CPSC 121: Models of Computation
Lab #5: Flip-Flops and Frequency Division

Objectives

In this lab, you will be introduced to two types of circuits called *latches* and *flip-flops*. Latches and flip-flops can be used to store information; this is useful when a system's current output depends on one or more of its previous outputs. You will also be working with different clock frequencies and using them to make 2 bit and 3 bit counters.

1 Pre-lab

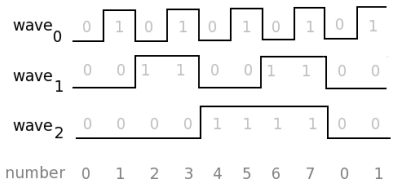
1.1 Counting

Notice that there is a parallel between binary numbers and truth tables, and a pattern can be seen in both:

0	0	0
0	0	1
0	1	0
0	1	1
...	...	...
1	1	1

The right-most column repeats 01010101, the one immediately to the left repeats 00110011, the next one repeats 00001111, and so on. In circuits, if we want to repeat a pattern like this, we use an electrical wave like the clock output on the Magic Box, which regularly alternates between 0 and 1 at a set frequency.

Using the three waves in the figure below, we can produce the same pattern and repeatedly count up to 7:



What differs between the three waves is their *frequency*: wave_0 is twice as frequent as wave_1, which is twice as frequent as wave_2. If we want to be able to count up to 15, all we need to do is add another wave, which is half as frequent as wave_2. Have a look at this animation, which has such a wave. (Recall that 1 Hz means once every second; 2 Hz means twice every second, etc.)

TODO (pre-lab): If we wanted the animation to count to 31, what frequency (in Hz) would wave_4 have?

1.2 Sequential circuits

Here's another way of thinking about counting. Consider the mathematical formula:

$$t_n = t_{n-1} + 1$$

where $t_0 = 0$. Here, we're taking the previous value of t and using it to get the next one. For example,

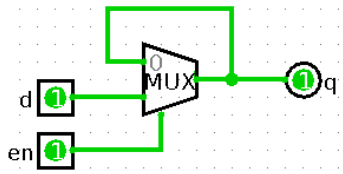
$$t_1 = t_0 + 1 = 0 + 1 = 1$$
$$t_2 = t_1 + 1 = 1 + 1 = 2$$

Notice how the value of t_n depends on t_{n-1}, the value that comes before it. This type of behavior can also be seen in circuits. When we store and use the previous output of a system, we call it a **sequential circuit**.

A latch is a type of sequential circuit, which works like this:

- If en $= 0$, the latch's output (q) is its previous output (q_{prev})
- If en $= 1$, the latch's output (q) is its input d

The latch allows us to remember a value; it is a simple form of memory. It looks like this:



The truth table for this circuit differs from ones you've seen before: now we have a variable, q_{prev}, representing the latch's previous q value:

en	d	q
0	0	q_{prev}
0	1	q_{prev}
1	0	0
1	1	1

TODO (pre-lab): Answer the following questions:

1. Suppose we turned on the latch with en=1 and d=0. What would be the value of the latch's output q?

2. Now suppose we, after doing that step above, switched en=0. What would be the value of the latch's output q?

3. And now, after that step, suppose we switched d=1. What would be the value of the latch's output q?

4. Now after that step, suppose we switched en=1. What would be the value of the latch's output q?

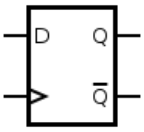
5. Lastly, after that, suppose we switched en=0. What would be the value of the latch's output q?

1.3 Flip-Flops

A flip-flop is another type of sequential circuit and is very similar to a latch. However, *when* the flip-flop will load in a new value is different. With the latch, we took in a new value whenever en was on – so if we left en on, and flipped d on and off, we would see q go on and off in response. Flip-flops only take in a new value when en ***becomes*** **on.**

Once en is on, nothing changes: we have to turn en off and then on again to load in a new value. Because of this, en is typically connected to a clock wave – this allows it to regularly load in new values. A new value is loaded every time the clock wave becomes on; otherwise we hold onto the old value.

We represent flip-flops with this symbol:



In the flip-flop, D is our input, Q is the value we're remembering and $>$ represents the clock input, replacing the en of the latch. **TODO (pre-lab): What is the $\overline{Q}$ output? Look this up and cite your source.** (If you are searching on the internet, it will help to look for a "D Flip-Flop").

2 A Flip-Flop

Connect up a 74LS175 Quad D Flip-Flop chip to power and ground on a Magic Box. Then, power the $\overline{Clear}$ input in the lower left corner of the chip. $\overline{Clear}$ can be read as "not *Clear*" (The complement or inversion of a signal is typically written with a line over top). For one of the four flip-flops in the circuit, connect 1D to a switch, 1Q and $1\overline{Q}$ to LEDs, and CLOCK to a connection in the clock output box.

Figure 1: The clock output box of a Magic Box

Unlike in Lab 1 where we used a clock wave that was generated by the *Magic Box*, here we want to use a manual clock – there is a white button above the clock output that you can press (it's at the top in the figure above). Each time you press the button, a new clock cycle begins (the clock begins outputting a value of 1); when you release it, the clock cycle is complete (the output of the clock again becomes 0). **For how to switch to manual clocking, see page 6 of the** Magic Box User's Manual

TODO: Determine a truth table for this circuit. For some entries, you may want to define and use variables like q_{prev}, or describe what happens in English.

D	Clock button	Q	$\overline{Q}$
0	is pressed		
1	is pressed		
0	is not pressed		
1	is not pressed		

3 Shift Registers

Flip flops can be connected together so that their states can be passed down a chain. This is called a *shift register*. Using your D Flip-Flop, connect 1Q to 2D, and 2Q to 3D, to connect three flip-flops in a chain. Also connect 1Q, 2Q and 3Q to green, yellow and red LEDs, respectively. Keep the clock connected to the clock button, and 1D connected to a switch. Figure 2 shows a diagram of the circuit you are implementing.

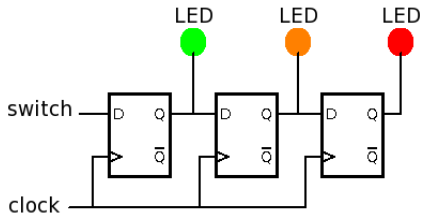


Figure 2: Three flip-flops connected in series.

TODO: Test your circuit to see what happens. Play with different values for 1D and see what happens as you clock the circuit manually. What happens?

4 Wiring a Counter

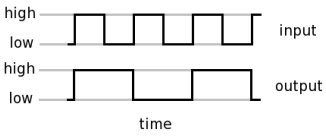
In this section, you will wire up a 2-bit counter with two Magic Boxes:

1. Find another team that has a Magic Box.
2. Turn off the power to the *Magic Box*es while wiring them together.
3. Leaving your original power and ground wires still connected, connect the ground input of one Magic Box to the ground input of the other one. This gives the two circuits a common reference for voltages.
   Do NOT connect the power signals together. Each Magic Box has its own voltage regulator that converts 9 volts from the AC power adapter to 5 volts for the breadboards. While the output voltages are very close to 5 volts, there will be slight differences between the boards. If you connect the outputs of two voltage regulators together, they can fight over what the right output voltage is. This violates the specification for how the regulators are supposed to be used and could lead to a malfunction.
4. One team should set their Magic Box's clock frequency to 1 Hz, and the other team should set theirs to 2 Hz. See page 6 of the Magic Box User's Manual for how this is done.
5. Now connect the 2 Hz clock wave to the rightmost LED on your board (it is orange and labeled B0).
6. Connect the 1 Hz clock wave to the LED beside it (it is green and labeled B1).
7. Turn on your Magic Boxes and look at the number display. It should be counting 0-1-2-3 over and over again.

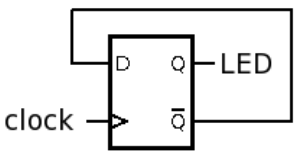
TODO: Once you have it counting to 3, show it to your TA.

5 Frequency Division and Counting

Our Magic Boxes only have one clock wave generator each: if we want to make a counter with one Magic Box, we need to be able to get multiple clock wave frequencies out of one generator. We accomplish this by making a *frequency divider*. It is a circuit which takes in a wave, and outputs a wave that is half as frequent:

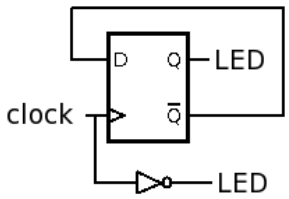


We can make a frequency divider with one flip-flop, like so[1]:



[1] Thanks to the physics of these integrated circuits, no D input is needed to get the circuit to work.

If you connect both your clock signal, and the output of your frequency divider, to your number display, you should get it counting 3-2-1-0 repeatedly. Check to see that's what happens. To get your circuit to count upwards, invert your original clock signal:



You should be seeing it count 0-1-2-3 repeatedly – we call this a 2-bit counter. **TODO: Show your 2-bit counter to your TA.**

Now, build a second frequency divider with a different LS175 chip. Verify that it works. You can now build a circuit that repeatedly counts up to 7, using both of your frequency dividers. You might want to figure out how you want to connect the two together on paper beforehand, since you will have to figure out which inputs and outputs go together! If you get stuck, ask your TAs for help!

TODO: Show your 3-bit counter to your TA once you have it working.

6 Further Analysis Questions

TODO (further analysis): Because sequential circuits take their outputs as an input, they typically start off having an *undetermined* state. When does a flip-flop have a determined state and what could some problems of having an undetermined state be?

7 End of Lab Survey

TODO: To help us improve these labs both this term and for future offerings, complete the survey at http://www.tinyurl.com/cs121labs.

8 Magic Box cleanup

TODO: Before leaving the lab, show your Magic Box to your TA.

A Challenge Problem

Serial communication is the process of sending data one bit at a time, while parallel communication is capable of sending multiple bits simultaneously. A simple illustration can clarify the difference. In this lab, we showed you a *serial in, parallel out* (SIPO) shift register:

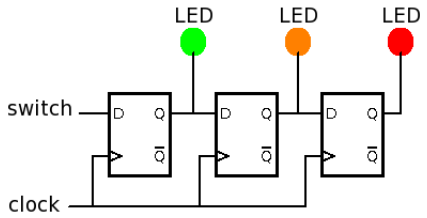
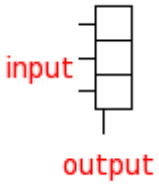


Figure 3: Three flip-flops connected in series, with parallel outputs.

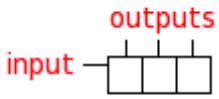
As there are three bits of memory in this shift register (one in each flip-flop), we call this a 3-bit SIPO shift register. Shift registers are used for the storage or transfer of data, and can be used to convert data either from a serial to parallel format, or vice versa. In a SIPO shift register, data is loaded in serially: one bit at a time. This stored data is then available in parallel form: all bits in the register can be read simultaneously.

A different type of shift register is a *parallel in, serial out* (PISO) shift register. In a PISO shift register, data bits are loaded in simultaneously, and then the stored data can be shifted out of the register and read serially, one bit at a time.

Shorthand for the PISO shift register is illustrated on the left below, and for the SIPO shift register, on the right:

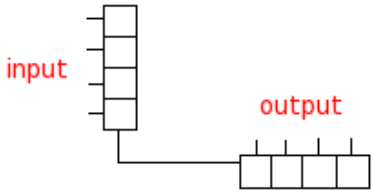


PISO shift register

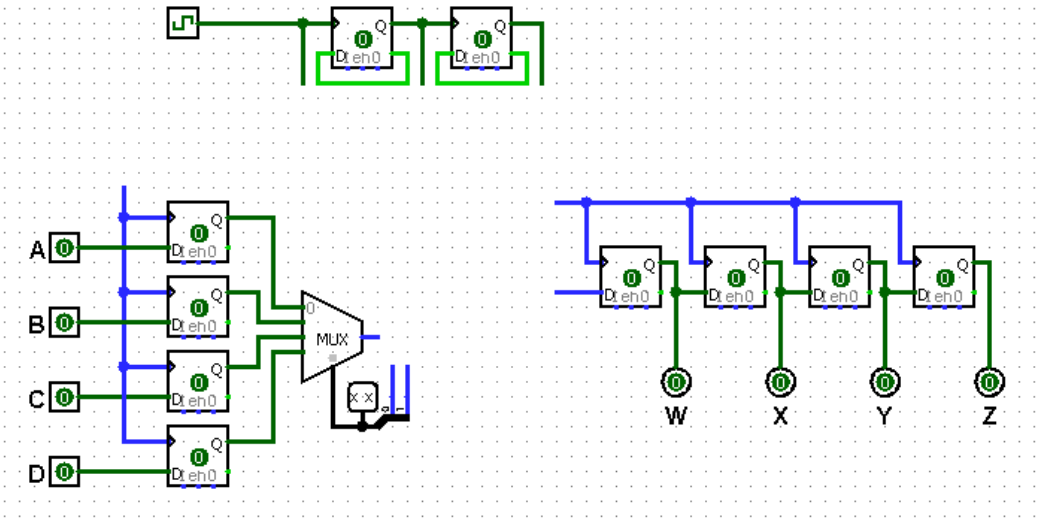


SIPO shift register

Consider the following problem: A student tries to make a 4-bit parallel-to-serial-to-parallel circuit, with this in mind:



But the student gets stuck on how to clock the circuit, after coming up with the following:



After precisely 4 clock cycles have completed, and before the 5th clock cycle begins, the student wants W to output the value of A, X to output the value of B, Y to output the value of C and Z to output the value of D. In Logisim, connect up the dangling wires so that the circuit works as intended. To get the bonus mark, you must be able to explain what each part of the circuit above is doing as you clock it.

Download the file `piso_sipo.circ` from the course website, and load it in Logisim. It might be helpful to set initial values for A, B, C and D (for example, A=1, B=0, C=0, D=1), and track how the values are passed along the circuit. Note that in Logisim, the D input for a flip-flop is on the lower-left, and the clock input is on the upper-left. **To reset the circuit, press Ctrl + R.**

B Marking scheme

All labs are out of ten marks. Two marks for pre-labs, and eight marks are for in-lab work:

- Two marks - Pre-lab questions
- Five marks - In this lab, it is 1 point for Section 2 (flip flop), 1 point for Section 3 (shift registers), 1 point for Section 4 (wiring a counter) and 2 points for Section 5 (frequency division/counters).
- One mark - Further analysis.
- One mark - End of lab survey.
- One mark - Magic Box cleanup.

TAs may at their discretion award one bonus mark, such as for completing a challenge problem.