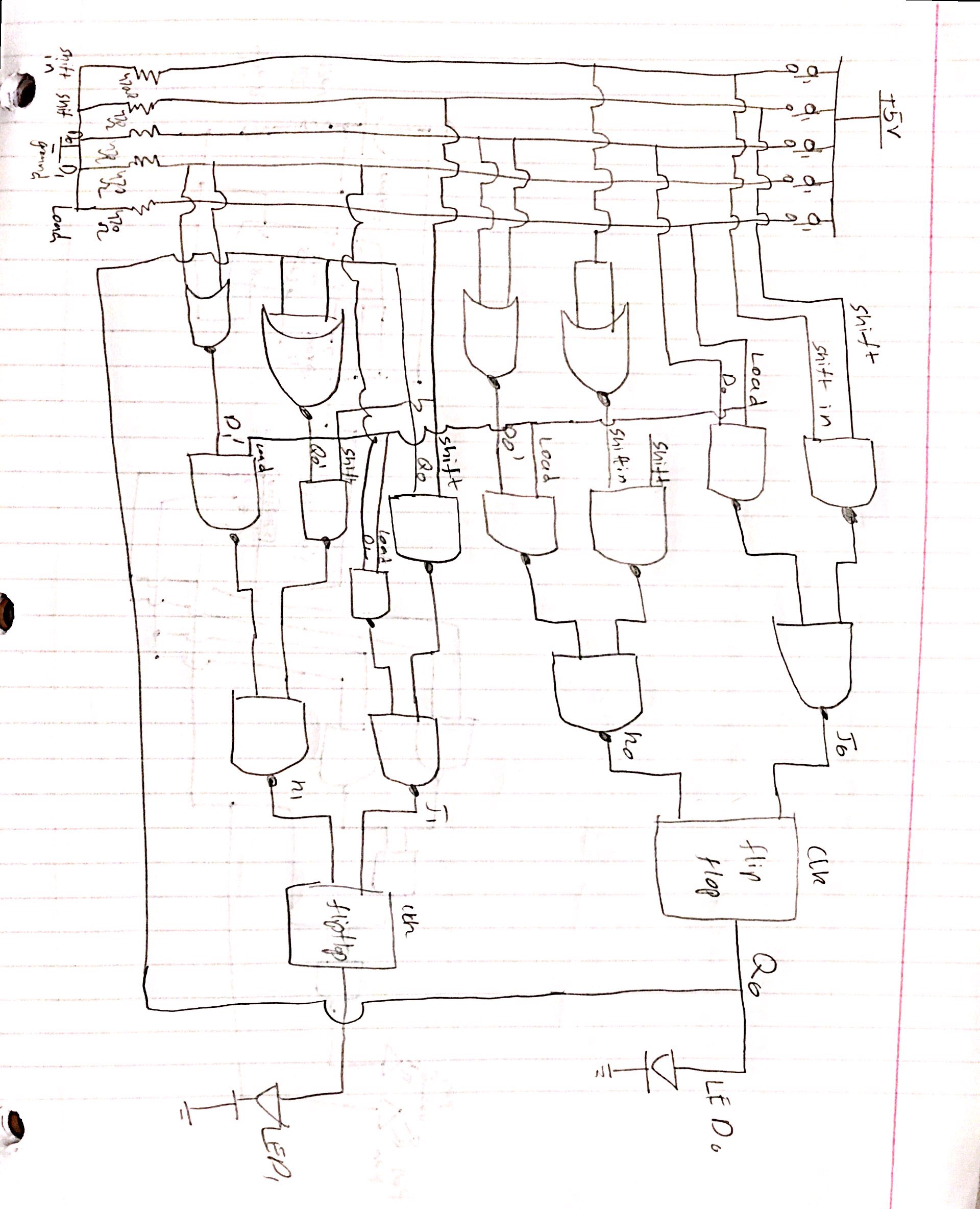
K0 = (SHIFT and SHIFT\_IN’) or (LOAD and D0’)

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

J1 = (SHIFT and Q0) or (LOAD and D1)

K1 = (SHIFT and Q0’) or (LOAD and D1’)

**Problem 3 [30 points]** Draw the circuit diagram for your design.



**Problem 4 [30 points]** Construct your circuit. Demonstrate your 2-bit shift register to your TA in your next lab session.

NOTE: 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.

**Problem 5 [10 points]** Link your breadboard with those of ~4 other students anddemonstrate your ~10-bit shift register to your TA.

**Instructions**

A K-bit shift register holds K bits of memory and can shift the bits in the register to adjacent bit positions. Suppose we have an 8-bit shift register, and the current bit string in the shift register is:

0110 1001

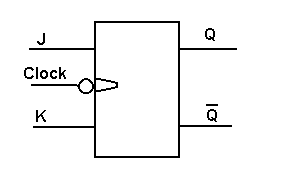
After performing a “left-shift” operation, all of the bits in the register move “left” by 1 bit position, discarding the leftmost bit and injecting a new rightmost bit of value determined by us as designers.If we choose to shift in a zero, then the register contents after the left-shift is:

1101 001**0**

Traditionally, the leftmost bit, which is lost to the circuit, is said to have been “put in the bit bucket.”

**1. The J-K Flip-flop**

J-K flip-flop



There are many types of flip-flops. Flip-flops, by definition, respond to a clock input signal. The simplest flip-flop in terms of functionality is the D flip-flop: when the clock input transitions high, the value stored by the flip-flop (memory) takes on 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 capable. It has 2 inputs called J and K. When the clock transitions, the two inputs J and K determine the logic level to be stored and output on Q:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 table 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 it shows what inputs to J and K are needed to “excite” or drive the flip-flop from one state to another for all possible combinations.

**2. Shift Register Inputs and Control Signals**

In this lab, you will build the circuit for a 2-bit left-shift register. The 74LS112 chip, containing two J-K flip-flops, will hold the two bits of memory required. The 74LS112 flip-flops are negative-edge triggered, meaning their outputs change on the negative edge of the clock signal (as the clock transitions from high to low).

Your shift register should be clocked using the 555 timer chip, with a clock interval of 1 second. There should be three input signals to the shift register: D0, D1 and SHIFT\_IN. All three input signals should be provided by push-button voltage divider circuits.

In addition, there should be two control inputs to the shift register, also sourced from push-button voltage divider circuits. The control signals are LOAD and SHIFT. These control inputs will direct the shift register operation as follows:

**2.1 LOAD, an active-high signal**

When asserted, the shift register will load the current values of the data inputs, D0 and D1, with D0 representing the least significant bit. When the clock transitions from high to low, each flip-flop should store their respective value. That is, Q0 (or FF0), should assume the value of D0; and Q1 (or FF1) should assume the value of D1. In equation form,  
Q0(t+1)=D0  
Q1(t+1)=D1  
with the negative clock edge.

**2.2 SHIFT, an active-high signal**

When asserted, the shift register should perform a 1-bit left shift for every negative clock edge. That is, the output of flip-flop one (Q1 or FF1), the most significant bit, should take on the value of flip-flop zero (Q0 or FF0), the least significant bit. The output of flip-flop 0 (Q0 or FF0) should take on the value of SHIFT\_IN. In equation form,  
Q1(t+1) = Q0(t)  
Q0(t+1) = SHIFT\_IN  
  
The bit shifted out of the most significant position must go into a bit bucket that you implement. Don’t overthink what is required to implement a bit bucket.

**2.3 Remaining functionality**

* You may assume that SHIFT and LOAD will not be asserted simultaneously. If they are, the operation of your circuit will be undefined.
* When neither LOAD nor SHIFT are asserted, the shift register should retain the currently stored bits for as long as electrical power is supplied. Mathematically, this means:  
  Q0(t+1) = Q0(t) and   
  Q1(t+1) = Q1(t)

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

* The four shift register inputs CLK, 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.

**3. Circuit Implementation – IMPORTANT**

**We will be combining the breadboards of ~5 students to demonstrate a ~10-bit shift register. Breadboards will first be positioned so that their long sides are adjacent and row numbers match up. In this configuration, breadboards may be interconnected using the small “ears” along the long edge closes to tie points column A.**

**To facilitate connecting the circuit on one breadboard to the circuit on the next breadboard, and to improve the viewing experience for the combined breadboards circuit, the following implementation requirements, a form of hardware interface definition, are imposed:**

1. **Pin 8 of 74112 (JK chip) must be placed in breadboard tie point E63.**
2. **The LED for Q0 must be placed at 74112 pin 9, and LED for Q1 at pin 5.**
3. **A wire sufficiently long to reach from tiepoint J59 (74112 Clock 2 pin) to tiepoint J59 on an adjacent breadboard must be available (for daisy-chaining clock signals from one breadboard of the ~5 that are linked.**
4. **All chips used in your design should be placed as close together as possible so that the circuit can fit on the breadboard.**
5. **The pushbutton inputs must be placed side by side at the row 1 end of the breadboard with the button for SHIFT-IN closest of all the pushbuttons to row 1.**
6. **SHIFT-IN and LOAD must be implemented as active high pushbutton inputs.**
7. **Each student will know how to quickly disconnect their on-board SHIFT\_IN signal from their circuit and replace this signal with a connection to tie point A60 (74112 pin 5), the Q1 signal, on the breadboard containing the register bit immediately to the right in the linked circuit.**
8. **A wire sufficiently long to reach from tiepoint J59 (74112 Clock 2 pin) to tiepoint J59 on an adjacent breadboard must be available (for daisy-chaining clock signals from one breadboard of the ~5 that are linked.**
9. **Two wires sufficiently long to link the ground and power rails of one breadboard to the adjacent breadboard must be available. (Only one of the linked breadboards will be connected to USB power. This breadboard will also supply the clock signal to the circuits on the other boards.)**

To design the shift register, you will need to determine the Boolean functions for the J and K inputs for each flip-flop. The circuits implementing these functions will in turn “command” the flip-flops appropriately by providing the necessary input values for J and K, read on the falling edge of the CLK signal.

This time it is important to consider the sequential behavior of the circuit, not just the combinational behavior. We want to use the J and K inputs to excite or drive the flip-flop to the correct next state.

For our purposes, we can carefully provide the JK input values in a way that causes the flip-flop to behave as a D-type flip-flop instead. See the revised characteristic table below. While this approach additionally provides us with toggle capability, we do not need it to implement the shift register.

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

Using the new operational semantics specified above, we can derive the Boolean expressions for J0 and K0 as follows:

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

**4. Notes on the 74LS112**

* The following webpage contains a list of datasheets for the 74LS112 chip: <http://www.datasheetcatalog.com/datasheets_pdf/7/4/L/S/74LS112.shtml>
* The J-K flip-flops provided by the 74LS112 are triggered by negative clock edges: they are triggered when the clock input goes from high to low, as opposed to the counter chip, 74163, which is positive edge triggered. 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 in the 74LS112 have two additional control inputs for each flip-flop: PRESET and RESET, both of which are active-low and asynchronous. If RESET is asserted, the flip-flop output changes to 0; if PRESET is asserted, the flip-flop output changes to 1. Because these inputs are asynchronous, the PRESET/RESET will happen at once (immediately) when they are asserted, rather than waiting for the clock to trigger the operation. Therefore, you should not use these inputs. You may deactivate them by connecting the pins to a constant high voltage (thus de-asserting the active-low inputs).

**5. Due Date**

At the beginning of your lab session next week.