

Sequential Circuits

This week, we want you to use Logisim to construct designs for two simple sequential circuits. The first will implement Euclid’s algorithm to compute the greatest common divisor of two numbers. The second will function as a hyperactive digital clock.

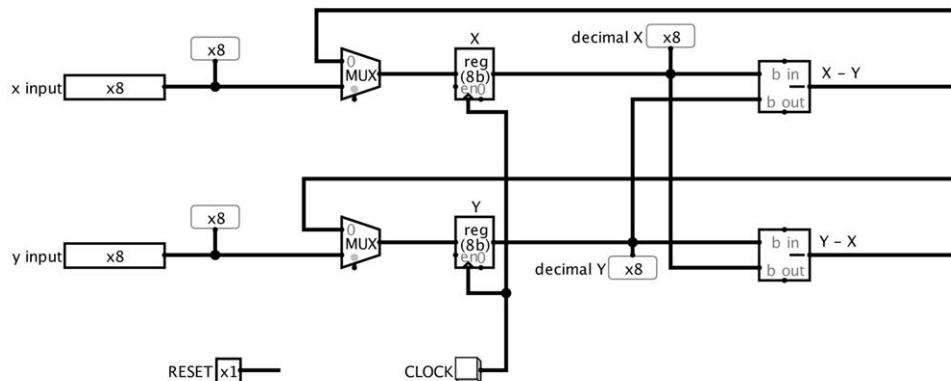
The Greatest Common Divisor

In mathematics, the greatest common divisor (gcd), also known as the greatest common factor (gcf), or highest common factor (hcf), of two or more non-zero integers, is the largest positive integer that divides the numbers without a remainder. For example, the GCD of 8 and 12 is 4. (From Wikipedia)

Euclid’s Algorithm

```
int gcd( int x, int y ) {
    while ( x != y ) {
        if ( x < y ) {
            y = y - x;
        } else {
            x = x - y;
        }
    }
    return x;
}
```

We want you to complete a circuit that implements Euclid’s algorithm for positive integers. To make your lives (and the grader’s) easier, we are going to give you a little help. We will provide a circuit designed to provide the necessary data path components for this algorithm. Our circuit is shown below. All you need to do is to add the necessary control circuitry.



The 8-bit pins at the left side of the diagram are intended to provide a means to enter initial values for x and y . Since Logisim pins only display values in binary, we have placed an unsigned decimal probe next to each input pin so that you can immediately verify that you entered the desired values in binary. We have put similar probes next to the registers that hold the intermediate values of X and Y as the algorithm executes.

Once input values are available in the pins on the left, pressing the reset button should initialize the registers labeled X and Y with the values in the pins. After that, repeatedly toggling the clock button should cause the circuit to execute Euclid’s algorithm. Once the GCD has been found (i.e. the values in the registers are equal), additional clock ticks should cause no further changes.

A Digital Clock

As a second exercise, we want you to construct a circuit that will cause 4 Logisim “Hex Digit Displays” to display values that cycle from 1:00 to 12:50 and back again as the digits on a digital

clock.

To guide (or constrain) your design a bit we want you to first design a sub-circuit that should aid you in constructing your clock. We will call this circuit a “bounded counter”. It will contain a 4-bit register which will function as a counter that automatically cycles between 0 and a specified bound.

Your bounded counter should have 3 input signals and 2 outputs:

Clock (input) The counter will increment (or reset to 0) on the leading edge of each clock pulse.

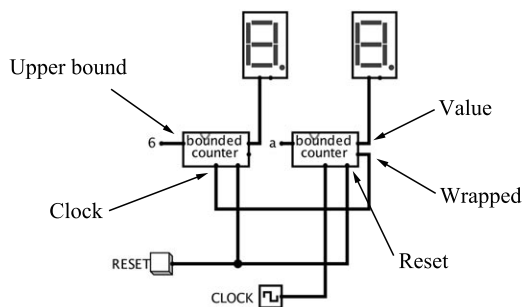
Upper Bound (input) A 4-bit bus interpreted as an unsigned binary number. The value of the counter will reset to 0 if incrementing it would make it equal to this number.

reset (input) When the reset signal is 1, the counter should reset to 0 independent of the clock.

Value (output) A 4-bit bus holding the current value of the counter.

Wrapped (output) A single bit which should be set to 1 only when the counter is displaying 0 because it just reset upon reaching its upper bound.

To give you a good start, the diagram below shows how bounded counters could be used to implement the two “minutes” digits of a digital clock face. Your job (and the challenge) is to implement the bounded counter sub-circuit and the two digits that display the hour. Remember your clock should cycle from 1:00 to 12:59!



Submitting Your Work

As we did for lab 3, we want you to submit both the .circ file for your work on this lab and a file containing drawings of all of your circuits so that we can print these drawings easily. To produce this drawing file:

- Select the “Print” item from the Logisim File menu. A dialog box should appear.
- Shift click or drag to select the names of all of your circuits in the list in this window.
- Edit the “Header” field in the window by adding your name. Click OK. A new dialog box should appear. Make sure the Print to File check box is selected. Then click Print.
- Use the next dialog box to navigate into the folder where your .circ file is saved and save the file containing drawings of your circuits as out.ps in this folder.

Within a terminal window, use the cd command to make the directory containing your .circ and out.ps files your current working directory. Then type the command:

```
turnin -c 237 out.ps Lab5Circuits.circ
```

Respond to the prompts appropriately and your work should be submitted.