Lab 10

Lab 10 assignment is due on Friday, April 17, by 9pm in Blackboard.

Please ask all questions about this assignment during the office hours or post questions on piazza.

This lab requires to assemble a digital circuit on a prototyping board. This requires access to a hardware lab, which is not possible at the moment. Therefore, we will not be assembling any hardware at this time. Instead, please carefully read the provided lab materials and complete and submit the lab worksheet provided at the end of this page.

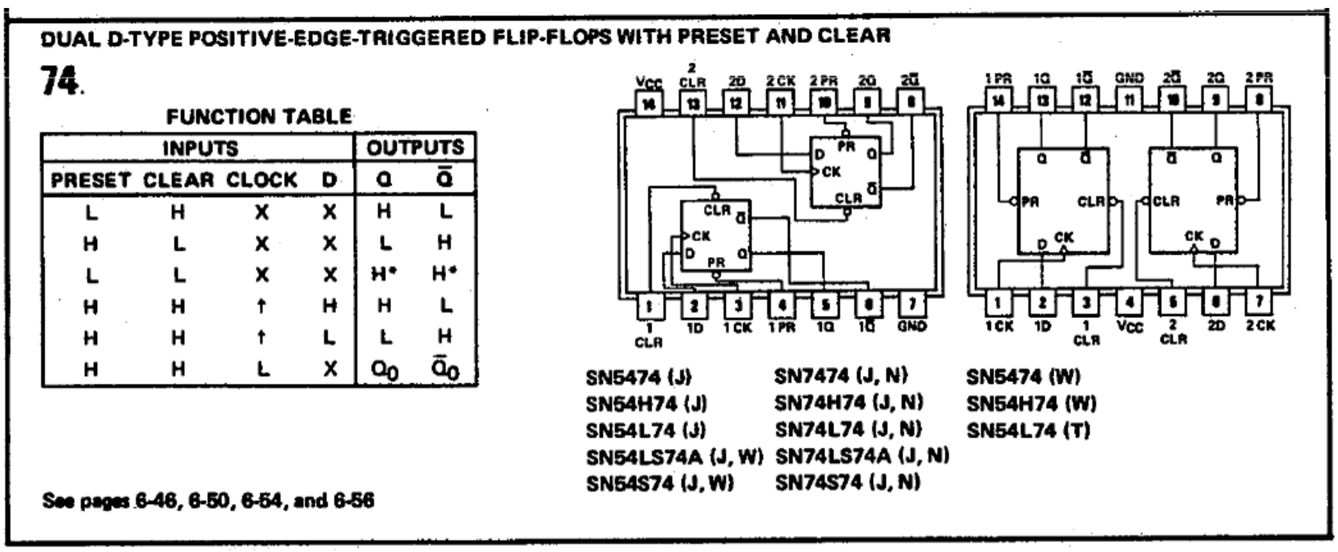
Sequential logic implementation

In this lab, you will implement on your protoboard the vending machine example from Section 3.3 of the Lumetta course notes, “Design of the Finite State Machine for the Lab.” In Lab 8, you implemented a combinational logic circuit that produced two output signals **A** (for "accept coin") and **P** (for "dispense product") based on the state $S_2S_1S_0$. In lab 9, you designed and simulated the finite state machine using Altera Quartus. In this lab, you will complete the protoboard implementation of the state machine. You will test it with the sensor/actuator device described below.

Implementing the State Machine on the Protoboard

If needed, redesign your next-state logic design from Lab 9 so that it uses only NAND, NOR, and NOT gates.

For flip-flops, we recommend that you use either the 7474 DIP or the 74175 DIP. The top data sheet shows the pin layout for the 7474 DIP which has two distinct positive-edge-triggered D flip-flops. Note the behaviors of the PRESET and CLEAR. The lower data sheet shows the pin layout for the 74175 DIP which has four positive-edge-triggered D flip-flops which share common CLEAR and CLOCK signals. When the CLEAR signal is 0 it asynchronously resets all of the flip-flops to store 0.


DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

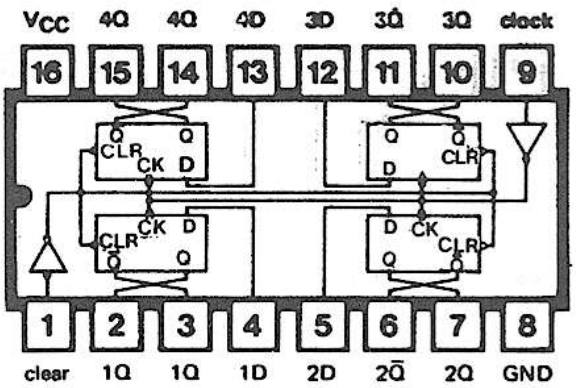
74.

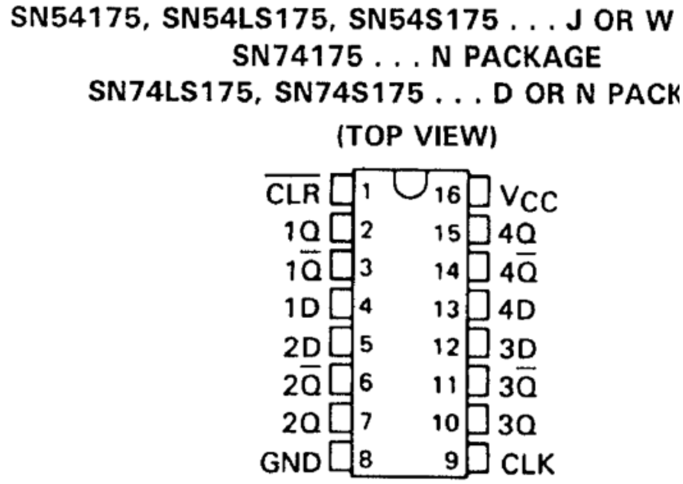
FUNCTION TABLE

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\bar{Q}_0$

SN5474 (J) SN7474 (J, N) SN5474 (W)
SN54H74 (J) SN74H74 (J, N) SN54H74 (W)
SN54L74 (J) SN74L74 (J, N) SN54L74 (T)
SN54LS74A (J, W) SN74LS74A (J, N)
SN54S74 (J, W) SN74S74 (J, N)

See pages 6-46, 6-50, 6-54, and 6-56



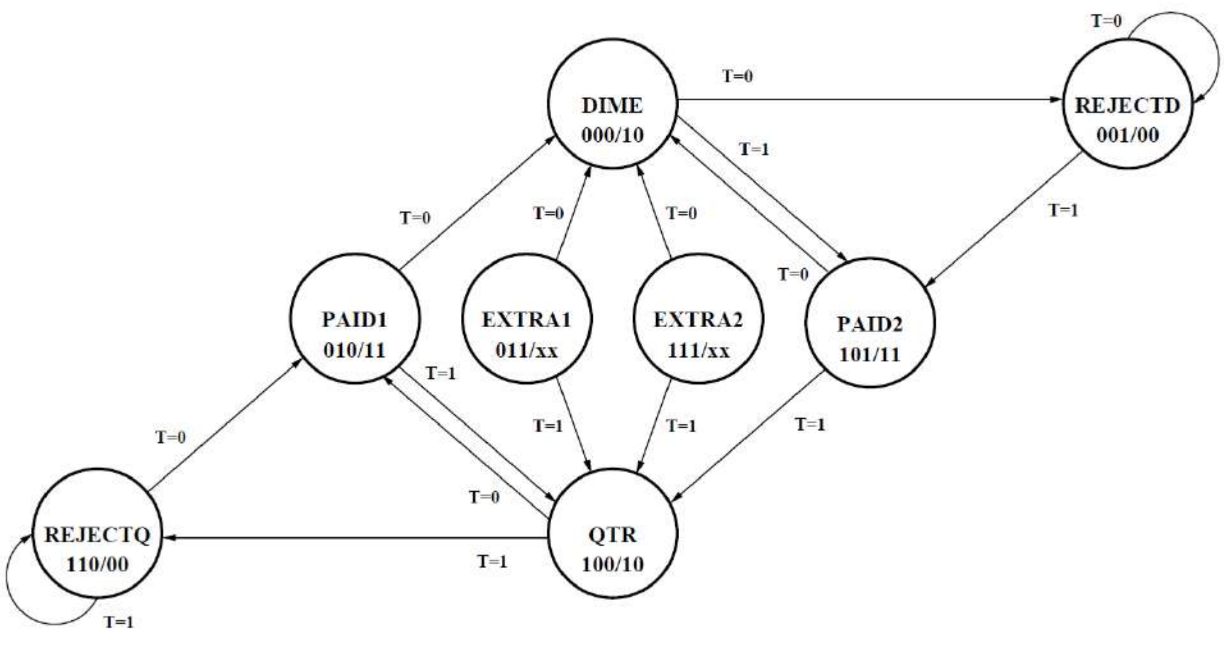


Implement your state machine and connect it to your circuit from Lab 8 that generates your outputs A and P. **Don't forget to first detach your switch inputs to your Lab 8 circuit in order to avoid driving them against the state outputs and possibly burning out your flip-flops.**

Connect your current state variables S_2, S_1, and S_0, and outputs A and P to LEDs so that you can easily check to see whether your circuit implements the correct state transitions and outputs.

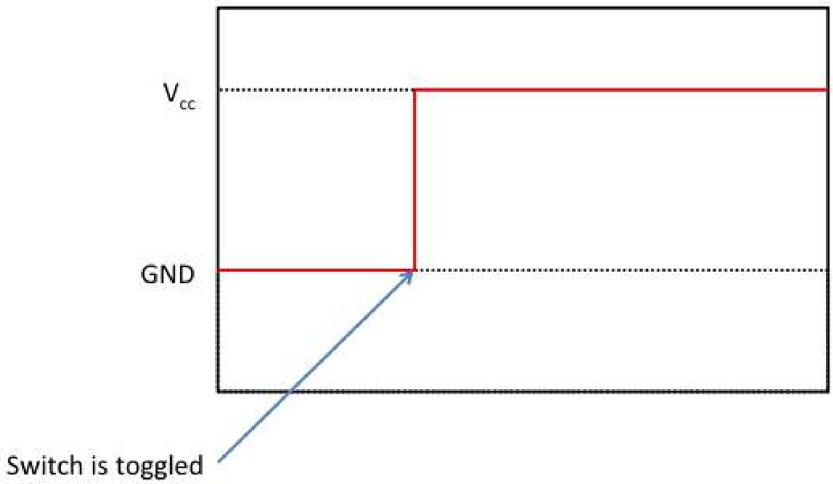
Debugging

The key for successfully demoing this lab is to make sure you **throughly debug your circuit before you demonstrate the ciorcuit**. To do that, you need to go through all possible state transitions:

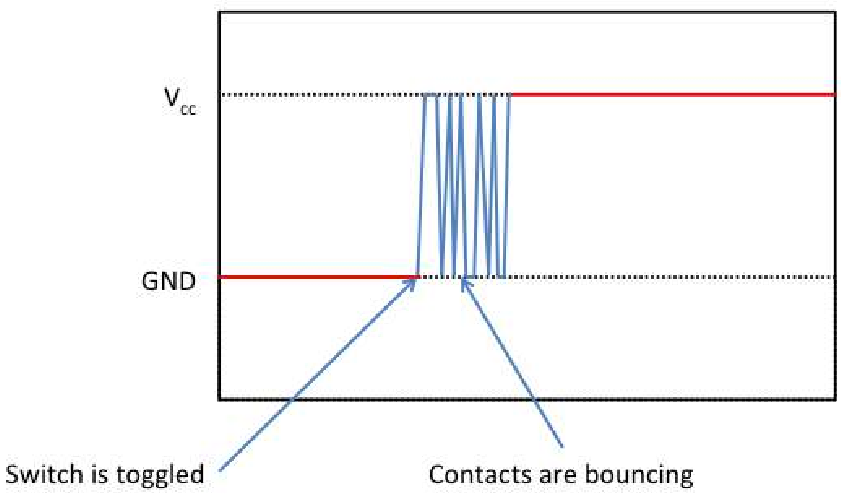


Be aware that you may not be able to reach states EXTRA1 and EXTRA2, so it is OK not to test those four state transitions leaving the unreachable states. You are expected to test all other state transitions though.

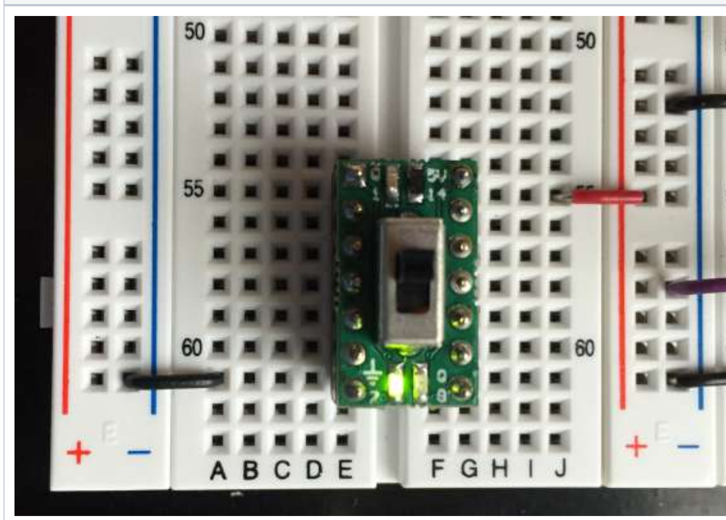
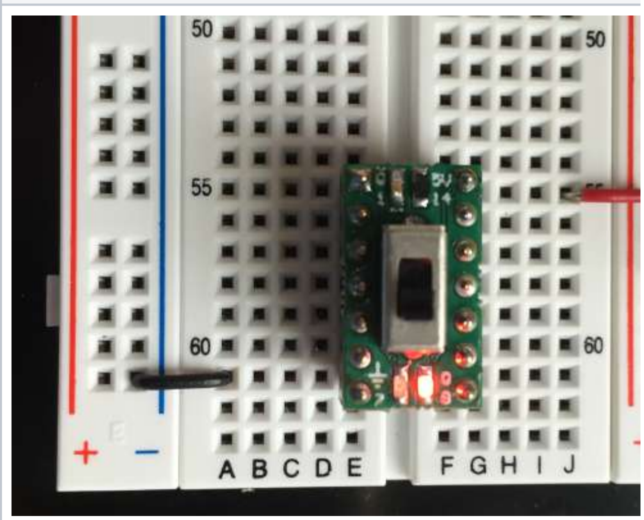
Test your circuit by simulating the T and CLK signals. To simulate T, you can use one switch input similarly to how you simulated S_2, S_1 and S_0 for Lab 7. Simulating CLK is trickier though, because you need a crisp 01 transition. So far, we have assumed that when you activate a switch, this is what happens:



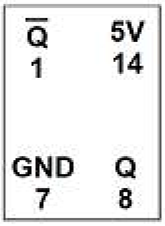
In real life, this is what actually happens:



This problem is known as contact bouncing, and makes fail any circuit that is edge-triggered, like the flip-flops you are using. To avoid that, we need a switch debouncing circuit:

Debouncing switch is in the ON (1) position (green light)	Debouncing switch is in the OFF (0) position (red light)
	

The debouncing switch only uses four pins:



Use the pin labeled as Q to simulate CLK, which will provide you with a crisp 01 transition.

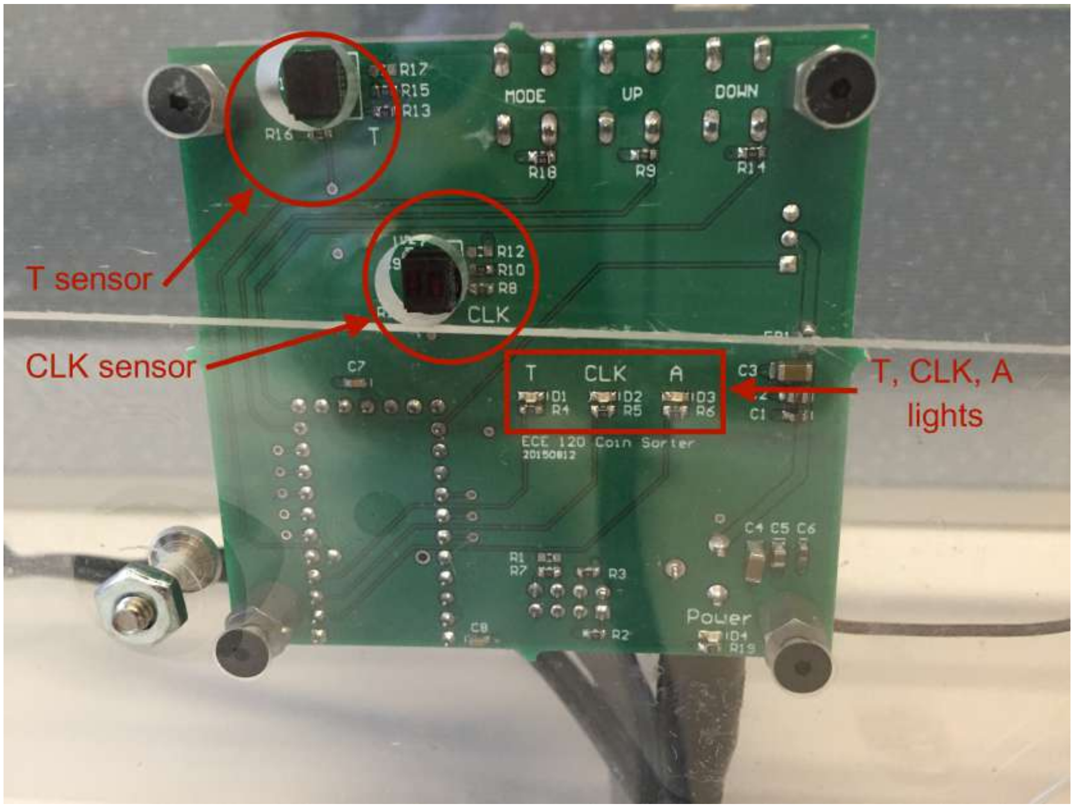
By simulating T and CLK, you can check and see whether your state machine is consistent, makes correct state transitions, and produces the correct outputs A, P, and catch some bugs that way.

If you have any issues with the debouncing switch, ask for help, but **in any case do NOT get rid of the debouncing switch; it can be repaired**.

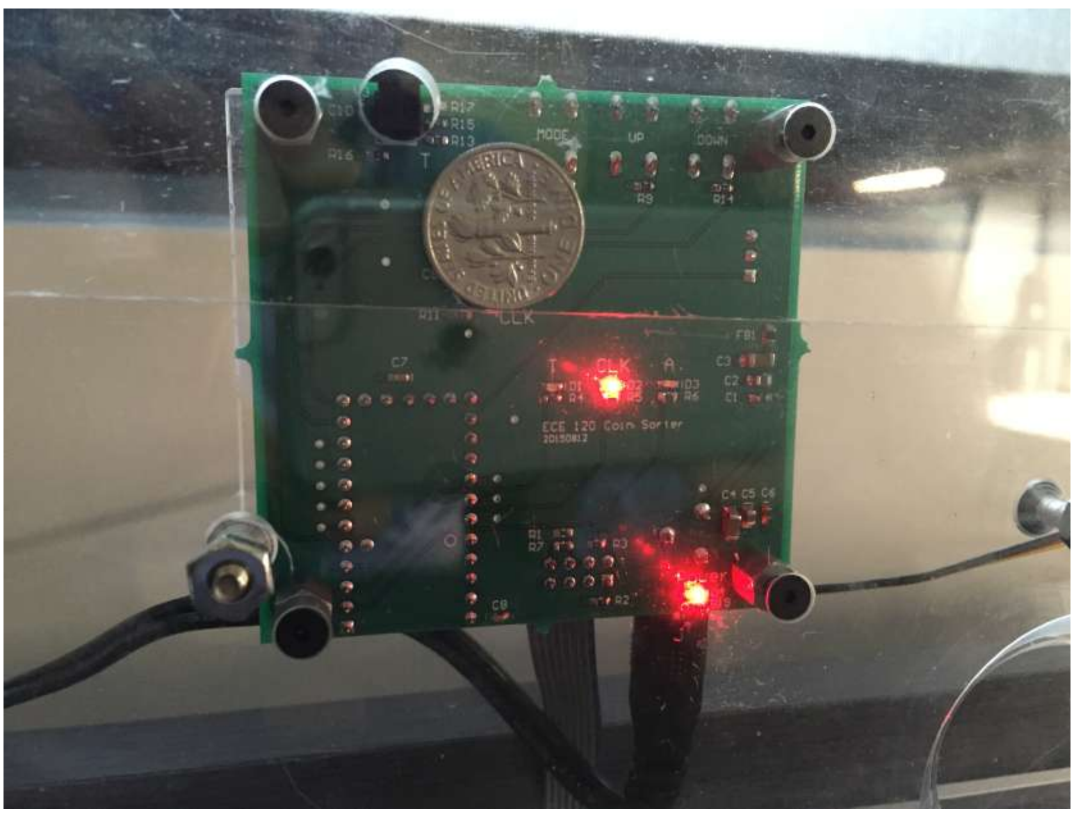
The interface to the vending machine

This is the hardware that we will use to simulate a vending machine:

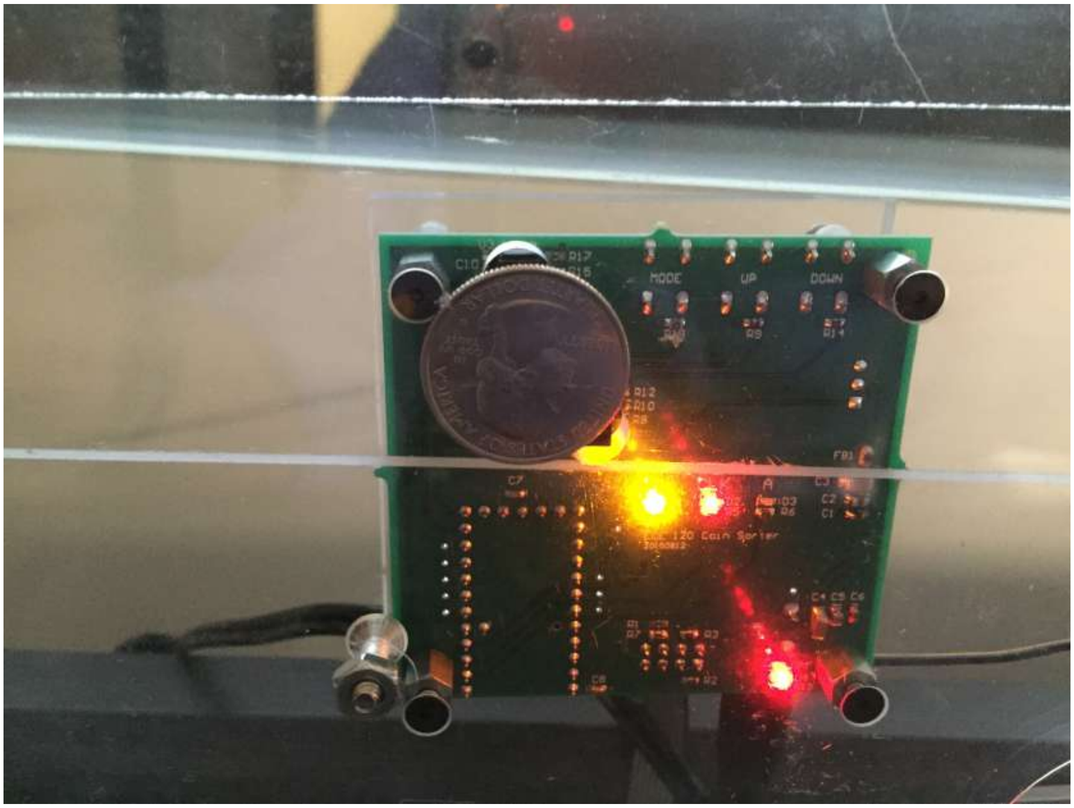
It has two optical sensors (T, CLK) and three LEDs (T, CLK, A) for visual feedback:



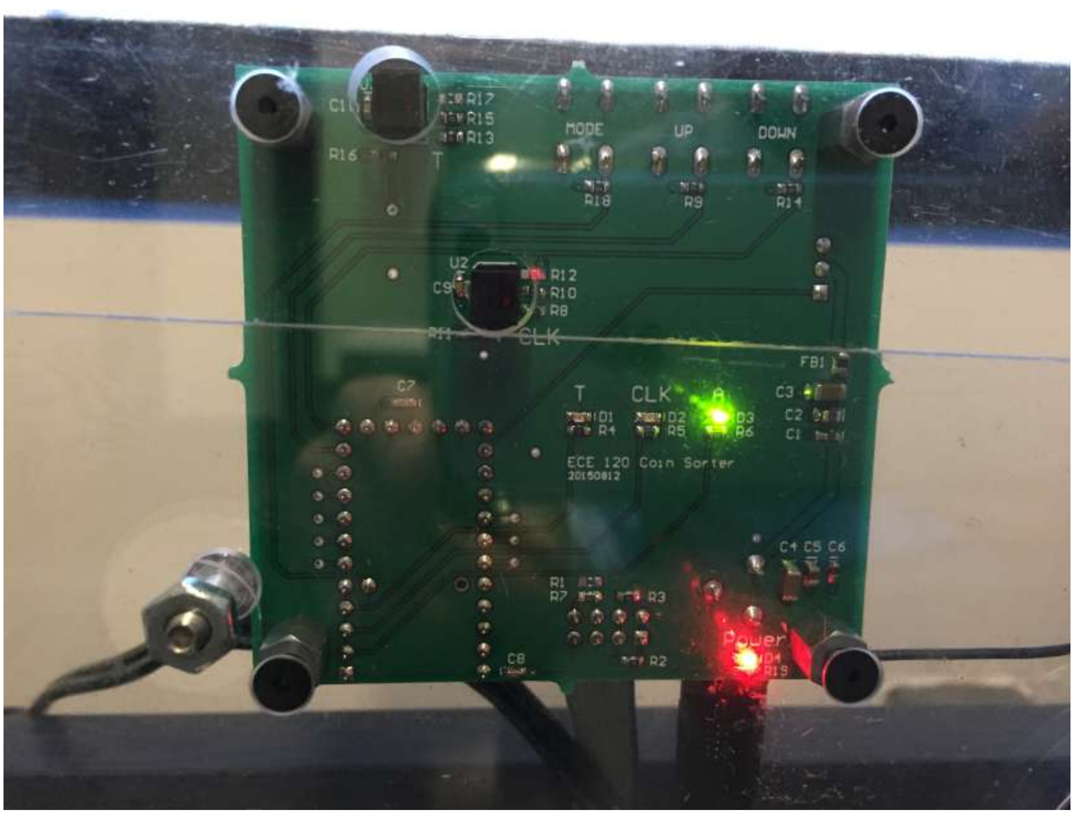
When a dime is passing, the CLK LED will light, to indicate the CLK sensor has detected a coin:



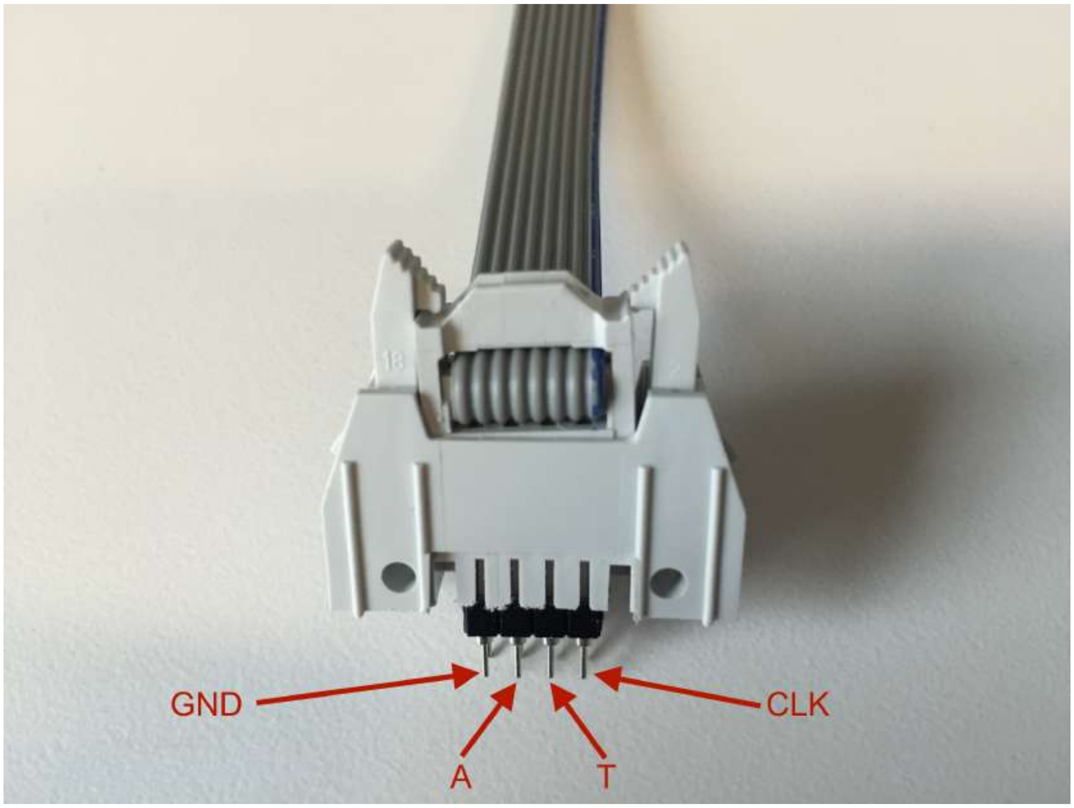
When a quarter is passing, the T, CLK LEDs will light, to indicate the T, CLK sensors have detected a coin:



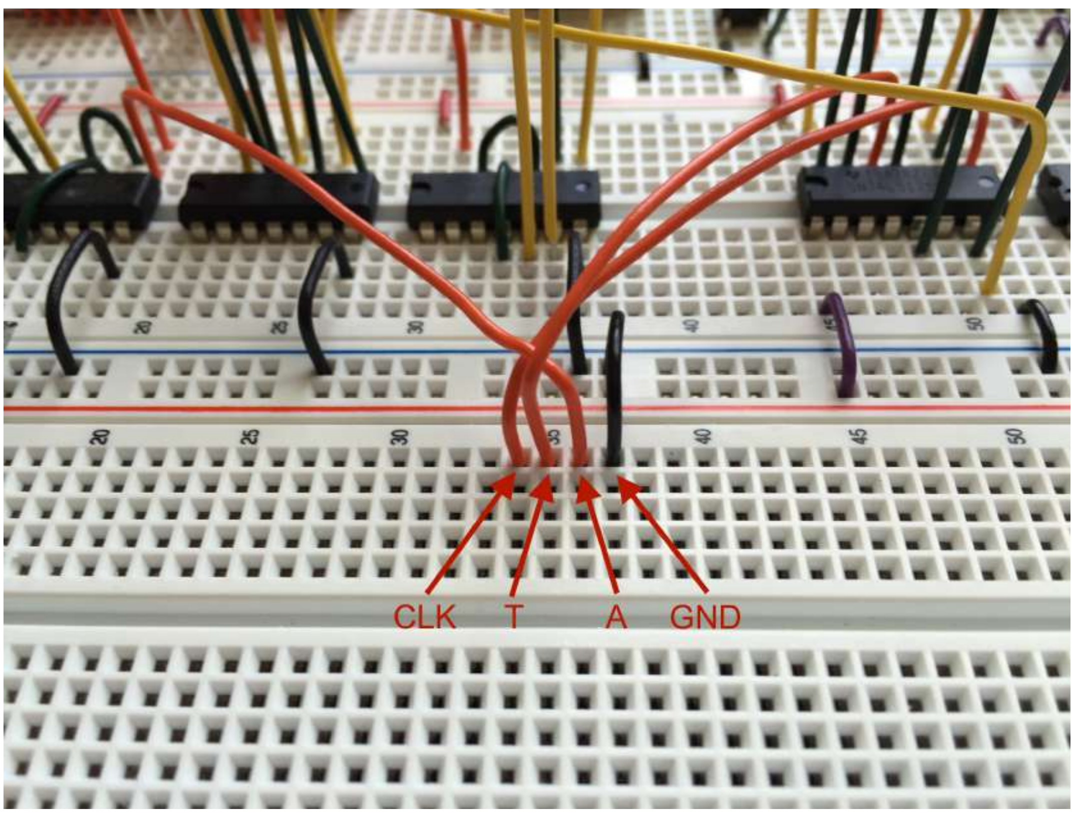
Similarly, when your circuit generates output A=1, the A LED will light, to indicate the signal has been correctly received:



The interface to the protoboard is given by a connector (note the **blue mark** on the right of the ribbon cable):

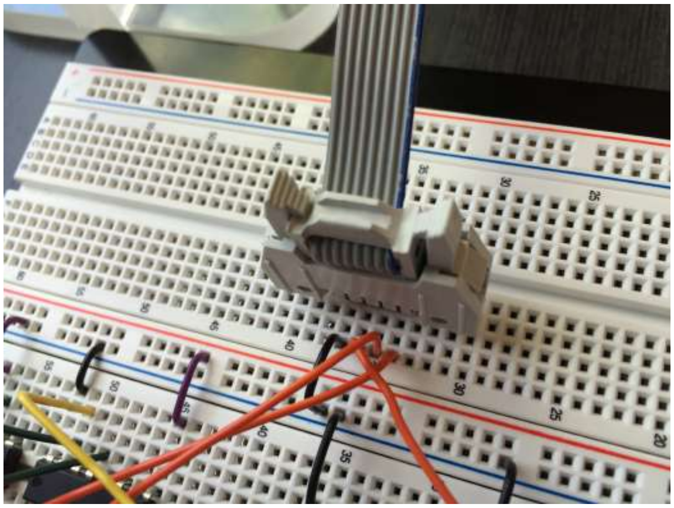


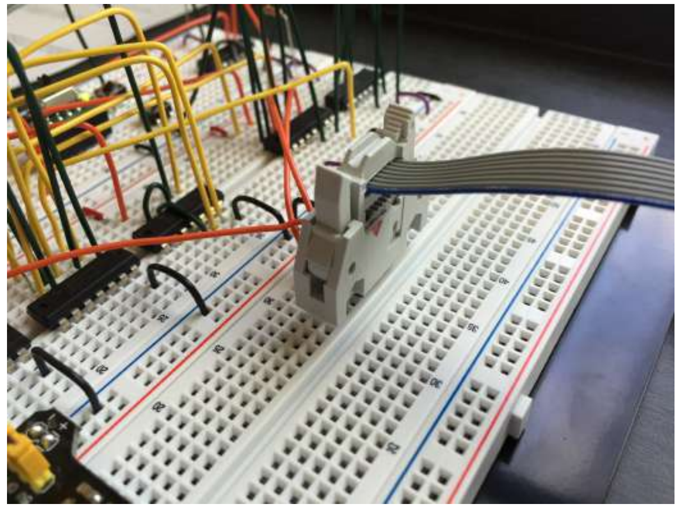
The connector has a total of 8 pins, but for simplicity in this lab, we are only using 4 signals: GND, A, T and CLK. In the protoboard, you will have to wire the required signals in preparation for the connection. Make sure to leave enough space on your protoboard for the interface to the vending machine:



Do not forget to first detach your T and CLK debugging signals, in order to prevent them driving against the vending machine and possibly damaging your circuit or the vending machine.

Insert the connector into the terminal strip so that it does **NOT** bridge the center channel, and keeping in mind the **blue mark** on the side of the ribbon cable to make sure signals are connected correctly:

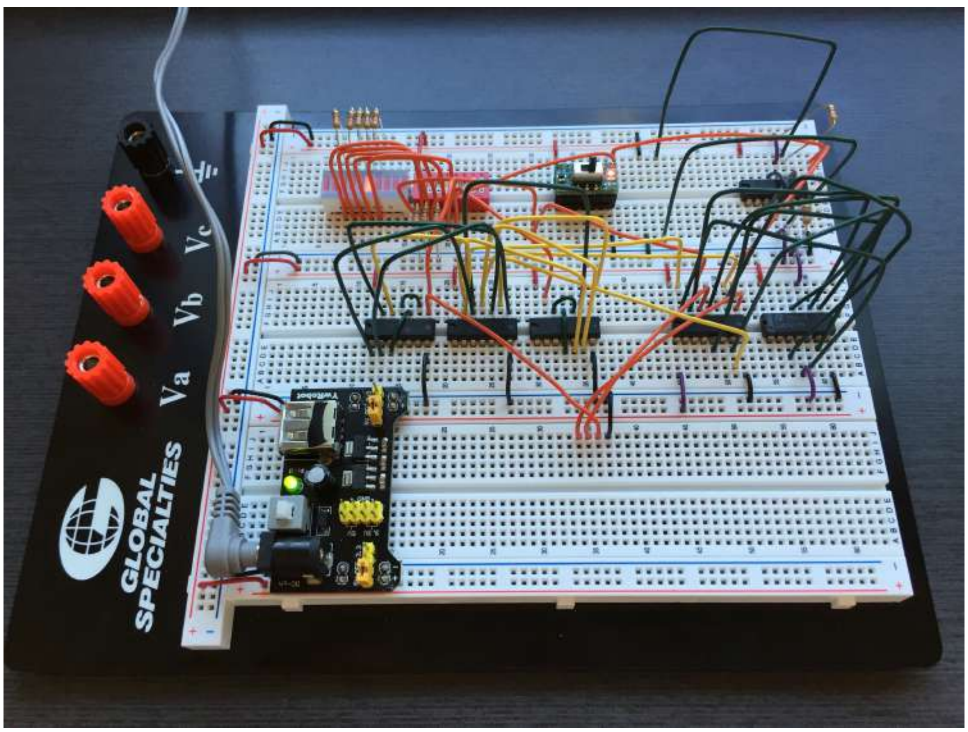




Note that the coin sorter device itself needs to be powered up with a 5V power supply. This is a different power supply; be careful not to use the protoboard's 9V power supply for powering up the coin sorter!

Some Useful Tips ...

Here is an example of a completed circuit. Notice the debugging circuit on the top left of the protoboard, and the debouncing switch in the top center of the breadboard. Also, the interface to the vending machine is in the bottom center of the protoboard, with plenty of space to allow for the connector.



- The circuit will be complicated. Making a rough sketch of your connections will speed up your debugging time.
- Try to keep the layout clean and neat so that you do not have a "short" between 2 wires.
- Make sure to leave some space for the interface to the vending machine.
- LEDs will be your best friend when debugging. Use them wisely!
- Test in phases! Make sure your FSM implementation is correct before you move on to testing the machine.

Refer to this video for tips for assembling and debugging this lab, courtesy of the ECE Student Advancement Committee.

What to submit

- Your protoboard layout sheet (same sheet as in Lab 8), including the circuit you implemented in Labs 8 and 10.