COS 116
The Computational Universe
Laboratory 7: Sequential and Synchronous Circuits

Last week you built combinational circuits out of logical gates. This week, you'll build simple sequential circuits and synchronous circuits on your breadboard. Recall from lecture that these use memory elements to maintain some *state* which changes over time. Recall also that the notion of "time" derives from a fluctuating voltage signal ("clock").

You will use *signal generators* to produce the 'clock' signal and *oscilloscopes* to get a graphical display of time-varying signals on your PC. Each oscilloscope and generator may need to be shared by two or more students (who will have separate breadboards).

If you get stuck at any point, feel free to discuss the problem with another student or a TA. However, you are not allowed to copy another student's answers.

Hand in your lab report at the beginning of lecture on Tuesday, April 18. Include responses to questions printed in bold. (Number them by Part and Step.) When you assemble the circuit for the traffic light, you need to show it to your TA. Otherwise you will not get credit for the lab.

The lab TA will walk you through the parts that you will need out of your plastic kit box.

- 1 set of power supply connectors
- 1 breadboard (with pre-wired switches and pre-wired chip)
- 1 set of breadboard jumper wires, multiple colors
- 3 LED (red, yellow, and green)
- 1 resistor
- 1 74LS04 chip (6 "Not" gates)
- 1 74LS08 chip (4 "And" gates)
- 1 74LS32 chip (4 "Or" gates)
- 1 74LS175 chip (4 D flip-flops)

Note: You may have noticed the addition a new chip on your breadboard besides the switch. This chip "cleans up" the output of Switch 4 to allow it to be used as a manual clock signal. Do not remove it or its wiring. Before you leave, remove any wires or components that you added to the breadboard and replace them in the plastic kit box, and return everything to the TA.

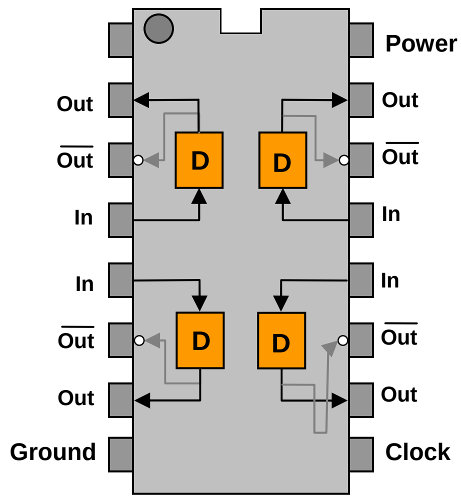
Connect the red terminal of the power supply to the red post on the breadboard, and the black terminal to the black post. Each power supply can feed multiple breadboards since the jacks can stack on top of each other. Switch the power supply on, and verify that its voltage reading is 5 volts. If not, contact the TA.

<u>Caution:</u> To avoid damaging chips, before you insert or remove any components from the breadboard, always disconnect the power by unplugging the red power supply connector from the breadboard's connector post.

Part 1: Understanding the D flip-flop using a manual "clock"

As you learned in lecture, the D ("data") flip-flop is a simple memory element. In this experiment you understand its properties using a manual "clock" you will provide yourself with a toggle switch. Each 74LS175 chip packages four D flip-flops:

74LS175 chip



4 D flip-flops

Notice the notch and dot at the top of the chip—you'll use these to orient the chip correctly on the breadboard. In lecture, we learned about the 'airlock' or master-slave version of the D flip-flop (see slides 19 – 28 of Lecture 12). This kind of flip-flop captures its input when the clock signal becomes 1 (the so-called *rising edge* of the clock) but changes the output only when the clock become 0 again (the *falling edge*). The D flip-flop in your kit does <u>not</u> have this airlock feature. Your flip-flop also captures the input on the rising edge, but makes it available on the output <u>immediately</u>. <u>The output can change on *rising* edges only, and at no other times</u>.

Connect the upper-right pin to power (the + column on the breadboard) and connect the lower-left pin to ground (the – column). The lower-right pin receives a clock signal which drives all four flip-flops. Each flip-flop has a single input and two outputs; one output is the regular output value and the other, written with an overbar in the diagram above, is the opposite (i.e., negation) of the output value.

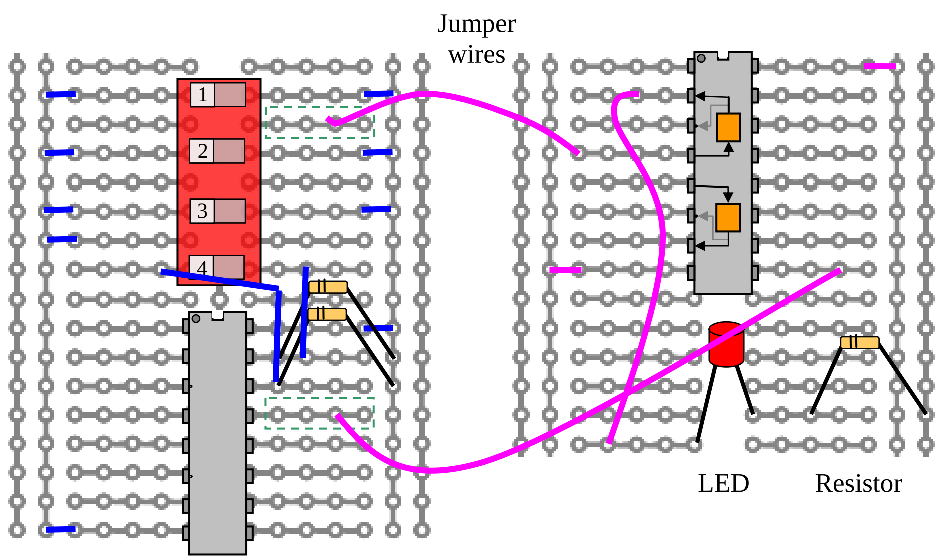
1. **Disconnect power by unplugging the red power supply wire from the breadboard.**

2. **Insert the LED, resistor, and 74LS175 (D flip-flop) chip into the right half of the breadboard, as shown in the picture below. Make sure the U-shaped notch in the chip is on the top, and make sure the longer lead of the LED is on the left.** <u>The chip may be damaged if you insert it upside down or wire it incorrectly.</u>

    a. **Connect the output of Switch 1 to the input of one of the D flip-flops on the chip.**

    b. **The manual clock is implemented by toggling Switch 4, but this output is passed through another chip (74LS279), which was pre-wired into your breadboard. (See the questions at the end for the reason for this extra chip.) Connect the output of the 74LS279 as shown below to the clock input on the D flip-flop.**

    c. **Connect the output of the D flip-flop to the LED.**

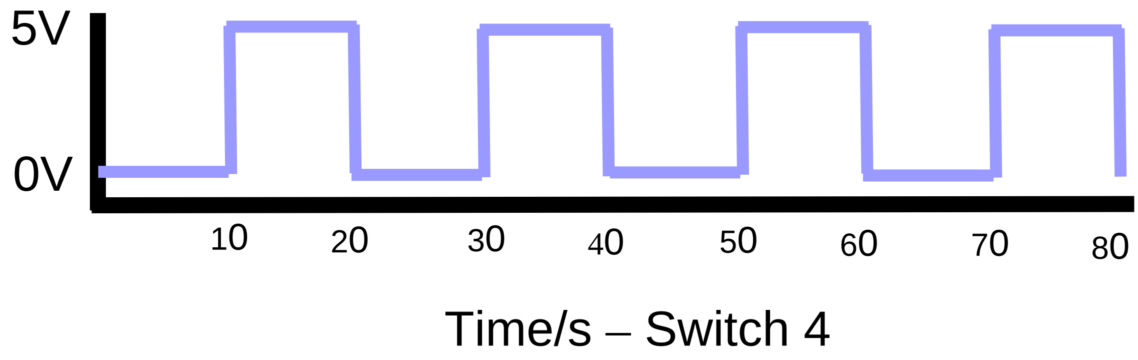


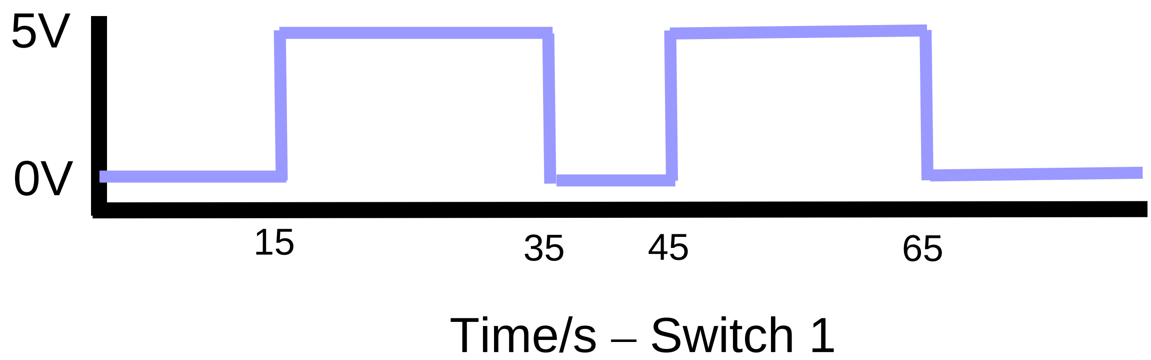
3. **Reconnect the power supply to the board.**

4. **Test your circuit. Provide a clock signal by flipping Switch 4 back and forth. Each time you turn Switch 4 on, the flip-flop updates its output. The LED will light up depending upon whether or not Switch 1 (connected to the flip-flop's input) was ON at the instant of the rising edge of the clock signal.**

Now we will study *timing diagrams*, which describe the behavior of sequential circuits. The diagram below shows two time-varying signals (waveforms). Your goal is to figure out how to toggle Switch 1 and Switch 4 so as to enter these two waveforms into the D flip-flop. Keep track of time using either your wristwatch or the clock on your PC. Very precise timing is unimportant so long as you get the order of switch flips right. In your lab report mention in what sequence you toggled

the two switches, and at what times the LED turned on and off. Infer the output waveform using these observations, sketch it in your report and explain it in terms of the properties of the D flip-flop. (If you ever lose track of where you are in the signal, unplug power from the circuit and start over. Both plots begin with the voltage 0, which means that both switches are at the zero position.) You should sketch the output diagram and understand it completely before moving on to the next experiment.





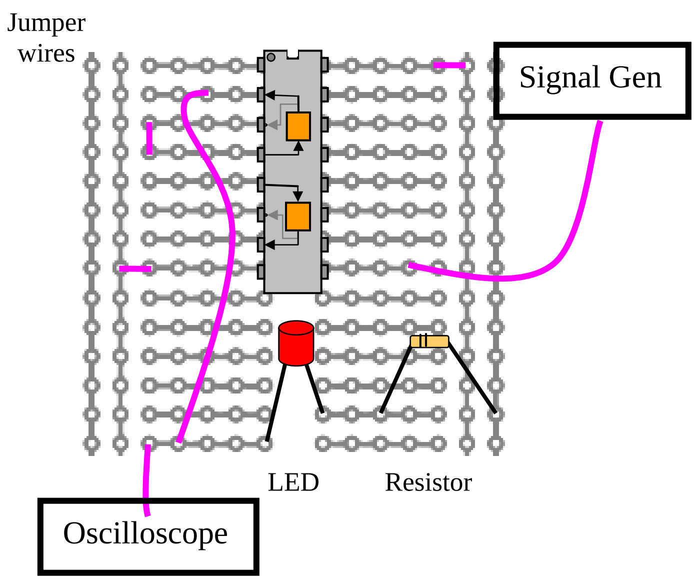
In your report, also describe what the output waveform would be if we had used the 'airlock' version of the D flip-flop? Recall that, in an 'airlock' flip-flop, the input is captured on the rising edge, but is only available on the output on the falling edge.

Part 2: Using a signal generator

In real applications, of course, flip-flops are driven by a clock signal. We use a signal generator to generate our clock waveform. You will also see a simple example of a sequential circuit that uses its own output as an input. You should pair up with a neighbor for this experiment.

1. **Disconnect power by unplugging the red power supply wire from the breadboard. Remove the wire going from the flip-flop to the LED, and make sure Switch 4 is turned off. Also, remove the wire connecting the output of Switch 4 to the Clock input of the D flip-flops.**

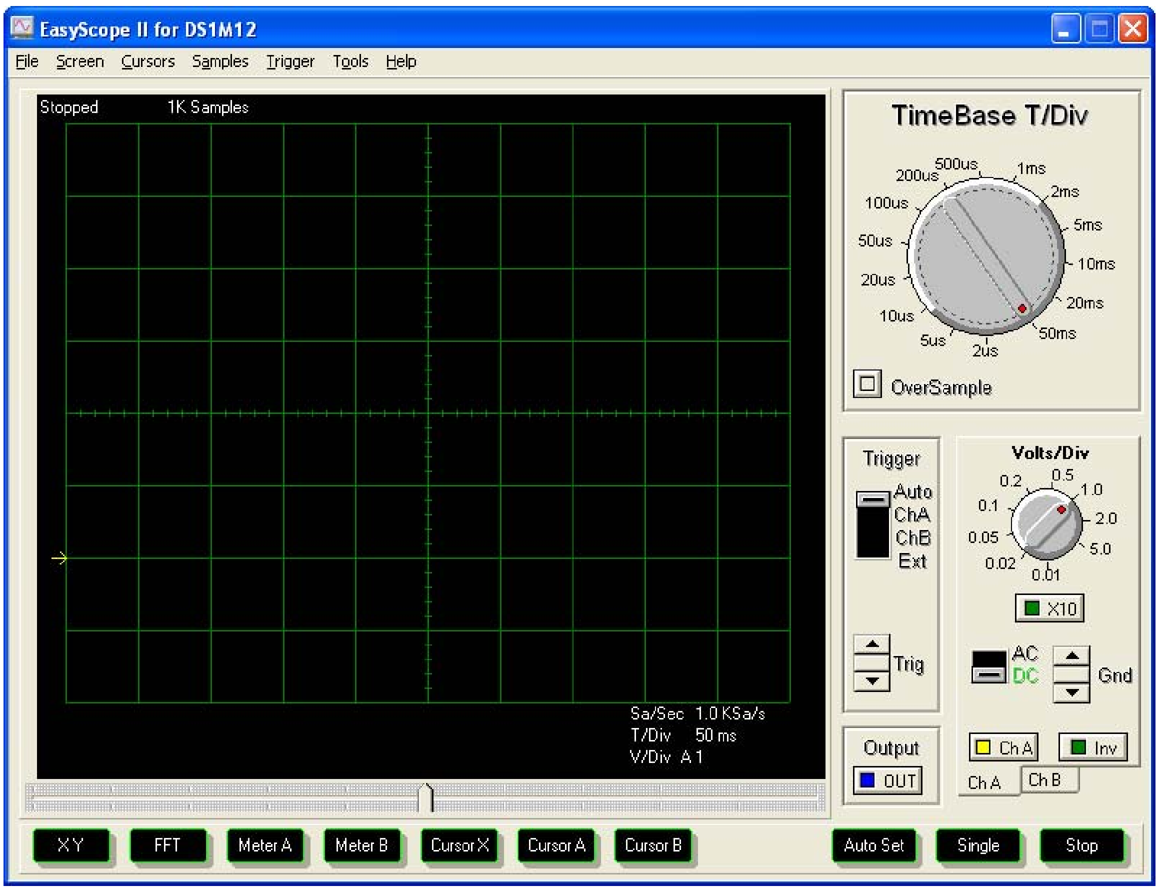
2. **Get a signal generator from your TA. Make sure it is powered on. Turn the wheels on it to set it to output a 1 Hz signal (1 clock cycle per second). The signal generators have probes coming out of them with a red and a black "crocodile" clip. Attach a little wire to each crocodile clip and insert the wire from the black clip to some blue column on the breadboard.**
3. **Connect the red crocodile clip to the input of the LED.**
4. **Using a wristwatch, count the number of times the LED turns on/off in 10 seconds and use this to infer the frequency of the waveform that is being fed into the LED. In your report write down this frequency.**
5. **Disconnect the signal generator's red crocodile clip from the LED. Replace the wire you had taken out in step 1.**
6. **Remove the wire connecting Switch 1 to the D flip-flop and use it to instead connect pin 3 and 4 of the 74LS175 IC. This connects the input of the flip-flop to the opposite (or negation) of its output. Your circuit should look like the figure below except for the oscilloscope, which will be added in the next experiment.**
7. **Now connect the red crocodile clip from the signal generator to the CLOCK pin of the D flip-flop. Observe the LED and again estimate the frequency of the waveform being applied to it. In your report mention this frequency and explain it in a line or two.**



Part 3: Observing fast voltage signals using the oscilloscope

Now we will use a digital oscilloscope to visualize these waveforms on a PC. Since, most real sequential circuits, like a computer's control unit, operate at very high frequency clock signals we want to test our circuit with such a clock signal. In this experiment, we will ‘run’ the previous circuit at higher clock frequencies. The output frequency will consequently be very high as well, and so an LED cannot be used to visualize it. Hence we will use an oscilloscope to visualize the output.

1. **Make sure the oscilloscope is connected to the computer. Next open the oscilloscope program ‘EasyScope II for DS1M12’. Set the TimeBase T/Div dial to 50ms. This sets the horizontal axis in the oscilloscope display to 50 ms per major division. Also, set the Volts/Div Dial to 1.0. This sets the scale of the vertical axis to 1 V per major division. Next drag the vertical offset, the small yellow arrow to the left of the graph, as shown below. This makes 0 V appear at that height. Press Run. The screen should look like the one below.**



2. **The oscilloscope has a probe similar to the one that the signal generator had. Connect the black crocodile clip to ‘ground’ (any blue column). Connect the red crocodile clip to the input of the LED in the circuit you built for experiment 2. Note that this is the output of the D flip-flop. (Make sure not to remove any connections made in part 2. Use an additional piece of wire to connect the input of the LED to the red clip as in the figure above.)**

3. **Now increase the frequency in the signal generator from 1 Hz to 20 Hz.**

4. **Sketch the waveform in your write-up. What is the amplitude and period of the output signal? Recall that the period of a signal is the length of time it takes to complete one cycle. You can play around with higher frequencies. You will have to adjust the TimeBase dial appropriately to visualize the resulting waveform.**

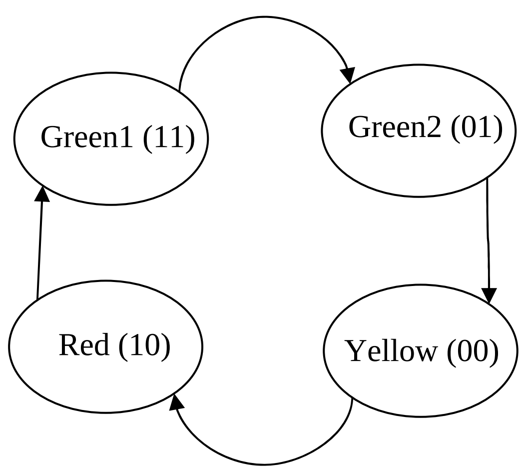
Part 4: Building Synchronous Sequential Circuits

In this experiment you implement a simple synchronous circuit for a simple "traffic light." The light has to behave as follows over and over: green for 2 clock cycles, yellow for 1 clock cycle, and red for 1 clock cycle. This behavior can be realized using a finite state machine with four states. We need two bits for representing 4 states. Since each flip flop can store 1 bit, you will need to use 2 flip-flops for this experiment. Call the flip flops 1 and 2.

We recommend the following state assignments:

Green1 = 11
Green2 = 01
Yellow = 00
Red = 10

The FSM cycles through these states. See the transition diagram below. Notice that this FSM has no inputs (i.e. the transition arrows are not labeled).



1. **Fill in the state transition table of this FSM. The first row is filled out for you. CurrentState1 and CurrentState2 are the two bits of the current state. NextState1 and NextState2 are the two bits of the next state. The output bits are the signals that turn the three LEDs on/off.**

CurrentState1	**CurrentState2**	**NextState1**	**NextState2**	**Green LED**	**Yellow LED**	**Red LED**
1	**1**	**0**	**1**	**1**	**0**	**0**
0	**1**					
0	**0**					
1	**0**					

2. **Write a simple synchronous circuit to implement this FSM. Note that the D flip-flops give you both the output and its negation, so you don't need to use NOT gates.**

3. **Figure out what the LEDs should be connected to? If you get stuck, ask the TA. [Hints: The green LED should be connected to the output of flip flop 2. You may need an AND chip for the red and yellow LEDs.]**

4. **Wire up the circuit. First put in the 3 LEDs with appropriate resistors. Next connect the flip flops to each other as you decided in step 2. Next put in the AND chip, connect the power and ground pins appropriately. Then based on your answer in step 3, connect the output of the flip flops to the LEDs 'through' the AND gates.**

5. **Connect the clock inputs of the D flip-flops to Switch 4. Verify that your circuit works by toggling the switch 12 times.**

6. **Disconnect the clock from Switch 4 and connect it to the signal generator. Change the frequency of the signal generator to about 1 Hz and watch your traffic light work, all by itself.**

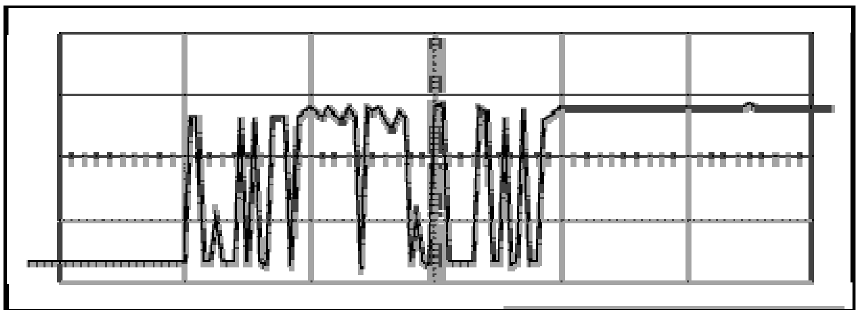
7. **Show your working circuit to your TA to get credit for finishing the lab.**

Further Questions to answer in your report

1. **Suppose we want a different traffic light behavior: red for 3 clock cycles, yellow for 1 clock cycles and green for 2 clock cycles. Sketch a finite state machine that corresponds to this behavior. How many flip flops are needed to implement it? Briefly describe how you would implement it, and include the truth table and the Boolean circuit needed. (You do not need to describe the implementation of this Boolean circuit on the breadboard.)**

2. **Use web search to get an estimate of the number of atoms in the universe. Report this number. Approximately how many flip flops would you need to implement a finite state machine in which the number of states exceeds this number?**

3. **This question explores the need for the extra chip that your breadboard came with today. Its purpose is to "debounce" the switch output. Our switches use mechanical contacts and a spring, so when you flip a switch, the output oscillates (bounces) a bit before settling down. When you connect the output of the switch to an oscilloscope, you will see something like the following. [The details of the oscillation are different for different types of switches.] Explain why were the oscillations not a problem with combinational circuits used in the last lab, but a major problem with sequential circuits used today?**



4. **(Extra credit) The extra chip is an R-S latch which also appeared in the lecture on Mar 16. Examine the chip debouncing circuit and explain how it works.**